

**TILT RIDERS:
CREATING AUTHORIAL-AGENTS FOR INTERACTIVE
STORYTELLING**

ANTÓNIO PIMENTEL BRISSON LOPES

SUPERVISOR: DOCTOR ANA MARIA SEVERINO DE ALMEIDA E PAIVA

THESIS APPROVED IN PUBLIC SESSION TO OBTAIN THE PhD DEGREE IN
INFORMATION SYSTEMS AND COMPUTER ENGINEERING

JURY FINAL CLASSIFICATION: PASS WITH MERIT

JURY

CHAIRPERSON: CHAIRMAN OF THE IST SCIENTIFIC BOARD

MEMBERS OF THE COMMITTEE:

DOCTOR ANA MARIA SEVERINO DE ALMEIDA E PAIVA

DOCTOR BRIAN SADLER MAGERKO

DOCTOR MIRJAM PALOSAARI ELADHARI

DOCTOR TERESA ISABEL LOPES ROMÃO

DOCTOR MARIA LUÍSA TORRES RIBEIRO MARQUES DA SILVA COHEUR

DOCTOR DANIEL JORGE VIEGAS GONÇALVES

**TILT RIDERS:
CREATING AUTHORIAL-AGENTS FOR INTERACTIVE
STORYTELLING****ANTÓNIO PIMENTEL BRISSON LOPES****SUPERVISOR:** DOCTOR ANA MARIA SEVERINO DE ALMEIDA E PAIVATHESIS APPROVED IN PUBLIC SESSION TO OBTAIN THE PhD DEGREE IN
INFORMATION SYSTEMS AND COMPUTER ENGINEERING

JURY FINAL CLASSIFICATION: PASS WITH MERIT

JURY**CHAIRPERSON:** CHAIRMAN OF THE IST SCIENTIFIC BOARD**MEMBERS OF THE COMMITTEE:**DOCTOR ANA MARIA SEVERINO DE ALMEIDA E PAIVA, ASSOCIATE PROFESSOR (WITH
HABILITATION) AT INSTITUTO SUPERIOR TÉCNICO, UNIVERSIDADE DE LISBOADOCTOR BRIAN SADLER MAGERKO, ASSISTANT PROFESSOR AT GEORGIA INSTITUTE OF
TECHNOLOGY, SCHOOL OF LITERATURE, COMMUNICATION AND CULTURE, ATLANTADOCTOR MIRJAM PALOSAARI ELADHARI, ASSISTANT PROFESSOR AT UNIVERSITY OF MALTA,
MALTADOCTOR TERESA ISABEL LOPES ROMÃO, ASSISTANT PROFESSOR AT FACULDADE DE CIÊNCIAS E
TECNOLOGIA, UNIVERSIDADE NOVA DE LISBOADOCTOR MARIA LUÍSA TORRES RIBEIRO MARQUES DA SILVA COHEUR, ASSISTANT PROFESSOR AT
INSTITUTO SUPERIOR TÉCNICO, UNIVERSIDADE DE LISBOADOCTOR DANIEL JORGE VIEGAS GONÇALVES, ASSISTANT PROFESSOR AT INSTITUTO SUPERIOR
TÉCNICO, UNIVERSIDADE DE LISBOA**FUNDING INSTITUTIONS**FUNDAÇÃO PARA A CIÊNCIA E TECNOLOGIA THROUGH PhD GRANT SFRH/BD/37476/2007 AND
UNDER THE PROJECT GALA (GAMES LEARNING ALLIANCE) NETWORK OF EXCELLENCE FUNDED BY
EU FP7-ICT-2009-5 UNDER GRANT AGREEMENT NO: 258169 AND FCT - FUNDAÇÃO PARA A
CIÊNCIA E TECNOLOGIA, UNDER PROJECT PEST-OE/EEI/LA0021/2013

2013

A investigação em agentes autónomos para narrativas interactivas tem-se baseado no princípio de que é possível gerar histórias a partir das interacções entre agentes, que implementem papéis de personagens bem definidos. Esta aproximação implica grandes custos na configuração das personagens e comportamentos dos agentes. Propomo-nos a reduzir estes custos transferindo parte da responsabilidade de criação de histórias e das suas personagens para agentes autónomos.

Apresentamos o conceito de *authorial-agents*, agentes autónomos que raciocinam sobre o impacto dos seus comportamentos no desenvolvimento de uma história. A criação destes agentes implica a implementação de técnicas de interpretação que permitam negociar o desenvolvimento de uma história.

Neste trabalho, implementamos modelos computacionais baseados em: estudos de comportamento de actores em contextos de criação cooperativa e na análise de técnicas de interpretação. Contribuímos para a criação de *authorial-agents* identificando e implementando três diferentes técnicas. Os agentes usados nos dois primeiros casos de estudo implementam dois mecanismos de *tilt* distintos, detectados no nosso próprio estudo comportamental de actores de teatro de improvisação: *status shift* e *property inconsistency*. O agente estudado no terceiro caso implementa o mecanismo de *escalada emocional* fundamentado na nossa análise de técnicas de interpretação e estruturação de *sketches* humorísticos.

Abstract

Autonomous Agents research in Interactive Storytelling (IS) has been following the principle that stories can be dynamically generated by the interaction between characters in real-time systems, as long as they implement well-defined characters. Such approach requires a heavy authorial effort. We contend that this could be reduced by shifting some of the authorial power to the agents.

We present the concept of *authorial-agents*, autonomous agents able of reasoning over the impact of their own actions in the story development and change their actions accordingly. The creation of authorial-agents requires the implementation of acting techniques that negotiate story development in all its dimensions.

Our approach for creating authorial-agents consists on the implementation of computational models based on cognitive studies over actors behaviors or detailed analysis of acting techniques. We contribute towards the creation of authorial-agents by identifying and implementing three different acting techniques for authorial-agents. Each technique is presented in a case study. The authorial-agents for the two first case studies apply two different *tilt* mechanisms that were identified in our cognitive study of improvisational actors: *status shift* and *property inconsistency*. The third case study implements an *emotional escalation* mechanism supported by our analysis of the acting techniques used to create comic sketches.

Palavras-chave

Agentes Autorais

Agentes Autônomos

Narrativas Interactivas

Improvisação

Tilt

Escalada Emocional

Keywords

Authorial-Agents

Autonomous Agents

Interactive Storytelling

Improv

Tilt

Emotional Escalation

Acknowledgments



Contents

1	Introduction	1
1.1	Problem	2
1.2	Approach	4
1.2.1	Improv Analysis	4
1.2.2	Humour	5
1.2.3	Conceptual Model and Implementation	6
1.3	Contributions	6
1.3.1	Resulting Publications	7
1.4	Document Structure	8
2	Improv and Emotional Escalation Background	9
2.1	Improv Background	9
2.1.1	Brief History	10
2.1.2	Improvised Dialogues	10
2.1.3	Storytelling in Improv	11
2.1.4	Tilt	13
2.1.5	Status	13
2.2	Emotional Escalation Background	14
2.3	Conclusions	14
2.4	Summary	15
3	Related Work	17
3.1	Plot-Centered Approaches	17
3.1.1	Mimesis	18

3.1.2	FAçaDE	19
3.1.3	Discussion	22
3.2	Character Approaches to IS	22
3.2.1	Interactive Storytelling Prototypes.	23
3.2.2	Thespian	25
3.2.3	FearNot!	28
3.2.4	Double Appraisal	31
3.2.5	Discussion	32
3.3	Improv in IS	34
3.3.1	IMPROV at NYU	34
3.3.2	Directed Improvisation	35
3.3.3	Digital Improv	39
3.3.4	Back-Leading through Character Status	46
3.3.5	Late Commitment and Open World Assumption	47
3.3.6	Discussion	48
3.4	Summary	50
4	A Conceptual Model for Story Development Using Tilt	53
4.1	Video Data Description	54
4.2	Approach	55
4.3	Analysis	59
4.3.1	Tilt Occurrences	59
4.3.2	Location Divergence	60
4.4	A Conceptual Story Development Model With Tilt	61
4.4.1	Platform Buildup	62
4.4.2	Tilt Build Up and Problem	63
4.4.3	Resolution	64
4.5	Tilt strategies	65
4.5.1	Platform Representation	65
4.5.2	Knowledge Structure	66
4.5.3	Tilt: Status Shift	67
4.5.4	Tilt: Property Inconsistency	68
4.6	Discussion	68
4.7	Summary	69
5	Tilt Agents Implementation	71
5.1	Architecture Overview	71
5.2	Knowledge Structure	73

5.3	Populating Improv Knowledge Base	74
5.4	Assumption Engine	74
5.4.1	Assumptions Update	76
5.4.2	Platform Update and Tilt Detection	77
5.5	Status Shift Implementation	78
5.5.1	Tilt Status Function	78
5.5.2	Portray Status Antagonist Function	79
5.6	Property Inconsistency Implementation	79
5.6.1	Tilt Property Inconsistency Function	79
5.6.2	Tilt Started Function	80
5.7	Discussion	81
5.8	Summary	81
6	Case Studies	83
6.1	Tilt Agents	83
6.1.1	Tilt: Status Shift	84
6.1.2	Tilt: Property Inconsistency	85
6.1.3	Discussion	86
6.2	Emotional Escalation Agent	86
6.2.1	Implementation Overview	87
6.2.2	Scenario	87
6.2.3	Authoring	88
6.2.4	Evaluation	88
6.2.5	Emotional Escalation Discussion	91
6.3	Case Studies Discussion	92
6.4	Summary	93
7	Conclusions and Future Work	95
7.1	Improv Agents	95
7.1.1	Further Cognitive Assessment for Story Development	97
7.2	Emotional Escalation Model	98
7.3	Further Conclusions	99
7.4	Summary	99
	Bibliography	108
	A Coffeshop Scene Analysis	109
	B Wife Zoo Scene Analysis	139

C LaughToMe! Questionnaire	225
D LaughToMe! Output	229

List of Figures

1.1	Research approach	4
2.1	Collaborative emergence process	11
2.2	Storytelling 3 beat sequence	12
3.1	Mimesis Components	18
3.2	FAçaDE screenshot	20
3.3	FAçaDE architecture	21
3.4	Character centered authoring	23
3.5	HTN example	24
3.6	Screenshots of different Interactive Storytelling Prototypes.	25
3.7	Screenshot from Tactical Language System	26
3.8	Proactive authoring cycle	27
3.9	Facilitator architecture	29
3.10	Story facilitator cycle	29
3.11	Spectrum of Directed Improvisation	36
3.12	Logical system organization for two agents in virtual world.	37
3.13	Otto and Gregor in a master-servant scenario.	38
3.14	Improv domain theory map	41
3.15	Improv agent decision cycle	43
3.16	Knowledge representation	44
3.17	Theory of Force Dynamics	47
4.1	Data gathering process	54
4.2	Video analysis process	56

4.3	Coding sequence for a regular offer confirmation interaction	58
4.4	Conceptual story development model using tilt	62
4.5	Platform elements representation	65
4.6	Platform elements instantiation	66
4.7	Knowledge structure for tilt analysis	67
5.1	Tilt agent architecture	72
5.2	Knowledge structure	73
5.3	Screenshot of the KB menu for populating DOA	74
5.4	EER diagram of MySQL implementation	75
5.5	Inference strategies	75
5.6	Offer / Confirmation sequence for a strong offer with a strong confirmation.	77
5.7	Tilt: status shift implementation overview	80
6.1	<i>Laugh To Me</i> screenshot	88
6.2	Emotional goal DispleaseClient-A	89
6.3	Emotional goal KeepPride	89

List of Tables

3.1	Improvisation forms	36
4.1	Video data details	55
4.2	Tilt examples from the scenes analyzed	55
4.3	Coding scheme	57
4.4	Annotated platform variables	58
4.5	Quantification of the variables annotated for each actor's story frame in Scene A. .	59
4.6	Quantification of the variables annotated for each actor's story frame in Scene B. .	59
4.7	Tilt occurrences	60
4.8	Non consensual location example	60
4.9	Sub-set of the actor' story frames at Turn 2.	61
4.10	Story steps primary goals	61
4.11	Platform offer	62
4.12	Platform exploration	63
4.13	Tilt offer	63
4.14	Tilt riding	64
4.15	Tying loose ends	64
5.1	Platform elements collection	72
6.1	Tilt example: Status Shift	84
6.2	Tilt example: Property Inconsistency	85
6.3	Questions stated in the online questionnaire	90
7.1	Improv games for further assessment	98

CHAPTER 1

Introduction

Each media breakthrough brings new opportunities for storytellers, either by improving previous paradigms or by establishing new ones. The digital breakthrough of the last few decades is no exception. Our heroes are no longer stuck in our television set, they are on the web, on our smartphones, they are everywhere and, most importantly, they are interacting with us!

At least ever since Meehan and Schank's TALE-SPIN (Meehan, 1977; Schank and Riesbeck, 1981) that the Interactive Storytelling (IS) community has been exploring the possibilities of computation for creating stories. We are not just interested in representing stories in a computational form, but in using technology to promote user's sense of immersion, agency and transformation (Murray, 1998). We want users to feel part of a story by playing a decisive role in the action development.

Artificial Intelligence (AI) research in the domain of autonomous agents has been playing an extremely relevant role in IS. Agent approaches to IS rely on the principle that stories can result from the actions and interactions between those agents in real-time systems, as long as they implement well defined roles (Young, 1999). This interest goes back to seminal projects such as the OZ project (Bates et al., 1994) or CAIT (Hayes-Roth et al., 1995) and continues to be followed (Aylett et al., 2006; Si et al., 2005b) nowadays in diverse IS application areas.

The implementation of agents with a higher level story perspective that goes beyond the character perspective, poses too many difficulties and most researchers have been privileging the implementation of agents that have well defined roles. Recent developments towards the creation of rich autonomous characters that plan with emotional consistency citepAylett:2006kx,Si:2010qf, increase story variability (Cavazza et al., 2002) or adapt character social behaviors to particular cultural aspects (Mascarenhas et al., 2009; Si et al., 2006) are indeed role constrained. However, by

limiting such complex agents' behaviors to well defined roles, and given that each agent needs to be authored independently of the others due to its particular role in the story, the use of autonomous characters has lead to the increase of the authorial burden.

An alternative approach to these as been taken by two different projects *Versu* (Evans, 2013) and *Prom Week* (McCoy et al., 2011, 2012). Both cases approach the definition of character behaviors as a result of the social context in which the characters are embedded. At the time of writing there are still no big insights about *Versu*, which is a commercial product, except for some of Richard Evans' invited talks, as the one that occurred at ICIDS 2012. In *Versu*, social behaviour is apparently achieved through careful authoring of social conflict situations written by a professional writer. In *Prom Week* the social behavior is autonomously generated using rule models inspired in social sciences. These two alternatives are very recent, and while a critical assessment of *Versu* would require more details it seems that its dependence of very detailed authoring does not allow it to make the case for reducing the authorial burden. In the other hand, behaviour generation in *Prom Week* looks very promising and it is widely recognized to have very good impact in gameplay, although this is still achieved in a sequence of very small and strict situations. Furthermore, we argue that these approaches model social interesting situations but do not model story development itself since the agents continue to be constrained in a social role.

The work presented in this thesis is motivated by the argument that the authorial burden of autonomous agents for IS can be reduced by endowing agents with their own authorial abilities. This means creating agents that implement real time processes of evaluation and creation of stories. In a way, we are trying to move from the creation of "virtual characters" in a story to the creation of "virtual story improvisers" that make stories emerge.

We propose the concept of *authorial-agents* as autonomous agents that are able to reason about the impact of their own actions in the story development and change their actions accordingly. Authorial agents rely on the implementation of their own story models which emerge in collaboration with other agents.

1.1 Problem

Authorial-agents do not play roles that are completely pre-determined. In a system with authorial-agents, the narrative itself is not pre-determined. The narrative is defined from the agent's interactions with each other within story action, hence they have explicit intentions regarding their performance that go beyond the simple sequence of actions. This is similar to the problem that real life actors have when improvising or interpreting a scene. They act in front of an audience while at the same time they negotiate with each other the progress of the scene.

The creation of authorial-agents requires the implementation of acting techniques that negotiate story development in all its dimensions. One fundamental reason behind the disinclination of

research towards authorial-agents is the lack of a clear assessment of the cognitive processes and acting techniques involved in a collaborative story creation. Such knowledge could support the design of computational models to be used in a multi-agent context. This brings us to the following research question:

How can we create authorial-agents that embed acting techniques and story models for collaborative story creation environments?

Collaborative story creation can be seen as a complex negotiation process where each authorial-agent proposes, accepts or rejects story elements based on their story model. In order to create a collaborative story model one must address several sub-problems:

- Which story elements are relevant for a story? - stories can be decomposed onto many different elements, such as characters, characters' motifs, location, relations, objects, events, premise. A story model for collaborative environments should be represented in terms of the processes of detecting and manipulating the elements that better contribute to the story creation.
- Which elements move a story forward? - The addition of story elements to a scene offers new opportunities for story development. Authorial-agents should identify the story elements that better create such opportunities, by reasoning upon the impact that a new story element might have in a ongoing story.
- Which story elements are more suitable for each story moment? - Different stories have different paces. Authorial-agents should be able of reasoning over story development in order to understand story pace and perceive the right timing for adding elements to a scene.

Story pace is deeply related to its emotional evolution. It is from the display of the characters' inner conflicts and emotions that tension grows. Authorial-agents in digital environments share the responsibility of conveying the inner perspective of the characters that better contribute to the story impact on the audience. This implies the following sub-questions:

- Which techniques can be used by an agent to increase its expressivity? - an authorial-agent should implement strategies for portraying inner character states in order to affect the emotional tension of a story.
- How can an authorial-agent deliberately affect the emotion of a scene? - in order to control the impact of their own performance in a story, authorial-agents should be able of reasoning over their impact on the emotional development of a scene.

1.2 Approach

We consider authorial-agents as a virtual metaphor for actors in a non scripted environment. This novel approach to IS's autonomous agents presents many research challenges for years to come. In this thesis, we hypothesize that valuable contributions towards the creation of authorial-agents can result from the analysis of real actors and acting techniques. Each such technique should be analyzed and implemented in small authorial-agents prototypes, thereby isolating each contribution in order to facilitate its validation. We propose to address authorial-agents research in a three stage approach consisting of: analysis, conceptual model and implementation. The first is the analysis of relevant acting techniques and actors behaviors. In this analysis step we will consider two particular cases: Improvisational Theatre (Improv) and Humour. After this, we will then formalize the results of the analysis into functional conceptual models. In the last step we implement the conceptual models in an authorial-agent (Fig. 1.1).

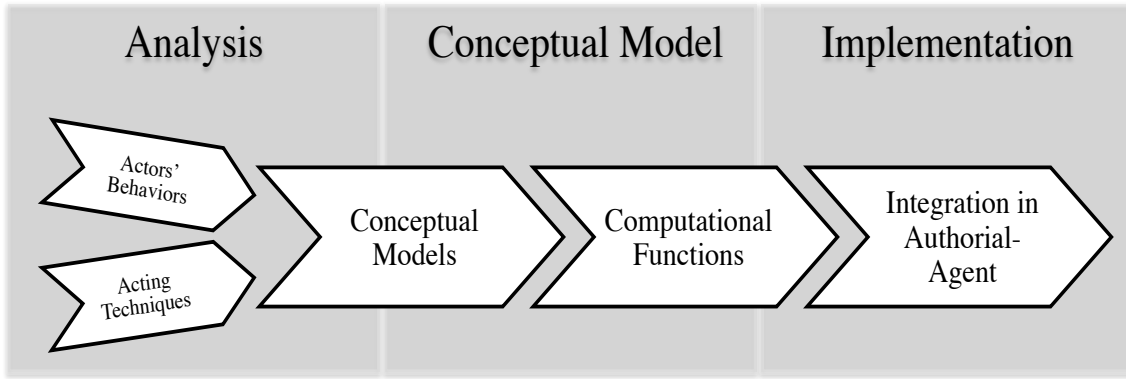


Figure 1.1: Research approach

1.2.1 Improv Analysis

Several research areas have studied improvisation and its implications. From a management perspective improvisation has been defined *"as the degree to which the composition and execution of an action converge in time"* (Moorman and Miner, 1998), while at the same time it has been regarded in other disciplines as the act of real-time dynamic problem solving (Johnson-Laird, 2002). In general terms we can define improvisation as a single process that merges the execution and conceptualization of an object or task.

Improv is a term commonly used to identify the theatre approach to improvisation. Improv has been defined as *"a form of unscripted performance that uses audience suggestions to initiate or shape scenes or plays created spontaneously and cooperatively according to agreed-upon rules or game structures"* (Seham, 2001) and as *"the creation of an artifact with aesthetic goals in real-time that is not completely prescribed in terms of functional and/or content constraints"* (Magerko et al., 2009).

We take improv as the real life cooperative story making example that is closest to the concept of *authorial-agents*, because in both contexts the story results from the on-the-fly contributions of individuals that participate in the story as characters. We argue that the problem of creating a story in improv is similar to the problem of creating stories in a multi-agent context, although one must consider the AI limitations in this particular context.

We approach improv as the main source of analysis for creating *authorial-agents*. In improv, actors develop stories by developing a shared understanding about a platform, which is the collection of story elements that establishes *who*, *what* and *where* the story being created is happening (Johnstone, 1999). An improv scene with two actors, might start with one actor acting like superman showing-off his super powers and flying around. Then the other actor could start pretending to be a kitten stuck in a tree branch. This would establish who the characters are (powerful superman and friendly kitten), where they are (near a tree), and what they are doing (superman is rescuing the kitten). Such platform establishes a consistent story setting, but it does not provide much more action to explore beyond the easy rescue of the kitten. Where is this story heading?

Good improv actors are not only experts in the cooperative process of establishing a platform, but also in adding story elements that *tilt* this platform in order to create story movement, provide climax and guide the sequence of actions. Continuing with our previous example, the actor playing the kitten could *tilt* the established platform by showing a *kryptonite* rock attached to his leash. The *platform* changes, because superman is no longer powerful and now one will doubt about the kitten's friendliness. Now the actors have the need to justify these changes, and new action will emerge from this need. The relevance of the tilt process in improv's collaborative story creation makes it a natural candidate for the first step of our approach. As such, we proposed to assess the information used by real improv actors and the techniques by which this information is manipulated to produce a *tilt*. We will pay special attention to status changes, i.e changes in the character's levels of dominance over the other characters and scene elements. Status changes are mentioned by practitioners as a most relevant instrument for encouraging actors to add new dramatic content that justifies changes (Johnstone, 1987, 1999).

The second step of the analysis stage is to build conceptual models of the techniques identified using very low level functions. This step is bounded by the research goal of creating computational solutions for autonomous agents oriented towards the use and manipulation of explicit information. Because of this, the assessment of the tilt process is intrinsically related to the story development model and, as such, a conceptual model for tilt must fit into a general story development model.

1.2.2 Humour

Humour is "the quality of being amusing or comic" (Dictionaries, 2012), being amusing or comic a quality of what causes laughter. We can consider humour as a process of evoking laughter. We regard humour as highly emotional context, in which the actors sense of timing is decisive to their

comic impact. Adding to this, humour is highly present in improv, hence, we defend that the study of humour could also contribute the development of authorial-agents.

There are many acting techniques and story development forms related to humour, from which comic sketches is perhaps the most popular and the one that is closest to short improv presentations, in terms of structure. While performing a comical sketch, actors are concerned about building up tension in order to create the right timing for a punchline, the final line that causes the comic effect. This concern establishes two parallel processes. While in one hand actors are concerned about the action sequence, on the other hand they are concerned about building up the right tension before releasing the punchline. We argue that this parallelism is a real life example of an authorial-agent that is concerned about the impact of its own actions on the overall story structure, rather than being just focused on maintaining the character role consistency.

We propose to analyze how this buildup is managed by an actor in terms of character emotions. We need to assess a computational model to be used by authorial-agents that endows them with the ability of actively considering the emotional evolution of the characters in a scene. We call this model *Emotional Escalation*.

1.2.3 Conceptual Model and Implementation

The conceptual model stage takes on the result of the analysis and organizes them in terms of functions. The implementation stage starts with the implementation of the functions identified in the previous conceptual models. In this step the formalisms for representing each function are validated in algorithmic representation. This stage is completed with the integration of each technique in an authorial-agent scenario that allows the demonstration of the use of each model.

1.3 Contributions

In this work we present the concept of "authorial-agents" as autonomous characters that reason about the impact of their actions in the development of a story. As mentioned earlier, this is a novel approach to autonomous agents research in IS that involves many research challenges for years to come. We contribute towards the creation of authorial-agents by extracting and validating, through implementation, relevant conceptual models of acting techniques based on improv and humour. More specifically, we present the following contributions:

- A collaborative story development model using tilt for authorial-agents - although others have approached improv as a metaphor for interactive storytelling, this has only been done through shallow understandings of improv theory (Hayes-Roth et al., 1995; Louchart and Aylett, 2002; Perlin and Goldberg, 1996; Swartjes, 2010). We present a collaborative story development model that is based on the cognitive study using real-life improv actors (Brisson et al., 2011a, 2013).

- Emotional escalation model - we present the concept of *emotional escalation* as a strategy for autonomous agents to manipulate the tension of a scene using emotions. *Emotional escalation* endows authorial-agents with the ability to cooperate in the creation of the emotional development of a scene. By controlling the timing and emotional impact of their actions authorial-agents can proactively build up tension to prepare relevant story moments (Carvalho et al., 2012).
- Implementation and discussion of three different authorial-agents - The first two agents are called *improv agents* due to their relation to the collaborative story development model using tilt. The *improv agents* implement tilt strategies identified in (Brisson et al., 2011b, 2013). The last agent is an *emotional escalation agent* based on the analysis of humour and comic sketch (Carvalho et al., 2012):

Status Shift Tilt Agent - applies a tilt strategy using status. This strategy was identified in the analysis of the collaborative story development model.

Property Inconsistency Tilt Agent - applies a tilt strategy using property inconsistencies. This strategy was identified in the analysis of the collaborative story development model.

Emotional Escalation Agent - applies the concept of emotional escalation in context of a comic sketch generation.

1.3.1 Resulting Publications

The above mentioned contributions have been published in the following:

- A. Brisson, B. Magerko, and A. Paiva. A Bottom Up Approach to Create Authorial-Agents based on Improvisational Theatre. *Special Issue on Computational Narrative and Games. IEEE Computer Society (to appear)*, 2013.
- A. Carvalho. Laugh to me: Exploring computational humour. *Master's thesis, Instituto Superior Técnico, Technical University of Lisbon*, 2012.
- A. Carvalho, A. Brisson, and A. Paiva. Laugh to me! implementing emotional escalation on autonomous agents for creating a comic sketch. *In ICIDS 2012 Proceedings of the 5th Joint International Conference on Interactive Digital Storytelling: Interactive Storytelling, Lecture Notes in Computer Science. Springer-Verlag Berlin*, November 2012.
- A. Brisson, B. Magerko, and A. Paiva. A computational model for finding the tilt in an improvised scene. *In Fourth International Conference on Interactive Digital Storytelling, ICIDS 2011*, volume 7069 of Lecture Notes in Computer Science, pages 158 - 163, Vancouver, Canada, November 2011b. Springer.

- A. Brisson, B. Magerko, and A. Paiva. Tilt riders: Improvisational agents who know what the scene is about. *In Proceedings of the 10th Intelligent Virtual Agents*, volume 6895 of Lecture Notes in Computer Science, pages 35 - 41, Reykjavik, Iceland, September 2011a. Springer.

1.4 Document Structure

The document is structured as follows: we start by presenting the Improv and Emotional Escalation Background (Chapter 2) that is more relevant for our research. Afterwards we present and discuss related work in the area of agents for IS (Chapter 3). The cognitive study that supports our improv approach is described in Chapter 4 along with the story development model and tilt strategies identified in it. We then follow with the presentation of the implementation of a tilt agent on Chapter 5. Chapter 6 describes the simulation scenarios of three authorial agents: Status Shift Tilt Agent, Property Inconsistency Tilt Agent and Emotional Escalation Agent. In Chapter 7 we detail the conclusions of our contributions and discuss future relevant research directions for authorial-agents.

Improv and Emotional Escalation Background

In the previous chapter we proposed to contribute to creation of authorial-agents by implementing computational models of acting techniques. In this chapter we summarize the relevant backgrounds of Improv and Humour that fundament our research.

2.1 Improv Background

Improv (Improvvisational Theatre) is *"a form of unscripted performance that uses audience suggestions to initiate or shape scenes or plays created spontaneously and cooperatively according to agreed-upon rules or game structures"*(Seham, 2001). In this special theatre genre, players (the term commonly used in improv for actors due to the strong presence of play/game models in improv) are free to change, adapt, add or remove elements to the scene whenever they want, with the goal of creating a good show for an audience.

As opposed to traditional acting, where the actors prepare themselves to portray a strongly detailed character before acting a pre-authored story, improv players know almost nothing about their characters or the story that is being presented when they step on stage. Character motivations, elements for justifying their motivation and behaviors, are constantly being created on the fly in cooperation with other players. This leads to a cooperative creation of a story world. Improv players are not serving a pre-defined story within a strongly defined character. Conversely, they share the creative responsibility of the show, which includes building their own character and story.

While it may seem that improv's intrinsic unpredictability prevents its use as source of study for computational models, it is clear that improv actors have the need to achieve consensus in a

performance. According to improv theorists, such as Johnstone (Johnstone, 1987, 1999) or Sawyer (Sawyer, 2003), improv performances tend to fall into very high-level structural forms. In this chapter, we will present the improv concepts used in our research. We start by presenting an historical background. We then present the concept of collaborative emergence as it is proposed by Sawyer (Sawyer, 2003) and the storytelling fundamentals of improv.

2.1.1 Brief History

The Improv Theatre's roots go a long way back at least until 16th century's *Commedia dell'arte*, a street theatre style performed by itinerant companies across Europe. *Commedia dell'arte* used stereotypical masks and improvised performances based on sketches or small scenarios. Famous theatre theorists and directors considered improvisation as a powerful acting instrument for learning and rehearsal. One example of this was Konstantin Stanislavsky, who repeatedly included improvisational exercises in his actor training Method (Stanislavski, 1989). In early 1900's Jacques Copeau a very influential french director, was the first to use semi-improvisational games in rehearsals (Copeau, 1990). Also by this time Moreno created what might have been the first professional company to perform without a script "The Theatre of Spontaneity" (Moreno, 1983).

The interaction between all these theatre schools resulted in a prosperous ground for practitioners particularly focused on improvisation, such as Viola Spolin (Spolin, 1999). The social turmoil around upper class theatre elites and the need to bring theatre closer to the common working class was the driving force that led some of these practitioners to the creation of the first Chicago Improv Theatre, "*The Compass*" (Coleman, 1991). This company was the origin of a long history of success and failures of improv theaters across the United States that eventually led to the creation of numerous improv forms and improv theatres around the world, and established Chicago as the birthplace of improv theatre.

2.1.2 Improvised Dialogues

In "Improvised Dialogues - Emergence and Creativity in Conversation" (Sawyer, 2003), Sawyer presents a very detailed ethnographic and sociological analysis of the cooperative creative process in improv. Sawyer's work was based not only on video recordings of performances, but also on notes taken from his privileged position as a piano player for some Chicago improv groups in the period between 1992-94. Among other aspects, this analysis regards the dynamic creation of a collective conversational frame and its influence in each individual performance. It also includes very detailed performance transcriptions and dissections, as well as a reference for examining collective behaviors in this context. From all the contributions of this research there are four elements that are essential to our work: *Dramatic Frame*, *Collaborative Emergence*, *Offer and Response*.

Following Sawyer (Sawyer, 2003), when two or more improv actors improvise, their dialog results in the creation of a *dramatic frame*, which is a shared understanding of all performance elements

brought to scene. Such elements include among other: the characters enacted by each actor; their motives and relations; the joint activity in which they are engaged; action location; time period; story genre; relation of the current joint activity to that plot and a large amount of implicit information (information that is presented without being directly referred), such as contextual information about an activity, place or time.

The *dramatic frame* configuration results from a turn-by-turn interaction called *collaborative emergence*, which is supported by two major functions:

- *offer*: proposal of a new element to add to the frame, this may be related to any element of the dramatic frame.
- *response*: validation of early proposals by integrating them in the dramatic frame or rejection of early proposals by disabling its integration or making it more difficult.

It is the interaction between offers and responses that determines the collaborative property of the process. *Offers* with no *response* are kept in the individual frame of each actor. It is only from *offer* / *response* agreement or disagreement that new elements may take part or be excluded from the dramatic frame. Only confirmed offers can be moved to an emergent dramatic frame, which then influences the next actions of the actors (vectors of causation).

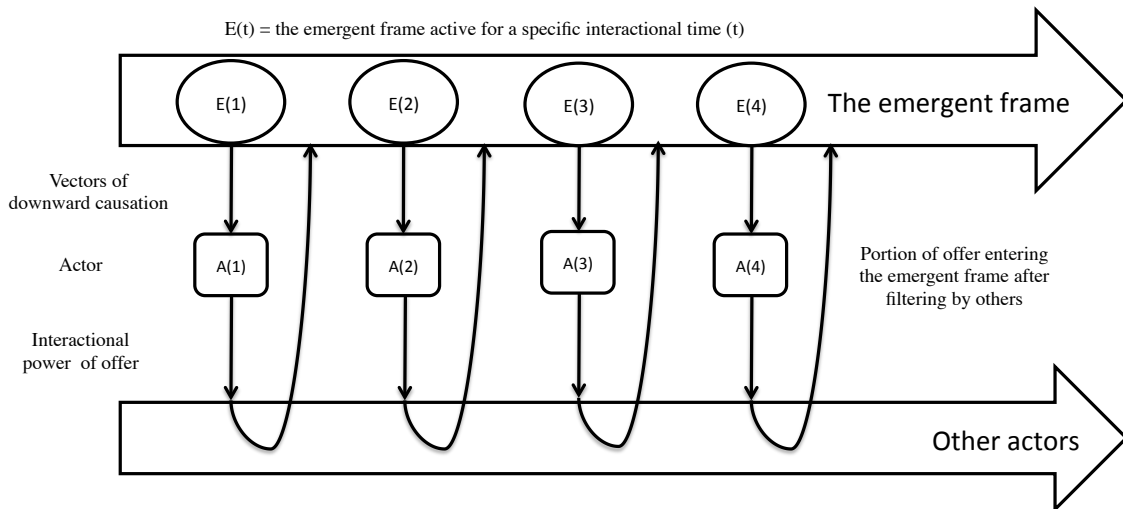


Figure 2.1: Collaborative emergence process, adapted from (Sawyer, 2003) pp.68

2.1.3 Storytelling in Improv

In spite of all the creative freedom involved in improv, improv players seem to have a tendency to fall into very high level patterns when creating a story. A possible reason behind this is that storytelling is not only an ancestral art but also a very important component of human thought. As such, storytelling development is linked to the development of thought itself, and influences our way of acting and telling stories. Improv players are no exception to this.

At least since 353 BCE , the approximate date of the earliest-surviving work of dramatic-theory Aritotle's *Poetics* (Buckley, 1992) authors have been privileging three act structures in their stories. It is taken as a common rule of theatre that three acts is a safe form that includes exposition, climax and resolution. Egri addresses this issue as a *Dialectic* process that starts with the presentation of a statement proposition, a *Thesis*, followed by an *Antithesis* that opposes the initial statement and is concluded by a *Synthesis* that resolves the conflict between the two early elements. He defends that this process provides the most important element of drama movement, conflict (Egri, 2004). In cinema McKee addresses this element as "*The Law of Conflict*", stating "*Nothing moves forward in story except through conflict*" (McKee, 1997), and places it as the motion source of his three act adaptation to cinema *Central Plot* (McKee, 1997).

Three Beat Sequence

Improv players tend to fall into very high level structural forms similar to the rule of three. Note that we are referring to very high levels of a collective creative process, is not the same than saying that experienced players fall or tend to fall into predefined plots, or story settings.

An example of this tendency that is extremely important to our research is the three beat storytelling process observed by Johnstone(Johnstone, 1999) and also by Sawyer(Sawyer, 2003). Both authors observed that storytelling in improv tends to follow a pattern that starts with the establishment of a routine, and is followed by a disruption of that same routine that leads to the need of resolving the discrepancies elicited by the earlier disruption. During the first two beats that normally represent half of a scene "*actors are encouraged to offer new material*" (Sawyer, 2003) and in the last beat they are encouraged to connect the elements introduced earlier in the scene resolving all or the most relevant discrepancies raised in the breaking of the routine (see Figure 2.2).

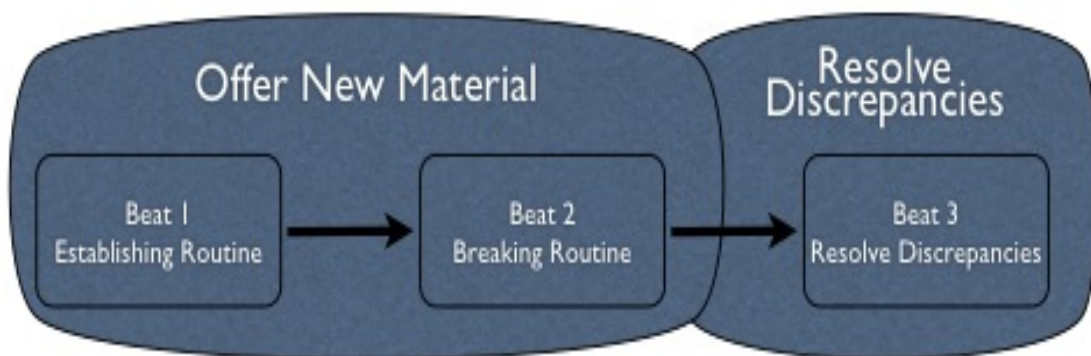


Figure 2.2: Storytelling 3 beat sequence

2.1.4 Tilt

In improv, players develop stories by developing a shared understanding about a platform, a collection of story elements that establishes *who*, *what* and *where* the story is happening (Sawyer, 2003). However, the simple creation of a platform, e.g. a mellow/passionate couple (*who*) in heir honeymoon (*what*) entering their hotel room (*where*), is not much more than a description of the story environment, "*the stability that precedes the chaos*" (Johnstone, 1999), and it does not produce an interesting story by itself. It is up to improv players to introduce new elements to the scene that unbalance the established platform. An example of this scenario could be the husband finding his spouse wearing his clothes and then have to adapt to the fact that she is the man in the house.

Johnstone also reports the possible existence of several weak tilts on scene. He defends that weak tilts are tilts because they affect a platform's balance, but do not introduce "a mystery" to a scene that strong tilts do.

We can define tilt as the process of proposing a new story direction by introducing a significant change in an established platform.

2.1.5 Status

From a sociological perspective, status is a complex social concept that defines a position in a social system (Linton, 1950). It is directly related to the social structures of roles that define hierarchies and the rights and duties of individuals. People are associated to status only through their participation in social systems that include them. Also, following Kemper (2011) status is both a scalar a value that defines social positions in a group, and also "*the actual behavior or acts or means by which the scalar standings, worth, prestige, honor of a person or its social position is conveyed in interaction*".

Johnstone (1987) argues that a very significant proportion of drama in improv results from status transitions, i.e. changes in character status, because in order to accommodate these transitions the actors need "*to find justifications for the changes*" (Johnstone, 1999).

Status is present in improv in three dimensions (Johnstone, 1987): demeanor, relationship and space. Demeanor is perceived through character postures, where high status is portrayed through erect and graceful postures while low status is associated to awkward and graceless postures. High relationship status is directly related to authority the same way that low status in this dimension is related to acquiescence. The space dimension is related to the way each actor can occupy space, the higher the status the more an actor can occupy space as opposite to low status actors try to avoid intruding upon space.

By combining the three status dimensions one observes that status can be addressed as a measure of social dominance that is attached to a subjects position in a group, but at the same

time it can be affected by the results and form of his interactions. This status omnipresence in characters' actions, led Johnstone to argument about the impossibility to enact a neutral-status character.

2.2 Emotional Escalation Background

The relevance of comedy in theatre and film has not passed unnoticed by the IS community (Cavazza et al., 2003; Olsen and Mateas, 2009; Thawonmas et al., 2003). The common ground between most IS approaches to comedy is that they find theoretical support on Schulz's incongruity-resolution theory (Martin, 2007). Schulz proposes that comic situations arise from a character failing to achieve a goal, such formalization has been adapted to IS through the implementation of planning formalisms that allow a character to fail in order to produce a comical effect.

One particular comical genre that can be related to this thesis research on authorial-agents is comical sketch. Following PERRET (1990) and Vorhaus (1994), sketches are short, isolated scenes that develop a certain *comic premise* (the incongruity that composes the initial idea of a comic story) before offering a *punchline* (the conclusion of a joke that resolves the incongruity and offers humour). Vorhaus also defends that this should be achieved by creating and gradually escalating a conflict between characters. Both authors stress the need to for building up the *tension* (set-up) before presenting a joke (pay-off).

Vorhaus (Vorhaus, 1994) underlines the importance of creating and developing a character conflict in comic sketch. He proposes the concept of *strong comic perspective*, which consists in developing unique character points of view. Such character's point of view should result from specific personality traits that are distorted from the common personalities.

In our research we are aware that comicality and its intrinsic need for building up the right tension before delivering a joke is one the hardest acting challenges. We define *Emotional Escalation* as the process by which an authorial-agent gradually increments the emotional tension of a scene in order to setup the right timing for a punchline.

2.3 Conclusions

Johnstone and Sawyer present formalizations about the processes embedded in improv acting. Such formalizations allow us to establish a formal approach to the analysis of improv for computational purposes. While collaborative emergence formalizes the process by which actors create a shared knowledge that influences their own performance, Johnstone proposes an intrinsic structure for how stories are developed in this context by proposing a three beat pattern. From a functional perspective this pattern could be implemented by three high-level functions:

1. Storyworld creation

2. Break of the regular development of the storyworld created previously
3. Reestablishment of the regular development of the storyworld

Each one of these functions is a huge challenge for AI and raises innumerable issues. The first two functions are more related to the association of ideas and concepts, posing problems of knowledge representation, data-mining or common sense reasoning, to name a few. The third function is more related with causality connections on real time which present reasoning and beliefs revision challenges.

Both authors emphasize the nuclear relevance of the tilt process on the storytelling process. Tilt clearly operates over a platform, which can be seen as a sub-set of the scene knowledge. Hence a detailed assessment of the tilt process could allow us to determine the knowledge and the functions needed for an authorial agent to provide movement to a story, and consequently contribute to a definition of a cooperative story development model.

Based on his experience as a practitioner and teacher, Johnstone argues that an improv actor is constantly conveying his status. This argument is supported by Kemper's status definition that not only considers the social position of a character but also his actions. We conclude that status can be addressed as a measure of social dominance that is attached to a subjects position in a group, but at the same time it can be affected by the results and form of his interactions.

Emotional Escalation is the process by which an authorial-agent gradually increments the emotional tension. This can also be seen as strategy for building up tension in order to prepare the delivery of a joke. The development of an *Emotional Escalation* model for authorial agents, should consider the concept of *strong comic perspective*(Vorhaus, 1994) and this could be done by endowing authorial-agents with personality representations to support their characters behaviors.

There are many similarities between the comic sketch theory supporting *emotional escalation* and the improv background. *Strong comic perspective* can be seen as a tilt of a characters personality from a normal personality, furthermore a punchline can also be seen as a tilt from an incongruent platform to congruent platform. Under this perspective, emotional escalation gains more relevance as it can also be used to prepare the right timing for a introducing a tilt.

2.4 Summary

In this chapter we presented an overview of improv's history and main concepts that relate to our research. We presented a very brief overview of the classical *rule of three*, specially its consistent presence in theatre and cinema and its impact in Improv. We summarized the process of collaborative emergence. This background supports the data analysis approach that will be presented in Chapter 4. We also presented the theoretical background for emotional escalation which supports the Emotional Escalation Agent implementation that will be presented in Section 6.2.

In the next chapter we will discuss related work and how the development of authorial-agents can improve the current interactive storytelling state-of-the-art.

CHAPTER 3

Related Work

Since its emergence, IS has been an extremely attractive application field for a wide variety of research disciplines. Each discipline brings novel techniques, advances and perspectives about IS. While multidisciplinary enriches our research field it also contributes to the lack of a standardized IS taxonomy shared by all researchers. In this chapter, we present and discuss research examples that are relevant for illustrating the contributions of this thesis. We will focus on describing different approaches to the development of models that embed story structure, its elements and the impact of such elements on story development. In this domain we make a clear distinction between systems in which the complete story development is centralized in a single entity and systems in which the the story representation is split among multiple entities with complementary knowledge about the story development. This distinction is motivated by the strong impact that each of these representations have on the systems' development process, potential and final results.

The works presented in this chapter are divided into two categories: *Character Centered* approaches where all story knowledge is encoded in the characters AI; *Plot-Based* approaches where the story knowledge is encoded in global action sequences. Additionally, because of the the relevance of improv in the research approach followed in this thesis, we extend our IS research overview with a detailed discussion of the main improv contributions to IS that are previous to this research

3.1 Plot-Centered Approaches

Plot-Centered approaches to IS are motivated by the intention of providing a more restrained authoring environment that enhances authorial control. The story plot is encoded in a unique set of rules that are normally used in a planning algorithm. The following sub-sections present

examples of approaches to IS systems where the story knowledge is encoded in global action sequences that affect multiple parts of a story world.

3.1.1 Mimesis

Mimesis (Young et al., 2004) is an IS system that uses a plot centered approach. A very interesting aspect of its architecture is that it distinguishes between the story content and its presentation to the user. This particular aspect allows the system to manipulate the story presentation without affecting significantly the authorial intentions included in the story content. This way, it is expected that Mimesis increases the capability of adapting the story development to the user behavior without breaking the story structure.

Architecture

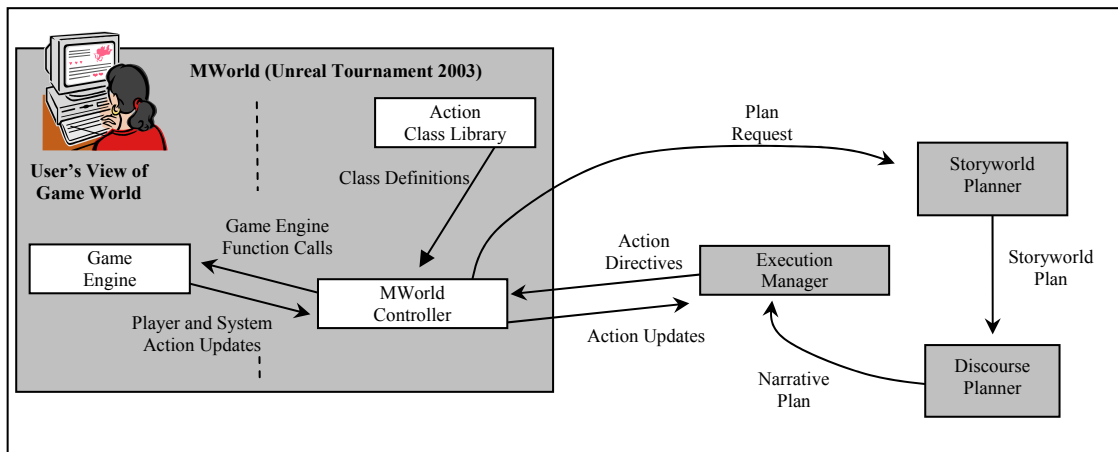


Figure 3.1: Mimesis components from (Young et al., 2004)

The Mimesis action described in Fig.3.1 begins with a plan request from the game engine to the *Storyworld Planner*, that includes a specific story problem that needs resolution. The *Storyworld Planner* receives this request and creates a new *Storyworld Plan* for the action with the goal of resolving the story problem provided by the engine. *Story Planner* then sends the plan to the *Discourse Planner* which is the the component responsible for planning the action sequence, schedule all the media resources needed for a story including the presentation sequence of the story (*discourse*). This planner integrates the *Storyworld* and the *Discourse* plan onto a general plan describing all system and user activity to be executed. Afterwords, the *Narrative Plan* is sent to the *Execution Manager*, that builds a Direct Acyclic Graph, and starts acting like a process scheduler selecting the actions for execution according to plan. These actions are sent to the *MWorld* in the form of *Action Directives* that include all the information of the action to be executed. The *MWorld* component that receives the *Action Directives* is the *MWorld Controller*. When this component receives an input, it maps the information contained in the *Action Directives*

onto their respective *Function Calls* in the Game Engine. In order to do this, the *Controller* uses the *Action Class Library* with all class definitions and a look-up table. After sending the functions for execution, the *MWorld Controller* receives a notification whenever an action is halted either by successful or unsuccessful completion and communicates it to the *Execution Manager*. The system keeps looping until the end of the plan is reached.

Intervention and Accommodation

When an IS system includes user interventions, it adds the user's unpredictability to the story. Problems arise in this context whenever the user executes actions that conflict with the active plan. More precisely, whenever a player decides to perform an action that generates a world state that conflicts with the constraints of the story plan. In Mimesis these actions are monitored via user's action commands, prior to its execution. The system chooses between two strategies when an exception is signaled:

- *Intervention* - The system intervenes by causing the action to fail.
- *Accommodate* - The system adjusts the structure of the plan to accommodate the new activity of the user.

With these two strategies, the system considers the computational costs associated to the generation of a new plan and the impact over the user's sense of agency (Murray, 1998). Mimesis optimizes this selection by analyzing each plan before executing it. This analysis aims at identifying points where enabled user actions can threaten its plan structure, before executing it.

Discourse Manipulation

Recent contributions from this research derive directly from the architectural detail of separating the *Discourse Planner* from the *Storyworld Planner*. This very particular detail can be used to manipulate camera control (Jhala and Young, 2009), rearrange the order of presentation of the story events in order to generate surprise and arousal in the user (Bae and Young, 2008) or for creating suspense (O'Neill and Riedl, 2011).

3.1.2 FAçaDE

FAçaDE is the most disseminated IS application with more than 500000 downloads since July 5Th 2005 from its website (Mateas and Stern, 2011). In this IS experience the user is invited to visit a special couple, Grace and Trip, who are his best friends. It is a great evening to remember old stories such as the wedding at which he was best-man, but soon the user finds out that his friends are passing a marriage crisis and expect him to pick a side of the conflict.

The story develops as the user interacts with his hosts in a three dimensional environment. Interaction is achieved by walking around the house (living-room and kitchen) and using some

interactive objects. It is also possible to interact with the other characters, which are behavior-based agents, either by using natural language text or by selecting special actions like kissing, comforting and hugging. The smooth and the effective interaction flow alongside a rhythmic story development are most certainly among the critical success elements of this application.



Figure 3.2: FAçaDE screenshot

The base of FAçaDE's success is the innovative narrative design approach taken by its authors. Instead of focusing solely on procedural generation processes for content generation and all its technological limitations, or on an extremely detailed graph and the combinatorial explosion that derives from this approach, they created a mixed solution of these two approaches based on the screenwriting concept of *Beats*.

According to McKee (McKee, 1997) a *Beat* "is an exchange of behavior in action/reaction. *Beat by Beat* these changing behaviors shape the turning of a scene". Mateas and Stern imported this concept to FAçaDE (Mateas, 2002; Mateas and Stern, 2003) and coined the concept of *Action Beats* as units of fragmentation of a story and all its possible story outcomes. *Story Beats* encapsulate story states with preconditions and effects, behaviors and discourse acts for the behavioral agents. The whole narrative representation is a sum of all the possible combinations of these elements according to the policies defined by the authors. One should note that *Story Beats* do not script exact action sequences but just define behaviors and goals for each given moment "offering a non-trivial simulation space" (Mateas and Stern, 2003). This way only one beat can be active at a given time.

To monitor the story evolution with all its elements (beat sequence, agent's behaviors and user interaction) and apply the author's policy FAçaDE's implementation includes a *Drama Manager*, that is responsible for activating the beat that better suits authorial intentions in each context. (see figure 3.3)

Whenever a beat successfully finishes or aborts, the *Drama Manager* receives as input everything that is happening in the story world, the past sequence of beats and user interactions, a cluster of beats that satisfy the current preconditions and a story arc with the desired tension values sequence for the story and chooses the next beat to activate according to the following algorithm:

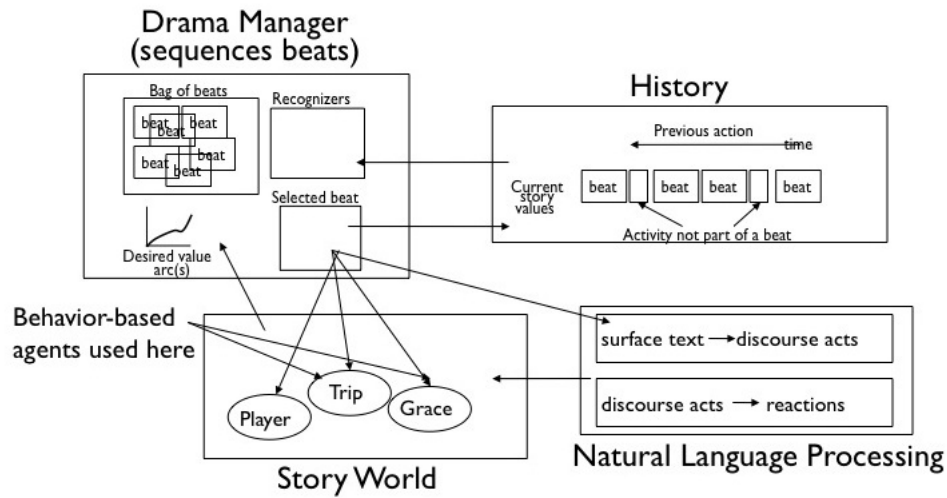


Figure 3.3: FAçaDE architecture

1. Initialize any beat-specific state that may play a role in beat selection.
2. Evaluate the preconditions for all the unused beats. This computes the *Satisfied* set with all beats with satisfied preconditions.
3. Evaluate the priority tests of each beat in *Satisfied*. Collect those with higher priority into the set *ScoredHighestPriority*.
4. Score each beat in the *HighestPriorityTest* using its effects to compare the beat with the desired story arc. This score evaluates the effects of a beat in a story variable named Tension. The result of this step is *ScoredHighestPriority*
5. Multiply each *Beats* score by its weight. Produces the *WeightedScoreHighestPriority*.
6. Select a beat randomly from the previous set according to a probability distribution defined in the weighted score.

During each *Beat* users interact with the environment and with the other characters with a high degree of freedom. Nevertheless, in what refers to *Agency* users are expected to act within character. There is no attempt whatsoever to integrate out of character actions into the story development, the user is not expected to add completely new elements to the scene, or to place himself in the center of the action. Instead, he can seriously influence characters' affinity among each other and that will have an influence on the scene tension and flow towards the next beat selection. Also, the expressive power of the natural language engine limits the user to written language and its timings.

Up until today, FAçaDE continues to be the most successful IS example ever disseminated. Its contributions go beyond the design, closing the gap between scientific IS and artistic Drama Theory, introducing some very important concepts such as *Action Beats* and *Story Arch*.

3.1.3 Discussion

In an overall appreciation we observe that plot-centered approaches have allowed a coherent integration of the user actions in the story development. In Mimesis, a reason behind this success is the separation between the story presentation and the story logic, while in Façade the authors succeeded in providing the successful integration between the artistic thought on story development, that results from the dynamic story structure provided by adapting the *Beat* concept to Interactive Drama. In both cases the users have the possibility of experiencing a better sense of agency, as long as they accept the story logic and do not try to diverge too much from the stories that they are invited to explore. Also, the continuity given to the Mimesis approach in several different projects, argues in favor of its adaptability to new story genres and its capacity to include storytelling techniques from other media.

On the other hand, the inclusion of user actions in plot-centered approaches typically results in a combinatorial explosion scenario. Authors are faced with the delicate balance between increasing authorial costs and reducing user influence. In Mimesis the authorial costs are diminished at the cost of the computational resources used to re-plan a new story plot, while in FAÇADE the authors report a total of more than two man year effort in the authoring process (Mateas and Stern, 2003).

We argue that the cost of adapting a story to the user actions can be heavily reduced by improving IS abilities to relate and interact with the user, i.e., including more realistic interactions between the user and the characters of a specific interactive storytelling system. Pre-determined plots, even when they are highly granular as in FAÇADE, are too restrictive and place the user in the position of exploring a pre-determined path instead of providing him with the illusion of being a true protagonists and the author of a story.

In the next section we will address Character Centered approaches and how the motivation of creating more human-like interactions and emotional consistent characters has contributed to the creation of emergent IS experiences.

3.2 Character Approaches to IS

Character approaches to IS are strongly motivated by the relevant impact that the interactions between characters have in a story development. This motivation requires for IS systems to be heavily populated by strongly designed characters, i.e., characters that embed behaviors are consistent with their roles, while at the same time are able to interact with other characters in a dramatical form.

The typical authoring process for a character centered approach follows recursive cycles that include three steps: character design, simulation and refinement (see Fig. 3.4). In the first step the author encodes each character's behaviors in the specificity of the AI architecture used in a multi-agent system. In the second step, the author runs a simulation to observe the agent's emergent

behavior. In the last step, the author evaluates how much the emergent behavior of the system represents the authorial intentions. The author then identifies the agent modifications needed and applies them in a new authoring cycle.

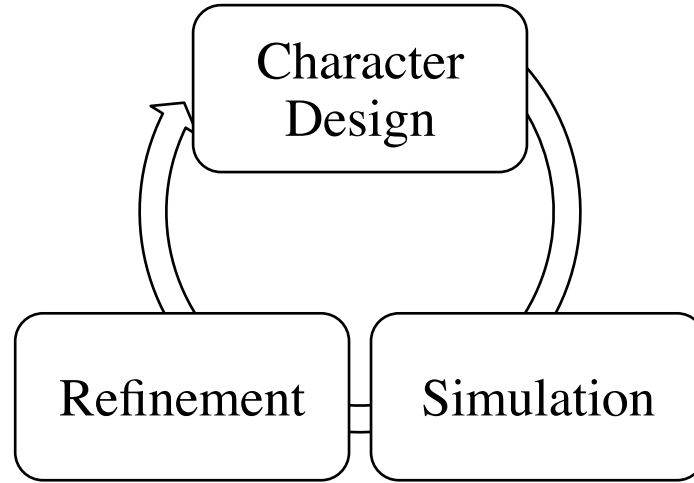


Figure 3.4: Character centered authoring

In this section we will present some relevant research examples that follow a character centered approach. For each example we provide a brief overview, its main contributions as well as the limitations that can be related to our research.

3.2.1 Interactive Storytelling Prototypes.

This character based approach (Cavazza et al., 2002), follows Michael Young's (Young, 1999) proposal that stories can be dynamically generated by the interaction between characters that use real-time planning systems, as long as the characters implement well defined roles. In this case each role is a character plan and it is from "on-stage" interaction between these plans that "situations of narrative relevance" occur (Cavazza et al., 2002). This idea establishes two very strict requisites for the character's planning algorithm:

- Consistency between plan definition, the character's role and narrative meaning.
- Real-Time adaptability to non-determined on-stage events. Characters should be ready to deal with unexpected events in the virtual world, specially those caused by user interaction.

Considering these requirements the authors decided to implement a planning system based on Hierarchical Task Networks (HTN) (see Figure 3.5). Each character goal is decomposed into alternative actions. Since each action has its own preconditions, one can define several ways of achieving a goal according to the environment conditions. This way, it is possible to define different sets of actions for each goal, creating a diversity of solutions. Nevertheless, this adaptability

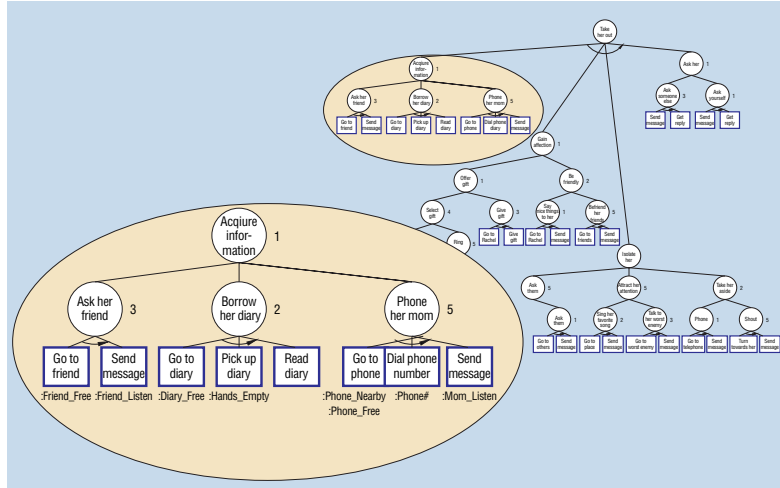


Figure 3.5: HTN example from Cavazza et al. (2002)

can generate inconsistencies in character behavior, as well as lead to blocking situations when a character is faced with an unexpected environment state for which there is no alternative plan.

When using these character/character interactions as a source of narrative relevant events, researchers use a bottom-up approach for story development, where characters act following their own plans. In such approaches, the character's interactions may produce undesirable story states, raising the need to develop strategies to overcome these states without losing narrative significance. This research includes two strategies to overcome this challenge: *Situated Reasoning* and *Action Repair*.

Situated Reasoning happens when the world conditions do not correspond to the character expectations to perform a task without causing it to fail. In such conditions, the character maintains its initial plan but finds a way of doing it with the actual conditions. If, by any chance, the initial plan of the previous character fails, he drops the failed plan and creates a new one. This is called *Action Repair*. An example shown in (Cavazza et al., 2002) is a character (man) that wants to perform a secret task (know more about the woman he likes) and decides to go to her room and read her PDA. When he is moving to the room he sees her moving toward him. Using *Situated Reasoning*, this character finds a way to be unseen, waits for the other to go away and resumes plan execution (walking through the corridor). If when performing the secret task, he notices that it is impossible to execute it (someone, maybe the user, has hidden the PDA and there is no way to find it), he builds a new plan (talk to a friend to get information about the girl), which is *Action Repair*.

In these prototypes the user is invited to watch a generated story in a interactive way, i.e., the user can influence the story directly in an third-person perspective, without taking any role as a character. The user interacts with the system by manipulating objects in the virtual environment, and by giving advices to characters using speech.



Figure 3.6: Screenshots of different Interactive Storytelling Prototypes.

This particular approach to IS has been applied in several scenarios based on works from other media, such as *Friends* (Cavazza et al., 2002) inspired in the homonymous sitcom, *Charbovary* (Cavazza et al., 2007) based on Flaubert’s novel of *Mde.Bovary* and *Merchant-of-Venice* (Julie Porteous, 2010), based on the homonym Shakespearian tragic comedy, each of them with specific contributions to research (see Fig. 3.6).

While *Friends* implemented the basis of this approach, the other two add new functionalities to the planning algorithm. *Mde.Bovary* included a representation of the affective state of the characters throughout the story, which was based on original annotations of Flaubert himself. Although there is no appraisal mechanism associated to this implementation, the values of the emotions are used to constrain the planning algorithm in order to maintain the characters evolution consistent with the authorial intentions.

The *Merchant-of-Venice* (Julie Porteous, 2010) prototype introduced the concept of *character point-of-view*, which consisted of a complementary higher level description of the story corresponding to the characters intentions in the plot. By switching between character’s perspective in a story the systems ends up generating different story developments and increasing story variability.

3.2.2 Thespian

Thespian (Si et al., 2005b, 2006) is a framework for authoring and simulating interactive-narratives. Until 2009, Thespian had been used to author more than thirty interactive narratives (Si et al., 2010). It has been applied in the creation of Interactive Pedagogical Dramas (interactive narratives with learning goals for the user) (Si et al., 2005b), particularly in foreign culture training scenarios for the military (Si et al., 2005a).

Thespian’s most relevant contributions to IS are related to the agents’ architecture and to the authoring process. The agents are built upon a multi-agent framework for simulating social-interactions called PsychSim (Marsella and Pynadath, 2004), which includes a Partially Observable Markov Decision Process (POMDP) to control the characters in a story and a theory-of-mind model, which allows agents to include models of others in their own decision process. Furthermore,



Figure 3.7: Screenshot from a Tactical Language System supported by Thespian. (from (Si et al., 2005a))

Thespian’s authoring process includes an authoring tool for identifying and refine critical plot sequences using a fitting algorithm. In the following we will summarize Thespian’s contributions.

Thespian Agents

The agent architecture implements a POMDP approach to control the characters. Each character’s personality and motivations are encoded as agent goals. An agent includes five main components:

- *State* - set of features representing the agent’s true state. It includes numeric representations of state features (e.g., age) and numeric representations of relation features between the agent and other agents (e.g. affinity).
- *Action dynamics* - define the impact of an action in a state. The impact can be defined in probabilistic terms, e.g., the author can define that when hunter shoots a wolf, the wolf dies 60% of the time.
- *Goals* - goals are implemented as reward functions that convert each world state into a numeric value. PsychSim uses two kinds of goals that attempt to minimize/maximize different things. *Feature goals* concern the value of a specific agent feature, while *Action goals* concerns the number of times a specific action is performed.
- *Policies* - define how agents select the better action in each state. The default policy used in Thespian is a bounded lookahead policy, by which agents simulate candidate actions and select the one that generates the most rewarding outcome. Nevertheless, agents can be configured for multiple levels of lookahead by which they can define a cycle in which after simulating the effect of their actions, they can simulate the effects of the actions of others and then simulate their actions on that new level and so fourth.

- *Beliefs* - an agent's subjective view of the world consists of his own state and a probability distribution of the elements defining the state of other agents. Each agent has a model of itself and one or more mental models for each of other agents (theory-of-mind) which are calculated by applying Bayes theorem. More details on the belief revision process that supports this component can be found in (Si et al., 2006).

The belief model and policy system have been extended to include more complex and realistic social behaviors. Examples of this are the social norms models presented in (Si et al., 2005b, 2006), used in a IPD scenario to train military officers in inter-cultural interaction. In these examples the state features were authored to represent critical social variables, such as trust, liking relationships, affinity, freedom. Also, the action dynamics and goals were tailored in order to manipulate and consider the previous referred variables. By using these elements in their behavior along with their theory-of-mind implementation, the agents are able to simulate the trust and other relational issues that are present in the relations between citizens of an occupied country and the military that seek information about the local armies.

Proactive Authoring

Authoring a PsychSim agent to play a specific character requires the definition of a large number of state variables, dynamics and goals that must be tuned in order to produce the desired results. Moreover, the inclusion of user actions in an interactive narrative adds a high level of unpredictability to the unfolding story paths, which escalates the authorial efforts. In order to facilitate this process, Thespian framework includes an authoring tool that facilitates this process. The authoring tool considers a recursive authoring cycle that follows four steps (see Fig.3.8)

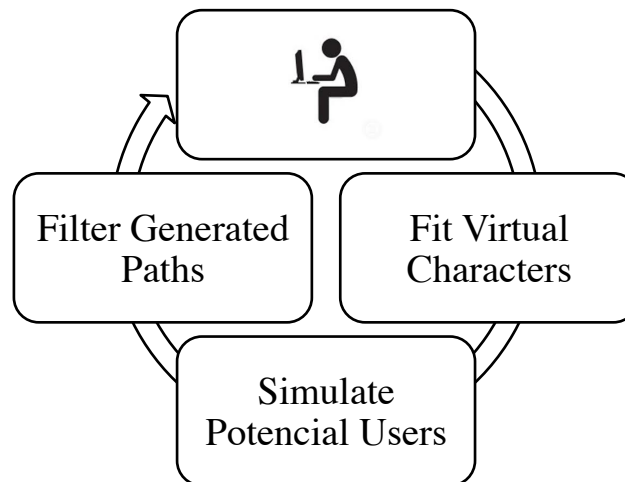


Figure 3.8: Proactive authoring cycle, adapted from (Si et al., 2007)

First the author defines the story state, goals and dynamics for each character, as well as the desired story paths including actions and speech acts sequences. In the second step Thespian runs

a fitting algorithm that uses the story path as a reference for producing a series of suggestions for configuring the agent's personalities. The suggestions consist on modifications to initial goal weights and the detection of goal conflicts (i.e. two or more goals that can not all be satisfied simultaneously), in such case the author must either correct the story path or the characters' configuration. Afterwards, the system simulates the users behaviors. In this step the user is assumed to be *well motivated* "have consistent motivations throughout the story" (Si et al., 2007), and modeled in a psychSim agent. The system can simulate different users by applying different mental models to the user agent. After generating all story paths, the system proactively filters story paths that although might be consistent to the author intentions, represent problematic sequences. The heuristics applied in this filtering use a percentage value and an integer value given by the author, to detect story paths in which an agent's behavior is repeated in a higher percentage or a number of times higher than the given values. The authoring cycle is concluded when the author reviews the story paths presented and eventually corrects them in order to start a new fitting step.

3.2.3 FearNot!

FearNot! was "developed to be used as an educational tool in schools to promote awareness about bullying behavior in schools." (Figueiredo et al., 2008) In this IS system the user takes on the role of a friend of a bullying victim to whom the user is asked to give advices from time to time.

From a narrative perspective, *FearNot!* is a sequence several instances of two different kinds of episodes:

- Interactive Episodes: where the user interacts with the victim autonomous character. The user provides advices using a written natural language interface. This interaction affects the autonomous character's emotional state and goals and will consequently affect its behavior.
- Emergent Non-Interactive Episodes: where autonomous characters take on the roles of bully victims, bullies, defenders and bystanders in the performance of typical bullying situations. The story unfolds from the interactions between the characters. One should notice that the user influence in the non-interactive episodes is achieved by the effects of his influence over the *Victim* in the interactive episodes.

The action begins with one or more non-interactive episodes that present the initial story setup to the user. Following this the narrative extends through a series of sets of one interactive episode followed by one or more non-interactive episodes.

The autonomous characters are implemented in FATIMA (Fearnot AffecTive Mind Architecture) (Dias and Paiva, 2005), an affective architecture for autonomous agents based on the OCC model of emotions. The autonomous agents are responsible for the emergent behaviors and for maintaining an emotional consistency of the characters. In order to guarantee that the emergent

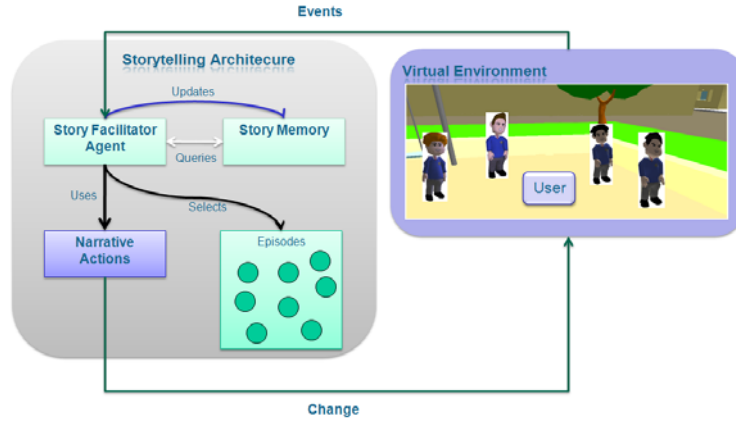


Figure 3.9: Facilitator architecture (Figueiredo et al., 2008)

behaviors comply with the educational goals of the project, FearNot! implements a metaphor for an educational role-play facilitator called *Story Facilitator* (Figueiredo et al., 2008). The purpose of the *Story Facilitator* is to mediate the possible conflicts between the emergent behaviors and the educational goals by selecting the sequence of episodes that are presented to the user.

Story Facilitator

Story Facilitator is a metaphor that refers to educational role-play. In educational role-play participants experience different perspectives of a social issue by playing the perspective of others. The action is monitored by a tutor (*facilitator*) who intervenes in the story in order to guarantee that the educational goal is achieved without breaking the role-play experience of the participants. Usually this is achieved by separating the story into episodes, allowing the *facilitator* to provide short representative backgrounds for each player character or add new elements to the scene at the beginning of each episode.

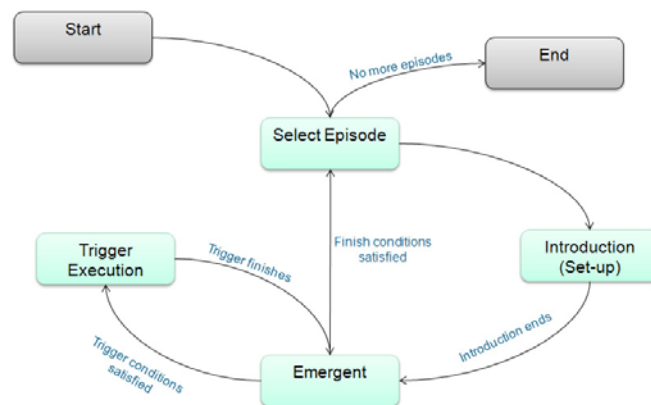


Figure 3.10: Story facilitator cycle (Figueiredo et al., 2008)

There are four major stages considered by this architecture in order to manage the episode sequence for each story: *SelectEpisode*, *Introduction (Set-up)*, *Emergent*, *Trigger Execution*. (see

Figure 3.10). Just before initiating a story, *SF* loads all episodes for the story and puts them in the *Story Memory* ready to be used. Afterwards, at *Select Episode* stage, the *SF* searches the story memory for the initial episode and marks it so it will not be selected again. At *Introduction (Set-up)* stage the *SF* executes all the *Narrative Actions* specified in the selected episode's set up. It loads the scenario, stores each character's properties into *Story Memory*, selects their goals and sends them to the corresponding autonomous agent that play the character, as defined in the *episode*. Following this, at *Emergent* stage, the autonomous agents that play the *episode* roles take full control of the action, letting the story unfold from their interactions. During this stage the *SF* monitors and stores all action events in *Story Memory*. Whenever the *Story Memory* state satisfies any author defined conditions in the *Episode* it triggers the *Trigger Execution* where the *SF* executes the actions defined by the author. Afterwards, the execution returns to *Emergent* stage until another trigger condition is satisfied or the *Episode's* finish conditions are satisfied, in which case the execution goes back to *Select Episode*, where a new *episodes* will always be initiated as long as there is any episode with satisfied pre-conditions, otherwise the story ends.

FATIMA Autonomous Characters

FATIMA (Dias and Paiva, 2005) is strongly based on OCC cognitive theory of emotions (Ortony et al., 1990), where emotions are defined as valanced (good or bad) reactions to events. The assessment of this relationship between events is called the appraisal process. In order to achieve believable and expressive agents, their behavior is influenced by their emotional state and personality. FATIMA includes a reactive level that provides a fast mechanism to appraise and react to a given event and a deliberative level takes longer to react but allows more complex goal-driven behavior.

In order to build agents in FATIMA one has to define the actions available for the domain (they will be used by the planner in the deliberative layer), and then to individually define each of the characters. The character's personality is strongly based on OCC and is defined by: a set of goals; a set of emotional reaction rules; action tendencies; emotional thresholds and decay rates for each of the 22 emotion types defined by OCC.

The emotional reaction rules assess how generic events are appraised and represent the character's standards and attitudes. The rules are used to influence interpersonal relations that are also modeled in FATIMA. For instance, if an agent performs an action that triggers negative emotions in another agent, the relation of the latter with the former will deteriorate. These relations are stored explicitly in the agent's model of the world and can be used to activate goals and other type of behavior. Action tendencies correspond to the reactive behavior and are simple action rules triggered by particular emotions. For example, we can have a character crying when very distressed.

For each emotion there is an emotional threshold and a decay rate. The threshold corresponds

to the character's resistance towards an emotion type, when an event is appraised, the created emotions are not necessarily "felt" by the character unless their appraised value is higher than the threshold. The decay value determines how fast does the emotion decay over time. In addition to goals, standards and attitudes, these emotional thresholds and decay rates are used to complement a character's personality. For example, a peaceful character will have a high threshold and a strong decay for the emotion type of *anger*, thus its anger emotions will be short and low. Thus, it is possible to have two characters with the same goals, standards and behaviors that react with different emotions to the same event (by having different thresholds).

Current research using FATIMA is motivated by two complementary perspectives. The first is to improve agent-agent and agent-human interaction believability by including a model of cultural behavior based on rituals (Mascarenhas et al., 2009), a model motivational drives (Lim et al., 2010) and a representation of personality traits (Doce et al., 2010). The second motivation is to endow FATIMA agents with an actor perspective over the character performance. This perspective has produced a Double Appraisal model (see Section. 3.2.4), a model for endowing agents with the awareness of the dramatic impact of their actions in a story. Also, in this thesis (see Section. 6.2) we will present a case study under this motivation that endows FATIMA with the ability of acting an emotional escalation in comic sketch scenario.

3.2.4 Double Appraisal

Double Appraisal(DA) (Aylett and Louchart, 2008) approaches IS with the motivation of endowing agents with the ability of reasoning over the impact of their own action in a story. "Rather than selecting the action with the highest value for the character state after appraisal, the one with the highest emotional impact is chosen" (Louchart and Aylett, 2002). DA extends FATIMA (see Section 3.2.3) and relies on its emotional elements to sense the dramatic intensity of a scene. Using a simulation theory, DA agents run parallel appraisal cycles in order to evaluate the possible emotional impact of their actions in other agents in order to relate it with the dramatic impact of their actions.

This approach includes three implementations, Reactive DA, Deliberative DA and DA with modeling(DAM):

- Reactive DA - instead of a single action, the appraisal output of the character produces a set of valid elements, consistent with the character role, ranked by emotional intensity. It then reappraises each of these elements in simulating their impact if they were targeted towards him. After running the second appraisal cycle with each one of the previously selected actions, the system chooses the action with the greatest emotional impact and executes it.
- Deliberative DA - in this case the main emotional intensity is used to select a plan that executes a character intention rather than a single action. Because each plan execution

consists in several sequences of actions, evaluating them through the appraisal of each one of their elements could easily result in a serious combinatorial explosion. To simplify this, the authors selected the action in the plan that finally satisfies the plan's post-condition, as the representative action.

- DAM - Is an extension of the early DA models that instead of appraising the emotional impact in a simulation where actions are executed towards himself and appraised with his own emotional state, uses a representation of the actual emotional state of each one of the other characters. After running this simulation the system chooses the single highest emotional impact for any of the agents in the scenario.

The system was evaluated in a text base scenario. Five agents played 5 different roles with intrinsic and conflicting goals, while another agent played as game master(GM). Five different kinds of stories were generated, each of them corresponding to a particular variation of the system. The resulting stories were ranked by users / readers according to their interestingness, and the events in the stories were marked according to their meaningfulness and dramatic intensity.

The most significant result of this evaluation is that the participants reported DA/DAM stories(Story 3,4 and 5) as being significantly more interesting than the story based on the original FATIMA architecture. As for DA vs DAM models the differences were marginal and consequently inconclusive. The authors argue that this fact might be the consequence of the simplification of the intention plan and of the short length of the scene and number of interactions.

3.2.5 Discussion

Character centered approaches have successfully contributed for integrating more believable behaviors and interactions in IS. Thespian and FearNot! achieve this by incorporating psychological models in their agents' minds that allow for emotional and social consistency. Nevertheless, this contribution continues to be made at the cost of authoring effort. Thespian successfully reduces part of this cost by implementing an authoring tool, but still the large amount of features that is required for authoring each agent continues to present a very hard challenge for the authors.

By failing to reduce authoring costs the developers choose to reduce user interaction power. In the Interactive Storytelling Prototypes the user interaction is very limited to the interaction with environment objects and with the expression of advices to some characters. We see this as a strategy for preventing the user from generating undesired story states that could callout the characters limitation to act beyond the pre-established behaviors defined by the author.

FearNot! disguises this issue by including the interactive and non interactive episodes mechanic consistent with the story fantasy. While the user interacts "in character" with the main character in an interactive episode, he is influencing the system selection of story episodes and also influencing characters starting emotional states in each episode. There is a well defined boundary between

interaction mode and emergent mode that allows an easier management of the user/author conflict. Nevertheless this is achieved by interrupting interaction mode, and consequently breaking the user sense of agency, whenever the user wants to intervene an emergent episode or fails to express his opinion in the textual interface.

Double Appraisal is an example of how character centered approaches are thriving to move out of the autonomous character paradigm to the boundaries of authorial-agent. The good results of DA and DAM vs. non DA are a solid example of the possibility of generating more interesting stories even when control is delegated to its participants. It also defines a new technique for selecting dramatically intense actions, that can be generically applied to any system populated by autonomous agents with cognitive appraisal mechanisms.

Nevertheless some important remarks should be considered. One could argue that the existence of a Game Master in the test scenario contradicts the essence of the character centered approach, because the story creation is still shared with a non-character. We consider that while the simple existence of a GM may contradict this essence, the characters still gain a significant autonomy that can not be discarded as an emergence factor. This argument can also be extended to the need for including a facilitator in *FearNot!*.

Another debatable aspect in DA is the principle that the most dramatically intense action available is always the best choice for story interestingness. It is known that there are different story moments and that choosing the right timing for each dramatic action is one of the hardest tasks that screenwriters and performers face. An example is overacting. An actor who overacts fails to deliver dramatic intensity to an audience, because his exaggerated emotional state leaves him with no options to buildup a tension process for the audience to follow. In the DA scenario the overacting issue does not occur because agent actions are conditioned by its current goals, hence conditioned by the authorial influence that defined his goals. The agent has no model defining his perspective over story development, hence he has no explicit intention of manipulating the story development. Instead, he is reasoning over a limited spectrum of actions available to his character at a given cycle based on a premise that the story is constantly asking for maximum dramatic intensity. A good improvement of this approach would be to allow agents to reason about dramatic intentions in a story. We will detail this argument in the motivation for the Emotional Escalation prototype (see Section 6.2).

The motivation towards reducing the authorial effort on character centered approaches clearly establishes two main research directions for the next few years. In one hand, IS will clearly benefit from research on cognitive and social intelligence models. The above mentioned examples, particularly FATIMA and Thespian, demonstrate that character centered approaches are good application field for such areas, as they contribute to increasing the believability of the interactions within an IS system. Such contributions contribute to the reduction of authorial costs in unexpected story contexts, by providing agents with more autonomous and believable interaction abilities. The

other direction is given by the attempt to endow agents with authorial abilities, such as explicit intentions over story development. In the next section we will see how improv has been influencing this research direction for IS.

3.3 Improv in IS

IS and Improv share the same common goal of creating stories in a collaborative way within an unpredictable environment. This common goal has already led some significant research to use Improv as a source of inspiration for IS. In this section we present some emblematic examples of such approaches.

3.3.1 IMPROV at NYU

In the late 90's the NYU's Media Research Lab conducted the IMPROV research project under the direction of Prof. Ken Perlin. This research focused on the development of three dimensional environments for human interaction, through avatar control, with "computer-controlled agents" (Goldberg, 1997). This research pioneered in several areas and contributed to several research topics for non-player characters, such as procedural animation, behavioral scripting and human-computer interaction. On the realm of interactive storytelling, IMPROV considered the development of autonomous agents with improvisational skills (virtual actors) as an opportunity for adding adaptability to virtual environments and reducing the effort of creating pre determined story content.

A common concern in all sub-topics of this research, was believability. Following a believability perspective based on the suspension of disbelief principle, the aim was not to create realistic virtual characters that completely imitate real life characters, but rather to create characters to which users can relate to (Perlin and Goldberg, 1996), characters about whom a user can perceive an internal personality, attitudes towards new circumstances or any given situation (Perlin, 2003).

At a technical level, "IMPROV is a system for the creation of real-time behavior based animated actors" (Athomas et al.), which is divided in two main components:

- Animation engine - focused on delivering procedural animation techniques to allow authors to create layered non-repetitive animations, that can be blended in soft transitions.
- Behavior engine - enables authors to define rules for generating character behaviors on how to communicate and take decisions.

The results produced by both components and its integration are still relevant references in current research, nonetheless because of its direct relation to this thesis work, we will give a particular attention to the behavior engine.

Following (Athomas et al.), the user is a system variable that prevents authors of interactive settings to create deterministic scenarios. All user actions towards an actor convey a particular user expectation that must be fulfilled by the actor when selecting a new action. This behavior will generate another unexpected response from the user. This is the main argument that led IMPROV research to enable authors with more than a tool for creating linear action sequences, but rather with a tool for creating layers of choices in the form of global and slowly changing plans that could rapidly adapt to changes in the environment.

Plans are encoded as small scripts for very specific situations, such as, *greeting someone* or *fidget*. Whenever an actor identifies a context, it loads the behavior script that corresponds to that specific context. One can see that this process generates the repetition of behaviors whenever an actor encounters a similar context. To prevent this robot-like behavior, and consequent loss of believability, the scripts include *tunable statistics*. *Tunable statistics* consists on enabling variability to a behavioral action or activity. When creating a script the author can associate different actions to the same task and associate these actions to values, e.g. Fidget - Twiddle thumbs 0.7, Scratch head 0.3, Pick nose 0.2 (Goldberg, 1997). Using this script format the agents can plan weighted decisions about the actions to perform, that will make that some options will occur more often than others.

Tunable statistics were extended to add variability to activities, and for defining the importance of particular skills for executing a particular activity (e.g. Difficulty .32 for Riding a Bike) or the effects of an action towards others (Belch can have an effect of setting other-actors sympathy towards the character to .01). Another use of tunable statistics, and perhaps the most relevant to this thesis, is that it allows to define personality traits for each character. These personality traits, such as, intelligence, amiability, strength and others allow the agents to select the scripts and action that more accurately convey the characters' personality. This selection is made using a fuzzy logic algorithm, that considers the distance between the personality and other character variables to those associated with the actions to perform, and weights these values before making a decision. This way the system can achieve a balance between character consistency and believability (through variability).

3.3.2 Directed Improvisation

The paradigm of Directed Improvisation presented by Hayes-Roth et al. consists of a constrained real-time creative performance "without detailed preparation and often by making use of the resources at hand" (Hayes-Roth et al., 1995). The test bed that implemented this paradigm was called Computer-Animated Interactive Theatre (CAIT). It followed two complementary approaches Improv Puppets, Improv Actors and also Mixed approach that combines these two. These approaches are based on the general assumption that "all acting is improvisation" (Applebee, 1980), and that there is a spectrum of different improvisation approaches according to the directorial influence (see

Autonomous Improvisation	Cooperative Improvisation
One-Step Improvisation	
Choose among alternative logically equivalent behaviors	Respond to a partner's behaviors
Sequential Improvisation	
Construct a coherent path to a dramatic moment	Interact with a partner's behavior sequence
Patterned Improvisation	
Instantiate an improvisational schema	Recognize and participate in a partner's schema

Table 3.1: Improvisation forms, adapted from (Hayes-Roth and Gent, 1997)

Figure 3.11).

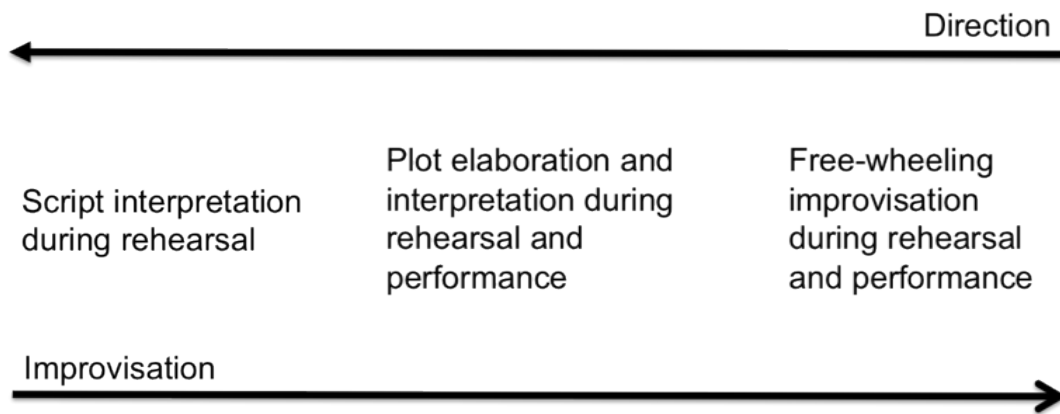


Figure 3.11: Spectrum of Directed Improvisation

The directorial influence is described using the spectrum of Directed Improvisation. This spectrum is defined by two symmetrical axis that represent Direction (directorial control) and Improvisation (the freedom to improvise), and two distinct moments on a play development (rehearsal and performance). At the Maximum Direction/Minimum Improvisation end are the plays that take advantage of interpretational freedom during rehearsal but follow rigid error recovery references in performance time. At an intermediate level are plays in which the plot is defined during rehearsals and in which the actors are allowed to freely interpret their characters inside the boundaries of the rehearsed plot. At the other end of the spectrum are plays in which actors are totally free to improvise everything in rehearsals and in performances.

Another important fundament for this research concerns the application of three different levels of complexity on two different forms of improvisation: *Autonomous Improvisation*, with a single performer; and *Cooperative Improvisation* with multiple and coordinated performers. (see Table 3.1)

Based on the Spectrum of Directed Improvisation Hayes-Roth et al. defined two different approaches to Improv Agents:

- Improv Puppets: follows the traditional puppets concept of serving as passive vehicles for story making, except that in this case Improv Puppets are animated synthetic characters that can be directed by the user in terms of verbal and physical behaviors in real-time (Hayes-Roth and Gent, 1997).
- Improv Actors: the synthetic characters act autonomously with preconceived "behavior scripts" that "synchronize sequences of physical and verbal behaviors to be performed". (Hayes-Roth et al., 1994)

The main difference between the two approaches is that in *Improv Actors* user control is strictly confined to pre-performance time. In the next section we will address some relevant technical aspects of the implementation of these two paradigms.

Improv Agents implementation distinguishes three major modules: Mind, Mind-Body Interface and Body. (see Figure 3.12)

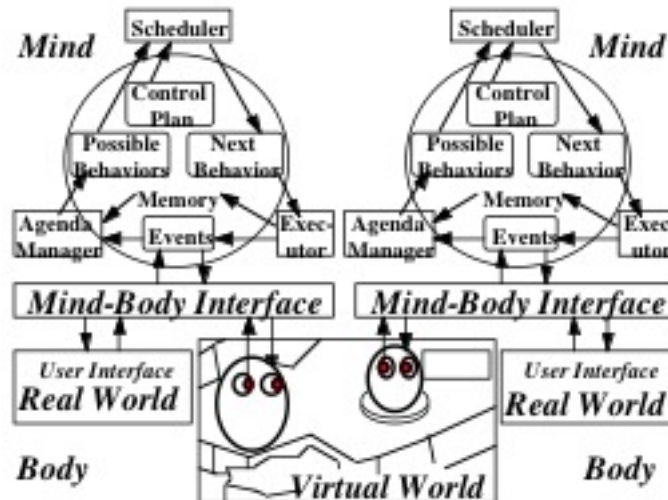


Figure 3.12: Logical system organization for two agents in virtual world.

The *Body* module is responsible for controlling the graphical representation of the agent. Among others it has the aim of controlling body transformations, gaits and providing an interface between world events and the *Mind-Body Interface* module.

All the sensorial input received by the *Mind-Body* interface is filtered and classified in this module before being transmitted to the *Mind* module. In the opposite data direction, it is also in this module that *Mind* outputs are converted into instructions for body effectors.

The *Mind* module iterates a three-step execution cycle that begins in the *Agenda Manager* component. This component uses recent perceptual and cognitive events, knowledge and state information to select behaviors that are possible and relevant in the present situation. In the second step the *Scheduler* selects one of these behaviors for execution according to the constraints defined in the *Control Plan*.

In order to help and better contextualize behavior selection, the system implements a representation of *mood* with three dimensions: *Emotional* (Happy-Sad), *Physiological* (Peppy-Tired) and *Social* (Friendly-Shy). *Mood* is used to categorize behaviors and to select them according to the dynamic *mood* of the agent's internal state.

As a result of the use of *mood*, the *Mind* is able to send new instructions to the interface not only when a new set of relevant behaviors are identified but also whenever it experiences a *mood* change.

Using status in a "Master and Servant" scenario

"Otto and Gregor" (Hayes-Roth et al., 1996, 1997) is a master vs. servant scenario inspired in a traditional improv exercise where players play a Boss that wants to fire his Employee. The interesting element of this scenario is that its development is influenced by the combinations of character statuses, e.g., in a situation where the Boss plays an extremely lower status than his Employee it may happen that the Boss not only ends up not firing his Employee but even with the Employee demanding a raise.



Figure 3.13: Otto and Gregor in a master-servant scenario.

In this scenario the original "Improv Actor" architecture was extended to include the notion of status. Following Johnstone (Johnstone, 1987) status is present in three dimensions: demeanor, relationship and space. High status in demeanor is perceived through erect postures and graceless as opposite to low status in demeanor that leads to awkward postures and graceless; high relationship status is directly related to authority the same ways that low status in this dimension is related to acquiescence; the space dimension is related to the way each actor can occupy space, the higher the status the more an actor can occupy space as opposite to low status actors try to avoid intruding upon space.

3.3.3 Digital Improv

In the previous sections of this chapter we addressed IS research projects that rely on hypothesis driven from observational information about improv. The Digital Improv Project (Baumer and Magerko, 2009; Magerko and Riedl, 2008; Magerko et al., 2009) takes a step forward by relying on an empirical understanding of the cognitive and social phenomena of the improv creative process. This research uses its own cognitive studies of improvisational theatre with real life improvisers.

In this research the working definition for *improv* is defined as *"the creation of an artifact and/or performance of aesthetic goals in real-time that is not completely prescribed in terms of functional and/or content constraints"* (Magerko et al., 2009). There is a special attention to the creative process itself. This approach is sensitive to the fact that the study of a creative process involves a wide variety of research fields, from which, an even wider range of research issues emerge. The need for creating well delimited scientific contributions in such a rich research landscape is the underlying motivation for using a bottom up approach, since it allows to isolate elements and contributions.

This bottom up approach starts with the cognitive study of real life improv actors. The general goal of the cognitive studies is to extract explicit data from the actors creative process. Once analyzed, the data can be used to ground specific solutions for problems that arise from the goal of creating micro-agents with improv abilities, i.e. agents that implement a specific improv ability.

In the following subsections we will present the empiric study approach that is being taken by this project to study the improv phenomena, the theory domain that resulted from the first analysis of data, the generic decision cycle generalizations and its fuzzy knowledge structure that was a main source of inspiration for the work presented in this thesis.

Empiric Study Approach

The goal of the empiric studies is to raise data to support a better understanding of the improv phenomena and the implementation of computational agents that apply improv techniques. To do this the Digital Improv project applies a retrospective protocol by which improv actors are invited to self-report their experience in pre-recorded sessions. This approach is supported by its non-pervasive nature. Since Improv already involves extremely complex social interactions, the use of pervasive analysis techniques, such as interrupting a scene, would contribute to distract the subjects and affect the quality of the results. Furthermore, the quality of the self reports can be considered of high fidelity, because actors are usually very experienced in self analyzing their works and performance since it is one their most important training strategies

The retrospective protocol includes two main steps: observation and data coding. The observation step is implemented as follows. Improv troupes are invited to perform in an environment specially prepared to accommodate improv exercises in front of a small audience. The space is fully equipped with video and sound recording devices that do not interfere with the performance area.

All participants are over 18, have at least some experience performing improvisational theatre and are reimbursed monetarily and with dinner for the evening sessions.

The observation consists on 4 stages: *Pre-Performance*, *Performance*, *Individual Retrospective Protocol* and *Group Interview*:

- **Pre-Performance** - participants are asked to fill out a questionnaire about their history, relationship with their troupe and level of expertise. Expertise is measured in terms of specific metrics, such as training, period for which they have been doing improv and professional experience (paid work).
- **Performance** - participants perform three kinds of improv games: Games with no content restrictions; with low content restrictions; and high content restrictions. Other games included Party Quirks and Film and Theatre Styles, which are common improv forms.
- **Individual Retrospective Protocol** - each participant participates in a private interview, where he is asked to remember and report his participation while watching a video playback of the group performance. In this step the participants are invited to report what they were thinking at each relevant moment of the scene.
- **Group Interview** - follows the same approach of the individual interview, this time with the whole group. This step is extremely useful for the actors to confront their experience with the one of their colleagues, detect misunderstandings and the processes by which they reach agreement.

The data coding consists on identifying relevant data in the combination of the interviews and performances videos. The goal is to establish a data driven approach to induce a working theory of cognition in improvisational theatre (improv Theory Domain) that can be used to determine high level categories for the data analysis.

Early analysis of the coded data produced two main contributions that we will summarize in the next sections. The Improv Theory Domain that isolates the computational challenges presented by improv and the definition of a decision cycle to be applied in an improv agent. We will also detail how these contributions supported the development of a fuzzy knowledge structure that was used to implement an improve agent that in party quirks scenario.

Improv Theory Domain

From the analysis of the data collected Magerko et al. (2009) defined a domain theory on improv cognition detected in the experiments that is summarized in Fig.3.14. This theory summarizes the cognitive challenges presented by improv.

Four different kinds of *Basic Cognition* were elicited in the studies:

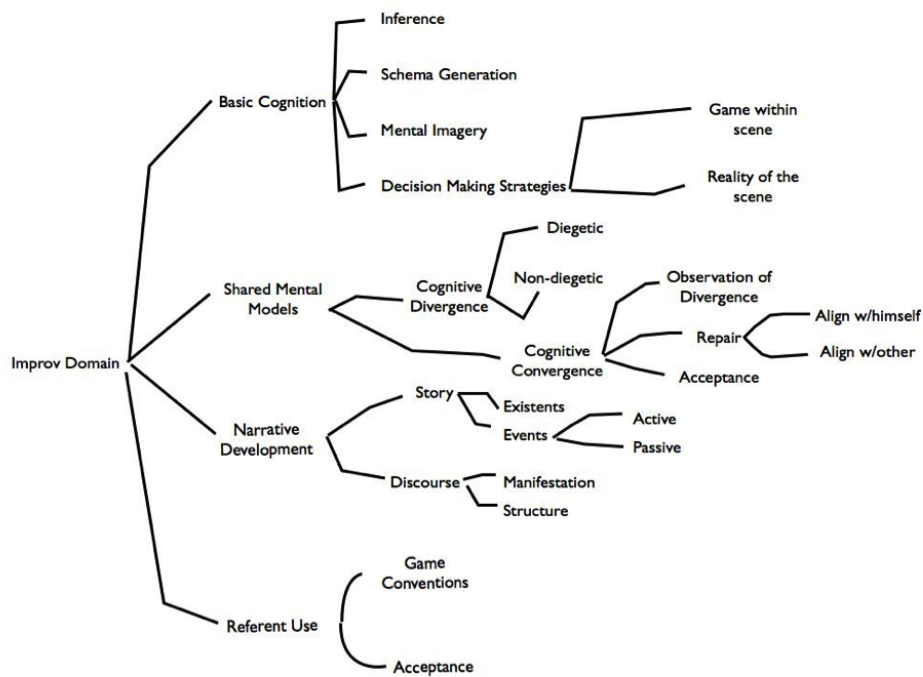


Figure 3.14: Improv domain theory map

- **Inference** - frequent reports of performers inferring information about the scene from other performers actions, more specifically inferring about scene location, improviser's goals and knowledge.

- **Schema Generation** - participants reported the use of schemas for character's features, activities and behaviors.

"I was just being a ridiculous caricature of my friends that are obsessed with video games... kind of anti-social..."

- **Mental Imagery** - in general, improvisers visualize their environment and consider how it would affect their actions.
- **Decision-Making Strategies** - Strategies used to take decisions upon story development.

Game Within Scene - strategies that use rules that emerge from the story. In this analysis the authors identified two games:

An example emerged for the selection of a particular kind of character prototype. Ex: *"I'm going to be the guy whose opinion differs from them"*. This game developed action around the continued opposition between the characters. Another example emerged from the reality of the scene. What happened when an actor was playing an invisible man eating chips? Another actor screamed Ex: *"Flying chips!!!"* pointing to the chips that were being

eaten by a player portraying invisible man. From this moment on the invisible can play with the others by scaring them with flying objects.

In some cases the subjects reported their difficulty in sharing their perspective of the scene. These reports were included in the *Shared Mental Models* sub-domain that includes two opposite states:

- **Cognitive Divergence**, occurs when the individual mental models of the characters differ from each other. Two different cases were observed:

Diegetic - when divergence is about facts of the story world. This is probably the most common form of cognitive divergence in Improv

Non-Diegetic - when divergence is out of the story world. An example may be a misinterpretation of the rules of an improv game by one of the players.

- **Cognitive Convergence** - "the process of establishing cognitive consensus" (Baumer and Magerko, 2009), although consensus may never be reached either because performers give up, or refuse to accept the mental model of actors by stubbornness or intended comedic affect. Normally the convergence process follows 3 steps:

1. Observation of Divergence
2. Repair - Align with other or with himself
3. Acceptance

Narrative Development, focuses on the processes and elements used to create a story in the observed presentations.

- **Story** - the content in the narrative

Existents - all sorts of props, people and spaces

Improvisers define characters by how they develop traits (e.g. relationships, goals, history, physical and mental attributes). Improvisers also report to keep track of the environment elements (e.g. location, objects, environment properties, etc...)

Events - what occurs in a narrative. According to (Ryan, 2006) there are 2 two kinds of events:

Active that affect the story state

Passive don't affect the story state

- **Discourse** - how the story is presented to the audience. This includes two dimensions:

Manifestation (the medium in which the story is expressed), this is not much relevant for this research since it only considers stage presentations.

Structure - narrative organization, that relates the story with the sequence of events and other presentation elements. In improv the narrative is created and presented at the same time in an offer / response process.

The last group of improv challenges corresponds to *Referent Use*, which are the references that improv players naturally bring to an improvisation. Two different types of referents were identified in this study:

- Past experiences - specific language elements or behaviors that are embedded in the actors acting by its training and previous performances.
- Game rules and conventions - the conditions imposed by improv training exercises or performance genders.

Decision Cycle

Another contribution of this work is the analysis of the improv decision cycle (see Fig. 3.15). This cycle starts with the improvisers reception of new Input(I) that can be either visual, audio or any other kind of stimuli derived from scene development and its participants. The second step of the cycle is the Creation step that encompasses the process of extracting knowledge(K) from the input, perceive the story state(S) and select possible events to add to the story state. Afterwards the player performs a Selection of the event(E) to be executed. Besides originating a space of rejected states(R) and a new story state S1 the execution of E is receive by the other performers as an offer, which is a new input for their decision cycle.

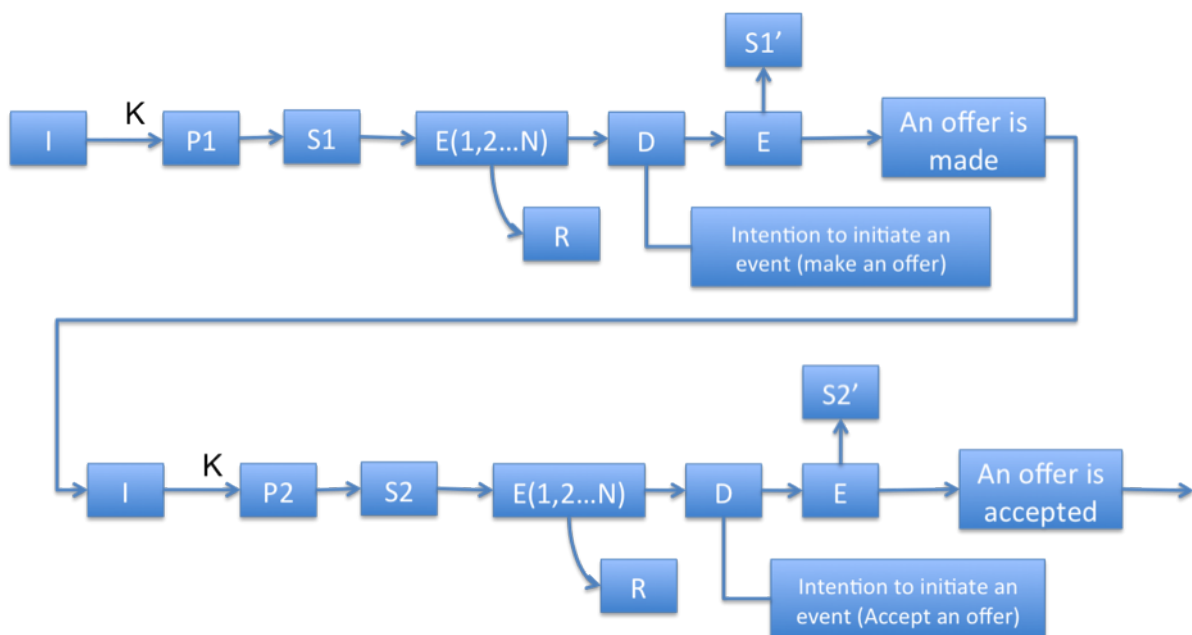


Figure 3.15: Improv agent decision cycle, adapted from (Baumer and Magerko, 2009)

Fuzzy Knowledge Representation

The Digital Improv's prototype that is more related to the research in this thesis is *Party Quirks* (Magerko et al., 2010a, 2011b) which is inspired in an homonymous improv game. In the real life version of party quirks one actor plays a party host, while the other actors play the guests. The audience provides secret quirks or characters to each guest. The task of the host is to guess the quirks or the secret character for each guest.

Although *knowledge disparity* and *cognitive convergence* (see Section 3.3.3) can be considered "*an ubiquitous aspect of improvisation*" (Magerko et al., 2011b), in *Party Quirks* these elements are particularly relevant, since it is the goal of the host to guess the information that is hidden from him and know by everyone else.

In order to simulate this cognitive disparity one needs to create a knowledge space where divergences and ambiguities can occur, i.e. a space that allows for a multiplicity of interpretations for the same event. In *Party Quirks* this was achieved by implementing a fuzzy knowledge representation (see Figure 3.16).

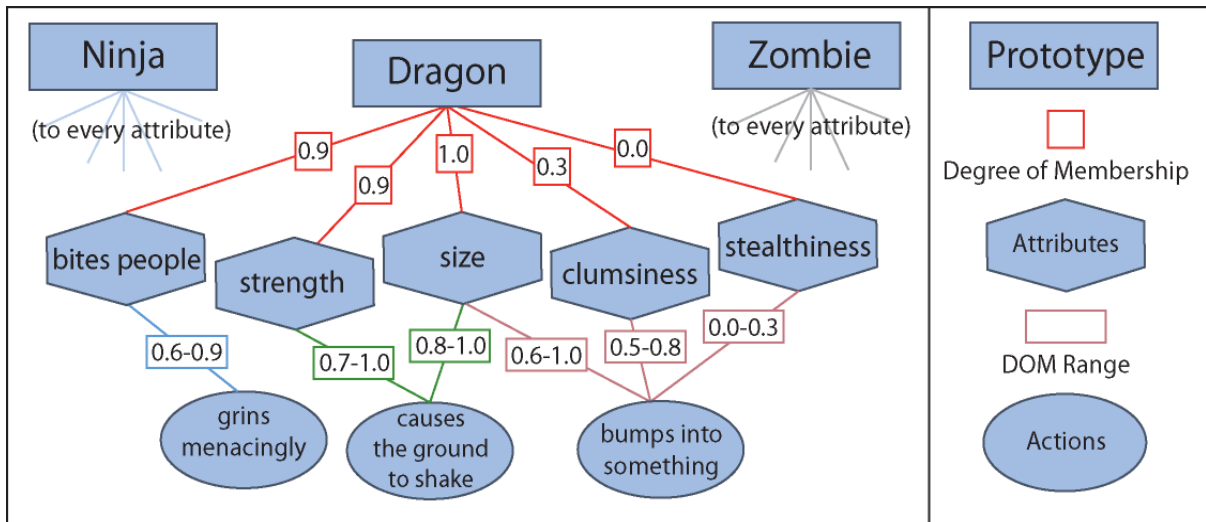


Figure 3.16: This picture illustrates the Knowledge Structure used in the *Party Quirks* prototype taken from Magerko et al. (2011b). We can clearly identify the conceptual hierarchy involving actions, properties and prototypes and the strength of the connections between these concepts.

Characters and quirks are modeled as *character prototypes*, where prototype refers to "*an idealized, socially recognizable construct that maps to a certain kind of character*" (Lakoff, 1999). Each prototype is composed by a collection of properties. Each property is related to one or more prototypes with a value between 0 and 1 called Degree Of Membership. Properties are portrayed by actions. Actions and properties are related with each other by a DOM Range which is an interval of values within the DOM values. In an example from (Magerko et al., 2011b) a *Ninja* has a high DOM value for *using a sword* and a low DOM value for *clumsiness*, while a *Town Drunk* has opposite values.

Prototypes are characterized by properties that are communicated by at least one action. Using the example in Figure 3.16, the property *size* can be represented by two <action, DOM Range> pairs: <*causes the ground to shake*, 0.8-1.0> and <*bumps_into_something*, 0.6-1.0>. The action *bumps_into_something* is also associated to another property *clumsiness* with a high DOM range <0.5-0.8>, which in this case can be associated to a *Town Drunk* prototype. Hence, the same action can be used to describe different properties of different prototypes. The final result of this approach is a knowledge structure that promotes the reusability of its elements, variability of the prototypes presentation and a shared resource for all characters to infer from.

Another relevant contribution of the fuzzy knowledge structure is that it allows to replicate knowledge disparity situations that are ever-present in improv. An example of a knowledge disparity scenario, could be one of an agent playing a *Dragon* that executes the *bumps_into_something* to portray the *Dragon's* large *size* and *clumsiness*, while at the same time another agent interprets this action as portrayal of the *clumsiness* of a *Town Drunk*.

A way to provide agents with the awareness that such divergences may occur, is to let them calculate the *ambiguity* of an action, i.e. a measure of much a prototype can be determined from a <attribute, value> pair. The ambiguity value considers how much the value for a pair is distant from the average value of the other pairs for the same attribute and on the number of other prototypes that can be described by the same attribute. The resulting value is normalized according to the other pair values of the prototype in order to facilitate comparisons.

Cognitive consensus is achieved through a series of offer / response sequences by which the guests can display queues about their quirks and characters. The game gets more interesting as the guests apply different strategies for suggesting their quirks. According to Magerko et al. (2011b), there are at least three character portrayal strategies that can be used by the guests:

- Reverse Scaffolding - the main purpose is to keep the scene interesting through delaying the hosts success without breaking the character. The guests achieve this by starting with less obvious offers, and by increasing the offers accuracy over time. The same is to say, the guests start with more ambiguous offers and incrementally reduce the ambiguity of their offers over time.
- Caricature - consists on exaggerating a particular prototype or quirk for achieving a comic purpose. This is achieved by repeating the selection of actions with very low ambiguity values.
- Opposing - when opposing, the guests inverts the DOM value for an attribute with low ambiguity and includes this new value in his action selection. A possible result of this could be a *barman* behaving exactly like a barman except that it has low value for *bottle juggling*.

The strategies presented above were implemented in a prototype demo presented at the 2011 Chicago Improv Festival where it was used by professional improv practitioners, that could interact

with a Party Quirks agent. This is still an ongoing research that continues to develop relevant contributions to the possibilities of improv in AI and vice-versa, recent examples of these are the contributions to cognitive consensus (Hodhod and Magerko, 2012a) and shared mental models (Hodhod and Magerko, 2012b).

3.3.4 Back-Leading through Character Status

In "Back-Leading through Character Status" (Zhu et al., 2011) the authors propose a new model for guiding the user experience on an interactive storytelling application. In this model, the authors recover the interactive theatre terminology of *inter-actors* (trained actors that interact with an audience) and *spect-actors* (participative audience). The interactive storytelling system is addressed as an inter-actor that guides spect-actor in an exciting narrative experience while transmitting the sense of control the user. This balance between guiding the experience while transmitting the sense of following is inspired in the *Back-Leading* techniques developed by experienced ballroom women dancers that need to lead the dance while appearing to be following their less skillful male partners.

Back-Leading through Character Status consists on maintaining a status negotiation with the user in order to promote the occurrence of status shifts that according Johnstone (1987) motivate the characters actions and contribute to the audience excitement. This way the system can guide the user through low status and high status conditions that establish the user action space without directly forcing the user to take any particular action.

Four different types of status transitions are considered in this model:

- Both lower status
- Both raise status
- One raises while the other lowers
- The status is reversed during the scene

The task of the inter-actor is to engage the spec-actor until the point at which the spect-actor is willing to take his own actions. The inter-actors' choice of status position offers cues to the spect-actor about its relations with the surrounding environment hence promoting the spect-actor awareness of the story setting and his role in it.

The above mentioned effect can be explained using the theory of Force Dynamics (FD) (Talmy, 1988) that models "how semantic entities interact with respect to force". FD establishes a framework that includes the concepts of " exertion of a force and the overcoming of such resistance, blockage of a force and the removal of such blockage, and so forth." It generalizes upon a basic pattern of two entities, an Agonist (the focal entity) and an Antagonist that exerts force over the Agonist (see Figure 3.17).

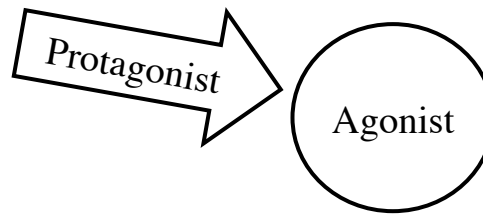


Figure 3.17: Theory of Force Dynamics is defined by a Protagonist that exerts force to an Antagonist.

An example of an application of the Back Leading through Character Status could be an Inter-actor (Protagonist) playing to be a police officer that orders a driver (spect-actor and agonist) to stop the vehicle, thereby exerting a force that leads the driver to stop driving around and prepare to present his documents. Afterwards the inter-actor could lower his own status by apologizing *"So sorry, Mr. Mayor couldn't recognize you. I hope there's no harm done."* and as a result offer a high status transition to the (spect-actor) back-leading him to act like a Mayor without limiting the spect-actor actions.

3.3.5 Late Commitment and Open World Assumption

A most distinct aspect of Improv is that its participants develop a common understanding of the story elements as the story creation process goes along. This aspect results from the fact that the actors have to build upon a story setting that is undefined at the beginning of the creative process.

In *The Virtual Storyteller* (Swartjes and Theune, 2008), Ivo Swartjes (Swartjes, 2010) focuses on the problem that arises when an agent has to plan in a storyworld that is not fixed at authoring time, but it is instead dynamically defined at runtime in favor of a better story development.

A dynamic definition of the storyworld, implies that an agent can add elements to the storyworld in runtime, e.g. when an agent playing a policeman says *"You'll rotten in this cell..."*, he is saying that there is a cell in the storyworld and that the other actor is in it, without this ever being completely defined in authoring time.

Two different types of goals are applied to the planning algorithm: Inner Character Goals (IC) and Out of Character Goals (OOC). Inner Character goals allow to define the behavior of the agent that emerges from the character's structure of beliefs and actions, while OOC goals are related to the agent's ability to define properties of the initial state of the storyworld. This distinction is most helpful when we consider the case in which an agent runs out of active character goals to explore the storyworld. In this case, the agent can use OOC goals to plan actions that will guide the agent to add content to the storyworld. The new content will update prepositions about the storyworld that will activate new IC goals for the agent to pursue. An example could be that of an agent playing a prisoner that runs out of active IC goals to pursue, detecting that if he adds a new statement to the storyworld (*has, shovel, SELF*) = *TRUE*, he can activate the *EscapeFromPrison*

IC goal.

Through Late Commitment (Swartjes, 2010), the agents only establish certain storyworld prepositions as being *FALSE* or *TRUE* during runtime. To achieve this behavior, Swartjes applied Open World Assumption (OWA) logics. Unlike classic Closed World Assumption (CWA) logic where every statement is valued as *FALSE* or *TRUE*, OWA considers a third value *UNKNOWN* to which every statement is set to, while it has not been evaluated. This allows for an agent to bring up new elements about to the storyworld that refer to statements that have not been previously addressed. Going back to the previous example, the value of the statement (location, object) for (cell, shovel) is *UNKNOWN* since the beginning of the scene, and because of this the agent can attribute a *TRUE* or *FALSE* value to it without bringing up inconsistencies.

This was not the first time OWA logics was applied in IS. In (Riedl and Young, 2005), the authors present the Initial State Revision (ISR) algorithm that considers the use of OWA to evaluate mutually exclusive sets of prepositions. These sets are valued as *UNKNOWN* at authoring time. When a set of prepositions is valued as *TRUE* all the other sets of prepositions that are mutually exclusive to it are set to false. In the case of the Impro-POP planner (Swartjes and Theune, 2008) there is no use of mutex sets. In this case the author defines *framing operators*, which are also independent sets of preconditions and effects that are used in the OOC dimension of the planning algorithm. According to its author, framing operators are not used to establish new facts about the storyworld settings, but rather "a commitment to truth or falsehood of facts" (Swartjes, 2010).

Another important feature of the Impro-POP algorithm is that it integrates OOC and IC perspectives with a special regard to its improv grounding. OOC operators are only used in the planning algorithm whenever the planner fails "to reuse plans already in the partial plans or choosing new IC operators" (Swartjes, 2010), this way the Impro-POP favors the reincorporation of previously defined story facts, whenever possible, before adding more elements to it, which is in line with one of improves principles of reincorporating material (Imp, 2012).

3.3.6 Discussion

The Improv project at NYU, pioneered in many different aspects, particularly in what concerns its contributions to the study of believability in virtual characters. Rather than following realism, believability is related to the user involvement and acceptance of the system rules, and consequently his willingness to accept that characters have their own life. This is related to the systems ability to convey the characters' inner life, through a clear portrayal of the character's personality through their behaviors. A relevant contribution to this thesis is the adoption of stochastic decision process methods to generate believable and non-repetitive behaviors, more particularly in a way to promote action variability without breaking character's consistency. Nonetheless, the system depends on heavy scripting efforts of activities and action to get a character prepared. Furthermore, the

large quantity of variables that be used to describe a personality and the characters tendencies brings even more weight to the authoring process. While the stochastic methods seems relevant for achieving a safe balance between variability and consistency, the authorial burden related to this approach is challenge that needs to be addressed.

Directed Improvisation is the first approach to IS that fully refers to Improv as source of inspiration. Its main contribution was the demonstration of how status can affect story variability. Nevertheless the term *Improv Actor* seems too ambitious for an agent whose behaviors seem to be strictly confined in a pre determined story space. Also, the notion of *Improv Puppet* seems to be pushing the limits of the improv essence and it can only be accepted in the context of the extremely vast scope of the improv definition used in this work.

Digital Improv establishes a systematization of the improv process, supported by empirical research. The improv domain presented identifies the most relevant challenges of an AI approach to improv and allows us to detect possible research contributions. We observe that the Digital Improv's *Improv Theory Domain* includes the problems raised in the discussion of the challenges of the three beat pattern presented in Section 2.3, this agreement should be seen has a reenforcement of the credibility of the analysis process and its results.

In terms of basic cognition the inference of a story state from the semantics implicit in action is a serious challenge that involves extracting knowledge from the sensory stimuli. This hard task is present in the first step of the cycle when the Performer extracts K from I to identify a story state (see Figure 3.15). In our opinion this process is seriously dependent of two critical design milestones: the development of an inference strategy; and the definition of a Story State representation, and none of them has a simple nor obvious solution.

Just the Story Sate definition, that is a central element of this model, is not straight forward, it is domain dependent and just the mere consideration of representing story states collides with the traditional agent roles that are defined in character and consequently collides with traditional architectures. This cycle implies the design of an agent architecture that can reason upon character and story state at the same time.

The Fuzzy Knowledge structure presented in Section 3.3.3 establishes an extremely relevant contribution towards defining a knowledge structure that can be used by autonomous agents that aim at mimicking the improv acting behaviors. It defines a knowledge structure that can be easily extended using crowd sourcing techniques wile at the same time it promotes the reusability of knowledge in different scenarios, contributing in this way to reducing the authorial burden implied in the creation of autonomous agents for IS.

Although the approach followed to define this knowledge structure was fully motivated towards generating character portrayal behavior, we argue that this same structure could be applied to other entities that are relevant for defining the knowledge space of an improv performance, such as, relations between characters, locations, objects or even activities. In Section 4.5.2 we propose

to adapt this knowledge structure to accomplish this extension. We will use it as reference in the experimental observation and data analysis approach in our research, in order to identify relevant knowledge entities and the strategies that operate over them that are used to develop tilts in an improv presentation.

Back-Leading is another good example of how interactive storytelling can extract useful lessons from interactive theatre and improv. This model sets the basis for a current ongoing empirical study of interactive theatre which is extremely related to the research presented in this thesis. Status shifts are presented as opportunities to guide the action directions and elicit audience excitement which, thereby sharing the same motivation than our research to study status shifts to generate contributions for the interactive storytelling community. At the time of writing this thesis there are no more contributions from the research on Back-Leading, nevertheless it is expected that future outputs will contribute to reinforce the contribution of status and status changes to interactive storytelling and consequently reinforce the relevance of both researches to the community.

The reincorporation of story material in Late Commitment, is a much relevant adoption of improv "best practices" to IS as it allows the creative process to be extended from the authoring stage to the execution stage. One of the reason that makes Late Commitment possible is the architectural decision that clearly separates *in character* goals and *out of character* goals. This division allows the agents to take individual decisions about the story development without constraining to the role of the character that they are playing. These are very solid adoptions of improv knowledge. This agent is based on a OOA planner that elaborates long planning decisions, which mimics the unpredictable state of an improv setting but also a decision process that plans many actions ahead. Our main criticism on this work is towards its planning approach, because real-life improv is based on fast and intuitive decision making, not only to provide more creative freedom but mainly because of the ubiquitous negotiation (offer / response) process, which prevents actors to raise too much expectations about their future action since at each turn they are dependent on the new inputs of other actors.

3.4 Summary

Both plot centered and character centered approaches have produced significant contributions for IS. FAÇADE continues to be the most successful IS system ever developed and autonomous agent's research for IS continues to produce relevant AI and HCI outcomes.

Part of FAÇADE's success results from its story beat representation for defining the plot. The story beat approach enhances authorial control in a dynamic environment that promotes story variability. Nonetheless, it is common for a user to experience loss of agency, particularly when the characters fail to respond to user actions that are not contemplated in the current story beat. Character centered approaches try to overcome such loss of agency by endowing the autonomous

agents that play a character with more realistic interactive behaviors.

While the creation of more socially intelligent agents is being achieved it is still made at the cost of increasing the agent's authoring complexity. Hence, while character centered approaches are improving their interaction possibilities, they could largely benefit from a reduction of the authorial effort.

In Chapter 1, we proposed authorial-agents as a solution for this problem based on a transfer of the authorial power from the author to the agents. This solution requires the definition of cooperative processes of story making based on improvisational theatre that could be applied to autonomous agents. In this context, Improv Project approach to improv analysis presents the opportunity for assessing the cognitive processes involved in the cooperative creation of stories. In the next chapter we will present how the Digital Improv approach was adapted for this purpose.

A Conceptual Model for Story Development Using Tilt

In the previous chapters we presented improv theory and how it could be used to facilitate the creation of authorial-agents. In Chapter 1, we present improv as the real life cooperative story making example that is closest to the concept of authorial-agents, because in both contexts the story results from the on-the-fly contributions of individuals that participate in the story as characters. Based on this, we argue that since the problem of creating a story in improv is comparable to the problem of creating stories in a multi-agent context, the creation of cooperative story models for authorial-agents could be informed by the analysis of real-life improv. In Chapter 2 we advocated in favor of the use of improv actors as a metaphor for authorial-agents and concluded that the improv collaborative story creation process used by improv actors is strongly supported by the tilt process. Following this, in Chapter 3 we underlined the opportunity presented by the Digital Improv project to assess the cognitive process involved in an improv performance.

In this chapter we report the approach followed to analyze the Improv Project's video data (see Section 3.3.3) in order to identify the information used by improv actors to develop stories and tilts in an improv a scene. The main purpose of this analysis was to use its results to support the modeling of the cognitive processes involved in improv story creation. Furthermore, given the relevance of the tilt process on the story development, we aimed at assessing the elements involved in the creation of a platform and the elements used in the process of tilting it. It is important to remark that we want to identify and extract relevant information from a complex phenomena in a subjective environment. This assessment involves an unmeasurable quantity of subjective data, as such, we can not aim at identifying a complete and indisputable model, but instead an informed modulation that can be used to explain an generate new phenomena which is very similar to the one analyzed. We will present the results of the analysis using our own coding approach to the Digital

Improv data. The coding approach is based on the collaborative emergence theory presented in Section 2.1.2. Partial contents of this chapter were presented in Brisson et al. (2011a,b, 2013)

4.1 Video Data Description

In this analysis we used video data from the Digital Improv project. The video data was collected according to description in Section 3.3.3. First a group of three actors was invited to perform an improv scene. The scene was recorded. After the improv performance, each actor was interviewed individually. The interviews were taken in parallel, recorded in video and in each interview the video of the performance was provided to the actors to comment and review their performance. After the individual interview the actors again in a group interview were they could collectively discuss and comment on the performance using a the performance video. This group interview was also video recorded (see Figure 4.1).

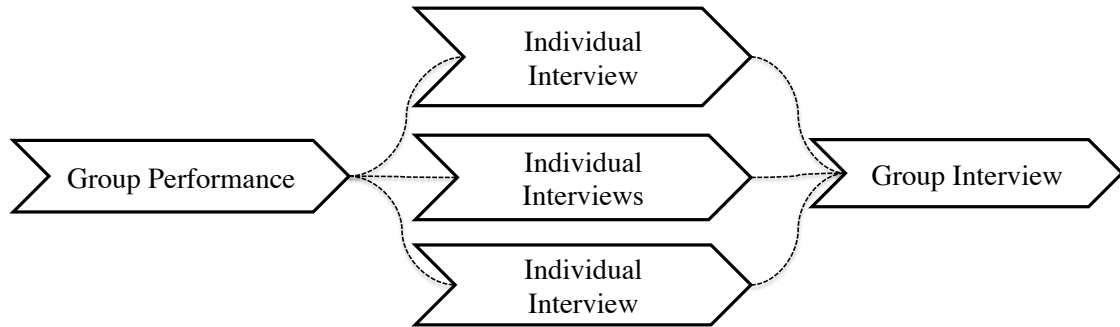


Figure 4.1: Data gathering process

At the time of this research the Digital Improv video repository included more than 70 hours of performance and interview footage (Magerko et al., 2010b). From this collection we selected the videos of the two scenes (Scene A and Scene B) with the best story examples. In Scene A, 3 actors were asked to perform a 3 minutes long scene and produced a 3 minutes 38 seconds scene with 65 actor turns. Scene B was performed by 3 other actors, that were asked to perform a 3 minutes scene of "three college friends at the zoo" and produced a 4 minutes 25 seconds long scene (see Table 4.1).

In scene A, the action starts with two man discussing the benefits of *fair trade* products. Both characters develop a routine in which they emphasize the benefits of *fair trade* to the world. The routine is broken when a third actor enters scene as a *fair trade* worker asking for food. Full transcription and analysis for scene A is provided in Annex A.

Scene B takes longer to develop than scene A. The actors quickly establish a platform of three college "*buddies*" where one of them is a rule follower who criticizes his friend whenever they break any rule. This routine is only broken further ahead when they find out that the "go by the rules character" is being a victim of domestic abuse, shifting the whole story direction and the characters

	Scene A	Scene B
Number of actors	3	3
Constraints	3 min	3 min; "three college friends at the zoo"
Scene length	3min 38s	4min 25s
Interviews footage	98 min	108min 37s
Number of actor turns	65	76

Table 4.1: Two scenes were analyzed, Scene A and Scene B. Each scene was performed by three different actors. In Scene A the actors were asked to perform a 3 minutes long scene, the final scene took 3 minutes and 38 seconds, the interviews summed up to 98 minutes of footage and the interaction between actors resulted 65 different turns in which an actor intervened. Scene B had an extra constraint that suggested the actors to be "three college buddies at a the zoo". Scene B took 4 minutes 25 seconds, 108 minutes and 37 seconds of footage were collected during the interviews and the actors interventions added up to 76 actor turns.

	Tilt Example (Scene A)	Tilt Example (Scene B)
Buildup	Two actors (1 and 2) emphasize how actor 1 is saving the world reselling Fair Trade products. .	Three actors (4, 5 and 6) Actor 6 acts as "serious no fun guy" that teaches his friends how to behave in public places.
Tilt	<Actor 3 enters as Fair Trade worker> Actor 3 - Please feed me! Actor 1 - <fails to explain himself>	Actor 5 - <towards Actor 6> Does she (Actor 6's wife) hit you?
Effect	Actor 2 - <shocked at Actor 1>	Actor 6 -I don't want to talk about it guys... <avoids eye contact>

Table 4.2: Tilt examples from the scenes analyzed. In Scene A the a tilt occurs when the actors break a routine in which they overvalue the benefits of *fair trade* when they introduce a *fair trade* worker that starves. In Scene B the tilt occurs when the character that was always playing as a role model is found out to be a victim of abuse.

relations. In Table 4.2 we provide examples of tilts that occurred in these scenes. Full transcription and analysis for scene B is provided in Annex B.

4.2 Approach

In Section 2.1 we present improv as "a form of unscripted performance that uses audience suggestions to initiate or shape scenes or plays created spontaneously and cooperatively according to agreed-upon rules or game structures" (Seham, 2001) and as "the creation of an artifact with aesthetic goals in real-time that is not completely prescribed in terms of functional and/or content constraints"(Magerko et al., 2009).

The task of assessing the information used in an undetermined, creative and collaborative environment presents many challenges, among these is the fact that acting conveys a lot of non-verbal and subjective information that can lead to ambiguous interpretations of the same actions. Such information is present in the actors' mental process and can not be directly assessed. Furthermore, the spontaneity inherent to an improv performance does not allow for pervasive interruptions that

could be used to enquire the participants *in loco*. This is why we can not use the immense collection of improv performances available online to accurately identify the cognitive processes involved in improv nor use classical approaches to extract statistical relations from them, because they do not have any information about the cognitive processes and the subjective elements of improv.

At the same time, the development of a computable conceptual model for story development in improv requires objective information that can be directly mapped into computational representations. Previous IS research inspired on improv (see Section 3.3) rely on shallow understandings of the improv phenomena. While such approaches produced significant contributions, the effort made by the Digital Improv research project to collect video data results in a data collection that contains not only the improv performances but also the individual comments of each actor and the discussions of the scene development process between the actors. (see Section 3.3.3) The size of the data sample does not stand for the whole improv phenomena, nonetheless, it is clear that this data is far more descriptive of the mental process associated to it than any other data source, making it an unique data source for detecting and resolving some of the most relevant ambiguities between the individual and collective perspectives of the actors about a scene.

The specificities of the video data, such as its size, the spontaneity of the performances recorded and the details about the actors' cognition, allow for deep understanding of the recorded phenomena. We argue that this understanding can be used to generalize a conceptual model for story development in an improv context. Similar to a grounded theory approach (Glaser and Strauss, 2012), designed to understand complex social phenomena, instead of deducing conclusions from a large data source, we propose to induce a conceptual model that fits the behavior of smaller and very detailed detailed data, where every elements of the model is clearly supported by the data source.

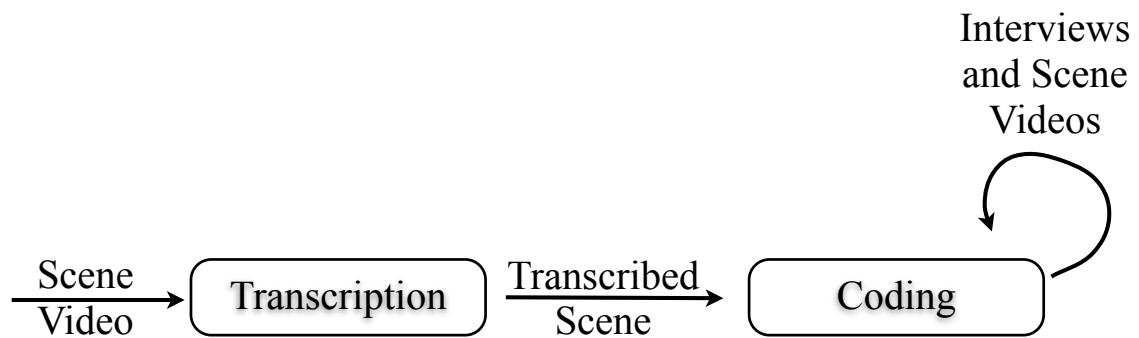


Figure 4.2: The first step of the analysis process was to transcribe the video scenes divided by actor turns. Afterwards the transcriptions were annotated according to the info provided by the videos using our coding scheme.

Figure 4.2 presents an overview of the analysis process, which started with the full transcription of a scene. Each actor turn was transcribed in detail. Afterwards each actor turn was annotated with the variables for each platform element using the coding scheme specially designed for this

Platform Elements	Variables Annotated
P (Proposed)	When any variable is presented to a scene, it is labeled with P on the frame of the actor that proposes it.
R (Received)	When an agent proposes a variable, the other agents interpret it and register the result of this interpretation with the value R.
A (Altered)	If the scene development leads into a state in which the value of an R variable is no longer consistent, the agent modifies its value to a possibly consistent value and marks it as A.
C (Confirmed)	When a variable is addressed by an agent other than that which originally proposed it or when a variable is repeatedly proposed or very strongly proposed, the variable is marked as C.

Table 4.3: Coding scheme

purpose and with annotations from the observer.

Our approach for identifying objective data about the platform elements in such an ambiguous and unstructured environment using video data from each actor’s perspective and their discussion, relied on two assumptions: principles:

- Explicit data assumption - the coding approach should discard all the ambiguous information that seems dependent of subjective interpretation and only consider information that is clearly presented during a scene, information confirmed or given by the actors in the post performance interviews.
- Tilt relevance - since tilt is considered by practitioners as the main element of story development in improv, we argue that a computational model of story development based on improv requires a strongly supported representation of a tilt function. In order to get this support, the data coding should be focused on the assessment of the cognitive processes used in tilt.

In order to avoid the misinterpretation of implicit information in the annotation process, we consider each improv actor as a virtual agent whose knowledge over the scene is completely provided by the explicit interactions with the other agents. We consider as explicit interactions the content of dialogues and clear mimics. Furthermore, we use the interviews to confirm information that could be annotated whenever possible.

We defined a data-coding scheme based on collaborative emergence (see Section 2.1.2) to be applied in transcriptions of improv scenes (see Table 4.3). The story elements annotated with our coding-scheme are organized according to the platform elements of *who?* *what?* and *where?* (see Table 4.4) and included into one of two kinds of data sets *story frame* and *dramatic frame*. *Story Frame* is the individual perspective of an agent that includes all the unconfirmed story elements that it considers to exist in the scene. Whenever a *Story Frame* element is confirmed by another

Platform Elements	Variables Annotated
Who?	Characters (name, occupation, habits, physical attributes, other attributes) Relations (affinity, status)
What?	Activities (subjects, targets, props)
Where?	Location (scenario elements, props)
Other	Explicit Offers (Subject, Variable, State)

Table 4.4: Annotated platform variables

agent or clearly reinforced by the agent that proposed the element, the value associated to the elements is passed to Confirmed (C). Only confirmed elements are sent to the *dramatic frame*.

At each turn, each variable present in each story frame is annotated with the corresponding code. We are able to annotate which elements compose each individual story frame and also which of those elements take part in the shared dramatic frame. Figure 4.3 shows the coding sequence for a regular offer / confirmation interaction. The interaction starts with Actor A proposing an element from his story frame. To do this A executes an offer. B interprets A's offer extracting from it the new elements to his story frame. We code these elements in B story frame with a received value. B's story frame influences his response as he tries to confirm the received offer. B confirms to A. When A receives the values offered by B we consider that both actors have confirmed values for the elements corresponding to A's offer and B's response. The confirmed values in both story frames are sent to the dramatic frame.

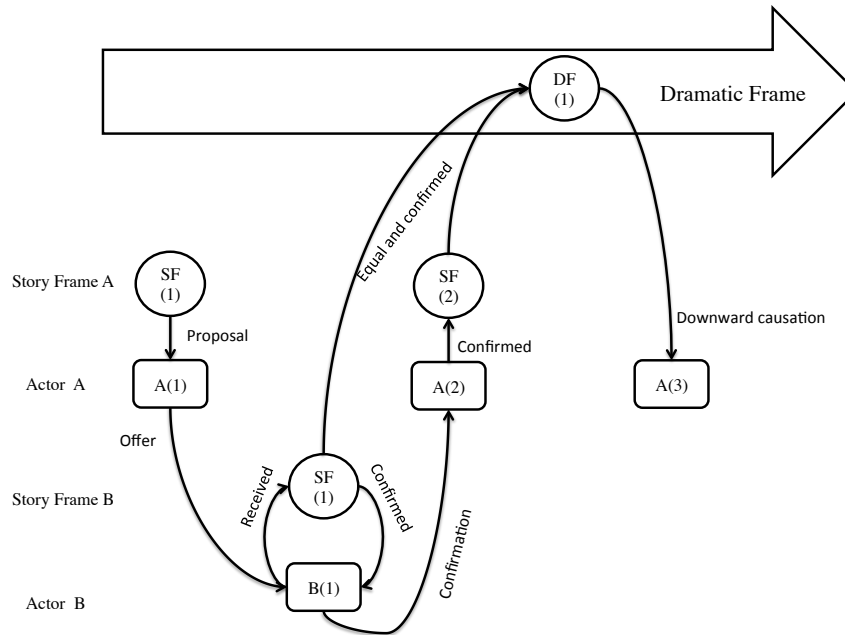


Figure 4.3: Coding sequence for a regular offer confirmation interaction

4.3 Analysis

The approach described in the previous section was applied to the two scenes with the best tilt examples from the data collection (see table 4.1). According to Table 4.5 and Table 4.6 the final annotations included 166 variables annotated for Scene A and 262 variables annotated for Scene B. These variables were distributed along the story frame of each actor and the collective story frames on each performance. The variables annotated were platform variables (Location, Activity, Characters and Props), constraints (such as the pre-determined scene length) and also the relations between the characters that includes annotations about status transitions and relation types, such as friends, clients, among others.

	Actors			Total
	D1	D2	D3	
Constraints	1	1	1	3
Location	3	2	2	7
Activity	14	13	12	39
Characters	23	23	23	69
Props	10	10	10	30
Relations	6	6	6	18
Total	57	55	54	166

Table 4.5: Quantification of the variables annotated for each actor’s story frame in Scene A.

	Actors			Total
	D5	D6	D7	
Constraints	2	2	2	6
Location	6	6	6	18
Activity	30	30	28	88
Characters	27	27	27	81
Props	15	15	15	45
Relations	8	8	8	24
Total	88	88	86	262

Table 4.6: Quantification of the variables annotated for each actor’s story frame in Scene B.

4.3.1 Tilt Occurrences

Following our tilt working definition (the proposal of a new story direction by introducing a significant change in an established platform, see Section 2.1.4), we annotated every instance of a tilt in the analyzed videos. We observed that more than one tilt happened within each scene, although for both scenes there was a tilt that had a greater impact over story development than the others (see Table 4.7).

	Tilt Occurrences (scene A)	Tilt Occurrences (scene B)
Weak Tilt	2	1
Strong Tilt	3	1

Table 4.7: Tilt occurrences

Turn	Actor	Action	Comments
1	D1	< is using a sort of broom on the floor> <making a scratchy noise>	D1 proposes "breaking leaves" in a garden D2 sees a sort of indoor maintenance
2	D2	<opens a door> <walks to the middle, informal, relaxed> Hey! How are you? Gotta get that Cup of Joe	D2 proposes Coffee Shop D1 receives as being inside

Table 4.8: Non consensual location example

4.3.2 Location Divergence

In scene A, we found evidence of a tilt that occurred without a shared value, between actors, for "Location" (i.e. there was no value for "Location" in the dramatic frame). Furthermore, the value within the story frame of each of the two actors, playing the two main characters was inconsistent at the moment the tilt occurred. In this case one of the actors performed the whole scene thinking he was in a coffee shop while his partner was not even sure if he was indoors or outdoors. Table 4.8 describes how this divergence results from the misinterpretation of an action gesture of one of the actors.

The scene starts with Actor (D1) performing what seems to be a "sweeping" gesture. In the post-performance interviews, his partner (D2) reports that he interpreted D1's gesture as some kind of indoor maintenance, while D1 reports that he was breaking (sweeping) leaves in a garden and got completely lost about the scene location when D3 started acting like if they were indoors: "Where are we? Does he think I was mopping?" This way D1 was forced to alter (A) his own activity and location to an unconfirmed value that could possibly be consistent with his partner's frame. Even though this divergence in their mental models was maintained almost until the end, the scene developed a good tilt. The resulting individual frames are presented in Table 4.9.

The final annotation of two short scenes with 3 characters included the annotation of 428 story elements. The interview videos not only facilitated the confirmation of some elements, but also allowed to detect elements that weren't easy to perceive from the performance videos. Nonetheless, the possibility of using these videos exponentially increased the size of the video data to be analyzed, which increased the duration of the coding task.

The final coding allows us to isolate a relevant part of the explicitly confirmed knowledge used in each moment of the performance. A preliminary analysis allowed us to observe that a tilt can

	D1 Story Frame	D2 Story Frame	Function
Activity	D1 Mopping (A) (Breaking leaves (P))	D1 Maintenance (R)	Platform Buildup
Location	Inside (A) (Outside(P))	Inside (P) Coffee Shop(P)	
Props	Mop(A) (Rake(P))		

Table 4.9: Sub-set of the actor’ story frames at Turn 2. D2 received (R) D1’s activity as maintenance and proposes (P) Coffee Shop for location. D1 is forced to alter (A) his previous proposed value for location.

Story Step	Priority
Platform Buildup	Create a Platform
Tilt Buildup	Unbalance the established platform and ride it.
Problem	Unbalance the established platform significantly and ride it.
Denouement	Balance the story elements in order to slow down the action, and tight loose ends.

Table 4.10: Story steps primary goals

occur without an established value for location. We also detected the occurrence of several tilts in each scene, which goes in line with Johnstone’s notion of weak and strong tilts presented in 2.1.4.

In the next sections we report how we used these results as a input to define a conceptual model of a cooperative story process for authorial-agents and analyze the strategies used to produce a tilt. The conceptual story development model included in the next section was presented in Brisson et al. (2011a).

4.4 A Conceptual Story Development Model With Tilt

Our first observation when approaching the definition of a conceptual story development model for authorial-agents based in improv was that players have different concerns over story development during a performance. An example is the concern about adding elements to the environment at the beginning of a scene, while at the end they are more concerned about establishing relations between all the elements presented in the scene, or the need to resolve a problem presented in the story. We call the story moments with distinct player concerns story-steps. In the analyzed scenes, we observed four different story-steps, each one with a distinct priority (see Table 4.10).

The four story steps were the first high-level elements for the definition of the conceptual model presented in Figure 4.4. In the following subsections we will detail all the elements of this model and the observations that led to their definition.

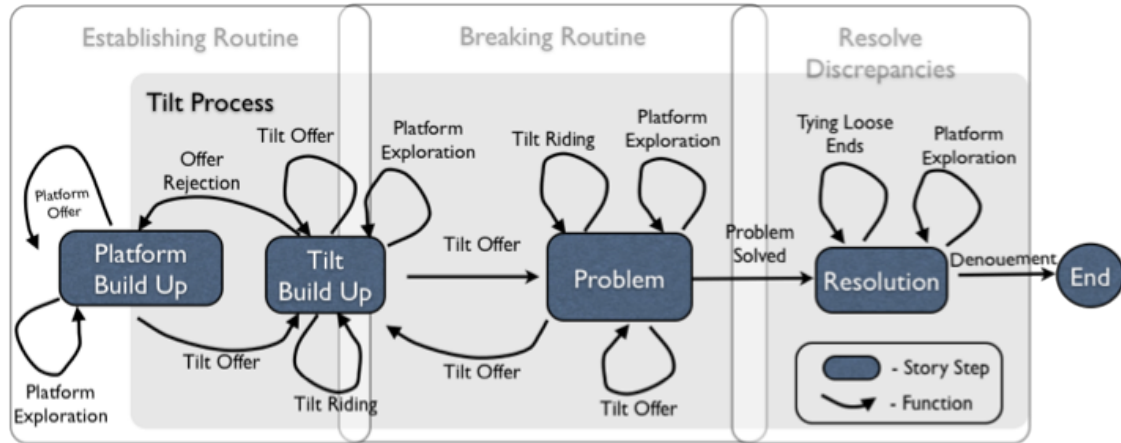


Figure 4.4: Conceptual story development model using tilt

Inputs	Output
Current Platform	New Platform Elements
Agents Frame	New Platform Restrictions
Dramatic Frame	

Table 4.11: Platform offer

4.4.1 Platform Buildup

The first step is where players create a platform and establish a routine. In scene B (see Table 4.2), player 6 reports the definition of his character (who?) in the established platform at the beginning of the scene before the occurrence of the first tilt: *"this is where I thought I was going to be the guy that plays by the rules. In this relationship of 3 guys from college I'm gonna be the dork."* Platform buildup includes two main functions: *Platform Offer* and *Platform Exploration*.

Platform Offer

Platform offer is used to add content to a story, such as character definition, story context, scenario, and other, with the goal of defining a platform that extends the space of possibilities in a story. The search for a platform offer is mostly associative and should use elements from the agents' perceived story frame. An example of this occurs in scene A when one of the players comments *"Hey, looks like you guys got a new line of muffins this morning,"* offering the existence of a muffin table in the scene. Explicit offer rejections can also be seen as offers of elements that can not be added to the scene. This requires the goal of building a platform that does not include the proposed offer (platform restriction), which means that a rejection may also be a platform offer.

Platform Exploration

It is the use of story elements within the current dramatic frame without adding any new story development. This function is recurrently used in every step without a direct impact on the story

Inputs	Output
Dramatic Frame	Character Action

Table 4.12: Platform exploration

Inputs	Output
Current Platform	Target Platform Tilt Variable

Table 4.13: Tilt offer

development either when a player finds no alternative or just wants to establish a relation with the scene elements. An example of this is when a character in scene A offers a muffin to another player: *"Here, have a muffin man. They're delicious."* These can be considered as default actions whenever no relevant action can occur.

4.4.2 Tilt Build Up and Problem

These are very similar story steps, which are responsible for breaking the routine. In both analyzed scenes we observed that a larger tilt occurs after the initial tilt that breaks the established platform. This larger tilt leaves a heavier mark on the scene and represents a nuclear problem to be addressed. This is in line with Johnstone's notion of minor and major tilts. Minor tilts are part of a buildup that moves the scene towards an inevitable major tilt. Player 6 from Scene B reports the occurrence of two different scene changes. The first tilt which we identified as a part of Tilt Buildup results from an insult of one of his friends to his wife: *"Yeah and I wish you didn't have that wife and those children!"*. Player 6 comments this insult: *"The scene is now shifting into something else, (...) where dealing with my wife which he doesn't like. I'm thinking should I not like my wife or should I like my wife? And that's going to base my opinion about what he said."* From this moment on, the scene develops around player 6's wife with growing conflicting opinions about her which end up leading to the offer *"Does she (wife) hit you?"* at this point player 6 reports: *"Now this is turning into a big tilt, which is a term for when a scene just takes a big turn into turning something else. A Big offer."* We detected two main functions in these steps: *Tilt Offer* and *Tilt Riding*.

Tilt Offer

Given a platform as a set of story variables that represent the who? what? and where? story values, a tilt can be seen as the transformation of a story platform, and its variables, into a new platform that is similar enough to avoid unresolvable inconsistencies and at the same time significantly different in some crucial variable sub-set (Tilt Variable). Following this perspective we define tilt offer as a function that brings new content to a platform with the goal of unbalancing it and present a tilt variable.

Inputs	Output
Current Platform Agents Frame Tilt Variable	New Platform Elements or a Platform Exploration fo- cused on the Tilt Variable

Table 4.14: Tilt riding

Inputs	Output
Inconsistent story ele- ments	Platform element that justifies the inconsistent state.

Table 4.15: Tying loose ends

Tilt Riding

Tilt Riding differs from platform exploration because it is not just a casual exploration of a platform, but an exploration of the platform elements that are more directly related with the tilt variable, with the purpose of increasing its importance and the characters attachment to it.

An example of tilt riding in scene B occurs after one player insults player 6's wife *"Yeah and I wish you didn't have that wife and those children!"* and the scene grows with new elements related with the new variable "player 6's wife". Other players keep adding elements to the scene against player 6's wife, while he purposely fails to counter them. They start by finding her ugly, "manish", too tall, "mammoth shewoman", until they reach the new tilt where they offer that she hits player 6. Player 6 accepts this tilt and rides it by exploring the fact that he is a victim of abuse *"I started acting like abuse victims act"*, changing his character.

4.4.3 Resolution

This is the step where players attempt to resolve the tilt reasons in order to make the action come to an end (e.g. *"at this point the clock in my head is going off. This should be wrapping up we should be finishing this scene."*).

Tying Loose Ends

This function aims at slowing down the pace of a story after its conclusion in order to bring it to an end. In example A one player took advantage of the fact that everybody in the scene was drugged to justify the cognitive divergence state generated from his initial activity. *"Oh my god! You've drugged this entire firm. No wonder I was breaking leaves in the break room."* This is also inline with improv theory *"A pointless story is one in which the recapitulation is missing or bungled, whereas a perfect story is one in which all the material is recycled"* (Johnstone, 1999). This function does not necessarily add new elements to the scene, but rather gives more importance to the definition of causality links between inconsistent or isolated story elements and the story.

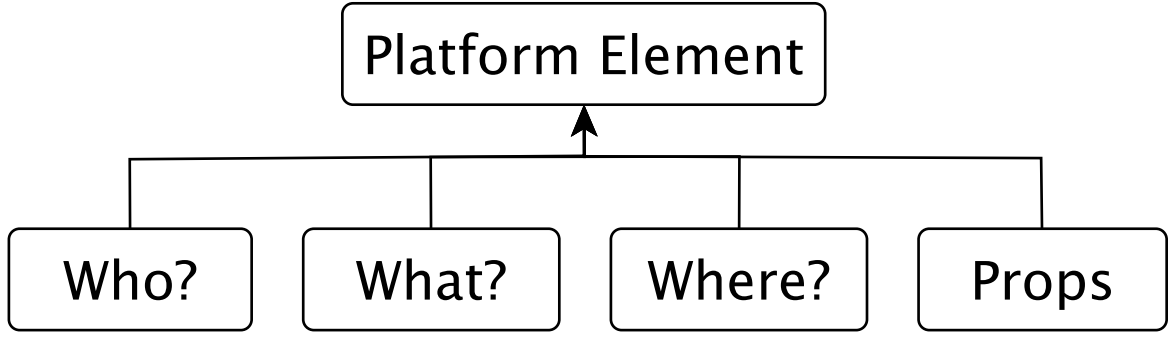


Figure 4.5: Platform elements representation

4.5 Tilt strategies

One of the main goals of our data analysis was to assess how story elements are manipulated in a tilt process. As we have seen earlier, tilt is a process that operates over a previously defined story platform. It manipulates story elements, unbalancing the platform and providing movement to the story. Therefore, before advancing with the creation of a tilt function that operates over the platform properties, we had to formalize a representation for a platform.

4.5.1 Platform Representation

We consider a platform as a collection of elements that establish the characters *who?*, activities *what?* and location *where?* of a scene. In our data analysis we observed that in Scene A all the tilts occurred without a commonly established *where?*. Although further analysis of more improv scenes are needed to take conclusions from this particular occurrence, this observation suggests that *where?* contributes to a platform definition but may not be essential for the tilt process. We argue that location's contribution to story development is similar to contribution of props (objects that exist on a scene). Props and location help to add more associations to the story elements, in order to feed the actors process of adding elements to a scene. The essential elements for representing a platform are the characters and their relations *who?* and a common understanding over the activities being performed (*what?*). Elements describing location and objects are helpful for associating new elements to a scene but are not necessary for tilting a platform, at least in the examples analyzed.

Based on this, we propose that the knowledge structure used by improv agents to represent a platform should include four elements: *who?*, *what?*, *where?*, and *props*, and tilt processes should be primarily directed towards *who?* and *what?* elements (see Fig.4.5).

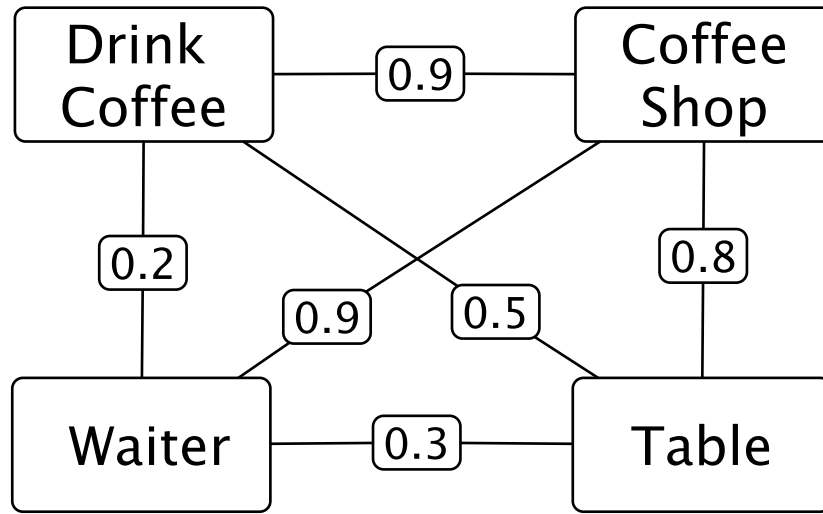


Figure 4.6: Platform elements instantiation

4.5.2 Knowledge Structure

Each specific platform is a unique combination of different instantiations of platform elements. Like in improv, there are no restrictions to the possible combinations of these elements in a platform. Nevertheless, some combinations of platform elements present stronger associations between elements than others. We follow the principle that some platform elements are more likely to occur in the same platform (co-occur) than others, e.g. the activity of eating is more likely to occur in a restaurant than in a work meeting. Two platform elements that are very likely to co-occur have a strong value associated to their relation, while platform elements that are very unlikely to co-occur have weak values associated to their relation. We represent this particular aspect with a Degree of Association (DOA), which represents a strength of the association between two elements. An example of an instantiation of platform elements related to a coffee shop is represented in Fig.4.6.

The instantiation of platform elements is made by executing the actions that portray them. As such, we need to establish a relation between the actions and the platform elements used. Our complete approach for the knowledge structure used in the analysis is described in Figure 4.7. It is an adaptation of the character prototype structure used in the Digital Improv Project’s Party Quirks system (Magerko et al., 2011a), where each element is characterized by a set of properties, e.g., a coffee shop can be characterized by being crowded, or an activity of drinking coffee can be characterized by being a lonely activity. The strength of the relationship between the elements and the attributes is represented by a DOM (non-boolean Degree Of Membership of an element belonging in a given set). The main purpose for DOM in our case is to provide a value for the importance of a property in the definition of an element.

For each property there is a set of actions portraying it. The relationship between a property

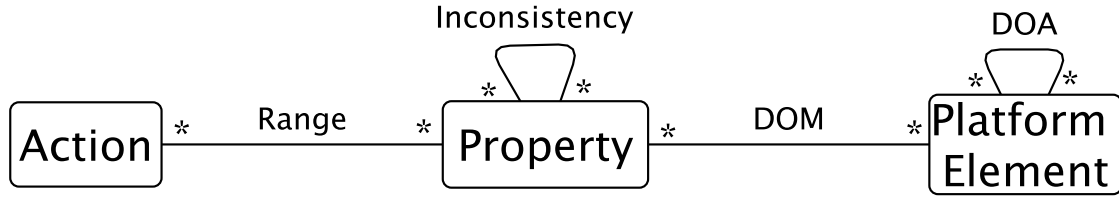


Figure 4.7: Knowledge structure for tilt analysis

and each of the actions that portray it includes a Range, which is a probabilistic distribution of how much an action portrays the corresponding property. Range is used to provide more variability when selecting an action to portray a property.

Relations between elements' properties include a value for inconsistency, which represents the strength of how much the semantic value of a property conflicts with another. The need for this representation is related with our observations of improv actors. In some particular examples, actors intentionally mix elements with semantic inconsistencies in a scene.

In the next subsections we detail two status strategies that were detected and analyzed in scene B from our data.

4.5.3 Tilt: Status Shift

In scene B actor A builds up the character of a *"serious no fun guy"* by repeatedly correcting his friends about how they should behave in a zoo. We could consider "serious no fun guy" as a character prototype that has properties of "respecting rules," "set an example," "patronizer". We observed that by portraying this character actor A raises his status over the others.

This platform is maintained for awhile, until the moment in which actor B looks aggressively at actor A saying "I don't have a wife and i don't have children, like you do!". Now actor A is being criticized instead of criticizing. Actor B continues "And i wish you didn't have that wife and those children!". In the post performance interview actor B reports "...i decided to endow him with a horrible wife". The scene changes, now actor A is no longer the "serious no fun guy" that tells others what to do, but the unlucky buddy that got married to the wrong woman. Player A status drops provoking a status shift from a high value to a low value.

Based on this we consider that a tilt agent using a status shift technique could follow these steps:

1. Detect if a particular actor is gaining relatively more status than the others. Define that actor as a target.
2. Endow the target actor with properties of a low status character prototype (ex: Married to horrible wife)

3. Endow the non-target actors with properties of a character prototype with higher status than character selected for target actor.

4.5.4 Tilt: Property Inconsistency

This example is a minor tilt that occurs in scene B, where at an initial step player A endows himself with the prototype of a zoologist. Following our knowledge structure approach this means that the zoologist (platform element for who?) is associated to player A. Supposing that in the knowledge base a zoologist has the property of respecting animals with a high DOM associated. Although "respecting animals" is a property with a high DOM to Zoologist it can not be immediately added to the platform, because it has not been addressed yet. Further on, player A endows himself the attribute of not "not respecting animals", by saying "*animals are lesser than humans,*" which is an action that portrays "no respect for animals." This action has a negative value for consistency with "respects animals", which is an attribute with a very strong DOM associated to the zoologist prototype enacted by player A. Now player A is a zoologist that does not respect animals, and this tilts the action because now there is a property that breaks the normal "routine" of a zoologist, which can be explored.

In this case a tilt agent using our knowledge structure would create a tilt by endowing a character with a property which is inconsistent with a *who?* element of the target character. We consider that an agent performing this technique would follow two steps:

1. Select a property of an active *who?* element with a high DOM value that has not been addressed.
2. Select an action that endows the character playing the *who?* with a property that is not consistent with the property used in the first step.

4.6 Discussion

The retrospective protocol approach followed on the collection of Digital Improv's data, produced a unique source of information about the improv process in terms of the information about the actors cognitive process that can be extracted from it. We approached this data from a bottom up perspective that allowed us to induce a conceptual model of story development.

The approach of annotating story frames and drama frames, allowed us to observe the constitution of each story platform along a story. This annotation was particularly useful for detecting that not every platform element has to be established when a tilt occurs and to detect the main transformations on the individual story frames when a tilt occurs.

Every element of the story development model and of the tilt formalizations is supported by video data and consistent with the improv theory principles presented in Chapter 2. Adding to

these we had the concern to use a clear and objective terminology that could have as much as possible be represented in computational terms. One should note the effort made to establish a consistent relation between platform elements, which is much similar to a knowledge base structure. This allows us to argue in favor of the potential of the model in our research. The consistency of the model with improv theory supported in the experience of practitioners is also signal of the potential of the model to be generalized.

As any grounded theory, the results of this analysis could be improved. We believe that this could be achieved by repeating the analysis process with more observers. From the results of this observations, new elements could be found and more importantly, stronger results could be inferred from the analysis the similarities between different annotations. We do not propose our model as a unique general solution, but rather as an informed and data supported model that is consistent with our data and with general improv theory, bringing more insights about the process and a representation that is concerned about facilitating a computational representation of the model.

4.7 Summary

The functions identified in the conceptual model for story development presented in this chapter establish an agenda for implementing a collaborative story model for authorial-agents. The implementation of this research agenda includes a large number of AI challenges for years to come, most of them already identified in Section 3.3.3.

Given the relevance of the tilt process and platform representation in this model, we decided to start this agenda with a formalization of a platform representation and an analysis of the tilt process. In the next chapter, we report how the platform formalization was used to define a knowledge base structure for authorial-agents and how this knowledge base was used to in the implementation of authorial-agents that execute the tilt strategies identified in this chapter.

Tilt Agents Implementation

The implementation of an authorial agent prototype for tilt respected three main concerns:

- computational implementation of the two tilt strategies detected in the data analysis.
- mapping of the concepts used in the tilt analysis.
- authoring concerns: allow an easy manipulation of the knowledge base content and its extensibility.

In order to maintain the focus of research in the tilt mechanism using just the knowledge inferred from explicit actions, we introduced some simplifications in the implementation scenario. We minimized coordination and dialogue issues by considering a scenario of only two agents, instead of using three actors like in the analysis scenarios. The agents take alternate turns to execute an action. We also simplified the offer/response process by not considering offers that endow elements to other characters. An agent can infer story elements from the actions of the other agents, but only takes actions to influence location, props, their own activities and their own characters. Cooperation is perceived when an agent accommodates itself to his perception of the other agent. Some elements of this chapter are also reported in (Brisson et al., 2013).

5.1 Architecture Overview

Figure 5.1 presents an overview of the tilt agent architecture. Whenever an action is executed in the story world, the *emergent frame* updates a list of *story events*. Each *story event* is identified with a unique ID and contains: the action executed; a reference to the subject that executed the action and the turn at which the action occurred.

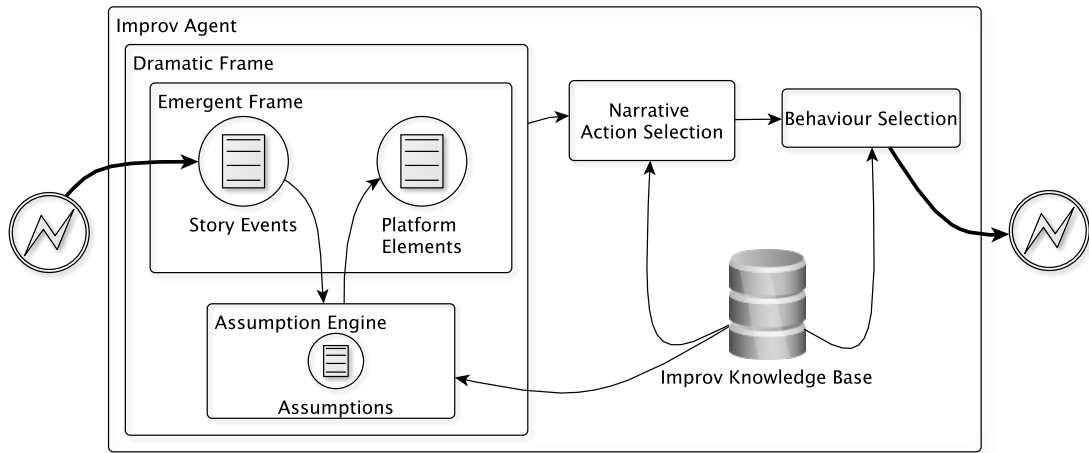


Figure 5.1: Tilt agent architecture

Platform Elements Collection			
Characters	Activities	Location	Props
[Id, Conf, Turn]	[Id, Conf, Turn]	[Id, Conf, Turn]	[Id, Conf, Turn]

Table 5.1: Platform elements collection

The Assumption Engine (AE) uses the last event to retrieve, from the Improv Knowledge Base, the platform elements that are most related with the event. The inference mechanism used in this step produces a collection of platform elements organized by lists of element types (see Table 5.1). Each retrieved element is identified with an Id value, the Turn at which the assumption was generated and scored with a Confidence value. The Confidence value represents how much the agent believes that the element is part of the scene at given turn (details on section 5.4).

Afterwards, the AE merges the new assumptions with the previous assumptions updating the confidence value for each element (details on section 5.4.1). Elements with a confidence value above an acceptance threshold for more than one turn are sent along with their confidence values to the Platform Elements collection, which represents the agent’s perspective over the story platform.

Using the Dramatic Frame as input, the Narrative Action Selection can detect if a platform is established or not, and based on that decide upon which narrative action to take (propose platform element, confirm element, tilt offer, tilt riding, etc...). Afterwards the Behavior Selection component selects, from the improv knowledge base, the action behavior that represents the previous options. In this implementation the Behavior Selection component was simplified to select the action that most strongly portrays an element, without repeating actions.

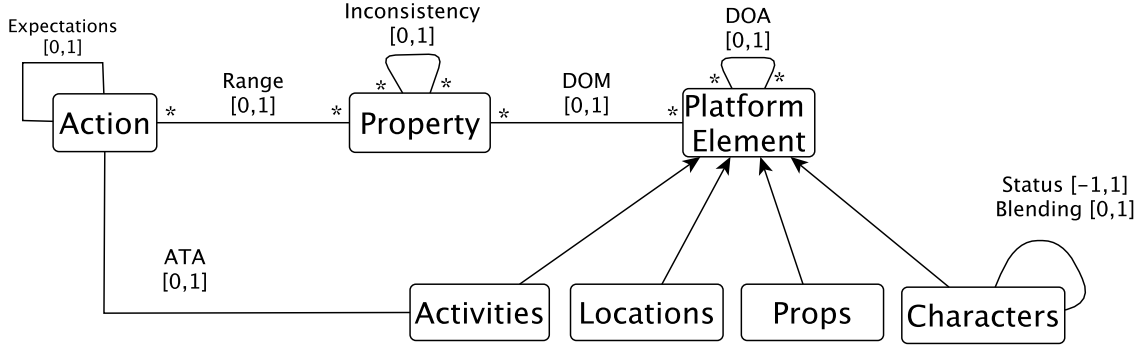


Figure 5.2: Knowledge structure

5.2 Knowledge Structure

The representation of Action, Property and Platform Element, in the Improv Knowledge Base, maps the model used previously in the tilt analysis, with the exception that their relations can now be quantified with fractional values between 0 and 1 (see Fig 5.2).

During the process of populating the knowledge base we observed that the properties used to describe activities were not very related with the properties of other elements and would not contribute to the tilt functions. We simplified the relation between actions and activities with a direct relation ATA (Action To Activity). ATA also includes a fractional value between 0 and 1, that represents how much an action can be used to describe an activity. This also simplified the process of populating activities.

Because of its complexity, the effort of creating a specific dialogue system for this prototype would cause our research to diverge from the main goal of isolating the study of tilt. We introduced a relation between actions called *Expectations*, which is used to represent how much the agent performing an action expects a specific range of actions in response. This way an agent that is expected to respond to another agent's action can choose between the possible responses those that better match his portraying intentions or even reject to respond.

The representation of the relations between characters includes two dimensions: status and blending. Status is represented with a value ranging from -1 to 1. The absolute value of status represents the status distance between two characters while its negative or positive values are used to define the direction of the status. E.g. a status of -3 between A and B, means that A has a lower status than B with a distance of 3, while a status of 7 between C and D means that C has a higher status than D with a distance of 7. Blending is a rough measure of how much an agent portraying a character can start portraying another specific character maintaining consistency. E.g. a Father character prototype has a low blending value with the Son character prototype, because it is not much expected that someone behaving like a Father suddenly starts behaving like a Son. This approach bootstraps the implementation of a conceptual blending mechanism (Fauconnier and Turner, 2003) which happens when two different concepts that share the same knowledge space are

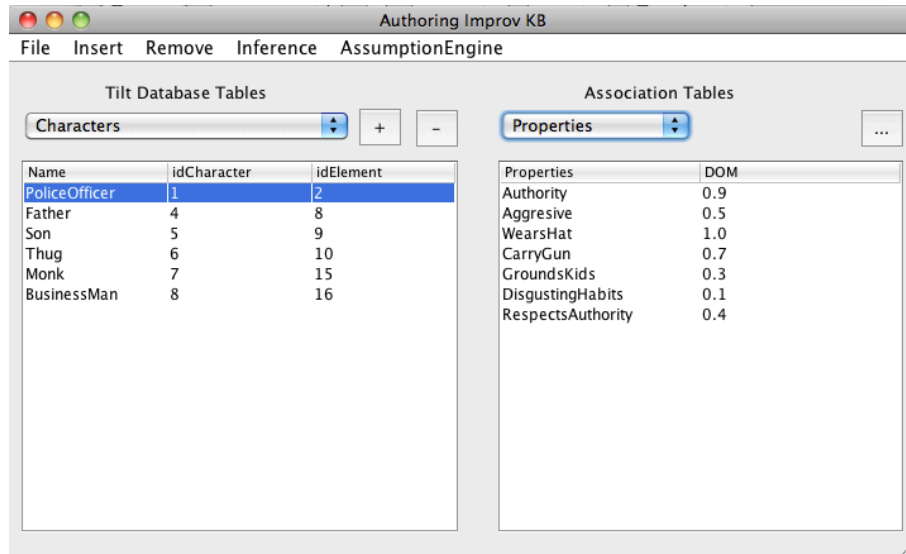


Figure 5.3: Screenshot of the KB menu for populating DOA

use to define a new idea or concept. Such, implementation could consist on developing functions that would generate new platform elements by gathering their properties in a new platform element and recalculate the corresponding DOM values. This would require further studies on the subject of conceptual blending that would take us beyond the scope of our research on tilt, nonetheless it presents a much relevant challenge for future work.

5.3 Populating Improv Knowledge Base

The knowledge base was populated by the developers using a visual tool (Fig. 5.3). The tool was specially developed for configuring the improv knowledge base. All the data is maintained in a relational data base management system using MySQL (see EER Model in Figure 5.4). The data base implements the knowledge structure for the improv knowledge base. It is possible to configure all the concepts and relations from the knowledge structure. It also allows to preview the results of the inference functions, or simulate sequences of assumptions. Details on the inference and assumptions update functions will be addressed in the following sections.

5.4 Assumption Engine

The assumption engine implements two main functions: *Knowledge Inference* (Fig. 5.5) and *Assumptions Update*:

Hierarchical Inference Hierarchical Inference receives actions as input for retrieving platform elements following two steps. First it retrieves all the properties related to the given action using the range value. Afterwards, it retrieves all the platform elements related to each property. The Inferred Confidence IC value for each inferred platform element e is a weighted average between the *Range* that relates the action a executed and the property p inferred, and the *DOM* values used to infer e from p :

$$IC(e) = \alpha Range(a, p) + (1 - \alpha) DOM(p, e), \quad (5.1)$$

where $0 \leq \alpha < 0.5$. The rationale behind this weighted average is to give more impact to the first step of the inference, which is more accurate than the second step that builds upon the first step results.

In the particular case of activities inferred using ATA the inference process is more straightforward and the IC value corresponds to the ATA value.

Associative Inference Given any platform element a , an associative inference returns all platform elements that might exist in the same story platform. The final IC value for a platform elements b resulting from an associative inference of a is a weighted average between the inferred confidence value of a and the DOA value that relates both elements:

$$Confidence(b) = \alpha Confidence(a) + (1 - \alpha) DOA(a, b) \quad (5.2)$$

where $0 \leq \alpha < 0.5$.

5.4.1 Assumptions Update

At the end of each turn, each agent calculates an Updated Confidence UC value for each element in the emergent frame. This value considers the IC values of a given element in each turn. Since whenever an actor makes a new offer in a scene he can reject or confirm the previous offers, we follow the rationale that the new IC values have a bigger contribution to the UC than the others. Based on this, we calculate the UC using a weighted average between the last IC for that element and the average of all its previous IC :

$$UC_n(e) = \alpha IC_n(e) + (1 - \alpha) \bar{IC}_{1:n-1}(e) \quad (5.3)$$

where n = number of turns when e is a platform element belonging to *Locations* or *Props* and n = number of subject turns when e belongs to *Characters* or *Activities*. $\alpha < 0.5$.

One should note that since the agents do not make offers to affect the other agents, the UC for characters or activities only considers ICs from the turns in which the agent that performed the

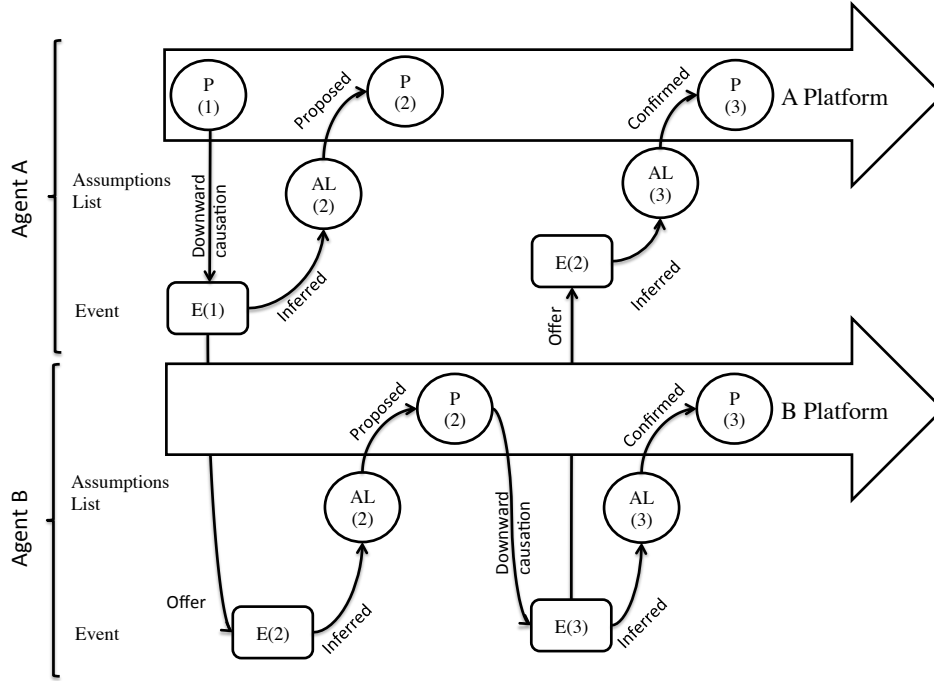


Figure 5.6: Offer / Confirmation sequence for a strong offer with a strong confirmation.

action took any action. It is also in this step that agents calculate a Confidence Increment (CI) for each UC, which is the difference between the new UC and the previous UC for an element. This value is used to detect the main changes in a platform and, consequently, detect tilt occurrences.

5.4.2 Platform Update and Tilt Detection

The process implemented for determining a platform (Figure 5.6) is an adaptation of the collaborative emergence process presented in Section 2.1.2 and of the coding process presented in Chapter 4. Instead of sharing a collection of elements in a shared frame, each agent builds its own platform composed by elements with high confidence values. The underlying assumption on this process is that each agent considers that the values with high confidence are consensual. Whenever the UC value for a given element is higher than the acceptance threshold, the element is sent to the platform collection. The element is marked as proposed when it is sent to the platform for the first time. This adaptation considers the emergence of a platform definition for each agent. When sent for the second time, the element is marked as confirmed. The main conceptual difference in this implementation is that we consider proposed elements as being part of the platform, while in the cooperative emergence process only confirmed elements are part of a platform. This allows us to use the proposed elements in the detection of tilt occurrences.

Following our working definition for tilt, *a significant change in the platform* and given our

observation, during the data analysis, that tilt is a process that results from the actors' offer / response interaction. We implemented Tilt Detection as a function that checks when tilt is being proposed. This function checks the assumptions in the platform for major variances in the confidence values. It does so by using the records of the CI (Confidence Increment) for each platform element and by considering that a tilt starts when at least one significant positive increment and a significant negative increment occur for the same type of platform element in a turn. E.g., an actor that switches from portraying a *police officer* to portray a *father* will have a strong negative CI for *police officer* and a strong CI for *father*.

5.5 Status Shift Implementation

Status shift is a tilt strategy based on the intensification and reversal of status dimension of the character's relations. During the intensification step, agents seek to gradually increase their status difference by portraying characters with a greater status difference, while maintaining the positions of high status and low status. The status reversal consists on the shift of the high status and low status direction.

Fig. 5.7 presents an overview of the tilt status implementation. The agent starts by checking if a status tilt is starting. To do this the agent checks if a tilt occurred in character assumptions for the other agent, and if the new strongest character for the other agent allows a status shift. When a tilt is not detected, the agent priority is to establish a relation with the other agent. The agent considers that a relation is established when both agents have confirmed character elements that have a relation defined in the improv knowledge base. To establish a relation with the other agent, the agent first checks if its own character is established and if his character has an antagonist relation, i.e. relations with a relevant status distance. If so, it reinforces its own character by executing an action that is highly related to his character. If a contrast relation is not found, it selects a new character with a high blending value with his strongest character. In the case of not having a character established, the agent either chooses to blend with the status antagonist of the other agent's character, or portrays a random character if the other agent does not have a character established. Next we address the two most relevant functions for this process: *Tilt Status* and *Portray Status Antagonist*.

5.5.1 Tilt Status Function

When an agent detects that both agents have established characters and that the established characters have an antagonist relation between them, it executes the *Tilt Status* function. *Tilt Status* searches the improv knowledge base for a target character element following these steps:

- Selects from all the *Relations* those in which one of the characters has a high blending value to its own established character and the other character has a high blending value to the

other agent's strongest character.

- Selects from the relations resulting from the previous step, all the relations in which the character with a status value opposed to his current status have a high blending value to his character.
- Selects from the resulting set the relation with the highest status difference.
- Portrays the character from the resulting relation, that has a high blending value to his character and an opposite status.

5.5.2 Portray Status Antagonist Function

After detecting that a status tilt was started by the other, the agent will try to ride the tilt by adapting his character to an antagonist of the other agent character. It does so by searching the other agent's character relations for the relation with a status shift, which has the strongest status difference and in which the other character is compatible to its current established. We consider that the highest status difference will produce more impact and that minimum blending value should contribute to maintain some consistency. Our implementation consists on the following steps:

- Selects from *Relations* all the pairs in which the other agent's character has a new shifted status.
- Selects from the previous set, the relation with the highest status difference that has a blending value, towards his strongest character, above a given threshold.

5.6 Property Inconsistency Implementation

Property Inconsistency was implemented with the goal of maintaining as much as possible the structure provided by the Status Shift tilt strategy (Fig. 5.7). This constraint was assumed as a contribution to facilitate the future integration of both tilt strategies in the same agent. *Tilt Started* was adapted to detect the occurrence of property inconsistencies, *Tilt Status* was replaced by *Tilt Property*. Since status change is not the goal of this strategy, although it might occur, instead of riding a tilt using *Portray Status Antagonist* the agent disregards the need for a status shift aiming only at portraying an antagonist of the new perceived character. Next we address the implementation of *Tilt Started* and *Tilt property Inconsistency*.

5.6.1 Tilt Property Inconsistency Function

This function is used by the agent to select an action that is inconsistent to his strongest character. From our tilt analysis, a property inconsistency happens when an actor executes an action that

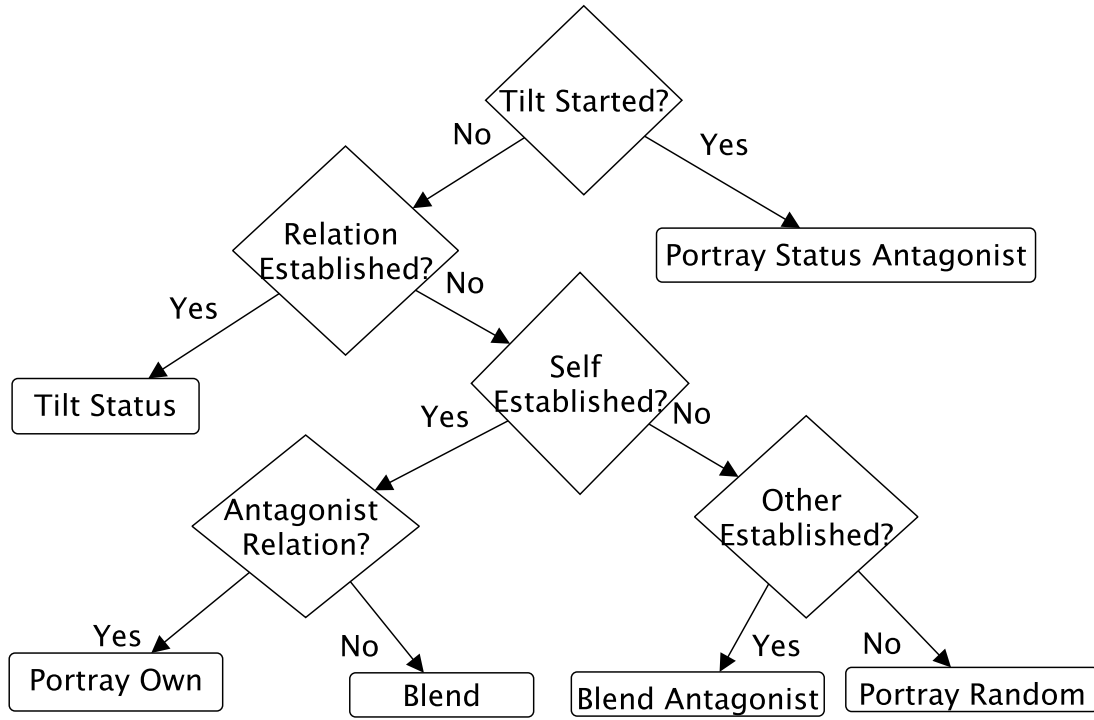


Figure 5.7: Tilt: status shift implementation overview

portrays a property that has a high inconsistency value with the character that the agent is portraying. We apply this definition in the Action Inconsistency (AI) function that measures the inconsistency between an action and a character. This function is given by:

$$AI(a, c) = \alpha I(p1, p2) + \beta Conf(c, p1) + \gamma Range(p2, a), \quad (5.4)$$

where $\alpha + \beta + \gamma = 1$, AI is the Action Inconsistency, I is the Inconsistency, $a \in Actions$, $c \in Characters$ and $p1, p2 \in Properties$. AI between a character c and an action a is calculated by the weighted average of three values: the inconsistency value between two properties $p1$, $p2$, in which $p1$ has a DOM relation with c and $p1$ is inconsistent with $p2$; the confidence value that relates the consistent property $p1$ and c ; the range that relates the inconsistent property to the action being evaluated.

5.6.2 Tilt Started Function

Tilt started was modified to detect the occurrence of a property inconsistency. At the beginning of the action selection the agent evaluates the inconsistency of the action that was executed in the previous turn. A tilt is detected when the agent acknowledges a high value for AI of the action executed in the previous turn.

5.7 Discussion

The agent architecture maintains consistency with the concepts of the improv theory presented in Chapter 2 and with the data coding approach presented in Chapter 4. The implementation approach of the tilt functions is in line the results of the data analysis. The improv knowledge base structure is extremely similar to the knowledge structure used in the data analysis. Nevertheless, some modifications, such as the *action to activity* or *expectations* were introduced in the implementation process. This is due to the fact that the data analysis procedure was designed to provide general insights about the cognitive processes used in the tilt process. In the other hand the tilt agent implementation focused specifically on implementing tilt on an environment with far more constrains.

The process of populating the knowledge base was extremely facilitated by the use of the visual tool. The implementation of the knowledge base in MySQL creates several opportunities for further improvements on the populating process, since it allows for an efficient management of large data bases. It also establishes a clear division between the knowledge used by the agents and their decision processes, since all the knowledge in the improv knowledge base is independent of the functions that manipulate it.

The authoring process for the scenarios was detached from the narrative creation process. This detachment contributed to reduce the authoring burden of the system, although it could be improved by allowing non-developers to contribute. Such crowd-sourcing approach would also allow for demonstrating the behaviors of the agents with a larger amount of data.

5.8 Summary

This chapter summarized the implementation of authorial agents that with tilt abilities as well as design and implementation of the knowledge base used by the agents. In the next chapter we will present the most representative results produced by these agents.

CHAPTER 6

Case Studies

The research approach presented in this thesis relies on the adaptation of real acting techniques to autonomous agents in order to enable the creation of authorial-agents. We focus on the effort of adapting acting techniques to the development of authorial-agents. Instead of creating full interactive storytelling scenarios with a complete story and user interaction, we choose to implement small acting contexts with no users where the agents can demonstrate the potential of their acting techniques in a controlled environment. This setting allows for a faster development cycle while maintaining the balance between the reduction of technical dependencies and the demonstration of the research results. In this chapter, we will present and discuss two case studies that follow this same motivation. The first case study is based on improv and corresponds to running simulations of the tilt agents introduced in Chapter 5 and reported in (Brisson et al., 2013). The second case study used *emotional escalation agents* that were built for humour context. Emotional escalation agents were implemented as a FATIMA (see Section 3.2.3) extension specially designed to consider the actors ability of buildup up tension on a comic sketch in order to create the right timing for a comic punchline. The agents in the second study are *Emotional Escalation Agents*. We discuss the issues that are exclusively related to each scenario within the corresponding case study and conclude with an overall discussion of the shared issues and their potential integration.

6.1 Tilt Agents

In this section we present a simulation for each of the tilt strategies described on Chapter 5. We start by presenting an example of a *status shift* and follow with the presentation of a *property inconsistency*. In both cases the scenarios setup include two agents acting an improvised scene

Turn	Agent	Description	Dialogue	Platform Elements	
1	A	Demanding to see license and registration	"License and Registration please Sir!"	A.Identifying(P) B.none	A.Police(P)
2	B	Looking in the eyes	"I will not do it."	A.Identifying(C) B.Thugh(P)	A.Police(P)
3	A	Adjust hat placement over own head	"hmm hmm"	A.Identifying(C) B.Thugh(P)	A.Police(C)
4	B	Punches the other in the face	POW !	A.Identifying(C) B.Thugh(C)	A.Police(C)
5	A	Pouts and screams	"You never let me do anything!"	A.Identifying(C) B.Thugh(C)	A.Son(P)
6	B	Pointing finger and shouting	"I forbid you to keep that behavior!"	A.Identifying(C) B.Father(P)	A.Son(P)
7	A	Pouts and kicks the ground	"No! No! No!"	A.Identifying(C) B.Father(P)	A.Son(C)
8	B	Point finger	"You should not do that child!"	A.Identifying(C) B.Father(C)	A.Son(C)
9	A	Begging with sweet voice	"Please, please can i get some candy?"	A.Identifying(C) B.Father(C)	A.Son(C)
10	B	Looking down at the other	"You know... When i was your age i had to work to buy food"	A.Identifying(C) B.Father(C)	A.Son(C)

Table 6.1: Tilt example: Status Shift

from scratch in alternated turns. The improv knowledge base presented in Sections 5.2 and 5.3 was the same for both scenarios and for all the characters.

6.1.1 Tilt: Status Shift

The testing scenario for the status shift technique includes two agents that interact with each other in alternate turns.

Table 6.1 presents an example of a scenario run using this setup. The action starts with agent A executing *askForDocuments*. The action executed by A produces a high IC for a *Police Officer* character. Since there is no relation established, B looks for an antagonist relation and chooses portray *Thug*. Because *askForDocuments* is an action that expects a response, B searches for an action with a high *expectations* value with *askForDocuments* and a IC value with *Thug* high enough to propose this character.

At the ending of the second turn, the platform for both agents includes a proposed character element for each of them. The characters are not established but have an antagonist relation. In turn 3, A reinforces his character causing *Police Officer* state to change to confirmed (C). B takes the same decisions in the fourth turn confirming *Thug*.

There is a confirmed character value for each character at turn five, in which A plays a *Police*

Turn	Agent	Description	Dialogue	Platform Elements
1	A	Demanding to see license and registration	"License and Registration please Sir!"	A.Identifying(P) A.Police(P) B.none
2	B	Looking in the eyes	"I will not do it."	A. Identifying(C) A.Police(P) B.Thugh(P)
3	A	Adjust hat placement over own head	"hmm hmm"	A. Identifying(C) A.Police(C) B.Thugh(P)
4	B	Punches the other in the face	POW !	A. Identifying(C) A.Police(C) B.Thugh(C)
5	A	Looking at the other	"May God forgive this child's sins"	A. Identifying(C) A.Police(C) A.Monk(P) B.Thugh(C)
6	B	Immobilizes the other	"Do not move!"	A. Identifying(C) A. Identifying(C) A.Police(C) A.Monk(P) B.Thugh(C)
7	A	Staring at the sky	"Bless this magnificent day...!"	A. Identifying(C) A.Police(C) A.Monk(C) B.Thugh(C)
8	B	Fires several shots to the air	"Yeehaa!"	A. Identifying(C) A.Police(C) A.Monk(c) B.Thugh(C)

Table 6.2: Tilt example: Property Inconsistency

Officer that has a higher status than a *Thugh*. Instead of continuing to explore this relation A considers that a platform is established and that it can start a tilt. A finds a tilt in the Father/Son relation and starts behaving like a son. In the following turn B responds by selecting the *Father* character and a new tilt buildup cycle begins, with A playing a *Son* and B playing a *Father*.

6.1.2 Tilt: Property Inconsistency

To illustrate the differences between property inconsistency and status shift, we simulated the buildup of the scene with property inconsistency (first four turns) to repeat the buildup of the scene with status shift. We let the agents run autonomously after the buildup process.

In the description of the scene (Table 6.2) we see that at the fifth turn A executes the action *forgiveSins*. This action has a high AI value for *Police*, resulting from the *Inconsistency* between *Authority*, property used in the improv knowledge base to characterize characters that have an official authority, and *ForgiveSins* property that characterizes characters with a religious authority. Because the chosen action also has a high value for representing a *Monk*, agent B also detects that is proposing a new character. B responds by antagonizing with the character with a highest assumption value for A at the begging of turn 6, which at that instant is *Monk*. Since in this case status shift is not relevant, B chooses to portray the character in the improv knowledge base with the biggest status difference to *Monk* which continues to be *Thug* and a new buildup occurs to set up the new *Monk* and *Thug* relation.

6.1.3 Discussion

The example runs presented bring to attention the success of the tilt strategies and the limitations of the tilt buildup and tilt riding functions.

During tilt buildup and tilt riding, the agents apply an action selection policy that consists of selecting from the actions that have not been executed, the action that more strongly portrays their current strongest character. The most positive side of this simplification is that it accelerates the tilt buildup process allowing the agents to quickly propose and confirm their characters, quickly establishing a platform that can be tilted. On the down side, in the tilt riding story step the agents start building a new platform whose only functions are to emphasize the tilt occurrence and provide the grounding for a new tilt to happen. The final result is a cyclic output of a tilting game, which clearly illustrates the goal of the scenario and presents new research challenges, being one of these challenges the creation of a resolution.

Regarding a comparison between the tilt function and relying solely on our observations, the implementation of status shift tilt seems to have greater impact than the property inconsistency tilt. One of the reasons behind this is that the status shift implementation was based on the analysis of a strong tilt, while property inconsistency was based on a weak tilt that developed into a big tilt. Also, the status shift produces by definition a change in both agents while the same does not necessarily happen in a property inconsistency tilt, thus reducing an impact of the tilt and tilt riding functions in the platform.

The tilt functions also differ in terms of consistency with the data analysis. While status shift clearly operates over platform data to detect a tilt, the tilt detection function for property inconsistency relies on the analysis of the action inconsistency between the action performed and the character of the agent that executed the action. This is because the platform representation in the agent does not include a representation of properties. Such properties are inferred from the character elements represented in a platform.

6.2 Emotional Escalation Agent

This agent implements the concept of Emotional Escalation, a gradual intensification of the characters emotions (see Section 2.2). We do this by extending the FATIMA architecture (see Section 3.2.3) in order to include *emotional goals* over his own emotions and the emotions of other agents, that influence the characters' planning. In the following we present a brief overview of the Emotional Escalation implementation, the test scenario used in the evaluation and a discussion of the main contributions of the case-study to this thesis. This agent was developed in the context of Laugh To Me a project presented in (Carvalho, 2012; Carvalho et al., 2012) which is a joint work between the author of this thesis and André Carvalho.

6.2.1 Implementation Overview

While other approaches to humour (Cavazza et al., 2003; Olsen and Mateas, 2009; Thawonmas et al., 2003) rely on the generation of action incongruence through planning formalisms for failing characters actions, we approach incongruity from an emotional perspective. We define an emotional incongruence as an incongruence that arises from the personality of the agent itself, i.e., how differently the agent appraises the world considering what it would be expected of him. Emotional incongruence can be easily adapted to FATIMA by authoring the agent’s personality properties.

Two scripted elements are used to facilitate the sketch structure: the beginning and the punchline (the final joke that closes the sketch). However the timing for delivering the punchline is determined by a set of pre-conditions that define the Emotional Escalation: *Emotional Goals* and *Emotional Guidelines*:

- *Emotional Guidelines* - are a function of the emotion potential value over story time. Story time units are number of actions of the character. Potential is a value between 1 and 10 used to calculate the intensity of an emotion. The intensity is the difference between the potential and threshold for an emotion. Each *emotional guideline* defines a desired potential for an emotion at time point. The agent is proactive in the search for actions that minimize the difference between his emotional values and the values defined in the *emotional guidelines*. To do this he simulates the impact of the actions to evoke the maximum value for the desired emotion and minimize the distance between the evoked value for that emotion and the guideline.
- *Emotional Goals* - are a set of pre-conditions that group *Emotional Guidelines* with a set of preconditions. When these preconditions are checked the *emotional guidelines* are activated and the agent will consider the guidelines in the action selection.

Emotional Goals and *Emotional Guidelines* allow the agent to be active in how he elicits specific emotions in other agents. The authoring of these elements can be used to establish a buildup towards the punchline. In Laugh To Me, these elements are authored using guidelines that determine a gradual intensification of the emotions, thus defining an Emotional Escalation.

6.2.2 Scenario

The test scenario was set in a pastry shop, where one agent plays a comic character (the Seller) and the another agent plays a Client (see Figure 6.1). The Client is a normal character who wants to buy a cake. The Seller is a comic character who refuses to sell a cake to the buyer. The Client is obese, and the attitude of the Seller ranges from being plain insulting to stress the fact that the Client is overweight as a reason not to sell the cake. The final punchline is a Seller’s attempt to sell something else. Since we defined a Seller that wants to make the Client angry, it seemed fitting to author the seller for trying to sell anti-depressants.

Figure 6.1: *Laugh To Me* screenshot

6.2.3 Authoring

The Seller is a comic character that wants to annoy the Buyer, hence we authored it in order to arouse Distress. The *emotional goal Displease Client-A*, consists of an initial sigmoid curve of the Client's Distress that is followed by an exponential growth curve. This makes for a change of pace in the sketch that starts slow but escalates fast after the initial setup.

Figure 6.2 presents the final output for the Seller's *emotional guideline*. One should note that since Distress is generated by negative desirability, the emotional values in Guideline refer to the Buyer's emotions.

The Buyer is a regular character. His personality is defined according to "normal standards" so that he appraises events, such as being insulted, as highly undesirable. The Client's authoring considers two main elements: the intention of buying a cake and the intention of opposing to the Seller's behavior. This was authored defining two *emotional guidelines*. The first *emotional guideline* concerns Joy and it is satisfied by performing the action *AskForCake*. This goal was only designed to define the beginning of the scene. The second *emotional guideline* is *KeepPride* and consists on performing *FormallyComplaint* actions that enable the Pride emotion.

Figure 6.3 presents the implementation of the *KeepPride* goal. The fast growing curve for Pride generates the selection of the *FormallyComplain* action. Although other actions were available to the agent this was the action that better satisfied the selection policy.

6.2.4 Evaluation

A prototype implementing the previously described scenario was evaluated in a study to determine if the *Emotional Escalation* was perceived and to assess the potential of this approach. Since

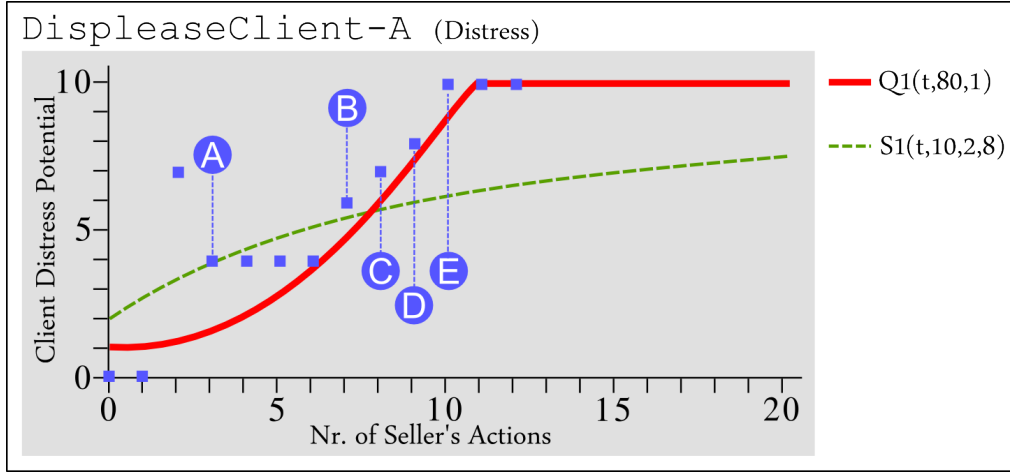


Figure 6.2: Emotional Goal DispleaseClient-A. Letters represent actions A-Raise Moral Issues, B-Reason, C-WarnHinderAppetite, D-MakeSarcasticRemark, E-MakeFatPeopleJoke, F-FormallyComplain. S_1 is the sigmoid curve and Q_1 is the quadratic curve.



Figure 6.3: Emotional goal KeepPride. Pride guidelines only, Joy not shown. Letters represent actions: A-AskCakeOrCandy, B-FormallyComplain

Question Number	Question
1	How did the Seller feel in the beginning of the sketch?
2	How did the Seller feel in the middle of the sketch?
3	How did the Seller feel in the end of the sketch?
4	[Do you agree] The Seller behaved as expected, given the situation?
5, 6, 7, 8, 9	[Do you agree] Same as 1-4 but regarding the Costumer?
10	[Do you agree] The sketch was too long?
11	[Do you agree] The sketch had a good ending?
12	[Do you agree] The ending should be better explained?
13	[Do you agree] The sketch was funny?

Table 6.3: Questions stated in the online questionnaire

Emotional Escalation was designed with goal of producing a growing emotional intensity we wanted to evaluate if this evolution was perceived by external viewers. In this context it is important to highlight the relevance of coordinating the emotional development of each character with the action development.

Based on this motivation we designed a questionnaire with tree main goals:

- Evaluate the subjects perception of the Seller’s emotional development.
- Evaluate the subjects perception of the Buyer’s emotional development.
- Evaluate the subjects opinion about the overall sketch development in terms of pace, comprehension, comic impact and characters emotional evolution.

The goals presented above were used to design a questionnaire with three sections (see Table 6.3). The first two sections were designed to evaluate the viewers perception about the emotional development of each character. To do this the user was first asked to select from a set of nominal values the one that corresponded the most to feelings of the character at the beginning, at the middle and at the ending of the sketch. Each of these sections ended with a question in which the subjects could classify the character’s consistency using a Likert scale. The third section aimed at evaluating the sketch development as whole in terms of perceived length, quality and comprehension of the punchline and the comic impact of the sketch.

The study was taken using an online questionnaire to which 75 subjects (37 males and 38 females) replied after viewing a screen cap video of the agents running the scenario. The participants clearly identified Happiness as the initial feeling of the Seller (60% of the answers to Q1) while a significative number did not identify the feeling as any present in the list (25,3%). The middle section of the sketch (Q2) presents less clear results, and Worry gathers only 36% of the responses and 28% are unable to find in the list a word that could express the Seller’s feelings. As for the ending part of the sketch (Q3) opinions are divided between answers associated with

Sadness (49,3%) and Disappointment (42,7%). The perceived emotions are thus consistent with both the actions and expressions of the Seller character. Initially the Seller feels glad for seeing the Client, thus Happiness seems the most appropriate answer. As the sketch unfolds, the Seller's smile fades to a neutral smile. Participants reported some doubts on how to interpret this, but decided the Seller was worried. In the ending part, the Seller fails to sell the antidepressant pills and, as a result, he sports an extremely sad smile. Participants recognized his sadness, and inferred the Seller got disappointed for not selling the antidepressants, from the actions and subsequent reaction

The Client's emotional escalation was even more straightforward than the Seller's. Being the regular character, most of the emotional escalation of the sketch was perceivable through him. The initial perception of the Client's feelings is similar to the Seller's, with Happiness being the mode answer to Q7 (57%). The evolution of the Client's feelings is then perceived as a growth of Anger (61,3% thought the Client was angry throughout the middle section – Q8 – of the sketch, and 70,7% – Q9 – in the ending part).

The majority of the participants (76%) totally disagree the Seller character behaved according to expectations (Q4). We can thus say the Seller was recognized as the *incongruent* character. In contrast, participants agree the Client behaved as expected.

The answers on whether the viewers thought the sketch was funny (Q18) was not conclusive with 3 as the median value selected. Some correlations with other questions may provide a better insight on *why* the participants deemed the sketch funny or unfunny.

A Spearman correlation test indicates an inverse relation between perceived length (Q13) and funniness, with a correlation factor (ρ) of $-0,366$ significant at the 0,01 level. This helps make the case that pacing is indeed an important subject in Interactive Comedy. Spearman-rho correlation tests also indicate funniness of the sketch relates directly with the quality of the ending (Q15, ρ of 0,597, significant at the 0,01 level) as well as inversely with the need of a better explanation for the ending (Q16, ρ of $-0,356$ significant at the 0,01 level). This relation stresses the importance of the punchline of the sketch, and the way it derives from the buildup.

6.2.5 Emotional Escalation Discussion

Although the evaluation of the comic effect of the scenario remained inconclusive, the fact that users identified the desired emotional evolution is a good indicator of the success of *emotional emergence* for establishing a consistent character evolution, i.e., a character emotional transformation that is inline with the characters' personality and role.

Emotional escalation aimed at providing agents with a new emotional perspective over story development. The agents use the emotional perspective to actively reason over the impact of their own actions on the tension of a scene. This was achieved through the implementation of *emotional guidelines* and *emotional goals*. The agents are not just focused on playing a character but are also

actively concerned about the impact of their own actions on the tension of the scene. The character's consistency (even when incongruence is part of the character's consistency) is guaranteed by the configuration of the agent's appraisal and emotional personality variables. *Emotional goals* establish an emotional reference for the authorial-agents to follow. This reference is completely authored, the agent is free to adapt his action selection according to the *emotional goals* at the cost of maintaining the general character properties defined by the author. Nonetheless, *emotional escalation* is a relevant contribution to the authorial-agents paradigm. It reduces the authorial effort because it moves the author focus way from the complex task of authoring of goals and actions to a much simpler method.

Further contributions to authorial-agents could result from the study of the dependencies and effects of the *emotional guidelines* over story development. A computational representation for such dependencies and effects, could allow agents to actively adapt their emotional impact according to the story needs or even breakthrough in the consideration of the emotional impact over an audience.

6.3 Case Studies Discussion

The authorial-agents presented in this chapter include some differences that result from the fact that the agents were developed in different contexts, nonetheless they include some relevant similarities.

Both approaches rely on the analysis of acting methods and knowledge. Tilt agents relied on improv theory, while emotional escalation agents relied on theories of humor from psychology, writing and acting. The main difference in this case is that while tilt agents are supported by a cognitive study specially designed to assess the agents low level implementation requirements, the emotional escalation approach followed a top-down strategy. Both approaches successfully met their goals of providing a tilt mechanism or a gradual emotional evolution, endowing the agents with actors' behaviors and acting techniques. We argue that the emotional escalation agent top-down approach success results from the use of an appraisal mechanism strongly supported by psychological theory previously used and validated for autonomous characters. In spite of this difference in the foundations of their conceptual models, we consider that in both cases the research approach proposed by this thesis was applied.

The possible integration between these two approaches to authorial-agents is a natural question that we should discuss. Could an authorial-agent integrate tilt with emotional escalation? Our experience in the development of both agents allows us to defend that emotional escalation is extremely related to the tilt build up process used. In fact this integration could improve the tilt buildup process by allowing agents to actively create tension before starting a tilt. Furthermore, we see emotional escalation as a particular strategy for exploring a platform, hence it could also

contribute for a more interesting exploration of the story space defined by a platform. Nevertheless, there is a particularly relevant obstacle that should be considered, which is related to the authoring process.

While in tilt agents the story is completely created by the agents, in emotional escalation this is not the case. Emotional escalation agents are still confined to a character personality that is completely authored. An integration of emotional escalation would have to deal with this difference in one of two ways, either endowing the agents with the ability of building up its own personality or restrain tilt agents with a character personality. The first approach would provide more authorial power to the agents, hence it is more in line with the concept of authorial-agents. On the other hand the first approach would require the need to research and analyze a model of personality buildup adding further obstacles to the implementation. The second approach also presents some challenges, one of which is the limitations that maintaining a consistent personality may present to the tilt functions.

6.4 Summary

We presented two authorial-agents case studies. The tilt agents result from the application of the improv background and data analysis presented earlier in the text, while emotional escalation was implemented in a humour context from a completely theoretical background. We presented and discussed transcriptions of running examples of the tilt agents and the evaluation of emotional escalation agent and discussed the issues related to a possible integration between these agents.

In the next chapter we will compare the results presented in these case studies with the research challenges presented in Chapter 1. From this we will extract the main contributions of this thesis.

Conclusions and Future Work

In Chapter 1 we propose a new paradigm for applying autonomous agents to IS called *authorial-agents*. This new paradigm aims at shifting the authorial effort towards the autonomous agents by endowing them with the ability to reason about the impact of their own actions in the story development and change their actions accordingly. We advocated that the creation of *authorial-agents* relied on the definition of conceptual models that adapt acting know how to computational functions. This approach was followed by a cognitive study of the story creation process in improv (Chapter 4) that was used to define the story development conceptual model that defined the implementation of the tilt agent (see Chapter 5).

A similar approach was followed in the implementation of the *emotional escalation agent* summarized in Section 6.2. In this case the analysis step did not include a cognitive study, but rather a profound bibliographic research on humour and principles of comical sketch writing.

In the following, we relate the results presented in this thesis with its expected contributions (see Chapter 1). We complement the Chapter with additional conclusions that result from broader perspective of the research and with the future research directions proposed by the improvement of our research.

7.1 Improv Agents

Improvisation is an extremely difficult cognitive and cooperative challenge even for real actors. Such challenge presents novel opportunities to several AI disciplines, such as dialogue, mental models or computational creativity. In this work, we focused on modeling and implementing the tilt process using a very small subset of the explicit information exchanged in an improv scene.

We contributed to assess the cognitive processes involved in the cooperative creation of stories. More particularly, we proposed two different tilt strategies based on the results of a cognitive study. This contribution validates our motivation towards using improv as a case study for cognitive processes that can enrich the authorial power of agents in storytelling contexts.

The story creation process in improv continues to present many challenges beyond tilt. The platform buildup process aims at cognitive consensus and as such it could be improved by extending the dialogue process with a theory mind.

Tilt riding could be improved, by extending the concept of blending to the portraying functions of an agent that changes character. The agent could consider that the new character is not the character prototype but a mixture of the characters that he portrayed in a the scene. Furthermore, the particular technique of property inconsistency could be improved by promoting a character blending between the original character a character that is highly related to the new property.

The tilt process itself can be improved by studying more improv examples with the purpose of raising evidence over other tilt strategies. One should note that the strategies reported in this thesis only use a very minor set of the existing knowledge in an improv scene. We argue that from retrieving inputs from further data analysis, together with further studies over the challenge in platform buildup and tilt riding, new tilt strategies will be found.

The resolution story step aims at resolving problems and discrepancies raised in the story. This presents causality representation and problem resolution challenges.

The *conceptual story development model using tilt* was essential in our attempt to formulate the functions involved in a collaborative story development process for authorial-agents. This story model allowed us to perceive and contextualize the computational complexity required to develop an authorial-agent based in improv. By contextualizing each function in a single model we were able to express the relevance of tilt in formal terms. Although this relevance was already clear for theorists and practitioners, including the subjects of the cognitive study, it had no been proposed yet in a conceptual model specifically designed for the purpose of a computational implementation. The relevance of tilt in this formal model reinforced our initial motivation for approaching the authorial-agent from the tilt perspective, since the implementation of such a nuclear element will certainly influence the design of the other functions.

We defend that the *conceptual story development model using till* is a relevant contribution for authorial-agents, because it establishes an integrated research agenda. Each function of the story development model can be associated to specific acting techniques that should be studied and analyzed for supporting new models. The resulting models should be applied in the creation new authorial-agents that can be integrated in a single authorial-agent that applies the whole story development model.

7.1.1 Further Cognitive Assessment for Story Development

Future work on this topic should include new cognitive studies similar to the one presented in this thesis, meaning that new data should be collected following the same protocol but in different scenarios and that it should be evaluated using the same approach used in this research. The studies should include improv performances with constraints and interviews specially designed to consider four main goals: platform exploration, tilt riding, problem resolution and tying loose ends. We propose six different improv games that could be used in the studies.

- *Tilt Tilt Tilt* - In this game two actors must continuously tilt a scene as much as possible.
- *Make More Interesting* - "One player starts a scene. As soon as the trainer claps his hands, the player needs to make whatever he was doing more interesting, without advancing. If our player was opening a briefcase at the clap, he needs to make 'opening a briefcase' more interesting; he is not allowed to take something out of the briefcase, as that would be advancing; player needs to stick to 'opening the briefcase'." (Imp, 2012)
- *Tilt Now!* - Whenever the outsider says "TILT!", the actors must tilt the scene. This way the platform exploration and tilt riding can be extended without an explicit intention of tilting a scene allowing to evaluate the particularities of these story moments.
- *Major Problem* - similar to the previous scenario, although in this case the outsider would say "MAJOR PROBLEM!" and the actors would have to propose and adapt to a highly dramatic problem and resolve it.
- *Supermen* - "First player gets from the audience a silly little problem, like My Shoelaces are Untied, and a simple object, say a kitchen cabinet. First player starts a scene in which the Problem arises, and her character is unable to fix the problem. Hence she calls in (explicitly) the help of Kitchen Cabinet Man. This is a SuperHero, like Superman, Spiderman, you know that kind of cartoon-character hero types. Our Superman comes in with lots of brou-haha (high Status) only to make the problem worse (and a status switch to low status). So our hero calls in the help of yet another Superman. Use whatever you can think of first. SayToothpaste Man. Again this hero comes in high status, screws up even more, does a status switch and calls in yet another hero, who finally fixes the problem." (Imp, 2012)
- *Crisis Situation* - "2 players on stage approach each other with a crisis, and an object unrelated to the crisis. After each has presented his or hers, each solves the other's crisis with their own object. Replies must be instantaneous and may be ridiculous." (Imp, 2012)

In table 7.1 we associate each of the proposed games to its potential contributions to the improvement of the collaborative story model.

Game	Platform Explo- ration	Tilt	Tilt Riding	Problem Resolution	Tying Loose Ends
Tilt Tilt Tilt		X			
Make More Interesting	X				
Tilt Now!	X	X	X		
Major Problem	X	X	X		
Supermen		X	X	X	
Crisis Situation				X	X

Table 7.1: Each game proposed was designed to elicitate contributions to specific research goals

7.2 Emotional Escalation Model

Action development in the sketch structure used in Laugh To Me is based on the concept of *emotional escalation*. The agents that apply *emotional escalation* are *authorial agents*, because they behave not only as characters but also as actors that play characters. The agents guide their action selection according to the *emotional guidelines*. The pacing of the sketch can be controlled by the shape of these *guidelines*, and how fast or slow they contribute to the *emotional escalation*, which is the evolution of emotions towards an emotional peak in which the sketch is resolved.

The preliminary results of our study indicate a promising start for this approach on *authorial agents*, since the viewers identified this process and the evolution of emotions in the agents.

The *emotional escalation authorial-agent* was implemented as a prototype built upon the FA-tiMA agent architecture, and tied it to an animation system that is capable of expressing the agents emotions and thus portraying the emotional escalation. The assessment of the comic value of the resulting sketch is encouraging albeit, non-conclusive. The relation between the perceived length of the sketch and its funniness suggests pacing should be a topic of interest in Interactive Comedy.

This work contributes to how Interactive Storytelling, the comedy genre and autonomous affective agents may complement each other in the form of authorial-agents. The *emotional escalation* model still relies on considerable authoring, both for the characters personalities and on the *emotional guidelines*. Nonetheless, this effort would be substantially heavier if we considered the use of the classical FA-tiMA authoring approach with the aim of producing the same emotional development. Furthermore, the authoring of the comic character’s personality based on incongruence avoids the repetitive task of tweaking actions and goals in order to produce the same effect.

Another advantage results from the fact of *emotional guidelines* being represented in terms of emotional potential, instead of dealing directly with the actions and goals of the agent. This fact contributes to enhance its reusability, either the reusability of the characters or the reusability of

the emotional passing in another context.

With further understanding of how the evolution of emotions of characters takes place in a collaborative story creation environment, we could improve the agents ability to change his behavior according to his appraisal of the story development.

7.3 Further Conclusions

Our data analysis approach has proven to be very time consuming, nevertheless it also has proved to contribute to raise evidences that can be applied to computational models. This is why we conclude that the analysis process is a valid approach for the purpose of improving authorial-agents.

The improv knowledge base structure is independent of the authorial mechanisms used in the agents. Instead of authoring planning mechanism where the story development is mixed with its content, the development of the tilt agents is clearly focused on the development of functions that could better use the world knowledge. This is a clear contribution to move the authorial efforts from the author to the agents.

The separation between improv knowledge base and the improv agents implementation facilitates the data extensibility. In future work we want to extend the content of knowledge using crowd sourcing techniques, in order to allow the agents to produce a larger variability of scenarios.

The collaborative efforts that actors take in the unpredictable environment of improv makes it still the best real life metaphor for interactive storytelling. Autonomous agents can improve their storytelling capabilities by implementing computational representations of the large quantity of storytelling techniques used in improv. It is possible to develop computational models of interactive storytelling techniques based on the analysis of improv. By bringing these techniques from the real world to the autonomous agents realm, we are contributing to the creation of *authorial-agents* and thereby reducing the authorial burden.

7.4 Summary

Research in *authorial-agents* contributes to shift authorial effort towards autonomous agents. This was achieved in two forms, in the form of *tilt agents* and in the form of *emotional escalation agents*. The tilt strategies applied in the *improv agents* establish a first bridge between two future research directions, one concerning the platform buildup step and the other concerning the processes that follow tilt. Further research of these topics should consider new cognitive studies for which we propose a new set of improv games that could be used for specific contributions. The *emotional escalation agents* contributed towards the reduction of the authorial burden in comic sketch context. Further studies assessing the role of emotions in collaborative story creation, could contribute to an adaptation of *emotional escalation* to other genders and particularly to IS using authorial-agents.

Bibliography

Improv encyclopedia. Website: <http://improvincyclopedia.org>, October 2012.

A. N. Applebee. *Child's Concept of Story: Ages Two to Seventeen*. University of Chicago Press, 1980. ISBN 0226021181. URL <http://www.amazon.co.uk/Childs-Concept-Story-Ages-Seventeen/dp/0226021181>.

E. S. Athomas, E. Singer, A. Goldberg, K. Perlin, C. Castiglia, and S. Liao. Improv: Interactive improvisational animation and music. URL <http://130.203.133.150/viewdoc/similar;jsessionid=76CEE574C160A529DFDB80511372FA7B?doi=10.1.1.11.8594&type=sc>.

R. Aylett and S. Louchart. If i were you: double appraisal in affective agents (short paper). In *Seventh International Conference on Autonomous agents and Multiagent Systems*, 2008. doi: 10.1.1.149.4114.

R. Aylett, S. Louchart, J. Dias, A. Paiva, M. Vala, S. Woods, and L. Hall. Unscripted narrative for affectively driven characters. *IEEE Computer Graphics and Applications*, 26(4):42–52, 2006.

B.-C. Bae and R. M. Young. A use of flashback and foreshadowing for surprise arousal in narrative using a plan-based approach. In *Proceedings of the International Conference on Interactive Digital Storytelling (ICIDS 08)*, pages 156–167, Erfurt, Germany, November 2008.

J. Bates, A. B. Loyall, and W. S. Reilly. An architecture for action, emotion, and social behavior. In *Selected papers from the 4th European Workshop on on Modelling Autonomous Agents in a Multi-Agent World, Artificial Social Systems*, MAAMAW '92, pages 55–68, London, UK, UK, 1994. Springer-Verlag. ISBN 3-540-58266-5. URL <http://dl.acm.org/citation.cfm?id=646907.710645>.

- A. Baumer and B. Magerko. Narrative development in improvisational theatre. In *ICIDS '09 Proceedings of the 2nd Joint International Conference on Interactive Digital Storytelling: Interactive Storytelling*, pages 140–151, Guimarães, Portugal, December 2009. Springer-Verlag Berlin.
- A. Brisson, B. Magerko, and A. Paiva. Tilt riders: Improvisational agents who know what the scene is about. In *Proceedings of the 10th Intelligent Virtual Agents*, volume 6895 of *Lecture Notes in Computer Science*, pages 35–41, Reyjavik, Iceland, September 2011a. Springer.
- A. Brisson, B. Magerko, and A. Paiva. A computational model for finding the tilt in an improvised scene. In *Fourth International Conference on Interactive Digital Storytelling, ICIDS 2011*, volume 7069 of *Lecture Notes in Computer Science*, pages 158–163, Vancouver, Canada, November 2011b. Springer.
- A. Brisson, B. Magerko, and A. Paiva. *A Bottom Up Approach to Create Authorial-Agents based on Improvisational Theatre*. Special Issue on Computational Narrative and Games. IEEE Computer Society (to appear), 2013.
- T. A. Buckley. *The Poetics (Great Books in Philosophy)*. Prometheus Books, 1992. ISBN 0879757760. URL <http://www.amazon.com/Poetics-Great-Books-Philosophy/dp/0879757760>.
- A. Carvalho. Laugh to me: Exploring computational humour. Master’s thesis, Instituto Superior Técnico, Technical University of Lisbon, 2012.
- A. Carvalho, A. Brisson, and A. Paiva. Laugh to me! implementing emotional escalation on autonomous agents for creating a comic sketch. In *ICIDS '12 Proceedings of the 5th Joint International Conference on Interactive Digital Storytelling: Interactive Storytelling*, Lecture Notes in Computer Science. Springer-Verlag Berlin, November 2012.
- M. Cavazza, F. Charles, and S. J. Mead. Character-based interactive storytelling. *IEEE Intelligent Systems*, 17:17–24, 2002. ISSN 1541-1672. doi: <http://doi.ieeecomputersociety.org/10.1109/MIS.2002.1024747>.
- M. Cavazza, F. Charles, and S. Mead. Generation of humorous situations in cartoons through plan-based formalisations. In *CHI-2003 Workshop: Humor Modeling in the Interface*. Citeseer, 2003.
- M. Cavazza, J.-L. Lugin, D. Pizzi, and F. Charles. Madame bovary on the holodeck: Immersive interactive storytelling. In *Proceedings of the 15th international conference on Multimedia*, pages 651–660. ACM, 2007. doi: 10.1.1.116.4348.
- J. Coleman. *The Compass: the improvisational theatre that revolutionized American comedy*. University of Chicago Press, 1991.

- J. Copeau. *Texts on Theatre*. Routledge, 1990. ISBN 041505253X. URL <http://www.amazon.co.uk/Texts-Theatre-Jacques-Copeau/dp/041505253X>.
- J. Dias and A. Paiva. Feeling and reasoning: A computational model for emotional characters. In C. Bento, A. Cardoso, and G. Dias, editors, *Progress in Artificial Intelligence*, volume 3808 of *Lecture Notes in Computer Science*, pages 127–140. Springer Berlin / Heidelberg, 2005. URL http://dx.doi.org/10.1007/11595014_13.
- O. Dictionaries. Humour. Website: <http://oxforddictionaries.com/definition/english/humour?q=humour>, October 2012.
- T. Doce, J. Dias, R. Prada, and A. Paiva. Creating individual agents through personality traits. *Intelligent Virtual Agents*, pages 257–264, 2010.
- L. Egri. *The Art of Dramatic Writing*. Simon & Schuster, New York, first edition, 2004. ISBN 978-0-671-21332-9.
- R. Evans. Linden research, inc. Website: <http://www.versu.com/>, June 2013.
- G. Fauconnier and M. Turner. *The Way We Think: Conceptual Blending And The Mind's Hidden Complexities*. Basic Books, Mar. 2003. ISBN 0465087868.
- R. Figueiredo, A. Brisson, R. Aylett, and A. Paiva. Emergent stories facilitated. In *Proceedings of the Second International Conference on Interactive Digital Storytelling, Erfurt, Germany.*, pages 218 – 229. Springer Berlin / Heidelberg, 2008.
- B. G. Glaser and A. L. Strauss. *The discovery of grounded theory: strategies for qualitative research*. Aldine Transaction, New Brunswick, N.J., 2012. ISBN 9780202302607 0202302601.
- A. Goldberg. Improv: A system for real-time animation of behavior-based interactive synthetic actors. *Creating personalities for synthetic actors*, pages 58–73, 1997.
- B. Hayes-Roth and R. V. Gent. Story-making with improvisational puppets. In *Proceedings of the first international conference on Autonomous agents*, pages 1–7, Marina del Rey, California, United States, 1997. doi: 0-89791-877-0. URL <http://portal.acm.org/citation.cfm?id=267660>.
- B. Hayes-Roth, E. Sincoff, L. Brownston, R. Huard, and B. Lent. Directed improvisation. Technical Report, 1994.
- B. Hayes-Roth, E. Sincoff, L. Brownston, R. Huard, and B. Lent. Directed improvisation with animated puppets. *Conference companion on Human factors in computing systems - CHI '95*, pages 79–80, 1995. doi: 10.1145/223355.223438. URL <http://portal.acm.org/citation.cfm?doid=223355.223438>.

- B. Hayes-Roth, R. V. Gent, E. Sincoff, R. Huard, T. Feldman, J. J.-j. Yu, K. Isbister, and B. Lent. Story-making with improvisational puppets and actors. Technical Report, 1996. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.54.8875>.
- B. Hayes-Roth, R. V. Gent, and D. Huber. Acting in Character. In *Autonomous Agents and Multi-Agent Systems*, pages 92–112. Springer, 1997.
- R. Hodhod and B. Magerko. Reaching cognitive consensus with improvisational agents. In *Eighth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2012a.
- R. Hodhod and B. Magerko. A formal model of shared mental models for computational improvisational agents. In *Proceedings of the Twelfth Annual Conference on Intelligent Virtual Agents*, Santa Cruz, CA, 2012b. Springer.
- A. Jhala and R. M. Young. Evaluation of intelligence camera control systems based on cognitive models of comprehension. In *Proceedings of the International Conference on the Foundations of Digital Games (ICFDG 09)*,, pages 26–30, Orlando, Fl. Usa, April 2009.
- P. Johnson-Laird. How jazz musicians improvise. *Music Perception*, 19(3):415–442, 2002.
- K. Johnstone. *Impro: Improvisation and the Theatre*. Routledge, 1987. ISBN 0878301178. URL <http://www.amazon.com/Impro-Improvisation-Theatre-Keith-Johnstone/dp/0878301178>.
- K. Johnstone. *Impro for Storytellers (Theatre Arts (Routledge Paperback))*. Routledge, 1999. ISBN 0878301054. URL <http://www.amazon.com/Impro-Storytellers-Theatre-Routledge-Paperback/dp/0878301054>.
- F. C. Julie Porteous, Marc Cavazza. Narrative generation through characters’ point of view. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Toronto, Canada, May 2010. Springer.
- T. Kemper. *Status, power and ritual interaction: a relational reading of Durkheim, Goffman, and Collins*. Ashgate Publishing, 2011.
- G. Lakoff. *Cognitive models and prototype theory*. MIT Press, Cambridge, MA, 1999.
- M. Lim, J. Dias, R. Aylett, and A. Paiva. Creating adaptive affective autonomous npcs. *Autonomous Agents and Multi-Agent Systems*, pages 1–25, 2010.
- R. Linton. *The study of man: an introduction*. Century Social Science Series. Appleton-Century-C, 1950. ISBN ICCN-50011774.
- S. Louchart and R. Aylett. Narrative theory and emergent interactive narrative. In *Proceedings of 2nd International Workshop on Narrative and Interactive Learning Environments*, Edinburgh, Scotland, August 2002. doi: 10.1.1.111.1320.

- B. Magerko and M. Riedl. What Happens Next?: Toward an Empirical Investigation of Improvisational Theatre. In *5th International Joint Workshop on Computational Creativity*, Madrid, Spain, 2008.
- B. Magerko, W. Manzoul, M. Riedl, A. Baumer, D. Fuller, K. Luther, and C. Pearce. An empirical study of cognition and theatrical improvisation. In *Creativity and Cognition*, pages 117–126, Berkeley, 2009. ACM. URL <http://portal.acm.org/citation.cfm?id=1640253>.
- B. Magerko, C. Fiesler, and A. Baumer. Fuzzy micro-agents for interactive narrative. In *Proceedings of the Sixth Annual AI and Interactive Digital Entertainment Conference*, 2010a.
- B. Magerko, C. Fiesler, A. Baumer, and D. Fuller. Bottoms up: improvisational micro-agents. In *Intelligent Narrative Technologies III Workshop INT3 10*, pages 1–8. ACM Press, 2010b. URL <http://portal.acm.org/citation.cfm?id=1822317>.
- B. Magerko, C. DeLeon, and P. Dohogne. Digital improvisational theatre: party quirks. In *Intelligent Virtual Agents*, pages 42–47. Springer, 2011a.
- B. Magerko, P. Dohogne, and C. DeLeon. Employing fuzzy concept for digital improvisational theatre. In *AIIDE*, 2011b.
- S. C. Marsella and D. V. Pynadath. Psychsim: Agent-based modelling of social interactions and influence. In *the International Conference on Cognitive Modeling*, pages 243–248, 2004.
- R. Martin. *The psychology of humor: An integrative approach*. Academic Press, 2007.
- S. Mascarenhas, J. M. A. Dias, N. Afonso, S. Enz, and A. Paiva. Using rituals to express cultural differences in synthetic character. In *Autonomous Agents and Multiagent Systems - AAMAS 2009*, volume 1, pages 305–312. IFAMAAS, 2009.
- M. Mateas. Towards integrating plot and character for interactive drama. In *AAAI symposium*, 2002.
- M. Mateas and A. Stern. FAçaDE: An experiment in building a fully-realized interactive drama. In *Game Developer’s Conference*, 2003.
- M. Mateas and A. Stern. Interactivestory, <http://www.interactivestory.net/>, 2011. URL <http://www.interactivestory.net/>.
- J. McCoy, M. Treanor, B. Samuel, M. Mateas, and N. Wardrip-Fruin. Prom week: social physics as gameplay. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, FDG ’11, pages 319–321, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0804-5. doi: 10.1145/2159365.2159425. URL <http://doi.acm.org/10.1145/2159365.2159425>.

- J. McCoy, M. Treanor, B. Samuel, A. A. Reed, N. Wardrip-Fruin, and M. Mateas. Prom week. In *Proceedings of the International Conference on the Foundations of Digital Games*, FDG '12, pages 235–237, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1333-9. doi: 10.1145/2282338.2282384. URL <http://doi.acm.org/10.1145/2282338.2282384>.
- R. Mckee. *Story: Substance, Structure, Style and The Principles of Screenwriting*. It Books, 1997. ISBN 0060391685. URL <http://www.amazon.com/Story-Substance-Structure-Principles-Screenwriting/dp/0060391685>.
- J. R. Meehan. Tale-spin, an interactive program that writes stories. In *In Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 91–98, 1977.
- C. Moorman and A. Miner. Organizational improvisation and organizational memory. *Academy of management Review*, pages 698–723, 1998.
- J. L. Moreno. *The Theatre of Spontaneity*. Beacon House, 1983. ISBN 0685308480. URL <http://www.amazon.co.uk/Theatre-Spontaneity-J-L-Moreno/dp/0685308480>.
- J. H. Murray. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. The MIT Press, 1998. ISBN 0262631873. URL <http://www.amazon.com/Hamlet-Holodeck-Future-Narrative-Cyberspace/dp/0262631873>.
- D. Olsen and M. Mateas. Beep! beep! boom!/: towards a planning model of coyote and road runner cartoons. In *Proceedings of the 4th International Conference on Foundations of Digital Games*, pages 145–152. ACM, 2009.
- B. O'Neill and M. Riedl. Toward a computational framework of suspense and dramatic arc. In *Proceedings of the 4th International Conference on Affective Computing and Intelligent Interaction*. Springer Berlin / Heidelberg, 2011.
- A. Ortony, G. L. Clore, and A. Collins. *The Cognitive Structure of Emotions*. Cambridge University Press, 1990. ISBN 0521386640. URL <http://www.amazon.com/Cognitive-Structure-Emotions-Andrew-Ortony/dp/0521386640>.
- K. Perlin. Building virtual actors who can really act. In O. Balet, G. Subsol, and P. Torguet, editors, *Virtual Storytelling. Using Virtual Reality Technologies for Storytelling*, number 2897 in Lecture Notes in Computer Science, pages 127–134. Springer Berlin Heidelberg, Jan. 2003. ISBN 978-3-540-20535-7, 978-3-540-40014-1. URL http://link.springer.com/chapter/10.1007/978-3-540-40014-1_17.
- K. Perlin and A. Goldberg. Improv: a system for scripting interactive actors in virtual worlds. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 205–216, New York, NY, USA, 1996. ACM. ISBN 0-89791-746-4. doi: 10.1145/237170.237258. URL <http://doi.acm.org/10.1145/237170.237258>.

- G. PERRET. *Comedy Writing Step-by-Step: how to write and sell your sense of humour*. Samuel French Trade,U.S.; 2nd Revised edition edition (1 April 1990), 1990.
- M. O. Riedl and R. M. Young. Open-world planning for story generation. In *Proceedings of the international joint conference on artificial intelligence (ijcai)*, pages 1719–1720, 2005.
- M.-L. Ryan. *Avatars of story*. The University of Minnesota Press, Minneapolis, 2006. ISBN 0816646864. URL <http://books.google.com/books?hl=pt-PT&lr=&id=dzX3n66QRiEC&pgis=1>.
- R. K. Sawyer. *Improvised dialogues: emergence and creativity in conversation*. Greenwood Publishing Group, 2003. ISBN 1567506771. URL <http://books.google.com/books?id=fMiTmZqPL0kC&pgis=1>.
- R. C. Schank and C. Riesbeck. *Overview of TALE-SPIN. Inside computer understanding: five programs plus miniatures*. Lawrence Erlbaum Associates, Hillsdale, N.J., 1981. ISBN 089859071X 9780898590715.
- A. Seham. *Whose improv is it anyway?: beyond second city*. Univ. Press of Mississippi, 2001.
- M. Si, S. C. Marsella, and D. V. Pynadath. Thespian: using multi-agent fitting to craft interactive drama. In *AAMAS '05 Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 21–28, ACM New York, NY, USA ©2005, 2005a.
- M. Si, S. C. Marsella, and D. V. Pynadath. Thespian: An architecture for interactive pedagogical drama. In *Proceedings of the 2005 conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology*, pages 595–602, Amsterdam, The Netherlands, The Netherlands, 2005b. IOS Press. ISBN 1-58603-530-4. URL <http://dl.acm.org/citation.cfm?id=1562524.1562605>.
- M. Si, S. C. Marsella, and D. V. Pynadath. Thespian: modeling socially normative behavior in a decision-theoretic framework. In *Proceedings of the 6th international conference on Intelligent Virtual Agents, IVA'06*, pages 369–382, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-37593-7, 978-3-540-37593-7. doi: 10.1007/11821830_30. URL http://dx.doi.org/10.1007/11821830_30.
- M. Si, S. C. Marsella, and D. V. Pynadath. Proactive authoring for interactive drama: An author's assistant. In *IVA '07 Proceedings of the 7th international conference on Intelligent Virtual Agents*, pages 225–237. Springer-Verlag Berlin, Heidelberg, 2007.
- M. Si, S. C. Marsella, and D. V. Pynadath. Modeling appraisal in theory of mind reasoning. *Autonomous Agents and Multi-Agent Systems*, 20(1):14–31, 2010. ISSN 1387-2532. doi: <http://dx.doi.org/10.1007/s10458-009-9093-x>.

- V. Spolin. *Improvisation for the Theater 3E: A Handbook of Teaching and Directing Techniques*. Northwestern University Press, 1999.
- C. Stanislavski. *An Actor Prepares*. Theatre Arts Book, 1989. ISBN 0878309837. URL <http://www.amazon.com/Actor-Prepares-Constantin-Stanislavski/dp/0878309837>.
- I. Swartjes. *Whose Story Is It Anyway? - How Improv Informs Agency and Authorship of Emergent Narrative*. PhD thesis, University of Twente, January 2010.
- I. M. T. Swartjes and M. Theune. The virtual storyteller: story generation by simulation. In *Proceedings of the Twentieth Belgian-Netherlands Conference on Artificial Intelligence, BNAIC 2008, Enschede*, pages 257–265, Enschede, October 2008. University of Twente.
- L. Talmy. Force dynamics in language and cognition. *Cognitive science*, 12(1):49–100, 1988.
- R. Thawonmas, K. Tanaka, and H. Hassaku. Extended hierarchical task network planning for interactive comedy. *Intelligent Agents and Multi-Agent Systems*, pages 205–213, 2003.
- J. Vorhaus. *The comic toolbox: How to be funny even if you're not*. Silman-James Press Los Angeles, 1994.
- R. Young. Notes on the use of plan structures in the creation of interactive plot. *AAAI Fall Symposium on Narrative Intelligence*, pages 164–167, 1999.
- R. Young, M. Riedl, M. Branly, A. Jhala, R. Martin, and C. Saretto. An architecture for integrating plan-based behavior generation with interactive game environments. *Journal of Game Development*, 1, 2004.
- J. Zhu, K. Ingraham, and J. Moshell. Back-leading through character status in interactive storytelling. *Interactive Storytelling*, pages 31–36, 2011.

APPENDIX A

Coffeshop Scene Analysis

Coffee Shop / Office

No Constrains

Transcription

Turn	Player	Action	Comments	Story Step	Function
1	D1	< is using a sort of broom on the floor > <making a scratchy noise>	D1 has newObject(used to clean) D1 cleans the floor	Platform Buildup	Build Platform
2	D3	<opens a door> <walks to the middle, informal, relaxed> Hey! How are you? Gotta get that cup of Joe <looking to front stage> Hey looks like you guys got a new line of muffins this morning!	Location(interior)=True D1 and D3 are equals, maybe friends NewObject(Muffins) NewObject(Table) Belongs(location,D1) Visiting(location,D3)	Platform Buildup	Build Platform
3	D1	<holding the accessories he created> <looking to front stage> We got something a little different,	Muffins(Special) = True Like(muffins, D1)++ Muffins(Organic)=True	Platform Buildup	Build Platform
		there all organic, they are made in Colombia you know. And these are all Fair Trade muffins.	Muffins(From)=Colombia Muffins(FairTrade)=True		
4	D3	<Emphasizes with some disbelief> Fair Trade!	Muffins.FairTrade(relevance)++;	Tilt Buildup	Tilt Offer <i>Tilt Variable = FairTrade</i>
5	D1	And you know what? It feels good!	D1 likes Fair Trade Muffins.FairTrade(relevance)++;	Tilt Buildup	Platform Riding Raising <i>FairTrade</i>
6	D3	Yes, it's not just a buzzword. It's saving lives.	Muffins.FairTrade(relevance)++;	Tilt Buildup	Platform Riding Raising <i>FairTrade</i>
7	D3	<enters from right stage> <faces D1 with servant posture, holding some sort of tray, and latin accent> Mr. Coffin I made up some more muffins for you from homeland.	D3 works for D1 NewObject(Tray with Mufins) NewActivity(sell muffins,D1,D3)	Tilt Buildup	Build Platform
8	D1	I appreciate it. <looks around> <takes a breath> <turns to D3>	D1 Likes muffins	Tilt Buildup	Platform Exploration
9	D3	Please feed me!	D3 passes hunger	Tilt Buildup	Tilt Offer <i>Tilt Variable = [FairTrade,</i>

1

2

10	D1	<surprised> Huu! Well, how can't the muffin girl feed herself?	Problem	Platform Riding
11	D1	<confused> You know lead a fish to water...	Problem	Platform Riding
12	D2	<pointing to the others> Are, are <like formulating a question>	D2 knows that something is wrong with the other two. D2 believes D3 in hunger isn't fair.	
13	D1	<more confused> I mean. If I make If I show you how to fish...	Problem	Platform Riding
14	D3	Yes...		
15	D1	<still confused, looking around> Anyway... I don't know <leaves to back right stage, to store his tools>	Problem	Platform Riding
16	D3	<looking at D1 and D2> I need for the food alone sir...	Problem	Platform Exploration
17	D1	<takes 2 muffins from the tray in D3 hands> <offers a muffin to D2> Here. Have a muffin man. They're delicious.	Problem	Platform Exploration
18	D2	<concerned. holding a muffin> Thanks. <Tries to formulate something> I...	Problem	Platform Exploration
19	D3	<beging D2> Please eat my muffins!	Problem	Platform Exploration
20	D1	<in high status holding muffin> <facing D2, advising> Don't let ammm, Eurica come to you...	Problem	Platform Exploration
21	D2	<to D3> hummm, eurica!	Problem	Platform Exploration
22	D3	<to D2 Giggles>	Problem	Platform Exploration
23	D2	<to D3> Eurica, can I have a word with you?	Problem	Platform Exploration
24	D3	<submissive to D1, asking permission>	Problem	Platform Exploration

		<walks to center front stage next do D2>			
25	D1	Sure I'll leave you to alone		Problem	Platform Exploration
		<steps out to right stage, pretends to eat his muffin>			
26	D2	Hey thanks a lot Scott.		Problem	Platform Exploration
27	D1	Any time.		Problem	Platform Exploration
28	D3	<facing D2>		Problem	Platform Exploration
		Hey!			
29	D2	<to D3> Hi, do you like Scott?	D2 tries to establish relation between D3 and D1	Problem	Platform Exploration
30	D3	I do. I do. Do you like my muffins?		Problem	Build Platform
31	D2	I do. I love your muffins.		Problem	Platform Exploration
32	D3	Every time you eat my muffins, one of my children doesn't die.	Children dying is a new problem added to the scene that raises the stakes of the previous problem, that D2 will try to resolve by eating muffins.	Problem	Platform Riding <i>Tilt Variable = [FairTrade, D3.Hunger, D3.ChildrenDying]</i>
33	D2	<starts eating muffin very fast> <takes more muffins and puts them in his mouth>		Problem	Platform Exploration
34	D3	Please, please! Keep eating my muffins!		Problem	Platform Exploration
5					
35	D2	<with his mouth full of muffins> How many kids do you have?		Problem	Platform Exploration
36	D3	I have at least 15... Please, keep eating my muffins		Problem	Platform Riding Raising dying children stakes
37	D2	<still with mouth full of muffins> <acknowledges with his head> I'll just go and get this whole tray from your hands. <takes tray from her hands>	Tries to resolve Dying Kids problem and D3.Hunger	Problem	Tilt Offer <i>Tilt Variable = [FairTrade, D3.Hunger, D3.ChildrenDying]</i>
38	D3	<happily surprised> oow! Uau!		Problem	Platform Exploration
39	D2	<breathless> And a, you know, would you like to come with me? To leave this place?		Problem	Tilt Offer <i>Tilt Variable = [FairTrade, D3.Hunger, D3.ChildrenDying, D3.Escape]</i>
40	D3	<look rapidly to D1 and then to D2 again> (whispering) More than anything in the world!		Problem	Platform Exploration
41	D1	<staring at D3 and D2> Scott, can I have a word with you		Problem	Platform Exploration

		for a second?		
42	D2	<to D1> <condescendent> Scott, if I could just have a word with you for a second?	Problem	Platform Exploration
43	D1	<more condescendent> SCOTT ANDERSEN, IF I COULD HAVE A WORD WITH YOU FOR A SECOND?	Problem	Platform Exploration
44	D2	<more condescendent> SCOTT PHELPS, <places tray on muffins table front stage> <moves right stage to D1> YOU'RE ABOUT TO HAVE A WORD WITH ME!	Problem	Platform Exploration
44	D3	Huu! <moves scared with hands on her ears to front stage>	Problem	Platform Exploration
45	D2	<pointing at D1> You are taking advantage of this young Colombian!	Problem	Platform Exploration
46	D1	<Irony> Am i?	Problem	Platform Exploration

7

47	D2	<angry> Yes, she has 15 children at home that will die if these muffins don't get eaten!	Problem	Platform Exploration
48	D1	<angry> You know what? This all muffin business is a <u>counter fort</u> . Did you ever hear about <u>Pablo Escobar</u> ?	D1 pulls back the story by tilting it, it invalidates early assumptions about D3 explored family. It rejects the early problem.	Tilt Offer <i>Tilt Variable = [FairTrade, D3.Hunger, D3.ChildrenDying, D3.Escape, D3.DrugDealer]</i>
49	D3	<revolted> No, no! You did not Mr.Phelps!	Tilt Buildup	Platform Riding
50	D1	<to D3> EXCUSE ME Eurica, this is a private conversation!	Tilt Buildup	Platform Exploration
51	D3	<Angry pointing at D1> Mr. Phelps do not bring Mr.Escobar. <Anxious. Bytes her hand and tries to calm herself> Ai.....	Tilt Buildup	Platform Riding
52	D1	<to D2> How do you think she had does 15 kids and got them to the States?	Tilt Buildup	Tilt Riding

8

		She used to be a drug mule man.		Tilt Buildup	Build Platform
53	D2	<astonished to D3> Is this right? Is this true?		Tilt Buildup	Platform Exploration
54	D1	<to D2> Scott Andersen, I wouldn't lie to you.		Tilt Buildup	Platform Exploration
54	D3	<In high status> <takes the muffin tray and faces D1 and D2> Mr. Andersen, your 15 th muffin has a happy ending to it...		Tilt Buildup	Build Platform
55	D2	<takes another muffin, smiling>		Tilt Buildup	Platform Exploration
56	D1	I can't compete with that...		Tilt Buildup	Platform Exploration
57	D2	I'm already starting to feel a little... sociable...		Tilt Buildup	Platform Exploration
58	D3	<peaceful, to D2> Look at you, you're happy. They're happy muffins. <again to D1, with evilness> They're happy muffins Mr. Phelps		Tilt Buildup	Platform Exploration
59	D1	<scared> Oh my god! You've drugged this entire firm. No wonder I was breaking leaves in the break room.	In this turn D1 not only offers that everyone in the company is drugged, but also ties a loose end from his initial activity.	Tilt Buildup	Platform Riding
60	D3	So I have. And now I stole you're accountings secrets, now Escobar can go richest.		Problem	Tilt Offer <i>Tilt Variable = [FairTrade, <u>D3.Hunger</u>, <u>D3.ChildrenDying</u>, <u>D3.Escape</u>, D3.DrugDealer, D3.Robs company]</i>
61	D1	<laughing, dropping out of character> That's gonna be tough, considering where he is in the present day.	Story advances to resolution without resolving the company problem?	Problem	Platform Exploration
	Audience	Laughs			
62	D2	Moves closer to D1		Problem	Platform Exploration
63	D3	I have ????? <moves out celebrating>	D3 leaves with all the money and the scene starts to end without resolving the problem	Problem	
	Audience	Laughs			
64	D1	<condescendent to D2> Where's the third world now?		Resolution	Explore Platform
65	D2	<high> You're shirt tastes amazing		Resolution	Explore Platform
	Audience	Applause			

Analysis

D1 Frame	D2 Frame	D3 Frame	Comments

Turn 1

In the beginning of the scene D1 is performing a random activity “Breaking Leaves” outside, but D2 perceives that D1 “is sweeping or making some sort of maintenance”, “the first thing I saw was a café”(D2). D1 made an offer in form of an action that as interpreted activated the D2’s Café frame.

D1 Frame	D2 Frame	D3 Frame	Comments
Activity: <ul style="list-style-type: none"> D1 Breaking leaves(P) Location: <ul style="list-style-type: none"> Outside(P) Props: <ul style="list-style-type: none"> Rake(P) 	Activity: <ul style="list-style-type: none"> D1 Maintenance(R) Location: <ul style="list-style-type: none"> Café(R) Props: <ul style="list-style-type: none"> Mop?(R) 	Activity: <ul style="list-style-type: none"> D1 sweeping(R) 	Platform build up

Turn 2

D2 follows his Frame of Friendly Café, opens a door and enters with a very friendly attitude. At this point D1 realizes he is working inside and that D2 as close relation with him “There was sort of sharing of status... he is very comfortable with me and our surroundings”D2. He accepts the offer of being working inside because he realizes that his action might have been interpreted as being mopping the floor “where are we? does he think I was mopping?”

Comfortable relation in comfortable surrounding for both while he D1 working inside activates his Office frame with a good colleague relation. In this context muffins takes him to the break room.

Also from D3 perspective, since she was on the opposite side of the door created by D2 she assumes she is on the inside, so she could work in that place.

D1 Frame	D2 Frame	D3 Frame	Comments
Activity:	Activity:	Activity:	Platform established this

11

<ul style="list-style-type: none"> D1 Mopping(A) <ul style="list-style-type: none"> breaking leaves(A) D1 taking break(P) D2 taking break(P) Location: <ul style="list-style-type: none"> Inside(A) <ul style="list-style-type: none"> Outside Office(P) Characters <ul style="list-style-type: none"> D1 worker(P) D3 worker(P) Relation <ul style="list-style-type: none"> Co-workers(P) Status: Equal(P) Props: <ul style="list-style-type: none"> Rake or mop? (A) Muffins(R) Door(R) 	<ul style="list-style-type: none"> D1 Maintenance(R) D2 ? Location: <ul style="list-style-type: none"> Inside(P) Café(P) Characters: <ul style="list-style-type: none"> D1 Café worker(P) D2 Client(P) Relation <ul style="list-style-type: none"> Status: Equal(P) Props: <ul style="list-style-type: none"> Mop?(R) Muffins(P) Door(P) 	<ul style="list-style-type: none"> D1 sweeping(R) Location: <ul style="list-style-type: none"> CoffeeShop(R) Characters: <ul style="list-style-type: none"> D1 Café worker(P) D2 Client(P) D3 Someone from the inside. That already works in the institution Relation <ul style="list-style-type: none"> Status: Equal(P) Props: <ul style="list-style-type: none"> Broom (R) Muffins(R) Door(R) 	doesn’t mean that platform build up as ended, just that it is enough to start looking for a tilt. In spite of some divergences, both players are eluded to be in consensus about location, relation, character roles
---	--	---	---

Turn 3

D1 start thinking “What’s different about our muffins?”, D1 wants to make these muffins special and I thought they’re all natural, going forward to Fare Trade “toward politicizing this equal office”, D2 takes this offer as a set up brings attention to it “we’re building a set up, for exactly what she(D3) came for... we are going to be wrong about it... great lets go with that is gonna create the feeling of like you now, peace and help, and I’m expecting it all to go wrong and it’s gonna go in that track”.

D1 Frame	D2 Frame	D3 Frame	Comments
Activity: <ul style="list-style-type: none"> D1 Mopping(A) <ul style="list-style-type: none"> breaking leaves(A) D1 taking break(P) D2 taking break(P) Location: <ul style="list-style-type: none"> Inside(A) <ul style="list-style-type: none"> Outside 	Activity: <ul style="list-style-type: none"> D1 Maintenance(R) D2 ? Location: <ul style="list-style-type: none"> Inside(P) Café(P) Characters: <ul style="list-style-type: none"> D1 Café worker(P) 	Activity: <ul style="list-style-type: none"> D1 sweeping(R) Location: <ul style="list-style-type: none"> CoffeeShop(R) Characters: <ul style="list-style-type: none"> D1 Café worker(R) D2 Client(R) Relation	Platform established this doesn’t mean that platform build up as ended, just that it is enough to start looking for a tilt. In spite of some divergences, both players are eluded to be in consensus about location, relation, character

12

<ul style="list-style-type: none"> Office(P) Characters <ul style="list-style-type: none"> D1 worker(P) D3 worker(P) Relation <ul style="list-style-type: none"> Co-workers(P) Status: Equal(P) Props: <ul style="list-style-type: none"> Rake or mop? (A) Muffins(C) <ul style="list-style-type: none"> Colombian(P) Organic(P) FairTrade(P) Door(R) 	<ul style="list-style-type: none"> D2 Client(P) Relation <ul style="list-style-type: none"> Status: Equal(P) Props: <ul style="list-style-type: none"> Mop?(R) Muffins(C) <ul style="list-style-type: none"> Colombian(R) Organic(R) FairTrade(R) Door(P) 	<ul style="list-style-type: none"> Status: Equal(R) Props: <ul style="list-style-type: none"> Mop?(R) Muffins(C) <ul style="list-style-type: none"> Colombian(R) Organic(R) FairTrade(R) Door(P) 	roles
---	--	--	-------

Turn 4

D2 Confirms the Fair Trade value, by repeating it with a mocking tone.

D1 Frame	D2 Frame	D3 Frame	Comments
Activity: <ul style="list-style-type: none"> D1 Mopping(A) <ul style="list-style-type: none"> breaking leaves(A) D1 taking break(P) D2 taking break(P) Location: <ul style="list-style-type: none"> Inside(A) <ul style="list-style-type: none"> Outside Office(P) Characters <ul style="list-style-type: none"> D1 worker(P) D3 worker(P) Relation <ul style="list-style-type: none"> Co-workers(P) Status: Equal(P) 	Activity: <ul style="list-style-type: none"> D1 Maintenance(R) D2 ? Location: <ul style="list-style-type: none"> Inside(P) <ul style="list-style-type: none"> Café(P) Characters: <ul style="list-style-type: none"> D1 worker(P) D2 Client(P) Relation <ul style="list-style-type: none"> Status: Equal(P) Props: <ul style="list-style-type: none"> Mop?(R) Muffins(C) <ul style="list-style-type: none"> Colombian(R) 	Activity: <ul style="list-style-type: none"> D1 sweeping(R) Location: <ul style="list-style-type: none"> Inside(R) CoffeeShop(R) Characters: <ul style="list-style-type: none"> D1 worker(R) D2 Client(R) Relation <ul style="list-style-type: none"> Status: Equal(R) Props: <ul style="list-style-type: none"> Broom (R) Muffins(C) <ul style="list-style-type: none"> Colombian(P) Organic(P) 	Platform established this doesn't mean that platform build up as ended, just that it is enough to start looking for a tilt. In spite of some divergences, all players are eluded to be in consensus about location, relation, character roles

13

Props: <ul style="list-style-type: none"> Rake or mop? (A) Muffins(C) <ul style="list-style-type: none"> Colombian(P) Organic(P) FairTrade(C) Door(R) 	<ul style="list-style-type: none"> Organic(R) FairTrade(C) <ul style="list-style-type: none"> Door(P) 	<ul style="list-style-type: none"> FairTrade(C) <ul style="list-style-type: none"> Door(R) 	
--	--	--	--

Consistency

At this moment if we only consider the consistent information between the players we get

Activity: <ul style="list-style-type: none"> ? Location: <ul style="list-style-type: none"> Inside(A) Characters <ul style="list-style-type: none"> D1 worker(P) D2 ? D3 ? 	Relation <ul style="list-style-type: none"> Status: Equal(P) Props: <ul style="list-style-type: none"> Muffins(C) <ul style="list-style-type: none"> Colombian(P) Organic(P) FairTrade(C) Door(R)
---	--

Emergent Frame

At this moment if we consider Sawyer's frame definition, where only accepted offers can represent the emergent frame we get the following emergent frame representation:

Activity: <ul style="list-style-type: none"> ? Location: <ul style="list-style-type: none"> Inside(A) Characters <ul style="list-style-type: none"> D1 worker(P) D2 ? 	<ul style="list-style-type: none"> D3 ? Relation <ul style="list-style-type: none"> Status: Equal(P) Props: <ul style="list-style-type: none"> Muffins(C) FairTrade(C)
---	--

14

Turn 5

D1 positions himself as Fair Trade supporter.

D1 Frame	D2 Frame	D3 Frame	Comments
Activity: <ul style="list-style-type: none"> D1 Mopping(A) <ul style="list-style-type: none"> breaking leaves(A) D1 taking break(P) D2 taking break(P) Location: <ul style="list-style-type: none"> Inside(A) <ul style="list-style-type: none"> Outside Office(P) Characters <ul style="list-style-type: none"> D1 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer(P) D3 worker(P) Relation <ul style="list-style-type: none"> Co-workers(P) Status: Equal(P) Props: <ul style="list-style-type: none"> Rake or mop? (A) Muffins(C) <ul style="list-style-type: none"> Colombian(P) Organic(P) FairTrade(C) Door(R) 	Activity: <ul style="list-style-type: none"> D1 Maintenance(R) D2 ? Location: <ul style="list-style-type: none"> Inside(P) Café(P) Characters: <ul style="list-style-type: none"> D1 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer(R) D2 Client(P) Relation <ul style="list-style-type: none"> Status: Equal(P) Props: <ul style="list-style-type: none"> Mop?(R) Muffins(C) <ul style="list-style-type: none"> Colombian(R) Organic(R) FairTrade(C) Door(P) 	Activity: <ul style="list-style-type: none"> D1 sweeping(R) Location: <ul style="list-style-type: none"> Inside(R) CoffeeShop(R) Characters: <ul style="list-style-type: none"> D1 worker(R) <ul style="list-style-type: none"> Fair Trade Buyer(R) D2 Client(R) Relation <ul style="list-style-type: none"> Status: Equal(R) Props: <ul style="list-style-type: none"> Broom (R) Muffins(C) <ul style="list-style-type: none"> Colombian(P) Organic(P) FairTrade(C) Door(R) 	Tilt Build Up D1 is positioning himself in a place where he knows someone might catch him in a wrong situation. His Offering tilt ground according to the previous established platform.

Turn 6

D2 Acknowledges D1 position on Fair Trade and confirms it. Proposing himself also as a Fair Trade buyer.

D1 Frame	D2 Frame	D3 Frame	Comments
Activity: <ul style="list-style-type: none"> D1 Mopping(A) <ul style="list-style-type: none"> breaking leaves(A) 	Activity: <ul style="list-style-type: none"> D1 Maintenance(R) D2 ? 	Activity: <ul style="list-style-type: none"> D1 sweeping(R) Location:	D1 is positioning himself in a place where he knows someone might catch him

15

<ul style="list-style-type: none"> D1 taking break(P) D2 taking break(P) Location: <ul style="list-style-type: none"> Inside(A) <ul style="list-style-type: none"> Outside Office(P) Characters <ul style="list-style-type: none"> D1 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer(C) D2 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer (R) Relation <ul style="list-style-type: none"> Co-workers(P) Status: Equal(P) Props: <ul style="list-style-type: none"> Rake or mop? (A) Muffins(C) <ul style="list-style-type: none"> Colombian(P) Organic(P) FairTrade(C) Door(R) 	Location: <ul style="list-style-type: none"> Inside(P) Café(P) Characters: <ul style="list-style-type: none"> D1 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer(C) D2 Client(P) <ul style="list-style-type: none"> Fair Trade Buyer (P) Relation <ul style="list-style-type: none"> Status: Equal(P) Props: <ul style="list-style-type: none"> Mop?(R) Muffins(C) <ul style="list-style-type: none"> Colombian(R) Organic(R) FairTrade(C) Door(P) 	<ul style="list-style-type: none"> Inside(R) CoffeeShop(R) Characters: <ul style="list-style-type: none"> D1 worker(R) <ul style="list-style-type: none"> Fair Trade Buyer(C) D2 Client(R) <ul style="list-style-type: none"> Fair Trade Buyer (R) Relation <ul style="list-style-type: none"> Status: Equal(R) Props: <ul style="list-style-type: none"> Broom (R) Muffins(C) <ul style="list-style-type: none"> Colombian(P) Organic(P) FairTrade(C) Door(R) 	in a wrong situation. His Offering tilt ground according to the previous established platform.
---	--	---	--

Consistency

At this moment if we only consider the consistent information between the players we get :

- Activity
 - ?
- Location:
 - Inside(A)
- Characters
 - D1 worker(P)
 - Fair Trade Buyer(C)
 - D2 ?
 - Fair Trade Buyer(P)
- D3 ?
 - Relation
 - Status: Equal(P)
- Props:
 - Muffins(C)
 - Colombian(P)
 - Organic(P)
 - FairTrade(C)
 - Door(R)

16

Emergent Frame

At this moment if we consider Sawyer's frame definition, where only accepted offers can represent the emergent frame we get the following emergent frame representation:

Activity ?	• D3 ?
Location:	Relation • Status: Equal(P)
• Inside(A)	Props:
Characters	• Muffins(C)
• D1 worker(C)	• FairTrade(C)
• Fair Trade Buyer(C)	• Door (R)
• D2 ?	

Turn 7

"The Colombian, Organic, Free Trade Muffins seemed like a ridiculous idea to me, it seemed to me that he was really pushing the muffins making a hard sell on the muffins, didn't know if D2 was really buying it or not"

Why would D3 pick this setup? Why is this really a setup?

Because the Fair Trade Muffins was the most confirmed offer at stake. So the question isn't why D3 picked the muffins to go on, but why did they get so important? What did D2 and D1 saw on Fair Trade that allowed them to prepare a setup?

7	D3	<enters from right stage> <faces D1 with servant posture, holding some sort of tray, and spanish accent> Mr. CoffeeMan made up some more muffins for you from homeland.	D3 Character Free Trade Worker Colombian D1 Character Free Trade Boss Status D3 < D1 D3 offers D1 to be a coffee owner, but he doesn't understand "CoffeeMan" but Cufman.... (GroupInterv. 11:33)
---	----	---	--

D1[Group Interview]: I've never thought coffer shop at to some point.... I thought office (For the entire scene?) yeah!

17

D1 Frame	D2 Frame	D3 Frame	Comments
Activity: <ul style="list-style-type: none"> D1 Mopping(A) <ul style="list-style-type: none"> breaking leaves(A) D1 taking break(P) D2 taking break(P) D3 offering muffins to D1 (P) Location: <ul style="list-style-type: none"> Inside(A) <ul style="list-style-type: none"> Outside Office(P) Characters <ul style="list-style-type: none"> D1 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer(C) D2 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer (R) D3 producer(R) <ul style="list-style-type: none"> Fair Trade Producer(R) Colombian(R) Relation <ul style="list-style-type: none"> Co-workers(P) StatusD1D2: Equal(P) D1 D3(R): <ul style="list-style-type: none"> Boss(D3, D1)(R) D3.Status <<< D1.Status(R) Props: <ul style="list-style-type: none"> Rake or mop? (A) Muffins(C) <ul style="list-style-type: none"> Colombian(C) Organic(P) FairTrade(C) Door(R) Muffin Tray(R) 	Activity: <ul style="list-style-type: none"> D1 Maintenance(R) D2 ? D3 offering muffins to D1 (P) Location: <ul style="list-style-type: none"> Inside(P) <ul style="list-style-type: none"> Café(C) Characters: <ul style="list-style-type: none"> D1 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer(C) D2 Client(P) <ul style="list-style-type: none"> Fair Trade Buyer (P) D3 producer(R) <ul style="list-style-type: none"> Fair Trade Producer(R) Colombian(R) Relation <ul style="list-style-type: none"> StatusD1D2: Equal(P) D1 D3(R): <ul style="list-style-type: none"> Boss(D3, D1)(R) D3.Status <<< D1.Status(R) Props: <ul style="list-style-type: none"> Mop?(R) Muffins(C) <ul style="list-style-type: none"> Colombian(C) Organic(R) FairTrade(C) Door(P) Muffin Tray(R) 	Activity: <ul style="list-style-type: none"> D1 sweeping(R) D3 offering muffins to D1 (P) Location: <ul style="list-style-type: none"> Inside(R) CoffeeShop(C) Characters: <ul style="list-style-type: none"> D1 worker(R) <ul style="list-style-type: none"> Fair Trade Buyer(C) D2 Client(R) <ul style="list-style-type: none"> Fair Trade Buyer (R) D3 producer(P) <ul style="list-style-type: none"> Fair Trade Producer(P) Colombian(P) Relation <ul style="list-style-type: none"> StatusD1D2: Equal(R) D1 D3(P): <ul style="list-style-type: none"> Boss(D3, D1)(P) D3.Status <<< D1.Status(P) Props: <ul style="list-style-type: none"> Broom (R) Muffins(C) <ul style="list-style-type: none"> Colombian(C) Organic(P) FairTrade(C) Door(R) Muffin Tray(P) 	Tilt Build Up D3 enters briggig more information about the platform established about Fair Trade. She assumes a Fair Trade related role that with very low status towards D1, continuing to increase D1's trap and dependency on Fair Trade.

18

Turn 8

D1 accepts the activity proposed by D3 “I appreciate it”

8 D1 I appreciate it. D1 Likes muffins
 <looks around>
 <turns to D2 and whispers>
 God, she’s annoying...
 <takes a breath>
 <turns to D3>

D1 Frame	D2 Frame	D3 Frame	Comments
Activity: <ul style="list-style-type: none"> D1 Mopping(A) <ul style="list-style-type: none"> breaking leaves(A) D1 taking break(P) D2 taking break(P) D3 offering muffins to D1 (C) Location: <ul style="list-style-type: none"> Inside(A) <ul style="list-style-type: none"> Outside Office(P) Characters <ul style="list-style-type: none"> D1 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer(C) D2 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer (R) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) Colombian(R) Annoying(P) Relation <ul style="list-style-type: none"> Co-workers(P) StatusD1D2: Equal(P) D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) 	Activity: <ul style="list-style-type: none"> D1 Maintenance(R) D2 ? D3 offering muffins to D1 (C) Location: <ul style="list-style-type: none"> Inside(P) <ul style="list-style-type: none"> Café(P) Characters: <ul style="list-style-type: none"> D1 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer(C) D2 Client(P) <ul style="list-style-type: none"> Fair Trade Buyer (P) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) Colombian(P) Annoying(R) Relation <ul style="list-style-type: none"> StatusD1D2: Equal(P) D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) D3.Status <<< D1.Status(C) Props: <ul style="list-style-type: none"> Mop?(R) Muffins(C) 	Activity: <ul style="list-style-type: none"> D1 sweeping(R) D3 offering muffins to D1 (C) Location: <ul style="list-style-type: none"> Inside(R) CoffeeShop(R) Characters: <ul style="list-style-type: none"> D1 worker(R) <ul style="list-style-type: none"> Fair Trade Buyer(C) D2 Client(R) <ul style="list-style-type: none"> Fair Trade Buyer (R) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) Colombian(R) Annoying(R) Relation <ul style="list-style-type: none"> StatusD1D2: Equal(R) D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) D3.Status <<< D1.Status(C) Props: <ul style="list-style-type: none"> Broom (R) Muffins(C) <ul style="list-style-type: none"> Colombian(C) 	Tilt Build Up D1 is positioning himself in a place where he knows someone might catch him in a wrong situation. His Offering tilt ground according to the previous established platform.

19

<ul style="list-style-type: none"> D3.Status <<< D1.Status(C) Props: <ul style="list-style-type: none"> Rake or mop? (A) Muffins(C) <ul style="list-style-type: none"> Colombian(C) Organic(P) FairTrade(C) Door(R) Muffin Tray(C) 	<ul style="list-style-type: none"> Colombian(C) Organic(R) FairTrade(C) <ul style="list-style-type: none"> Door(P) Muffin Tray(C) 	<ul style="list-style-type: none"> Organic(P) FairTrade(C) <ul style="list-style-type: none"> Door(R) Muffin Tray(C) 	
---	--	--	--

Consistency

At this moment if we only consider the consistent information between the players we get :

Activity: <ul style="list-style-type: none"> D3 offering muffins to D1 (C) Location: <ul style="list-style-type: none"> Inside(A) Characters <ul style="list-style-type: none"> D1 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer(C) D2 ? <ul style="list-style-type: none"> Fair Trade Buyer(P) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) Colombian(P) Annoying(P) 	Relation <ul style="list-style-type: none"> Status: Equal(P) D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) D3.Status <<< D1.Status(C) Props: <ul style="list-style-type: none"> Muffins(C) <ul style="list-style-type: none"> Colombian(P) Organic(P) FairTrade(C) Door(R) Muffin Tray(C)
---	---

Emergent Frame

At this moment if we consider Sawyer’s frame definition, where only accepted offers can represent the emergent frame we get the following emergent frame representation:

Location <ul style="list-style-type: none"> Inside(A) 	Activity: <ul style="list-style-type: none"> D3 offering muffins to D1 (C) Characters: <ul style="list-style-type: none"> D1?
---	---

20

- Fair Trade Buyer(C)
 - D3 producer(C)
 - Fair Trade Producer(C)
- Relation:
- D1 D3(C):
 - Boss(D3, D1)(C)
- D3.Status <<< D1.Status(C)
- Props:
- Muffins(C)
 - FairTrade(C)
 - Door(R)
 - Muffin Tray(C)

Turn 9

It is at this moment that D3 presents new data to the scene that alters significantly the characters state. The fact that she's starving is related to being explored by the Fair Trade ideology which breaks D1's apparent hero position.

9 D3 Please feed me! D3 passes hunger

D1[Group interview]: "Immediately my offer from D3 that I read was we're hypocrites"

D3[Group interview]: "Yeah. Yes" <looking at D1>

D1[Group interview]: We are doing everything that is politically savvy or correct, yet in our own behaviors realistically we subjugate and we have no interest in treating people equally and fairly."

D1 Frame	D2 Frame	D3 Frame	Comment
Activity: <ul style="list-style-type: none"> • D1 Mopping(A) <ul style="list-style-type: none"> ○ breaking leaves(A) • D1 taking break(P) • D2 taking break(P) • D3 offering muffins to D1 (C) • D3 begs for food (R) Location: <ul style="list-style-type: none"> • Inside(A) <ul style="list-style-type: none"> ○ Outside • Office(P) Characters <ul style="list-style-type: none"> • D1 worker(P) <ul style="list-style-type: none"> ○ Fair Trade Buyer(C) • D2 worker(P) <ul style="list-style-type: none"> ○ Fair Trade Buyer (R) 	Activity: <ul style="list-style-type: none"> • D1 Maintenance(R) • D2 ? • D3 offering muffins to D1 (C) • D3 begs for food (R) Location: <ul style="list-style-type: none"> • Inside(P) <ul style="list-style-type: none"> • Café(P) Characters: <ul style="list-style-type: none"> • D1 worker(P) <ul style="list-style-type: none"> ○ Fair Trade Buyer(C) • D2 Client(P) <ul style="list-style-type: none"> ○ Fair Trade Buyer (P) • D3 producer(C) <ul style="list-style-type: none"> ○ Fair Trade Producer(C) ○ Colombian(R) 	Activity: <ul style="list-style-type: none"> • D1 sweeping(R) • D3 offering muffins to D1 (C) • D3 begs for food (P) Location: <ul style="list-style-type: none"> • Inside(R) • CoffeeShop(R) Characters: <ul style="list-style-type: none"> • D1 worker(R) <ul style="list-style-type: none"> ○ Fair Trade Buyer(C) • D2 Client(R) <ul style="list-style-type: none"> ○ Fair Trade Buyer (R) • D3 producer(C) <ul style="list-style-type: none"> ○ Fair Trade Producer(C) ○ Colombian(P) ○ Annoying(R) 	Tilt Acceptance D3 tilts the previous established platform by changing contradicting the Fair Trade frame to which D1 and D2 got intensively involved. What's the role of D3

21

<ul style="list-style-type: none"> • D3 producer(C) <ul style="list-style-type: none"> ○ Fair Trade Producer(C) ○ Colombian(R) ○ Annoying(P) ○ Starving(R) ○ Poor(R) Relation <ul style="list-style-type: none"> • Co-workers(P) • StatusD1D2: Equal(P) • D1 D3(C): <ul style="list-style-type: none"> ○ Boss(D3, D1)(C) ○ D3.Status <<< D1.Status(C) Props: <ul style="list-style-type: none"> • Rake or mop? (A) • Muffins(C) <ul style="list-style-type: none"> ○ Colombian(C) ○ Organic(P) ○ FairTrade(C) • Door(R) • Muffin Tray(C) 	<ul style="list-style-type: none"> ○ Annoying(R) ○ Starving(P) ○ Poor(P) Relation <ul style="list-style-type: none"> • StatusD1D2: Equal(P) • D1 D3(C): <ul style="list-style-type: none"> ○ Boss(D3, D1)(C) ○ D3.Status <<< D1.Status(C) Props: <ul style="list-style-type: none"> • Mop?(R) • Muffins(C) <ul style="list-style-type: none"> ○ Colombian(C) ○ Organic(R) ○ FairTrade(C) • Door(P) • Muffin Tray(C) 	<ul style="list-style-type: none"> ○ Starving(R) ○ Poor(R) Relation <ul style="list-style-type: none"> • StatusD1D2: Equal(R) • D1 D3(C): <ul style="list-style-type: none"> ○ Boss(D3, D1)(C) ○ D3.Status <<< D1.Status(C) Props: <ul style="list-style-type: none"> • Broom (R) • Muffins(C) <ul style="list-style-type: none"> ○ Colombian(C) ○ Organic(P) ○ FairTrade(C) • Door(R) • Muffin Tray(C) 	being poor or starving?
--	--	---	----------------------------

Consistency

At this moment if we only consider the consistent information between the players we get :

- Location
- Inside(A)
- Activity:
- D3 offering muffins to D1 (C)
 - D3 begs for food (P)
- Characters:
- D1?
 - Fair Trade Buyer(C)
 - D3 producer(C)
- Relation:
- StatusD1D2: Equal(R)
 - D1 D3(C):
 - Boss(D3, D1)(C)
 - D3.Status <<< D1.Status(C)
- Props:
- Fair Trade Producer(C)
 - Colombian(R)
 - Annoying(R)
 - Starving(P)
 - Poor(P)

22

- Muffins(C)
 - FairTrade(C)
- Door(R)

Emergent Frame

At this moment if we consider Sawyer's frame definition, where only accepted offers can represent the emergent frame we get the following emergent frame representation:

Location

- Inside(A)

Activity:

- D3 offering muffins to D1 (C)

Characters:

- D1?
 - Fair Trade Buyer(C)
- D3 producer(C)
 - Fair Trade Producer(C)

Relation:

[Turn 10 to 15](#)

- Muffin Tray(C)

- StatusD1D2: Equal(R)

- D1 D3(C):

- Boss(D3, D1)(C)
- D3.Status <<< D1.Status(C)

Props:

- Muffins(C)
 - FairTrade(C)
- Door(R)

Muffin Tray(C)

D1 "I think an I have a tendency to do this has an improviser. Sometimes it's amazing, sometimes it's confusing. I kind of doubled up on an offer, where she's made an offer that we're kind of hypocritical, so I'm like ok I'll go with that I miss treat her and talk condescendingly to her... although on the surface I seem to care about political issues and the welfare of everyone at large, and so even worse than that is to be completely condescending and at the same time a complete ignorant..."

10 D1 <surprised>
Huu! Well, how can't the muffin girl feed herself?

11 D1 <confused>
You know lead a fish to water...

13 D1 <more confused>
I mean...
If I make....
If I show you how to fish...

14 D3 Yes...

23

15 D1 <still confused, looking around>
Anyway... I don't know
<leaves to back right stage, to store his tools>

D1 Frame	D3 Frame	D3 Frame	Model State
D1 Frame Activity: <ul style="list-style-type: none"> • D1 Mopping(A) <ul style="list-style-type: none"> ◦ breaking leaves(A) • D1 taking break(P) • D2 taking break(P) • D3 offering muffins to D1 (C) • D3 begs for food (C) Location: <ul style="list-style-type: none"> • Inside(A) <ul style="list-style-type: none"> ◦ Outside • Office(P) Characters <ul style="list-style-type: none"> • D1 worker(P) <ul style="list-style-type: none"> ◦ Fair Trade Buyer(C) ◦ Ignorant(P) ◦ Arrogant(P) • D2 worker(P) <ul style="list-style-type: none"> ◦ Fair Trade Buyer (C) • D3 producer(C) <ul style="list-style-type: none"> ◦ Fair Trade Producer(C) ◦ Colombian(R) ◦ Annoying(P) ◦ Starving(C) ◦ Poor(C) Relation <ul style="list-style-type: none"> • Co-workers(P) • StatusD1D2: Equal(P) • D1 D3(C): <ul style="list-style-type: none"> ◦ Boss(D3, D1)(C) 	D2 Frame Activity: <ul style="list-style-type: none"> • D1 Maintenance(R) • D2 ? • D3 offering muffins to D1 (C) • D3 begs for food (C) Location: <ul style="list-style-type: none"> • Inside(P) <ul style="list-style-type: none"> • Café(P) Characters: <ul style="list-style-type: none"> • D1 worker(P) <ul style="list-style-type: none"> ◦ Fair Trade Buyer(C) ◦ Ignorant(R) ◦ Arrogant(R) • D2 Client(P) <ul style="list-style-type: none"> ◦ Fair Trade Buyer (C) • D3 producer(C) <ul style="list-style-type: none"> ◦ Fair Trade Producer(C) ◦ Colombian(R) ◦ Annoying(R) ◦ Starving(C) ◦ Poor(C) Relation <ul style="list-style-type: none"> • StatusD1D2: Equal(P) • D1 D3(C): <ul style="list-style-type: none"> ◦ Boss(D3, D1)(C) ◦ D3.Status <<< D1.Status(C) Props: <ul style="list-style-type: none"> • Mop?(R) • Muffins(C) 	D3 Frame Activity: <ul style="list-style-type: none"> • D1 sweeping(R) • D3 offering muffins to D1 (C) • D3 begs for food (C) Location: <ul style="list-style-type: none"> • Inside(R) • CoffeeShop(R) Characters: <ul style="list-style-type: none"> • D1 worker(R) <ul style="list-style-type: none"> ◦ Fair Trade Buyer(C) ◦ Ignorant(R) ◦ Arrogant(R) • D2 Client(R) <ul style="list-style-type: none"> ◦ Fair Trade Buyer (C) • D3 producer(C) <ul style="list-style-type: none"> ◦ Fair Trade Producer(C) ◦ Colombian(P) ◦ Annoying(R) ◦ Starving(C) ◦ Poor(C) Relation <ul style="list-style-type: none"> • StatusD1D2: Equal(R) • D1 D3(C): <ul style="list-style-type: none"> ◦ Boss(D3, D1)(C) ◦ D3.Status <<< D1.Status(C) Props: <ul style="list-style-type: none"> • Broom (R) • Muffins(C) <ul style="list-style-type: none"> ◦ Colombian(C) 	Tilt Riding D1 is offering what he thinks that the new platform makes of his characters, by acting as an hypocrite, ignorant and condescending man. He is also confirming D3's offers about she being poor, low status and starving. This might also have an impact on the D1's relations although there's no info about it.

<ul style="list-style-type: none"> o D3.Status <<< D1.Status(C) Props: <ul style="list-style-type: none"> • Rake or mop? (A) • Muffins(C) <ul style="list-style-type: none"> o Colombian(C) o Organic(P) o FairTrade(C) • Door(R) • Muffin Tray(C) 	<ul style="list-style-type: none"> o Colombian(C) o Organic(R) o FairTrade(C) <ul style="list-style-type: none"> • Door(P) • Muffin Tray(C) 	<ul style="list-style-type: none"> o Organic(P) o FairTrade(C) <ul style="list-style-type: none"> • Door(R) • Muffin Tray(C) 	
---	--	--	--

Consistency

At this moment if we only consider the consistent information between the players we get

Activity:

- D3 offering muffins to D1 (C)
- D3 begs for food (C)

Location:

- Inside(A)

Characters

- D1 worker(P)
 - o Fair Trade Buyer(C)
 - o Ignorant(P)
 - o Arrogant(P)
- D2
 - o Fair Trade Buyer (C)
- D3 producer(C)
 - o Fair Trade Producer(C)
 - o Colombian(R)

- o Annoying(P)
- o Starving(C)
- o Poor(C)

Relation

- StatusD1D2: Equal(P)
- D1 D3(C):
 - o Boss(D3, D1)(C)
 - o D3.Status <<< D1.Status(C)

Props:

- Muffins(C)
 - o Colombian(C)
 - o Organic(P)
 - o FairTrade(C)
- Door(R)

Muffin Tray(C)

Emergent Frame

At this moment if we consider Sawyer's frame definition, where only accepted offers can represent the emergent frame we get the following emergent frame representation:

25

Activity:

- D3 offering muffins to D1 (C)
- D3 begs for food (C)

Location:

- Inside(A)

Characters

- D1
 - o Fair Trade Buyer(C)
- D2

- o Fair Trade Buyer (C)
- D3 producer(C)
 - o Fair Trade Producer(C)

Relation

- StatusD1D2: Equal(P)
- D1 D3(C):
 - o Boss(D3, D1)(C)
 - o D3.Status <<< D1.Status(C)

Props:

- Muffins(C)
 - o Colombian(C)
 - o FairTrade(C)
- Door(C)

Muffin Tray(C)

Turn 16 to 22

16	D3	<looking at D1 and D2> I need for the food alone sir...	D3 - Reinforcing she needs food D3 – Shows low status towards D2
17	D1	<takes 2 muffins from the tray in D3 hands> <offers a muffin to D2> Here. Have a muffin man. They're delicious.	New activity, D1 and D2 eating muffins New property D1 finds muffins delicious
18	D2	<concerned. holding a muffin> Thanks. <Tries to formulate something> I....	D2 confirms eating activity
19	D3	<begging D2> Please eat my muffins!	D3 reinforces need to sell muffins
20	D1	<in high status holding muffin> <facing D2, advising> Don't let ammm, Eurica come to you...	D1 reinforces his status, D1 reinforces his perception of Eurica being annoying D1 offers name to D3
21	D2	<to D3>	D3 confirms D2 name as Eurica

26

D1 Frame	D2 Frame	D3 Frame	Comments
<p>Activity:</p> <ul style="list-style-type: none"> D1 Mopping(A) <ul style="list-style-type: none"> breaking leaves(A) D1 taking break(P) D2 taking break(P) D3 offering muffins to D1 (C) D3 begs for food (C) D1 eating(Muffins)(C) D2 eating(Muffins)(C) <p>Location:</p> <ul style="list-style-type: none"> Inside(A) <ul style="list-style-type: none"> Outside Office(P) <p>Characters</p> <ul style="list-style-type: none"> D1 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer(C) Ignorant(P) Arrogant(P) D2 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer (C) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) Colombian(R) Annoying(P) Starving(C) Poor(C) Name = Eurica (C) <p>Relation</p> <ul style="list-style-type: none"> Co-workers(P) StatusD1D2: Equal(P) D1 D3(C): 	<p>Activity:</p> <ul style="list-style-type: none"> D1 Maintenance(R) D2 ? D3 offering muffins to D1 (C) D3 begs for food (C) D1 eating(Muffins)(C) D2 eating(Muffins)(C) <p>Location:</p> <ul style="list-style-type: none"> Inside(P) <ul style="list-style-type: none"> Café(P) <p>Characters:</p> <ul style="list-style-type: none"> D1 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer(C) Ignorant(R) Arrogant(R) D2 Client(P) <ul style="list-style-type: none"> Fair Trade Buyer (C) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) Colombian(R) Annoying(R) Starving(C) Poor(C) Name = Eurica (C) <p>Relation</p> <ul style="list-style-type: none"> StatusD1D2: Equal(P) D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) D3.Status <<< D1.Status(C) <p>Props:</p>	<p>Activity:</p> <ul style="list-style-type: none"> D1 sweeping(R) D3 offering muffins to D1 (C) D3 begs for food (C) D1 eating(Muffins)(C) D2 eating(Muffins)(C) <p>Location:</p> <ul style="list-style-type: none"> Inside(R) <ul style="list-style-type: none"> CoffeeShop(R) <p>Characters:</p> <ul style="list-style-type: none"> D1 worker(R) <ul style="list-style-type: none"> Fair Trade Buyer(C) Ignorant(R) Arrogant(R) D2 Client(R) <ul style="list-style-type: none"> Fair Trade Buyer (C) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) Colombian(P) Annoying(R) Starving(C) Poor(C) Name = Eurica (C) <p>Relation</p> <ul style="list-style-type: none"> StatusD1D2: Equal(R) D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) D3.Status <<< D1.Status(C) <p>Props:</p> <ul style="list-style-type: none"> Broom (R) Muffins(C) 	<p>Tilt Riding D1 and D3 seem to have plenty on character info to display in the current context. The action is an extension of the previous status quo.</p>

27

<ul style="list-style-type: none"> Boss(D3, D1)(C) D3.Status <<< D1.Status(C) <p>Props:</p> <ul style="list-style-type: none"> Rake or mop? (A) Muffins(C) <ul style="list-style-type: none"> Colombian(C) Organic(P) FairTrade(C) Delicious(P) Door(R) Muffin Tray(C) 	<ul style="list-style-type: none"> Mop?(R) Muffins(C) <ul style="list-style-type: none"> Colombian(C) Organic(R) FairTrade(C) Delicious(C) Door(P) Muffin Tray(C) 	<ul style="list-style-type: none"> Colombian(C) Organic(P) FairTrade(C) Delicious(C) <ul style="list-style-type: none"> Door(R) Muffin Tray(C) 	
--	--	--	--

Consistency

At this moment if we only consider the consistent information between the players we get

<p>Activity:</p> <ul style="list-style-type: none"> D3 offering muffins to D1 (C) D3 begs for food (C) D1 eating(Muffins)(C) D2 eating(Muffins)(C) <p>Location:</p> <ul style="list-style-type: none"> Inside(A) <p>Characters</p> <ul style="list-style-type: none"> D1 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer(C) Ignorant(P) Arrogant(P) D2 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer (C) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) Colombian(R) Annoying(P) 	<ul style="list-style-type: none"> Starving(C) Poor(C) Name = Eurica (C) <p>Relation</p> <ul style="list-style-type: none"> StatusD1D2: Equal(P) D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) D3.Status <<< D1.Status(C) <p>Props:</p> <ul style="list-style-type: none"> Rake or mop? (A) Muffins(C) <ul style="list-style-type: none"> Colombian(C) Organic(P) FairTrade(C) Delicious(P) Door(R) Muffin Tray(C)
--	---

28

Emergent Frame

At this moment if we consider Sawyer's frame definition, where only accepted offers can represent the emergent frame we get the following emergent frame representation:

Activity:

- D3 offering muffins to D1 (C)
- D3 begs for food (C)
- D1 eating(Muffins)(C)
- D2 eating(Muffins)(C)

Location:

- Inside(A)

Characters

- D1 worker(P)
 - Fair Trade Buyer(C)
- D2 worker(P)
 - Fair Trade Buyer (C)
- D3 producer(C)
 - Fair Trade Producer(C)
 - Starving(C)

- Poor(C)
- Name = Eurica (C)

Relation

- StatusD1D2: Equal(P)
- D1 D3(C):
 - Boss(D3, D1)(C)
 - D3.Status <<< D1.Status(C)

Props:

- Muffins(C)
 - Colombian(C)
 - Organic(P)
 - FairTrade(C)
- Door(R)

Muffin Tray(C)

Turn 23 to 28

During these turns D2 and D3 just move according to their characters in order to start a conversation. A head to head. D3 uses this moment to reinforce the status in her relation with D1 by asking him permission to talk with D

D2 Individual Interview: "Checking with D1 and making sure he is my boss... checking with him gives him the higher status and gives him again like **villain points...** or makes me **more sympathetic...**"

23 D2 <to D3>
Eurica, can I have a word with you?

24 D3 <submissive to D1, asking permission>
<walks to center front stage next do D2>

25 D1 Sure I'll leave you two alone
<steps out to right stage, pretends to eat his muffin>

29

26 D2 Hey thanks a lot Scott.

27 D1 Any time.

28 D3 <facing D2>
Hey!

D1 Frame	D2 Frame	D3 Frame	Comments / Action
Activity: <ul style="list-style-type: none"> • D2 and D3 talking in private(C) (..) 	Activity: <ul style="list-style-type: none"> • D2 and D3 talking in private(C) 	Activity: <ul style="list-style-type: none"> • D2 and D3 talking in private(C) 	Tilt Riding D1 and D3 seem to have plenty on character info to display in the current context. The action is an extension of the previous status quo.

Consistency

At this moment if we only consider the consistent information between the players we get a new activity: D3 and D2 talk in private.

Activity:

- D3 offering muffins to D1 (C)
- D3 begs for food (C)
- D1 eating(Muffins)(C)
- D2 eating(Muffins)(C)
- D2 and D3 talking in private(C)

Location:

- Inside(A)

Characters

- D1 worker(P)
 - Fair Trade Buyer(C)
 - Ignorant(P)
 - Arrogant(P)

- D2 worker(P)
 - Fair Trade Buyer (C)
- D3 producer(C)
 - Fair Trade Producer(C)
 - Colombian(R)
 - Annoying(P)
 - Starving(C)
 - Poor(C)
 - Name = Eurica (C)

Relation

- StatusD1D2: Equal(P)
- D1 D3(C):
 - Boss(D3, D1)(C)

30

- D3.Status <<< D1.Status(C)
- Props:
- Rake or mop? (A)

- Muffins(C)
 - Colombian(C)

Emergent Frame

At this moment if we consider Sawyer's frame definition, where only accepted offers can represent the emergent frame we get the following emergent frame representation:

Activity:

- D3 offering muffins to D1 (C)
- D3 begs for food (C)
- D1 eating(Muffins)(C)
- D2 eating(Muffins)(C)
- D2 and D3 talking in private(C)

Location:

- Inside(A)

Characters

- D1 worker(P)
 - Fair Trade Buyer(C)
- D2 worker(P)
 - Fair Trade Buyer (C)
- D3 producer(C)
 - Fair Trade Producer(C)
-

- Starving(C)
- Poor(C)
- Name = Eurica (C)

Relation

- StatusD1D2: Equal(P)
- D1 D3(C):
 - Boss(D3, D1)(C)
 - D3.Status <<< D1.Status(C)

Props:

- Muffins(C)
 - Colombian(C)
 - Organic(P)
 - FairTrade(C)
- Door(R)
- Muffin Tray(C)

Turn 29 to 31

D3 individual interview: "getting him to help me. Giving myself enough negative things so he has no choice but to help me, and no choice but for as to gain on the boss"

D1 Individual Interview: "**Offer her an opportunity to leave this place ...** because the way she presented the character it seemed really vulnerable and seems she was being taken advantage of"

D1 Group Interview: At this time she already presented herself as a...." D3 "...low status..." D1 "a protagonist.. lets just start with something and get you out of here"

31

29	D2	<to D3> Hi, do you like Scott?	D2 tries to establish relation between D3 and D1
30	D3	I do. I do. Do you like my muffins?	D2 establishes her relation with D1 D3 tries to establish relation between D2 and muffins
31	D2	I do. I love your muffins.	D2 establishes his relation with muffins.

Consistency

At this moment if we only consider the consistent information between the players we get :

Activity:

- D3 offering muffins to D1 (C)
- D3 begs for food (C)
- D1 eating(Muffins)(C)
- D2 eating(Muffins)(C)
- D2 and D3 talking in private(C)

Location:

- Inside(A)

Characters

- D1 worker(P)
 - Fair Trade Buyer(C)
 - Ignorant(P)
 - Arrogant(P)
- D2 worker(P)
 - Fair Trade Buyer (C)
 - Likes muffins(C)
- D3 producer(C)

- Fair Trade Producer(C)
- Colombian(R)
- Annoying(P)
- Starving(C)
- Poor(C)
- Name = Eurica (C)
- Likes D1(C)

Relation

- StatusD1D2: Equal(P)
- D1 D3(C):
 - Boss(D3, D1)(C)
 - D3.Status <<< D1.Status(C)

Props:

- Rake or mop? (A)
- Muffins(C)
 - Colombian(C)

Emergent Frame

At this moment if we consider Sawyer's frame definition, where only accepted offers can represent the emergent frame we get the following emergent frame representation:

32

Activity:

- D3 offering muffins to D1 (C)
- D3 begs for food (C)
- D1 eating(Muffins)(C)
- D2 eating(Muffins)(C)
- D2 and D3 talking in private(C)

Location:

- Inside(A)

Characters

- D1 worker(P)
 - Fair Trade Buyer(C)
- D2 worker(P)
 - Fair Trade Buyer (C)
 - Likes muffins(C)
- D3 producer(C)
 - Fair Trade Producer(C)

- Colombian(R)
- Annoying(P)
- Starving(C)
- Poor(C)
- Name = Eurica (C)
- Likes D1(C)

Relation

- StatusD1D2: Equal(P)
- D1 D3(C):
 - Boss(D3, D1)(C)
 - D3.Status <<< D1.Status(C)

Props:

- Rake or mop? (A)
- Muffins(C)
 - Colombian(C)

Turn 32 to 40

D3 "I am looking at him (D2) to get me out of this position, my extreme low status position"

Interviewer: Were you both at the coffee shop at this moment? D1 "I don't think that location had any bearing at this point. It's really about the..."

D3 "...people and how they are interacting as opposed to the environment"

32	D3	Every time you eat my muffins, one of my children doesn't die.	D3 Reinforces the importance of her poverty with the chance of saving her children D3 adds new rule to D2: eat to protect her children.
33	D2	<start eating muffin very fast> <takes more muffins and puts them in his	D3 follows the good guy role doing everything he can to save

33

		mouth>	children
34	D3	Please, please! Keep eating my muffins!	D3 reinforces again the importance of saving her children
35	D2	<with his mouth full of muffins> How many kids do you have?	
36	D3	I have at least 15... Please, keep eating my muffins	
37	D2	<still with mouth full of muffins> <acknowledges with his head> I'll just go and get this whole tray from your hands. <takes tray from her hands>	
38	D3	<happily surprised> oow! Uau!	
39	D2	<breathless> And a, you know, would you like to come with me? To leave this place?	
40	D3	<look rapidly to D1 and then to D2 again> (whispering) More than anything in the world!	

D1 Frame	D2 Frame	D3 Frame	Comments
----------	----------	----------	----------

34

<p>Activity:</p> <ul style="list-style-type: none"> D1 Mopping(A) <ul style="list-style-type: none"> breaking leaves(A) D1 taking break(P) D2 taking break(P) D3 offering muffins to D1 (C) D3 begs for food (C) D1 eating(Muffins)(C) D2 eating(Muffins)(C) D2 eat(muffins) for saving D3 children (C) <p>Location:</p> <ul style="list-style-type: none"> Inside(A) <ul style="list-style-type: none"> Outside Office(P) <p>Characters</p> <ul style="list-style-type: none"> D1 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer(C) Ignorant(P) Arrogant(P) D2 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer (C) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) Colombian(R) Annoying(P) Starving(C) Poor(C) Name = Eurica (C) Mother <ul style="list-style-type: none"> <15 kids (C) wants escape(C) <p>Relation</p> <ul style="list-style-type: none"> Co-workers(P) StatusD1D2: Equal(P) 	<p>Activity:</p> <ul style="list-style-type: none"> D1 Maintenance(R) D2 ? D3 offering muffins to D1 (C) D3 begs for food (C) D1 eating(Muffins)(C) D2 eating(Muffins)(C) D2 eat(muffins) for saving D3 children (C) <p>Location:</p> <ul style="list-style-type: none"> Inside(P) <ul style="list-style-type: none"> Café(P) <p>Characters:</p> <ul style="list-style-type: none"> D1 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer(C) Ignorant(R) Arrogant(R) D2 Client(P) <ul style="list-style-type: none"> Fair Trade Buyer (C) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) Colombian(R) Annoying(R) Starving(C) Poor(C) Name = Eurica (C) Mother <ul style="list-style-type: none"> <15 kids (C) wants escape(C) <p>Relation</p> <ul style="list-style-type: none"> StatusD1D2: Equal(P) D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) D3.Status <<< D1.Status(C) 	<p>Activity:</p> <ul style="list-style-type: none"> D1 sweeping(R) D3 offering muffins to D1 (C) D3 begs for food (C) D1 eating(Muffins)(C) D2 eating(Muffins)(C) D2 eat(muffins) for saving D3 children (P) <p>Location:</p> <ul style="list-style-type: none"> Inside(R) <ul style="list-style-type: none"> CoffeeShop(R) <p>Characters:</p> <ul style="list-style-type: none"> D1 worker(R) <ul style="list-style-type: none"> Fair Trade Buyer(C) Ignorant(R) Arrogant(R) D2 Client(R) <ul style="list-style-type: none"> Fair Trade Buyer (C) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) Colombian(P) Annoying(R) Starving(C) Poor(C) Name = Eurica (C) Mother <ul style="list-style-type: none"> <15 kids (C) wants escape(C) <p>Relation</p> <ul style="list-style-type: none"> StatusD1D2: Equal(R) D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) D3.Status <<< D1.Status(C) 	<p>The action continues to extend by exploring D3's low status</p>
--	--	---	--

35

<ul style="list-style-type: none"> D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) D3.Status <<< D1.Status(C) <p>Props:</p> <ul style="list-style-type: none"> Rake or mop? (A) Muffins(C) <ul style="list-style-type: none"> Colombian(C) Organic(P) FairTrade(C) Delicious(P) Door(R) Muffin Tray(C) 	<p>Props:</p> <ul style="list-style-type: none"> Mop?(R) Muffins(C) <ul style="list-style-type: none"> Colombian(C) Organic(R) FairTrade(C) Delicious(C) Door(P) Muffin Tray(C) 	<p>Props:</p> <ul style="list-style-type: none"> Broom (R) Muffins(C) <ul style="list-style-type: none"> Colombian(C) Organic(P) FairTrade(C) Delicious(C) Door(R) Muffin Tray(C) 	
---	--	--	--

Consistency

At this moment if we only consider the consistent information between the players we get

<p>Activity:</p> <ul style="list-style-type: none"> D3 offering muffins to D1 (C) D3 begs for food (C) D1 eating(Muffins)(C) D2 eating(Muffins)(C) D2 eat(muffins) for saving D3 children (P) <p>Location:</p> <ul style="list-style-type: none"> Inside(R) <p>Characters:</p> <ul style="list-style-type: none"> D1 worker(R) <ul style="list-style-type: none"> Fair Trade Buyer(C) Ignorant(R) Arrogant(R) Fair Trade Buyer (C) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) 	<ul style="list-style-type: none"> Colombian(P) Annoying(R) Starving(C) Poor(C) Name = Eurica (C) Mother <ul style="list-style-type: none"> <15 kids (C) wants escape(C) <p>Relation</p> <ul style="list-style-type: none"> StatusD1D2: Equal(R) D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) D3.Status <<< D1.Status(C) <p>Props:</p> <ul style="list-style-type: none"> Muffins(C) <ul style="list-style-type: none"> Colombian(C)
--	---

36

- Organic(P)
- FairTrade(C)
- Delicious(C)

- Door(R)
- Muffin Tray(C)

Emergent Frame

At this moment if we consider Sawyer's frame definition, where only accepted offers can represent the emergent frame we get the following emergent frame representation:

Activity:

- D3 offering muffins to D1 (C)
- D3 begs for food (C)
- D1 eating(Muffins)(C)
- D2 eating(Muffins)(C)
- D2 eat(muffins) for saving D3 children (C)

Location:

- Inside(R)

Characters:

- D1 worker(R)
 - Fair Trade Buyer(C)
- D3 producer(C)
 - Fair Trade Producer(C)
 - Colombian(C)
 - Starving(C)
 - Poor(C)

- Name = Eurica (C)
- Mother
 - <15 kids (C)
- wants escape(C)

Relation

- StatusD1D2: Equal(C)
- D1 D3(C):
 - Boss(D3, D1)(C)
 - D3.Status <<< D1.Status(C)

Props:

- Muffins(C)
 - Colombian(C)
 - Organic(C)
 - FairTrade(C)
 - Delicious(C)

Muffin Tray(C)

Turn 41 to 48

D1 "Why don't we pull the game of pulling people aside to reveal things"

37

41	D1	<staring at D3 and D2> Scott, can I have a word with you for a second?	D1 offers D2.name = Scott D1 proposes to continue the game of pulling people aside D1 proposes conversation with D2
42	D2	<to D1> <condescendent> Scott, if I could just have a word with you for a second?	D2 confirms D2.name = Scott D2 proposes D1.name = Scott D2 follows the game of pulling people aside
43	D1	<more condescendent> SCOTT ANDERSEN, IF I COULD HAVE A WORD WITH YOU FOR A SECOND?	Tension grows but it is not clear a stage in status. D1 strongly proposes D2.name = Scott Andersen
44	D2	<more condescendent> SCOTT PHELPS, <places tray on muffins table front stage> <moves right stage to D1> YOU'RE ABOUT TO HAVE A WORD WITH ME!	D2 strongly proposes D1.name = Scott Phelps D2 accepts conversation with D1

D1 Frame	D2 Frame	D3 Frame	Comments
Activity: <ul style="list-style-type: none"> • D1 Mopping(A) <ul style="list-style-type: none"> ○ breaking leaves(A) • D1 taking break(P) • D2 taking break(P) • D3 offering muffins to D1 (C) 	Activity: <ul style="list-style-type: none"> • D1 Maintenance(R) • D2 ? • D3 offering muffins to D1 (C) • D3 begs for food (C) • D1 eating(Muffins)(C) 	Activity: <ul style="list-style-type: none"> • D1 sweeping(R) • D3 offering muffins to D1 (C) • D3 begs for food (C) • D1 eating(Muffins)(C) • D2 eating(Muffins)(C) 	Status negotiation between D1 and D2 doesn't produce any outcome, but

38

<ul style="list-style-type: none"> D3 begs for food (C) D1 eating(Muffins)(C) D2 eating(Muffins)(C) D2 eat(muffins) for saving D3 children (C) D1 talking aside to D2 (C) <p>Location:</p> <ul style="list-style-type: none"> Inside(A) <ul style="list-style-type: none"> Outside Office(P) <p>Characters</p> <ul style="list-style-type: none"> D1 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer(C) Ignorant(P) Arrogant(P) D2.name= Scott Andersen (C) D2 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer (C) D2.name= Scott Phelps (C) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) Colombian(R) Annoying(P) Starving(C) Poor(C) Name = Eurica (C) Mother <ul style="list-style-type: none"> <15 kids (C) wants escape(C) <p>Relation</p> <ul style="list-style-type: none"> Co-workers(P) StatusD1D2: Equal(P) D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) 	<ul style="list-style-type: none"> D2 eating(Muffins)(C) D2 eat(muffins) for saving D3 children (C) D1 talking aside to D2 (C) <p>Location:</p> <ul style="list-style-type: none"> Inside(P) Café(P) <p>Characters:</p> <ul style="list-style-type: none"> D1 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer(C) Ignorant(R) Arrogant(R) D2.name= Scott Andersen (C) D2 Client(P) <ul style="list-style-type: none"> Fair Trade Buyer (C) D2.name= Scott Phelps (C) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) Colombian(R) Annoying(R) Starving(C) Poor(C) Name = Eurica (C) Mother <ul style="list-style-type: none"> <15 kids (C) wants escape(C) <p>Relation</p> <ul style="list-style-type: none"> StatusD1D2: Equal(P) D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) D3.Status <<< D1.Status(C) <p>Props:</p>	<ul style="list-style-type: none"> D2 eat(muffins) for saving D3 children (P) D1 talking aside to D2 (C) <p>Location:</p> <ul style="list-style-type: none"> Inside(R) CoffeeShop(R) <p>Characters:</p> <ul style="list-style-type: none"> D1 worker(R) <ul style="list-style-type: none"> Fair Trade Buyer(C) Ignorant(R) Arrogant(R) D2.name= Scott Andersen (C) D2 Client(R) <ul style="list-style-type: none"> Fair Trade Buyer (C) D2.name= Scott Phelps (C) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) Colombian(P) Annoying(R) Starving(C) Poor(C) Name = Eurica (C) Mother <ul style="list-style-type: none"> <15 kids (C) wants escape(C) <p>Relation</p> <ul style="list-style-type: none"> StatusD1D2: Equal(R) D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) D3.Status <<< D1.Status(C) <p>Props:</p> <ul style="list-style-type: none"> Broom (R) 	both follow the strategy of talking aside to reveal new facts
---	--	---	---

39

<ul style="list-style-type: none"> D3.Status <<< D1.Status(C) <p>Props:</p> <ul style="list-style-type: none"> Rake or mop? (A) Muffins(C) <ul style="list-style-type: none"> Colombian(C) Organic(P) FairTrade(C) Delicious(P) Door(R) Muffin Tray(C) 	<ul style="list-style-type: none"> Mop?(R) Muffins(C) <ul style="list-style-type: none"> Colombian(C) Organic(R) FairTrade(C) Delicious(C) Door(P) Muffin Tray(C) 	<ul style="list-style-type: none"> Muffins(C) <ul style="list-style-type: none"> Colombian(C) Organic(P) FairTrade(C) Delicious(C) Door(R) Muffin Tray(C) 	
---	--	---	--

Consistency

At this moment if we only consider the consistent information between the players we get

Activity:

- D3 offering muffins to D1 (C)
- D3 begs for food (C)
- D1 eating(Muffins)(C)
- D2 eating(Muffins)(C)
- D2 eat(muffins) for saving D3 children (P)
- D1 talking aside to D2 (C)

Location:

- Inside(R)

Characters:

- D1 worker(R)
 - Fair Trade Buyer(C)
 - Ignorant(R)
 - Arrogant(R)
 - Fair Trade Buyer (C)
 - D1.name = Scott Phelps
- D2
 - D2.name= Scott Andersen (C)

- D3 producer(C)
 - Fair Trade Producer(C)
 - Colombian(P)
 - Annoying(R)
 - Starving(C)
 - Poor(C)
 - Name = Eurica (C)
 - Mother
 - <15 kids (C)
 - wants escape(C)

Relation

- StatusD1D2: Equal(R)
- D1 D3(C):
 - Boss(D3, D1)(C)
 - D3.Status <<< D1.Status(C)

Props:

- Muffins(C)

40

- Colombian(C)
- Organic(P)
- FairTrade(C)
- Delicious(C)

- Door(R)
- Muffin Tray(C)

Emergent Frame

At this moment if we consider Sawyer's frame definition, where only accepted offers can represent the emergent frame we get the following emergent frame representation:

Activity:

- D3 offering muffins to D1 (C)
- D3 begs for food (C)
- D1 eating(Muffins)(C)
- D2 eating(Muffins)(C)
- D2 eat(muffins) for saving D3 children (P)
- D1 talking aside to D2 (C)

- Starving(C)
- Poor(C)
- Name = Eurica (C)
- Mother
 - <15 kids (C)
- wants escape(C)

Location:

- Inside(R)

Characters:

- D1 worker(R)
 - Fair Trade Buyer(C)
 - Fair Trade Buyer (C)
 - D1.name = Scott Phelps
- D2
 - D2.name= Scott Andersen (C)
- D3 producer(C)
 - Fair Trade Producer(C)
 - Colombian(P)
 - Annoying(R)

Relation

- StatusD1D2: Equal(R)
- D1 D3(C):
 - Boss(D3, D1)(C)
 - D3.Status <<< D1.Status(C)

Props:

- Muffins(C)
 - Colombian(C)
 - Organic(P)
 - FairTrade(C)
 - Delicious(C)
- Door(R)

Muffin Tray(C)

Turn 41 to 48

41

D2 continues to follow his strategy of liberating D3 from oppression while D1 takes the opportunity to offer a new tilt.

D1 "I have something to offer now... I feel we established everything... we've got our platform... you (D3) are being oppressed by us... that's our platform the platform we had a minute ago, what about a status shift? If I made her someone more important, or someone that she somehow seems not to be?"

44	D3	Huu!	D3 is playing low status
		<moves scared with hands on her ears to front stage>	
45	D2	<pointing at D1>	D2 offers D1 exploits D3
		You are taking advantage of this young Colombian!	
46	D1	<Ironic>	D1 acknowledges the offer but doesn't confirm
		Am i?	
47	D2	<angry>	D2 reinforces D1 exploits D3
		Yes, she has 15 children at home that will die if these muffins don't get eaten!	
48	D1	<angry>	Tilt Offer
		You know what? This all muffin business is a counter fort. Did you ever hear about Pablo Escobar?	D1 acknowledges the offer but doesn't reject it
			D1 offers D3 does counterfeit muffins
			D1 offers Pablo Escobar

D1 Frame	D2 Frame	D3 Frame	Comments
----------	----------	----------	----------

42

<p>Activity:</p> <ul style="list-style-type: none"> D1 Mopping(A) <ul style="list-style-type: none"> breaking leaves(A) D1 taking break(P) D2 taking break(P) D3 offering muffins to D1 (C) D3 begs for food (C) D1 eating(Muffins)(C) D2 eating(Muffins)(C) D2 eat(muffins) for saving D3 children (C) D1 talking aside to D2 (C) D1 exploits D3 (R) D3 does counterfeit (P) <p>Location:</p> <ul style="list-style-type: none"> Inside(A) <ul style="list-style-type: none"> Outside Office(P) <p>Characters</p> <ul style="list-style-type: none"> D1 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer(C) Ignorant(P) Arrogant(P) D2.name= Scott Andersen (C) D2 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer (C) D2.name= Scott Phelps (C) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) Colombian(R) Annoying(P) Starving(C) Poor(C) Name = Eurica (C) Mother 	<p>Activity:</p> <ul style="list-style-type: none"> D1 Maintenance(R) D2 ? D3 offering muffins to D1 (C) D3 begs for food (C) D1 eating(Muffins)(C) D2 eating(Muffins)(C) D2 eat(muffins) for saving D3 children (C) D1 talking aside to D2 (C) D1 exploits D3 (P) D3 does counterfeit (R) <p>Location:</p> <ul style="list-style-type: none"> Inside(P) <ul style="list-style-type: none"> Café(P) <p>Characters:</p> <ul style="list-style-type: none"> D1 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer(C) Ignorant(R) Arrogant(R) D2.name= Scott Andersen (C) D2 Client(P) <ul style="list-style-type: none"> Fair Trade Buyer (C) D2.name= Scott Phelps (C) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) Colombian(R) Annoying(R) Starving(C) Poor(C) Name = Eurica (C) Mother <ul style="list-style-type: none"> <15 kids (C) wants escape(C) Pablo Escobar (R) 	<p>Activity:</p> <ul style="list-style-type: none"> D1 sweeping(R) D3 offering muffins to D1 (C) D3 begs for food (C) D1 eating(Muffins)(C) D2 eating(Muffins)(C) D2 eat(muffins) for saving D3 children (P) D1 talking aside to D2 (C) D1 exploits D3 (R) D3 does counterfeit (P) <p>Location:</p> <ul style="list-style-type: none"> Inside(R) <ul style="list-style-type: none"> CoffeeShop(R) <p>Characters:</p> <ul style="list-style-type: none"> D1 worker(R) <ul style="list-style-type: none"> Fair Trade Buyer(C) Ignorant(R) Arrogant(R) D2.name= Scott Andersen (C) D2 Client(R) <ul style="list-style-type: none"> Fair Trade Buyer (C) D2.name= Scott Phelps (C) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) Colombian(P) Annoying(R) 	Tilt Offer
---	--	---	------------

43

<ul style="list-style-type: none"> <15 kids (C) <ul style="list-style-type: none"> wants escape(C) Pablo Escobar (P) <p>Relation</p> <ul style="list-style-type: none"> Co-workers(P) StatusD1D2: Equal(P) StatusD1D2: Equal(C) D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) RobberVictim(D3,D1D2) D3.Status >>>> D1.Status(C) D3.Status <<< D1.Status(P) <p>Props:</p> <ul style="list-style-type: none"> Rake or mop? (A) Muffins(C) <ul style="list-style-type: none"> Colombian(C) Organic(P) FairTrade(C) Delicious(P) Counterfeit (P) Door(R) Muffin Tray(C) 	<p>Relation</p> <ul style="list-style-type: none"> D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) RobberVictim(D3,D1D2) D3.Status >>>> D1.Status(C) D3.Status <<< D1.Status(P) Mop?(R) Muffins(C) <ul style="list-style-type: none"> Colombian(C) Organic(R) FairTrade(C) Delicious(C) Counterfeit (R) Door(P) Muffin Tray(C) 	<ul style="list-style-type: none"> Starving(C) Poor(C) Name = Eurica (C) Mother <ul style="list-style-type: none"> <15 kids (C) wants escape(C) Pablo Escobar (R) <p>Relation</p> <ul style="list-style-type: none"> D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) RobberVictim(D3,D1D2) D3.Status >>>> D1.Status(C) D3.Status <<< D1.Status(P) <p>Props:</p> <ul style="list-style-type: none"> Broom (R) Muffins(C) <ul style="list-style-type: none"> Colombian(C) Organic(P) FairTrade(C) Delicious(C) Counterfeit (R) Door(R) Muffin Tray(C) 	
---	--	--	--

Consistency

At this moment if we only consider the consistent information between the players we get new entries resulting from D1 tilt offers

Activity:

- D3 offering muffins to D1 (C)

44

- D3 begs for food (C)
- D1 eating(Muffins)(C)
- D2 eating(Muffins)(C)
- D2 eat(muffins) for saving D3 children (P)
- D1 talking aside to D2 (C)
- D1 exploits D3 (R)
- D3 does counterfeit (P)

Location:

- Inside(R)

Characters:

- D1 worker(R)
 - Fair Trade Buyer(C)
 - Ignorant(R)
 - Arrogant(R)
 - Fair Trade Buyer (C)
 - D1.name = Scott Phelps
- D2
 - D2.name= Scott Andersen (C)
- D3 producer(C)
 - Fair Trade Producer(C)
 - Colombian(P)

- Annoying(R)
- Starving(C)
- Poor(C)
- Name = Eurica (C)
- Mother
 - <15 kids (C)
- wants escape(C)
- Pablo Escobar (P)

Relation

- StatusD1D2: Equal(R)
- D1 D3(C):
 - Boss(D3, D1)(C)
 - D3.Status <<< D1.Status(C)

Props:

- Muffins(C)
 - Colombian(C)
 - Organic(P)
 - FairTrade(C)
 - Delicious(C)
- Door(R)

Muffin Tray(C)

Emergent Frame

At this moment if we consider Sawyer's frame definition, where only accepted offers can represent the emergent frame the tilt variables have not entered the emergent frame. We get the following emergent frame representation:

Activity:

- D3 offering muffins to D1 (C)
- D3 begs for food (C)
- D1 eating(Muffins)(C)
- D2 eating(Muffins)(C)
- D2 eat(muffins) for saving D3 children (P)
- D1 talking aside to D2 (C)

Location:

- Inside(R)

Characters:

- D1 worker(R)
 - Fair Trade Buyer(C)
 - Fair Trade Buyer (C)
 - D1.name = Scott Phelps
- D2
 - D2.name= Scott Andersen (C)
- D3 producer(C)

45

- Fair Trade Producer(C)
- Colombian(P)
- Annoying(R)
- Starving(C)
- Poor(C)
- Name = Eurica (C)
- Mother
 - <15 kids (C)
- wants escape(C)

Relation

- StatusD1D2: Equal(R)

- D1 D3(C):
 - Boss(D3, D1)(C)
 - D3.Status <<< D1.Status(C)

Props:

- Muffins(C)
 - Colombian(C)
 - Organic(P)
 - FairTrade(C)
 - Delicious(C)
- Door(R)

Muffin Tray(C)

Turn 49 to 51

D1 "I was trying to make the offer that she was much... higher status than she appeared to be, that she was covering as a lower medium labor individual but actually she had money and connections that were far more powerful than any of us (D1, D2) could ever match."

D3 "I remembered the name that he (Pablo Escobar) was in a main Colombian cartel". "My decision to react strongly... clear specific endowment about my character... to have a emotional reaction about something about myself, makes it more weight something more interesting to explore, keep the character interesting and keeping the scene interesting to explore..."

49	D3	<revolted> No, no! You did not Mr.Phelps!	Tilt Acceptance D3 confirms
50	D1	<to D3> EXCUSE ME Eurica, this is a private conversation!	D1 plays the game
51	D3	<Angry pointing at D1> Mr. Phelps do not bring Mr.Escobar. <Anxious. Bytes her hand and tries to calm herself>	D3 confirms Escobar

46

Ai.....

- 52 D1 <to D2> D1 proposes D3 is a Drug Mule
- How do you think she had those 15 kids and got them to the States?
- She used to be a drug mule man.
- 53 D2 <astonished to D3>
- Is this right? Is this true?

D1 Frame	D2 Frame	D3 Frame	Comments
<p>Activity:</p> <ul style="list-style-type: none"> D1 Mopping(A) <ul style="list-style-type: none"> breaking leaves(A) D1 taking break(P) D2 taking break(P) D3 offering muffins to D1 (C) D3 begs for food (C) D1 eating(Muffins)(C) D2 eating(Muffins)(C) D2 eat(muffins) for saving D3 children (C) D1 talking aside to D2 (C) D1 exploits D3 (R) D3 does counterfeit (P) <p>Location:</p> <ul style="list-style-type: none"> Inside(A) <ul style="list-style-type: none"> Outside Office(P) <p>Characters</p> <ul style="list-style-type: none"> D1 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer(C) 	<p>Activity:</p> <ul style="list-style-type: none"> D1 Maintenance(R) D2 ? D3 offering muffins to D1 (C) D3 begs for food (C) D1 eating(Muffins)(C) D2 eating(Muffins)(C) D2 eat(muffins) for saving D3 children (C) D1 talking aside to D2 (C) D1 exploits D3 (P) D3 does counterfeit (R) <p>Location:</p> <ul style="list-style-type: none"> Inside(P) <ul style="list-style-type: none"> Café(P) <p>Characters:</p> <ul style="list-style-type: none"> D1 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer(C) Ignorant(R) Arrogant(R) 	<p>Activity:</p> <ul style="list-style-type: none"> D1 sweeping(R) D3 offering muffins to D1 (C) D3 begs for food (C) D1 eating(Muffins)(C) D2 eating(Muffins)(C) D2 eat(muffins) for saving D3 children (P) D1 talking aside to D2 (C) D1 exploits D3 (R) D3 does counterfeit (P) <p>Location:</p> <ul style="list-style-type: none"> Inside(R) <ul style="list-style-type: none"> CoffeeShop(R) <p>Characters:</p> <ul style="list-style-type: none"> D1 worker(R) <ul style="list-style-type: none"> Fair Trade Buyer(C) Ignorant(R) 	Tilt Offer

47

<ul style="list-style-type: none"> Ignorant(P) Arrogant(P) D2.name= Scott Andersen (C) D2 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer (C) D2.name= Scott Phelps (C) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) Colombian(R) Annoying(P) Starving(C) Poor(C) Name = Eurica (C) Mother <ul style="list-style-type: none"> <15 kids (C) wants escape(C) Drug Mule(P) Pablo Escobar (P) <p>Relation</p> <ul style="list-style-type: none"> StatusD1D2: Equal(C) D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) RobberVictim(D3,D1D2) D3.Status >>>> D1.Status(A) D3.Status <<< D1.Status(C) <p>Props:</p> <ul style="list-style-type: none"> Rake or mop? (A) Muffins(C) <ul style="list-style-type: none"> Colombian(C) Organic(P) FairTrade(C) Delicious(P) Counterfeit (P) Door(R) Muffin Tray(C) 	<ul style="list-style-type: none"> D2.name= Scott Andersen (C) D2 Client(P) <ul style="list-style-type: none"> Fair Trade Buyer (C) D2.name= Scott Phelps (C) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) Colombian(R) Annoying(R) Starving(C) Poor(C) Name = Eurica (C) Mother <ul style="list-style-type: none"> <15 kids (C) wants escape(C) Drug Mule(P) Pablo Escobar (R) <p>Relation</p> <ul style="list-style-type: none"> StatusD1D2: Equal(C) D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) RobberVictim(D3,D1D2) D3.Status >>>> D1.Status(A) D3.Status <<< D1.Status(C) <p>Props:</p> <ul style="list-style-type: none"> Mop?(R) Muffins(C) <ul style="list-style-type: none"> Colombian(C) Organic(R) FairTrade(C) Delicious(C) Counterfeit (R) Door(P) Muffin Tray(C) 	<ul style="list-style-type: none"> Arrogant(R) D2.name= Scott Andersen (C) D2 Client(R) <ul style="list-style-type: none"> Fair Trade Buyer (C) D2.name= Scott Phelps (C) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) Colombian(P) Annoying(R) Starving(C) Poor(C) Name = Eurica (C) Mother <ul style="list-style-type: none"> <15 kids (C) wants escape(C) Drug Mule(P) Pablo Escobar (R) <p>Relation</p> <ul style="list-style-type: none"> StatusD1D2: Equal(C) D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) RobberVictim(D3,D1D2) D3.Status >>>> D1.Status(A) D3.Status <<< D1.Status(C) <p>Props:</p> <ul style="list-style-type: none"> Broom (R) Muffins(C) <ul style="list-style-type: none"> Colombian(C) 	
--	---	--	--

48

		<ul style="list-style-type: none"> ○ Organic(P) ○ FairTrade(C) ○ Delicious(C) ○ Counterfeit (R) • Door(R) • Muffin Tray(C) 	
--	--	--	--

Consistency

At this moment if we only consider the consistent information between the players we get new entries resulting from D1 tilt offers

Activity:

- D3 offering muffins to D1 (C)
- D3 begs for food (C)
- D1 eating(Muffins)(C)
- D2 eating(Muffins)(C)
- D2 eat(muffins) for saving D3 children (P)
- D1 talking aside to D2 (C)
- D1 exploits D3 (R)
- D3 does counterfeit (P)

Location:

- Inside(R)

Characters:

- D1 worker(R)
 - Fair Trade Buyer(C)
 - Ignorant(R)
 - Arrogant(R)
 - Fair Trade Buyer (C)
 - D1.name = Scott Phelps
- D2
 - D2.name= Scott Andersen (C)
- D3 producer(C)
 - Fair Trade Producer(C)

Relation

- StatusD1D2: Equal(R)
- StatusD1D2: Equal(C)
- D1 D3(C):
 - Boss(D3, D1)(A)
 - RobberVictim(D3,D1D2)
 - D3.Status >>>> D1.Status(A)
 - D3.Status <<<< D1.Status(C)

Props:

- Muffins(C)
 - Colombian(C)
 - Organic(P)

49

- FairTrade(C)
- Delicious(C)

- Door(R)
- Muffin Tray(C)

Emergent Frame

At this moment if we consider Sawyer's frame definition, where only accepted offers can represent the emergent frame we get the following emergent frame representation:

Activity:

- D3 offering muffins to D1 (C)
- D3 begs for food (C)
- D1 eating(Muffins)(C)
- D2 eating(Muffins)(C)
- D2 eat(muffins) for saving D3 children (P)
- D1 talking aside to D2 (C)

Location:

- Inside(R)

Characters:

- D1 worker(R)
 - Fair Trade Buyer(C)
 - Fair Trade Buyer (C)
 - D1.name = Scott Phelps
- D2
 - D2.name= Scott Andersen (C)
- D3 producer(C)
 - Fair Trade Producer(C)
 - Colombian(P)
 - Annoying(R)

Relation

- StatusD1D2: Equal(R)
- D1 D3(C):
 - Boss(D3, D1)(C)
 - D3.Status <<<< D1.Status(C)

Props:

- Muffins(C)
 - Colombian(C)
 - Organic(P)
 - FairTrade(C)
 - Delicious(C)

- Door(R)
- Muffin Tray(C)

Turn 54 to 64

In this part of the scene D3 connects all loose ends and resolves the scene by herself in a high status. D1 and D2 move to low status and are left alone in a resolved scene, exploring the reality of it until they call a scene.

50

54 D3 <In high status>
<takes the muffin tray and faces D1 and D2>

54 D1 <to D2> D1 confirms Counterfeit
Scott Andersen, I wouldn't lie to you.

54 D3 Mr. Andersen, your 15th muffin has a happy ending to it... D3 confirms counterfeit muffins
D3 confirms Drug mule
D3 proposes muffins with drugs

55 D2 <takes another muffin, smiling>

56 D1 I can't compete with that... D1 confirms muffins with drugs

57 D2 I'm already starting to feel a little... sociable... D2 confirms muffins with drugs
D2 offers D2 under drug influence

58 D3 <peaceful, to D2> D3 explores the reality of muffins with drugs
Look at you, you're happy. They're happy muffins. D3 confirms D2 under drug influence
<again to D1, with evilness>
They're happy muffins Mr. Phelps

59 D1 <scared> D1 explores the reality of muffins with drugs.
Oh my god! You've drugged this entire accounting company. No wonder I was breaking leaves in the break room. D1 offers D1 under Drug influence
D1 Confirms Breaking leaves in the break room.

51

D1 proposes accounting firm
D1 proposes D3 drugs accounting firm

60 D3 So I have. And now I stole you're accountings secrets, now Escobar can go richest. D3 confirms D3 drugs accounting firm
D3 proposes to have stoles accounting secrets
D3 proposes to take money to Escobar
D3 confirms working for Escobar

61 D1 <laughing, dropping out of character> From here on D1 and D2 explore the reality of the scene
That's gonna be tough, considering where he is in the present day.

Audience Laughs

62 D2 Moves closer to D1

63 D3 I have ??????
<moves out celebrating>

Audience Laughs

64 D1 <condescendent to D2>
Where's the third world now?

65 D2 <high>
You're shirt tastes amazing

Audience Applause

D1 Frame	D2 Frame	D3 Frame	Comments
----------	----------	----------	----------

52

<p>Activity:</p> <ul style="list-style-type: none"> D1 Mopping(A) <ul style="list-style-type: none"> breaking leaves(C) D1 taking break(P) D2 taking break(P) D3 offering muffins to D1 (C) D3 begs for food (C) D1 eating(Muffins)(C) D2 eating(Muffins)(C) D2 eat(muffins) for saving D3 children (C) D1 talking aside to D2 (C) D1 exploits D3 (R) D3 does counterfeit (C) D3 drugs accounting firm(C) D3 steals firm(C) <p>Location:</p> <ul style="list-style-type: none"> Inside(A) <ul style="list-style-type: none"> Outside Office(P) <p>Characters</p> <ul style="list-style-type: none"> D1 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer(C) Ignorant(P) Arrogant(P) D2.name= Scott Andersen (C) D1 on drug influence (C) D2 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer (C) D2.name= Scott Phelps (C) D2 on drug influence (C) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) 	<p>Activity:</p> <ul style="list-style-type: none"> D1 Maintenance(R) <ul style="list-style-type: none"> breaking leaves(C) D2 ? D3 offering muffins to D1 (C) D3 begs for food (C) D1 eating(Muffins)(C) D2 eating(Muffins)(C) D2 eat(muffins) for saving D3 children (C) D1 talking aside to D2 (C) D1 exploits D3 (P) D3 does counterfeit (C) D3 drugs accounting firm(C) D3 steals firm(C) <p>Location:</p> <ul style="list-style-type: none"> Inside(P) Café(P) <p>Characters:</p> <ul style="list-style-type: none"> D1 worker(P) <ul style="list-style-type: none"> Fair Trade Buyer(C) Ignorant(R) Arrogant(R) D2.name= Scott Andersen (C) D1 on drug influence (C) D2 Client(P) <ul style="list-style-type: none"> Fair Trade Buyer (C) D2.name= Scott Phelps (C) D2 on drug influence (C) D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) Colombian(R) Annoying(R) Starving(C) Poor(C) 	<p>Activity:</p> <ul style="list-style-type: none"> D1 sweeping(R) <ul style="list-style-type: none"> breaking leaves(C) D3 offering muffins to D1 (C) D3 begs for food (C) D1 eating(Muffins)(C) D2 eating(Muffins)(C) D2 eat(muffins) for saving D3 children (P) D1 talking aside to D2 (C) D1 exploits D3 (R) D3 does counterfeit (C) D3 drugs accounting firm(C) D3 steals firm(C) <p>Location:</p> <ul style="list-style-type: none"> Inside(R) CoffeeShop(R) <p>Characters:</p> <ul style="list-style-type: none"> D1 worker(R) <ul style="list-style-type: none"> Fair Trade Buyer(C) Ignorant(R) Arrogant(R) D2.name= Scott Andersen (C) D1 on drug influence (C) D2 Client(R) <ul style="list-style-type: none"> Fair Trade Buyer (C) D2.name= Scott Phelps (C) D2 on drug influence (C) 	Tying Loose Ends
---	--	--	------------------

53

<ul style="list-style-type: none"> Colombian(R) Annoying(P) Starving(C) Poor(C) Name = Eurica (C) Mother <ul style="list-style-type: none"> <15 kids (C) wants escape(C) Drug Mule(C) Works for Escobar(C) <p>Pablo Escobar (P)</p> <p>Relation</p> <ul style="list-style-type: none"> StatusD1D2: Equal(C) D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) RobberVictim(D3,D1D2) D3.Status >>>> D1.Status(A) D3.Status <<<< D1.Status(C) <p>Props:</p> <ul style="list-style-type: none"> Rake or mop? (A) Muffins(C) <ul style="list-style-type: none"> Colombian(C) Organic(P) FairTrade(C) Delicious(P) Counterfeit (C) Drugged(C) Door(R) Muffin Tray(C) 	<ul style="list-style-type: none"> Name = Eurica (C) Mother <ul style="list-style-type: none"> <15 kids (C) wants escape(C) Drug Mule(C) Works for Escobar(C) <p>Pablo Escobar (R)</p> <p>Relation</p> <ul style="list-style-type: none"> StatusD1D2: Equal(C) D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) RobberVictim(D3,D1D2) D3.Status >>>> D1.Status(A) D3.Status <<<< D1.Status(C) <p>Props:</p> <ul style="list-style-type: none"> Mop?(R) Muffins(C) <ul style="list-style-type: none"> Colombian(C) Organic(R) FairTrade(C) Delicious(C) Counterfeit (C) Drugged(C) Door(P) Muffin Tray(C) 	<ul style="list-style-type: none"> D3 producer(C) <ul style="list-style-type: none"> Fair Trade Producer(C) Colombian(P) Annoying(R) Starving(C) Poor(C) Name = Eurica (C) Mother <ul style="list-style-type: none"> <15 kids (C) wants escape(C) Drug Mule(C) Works for Escobar(C) Pablo Escobar (R) <p>Relation</p> <ul style="list-style-type: none"> StatusD1D2: Equal(C) D1 D3(C): <ul style="list-style-type: none"> Boss(D3, D1)(C) RobberVictim(D3,D1D2) D3.Status >>>> D1.Status(C) D3.Status <<<< D1.Status(C) <p>Props:</p> <ul style="list-style-type: none"> Broom (R) Muffins(C) <ul style="list-style-type: none"> Colombian(C) Organic(P) FairTrade(C) Delicious(C) Counterfeit (C) Drugged(C) 	
--	--	--	--

54

		<ul style="list-style-type: none"> • Door(R) • Muffin Tray(C) 	
--	--	---	--

Consistency

At this moment if we only consider the consistent information between the players we get new entries resulting from D1 tilt offers

Activity:

- D1breaking leaves(C)
- D2 taking break(P)
- D3 offering muffins to D1 (C)
- D3 begs for food (C)
- D1 eating(Muffins)(C)
- D2 eating(Muffins)(C)
- D2 eat(muffins) for saving D3 children (C)
- D1 talking aside to D2 (C)
- D3 does counterfeit (C)
- D3 drugs accounting firm(C)
- D3 steals firm(C)

Location:

Characters

- D1 worker(P)
 - Fair Trade Buyer(C)
 - D2.name= Scott Andersen (C)
 - D1 on drug influence (C)
- D2 worker(P)
 - Fair Trade Buyer (C)
 - D2.name= Scott Phelps (C)
 - D2 on drug influence (C)
- D3 producer(C)
 - Fair Trade Producer(C)
 - Starving(C)

- Poor(C)
- Name = Eurica (C)
- Mother
 - <15 kids (C)
- wants escape(C)
- Drug Mule(C)
- Works for Escobar(C)
- Pablo Escobar (C)

Relation

- StatusD1D2: Equal(C)
- D1 D3(C):
 - Boss(D3, D1)(C)
 - RobberVictim(D3,D1D2)
 - D3.Status >>>> D1.Status(C)
 - D3.Status <<< D1.Status(C)

Props:

- Rake or mop? (A)
- Muffins(C)
 - Colombian(C)
 - FairTrade(C)
 - Delicious(C)
 - Counterfeit (C)
 - Drugged(C)

Muffin Tray(C)

55

Emergent Frame

At this moment if we consider Sawyer's frame definition, where only accepted offers can represent the emergent frame we get the following emergent frame representation:

Activity:

- D1breaking leaves(C)
- D2 taking break(P)
- D3 offering muffins to D1 (C)
- D3 begs for food (C)
- D1 eating(Muffins)(C)
- D2 eating(Muffins)(C)
- D2 eat(muffins) for saving D3 children (C)
- D1 talking aside to D2 (C)
- D3 does counterfeit (C)
- D3 drugs accounting firm(C)
- D3 steals firm(C)

Location:

Characters

- D1 worker(P)
 - Fair Trade Buyer(C)
 - D2.name= Scott Andersen (C)
 - D1 on drug influence (C)
- D2 worker(P)
 - Fair Trade Buyer (C)
 - D2.name= Scott Phelps (C)
 - D2 on drug influence (C)
- D3 producer(C)
 - Fair Trade Producer(C)
 - Starving(C)

- Poor(C)
- Name = Eurica (C)
- Mother
 - <15 kids (C)
- wants escape(C)
- Drug Mule(C)
- Works for Escobar(C)
- Pablo Escobar (C)

Relation

- StatusD1D2: Equal(C)
- D1 D3(C):
 - Boss(D3, D1)(C)
 - RobberVictim(D3,D1D2)
 - D3.Status >>>> D1.Status(C)
 - D3.Status <<< D1.Status(C)

Props:

- Rake or mop? (A)
- Muffins(C)
 - Colombian(C)
 - FairTrade(C)
 - Delicious(C)
 - Counterfeit (C)
 - Drugged(C)

Muffin Tray(C)

56

APPENDIX B

Wife Zoo Scene Analysis

Wife Zoo Scene

Constraints

Three friends from college at the zoo D7, D5, D6
Scene length – 3min.

Transcription with comments

Turn	Player	Action	Comments	Story Step	Function
1	D6	<D7 and D5 are at center stage> <D6 enters from right stage> D6 - I got them! Oh man! snowcones, they got all kind fflavours. mmm	Friends(D7, D5, D6) NewObject(Snowcone) Like(snowcones, D6)++ Has(Snowcone, D6,3)	Platform Buildup Platform Buildup	Build Platform
2	D6	<D6 gives a snowcone to each>	Like(snowcones, D7)++ Like(snowcones, D5)++ Has(snowcone, D6,1) Has(snowcone, D7, 1) Has(snowcone, D5,1)	Platform Buildup	
3	D7	(sorry) Ow I was eating my popcorn Thanks man! <D6 goes back to right stage>	NewObject(Popcorn) Has(Popcorn, D7,1) D7 and D6 equal status	Platform Buildup	Build Platform
4	D6	They put all the kinds of syrups in there. <looking at D6>	Snowcone(syrup) = True	Platform Buildup	Exploring Platform
5	D5	oh it's a suicide slushy	Syrup is a "Suicide Slushy"	Platform Buildup	Exploring Platform
6	D6	mhm		Platform Buildup	Exploring Platform
7	D5	<looking at D7> Hey you should feed your popcorn to the lions!	Offer(D7, feed(lions), popcorn)	Platform Buildup	Build Platform
8	D7	<looking at D5> <patronizing> hey hmm you know, something Ted they don't really haam want you to feed the animals because they're on specialized DIETS!!!	D7 appraise feed(lions) <<< D7 offers D5(name)= Ted Offer(D6, occupation related with animals)	Tilt Buildup	Tilt Offer (<i>Introducing conflict</i>) <i>Tilt Variable = Feed Lion</i> Build Platform
9	D5	You know you should know this since you're a.... <looking at D7>... a zoologist?! D6 <oversteamed by the others>— yup see that one over there...	D5(zoologist) = true	Tilt Buildup	Build Platform
10	D7	<looking at D5>... When you where at zoologist school....	D5(asBeen) = zoology school	Tilt Buildup	Build platform
11	D5	<assertive> yeah here's what I've learned...Animals are lesser than humans! (Pause) Humans are better!	D5 believes humans are better than animals. D5 harms animals	Tilt Buildup Tilt Buildup	Build Platform <i>D5 accepts conflict offer and raises stakes</i> <i>Tilt Variable = [Feed Lion, Animal Rights]</i> Tilt Riding
12	D7	<patronizing> You know that's not probably what they taught you....	D6(learned to Harm animals) = False	Tilt Buildup	Tilt Riding
13	D6	<looking and pointing at something down front stage for a while tries to speak but it's interrupted>	D6 is adding something to the scenario, maybe lions...	Tilt Buildup	Build Up
14	D5	No, that's what I've learned.. <looking right stage at D6>	D6(learned to Harm animals) = True	Tilt Buildup	Tilt Riding
15	D6	<looking at the other two> huu I don't take any part of that stuff, I just know for a fact that you can shuck stuff at them, and I'm gonna chuck this <holding his	Offer(D6, schuk(snowcone, lion)) AddElement(Lion)	Tilt Buildup Tilt Buildup	<i>D6 takes sides</i> Tilt Riding Build Platform <i>D6 creates a lion</i>

1

2

		slushy hand high> at that lion over there <pointing front stage with the other hand>			
16	D6	<hesitates and looks around> <D7 and D5 argue with each other>	D6 know he's breaking a rule.... D7 and D5 present arguments for their different appraisals of action Schuck() / Feed()	Tilt Buildup Tilt Buildup	Tilt Riding <i>Tilt Variables = [Feed Lion, Animal Rights, Breaking Rules]</i>
17	D5	It seems like a wise decision	D5 supports Feed(lion)	Tilt Buildup	Tilt Riding
18	D7	I'm so totally opposed to this	D7 rejects Feed(lion)	Tilt Buildup	Tilt Riding
19	D5	It doesn't seem that bad. <assertive> It doesn't seem that bad...		Tilt Buildup	Tilt Riding
20	D7	<Ironic> Really?		Tilt Buildup	Tilt Riding
21	D6	<D6 makes a throwing gesture and sound towards frontstage>	Has(snowcone, D6, 1) = False	Tilt Buildup	Tilt Riding
22	D7	<looking frontstage with disapproval> ohMan!	D7 disapproves D6 action	Tilt Buildup	Tilt Riding
23	D5	<assertive> Nothing ironic about him throwing a suicide snowcone to the lion.	Has(snowcons, lion,1)	Tilt Buildup	Tilt Riding
24	D6	<Looking down frontstage and mocking> What are you gonna do about it lion? It's awesome! Lick it! Lick it!	D6 harms animals Eats(snowcone, lion)	Tilt Buildup	Tilt Riding
25	D7	<calm and patronizing>You know Tony...it's not cool. Sorry	D7 criticizes D6	Tilt Buildup	Exploring Platform
26	D6	<calm and assertive> I don't have a wife, and I don't have a children <u>like you've got!</u>	STRONG OFFER Has(D7, wife) Has(D7, Children) D6 appraise(Wife)<<< D6 appraise(Children)<<<	Tilt Buildup	Tilt Offer Tilt variable = [D7.Wife, D7.Children]
27	D5	<raising hand towards D6> Yeah! Hi 5 Dude!	D5 agrees with D6 D5(affinity,D6) >>>	Tilt Buildup	Tilt Riding
28	D5	<hi 5ing D6> Yeah! I wish I didn't have a wife and children	D6(affinity,D5) >>> Has(D5, Wife) Has(D5, Children) D5 appraise(Wife)<<< D5 appraise(Children)<<<	Tilt Buildup	Exploring Platform
29	D6	<pointing at D7> <patronizing> Yeah and I wish you didn't have that wife and those children...	D6 appraise(D5.wife)<<< D6 appraise(D5.children)<<<	Tilt Buildup	Tilt Riding
30		(Pause) D7 is completely puzzled!!!!			
31	D7	What the Fuck dudes <complaining> Why are you trashing my wife? We do this once a year,, <D5 faces D7 in the eyes> That's right... (Like throw it out)	D7 offers action(insult(D7.wife))	Tilt Buildup Tilt Buildup	Tilt Riding Build Platform
		we get together in a funny little place that I never get to pick <D5 acknowledges with superiority> and you <UNCOMPREHENSIBLE WORDS> come here and you pick on my WIFE! <Pause> <Tries to calm speech> I Can't help it, that she's the most beautiful girl in the world....	Context (location picked by D5 and D6) (frequency once a year) D7 never picks location D7 status<<<	Tilt Buildup Tilt Buildup	Explore (confirming) Build Platform
32	D5	<D5 looks back from D7 to front stage> <unconvinced and ironic> oh yeah.... She's very, very..... very....	D7 offers(support(D7.wife)) to himself	Tilt Buildup	Build Platform
33	D7	Are you calling my wife ugly?	D7 offers D7.wife(ugly)=True	Tilt Buildup	Tilt Riding
34	D5	D5 <moving his head to the sides, with a mocking smile> I'm	D5 accepts offer D5 respects D7	Tilt Buildup	Tilt Riding

[illegible]

					physically abused]
54	D5	Why do you wear a.... Show us, show us around you're neck.	D5 explores beating signs	Problem	Platform Exploration
55	D7	<stretches his collar down, with suffering face>	Offers Has(D7, beatingMarks)	Problem	Platform Exploration
56	D6	<looking at D7 neck> oh my god!	D7 accepts offer Has(D7, beatingMarks)	Problem	Platform Exploration
57	D7	<suffering> she chains me to the wall. She chains me up there	D6 confirms	Problem	Platform Exploration
58	D6	She only lets you out once a year....and its to hangout with us.	D7 explores beating marks	Problem	Platform Exploration
59	D7	You know what I eat all year long? "Ground" of krystal burgers... It's Krystal burgers in a blender.. It serves this kind of mash it's all I eat.	D7 offers(D7(prisoner)=True) Offers(D7.wife(abuser)=True) D6 offers(annualLeave)	Problem	Platform Exploration
		Have you ever smelled one of my farts on one of these trips?	D7 explores D7(Prisoner) D7 offers Eats(D7, badfood) Offers D7 is badly treaten Explores effects of Bad food... Offers D7.farts(smell) = True	Problem	Tilt Offer = [D7.Wife, D7.Children, D7.Wife.Manish, D7 abused, saved7, D7 physically abused, smelly farts] Platform Exploration
60	D6	<looking down> I really didn't notice, don't want to talk about it.	D6 accepts offer with disgust	Problem	Platform Exploration
61	D7	You know what that is?	D7 describes in more detail	Problem	Platform Exploration
62	D5	I really thought it was the animals....	D5 accepts and describes it...	Problem	Platform Exploration
63	D7	No that's why I'm really happy you chose the zoo this year, because I can blend in...	D7 uses context of zoo in fart description	Problem	Platform Exploration
64	D6	<looking at D5> I really can't finish this snowcone...	D5 is disgusted.. D5 offers doing something to Snowcone	Problem	Platform Exploration
65	D7	Why don't you throw away to the poor animal? <ironic>	D7 offers(feedanimals(snowcone))	Problem	Platform Exploration
66	D6	You listen <points at D7> when we leave this zoo you can come	D6 offers solution to abuses	Problem	Tilt Offer Tilt Variable = 7
67	D7	stay with me.			D7.moveOut
68	D6	Really? That's right. - We've got a spare room in the house. I want you to... <D5 nods his head saying yes>	D6 plays character D6 describes solution D5 confirms it	Problem Problem	Platform Exploration Platform Exploration
69	D7	<interrupts> - She's got me chipped!(Pause) there's a microchip...	D7 offers new problem(chip)	Problem	Tilt Offer = Can't move out.]D7.chipped invalidates D7.moveOut]
70	D6	I'll cut it out. I'll cut it out. I've tagged animals. I know what it's like. <D7 puts an arm around D6>	D6 uses his knowledge to resolve problem	Problem	Tilt Offer = Cut chip. [D7-ChipRemoved, D7.MovesOut]
71	D5	<towards D7> she's not gonna know who you are, because you're taking her to court. You're dissolving this hideous monstrosity of a marriage.	D5 offers(dissolvingMarriage)	Resolution	Tying Loose ends
72	D6	Hey guys... let's cone that lion together. C'mon all hands on the cone. <D5 and D7 place hand on D6's cone>	D6 offers(feed(Lion), all)	Resolution	Denouement, Tying loose ends
73	D7	I'm still really opposed to this....	D7 plays character but accepts offer	Resolution	Tying loose ends
74	D6	Take that lion!	D6 plays characters	Resolution	Denouement
75	All	<all of them throw the cone together>		Resolution	Platform Exploration
76	D7	oh look at him he's got syrup in his eyes.....	D7 plays character		
77	D6	SCENE!!!!!!			

Notes on D7 Interview

6:49 - I was trying to introduce conflict You should have an opinion about something, so that you have a stance on something, so that you bases your character on something so that you can follow through on that somewhere and than onto what... something might happen that changes the environment which is the TILT....

7:47 – And I also just called him Ted it's important to try to name your partners, I eventually saw at the middle of the scene that I had no name I had no identity, I'm just the guy who is against to throwing thing at the lion.

Endowing offer(zoology)

1:30 – he probabaly named him Tony at the beginning bu it was just at this time that I heard it for the first time.

10:51 – The scene is now(2:21) shifting into something else, where no longer dealing with the lion and where dealing with my wife which he doesn't like, I'm thinking should I not like my wife or should I like my wife? And that's going to bas my opinion about what he said.

Game within the scene(I'm gonna be the outsider from which every opinion differs)

"Fuck you guys." I'm finally trying to have a say.

"cleft chin" – I'm thinking of something hilarious about cleft chin, the zoo doesn't make a difference, it's all about three guys arguing...

"2" taller than you are"

I'm thinking about my popcorn and snowcone still on my hand...

15:39 – Now this is turning into a big tilt, which is a term for when a still jus takes a big turn into turning something else. A Big offer because we were kind of going nowhere... "Wait hold on....."

"I don't wanna talk about It guys" act like abused people act like.....

smash! No! Smash! No!

Great because he's taking things out of my hand

Now you littered...

Just calling back

I was just trying to reestablish we are in a zoo. Keep it consistent for the audience. Reestablishing some facts.

"You want me to prove it"

9

I should've said YES which is a golden rule.....

I was processing the fact that allof the suddenly they were on my side,...

She chains me to the wall

I clearly got some kind of mark....

"Ground of Krystal Burger"

I tried to think of the most disgusting thing....

"Farts"

Back it out with some proof

"Microchip"

well she's gonna be able of tracking me some how.... Am I just being corny throwing an offer really late....

It seems there's a resolution coming up....

"It's so late in the scene I don't need to be throwing out things"

Wrapping it up, we resolved it we'r friends again, let's do something terrible together as friends....

Notes on D6 Interview

Sometimes you get steamrolled in a scene....

8:13 – Now that I've given them objects, now that we've got stuff, lets see what they have to say ... improv ain't about creativity is more about discovery , because people will say things that you've never expected them to say or maybe they didn't say it, or maybe they didn't say it wright? It's really a mistake and you got to react to that if you said 4 I might react to whole different context, and now you discovered it and we discovered something new.....

Steamrolling - an improviser with his own agenda....

I just wanted to make sure, whatever they're story was going to be I wanna get in there...

Around Turn 12 I'm making this OFFER you see that lion right there?!!! Any other type of people would go "yeah" even if they might not know what I'm thinking about. See that lion right there? He's dead, he's alive, he's a child, he's a baby... he already got on his mind to go somewhere else... but I actually wanted to bring that back around because it sort of got dropped and everything.

"That Lion right there." Since

I, gonna show you we can throw stuff at that lion over there....

What I was thinking is that D7 is a sad sad no fun kind of guy, his character already is one who says you can't throw shit, "oh now I've got a snowcone... you now" so what's the reason of all this torment? probably in college we all had a good time, but now for some reason his

life is muddy, so I decided to endow him with a horrible life. Then a thing happens in improv called the game ofn the scene (game -> repetitive behavior) randy and I agree that D7 wife sucks..... the cake is the real story that the audience is going to understand ... the emotional side of it is that his wife abuses him...

I was looking for the right word, I could couldn't say his wife wasn't ugly because of my character,.... Without saying what she is I'm saying what she is because I can't deny it or support what you said.....

Original in improv is a bad word, because you were so trying to be unique... you're trying to hard and it is really the flow... Raising the stakes....

The proximity he got, he entered his space he's in his bubble, what's the truth, what's the deal?

Notes on Group Interview

D7 - Here's what I'm thinking: oh shit! I already got popcorn on my hand

D6 - It was only after I came in (imitates D7) because they were two I wanted to do was (because they were a duo) I wanted to bring something in to create environment, because we usually start with only one improviser creating environment and if you start with two it pretty much gonna be talk and if you start with three you're god dam sure its gonna be talk, we're gonna look around each other and find out who each other is...

D5 - I tell you what I was seeing at the moment, because I didn't see you (points D7) eating popcorn. Cause I think you (looks at D6) start speaking as soon as you start coming into the scene, so the soon as the scene start you come in so I was thinking great tony start I'm gonna lay back.

D5 - when I come in obviously I had been in there, than I was gone and back. So obviously this had been going on all afternoon.

1:26- So this is where I thought I was going to be the guy that plays by the rules,, in this relation 3 guys from college I'm gonna be the dork "We shouldn't do that guys. We should blablalabla play by the rules" D6 - that's what o picked up.

D5 - Because the last thing you want to see is three jackasses...

D7- Yeah you don't wanna see three guys with exactly the same status that feel the same way about something

D6 - Even in pre-scene when she said three college guys that meet at the zoo, I immediately dismissed that "Uh we're all drunk at the zoo" I know what I didn't want to do, but I didn't know what I wanted to do....

D5 - When she said "three guys at the zoo" I almost ha my character there. Just like dushbag.-

D7 - Really? I didn't have my character until right here! ("Specialized diets")

D6 - I didn't have my character until after you (points D7) were the downward guy and everything was the rules... and then you (D5) became the common sense guy to me, and I was the guy that became durn and shout "hey watch this" (throwing gesture)

About D5: He wasn't wild he wasn't negative, so I just assumed he was common sense guy,,

D7 - Eventhough he's the one who offers to throw food to the lion.

D5 - I didn't know it then...

Reestablish - D5 the last thing you want is to be assuming the audience understands something that they don't ... the best way to the improviser is to make it stupidly clear.

Lion offer - D6 pints to a lion and other don't see him doing it, so he reestablishes it later without repeating the action bu by including this element in another action. D5 and D7 confirm they hadn't seen him pointing.

11

First lion offer is from D5 but it just refers lion as something to feed it doesn't present the lion in front of them.

All three agree on thinking about a story like San Francisco Zoo.

D6 - at this point we didn't get to what an established platform is.. (what's missing?) This thing that comes next which is the story of a scene D7 - Which is my life... D6 - yeah

D6 - where looking for the thing that makes it important. This is about the time where the tilt needs to be settled. We spent a lot of time gossiping

Whats the tilt?

D7 : I'm trying to defend why I like a giant....

D6: At that point when you said (looking at D7) "I like it rough" I thought "ok we can make this serious,,," make this important give it some gravity.

D7 as soon as he said "she's beating you up emotionally" I knew it was it...

D6: So I put on a serious face... put on a face of concern D7 has picked up on that face of concern

D7: I was like... shit you're right...

SMASH SMASH

D6 : his character (D7) is trying to put it back towards feeding animals and stuff like that, when actually D7 the performer is wanting the scene go to that place.

"A hundred to save you"

D7: This was a huge revelation for me, these guys are gonna back me up no matter what... now we have to deal with it, they've given me the offer "We'll do whatever it takes to help you out."

D6: I said... you want me to prove it? (to D7) And I was glad you said no,

D7: The scene wasn't about that...

D6: At this point the clock in my head is going off.. This should be wrapping up we should be finishing this scene.

You can come stay with me...

D6: in my mind it was a man shelter, a violence shelter for men, but that was by itself a scene, it was too much, so the next available scene was my house.

Chipped

D7 now I'm almost denying it... I should've accepted and we should've wrapped it, but then I thought maybe we can enrich it a little bit more. I'm adding more and I don't want to be adding more....

D6.: I wanted to say terrible marriage ...

D5: At this point a think D6 was tying up loose ends that's why I think I didn't build up more on it...

Analysis

D5 Frame	D6 Frame	D7 Frame	Comments
Constraint: • 3 min (C)	Constraint: • 3 min (C)	Constraint: • 3 min (C)	Platform build up
Location:	Location:	Location:	

12

<ul style="list-style-type: none"> • Zoo(C) Relations <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = 	<ul style="list-style-type: none"> • Zoo(C) Relations <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = 	<ul style="list-style-type: none"> • Zoo(C) Relations <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = 	
---	---	---	--

Turn 1 to 6

Turn	Player	Action	Comments
1	D6	<D7 and D5 are at center stage> <D6 enters from right stage> D6 - I got them! Oh man! snowcones, they got all kind flavours. mmm	Friends(D7, D5, D6) NewObject(Snowcone) Like(snowcones, D6)++ Has(Snowcone, D6,3)
2	D6	<D6 gives a snowcone to each>	Like(snowcones, D7)++ Like(snowcones, D5)++ Has(snowcone, D6,1) Has(snowcone, D7, 1) Has(snowcone, D5,1)
3	D7	(sorry) Ow I was eating my popcorn Thanks man! <D6 goes back to right stage>	NewObject(Popcorn) Has(Popcorn, D7,1) D7 and D6 equal status
4	D6	They put all the kinds of syrups in there. <looking at D5>	Snowcone(syrup) = True
5	D5	oh it's a suicide slushy	Syrup is a "Suicide Slushy"
6	D6	mhm	

Turn 1

D5 enters with snowcones on his hand. D7 was proposing he had popcorn.

D5 Frame	D6 Frame	D7 Frame	Comments
----------	----------	----------	----------

13

Constraint: <ul style="list-style-type: none"> • 3 min (C) Location: <ul style="list-style-type: none"> • Zoo(C) Relations <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = Activity <ul style="list-style-type: none"> • Eat(SnowCone)(P) Props <ul style="list-style-type: none"> • Snowcones(R) <ul style="list-style-type: none"> ◦ Owner D5 	Constraint: <ul style="list-style-type: none"> • 3 min (C) Location: <ul style="list-style-type: none"> • Zoo(C) Relations <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = Activity <ul style="list-style-type: none"> • Eat(SnowCone)(R) Props <ul style="list-style-type: none"> • 3 Snowcones(P) <ul style="list-style-type: none"> ◦ Owner D5 	Constraint: <ul style="list-style-type: none"> • 3 min (C) Location: <ul style="list-style-type: none"> • Zoo(C) Relations <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = Activity <ul style="list-style-type: none"> • Eat(SnowCone)(R) • Eat(Popcorn) (P) Props <ul style="list-style-type: none"> • 3 Snowcones(R) <ul style="list-style-type: none"> ◦ Owner D5 • Popcorns (P) <ul style="list-style-type: none"> ◦ Owner D7 	Platform build up
--	--	--	-------------------

14

Turn 2

D5 hands a snow cone each.

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Friend(D5, D7) <ul style="list-style-type: none"> Status = Friend(D6, D6) <ul style="list-style-type: none"> Status = <p>Activity</p> <ul style="list-style-type: none"> Eat(SnowCone)(C) <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 1 Snowcones(C) <ul style="list-style-type: none"> Owner D6 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Friend(D5, D7) <ul style="list-style-type: none"> Status = Friend(D6, D6) <ul style="list-style-type: none"> Status = <p>Activity</p> <ul style="list-style-type: none"> Eat(SnowCone)(C) <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 1 Snowcones(C) <ul style="list-style-type: none"> Owner D6 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Friend(D5, D7) <ul style="list-style-type: none"> Status = Friend(D6, D6) <ul style="list-style-type: none"> Status = <p>Activity</p> <ul style="list-style-type: none"> Eat(SnowCone)(C) Eat(Popcorn) (P) <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 1 Snowcones(C) <ul style="list-style-type: none"> Owner D6 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Popcorns (P) <ul style="list-style-type: none"> Owner D7 	Platform build up

15

Turn 3

D7 informs he was holding popcorns. D7 thanks informally

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Friend(D5, D7) <ul style="list-style-type: none"> Status = C Friend(D6, D6) <ul style="list-style-type: none"> Status = <p>Activity</p> <ul style="list-style-type: none"> Eat(D7,Popcorn) (C) Eat(SnowCone)(C) <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 1 Snowcones(C) <ul style="list-style-type: none"> Owner D6 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Popcorns (C) <ul style="list-style-type: none"> Owner D7 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Friend(D5, D7) <ul style="list-style-type: none"> Status = C Friend(D6, D6) <ul style="list-style-type: none"> Status = <p>Activity</p> <ul style="list-style-type: none"> Eat(D7,Popcorn) (C) Eat(SnowCone)(C) <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 1 Snowcones(C) <ul style="list-style-type: none"> Owner D6 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Popcorns (C) <ul style="list-style-type: none"> Owner D7 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Friend(D5, D7) <ul style="list-style-type: none"> Status = C Friend(D6, D6) <ul style="list-style-type: none"> Status = <p>Activity</p> <ul style="list-style-type: none"> Eat(Popcorn) (C) Eat(SnowCone)(C) <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 1 Snowcones(C) <ul style="list-style-type: none"> Owner D6 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Popcorns (C) <ul style="list-style-type: none"> Owner D7 	Platform build up

Turn 4

They put all kin sod syrups in there.....

D5 Frame	D6 Frame	D7 Frame	Comments
----------	----------	----------	----------

16

<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = C • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup? (R) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D6 ◦ Syrup? (R) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = C • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup? (P) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D6 ◦ Syrup? (P) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (P) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = C • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = <p>Activity</p> <ul style="list-style-type: none"> • Eat(Popcorn) (C) • Eat(SnowCone)(C) <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup? (R) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D6 ◦ Syrup? (R) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 	Platform build up
--	--	---	-------------------

Turn 5

D5 names syrup "Suicide Slushy"

D5 Frame	D6 Frame	D7 Frame	Comments
----------	----------	----------	----------

17

<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = C • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = <p>Activity</p> <ul style="list-style-type: none"> • Eat(D7,Popcorn) (C) • Eat(SnowCone)(C) <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(P) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D6 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(P) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(P) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = C • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = <p>Activity</p> <ul style="list-style-type: none"> • Eat(D7,Popcorn) (C) • Eat(SnowCone)(C) <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(R) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D6 ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(R) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(R) • Popcons (C) <ul style="list-style-type: none"> ◦ Owner D7 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = C • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = <p>Activity</p> <ul style="list-style-type: none"> • Eat(Popcorn) (C) • Eat(SnowCone)(C) <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(R) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D6 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(R) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(R) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 	Platform build up
---	---	---	-------------------

18

Turn 6

D6 hmhmm confirms "Suicide Slushy"

D5 Frame	D6 Frame	D7 Frame	Comments
Constraint: <ul style="list-style-type: none"> • 3 min (C) Location: <ul style="list-style-type: none"> • Zoo(C) Relations <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = C • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = Activity <ul style="list-style-type: none"> • Eat(D7,Popcorn) (C) • Eat(SnowCone)(C) Props <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D6 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 	Constraint: <ul style="list-style-type: none"> • 3 min (C) Location: <ul style="list-style-type: none"> • Zoo(C) Relations <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = C • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = Activity <ul style="list-style-type: none"> • Eat(D7,Popcorn) (C) • Eat(SnowCone)(C) Props <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D6 ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 	Constraint: <ul style="list-style-type: none"> • 3 min (C) Location: <ul style="list-style-type: none"> • Zoo(C) Relations <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = C • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = Activity <ul style="list-style-type: none"> • Eat(Popcorn) (C) • Eat(SnowCone)(C) Props <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D6 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 	Platform build up

19

<ul style="list-style-type: none"> ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 	<ul style="list-style-type: none"> ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 	<ul style="list-style-type: none"> ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 	
---	---	---	--

Turn 7 to 11

Turn	Player	Action	Comments
7	D5	<looking at D7> Hey you should feed your popcorn to the lions!	Offer(D7, feed(lions), popcorn)
8	D7	<looking at D5> <patronizing> hey hmm you know, something Ted they don't really haam want you to feed the animals because they're on specialized DIETS!!! You know you should know this since you're a....	D7 appraise feed(lions) <<< D7 offers D5(name)= Ted Offer(D6, occupation related with animals)
9	D5	<looking at D7>... a zoologist?! D6 <oversteamed by the others>— yup see that one right there...	D5(zoologist) = true D6 oversteamed
10	D7	<looking at D5>... When you where at zoologist	D5(asBeen) =

20

11	D5	school.... We we were in college.. <assertive> yeah here's what I've learned...Animals are lesser than humans! (Pause) Humans are better!	zoology school D5 believes humans are better than animals. D5 harms animals
----	----	---	---

Turn 7

D5 offers D7 the action of throwing popcorns to the lions.. There's no record on why D5 thought of this. From a global perspective it might seem as a Tilt offer, nevertheless from an individual perspective there's no evidence supporting that D5 was thinking of anything else but exploring the fact that D7 had two different things to eat, and that people in the zoo feed animals. In the other hand D7's opposition may seem more like a Conflict offer because he is deliberately opposing his friend, and patronizing and generating statuses variations. In some sense it might also be a tilt since it bringing a new platform into action in which the characters move from a friendly relation to a conflict relation....

D7 Interview:

6:49 - I was trying to introduce conflict You should have an opinion about something, so that you have a stance on something, so that you bases your character on something so that you can follow through on that somewhere and than onto what... something might happen that changes the environment which is the TILT....

7:47 – And I also just called him Ted it's important to try to name your partners, I eventually saw at the middle of the scene that I had no name I had no identity, I'm just the guy who is against to throwing thing at the lion.

D5 Frame	D6 Frame	D7 Frame	Comments
Constraint: <ul style="list-style-type: none"> 3 min (C) Location: <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(P) Explicit Offer <ul style="list-style-type: none"> D7 Feed(Lions, 	Constraint: <ul style="list-style-type: none"> 3 min (C) Location: <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) Explicit Offer <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) 	Constraint: <ul style="list-style-type: none"> 3 min (C) Location: <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) Explicit Offer <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) 	Platform build up

21

Popcorn) (P)	(R)	(R)	
Relations <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Friend(D5, D7) <ul style="list-style-type: none"> Status = C Friend(D6, D6) <ul style="list-style-type: none"> Status = Props <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D6 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 	Relations <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Friend(D5, D7) <ul style="list-style-type: none"> Status = C Friend(D6, D6) <ul style="list-style-type: none"> Status = Props <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) Snowcones(C) <ul style="list-style-type: none"> Owner D6 Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) Popcons (C) <ul style="list-style-type: none"> Owner D7 	Relations <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Friend(D5, D7) <ul style="list-style-type: none"> Status = C Friend(D6, D6) <ul style="list-style-type: none"> Status = Props <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D6 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 	

Turn 8

22

In this turn D7 positions his character against feeding animals in a zoo(TiltOffer). Endows D5 with a name and an offer D5 an animal related occupation... this Tilt offer will be accepted later when D5 accepts the conflict related with having an opposite position on feeding animals.

Other things such as Specialized diets and zoo rules are explorations of the platform to build arguments in turn of his position.

D7 uses a patronizing tone which raises his status over D5

D5 Frame	D6 Frame	D7 Frame	Comments
Constraint: <ul style="list-style-type: none"> • 3 min (C) Location: <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(P) ◦ Rules ◦ ↯feed(animals) (P) Explicit Offer <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) Characters <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (R) 	Constraint: <ul style="list-style-type: none"> • 3 min (C) Location: <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↯feed(animals)(R) Explicit Offer <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) Characters <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (R) Relations <ul style="list-style-type: none"> • Friend(D5, D6) 	Constraint: <ul style="list-style-type: none"> • 3 min (C) Location: <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↯feed(animals)(R) Explicit Offer <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) Characters <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(P) ◦ Occupation related with animals (P) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (P) Relations <ul style="list-style-type: none"> 	Tilt

23

Relations <ul style="list-style-type: none"> • Friend(D5, D6) • Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +D7 • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = Props <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS (C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D6 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS (C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS (C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 	Relations <ul style="list-style-type: none"> • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +D7 • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = Props <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D6 ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> ◦ Owner D7 	Relations <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +D7 • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = Props <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D6 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 	
---	--	---	--

Turn 9

D5 endows himself as a zoologist.... And D6 oversteamed

D5 Frame	D6 Frame	D7 Frame	Comments
----------	----------	----------	----------

24

<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) Rules <ul style="list-style-type: none"> ↯feed(animals)(P) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) <p>Characters</p> <ul style="list-style-type: none"> D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(P) D7 <ul style="list-style-type: none"> Against feeding animals (R) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Friend(D5, D7) <ul style="list-style-type: none"> Status = +D7 Friend(D6, D6) <ul style="list-style-type: none"> Status = 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules <ul style="list-style-type: none"> ↯feed(animals) (R) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) <p>Characters</p> <ul style="list-style-type: none"> D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(R) D7 <ul style="list-style-type: none"> Against feeding animals (R) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Friend(D5, D7) <ul style="list-style-type: none"> Status = +D7 Friend(D6, D6) <ul style="list-style-type: none"> Status = <p>Props</p>	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules <ul style="list-style-type: none"> ↯feed(animals)(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) <p>Characters</p> <ul style="list-style-type: none"> D5 <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(R) D7 <ul style="list-style-type: none"> Against feeding animals (P) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Friend(D5, D7) <ul style="list-style-type: none"> Status = +D7 Friend(D6, D6) <ul style="list-style-type: none"> Status = <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) 	Tilt
---	---	---	------

25

<p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D6 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 	<ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) Snowcones(C) <ul style="list-style-type: none"> Owner D6 Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) Popcons (C) <ul style="list-style-type: none"> Owner D7 A particular lion(P) 	<ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D6 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 	
--	---	--	--

Turn 10

D7 clarifies D5 a Zoologist and offers D5 as a former zoology school member.

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) ColleagueFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) ColleagueFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) ColleagueFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) 	Tilt

26

<ul style="list-style-type: none"> o Lions in front(P) o Rules o \negfeed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) <p>Characters</p> <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> o Name Ted(R) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (R) • D7 <ul style="list-style-type: none"> o Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +D7 • Friend(D6, D6) <ul style="list-style-type: none"> o Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D6 	<ul style="list-style-type: none"> o Lions in front(C) o A particular lion(P) o rules o \negfeed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) <p>Characters</p> <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> o Name Ted(R) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (R) • D7 <ul style="list-style-type: none"> o Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +D7 • Friend(D6, D6) <ul style="list-style-type: none"> o Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) 	<ul style="list-style-type: none"> o Lions in front(C) o rules o \negfeed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) <p>Characters</p> <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> o Name Ted(P) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (P) • D7 <ul style="list-style-type: none"> o Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +D7 • Friend(D6, D6) <ul style="list-style-type: none"> o Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) 	
---	---	---	--

27

<ul style="list-style-type: none"> o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 	<ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> o Owner D6 o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(P) 	<ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D6 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 	
--	---	---	--

Turn 11

D5 clarifies he went a Zoology school and endows himself has an animal abuser.

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • ColleagueFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(P) o Rules o \negfeed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) <p>Characters</p>	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • ColleagueFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o \negfeed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) <p>Characters</p>	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • ColleagueFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o \negfeed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) 	Tilt

28

<ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(P) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +D5 • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D6 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) 	<ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +D5 • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D6 ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) 	<p>Characters</p> <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(P) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +D5 • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D6 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 	
---	--	---	--

29

◦ Owner D7	<ul style="list-style-type: none"> • Popcons (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(P) 	<ul style="list-style-type: none"> ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 	
------------	---	---	--

Turn 12 to 25

Turn	Player	Action	Comments
12	D7	<patronizing> You know that's not probably what they taught you....	D6(learned to Harm animals) = False
13	D6	<looking and pointing at something down front stage for a while tries to speak but it's interrupted>	D6 is adding something to the scenario, maybe lions...
14	D5	No, that's what I've learned.. <looking right stage at D6>	D6(learned to Harm animals) = True
15	D6	<looking at the other two> huu I don't take any part of that stuff, I just know for a fact that you can chuck stuff at them, and I'm gonna chuck this <holding his slushy hand high> at that lion over there <pointing front stage with the other hand>	Offer(D6, schuk(snowcone, lion)) AddElement(Lion)
16	D6	<hesitates and looks around> <D7 and D5 argue with each other>	D6 know he's breaking a rule.... D7 and D5 present arguments for their different appraisals of action Schuck() / Feed()
17	D5	It seems like a wise decision	D5 supports Feed(lion)
18	D7	I'm so totally opposed to this	D7 rejects Feed(lion)
19	D5	It doesn't seem that bad. <assertive>	

30

20	D7	It doesn't seem that bad...	
21	D6	<Ironic> Really? <D6 makes a throwing gesture and sound towards frontstage>	Has(snowcone, D6, 1) = False
22	D7	<looking frontstage with disapproval> ohMan!	D7 disaproves D6 action
23	D5	<assertive> Nothing ironic about him throwing a suicide snowcone to the lion.	Has(snowcons, lion,1)
24	D6	<Looking down frontstage and mocking> What are you gonna do about it lion? It's awesome! Lick it! Lick it!	D6 harms animals Eats(snowcone, lion)
25	D7	<calm and patronizing>You know Tony...it's not cool. Sorry	D7 criticizes D6

Turn 12

D7 patronizes D5 lack of knowledge

D5 Frame	D6 Frame	D7 Frame	Comments
Constraint: <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) Location: <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(P) Rules ↯feed(animals)(C) Explicit Offer <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) Characters	Constraint: <ul style="list-style-type: none"> 3 min (C) ColleagueFriends(D5,6,7) Location: <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ↯feed(animals)(C) Explicit Offer <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) Characters	Constraint: <ul style="list-style-type: none"> 3 min (C) ColleagueFriends(D5,6,7) Location: <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ↯feed(animals)(C) Explicit Offer <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) Characters	Tilt

31

<ul style="list-style-type: none"> D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(P) D7 <ul style="list-style-type: none"> Against feeding animals (C) Relations <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Friend(D5, D7) <ul style="list-style-type: none"> Status = +D7 Friend(D6, D6) <ul style="list-style-type: none"> Status = Props <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D6 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 	<ul style="list-style-type: none"> D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) D7 <ul style="list-style-type: none"> Against feeding animals (C) Relations <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Friend(D5, D7) <ul style="list-style-type: none"> Status = +D7 Friend(D6, D6) <ul style="list-style-type: none"> Status = Props <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) Snowcones(C) <ul style="list-style-type: none"> Owner D6 Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) 	<ul style="list-style-type: none"> D5 <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) D7 <ul style="list-style-type: none"> Against feeding animals (C) Relations <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Friend(D5, D7) <ul style="list-style-type: none"> Status = +D7 Friend(D6, D6) <ul style="list-style-type: none"> Status = Props <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D6 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) 	
---	--	--	--

32

<ul style="list-style-type: none"> • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 	<ul style="list-style-type: none"> • Popcons (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(P) 	<ul style="list-style-type: none"> ◦ Owner D7 	
--	---	--	--

Turn 13

D5 tries offer the lion in front stage but the others don't seem to be interested or noticing

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(P) ◦ Rules ◦ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) <p>Characters</p> <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(P) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) <p>Relations</p>	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) <p>Characters</p> <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) <p>Relations</p>	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) <p>Characters</p> <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(P) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) 	Tilt

33

<ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +D7 • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D6 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 	<ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +D7 • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D6 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(PP) 	<ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +D7 • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D6 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 	
---	--	--	--

Turn 13

D5 reacts and seems to balance status. They're just arguing to see who takes the last word....

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) 	Tilt

34

<p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(P) ◦ Rules ◦ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) <p>Characters</p> <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(P) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +D5 • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 	<p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) <p>Characters</p> <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +D5 • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 	<p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) <p>Characters</p> <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(P) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +D5 • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) 	
---	---	--	--

35

<ul style="list-style-type: none"> ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D6 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 	<ul style="list-style-type: none"> ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D6 ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(P) 	<ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D6 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 	
---	--	---	--

Turn 14

D5 reacts and seems to balance status. They're just arguing to see who takes the last word....

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(P) ◦ Rules ◦ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) <p>Characters</p> <ul style="list-style-type: none"> • D5 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) <p>Characters</p> <ul style="list-style-type: none"> • D5 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) <p>Characters</p> <ul style="list-style-type: none"> • D5 	Tilt

36

<ul style="list-style-type: none"> o Name Ted(R) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(P) <ul style="list-style-type: none"> • D7 <ul style="list-style-type: none"> o Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +D5 • Friend(D6, D6) <ul style="list-style-type: none"> o Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D6 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) 	<ul style="list-style-type: none"> o Name Ted(R) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(R) <ul style="list-style-type: none"> • D7 <ul style="list-style-type: none"> o Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +D5 • Friend(D6, D6) <ul style="list-style-type: none"> o Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D6 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) 	<ul style="list-style-type: none"> o Name Ted(P) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(R) <ul style="list-style-type: none"> • D7 <ul style="list-style-type: none"> o Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +D5 • Friend(D6, D6) <ul style="list-style-type: none"> o Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D6 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 	
--	--	---	--

37

o Owner D7	o Owner D7		
	• A particular lion(P)		

Turn 15

Offer(D6, schuk(snowcone, lion)) AddElement(Lion)

I'm making this OFFER you see that lion right there?!!! Any other type of people would go "yeah" even if they might not know what I'm thinking about. See that lion right there? He's dead, he's alive, he's a child, he's a baby... he already got on his mind to go somewhere else... but I actually wanted to bring that back around because it sort of got dropped and everything.

"That Lion right there." Since

I', gonna show you we can throw stuff at that lion over there....

Notes: It seems almost as if D6 wanted to take an active role in the D5/D7 argument. He had to take sides to be a part of the action, Speculations about D6 thought:

- o he couldn't be active by not doing an action such as by not throwing food, he would have to do something to prevent the others from throwing food such as taking the food from their hands....
- o He's character was very decided about his positions and actions, the weaker side of the debate was D7 because he was preventing them to amuse themselves, by being to patronizing.
- o It also might happen that D7 status was lower and more vulnerable...

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(P) o Rules o ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) 	Tilt

38

Characters <ul style="list-style-type: none"> D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(P) D7 <ul style="list-style-type: none"> Against feeding animals (C) Relations <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Friend(D5, D7) <ul style="list-style-type: none"> Status = +D5 Friend(D6, D6) <ul style="list-style-type: none"> Status = Props <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D6 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (R) 	Characters <ul style="list-style-type: none"> D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) D7 <ul style="list-style-type: none"> Against feeding animals (C) Relations <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Friend(D5, D7) <ul style="list-style-type: none"> Status = +D5 Friend(D6, D6) <ul style="list-style-type: none"> Status = Props <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D6 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) 	Characters <ul style="list-style-type: none"> D5 <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) D7 <ul style="list-style-type: none"> Against feeding animals (C) Relations <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Friend(D5, D7) <ul style="list-style-type: none"> Status = +D5 Friend(D6, D6) <ul style="list-style-type: none"> Status = Props <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D6 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) 	
--	--	--	--

39

<ul style="list-style-type: none"> <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	<ul style="list-style-type: none"> <ul style="list-style-type: none"> SuicideS(C) Popcons (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	<ul style="list-style-type: none"> Owner D7 A particular lion(C) 	
---	--	--	--

Turn 16 to 20

D7 and D5 explore their arguments over D6 offer. D6 explores his action by portraying he knows his action is against the rules..

Why doesn't D6 throw the snowcone immediately, is D6 riding the conflict is he choosing sides....it creates suspense....

D6 know he's breaking a rule....

D7 and D5 present arguments for their different appraisals of action Schuck() / Feed()

D5 supports Feed(lion)

D7 rejects Feed(lion)

D5 Frame	D6 Frame	D7 Frame	Comments
Constraint: <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) Location: <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(P) Rules ↪feed(animals)(C) Explicit Offer <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) Characters <ul style="list-style-type: none"> D5 	Constraint: <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) Location: <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ↪feed(animals)(C) Explicit Offer <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) Characters <ul style="list-style-type: none"> D5 	Constraint: <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) Location: <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ↪feed(animals)(C) Explicit Offer <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) Characters <ul style="list-style-type: none"> D5 	Tilt

40

<ul style="list-style-type: none"> o Name Ted(R) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(P) <ul style="list-style-type: none"> • D7 <ul style="list-style-type: none"> o Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +D5 • Friend(D6, D6) <ul style="list-style-type: none"> o Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D6 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) 	<ul style="list-style-type: none"> o Name Ted(R) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(R) <ul style="list-style-type: none"> • D7 <ul style="list-style-type: none"> o Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +D5 • Friend(D6, D6) <ul style="list-style-type: none"> o Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D6 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> o Name Ted(P) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(R) <ul style="list-style-type: none"> • D7 <ul style="list-style-type: none"> o Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +D5 • Friend(D6, D6) <ul style="list-style-type: none"> o Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D6 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	
--	---	---	--

41

<ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 		
--	--	--	--

Turn 21

D6 throws Snowcone to feed the Lion.

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(P) o Rules o ↵feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) <p>Characters</p> <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> o Name Ted(R) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(P) • D7 <ul style="list-style-type: none"> o Against feeding animals 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o ↵feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) <p>Characters</p> <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> o Name Ted(R) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(R) • D7 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o ↵feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) <p>Characters</p> <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> o Name Ted(P) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(R) • D7 	Tilt

42

<p>(C)</p> <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +D5 • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<p>◦ Against feeding animals (C)</p> <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +D5 • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<p>◦ Against feeding animals (C)</p> <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +D5 • Friend(D6, D6) <ul style="list-style-type: none"> ◦ Status = <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	
--	--	---	--

Turn 22

D7 disapproves D6 activity and shows his disapproval.

D5 Frame	D6 Frame	D7 Frame	Comments
Constraint:	Constraint:	Constraint:	Tilt

43

<ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(P) ◦ Rules ◦ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) <p>Characters</p> <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(P) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +D5 	<ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) <p>Characters</p> <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +D5 • Friend(D6, D7) 	<ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) <p>Characters</p> <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(P) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +D5 • Friend(D6, D7) 	
--	--	--	--

44

<ul style="list-style-type: none"> • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status =+D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ◦ Status =+D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ◦ Status =+D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	
---	---	--	--

Turn 23

D5 defends D6

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(P) ◦ Rules ◦ ↩feed(animals)(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↩feed(animals)(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↩feed(animals)(C) 	Tilt

45

<p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) <p>Characters</p> <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(P) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity + • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = ++D5 • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status =+D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 	<p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) <p>Characters</p> <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity + • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = ++D5 • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status =+D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) 	<p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) <p>Characters</p> <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(P) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity + • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = ++D5 • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status =+D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) 	
---	--	--	--

46

<ul style="list-style-type: none"> o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	
--	--	--	--

Turn 24

D6 mocks lion and offers lion eating snowcone.

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(P) o Rules o ↵feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o ↵feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o ↵feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) 	Tilt

47

<ul style="list-style-type: none"> • Lion eat(Snowcone)(R) • D6 mocking(Lion)(R) <p>Characters</p> <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> o Name Ted(R) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(P) • D7 <ul style="list-style-type: none"> o Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity + • Friend(D5, D7) <ul style="list-style-type: none"> o Status = ++D5 • Friend(D6, D7) <ul style="list-style-type: none"> o Status = ++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) 	<ul style="list-style-type: none"> • Lion eat(Snowcone)(P) • D6 mocking(Lion)(P) <p>Characters</p> <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> o Name Ted(R) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(R) • D7 <ul style="list-style-type: none"> o Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity + • Friend(D5, D7) <ul style="list-style-type: none"> o Status = ++D5 • Friend(D6, D7) <ul style="list-style-type: none"> o Status = ++D6 o Status = ++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (P) 	<ul style="list-style-type: none"> • Lion eat(Snowcone)(R) • D6 mocking(Lion)(R) <p>Characters</p> <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> o Name Ted(P) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(R) • D7 <ul style="list-style-type: none"> o Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity + • Friend(D5, D7) <ul style="list-style-type: none"> o Status = ++D5 • Friend(D6, D7) <ul style="list-style-type: none"> o Status = ++D6 o Status = ++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) 	
---	---	---	--

48

<ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (P) ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	
--	---	--	--

Turn 25

D7 disapproves D6 activity, which confirms that it was received. And diminishes D7 and D6 affinity, and increases D7 status..

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(P) ◦ Rules ◦ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) <p>Characters</p> <ul style="list-style-type: none"> • D5 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(P) • D6 mocking(Lion)(C) <p>Characters</p> <ul style="list-style-type: none"> • D5 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) <p>Characters</p> <ul style="list-style-type: none"> • D5 	Tilt

49

<ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(P) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity + • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = ++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status = ++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) 	<ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity + • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = ++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status = +D6 ◦ Status = ++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) 	<ul style="list-style-type: none"> ◦ Name Ted(P) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity + • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = ++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status = +D6 ◦ Status = ++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) 	
--	---	---	--

50

<ul style="list-style-type: none"> • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> • Popcons (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	
--	---	--	--

Turn 26 to 29

26	D6	<calm and assertive> I don't have a wife, and I don't have a children <u>like you've got!</u>	STRONG OFFER Has(D7, wife) Has(D7, Children) D6 appraise(Wife)<<< D6 appraise(Children)<<<
27	D5	<raising hand towards D6> Yeah! Hi 5 Dude!	D5 agrees with D6 D5(affinity,D6) >>> D6(affinity,D5) >>>
28	D5	<hi 5ing D6> Yeah! I wish I didn't have a wife and children	Has(D5, Wife) Has(D5, Children) D5 appraise(Wife)<<< D5 appraise(Children)<<<
29	D6	<pointing at D7> <patronizing> Yeah and I wish you didn't have that wife and those children...	D6 appraise(D5.wife)<<< D6 appraise(D5.children)<<<

Turn 26

This is a clear tilt offer because D5 is deliberately trying to move focus away from the animal issues to D7's personal life..
D6 "What I was thinking is that D7 is a sad sad no fun kind of guy, his character already is one who says you can't throw shit, "oh now I've got a snowcone... you now" so what's the reason of all this torment? probably in college we all had a good time, but now for some reason his life is muddy, so I decided to endow him with a horrible life. Then a thing happens in improv called the game ofn the scene (game -> repetitive behavior) randy and I agree that D7 wife sucks..... the cake is the real story that the audience is going to understands ... the emotional side of it is that his wife abuses him..."

D5 Frame	D6 Frame	D7 Frame	Comments
Constraint: • 3 min (C)	Constraint: • 3 min (C)	Constraint: • 3 min (C)	Tilt Offer

51

<ul style="list-style-type: none"> • CollegeFriends(D5,6,7) Location: <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(P) ◦ Rules ◦ ↯feed(animals)(C) Explicit Offer <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) Activity <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) Characters <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(P) ◦ Has bad wife(R) ◦ Has bad children(R) ◦ • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(R) ◦ Has bad children(R) 	<ul style="list-style-type: none"> • CollegeFriends(D5,6,7) Location: <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↯feed(animals)(C) Explicit Offer <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) Activity <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(P) • D6 mocking(Lion)(C) Characters <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(P) ◦ Has bad children(P) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(P) ◦ Has bad children(P) Relations	<ul style="list-style-type: none"> • CollegeFriends(D5,6,7) Location: <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↯feed(animals)(C) Explicit Offer <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) Activity <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) Characters <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(P) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(R) ◦ Has bad children(R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(R) ◦ Has bad children(R) Relations	
--	---	---	--

52

Relations <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity + • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = ++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status = ++D6 Props <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity + • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = ++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status = +D6 ◦ Status = ++D6 Props <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity + • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = ++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status = +D6 ◦ Status = ++D6 Props <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	
---	---	---	--

Turn 27 and 28

D5 confirms content of D6 offer in spite the fact that he initially understood it as an endowment exclusively addressed to D7. This also increases D6 and D5 affinities.

53

D5 Frame	D6 Frame	D7 Frame	Comments
Constraint: <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) Location: <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(P) ◦ Rules ◦ ↯feed(animals)(C) Explicit Offer <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) Activity <ul style="list-style-type: none"> • D6 Feed(Lion, Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) Characters <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(P) ◦ Has bad wife(C) ◦ Has bad children(C) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(C) 	Constraint: <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) Location: <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↯feed(animals)(C) Explicit Offer <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) Activity <ul style="list-style-type: none"> • D6 Feed(Lion, Snowcone)(C) • Lion eat(Snowcone)(P) • D6 mocking(Lion)(C) Characters <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(C) 	Constraint: <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) Location: <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↯feed(animals)(C) Explicit Offer <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) Activity <ul style="list-style-type: none"> • D6 Feed(Lion, Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) Characters <ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(P) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(C) 	Tilt Offer

54

<ul style="list-style-type: none"> Has bad children(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = ++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status = ++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	<ul style="list-style-type: none"> Has bad children(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = ++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status = +D6 Status = ++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) Popcons (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	<ul style="list-style-type: none"> Has bad children(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = ++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status = +D6 Status = ++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	
--	--	---	--

55

Turn 29

D6 agrees with D5 increasing Affinity and focuses on D7's wife as a problem!

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(P) Rules ←feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion, Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) <p>Characters</p> <ul style="list-style-type: none"> D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(P) Has bad wife(C) Has bad children(C) D7 <ul style="list-style-type: none"> Against feeding animals 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ←feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion, Snowcone)(C) Lion eat(Snowcone)(P) D6 mocking(Lion)(C) <p>Characters</p> <ul style="list-style-type: none"> D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ←feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion, Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) <p>Characters</p> <ul style="list-style-type: none"> D5 <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) 	Tilt Offer

56

(C) o Has bad wife(CC) o Has bad children(CC)	o Against feeding animals (C) o Has bad wife(CC) o Has bad children(CC)	o Has bad wife(CC) o Has bad children(CC)	
Relations <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status = ++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status = ++D6 	Relations <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status = ++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status = ++D6 	Relations <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status = ++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status = ++D6 	
Props <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	Props <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	Props <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	

57

Consistency and Emergent Frame

Are almost exactly the same right now because every data except for the lion is eating snowcone is confirmed and there are no inconsistencies.

An interesting aspect to notice is that D7 knows that he has a bad wife, but at the same time he can chose not to agree with what the others present as a fact

Turn 30 to 34

Turn	Player	Action	Comments
31	D7	What the Fuck dudes <complaining> Why are you trashing my wife? We do this once a year,, <D5 faces D7 in the eyes> That's right... (Like throw it out) we get together in a funny little place that I never get to pick <D5 acknowledges with superiority> and you <UNCOMPREHENSIBLE WORDS> come her and you pick on my WIFE! <Pause> <Tries to calm speech> I Can't help it, that she's the most beautifull girl in the world....	D7 offers action(insult(D7.wife)) Context (location picked by D5 and D6) (frequency once a year) D7 never picks location D7 status<<<
32	D5	<D5 looks back from D7 to front stage> <unconvinced and ironic> oh yeah... She's very, very..... very....	D5 accepts the offer of insulting(D7.wife)
33	D7	Are you calling my wife ugly?	D7 offers D7.wife(ugly)=True
34	D5	D5 <moving his head to the sides, with a mocking smile> I'm not calling you're wife ugly.	D5 accepts offer D5 respects D7

58

Turn 30 to 31

D7 “The scene is now(2:21) shifting into something else, where no longer dealing with the lion and where dealing with my wife which he doesn’t like, I’m thinking should I not like my wife or should I like my wife? And that’s going to bas my opinion about what he said.”

Game within the scene – “I’m gonna be the outsider from which every opinion differs”

D7 offers D6 and D5 are insulting his wife, which is accepted by D6 “yeah”, at the same time he provides context about how his opinios are never considered by his friends, “I never get to pick”

D7 complements Wife just to oppose the insults of the other two players.

D5 Frame	D6 Frame	D7 Frame	Comments
Constraint: <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) Location: <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(P) Rules ↔feed(animals)(C) Explicit Offer <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) Activity <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(R) 	Constraint: <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) Location: <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ↔feed(animals)(C) Explicit Offer <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) Activity <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(P) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(R) 	Constraint: <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) Location: <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ↔feed(animals)(C) Explicit Offer <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) Activity <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(P) 	Tilt Offer

59

Characters	Characters	Characters	
<ul style="list-style-type: none"> D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(P) Has bad wife(C) Has bad children(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(R) Relations <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =++D6 Status =+++D6 Props <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup? (R) 	<ul style="list-style-type: none"> D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(R) Relations <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status =+++D6 Props <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) 	<ul style="list-style-type: none"> D5 <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(P) Relations <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status=+++D6 Props <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) 	

60

<ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	
--	---	--	--

Turn 32

D6 accepts the offer of insulting (D7.wife)

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(P) ◦ Rules ◦ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) 	Tilt Offer

61

<ul style="list-style-type: none"> • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ ↯findW7Wife(beautiful)(R) • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(P) ◦ Has bad wife(C) ◦ Has bad children(C) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +++D5 • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +++D5 	<ul style="list-style-type: none"> • Lion eat(Snowcone)(P) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ ↯findW7Wife(beautiful)(P) • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +++D5 • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status = +D6 	<ul style="list-style-type: none"> • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ ↯findW7Wife(beautiful)(R) • D5 <ul style="list-style-type: none"> ◦ Name Ted(P) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status =Status = +++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status = +D6 	
---	---	---	--

62

<ul style="list-style-type: none"> o Affinity - • Friend(D6, D7) o Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> o Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> o Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	
---	---	---	--

Turn 33

D7 confirms that D6 doesn't find D7.Wife the most beautiful girl in the world. And offers D6 finds D7.wife ugly

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(P) o Rules 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules 	Tilt Offer

63

<ul style="list-style-type: none"> o ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> o ↯findW7Wife(beautiful)(C) o FindsD7Wife(Ugly)(R) • D5 <ul style="list-style-type: none"> o Name Ted(R) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(P) o Has bad wife(C) o Has bad children(C) • D7 <ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC) o Has bad children(CC) o FindsWife(MostBeautif 	<ul style="list-style-type: none"> o ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(P) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> o ↯findW7Wife(beautiful)(C) o FindsD7Wife(Ugly)(R) • D5 <ul style="list-style-type: none"> o Name Ted(R) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(R) o Has bad wife(C) o Has bad children(C) • D7 <ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC) o Has bad children(CC) o FindsWife(MostBeautiful)(C) 	<ul style="list-style-type: none"> o ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> o ↯findW7Wife(beautiful)(C) o FindsD7Wife(Ugly)(P) • D5 <ul style="list-style-type: none"> o Name Ted(P) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(R) o Has bad wife(C) o Has bad children(C) • D7 <ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC) o Has bad children(CC) o FindsWife(MostBeautiful)(C) 	
--	--	--	--

64

<ul style="list-style-type: none"> ul)(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status =+D6 ◦ Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status =Status = +++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status =+D6 ◦ Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	
--	--	--	--

65

Turn 34

D6 “I was looking for the right word, I could couldn’t say his wife wasn’t ugly because of my character,.... Without saying what she is I’m saying what she is because I can’t deny it or support what you said.....”

D6 confirms that he finds D7.Wife ugly

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(P) ◦ Rules ◦ ↵feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↵feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(P) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↵feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(P) ◦ Occupation related with animals (C) 	<p>Tilt Offer</p>

66

<ul style="list-style-type: none"> o Zoologist(C) o ExZoology Student (C) o Animal Abuser(P) o Has bad wife(C) o Has bad children(C) <ul style="list-style-type: none"> • D7 <ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC) o Has bad children(CC) o FindsWife(MostBeautiful)(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) 	<ul style="list-style-type: none"> o Zoologist(C) o ExZoology Student (C) o Animal Abuser(R) o Has bad wife(C) o Has bad children(C) <ul style="list-style-type: none"> • D7 <ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC) o Has bad children(CC) o FindsWife(MostBeautiful)(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+D6 o Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 	<ul style="list-style-type: none"> o Zoologist(C) o ExZoology Student (C) o Animal Abuser(R) o Has bad wife(C) o Has bad children(C) <ul style="list-style-type: none"> • D7 <ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC) o Has bad children(CC) o FindsWife(MostBeautiful)(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status =Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+D6 o Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) 	
---	---	--	--

67

<ul style="list-style-type: none"> o Owner D7 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) <ul style="list-style-type: none"> • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) <ul style="list-style-type: none"> • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> o Owner D7 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) <ul style="list-style-type: none"> • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	
--	--	--	--

Turn 35

D5 offers D7.Wife(Hansome), he might have been locked somewhere in between not offending and not lying to his friend.

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(P) o Rules o ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) <p>Characters</p>	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(P) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) <p>Characters</p>	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) <p>Characters</p>	Tilt Offer

68

<ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(P) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (P) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 	<ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status =+D6 ◦ Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) 	<ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(P) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status =Status = +++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status =+D6 ◦ Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 	
---	---	---	--

69

<ul style="list-style-type: none"> ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	
--	--	---	--

Turn 36

D7 offers D5 finds D7.Wife Manish

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(P) ◦ Rules ◦ ↵feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↵feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↵feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) 	Tilt Offer

70

<ul style="list-style-type: none"> Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(P) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (P) D7.Wife(Manish)(R) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) 	<ul style="list-style-type: none"> Lion eat(Snowcone)(P) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(R) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - 	<ul style="list-style-type: none"> Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(P) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =Status = +++D5 Affinity - 	
--	--	--	--

71

<ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	<ul style="list-style-type: none"> Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) Popcons (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	<ul style="list-style-type: none"> Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	
---	--	--	--

Turn 37

D5 confirms Manish and endows D7.Wife with a cleft chin

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(P) 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) 	Tilt Offer

72

<ul style="list-style-type: none"> Rules \negfeed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion, Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(P) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (P) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(P) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) 	<ul style="list-style-type: none"> rules \negfeed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion, Snowcone)(C) Lion eat(Snowcone)(P) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(R) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) 	<ul style="list-style-type: none"> rules \negfeed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion, Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(R) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) 	
--	--	--	--

73

<ul style="list-style-type: none"> Has bad children(CC) FindsWife(MostBeautiful)(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status = +++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	<ul style="list-style-type: none"> FindsWife(MostBeautiful)(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status = +D6 Status = +++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	<ul style="list-style-type: none"> FindsWife(MostBeautiful)(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status = +D6 Status = +++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	
--	---	--	--

74

Turn 38

D7 confirms cleft chin and explores it trying to be funny

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(P) Rules ←feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion, Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ←feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion, Snowcone)(C) Lion eat(Snowcone)(P) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ←feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion, Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) 	Tilt Offer

75

<ul style="list-style-type: none"> Animal Abuser(P) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (P) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) <ul style="list-style-type: none"> D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status = +++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) 	<ul style="list-style-type: none"> Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) <ul style="list-style-type: none"> D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status = +D6 Status = +++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (P) 	<ul style="list-style-type: none"> Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) <ul style="list-style-type: none"> D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> Use(chin, mostard) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status = +D6 Status = +++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) 	
---	---	--	--

76

<ul style="list-style-type: none"> ○ Owner Lion(C) ○ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ○ Owner D7 ○ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ○ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ○ Owner D7 ○ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> ○ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ○ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ○ Owner D7 ○ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ○ Owner D7 • A particular lion(C) 	
---	---	--	--

Turn 39

D5 offers D7.Wife taller than D7

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ○ Lions in front(P) ○ Rules ○ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ○ Lions in front(C) ○ rules ○ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(P) • D6 mocking(Lion)(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ○ Lions in front(C) ○ rules ○ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) 	Tilt Offer

77

<ul style="list-style-type: none"> • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ○ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ○ Name Ted(R) ○ Occupation related with animals (C) ○ Zoologist(C) ○ ExZoology Student (C) ○ Animal Abuser(P) ○ Has bad wife(C) ○ Has bad children(C) ○ D7.Wife(Hansome) (P) ○ D7.Wife(Manish)(C) ○ D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> ○ Against feeding animals (C) ○ Has bad wife(CC) ○ Has bad children(CC) ○ FindsWife(MostBeautiful)(C) ○ Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ○ Height<D7.Wife.Height(P) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ○ Status = 	<ul style="list-style-type: none"> • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ○ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ○ Name Ted(R) ○ Occupation related with animals (C) ○ Zoologist(C) ○ ExZoology Student (C) ○ Animal Abuser(R) ○ Has bad wife(C) ○ Has bad children(C) ○ D7.Wife(Hansome) (R) ○ D7.Wife(Manish)(C) ○ D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> ○ Against feeding animals (C) ○ Has bad wife(CC) ○ Has bad children(CC) ○ FindsWife(MostBeautiful)(C) ○ Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ○ Height<D7.Wife.Height(R) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ○ Status = ○ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ○ Status = +++D5 	<ul style="list-style-type: none"> • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ○ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ○ Name Ted(P) ○ Occupation related with animals (C) ○ Zoologist(C) ○ ExZoology Student (C) ○ Animal Abuser(R) ○ Has bad wife(C) ○ Has bad children(C) ○ D7.Wife(Hansome) (R) ○ D7.Wife(Manish)(C) ○ D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> ○ Against feeding animals (C) ○ Has bad wife(CC) ○ Has bad children(CC) ○ FindsWife(MostBeautiful)(C) ○ Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ○ Height<D7.Wife.Height(R) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ○ Status = ○ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ○ Status =Status = +++D5 	
--	---	---	--

78

<ul style="list-style-type: none"> o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+D6 o Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+D6 o Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	
--	--	--	--

Turn 40

D6 confirms D7.Wife taller than D7

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) 	Tilt Offer

79

<ul style="list-style-type: none"> o Lions in front(P) o Rules o ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> o FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> o Name Ted(R) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(P) o Has bad wife(C) o Has bad children(C) o D7.Wife(Hansome) (P) o D7.Wife(Manish)(C) o D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> o Against feeding animals (C) 	<ul style="list-style-type: none"> o Lions in front(C) o rules o ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(P) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> o FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> o Name Ted(R) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(R) o Has bad wife(C) o Has bad children(C) o D7.Wife(Hansome) (R) o D7.Wife(Manish)(C) o D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC) 	<ul style="list-style-type: none"> o Lions in front(C) o rules o ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> o FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> o Name Ted(P) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(R) o Has bad wife(C) o Has bad children(C) o D7.Wife(Hansome) (R) o D7.Wife(Manish)(C) o D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC) 	
--	--	--	--

80

<ul style="list-style-type: none"> Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) 	<ul style="list-style-type: none"> Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) Popcons (C) <ul style="list-style-type: none"> Owner D7 	<ul style="list-style-type: none"> Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 	
--	---	---	--

81

<ul style="list-style-type: none"> Owner D7 A particular lion(C) 	<ul style="list-style-type: none"> A particular lion(C) 	<ul style="list-style-type: none"> A particular lion(C) 	
--	--	--	--

Turn 41

Height differences starts getting the focus of the action. D7 plays he doesn't care unconvincingly... Which places himself in the opposite extreme of Height Differences..

Turn 42

D5 endows D7.Wife as a Mamoth Shewoman / something evil....

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(P) Rules ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) <p>Characters</p>	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(P) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) <p>Characters</p>	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) <p>Characters</p>	Tilt Offer

82

<ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(P) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (P) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) ◦ Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ◦ Height<D7.Wife.Height(C) ◦ Wife=MammothSheWoman (P) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +++D5 ◦ Affinity - • Friend(D5, D7) 	<ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (R) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) ◦ Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ◦ Height<D7.Wife.Height(C) ◦ Wife=MammothSheWoman (R) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +++D5 ◦ Affinity - • Friend(D6, D7) 	<ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(P) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (R) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) ◦ Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ◦ Height<D7.Wife.Height(C) ◦ Wife=MammothSheWoman (R) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status =Status = +++D5 ◦ Affinity - • Friend(D6, D7) 	
---	---	---	--

83

<ul style="list-style-type: none"> ◦ Status = +++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ◦ Status =+D6 ◦ Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ◦ Status =+D6 ◦ Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	
---	--	---	--

Turn 43

D7 confirms mammoth Shewoman and offers he might like it rug, he might be maso,, ,

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(P) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) 	Tilt Offer

84

<ul style="list-style-type: none"> Rules ¬feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion, Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (R) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(P) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (P) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 <ul style="list-style-type: none"> Against feeding animals 	<ul style="list-style-type: none"> rules ¬feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion, Snowcone)(C) Lion eat(Snowcone)(P) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (R) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) 	<ul style="list-style-type: none"> rules ¬feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion, Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (P) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) 	
--	--	--	--

85

<ul style="list-style-type: none"> (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (P) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 	<ul style="list-style-type: none"> Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (R) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =+++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) 	<ul style="list-style-type: none"> Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner D7 Syrup? (R) 	
--	---	---	--

86

<ul style="list-style-type: none"> ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> • Popcons (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	
---	---	---	--

Turn 44

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(P) ◦ Rules ◦ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (R) • D7.Wife beatEmotionally(D7, D7 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(P) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (R) • D7.Wife beatEmotionally(D7, D7 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (P) • D7.Wife beatEmotionally(D7, D7 	Tilt Offer

87

<p>.Wife)(P)</p> <ul style="list-style-type: none"> • D7.Wife beatPhysically(D7, D7.Wife)(P) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(P) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (P) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) ◦ Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ◦ Height<D7.Wife.Height(C) ◦ Wife=MammothSheWoman (P) 	<p>.Wife)(R)</p> <ul style="list-style-type: none"> • D7.Wife beatPhysically(D7, D7.Wife)(R) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (R) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) ◦ Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ◦ Height<D7.Wife.Height(C) ◦ Wife=MammothSheWoman (R) ◦ Victim(R) <p>Relations</p>	<p>.Wife)(R)</p> <ul style="list-style-type: none"> • D7.Wife beatPhysically(D7, D7.Wife)(R) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(P) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (R) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) ◦ Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ◦ Height<D7.Wife.Height(C) ◦ Wife=MammothSheWoman (C) ◦ Victim(R) <p>Relations</p>	
--	--	--	--

88

<ul style="list-style-type: none"> ◦ Victim(P) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status =+D6 ◦ Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status =Status = +++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status =+D6 ◦ Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	
---	---	---	--

Turn 45

D7 seems to accept offer of being a Victim of physical abuse.

89

D7 “right there it’s the turn away, I’ like lets not talk about this, and they know, they follow me, it’s clearly like (whining voice) ‘I don’t know. How do abused people act like?... I don’t know...’ ”

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(P) ◦ Rules ◦ ↩feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (R) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↩feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(P) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (R) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↩feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (P) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 	Tilt Offer

90

<ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(P) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (P) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) ◦ Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ◦ Height<D7.Wife.Height(C) ◦ Wife=MammothSheWoman (P) ◦ Victim(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +++D5 	<ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (R) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) ◦ Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ◦ Height<D7.Wife.Height(C) ◦ Wife=MammothSheWoman (R) ◦ Victim(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +++D5 ◦ Affinity - • Friend(D6, D7) 	<ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(P) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (R) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) ◦ Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ◦ Height<D7.Wife.Height(C) ◦ Wife=MammothSheWoman (C) ◦ Victim(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status =Status = +++D5 ◦ Affinity - • Friend(D6, D7) 	
--	--	--	--

91

<ul style="list-style-type: none"> ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ◦ Status =+D6 ◦ Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ◦ Status =+D6 ◦ Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	
---	--	---	--

Turn 46

D5 proposes activity of D7 exposing his abuses...

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(P) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) 	Tilt Offer

92

<ul style="list-style-type: none"> Rules \negfeed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion, Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (R) D7.Wife beatEmotionally(D7, D7.Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(P) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(P) Has bad wife(C) Has bad children(C) 	<ul style="list-style-type: none"> rules \negfeed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion, Snowcone)(C) Lion eat(Snowcone)(P) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (R) D7.Wife beatEmotionally(D7, D7.Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(R) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) 	<ul style="list-style-type: none"> rules \negfeed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion, Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (P) D7.Wife beatEmotionally(D7, D7.Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(R) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) 	
--	--	--	--

93

<ul style="list-style-type: none"> D7.Wife(Hansome) (P) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) <ul style="list-style-type: none"> D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (P) Victim(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 	<ul style="list-style-type: none"> D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) <ul style="list-style-type: none"> D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (R) Victim(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) 	<ul style="list-style-type: none"> D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) <ul style="list-style-type: none"> D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (C) Victim(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) 	
--	---	--	--

94

<ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D7 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	
---	--	---	--

Turn 47

D7 acts as a victim, trying to avoid exposure by transferring focus, which confirms he received the early offer to expose the situation
Some possible speculations around this attitude:

- It doesn't seem like a real offer for a new activity because D7 is just playing a character and knows that his partners would change the subject.
- Nevertheless it leaves an opportunity for a change on story focus to a less interesting subject, so we might consider it as a very weak offer.
- It also might happen that the player is making a real offer that drives from his character behavior, but his character status is so low at this point that his suggestions for scope transfer can be completely disregarded by the other characters.

D5 Frame	D6 Frame	D7 Frame	Comments
Constraint: <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) Location: <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(P) ◦ Rules ◦ ¬feed(animals)(C) 	Constraint: <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) Location: <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ¬feed(animals)(C) 	Constraint: <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) Location: <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ¬feed(animals)(C) 	Tilt Offer

95

Explicit Offer <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) Activity <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (R) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> ◦ D7 ChangeSubject(R) Characters <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(P) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (P) 	Explicit Offer <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) Activity <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(P) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (R) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> ◦ D7 ChangeSubject(R) Characters <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (R) 	Explicit Offer <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) Activity <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (P) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> ◦ D7 ChangeSubject(P) Characters <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(P) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (R) 	
---	---	---	--

96

<ul style="list-style-type: none"> o D7.Wife(Manish)(C) o D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC) o Has bad children(CC) o FindsWife(MostBeautiful)(C) o Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) o Height<D7.Wife.Height(C) o Wife=MammothSheWoman (P) o Victim(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) 	<ul style="list-style-type: none"> o D7.Wife(Manish)(C) o D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC) o Has bad children(CC) o FindsWife(MostBeautiful)(C) o Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) o Height<D7.Wife.Height(C) o Wife=MammothSheWoman (R) o Victim(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+D6 o Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (P) 	<ul style="list-style-type: none"> o D7.Wife(Manish)(C) o D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC) o Has bad children(CC) o FindsWife(MostBeautiful)(C) o Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> ▪ Use(chin, mostard) o Height<D7.Wife.Height(C) o Wife=MammothSheWoman (C) o Victim(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status =Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+D6 o Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) 	
---	--	--	--

97

<ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D7 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	
---	---	--	--

Turn 48

D5 rejects change of subject and reinforces his position over feeding animals by taking D7's food and throwing it away to the lions. D5 gestures weren't very clear as to where the objects were thrown. And D7 interpreted it as the floor (littering).

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(P) o Rules o ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(P) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) 	Tilt Offer

98

<ul style="list-style-type: none"> D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (R) D7.Wife beatEmotionally(D7, D7.Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED FeedLions(PopCorn)(P) FeedLions(Snowcone)(P) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(P) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (P) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) 	<ul style="list-style-type: none"> D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (R) D7.Wife beatEmotionally(D7, D7.Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED FeedLions(PopCorn)(R) FeedLions(Snowcone)(R) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) 	<ul style="list-style-type: none"> D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (P) D7.Wife beatEmotionally(D7, D7.Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED Littering(PopCorn)(R) Littering(Snowcone)(R) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC)
--	--	--

99

<ul style="list-style-type: none"> Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (P) Victim(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lions(P) 	<ul style="list-style-type: none"> Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (R) Victim(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =+++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lions(R) Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) 	<ul style="list-style-type: none"> Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (C) Victim(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =Status =+++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner none(R) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C)
--	--	---

100

<ul style="list-style-type: none"> ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> • Popcons (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	
---	---	--	--

Turn 49

D7 confirms Littering as his perceived activity. And tries to switch the scope of the action.

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(P) ◦ Rules ◦ ↩feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (R) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↩feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(P) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (R) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↩feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (P) 	Tilt Offer

101

<ul style="list-style-type: none"> • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> ◦ D7 ChangeSubject(C) REJECTED • FeedLions(Popcorn) <ul style="list-style-type: none"> ◦ Littering(Popcorn)(A) • FeedLions(Snowcone) <ul style="list-style-type: none"> ◦ Littering(Popcorn)(A) • CommitsCrimes(D5)(R) • CommitsCrimes(D6)(R) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(P) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (P) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) 	<ul style="list-style-type: none"> • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> ◦ D7 ChangeSubject(C) REJECTED • FeedLions(Popcorn) <ul style="list-style-type: none"> ◦ Littering(Popcorn)(A) • FeedLions(Snowcone) <ul style="list-style-type: none"> ◦ Littering(Popcorn)(A) • CommitsCrimes(D5)(R) • CommitsCrimes(D6)(R) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (R) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) 	<ul style="list-style-type: none"> • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> ◦ D7 ChangeSubject(C) REJECTED • Littering(PopCorn)(C) • Littering(Snowcone)(C) • CommitsCrimes(D5)(P) • CommitsCrimes(D6)(P) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(P) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (R) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) ◦ Betty Davis(CleftChin) (P) 	
---	---	--	--

102

<ul style="list-style-type: none"> Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (P) Victim(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lions <ul style="list-style-type: none"> none(A) 	<ul style="list-style-type: none"> FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (R) Victim(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lions <ul style="list-style-type: none"> none(A) Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) 	<ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) <ul style="list-style-type: none"> Wife=MammothSheWoman (C) Victim(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner none(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 	
--	---	--	--

103

<ul style="list-style-type: none"> Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	<ul style="list-style-type: none"> Popcons (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	<ul style="list-style-type: none"> A particular lion(C) 	
---	---	--	--

Turn 50

D5 offers to save D7. D6 positions himself in a position of committing crimes and saving D7.

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(P) Rules ←feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexualy(D7, D7.Wife) (R) 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ←feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(P) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexualy(D7, D7.Wife) (R) 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ←feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexualy(D7, D7.Wife) (P) 	Tilt Offer

104

<ul style="list-style-type: none"> D7.Wife beatEmotionally(D7, D7.Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) FeedLions(Popcorn) <ul style="list-style-type: none"> Littering(Popcorn)(A) FeedLions(Snowcone) <ul style="list-style-type: none"> Littering(Popcorn)(A) CommitsCrimes(D5)(C) CommitsCrimes(D6)(C) Save(D7)(P) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(P) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (P) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) 	<ul style="list-style-type: none"> D7.Wife beatEmotionally(D7, D7.Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) FeedLions(Popcorn) <ul style="list-style-type: none"> Littering(Popcorn)(A) FeedLions(Snowcone) <ul style="list-style-type: none"> Littering(Popcorn)(A) CommitsCrimes(D5)(C) CommitsCrimes(D6)(C) Save(D7)(P) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) 	<ul style="list-style-type: none"> D7.Wife beatEmotionally(D7, D7.Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) Littering(PopCorn)(C) Littering(Snowcone)(C) CommitsCrimes(D5)(C) CommitsCrimes(D6)(C) Save(D7)(P) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C)
---	---	--

105

<ul style="list-style-type: none"> Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (P) Victim(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lions 	<ul style="list-style-type: none"> Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (R) Victim(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lions none(A) Syrup? (P) 	<ul style="list-style-type: none"> Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (C) Victim(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner none(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7
---	--	--

106

<ul style="list-style-type: none"> ▪ none(A) ○ Syrup? (R) ▪ SuicideS(C) • Popcorns (C) ○ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) ○ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> • A particular lion(C) 	
--	---	--	--

Turn 51

D6 positions himself in a position of committing crimes and saving D7.

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ○ Lions in front(P) ○ Rules ○ ↩feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (R) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ○ Lions in front(C) ○ rules ○ ↩feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(P) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (R) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ○ Lions in front(C) ○ rules ○ ↩feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (P) 	Tilt Offer

107

<ul style="list-style-type: none"> • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> ○ D7 ChangeSubject(C) REJECTED • FeedLions(Popcorn) <ul style="list-style-type: none"> ○ Littering(Popcorn)(A) • FeedLions(Snowcone) <ul style="list-style-type: none"> ○ Littering(Popcorn)(A) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(R) • Save(D7)(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ○ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ○ Name Ted(R) ○ Occupation related with animals (C) ○ Zoologist(C) ○ ExZoology Student (C) ○ Animal Abuser(P) ○ Has bad wife(C) ○ Has bad children(C) ○ D7.Wife(Hansome) (P) ○ D7.Wife(Manish)(C) ○ D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> ○ Against feeding animals 	<ul style="list-style-type: none"> • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> ○ D7 ChangeSubject(C) REJECTED • FeedLions(Popcorn) <ul style="list-style-type: none"> ○ Littering(Popcorn)(A) • FeedLions(Snowcone) <ul style="list-style-type: none"> ○ Littering(Popcorn)(A) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(P) • Save(D7)(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ○ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ○ Name Ted(R) ○ Occupation related with animals (C) ○ Zoologist(C) ○ ExZoology Student (C) ○ Animal Abuser(R) ○ Has bad wife(C) ○ Has bad children(C) ○ D7.Wife(Hansome) (R) ○ D7.Wife(Manish)(C) ○ D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> ○ Against feeding animals (C) 	<ul style="list-style-type: none"> • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> ○ D7 ChangeSubject(C) REJECTED • Littering(PopCorn)(C) • Littering(Snowcone)(C) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(R) • Save(D7)(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ○ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ○ Name Ted(P) ○ Occupation related with animals (C) ○ Zoologist(C) ○ ExZoology Student (C) ○ Animal Abuser(R) ○ Has bad wife(C) ○ Has bad children(C) ○ D7.Wife(Hansome) (R) ○ D7.Wife(Manish)(C) ○ D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> ○ Against feeding animals (C) ○ Has bad wife(CC) ○ Has bad children(CC) 	
---	---	---	--

108

<ul style="list-style-type: none"> (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (P) Victim(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) 	<ul style="list-style-type: none"> Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (R) Victim(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lions none(A) 	<ul style="list-style-type: none"> FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (C) Victim(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner none(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) 	
---	--	---	--

109

<ul style="list-style-type: none"> Owner Lions <ul style="list-style-type: none"> none(A) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	<ul style="list-style-type: none"> Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) Popcons (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	<ul style="list-style-type: none"> Owner D7 A particular lion(C) 	
---	--	--	--

Turn 52

D5 accepts offer of hundred crimes. And asks D7 if he wants him to prove it.

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(P) Rules ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(P) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) 	Tilt Offer

110

D7.Wife) (R) <ul style="list-style-type: none"> D7.Wife beatEmotionally(D7, D7.Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED FeedLions(Popcorn) <ul style="list-style-type: none"> Littering(Popcorn)(A) FeedLions(Snowcone) <ul style="list-style-type: none"> Littering(Popcorn)(A) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) Characters <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(P) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (P) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 	(R) <ul style="list-style-type: none"> D7.Wife beatEmotionally(D7, D7.Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED FeedLions(Popcorn) <ul style="list-style-type: none"> Littering(Popcorn)(A) FeedLions(Snowcone) <ul style="list-style-type: none"> Littering(Popcorn)(A) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) Characters <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 	(P) <ul style="list-style-type: none"> D7.Wife beatEmotionally(D7, D7.Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED Littering(PopCorn)(C) Littering(Snowcone)(C) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) Characters <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) 	
---	--	---	--

111

<ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (P) Victim(C) Relations <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+++D6 Props <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 	<ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (R) Victim(C) Relations <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =+++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status=+++D6 Props <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lions 	<ul style="list-style-type: none"> Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (C) Victim(C) Relations <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status=+++D6 Props <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner none(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 	
---	--	---	--

112

<ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lions <ul style="list-style-type: none"> none(A) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	<ul style="list-style-type: none"> <ul style="list-style-type: none"> none(A) Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) Popcons (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	<ul style="list-style-type: none"> Popcorns (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	
---	---	--	--

Turn 53

D7 "I should've accepted his offer of committing a hundred crimes (...) but they're were for the first time siding with me and I was processing that..."

D7 exposes his wife as an abuser and names her Jessica.

D5 Frame	D6 Frame	D7 Frame	Comments
Constraint: <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) Location: <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(P) Rules ↯feed(animals)(C) Explicit Offer <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) Activity <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) 	Constraint: <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) Location: <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ↯feed(animals)(C) Explicit Offer <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) Activity <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(P) D6 mocking(Lion)(C) 	Constraint: <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) Location: <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ↯feed(animals)(C) Explicit Offer <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) Activity <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) 	Tilt Offer

113

<ul style="list-style-type: none"> D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (R) D7.Wife beatEmotionally(D7, D7 .Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED <ul style="list-style-type: none"> D7 executes(C) FeedLions(Popcorn) <ul style="list-style-type: none"> Littering(Popcorn)(A) FeedLions(Snowcone) <ul style="list-style-type: none"> Littering(Popcorn)(A) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) Characters <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(P) Has bad wife(C) Has bad children(C) 	<ul style="list-style-type: none"> D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (R) D7.Wife beatEmotionally(D7, D7 .Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED <ul style="list-style-type: none"> D7 executes(C) FeedLions(Popcorn) <ul style="list-style-type: none"> Littering(Popcorn)(A) FeedLions(Snowcone) <ul style="list-style-type: none"> Littering(Popcorn)(A) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) Characters <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) 	<ul style="list-style-type: none"> D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (P) D7.Wife beatEmotionally(D7, D7 .Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED <ul style="list-style-type: none"> D7 executes(C) Littering(PopCorn)(C) Littering(Snowcone)(C) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) Characters <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) 	
--	--	--	--

114

<ul style="list-style-type: none"> o D7.Wife(Hansome) (P) o D7.Wife(Manish)(C) o D7.Wife.Chin(Cleft)(C) <ul style="list-style-type: none"> • D7 <ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC) o Has bad children(CC) o FindsWife(MostBeautiful)(C) o Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) o Height<D7.Wife.Height(C) o Wife=MammothSheWoman (P) o Victim(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) 	<ul style="list-style-type: none"> o D7.Wife(Hansome) (R) o D7.Wife(Manish)(C) o D7.Wife.Chin(Cleft)(C) <ul style="list-style-type: none"> • D7 <ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC) o Has bad children(CC) o FindsWife(MostBeautiful)(C) o Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) o Height<D7.Wife.Height(C) o Wife=MammothSheWoman (R) o Victim(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+D6 o Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) 	<ul style="list-style-type: none"> o D7.Wife.Chin(Cleft)(C) <ul style="list-style-type: none"> • D7 <ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC) o Has bad children(CC) o FindsWife(MostBeautiful)(C) o Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> ▪ Use(chin, mostard) o Height<D7.Wife.Height(C) o Wife=MammothSheWoman (C) o Victim(C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status =Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+D6 o Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C)
---	---	---

115

<ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lions <ul style="list-style-type: none"> ▪ none(A) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lions <ul style="list-style-type: none"> ▪ none(A) o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner none(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	
---	---	--	--

Turn 54

D6 follows new platform of abused husband and offers D7 to have abuse bruises on the neck...

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(P) o Rules o ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) 	Tilt Offer

116

<ul style="list-style-type: none"> Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (R) D7.Wife beatEmotionally(D7, D7 .Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED D7 executes(C) FeedLions(Popcorn) <ul style="list-style-type: none"> Littering(Popcorn)(A) FeedLions(Snowcone) <ul style="list-style-type: none"> Littering(Popcorn)(A) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(P) 	<ul style="list-style-type: none"> Lion eat(Snowcone)(P) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (R) D7.Wife beatEmotionally(D7, D7 .Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED D7 executes(C) FeedLions(Popcorn) <ul style="list-style-type: none"> Littering(Popcorn)(A) FeedLions(Snowcone) <ul style="list-style-type: none"> Littering(Popcorn)(A) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) 	<ul style="list-style-type: none"> Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (P) D7.Wife beatEmotionally(D7, D7 .Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED D7 executes(C) Littering(PopCorn)(C) Littering(Snowcone)(C) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) 	
---	---	---	--

117

<ul style="list-style-type: none"> Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (P) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) <ul style="list-style-type: none"> D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (P) Victim(C) Bruises(R) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) 	<ul style="list-style-type: none"> Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) <ul style="list-style-type: none"> D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (R) Victim(C) Bruises(P) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) 	<ul style="list-style-type: none"> D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) <ul style="list-style-type: none"> D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (C) Victim(C) Bruises(R) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) 	
---	---	---	--

118

<ul style="list-style-type: none"> o Owner D5 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lions <ul style="list-style-type: none"> ▪ none(A) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lions <ul style="list-style-type: none"> ▪ none(A) o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner none(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	
--	---	--	--

Turn 55 to 57

Bruises are confirmed... And D7 offers his wife chains him.

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(P) o Rules o ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o ↯feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) 	Tilt Offer

119

<p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexualy(D7, D7.Wife) (R) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> o D7 ChangeSubject(C) REJECTED o D7 executes(C) • FeedLions(Popcorn) <ul style="list-style-type: none"> o Littering(Popcorn)(A) • FeedLions(Snowcone) <ul style="list-style-type: none"> o Littering(Popcorn)(A) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(C) • Save(D7)(C) • D7.Wife chains(D7)(R) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> o FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> o Name Ted(R) o Occupation related with 	<p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(P) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexualy(D7, D7.Wife) (R) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> o D7 ChangeSubject(C) REJECTED o D7 executes(C) • FeedLions(Popcorn) <ul style="list-style-type: none"> o Littering(Popcorn)(A) • FeedLions(Snowcone) <ul style="list-style-type: none"> o Littering(Popcorn)(A) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(C) • Save(D7)(C) • D7.Wife chains(D7)(R) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> o FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> o Name Ted(R) o Occupation related with 	<p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexualy(D7, D7.Wife) (P) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> o D7 ChangeSubject(C) REJECTED o D7 executes(C) • Littering(PopCorn)(C) • Littering(Snowcone)(C) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(C) • Save(D7)(C) • D7.Wife chains(D7)(P) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> o FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> o Name Ted(P) o Occupation related with animals (C) o Zoologist(C) 	
--	--	--	--

120

<ul style="list-style-type: none"> animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(P) o Has bad wife(C) o Has bad children(C) o D7.Wife(Hansome) (P) o D7.Wife(Manish)(C) o D7.Wife.Chin(Cleft)(C) <ul style="list-style-type: none"> D7 <ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC) o Has bad children(CC) o FindsWife(MostBeautiful)(C) o Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) o Height<D7.Wife.Height(C) o Wife=MammothSheWoman (P) o Victim(C) o Bruises(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) 	<ul style="list-style-type: none"> animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(R) o Has bad wife(C) o Has bad children(C) o D7.Wife(Hansome) (R) o D7.Wife(Manish)(C) o D7.Wife.Chin(Cleft)(C) <ul style="list-style-type: none"> D7 <ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC) o Has bad children(CC) o FindsWife(MostBeautiful)(C) o Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) o Height<D7.Wife.Height(C) o Wife=MammothSheWoman (R) o Victim(C) o Bruises(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status = +D6 Status = +++D6 	<ul style="list-style-type: none"> o ExZoology Student (C) o Animal Abuser(R) o Has bad wife(C) o Has bad children(C) o D7.Wife(Hansome) (R) o D7.Wife(Manish)(C) o D7.Wife.Chin(Cleft)(C) <ul style="list-style-type: none"> D7 <ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC) o Has bad children(CC) o FindsWife(MostBeautiful)(C) o Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> Use(chin, mostard) o Height<D7.Wife.Height(C) o Wife=MammothSheWoman (C) o Victim(C) o Bruises(C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status = +D6 Status = +++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C)
--	--	--

121

<ul style="list-style-type: none"> o Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lions <ul style="list-style-type: none"> none(A) o Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> o Owner D7 A particular lion(C) 	<p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> SuicideS(C) Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lions <ul style="list-style-type: none"> none(A) o Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) Popcons (C) <ul style="list-style-type: none"> o Owner D7 A particular lion(C) 	<ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> SuicideS(C) <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> o Owner none(C) o Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> o Owner D7 A particular lion(C)
---	---	---

Turn 58

D5 offers annual leave to hangout which confirms he received the chain to wall offer.

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(P) Rules ↪feed(animals)(C) <p>Explicit Offer</p>	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ↪feed(animals)(C) <p>Explicit Offer</p>	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ↪feed(animals)(C) <p>Explicit Offer</p>	Tilt Offer

122

<ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexualy(D7, D7.Wife) (R) D7.Wife beatEmotionally(D7, D7 .Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED D7 executes(C) FeedLions(Popcorn) <ul style="list-style-type: none"> Littering(Popcorn)(A) FeedLions(Snowcone) <ul style="list-style-type: none"> Littering(Popcorn)(A) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) D7.Wife chains(D7)(C) D7 anualLeave(P) <p>Characters</p>	<ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(P) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexualy(D7, D7.Wife) (R) D7.Wife beatEmotionally(D7, D7 .Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED D7 executes(C) FeedLions(Popcorn) <ul style="list-style-type: none"> Littering(Popcorn)(A) FeedLions(Snowcone) <ul style="list-style-type: none"> Littering(Popcorn)(A) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) D7.Wife chains(D7)(C) D7 anualLeave(R) <p>Characters</p> <ul style="list-style-type: none"> D6 	<ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexualy(D7, D7.Wife) (P) D7.Wife beatEmotionally(D7, D7 .Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED D7 executes(C) Littering(PopCorn)(C) Littering(Snowcone)(C) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) D7.Wife chains(D7)(C) D7 anualLeave(R) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5
---	--	--

123

<ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(P) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (P) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful) (C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (P) Victim(C) Bruises(C) Prisoner(P) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) 	<ul style="list-style-type: none"> <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (R) Victim(C) Bruises(C) Prisoner(R) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =Status = +++D5 Affinity - Friend(D6, D7) 	<ul style="list-style-type: none"> <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (C) Victim(C) Bruises(C) Prisoner(R) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =Status = +++D5 Affinity - Friend(D6, D7)
---	---	---

124

<ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lions <ul style="list-style-type: none"> ▪ none(A) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+D6 o Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lions <ul style="list-style-type: none"> ▪ none(A) o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> o Status =+D6 o Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner none(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	
--	--	---	--

Turn 59 to 61

Players explore D7 is a prisoner. D7 offers he only eats Krystal Burgers in a blender all year long, and that it makes him fart.

D5 Frame	D6 Frame	D7 Frame	Comments
Constraint: <ul style="list-style-type: none"> • 3 min (C) 	Constraint: <ul style="list-style-type: none"> • 3 min (C) 	Constraint: <ul style="list-style-type: none"> • 3 min (C) 	Tilt Offer

125

<ul style="list-style-type: none"> • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(P) o Rules o ↩feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (R) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> o D7 ChangeSubject(C) REJECTED o D7 executes(C) • FeedLions(Popcorn) <ul style="list-style-type: none"> o Littering(Popcorn)(A) • FeedLions(Snowcone) <ul style="list-style-type: none"> o Littering(Popcorn)(A) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) 	<ul style="list-style-type: none"> • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o ↩feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(P) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (R) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> o D7 ChangeSubject(C) REJECTED o D7 executes(C) • FeedLions(Popcorn) <ul style="list-style-type: none"> o Littering(Popcorn)(A) • FeedLions(Snowcone) <ul style="list-style-type: none"> o Littering(Popcorn)(A) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) 	<ul style="list-style-type: none"> • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o ↩feed(animals)(C) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (P) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> o D7 ChangeSubject(C) REJECTED o D7 executes(C) • Littering(PopCorn)(C) • Littering(Snowcone)(C) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(C) • Save(D7)(C) 	
---	---	--	--

126

<ul style="list-style-type: none"> CommitHundredCrimes(C) Save(D7)(C) D7.Wife chains(D7)(C) D7 anualLeave(P) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(P) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (P) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (P) 	<ul style="list-style-type: none"> CommitHundredCrimes(C) Save(D7)(C) D7.Wife chains(D7)(C) D7 anualLeave(R) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (R) Victim(C) Bruises(C) Prisoner(C) 	<ul style="list-style-type: none"> D7.Wife chains(D7)(C) D7 anualLeave(R) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (C) Victim(C) Bruises(C) Prisoner(C) Eats(KrtystalBurgers)(C) Farts(C)
---	---	--

127

<ul style="list-style-type: none"> Victim(C) Bruises(C) Prisoner(C) <ul style="list-style-type: none"> Smell <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lions <ul style="list-style-type: none"> none(A) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	<ul style="list-style-type: none"> Eats(KrtystalBurgers)(C) Farts(C) <ul style="list-style-type: none"> Smell <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lions <ul style="list-style-type: none"> none(A) Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) Popcons (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	<ul style="list-style-type: none"> Smell <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner none(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C)
---	--	---

128

Turn 62

D5 offers there is a smell is caused by the animals in the zoo... or offers D7 farts to smell as bad as an animal.

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(P) Rules ↯feed(animals)(C) Smell(P) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (R) D7.Wife beatEmotionally(D7, D7 .Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ↯feed(animals)(C) Smell(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(P) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (R) D7.Wife beatEmotionally(D7, D7 .Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ↯feed(animals)(C) Smell(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (P) D7.Wife beatEmotionally(D7, D7 .Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) 	Tilt Offer

129

<p>REJECTED</p> <ul style="list-style-type: none"> D7 executes(C) FeedLions(Popcorn) <ul style="list-style-type: none"> Littering(Popcorn)(A) FeedLions(Snowcone) <ul style="list-style-type: none"> Littering(Popcorn)(A) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) D7.Wife chains(D7)(C) D7 anualLeave(P) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(P) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (P) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) 	<p>REJECTED</p> <ul style="list-style-type: none"> D7 executes(C) FeedLions(Popcorn) <ul style="list-style-type: none"> Littering(Popcorn)(A) FeedLions(Snowcone) <ul style="list-style-type: none"> Littering(Popcorn)(A) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) D7.Wife chains(D7)(C) D7 anualLeave(R) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) 	<p>REJECTED</p> <ul style="list-style-type: none"> D7 executes(C) Littering(PopCorn)(C) Littering(Snowcone)(C) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) D7.Wife chains(D7)(C) D7 anualLeave(R) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> Use(chin, mostard) 	
--	---	---	--

130

<ul style="list-style-type: none"> o FindsWife(MostBeautiful)(C) o Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) o Height<D7.Wife.Height(C) o Wife=MammothSheWoman (P) o Victim(C) o Bruises(C) o Prisoner(C) o Eats(KrtystalBurgers)(C) o Farts(C) <ul style="list-style-type: none"> ▪ Smell <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) 	<ul style="list-style-type: none"> o Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) o Height<D7.Wife.Height(C) o Wife=MammothSheWoman (R) o Victim(C) o Bruises(C) o Prisoner(C) o Eats(KrtystalBurgers)(C) o Farts(C) <ul style="list-style-type: none"> ▪ Smell <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+D6 o Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) 	<ul style="list-style-type: none"> o Height<D7.Wife.Height(C) o Wife=MammothSheWoman (C) o Victim(C) o Bruises(C) o Prisoner(C) o Eats(KrtystalBurgers)(C) o Farts(C) <ul style="list-style-type: none"> ▪ Smell <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status =Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+D6 o Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner none(C) o Syrup? (R) 	
---	---	---	--

131

<ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lions <ul style="list-style-type: none"> ▪ none(A) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> o Owner Lions <ul style="list-style-type: none"> ▪ none(A) o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	
--	---	---	--

Turn 63

D7 accepts offer of smelly zoo and adds to it that it is caused by his farts

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(P) o Rules o ↯feed(animals)(C) o Smell(C) o D7 blinds in(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o ↯feed(animals)(C) o Smell(C) o D7 blinds in(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(P) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o ↯feed(animals)(C) o Smell(C) o D7 blinds in(P) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) 	Tilt Offer

132

<ul style="list-style-type: none"> D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (R) D7.Wife beatEmotionally(D7, D7.Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED D7 executes(C) FeedLions(Popcorn) <ul style="list-style-type: none"> Littering(Popcorn)(A) FeedLions(Snowcone) <ul style="list-style-type: none"> Littering(Popcorn)(A) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) D7.Wife chains(D7)(C) D7 anualLeave(P) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) 	<ul style="list-style-type: none"> D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (R) D7.Wife beatEmotionally(D7, D7.Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED D7 executes(C) FeedLions(Popcorn) <ul style="list-style-type: none"> Littering(Popcorn)(A) FeedLions(Snowcone) <ul style="list-style-type: none"> Littering(Popcorn)(A) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) D7.Wife chains(D7)(C) D7 anualLeave(R) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) 	<ul style="list-style-type: none"> D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (P) D7.Wife beatEmotionally(D7, D7.Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED D7 executes(C) Littering(PopCorn)(C) Littering(Snowcone)(C) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) D7.Wife chains(D7)(C) D7 anualLeave(R) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) 	
--	--	---	--

133

<ul style="list-style-type: none"> ExZoology Student (C) Animal Abuser(P) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (P) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) <ul style="list-style-type: none"> D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (P) Victim(C) Bruises(C) Prisoner(C) Eats(KrtystalBurgers)(C) Farts(C) <ul style="list-style-type: none"> Smell <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 	<ul style="list-style-type: none"> ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) <ul style="list-style-type: none"> D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (R) Victim(C) Bruises(C) Prisoner(C) Eats(KrtystalBurgers)(C) Farts(C) <ul style="list-style-type: none"> Smell <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) 	<ul style="list-style-type: none"> Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) <ul style="list-style-type: none"> D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (C) Victim(C) Bruises(C) Prisoner(C) Eats(KrtystalBurgers)(C) Farts(C) <ul style="list-style-type: none"> Smell <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status=+++D6 	
---	---	---	--

134

<ul style="list-style-type: none"> o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lions <ul style="list-style-type: none"> ▪ none(A) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> o Status =+D6 o Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lions <ul style="list-style-type: none"> ▪ none(A) o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner none(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	
---	--	---	--

Turn 64

D6 offers he can't eat snowcone. He will throw it away..

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(P) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) 	Tilt Offer

135

<ul style="list-style-type: none"> o Rules o ↯feed(animals)(C) o Smell(C) o D7 blinds in(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexualy(D7, D7.Wife) (R) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> o D7 ChangeSubject(C) • REJECTED <ul style="list-style-type: none"> o D7 executes(C) • FeedLions(Popcorn) <ul style="list-style-type: none"> o Littering(Popcorn)(A) • FeedLions(Snowcone) <ul style="list-style-type: none"> o Littering(Popcorn)(A) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(C) • Save(D7)(C) 	<ul style="list-style-type: none"> o rules o ↯feed(animals)(C) o Smell(C) o D7 blinds in(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(P) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexualy(D7, D7.Wife) (R) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> o D7 ChangeSubject(C) • REJECTED <ul style="list-style-type: none"> o D7 executes(C) • FeedLions(Popcorn) <ul style="list-style-type: none"> o Littering(Popcorn)(A) • FeedLions(Snowcone) <ul style="list-style-type: none"> o Littering(Popcorn)(A) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(C) • Save(D7)(C) 	<ul style="list-style-type: none"> o rules o ↯feed(animals)(C) o Smell(C) o D7 blinds in(P) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexualy(D7, D7.Wife) (P) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> o D7 ChangeSubject(C) • REJECTED <ul style="list-style-type: none"> o D7 executes(C) • Littering(PopCorn)(C) • Littering(Snowcone)(C) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(C) • Save(D7)(C) • D7.Wife chains(D7)(C) • D7 anualLeave(R) 	
---	---	--	--

136

<ul style="list-style-type: none"> D7.Wife chains(D7)(C) D7 anualLeave(P) Throw(D5.snowcone)(P) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(P) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (P) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (P) Victim(C) 	<ul style="list-style-type: none"> D7.Wife chains(D7)(C) D7 anualLeave(R) Throw(D5.snowcone)(P) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (R) Victim(C) Bruises(C) Prisoner(C) Eats(KrtystalBurgers)(C) 	<ul style="list-style-type: none"> Throw(D5.snowcone)(P) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (C) Victim(C) Bruises(C) Prisoner(C) Eats(KrtystalBurgers)(C) Farts(C) 	
--	---	--	--

137

<ul style="list-style-type: none"> Bruises(C) Prisoner(C) Eats(KrtystalBurgers)(C) Farts(C) <ul style="list-style-type: none"> Smell <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lions <ul style="list-style-type: none"> none(A) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 	<ul style="list-style-type: none"> Farts(C) <ul style="list-style-type: none"> Smell <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lions <ul style="list-style-type: none"> none(A) Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) Popcons (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	<ul style="list-style-type: none"> Smell <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner none(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	
--	--	---	--

138

<ul style="list-style-type: none"> A particular lion(C) 			
--	--	--	--

Turn 65

D7 builds up on D6 offer and offers him to throw the snowcone to lions.

Is this agains his character? There was a transformation. He's no more the no fun guy!

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(P) Rules ↯feed(animals)(C) Smell(C) D7 blinds in(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (R) D7.Wife beatEmotionally(D7, D7 .Wife)(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ↯feed(animals)(C) Smell(C) D7 blinds in(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(P) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (R) D7.Wife beatEmotionally(D7, D7 .Wife)(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ↯feed(animals)(C) Smell(C) D7 blinds in(P) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (P) D7.Wife beatEmotionally(D7, D7 .Wife)(C) 	Tilt Offer

139

<ul style="list-style-type: none"> D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED <ul style="list-style-type: none"> D7 executes(C) FeedLions(Popcorn) <ul style="list-style-type: none"> Littering(Popcorn)(A) FeedLions(Snowcone) <ul style="list-style-type: none"> Littering(Popcorn)(A) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) D7.Wife chains(D7)(C) D7 anualLeave(P) Throw(D5.snowcone)(P) <ul style="list-style-type: none"> Feed(Lion, Snowcone)(R) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(P) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (P) D7.Wife(Manish)(C) 	<ul style="list-style-type: none"> D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED <ul style="list-style-type: none"> D7 executes(C) FeedLions(Popcorn) <ul style="list-style-type: none"> Littering(Popcorn)(A) FeedLions(Snowcone) <ul style="list-style-type: none"> Littering(Popcorn)(A) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) D7.Wife chains(D7)(C) D7 anualLeave(R) Throw(D5.snowcone) <ul style="list-style-type: none"> Feed(Lion, Snowcone)(R) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) 	<ul style="list-style-type: none"> D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED <ul style="list-style-type: none"> D7 executes(C) Littering(PopCorn)(C) Littering(Snowcone)(C) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) D7.Wife chains(D7)(C) D7 anualLeave(R) Throw(D5.snowcone) <ul style="list-style-type: none"> Feed(Lion, Snowcone)(P) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) D7 	
--	---	---	--

140

<ul style="list-style-type: none"> o D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC) o Has bad children(CC) o FindsWife(MostBeautiful)(C) o Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) o Height<D7.Wife.Height(C) o Wife=MammothSheWoman (P) o Victim(C) o Bruises(C) o Prisoner(C) o Eats(KrtystalBurgers)(C) o Farts(C) <ul style="list-style-type: none"> ▪ Smell <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) 	<ul style="list-style-type: none"> o D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC) o Has bad children(CC) o FindsWife(MostBeautiful)(C) o Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) o Height<D7.Wife.Height(C) o Wife=MammothSheWoman (R) o Victim(C) o Bruises(C) o Prisoner(C) o Eats(KrtystalBurgers)(C) o Farts(C) <ul style="list-style-type: none"> ▪ Smell <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+D6 o Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 	<ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC) o Has bad children(CC) o FindsWife(MostBeautiful)(C) o Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> ▪ Use(chin, mostard) o Height<D7.Wife.Height(C) o Wife=MammothSheWoman (C) o Victim(C) o Bruises(C) o Prisoner(C) o Eats(KrtystalBurgers)(C) o Farts(C) <ul style="list-style-type: none"> ▪ Smell <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status =Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+D6 o Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) 	
--	--	--	--

141

<ul style="list-style-type: none"> o Owner D5 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lions <ul style="list-style-type: none"> ▪ none(A) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lions <ul style="list-style-type: none"> ▪ none(A) o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner none(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	
--	--	--	--

Turn 66

D5 offers to resolve abuse problem by asking D7 to move out and go live with D5.

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(P) o Rules o ↯feed(animals)(C) o Smell(C) o D7 blinds in(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o ↯feed(animals)(C) o Smell(C) o D7 blinds in(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o ↯feed(animals)(C) o Smell(C) o D7 blinds in(P) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) 	Tilt Offer

142

<ul style="list-style-type: none"> • D6 Feed(Lions, Snowcone)(C) Activity <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (R) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> ◦ D7 ChangeSubject(C) REJECTED ◦ D7 executes(C) • FeedLions(Popcorn) <ul style="list-style-type: none"> ◦ Littering(Popcorn)(A) • FeedLions(Snowcone) <ul style="list-style-type: none"> ◦ Littering(Popcorn)(A) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(C) • Save(D7)(C) • D7.Wife chains(D7)(C) • D7 anualLeave(P) • Throw(D5.snowcone)(P) <ul style="list-style-type: none"> ◦ Feed(Lion, Snowcone)(R) • MoveOut(D7) <ul style="list-style-type: none"> ◦ To D5(C) 	<ul style="list-style-type: none"> • D6 Feed(Lions, Snowcone)(C) Activity <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(P) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (R) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> ◦ D7 ChangeSubject(C) REJECTED ◦ D7 executes(C) • FeedLions(Popcorn) <ul style="list-style-type: none"> ◦ Littering(Popcorn)(A) • FeedLions(Snowcone) <ul style="list-style-type: none"> ◦ Littering(Popcorn)(A) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(C) • Save(D7)(C) • D7.Wife chains(D7)(C) • D7 anualLeave(R) • Throw(D5.snowcone) <ul style="list-style-type: none"> ◦ Feed(Lion, Snowcone)(R) • MoveOut(D7) <ul style="list-style-type: none"> ◦ To D5(C) 	<ul style="list-style-type: none"> • D6 Feed(Lions, Snowcone)(C) Activity <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (P) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> ◦ D7 ChangeSubject(C) REJECTED ◦ D7 executes(C) • Littering(PopCorn)(C) • Littering(Snowcone)(C) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(C) • Save(D7)(C) • D7.Wife chains(D7)(C) • D7 anualLeave(R) • Throw(D5.snowcone) <ul style="list-style-type: none"> ◦ Feed(Lion, Snowcone)(P) • MoveOut(D7) <ul style="list-style-type: none"> ◦ To D5(C) Characters
--	---	--

143

Characters <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(P) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (P) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful) (C) ◦ Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ◦ Height<D7.Wife.Height(C) ◦ Wife=MammothSheWoma n (P) ◦ Victim(C) ◦ Bruises(C) ◦ Prisoner(C) ◦ Eats(KrtystalBurgers)(C) ◦ Farts(C) 	Characters <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (R) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) ◦ Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ◦ Height<D7.Wife.Height(C) ◦ Wife=MammothSheWoman (R) ◦ Victim(C) ◦ Bruises(C) ◦ Prisoner(C) ◦ Eats(KrtystalBurgers)(C) ◦ Farts(C) <ul style="list-style-type: none"> ▪ Smell 	<ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(P) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (R) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) ◦ Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ◦ Height<D7.Wife.Height(C) ◦ Wife=MammothSheWoman (C) ◦ Victim(C) ◦ Bruises(C) ◦ Prisoner(C) ◦ Eats(KrtystalBurgers)(C) ◦ Farts(C) <ul style="list-style-type: none"> ▪ Smell Relations <ul style="list-style-type: none"> • Friend(D5, D6)
---	--	--

144

<ul style="list-style-type: none"> Smell <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lions <ul style="list-style-type: none"> none(A) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	<p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lions <ul style="list-style-type: none"> none(A) Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) Popcons (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	<ul style="list-style-type: none"> Status = Affinity ++ <ul style="list-style-type: none"> Friend(D5, D7) <ul style="list-style-type: none"> Status =Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner none(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	
---	--	---	--

145

Turn 67 and 68

D5 and D7 confirm invitation to the D5's spare room.

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(P) Rules ←feed(animals)(C) Smell(C) D7 blinds in(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexualy(D7, D7.Wife) (R) D7.Wife beatEmotionally(D7, D7 .Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) <p>REJECTED</p>	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ←feed(animals)(C) Smell(C) D7 blinds in(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(P) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexualy(D7, D7.Wife) (R) D7.Wife beatEmotionally(D7, D7 .Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) <p>REJECTED</p>	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ←feed(animals)(C) Smell(C) D7 blinds in(P) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexualy(D7, D7.Wife) (P) D7.Wife beatEmotionally(D7, D7 .Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) <p>REJECTED</p>	<p>Tilt Offer</p>

146

<ul style="list-style-type: none"> o D7 executes(C) • FeedLions(Popcorn) <ul style="list-style-type: none"> o Littering(Popcorn)(A) • FeedLions(Snowcone) <ul style="list-style-type: none"> o Littering(Popcorn)(A) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(C) • Save(D7)(C) • D7.Wife chains(D7)(C) • D7 anualLeave(P) • Throw(D5.snowcone)(P) <ul style="list-style-type: none"> o Feed(Lion, Snowcone)(R) • MoveOut(D7) <ul style="list-style-type: none"> o To D5(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> o FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> o Name Ted(R) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(P) o Has bad wife(C) o Has bad children(C) o D7.Wife(Hansome) (P) o D7.Wife(Manish)(C) o D7.Wife.Chin(Cleft)(C) o House with spare room(P) • D7 	<ul style="list-style-type: none"> o D7 executes(C) • FeedLions(Popcorn) <ul style="list-style-type: none"> o Littering(Popcorn)(A) • FeedLions(Snowcone) <ul style="list-style-type: none"> o Littering(Popcorn)(A) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(C) • Save(D7)(C) • D7.Wife chains(D7)(C) • D7 anualLeave(R) • Throw(D5.snowcone) <ul style="list-style-type: none"> o Feed(Lion, Snowcone)(R) • MoveOut(D7) <ul style="list-style-type: none"> o To D5(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> o FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> o Name Ted(R) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(R) o Has bad wife(C) o Has bad children(C) o D7.Wife(Hansome) (R) o D7.Wife(Manish)(C) o D7.Wife.Chin(Cleft)(C) o House with spare room(R) • D7 	<ul style="list-style-type: none"> o D7 executes(C) • Littering(PopCorn)(C) • Littering(Snowcone)(C) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(C) • Save(D7)(C) • D7.Wife chains(D7)(C) • D7 anualLeave(R) • Throw(D5.snowcone) <ul style="list-style-type: none"> o Feed(Lion, Snowcone)(P) • MoveOut(D7) <ul style="list-style-type: none"> o To D5(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> o FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> o Name Ted(P) o Occupation related with animals (C) o Zoologist(C) o ExZoology Student (C) o Animal Abuser(R) o Has bad wife(C) o Has bad children(C) o D7.Wife(Hansome) (R) o D7.Wife(Manish)(C) o D7.Wife.Chin(Cleft)(C) o House with spare room(R) • D7 <ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC)
--	---	--

147

<ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC) o Has bad children(CC) o FindsWife(MostBeautiful)(C) o Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) o Height<D7.Wife.Height(C) o Wife=MammothSheWoman (P) o Victim(C) o Bruises(C) o Prisoner(C) o Eats(KrtystalBurgers)(C) o Farts(C) <ul style="list-style-type: none"> ▪ Smell <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup? (R) 	<ul style="list-style-type: none"> o Against feeding animals (C) o Has bad wife(CC) o Has bad children(CC) o FindsWife(MostBeautiful)(C) o Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) o Height<D7.Wife.Height(C) o Wife=MammothSheWoman (R) o Victim(C) o Bruises(C) o Prisoner(C) o Eats(KrtystalBurgers)(C) o Farts(C) <ul style="list-style-type: none"> ▪ Smell <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+D6 o Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) 	<ul style="list-style-type: none"> o Has bad children(CC) o FindsWife(MostBeautiful)(C) o Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> ▪ Use(chin, mostard) o Height<D7.Wife.Height(C) o Wife=MammothSheWoman (C) o Victim(C) o Bruises(C) o Prisoner(C) o Eats(KrtystalBurgers)(C) o Farts(C) <ul style="list-style-type: none"> ▪ Smell <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status =Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+D6 o Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R)
--	--	--

148

<ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lions <ul style="list-style-type: none"> ▪ none(A) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> • Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lions <ul style="list-style-type: none"> ▪ none(A) ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner none(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	
--	--	---	--

Turn 69

D7 offers a problem to D5 solution, by endowing himself with a tracker chip.

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(P) ◦ Rules ◦ ↯feed(animals)(C) ◦ Smell(C) ◦ D7 blinds in(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↯feed(animals)(C) ◦ Smell(C) ◦ D7 blinds in(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> ◦ Lions in front(C) ◦ rules ◦ ↯feed(animals)(C) ◦ Smell(C) ◦ D7 blinds in(P) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) 	Tilt Offer

149

Activity	Activity	Activity	
<ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (R) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> ◦ D7 ChangeSubject(C) REJECTED ◦ D7 executes(C) • FeedLions(Popcorn) <ul style="list-style-type: none"> ◦ Littering(Popcorn)(A) • FeedLions(Snowcone) <ul style="list-style-type: none"> ◦ Littering(Popcorn)(A) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(C) • Save(D7)(C) • D7.Wife chains(D7)(C) • D7 anualLeave(P) • Throw(D5.snowcone)(P) <ul style="list-style-type: none"> ◦ Feed(Lion, Snowcone)(R) • MoveOut(D7) <ul style="list-style-type: none"> ◦ To D5(C) <p>Characters</p>	<ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(P) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (R) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> ◦ D7 ChangeSubject(C) REJECTED ◦ D7 executes(C) • FeedLions(Popcorn) <ul style="list-style-type: none"> ◦ Littering(Popcorn)(A) • FeedLions(Snowcone) <ul style="list-style-type: none"> ◦ Littering(Popcorn)(A) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(C) • Save(D7)(C) • D7.Wife chains(D7)(C) • D7 anualLeave(R) • Throw(D5.snowcone) <ul style="list-style-type: none"> ◦ Feed(Lion, Snowcone)(P) • MoveOut(D7) <ul style="list-style-type: none"> ◦ To D5(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) 	<ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (P) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> ◦ D7 ChangeSubject(C) REJECTED ◦ D7 executes(C) • Littering(PopCorn)(C) • Littering(Snowcone)(C) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(C) • Save(D7)(C) • D7.Wife chains(D7)(C) • D7 anualLeave(R) • Throw(D5.snowcone) <ul style="list-style-type: none"> ◦ Feed(Lion, Snowcone)(P) • MoveOut(D7) <ul style="list-style-type: none"> ◦ To D5(C) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) 	

150

<ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(P) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (P) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) ◦ House with spare room(P) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) ◦ Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ◦ Height<D7.Wife.Height(C) ◦ Wife=MammothSheWoman (P) ◦ Victim(C) ◦ Bruises(C) ◦ Prisoner(C) ◦ Eats(KrtystalBurgers)(C) ◦ Farts(C) <ul style="list-style-type: none"> ▪ Smell 	<ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (R) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) ◦ House with spare room(R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) ◦ Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ◦ Height<D7.Wife.Height(C) ◦ Wife=MammothSheWoman (R) ◦ Victim(C) ◦ Bruises(C) ◦ Prisoner(C) ◦ Eats(KrtystalBurgers)(C) ◦ Farts(C) <ul style="list-style-type: none"> ▪ Smell ◦ Chipd (C) 	<ul style="list-style-type: none"> • D5 <ul style="list-style-type: none"> ◦ Name Ted(P) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (R) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) ◦ House with spare room(R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) ◦ Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ◦ Height<D7.Wife.Height(C) ◦ Wife=MammothSheWoman (C) ◦ Victim(C) ◦ Bruises(C) ◦ Prisoner(C) ◦ Eats(KrtystalBurgers)(C) ◦ Farts(C) <ul style="list-style-type: none"> ▪ Smell ◦ Chipd (P) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) 	
---	--	--	--

151

<ul style="list-style-type: none"> ◦ Chipd (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lions <ul style="list-style-type: none"> ▪ none(A) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status =+D6 ◦ Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lions <ul style="list-style-type: none"> ▪ none(A) ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ <ul style="list-style-type: none"> • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status =Status = +++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status =+D6 ◦ Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner none(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ◦ Owner D7 • A particular lion(C) 	
---	--	---	--

152

Turn 70

D5 uses his own property of being a zoologist to take chip out.

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(P) Rules ¬feed(animals)(C) Smell(C) D7 blinds in(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (R) D7.Wife beatEmotionally(D7, D7 .Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) <p>REJECTED</p>	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ¬feed(animals)(C) Smell(C) D7 blinds in(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(P) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (R) D7.Wife beatEmotionally(D7, D7 .Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) <p>REJECTED</p>	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ¬feed(animals)(C) Smell(C) D7 blinds in(P) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> D6 Feed(Lion,Snowcone)(C) Lion eat(Snowcone)(R) D6 mocking(Lion)(C) D6 and D5 insult(D7.Wife)(C) D7 Supports(D7.Wife)(C) D7 .wife beatSexually(D7, D7.Wife) (P) D7.Wife beatEmotionally(D7, D7 .Wife)(C) D7.Wife beatPhysically(D7, D7.Wife)(C) D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) <p>REJECTED</p>	Tilt Offer

153

<ul style="list-style-type: none"> D7 executes(C) FeedLions(Popcorn) <ul style="list-style-type: none"> Littering(Popcorn)(A) FeedLions(Snowcone) <ul style="list-style-type: none"> Littering(Popcorn)(A) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) D7.Wife chains(D7)(C) D7 anualLeave(P) Throw(D5.snowcone)(P) <ul style="list-style-type: none"> Feed(Lion, Snowcone)(R) MoveOut(D7) <ul style="list-style-type: none"> To D5(C) D5 cutChip(D7)(P) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(P) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (P) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) House with spare room(P) 	<ul style="list-style-type: none"> D7 executes(C) FeedLions(Popcorn) <ul style="list-style-type: none"> Littering(Popcorn)(A) FeedLions(Snowcone) <ul style="list-style-type: none"> Littering(Popcorn)(A) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) D7.Wife chains(D7)(C) D7 anualLeave(R) Throw(D5.snowcone) <ul style="list-style-type: none"> Feed(Lion, Snowcone)(R) MoveOut(D7) <ul style="list-style-type: none"> To D5(C) D5 cutChip(D7)(R) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) House with spare room(R) 	<ul style="list-style-type: none"> D7 executes(C) Littering(PopCorn)(C) Littering(Snowcone)(C) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) D7.Wife chains(D7)(C) D7 anualLeave(R) Throw(D5.snowcone) <ul style="list-style-type: none"> Feed(Lion, Snowcone)(P) MoveOut(D7) <ul style="list-style-type: none"> To D5(C) D5 cutChip(D7)(R) <p>Characters</p> <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) House with spare room(R) D7 <ul style="list-style-type: none"> Against feeding animals (C) 	
---	--	---	--

154

<ul style="list-style-type: none"> D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (P) Victim(C) Bruises(C) Prisoner(C) Eats(KrtystalBurgers)(C) Farts(C) <ul style="list-style-type: none"> Smell Chiped (C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) 	<ul style="list-style-type: none"> D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (R) Victim(C) Bruises(C) Prisoner(C) Eats(KrtystalBurgers)(C) Farts(C) <ul style="list-style-type: none"> Smell Chiped (C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) 	<ul style="list-style-type: none"> Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (C) Victim(C) Bruises(C) Prisoner(C) Eats(KrtystalBurgers)(C) Farts(C) <ul style="list-style-type: none"> Smell Chiped (C) <p>Relations</p> <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =Status = +++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> 1 Snowcones(C) <ul style="list-style-type: none"> Owner D5 Syrup (C) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) 	
--	---	---	--

155

<ul style="list-style-type: none"> Owner D5 Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lions <ul style="list-style-type: none"> none(A) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	<ul style="list-style-type: none"> SuicideS(C) Snowcones(C) <ul style="list-style-type: none"> Owner Lion(C) Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner Lions <ul style="list-style-type: none"> none(A) Syrup? (P) <ul style="list-style-type: none"> SuicideS(C) Popcons (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	<ul style="list-style-type: none"> Owner Lion(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) 1 Snowcones(C) <ul style="list-style-type: none"> Owner none(C) Syrup? (R) <ul style="list-style-type: none"> SuicideS(C) Popcorns (C) <ul style="list-style-type: none"> Owner D7 A particular lion(C) 	
--	---	--	--

Turn 71

D6 offers prosecute(D7.Wife) activity to D7

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(P) Rules ←feed(animals)(C) Smell(C) D7 blinds in(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ←feed(animals)(C) Smell(C) D7 blinds in(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) 	<p>Constraint:</p> <ul style="list-style-type: none"> 3 min (C) CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> Zoo(C) <ul style="list-style-type: none"> Lions in front(C) rules ←feed(animals)(C) Smell(C) D7 blinds in(P) <p>Explicit Offer</p> <ul style="list-style-type: none"> D7 Feed(Lions, Popcorn) (C) 	Tilt Offer

156

<ul style="list-style-type: none"> • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (R) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> ◦ D7 ChangeSubject(C) REJECTED ◦ D7 executes(C) • FeedLions(Popcorn) <ul style="list-style-type: none"> ◦ Littering(Popcorn)(A) • FeedLions(Snowcone) <ul style="list-style-type: none"> ◦ Littering(Popcorn)(A) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(C) • Save(D7)(C) • D7.Wife chains(D7)(C) • D7 anualLeave(P) • Throw(D5.snowcone)(P) <ul style="list-style-type: none"> ◦ Feed(Lion, Snowcone)(R) • MoveOut(D7) <ul style="list-style-type: none"> ◦ To D5(C) 	<ul style="list-style-type: none"> • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(P) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (R) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> ◦ D7 ChangeSubject(C) REJECTED ◦ D7 executes(C) • FeedLions(Popcorn) <ul style="list-style-type: none"> ◦ Littering(Popcorn)(A) • FeedLions(Snowcone) <ul style="list-style-type: none"> ◦ Littering(Popcorn)(A) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(C) • Save(D7)(C) • D7.Wife chains(D7)(C) • D7 anualLeave(R) • Throw(D5.snowcone) <ul style="list-style-type: none"> ◦ Feed(Lion, Snowcone)(R) • MoveOut(D7) <ul style="list-style-type: none"> ◦ To D5(C) 	<ul style="list-style-type: none"> • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (P) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> ◦ D7 ChangeSubject(C) REJECTED ◦ D7 executes(C) • Littering(PopCorn)(C) • Littering(Snowcone)(C) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(C) • Save(D7)(C) • D7.Wife chains(D7)(C) • D7 anualLeave(R) • Throw(D5.snowcone) <ul style="list-style-type: none"> ◦ Feed(Lion, Snowcone)(P) • MoveOut(D7) <ul style="list-style-type: none"> ◦ To D5(C) • D5 cutChip(D7)(R) • D7 Prosecute(D7.Wife)(R) 	
--	---	--	--

157

<ul style="list-style-type: none"> • D5 cutChip(D7)(P) • D7 Prosecute(D7.Wife)(R) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(P) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (P) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) ◦ House with spare room(P) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) ◦ Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ◦ Height<D7.Wife.Height(C) ◦ Wife=MammothSheWoman (P) ◦ Victim(C) 	<ul style="list-style-type: none"> • D5 cutChip(D7)(R) • D7 Prosecute(D7.Wife)(P) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (R) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) ◦ House with spare room(R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) ◦ Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ◦ Height<D7.Wife.Height(C) ◦ Wife=MammothSheWoman (R) ◦ Victim(C) ◦ Bruises(C) ◦ Prisoner(C) ◦ Eats(KrystalBurgers)(C) 	<p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(P) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (R) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) ◦ House with spare room(R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) ◦ Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ◦ Height<D7.Wife.Height(C) ◦ Wife=MammothSheWoman (C) ◦ Victim(C) ◦ Bruises(C) ◦ Prisoner(C) ◦ Eats(KrystalBurgers)(C) ◦ Farts(C) <ul style="list-style-type: none"> ▪ Smell 	
---	---	--	--

158

<ul style="list-style-type: none"> o Bruises(C) o Prisoner(C) o Eats(KrystalBurgers)(C) o Farts(C) <ul style="list-style-type: none"> ▪ Smell o Chipped (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lions <ul style="list-style-type: none"> ▪ none(A) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 	<ul style="list-style-type: none"> o Farts(C) <ul style="list-style-type: none"> ▪ Smell o Chipped (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+D6 o Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lions <ul style="list-style-type: none"> ▪ none(A) o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> o Chipped (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> o Status = o Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> o Status =Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+D6 o Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner none(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	
---	---	---	--

159

<ul style="list-style-type: none"> • A particular lion(C) 			
--	--	--	--

Turn 72

D6 accepts D7 offer of feeding Lion, and because he feels the story is resolved he proposes all actors to perform this activity with him.

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(P) o Rules o ↯feed(animals)(C) o Smell(C) o D7 blinds in(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (R) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o ↯feed(animals)(C) o Smell(C) o D7 blinds in(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(P) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (R) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) <ul style="list-style-type: none"> o Lions in front(C) o rules o ↯feed(animals)(C) o Smell(C) o D7 blinds in(P) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (P) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, 	Tilt Offer

160

D7.Wife)(C) <ul style="list-style-type: none"> D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED <ul style="list-style-type: none"> D7 executes(C) FeedLions(Popcorn) <ul style="list-style-type: none"> Littering(Popcorn)(A) FeedLions(Snowcone) <ul style="list-style-type: none"> Littering(Popcorn)(A) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) D7.Wife chains(D7)(C) D7 anualLeave(P) Throw(D5.snowcone)(P) <ul style="list-style-type: none"> Feed(Lion, Snowcone) [All] Feed(Lion, Snowcone)(P) MoveOut(D7) <ul style="list-style-type: none"> To D5(C) D5 cutChip(D7)(P) D7 Prosecute(D7.Wife)(R) Characters <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) 	D7.Wife)(C) <ul style="list-style-type: none"> D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED <ul style="list-style-type: none"> D7 executes(C) FeedLions(Popcorn) <ul style="list-style-type: none"> Littering(Popcorn)(A) FeedLions(Snowcone) <ul style="list-style-type: none"> Littering(Popcorn)(A) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) D7.Wife chains(D7)(C) D7 anualLeave(R) Throw(D5.snowcone) <ul style="list-style-type: none"> Feed(Lion, Snowcone) [All] Feed(Lion, Snowcone)(R) MoveOut(D7) <ul style="list-style-type: none"> To D5(C) D5 cutChip(D7)(R) D7 Prosecute(D7.Wife)(P) Characters <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(R) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) 	D7.Wife)(C) <ul style="list-style-type: none"> D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> D7 ChangeSubject(C) REJECTED <ul style="list-style-type: none"> D7 executes(C) Littering(PopCorn)(C) Littering(Snowcone)(C) CommitsCrimes(D5)(CC) CommitsCrimes(D6)(CC) CommitHundredCrimes(C) Save(D7)(C) D7.Wife chains(D7)(C) D7 anualLeave(R) Throw(D5.snowcone) <ul style="list-style-type: none"> Feed(Lion, Snowcone) [All] Feed(Lion, Snowcone)(P) MoveOut(D7) <ul style="list-style-type: none"> To D5(C) D5 cutChip(D7)(R) D7 Prosecute(D7.Wife)(R) Characters <ul style="list-style-type: none"> D6 <ul style="list-style-type: none"> FindsD7Wife(Ugly)(C) D5 <ul style="list-style-type: none"> Name Ted(P) Occupation related with animals (C) Zoologist(C) ExZoology Student (C) Animal Abuser(R) 	
--	---	---	--

161

<ul style="list-style-type: none"> Animal Abuser(P) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (P) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) House with spare room(P) <ul style="list-style-type: none"> D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (P) Victim(C) Bruises(C) Prisoner(C) Eats(KrtystalBurgers)(C) Farts(C) <ul style="list-style-type: none"> Smell Chiped (C) Relations <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =+++D5 Affinity - Friend(D5, D7) 	<ul style="list-style-type: none"> Animal Abuser(R) Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) House with spare room(R) <ul style="list-style-type: none"> D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (R) Victim(C) Bruises(C) Prisoner(C) Eats(KrtystalBurgers)(C) Farts(C) <ul style="list-style-type: none"> Smell Chiped (C) Relations <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =+++D5 Affinity - Friend(D6, D7) 	<ul style="list-style-type: none"> Has bad wife(C) Has bad children(C) D7.Wife(Hansome) (R) D7.Wife(Manish)(C) D7.Wife.Chin(Cleft)(C) House with spare room(R) <ul style="list-style-type: none"> D7 <ul style="list-style-type: none"> Against feeding animals (C) Has bad wife(CC) Has bad children(CC) FindsWife(MostBeautiful)(C) Betty Davis(CleftChin) (P) <ul style="list-style-type: none"> Use(chin, mostard) Height<D7.Wife.Height(C) Wife=MammothSheWoman (C) Victim(C) Bruises(C) Prisoner(C) Eats(KrtystalBurgers)(C) Farts(C) <ul style="list-style-type: none"> Smell Chiped (C) Relations <ul style="list-style-type: none"> Friend(D5, D6) <ul style="list-style-type: none"> Status = Affinity ++ Friend(D5, D7) <ul style="list-style-type: none"> Status =Status =+++D5 Affinity - Friend(D6, D7) <ul style="list-style-type: none"> Status =+D6 	
--	--	---	--

162

<ul style="list-style-type: none"> o Status = +++D5 o Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> o Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lions <ul style="list-style-type: none"> ▪ none(A) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> o Status =+D6 o Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lions <ul style="list-style-type: none"> ▪ none(A) o Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> o Status=+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner D5 o Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner Lion(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> o Owner none(C) o Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> o Owner D7 • A particular lion(C) 	
---	--	--	--

Turn 73 till end

D7 accepts offer and joins the other two executing the activity of feeding the lion. Afterwards D5 calls “Scene”

D5 Frame	D6 Frame	D7 Frame	Comments
<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) 	<p>Constraint:</p> <ul style="list-style-type: none"> • 3 min (C) • CollegeFriends(D5,6,7) <p>Location:</p> <ul style="list-style-type: none"> • Zoo(C) 	Tilt Offer

163

<ul style="list-style-type: none"> o Lions in front(P) o Rules o ↯feed(animals)(C) o Smell(C) o D7 blinds in(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (R) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> o D7 ChangeSubject(C) REJECTED o D7 executes(C) • FeedLions(Popcorn) <ul style="list-style-type: none"> o Littering(Popcorn)(A) • FeedLions(Snowcone) <ul style="list-style-type: none"> o Littering(Popcorn)(A) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(C) 	<ul style="list-style-type: none"> o Lions in front(C) o rules o ↯feed(animals)(C) o Smell(C) o D7 blinds in(R) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(P) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (R) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> o D7 ChangeSubject(C) REJECTED o D7 executes(C) • FeedLions(Popcorn) <ul style="list-style-type: none"> o Littering(Popcorn)(A) • FeedLions(Snowcone) <ul style="list-style-type: none"> o Littering(Popcorn)(A) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(C) 	<ul style="list-style-type: none"> o Lions in front(C) o rules o ↯feed(animals)(C) o Smell(C) o D7 blinds in(P) <p>Explicit Offer</p> <ul style="list-style-type: none"> • D7 Feed(Lions, Popcorn) (C) • D6 Feed(Lions, Snowcone)(C) <p>Activity</p> <ul style="list-style-type: none"> • D6 Feed(Lion,Snowcone)(C) • Lion eat(Snowcone)(R) • D6 mocking(Lion)(C) • D6 and D5 insult(D7.Wife)(C) • D7 Supports(D7.Wife)(C) • D7 .wife beatSexually(D7, D7.Wife) (P) • D7.Wife beatEmotionally(D7, D7 .Wife)(C) • D7.Wife beatPhysically(D7, D7.Wife)(C) • D7 expose(abusesVictim)(C) <ul style="list-style-type: none"> o D7 ChangeSubject(C) REJECTED o D7 executes(C) • Littering(PopCorn)(C) • Littering(Snowcone)(C) • CommitsCrimes(D5)(CC) • CommitsCrimes(D6)(CC) • CommitHundredCrimes(C) • Save(D7)(C) • D7.Wife chains(D7)(C) 	
--	--	--	--

164

<ul style="list-style-type: none"> • Save(D7)(C) • D7.Wife chains(D7)(C) • D7 anualLeave(P) • Throw(D5.snowcone)(P) <ul style="list-style-type: none"> ◦ Feed(Lion, Snowcone) ◦ [All] Feed(Lion, Snowcone)(C) • MoveOut(D7) <ul style="list-style-type: none"> ◦ To D5(C) • D5 cutChip(D7)(P) • D7 Prosecute(D7.Wife)(R) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(R) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(P) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (P) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) ◦ House with spare room(P) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) 	<ul style="list-style-type: none"> • Save(D7)(C) • D7.Wife chains(D7)(C) • D7 anualLeave(R) • Throw(D5.snowcone) <ul style="list-style-type: none"> ◦ Feed(Lion, Snowcone) ◦ [All] Feed(Lion, Snowcone)(C) • MoveOut(D7) <ul style="list-style-type: none"> ◦ To D5(C) • D5 cutChip(D7)(R) • D7 Prosecute(D7.Wife)(P) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(P) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (R) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) ◦ House with spare room(R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) ◦ Betty Davis(CleftChin) (P) 	<ul style="list-style-type: none"> • D7 anualLeave(R) • Throw(D5.snowcone) <ul style="list-style-type: none"> ◦ Feed(Lion, Snowcone) ◦ [All] Feed(Lion, Snowcone)(C) • MoveOut(D7) <ul style="list-style-type: none"> ◦ To D5(C) • D5 cutChip(D7)(R) • D7 Prosecute(D7.Wife)(R) <p>Characters</p> <ul style="list-style-type: none"> • D6 <ul style="list-style-type: none"> ◦ FindsD7Wife(Ugly)(C) • D5 <ul style="list-style-type: none"> ◦ Name Ted(P) ◦ Occupation related with animals (C) ◦ Zoologist(C) ◦ ExZoology Student (C) ◦ Animal Abuser(R) ◦ Has bad wife(C) ◦ Has bad children(C) ◦ D7.Wife(Hansome) (R) ◦ D7.Wife(Manish)(C) ◦ D7.Wife.Chin(Cleft)(C) ◦ House with spare room(R) • D7 <ul style="list-style-type: none"> ◦ Against feeding animals (C) ◦ Has bad wife(CC) ◦ Has bad children(CC) ◦ FindsWife(MostBeautiful)(C) ◦ Betty Davis(CleftChin) (P)
---	---	---

165

<ul style="list-style-type: none"> ◦ FindsWife(MostBeautiful)(C) ◦ Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ◦ Height<D7.Wife.Height(C) ◦ Wife=MammothSheWoman (P) ◦ Victim(C) ◦ Bruises(C) ◦ Prisoner(C) ◦ Eats(KrystalBurgers)(C) ◦ Farts(C) <ul style="list-style-type: none"> ▪ Smell ◦ Chiped (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) 	<ul style="list-style-type: none"> ◦ Betty Davis(CleftChin) (R) <ul style="list-style-type: none"> ▪ Use(chin, mostard) ◦ Height<D7.Wife.Height(C) ◦ Wife=MammothSheWoman (R) ◦ Victim(C) ◦ Bruises(C) ◦ Prisoner(C) ◦ Eats(KrystalBurgers)(C) ◦ Farts(C) <ul style="list-style-type: none"> ▪ Smell ◦ Chiped (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status = +++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status =+D6 ◦ Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) 	<ul style="list-style-type: none"> ▪ Use(chin, mostard) ◦ Height<D7.Wife.Height(C) ◦ Wife=MammothSheWoman (C) ◦ Victim(C) ◦ Bruises(C) ◦ Prisoner(C) ◦ Eats(KrystalBurgers)(C) ◦ Farts(C) <ul style="list-style-type: none"> ▪ Smell ◦ Chiped (C) <p>Relations</p> <ul style="list-style-type: none"> • Friend(D5, D6) <ul style="list-style-type: none"> ◦ Status = ◦ Affinity ++ • Friend(D5, D7) <ul style="list-style-type: none"> ◦ Status =Status = +++D5 ◦ Affinity - • Friend(D6, D7) <ul style="list-style-type: none"> ◦ Status =+D6 ◦ Status =+++D6 <p>Props</p> <ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner D5 ◦ Syrup (C) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ◦ Owner Lion(C) ◦ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C)
--	--	--

166

<ul style="list-style-type: none"> ○ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • 1 Snowcones(C) <ul style="list-style-type: none"> ○ Owner Lions <ul style="list-style-type: none"> ▪ none(A) ○ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ○ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> • 1 Snowcones(C) <ul style="list-style-type: none"> ○ Owner Lions <ul style="list-style-type: none"> ▪ none(A) ○ Syrup? (P) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcons (C) <ul style="list-style-type: none"> ○ Owner D7 • A particular lion(C) 	<ul style="list-style-type: none"> ○ Owner none(C) ○ Syrup? (R) <ul style="list-style-type: none"> ▪ SuicideS(C) • Popcorns (C) <ul style="list-style-type: none"> ○ Owner D7 • A particular lion(C) 	
--	--	--	--

APPENDIX C

LaughToMe! Questionnaire

Questionnaire

Thank you for taking the time to answer this inquiry. The goal of this questionnaire is to evaluate a prototype made in the context of a master thesis at IST - Instituto Superior Técnico. The questionnaire estimated completion time is **5 - 7 minutes**.

The questions in this form are related to a video that tries to portray a sketch or a short scene. The sketch happens in a pastry shop and involves two characters: a seller and a client, represented by the images below.



Seller



Client

First watch the video, then proceed to answer the questions.

A - The following questions regard the seller.

For each question choose the answer which you think applies the most.

The following questions refer to the evolution of the seller's feelings throughout the sketch.

1. How did the seller feel in the **beginning** of the sketch? *

- ☐ Happiness
- ☐ Sadness
- ☐ Worry

- ☐ Surprise
- ☐ Anger
- ☐ Disappointment
- ☐ None of the above

2. What were the seller's feelings in the **middle** the sketch? *

- ☐ He got happy/happier
- ☐ He got sad/saddder
- ☐ He got (more) worried
- ☐ He got (more) surprised
- ☐ He got angry/ angrier
- ☐ He got (more) disappointed
- ☐ None of the above

3. How did the seller felt in the **end** of the sketch? *

- ☐ Happiness
- ☐ Sadness
- ☐ Worry
- ☐ Surprise
- ☐ Anger
- ☐ Disappointment
- ☐ None of the above

For the following questions give an answer from 1 to 5, in which 1 means you disagree completely, 5 that you agree completely according to the following scale.

1	2	3	4	5
Disagree completely	Disagree somehow	Neither agree nor disagree	Agree somehow	Agree completely

4. The seller behaved as expected, given the situation. *

- ☐ (1) Disagree completely ☐ (2) ☐ (3) Neither agree nor disagree ☐ (4) ☐ (5) Agree completely

5. The seller was coherent throughout the sketch. *

- ☐ (1) Disagree completely ☐ (2) ☐ (3) Neither agree nor disagree ☐ (4) ☐ (5) Agree completely

6. The seller became more aggressive as the sketch went on. *

- ☐ (1) Disagree completely ☐ (2) ☐ (3) Neither agree nor disagree ☐ (4) ☐ (5) Agree completely

B - The following questions regard the client:

For each question choose the answer which you think applies the most.

The following questions refer to the evolution of the client's feelings throughout the sketch.

7. How did the client feel in the **beginning** of the sketch? *

- ☐ Happiness
- ☐ Sadness
- ☐ Worry
- ☐ Surprise
- ☐ Anger
- ☐ Disappointment
- ☐ None of the above

8. How did the client's feelings evolved **throughout** the sketch? *

- ☐ He got happy/happier
- ☐ He got sad/saddder
- ☐ He got (more) worried
- ☐ He got (more) surprised
- ☐ He got angry/ angrier
- ☐ He got (more) disappointed
- ☐ None of the above

9. How did the client feel in the **end** of the sketch? *

- ☐ Happiness
- ☐ Sadness
- ☐ Worry
- ☐ Surprise
- ☐ Anger
- ☐ Disappointment

☐ None of the above

For the following questions give an answer from 1 to 5, in which 1 means you disagree completely, 5 that you agree completely according to the following scale.

1	2	3	4	5
Disagree completely	Disagree somehow	Neither agree nor disagree	Agree somehow	Agree completely

10. The client behaved as expected, given the situation. *

☐ (1) Disagree completely ☐ (2) ☐ (3) Neither agree nor disagree ☐ (4) ☐ (5) Agree completely

11. The client was coherent throughout the sketch. *

☐ (1) Disagree completely ☐ (2) ☐ (3) Neither agree nor disagree ☐ (4) ☐ (5) Agree completely

12. The client became more aggressive as the sketch went on. *

☐ (1) Disagree completely ☐ (2) ☐ (3) Neither agree nor disagree ☐ (4) ☐ (5) Agree completely

C - The following questions regard the scene as a whole:

For the following questions give an answer from 1 to 5, in which 1 means you disagree completely, 5 that you agree completely according to the following scale.

1	2	3	4	5
Disagree completely	Disagree somehow	Neither agree nor disagree	Agree somehow	Agree completely

13. The sketch was too long. *

☐ (1) Disagree completely ☐ (2) ☐ (3) Neither agree nor disagree ☐ (4) ☐ (5) Agree completely

14. The sketch was humorous. *

☐ (1) Disagree completely ☐ (2) ☐ (3) Neither agree nor disagree ☐ (4) ☐ (5) Agree completely

15. The sketch had a good ending. *

☐ (1) Disagree completely ☐ (2) ☐ (3) Neither agree nor disagree ☐ (4) ☐ (5) Agree completely

16. The ending should be better explained. *

☐ (1) Disagree completely ☐ (2) ☐ (3) Neither agree nor disagree ☐ (4) ☐ (5) Agree completely

17. The characters felt the same way in the beginning as they did in the end of the sketch. *

☐ (1) Disagree completely ☐ (2) ☐ (3) Neither agree nor disagree ☐ (4) ☐ (5) Agree completely

18. The sketch was funny. *

☐ (1) Disagree completely ☐ (2) ☐ (3) Neither agree nor disagree ☐ (4) ☐ (5) Agree completely

D - Generic Information

19. Age: * ☐ < 18
☐ 19 - 25
☐ 26-35
☐ 36-45
☐ 46-55
☐ >55

20. Gender: * ☐ Male
☐ Female

* = Input is required

This form was created at www.formdesk.com

APPENDIX D

LaughToMe! Output

Character	Action	Speech Act	Expression	Sketch Moment
Seller	Greet	Howdie!	SmileOpenly	Beginning
Client	Greet	Hi there!	SmileSlightly	Beginning
Seller	Offer Assistance	May I help you?	SmileOpenly	Beginning
Client	AskForCake	I'd like a chocolate cake.	SmileSlightly	Beginning
Seller	WarnHinderAppetite	You will ruin your appetite.	SmileOpenly	Middle
Client	AskForCake	Can I have a chocolate cake?	SmileSlightly	Middle
Seller	RaiseMoralIssues	I can't sell you cake, that's against my moral code.	SmileOpenly	Middle
Client	Reason	Can't we reach a common ground? An agreement?	SmileSlightly	Middle
Seller	RaiseMoralIssues	My religion opposes, you know, giving you cake.	SmileOpenly	Middle
Client	AskForCake	Can you please just give me a cake.	ShowSlightDistress	Middle
Seller	RaiseMoralIssues	Morally I'd have to say know. Otherwise... still a no.	SmileSlightly	Middle
Client	AskForCake	Chocolate. Cake. Can I get that.	ShowSlightDistress	Middle
Seller	RaiseMoralIssues	Selling cake would make for bad karma.	SmileSlightly	Middle
Client	AskForCake	I, myself, would fancy a cake.	ShowSlightDistress	Middle
Seller	Reason	Can't we reach a common ground? An agreement?	SmileSlightly	Middle
Client	AskForCake	I'd really like some cake.	ShowSlightDistress	Middle
Seller	WarnHinderAppetite	You will ruin your appetite.	SmileSlightly	Middle
Client	FormallyComplain	This is inadmissible. I want to write down a complaint.	ShowSlightDistress	Middle
Seller	MakeSarcasticComment	Sure, your grandma thinks you're thin too.	SmileSlightly	Middle

Table D.1 – continued from previous page

Character	Action	Speech Act	Expression	Sketch Moment
Client	FormallyComplain	This is inadmissible. I want to write down a complaint.	ShowSlightAnger	Middle
Seller	MakeFatPeopleJoke	There's only room here for one of us. Or ten of me.	SmileSlightly	Middle
Client	FormallyComplain	This is inadmissible. I want to write down a complaint.	ShowGreatAnger	Middle
Seller	TrySellAntiDepressants	You look a tad distressed. May I interest you in these amazing antidepressants?	SmileSlightly	Punchline
Client	RejectAntiDepressants	I don't want those antidepressants, I want a chocolate cake!	ShowGreatAnger	Punchline
Seller	ComplainAboutCrisis	People buy less and less these days. Surely it must be the crisis.	ShowGreatDistress	Punchline

