Mappets: An Interactive Plugin for Transmedia Machinima on Unity3D

Rafael Kuffner dos Anjos, Eugenio Di Tullio, and Rui Prada

Inteligent Agents and Synthetic Characters Group,
Instituto Superior Técnico, UTL, INESC-ID, Lisbon, Portugal
{rkuffner,etullio,rui.prada}@gaips.inesc-id.pt
http://gaips.inesc-id.pt

Abstract. The popularity of Machinima movies has increased greatly in the recent years. From a transmedia point of view, there was little development regarding tools to assist the production of Machinima. These are still mainly focused on the gaming community, and 3D animators. The developed tool aims to bring the typical workflow present on a normal movie set, into a machinima creation environment, expanding possibilities for transmedia productions. With Mappets as a plugin for the Unity3D game engine, we allow a translation from the typical movie dimension to a virtual one. This work evaluates the current state of art of machinima development tools and presents a working solution more adequate for transmedia productions and non-expert users interested in the production of machinima.

Keywords: TransMedia, Entertainment, Machinima, Narratives, Authoring System,Digital Entretainment, Entretainment Technology,Artificial Inteligence

1 Introduction

When thinking about Transmedia [1], movies come up naturally as an alternative medium to tell a story that is written on a book or comic, or just passed by orally. While anyone can create the story, communicating it through a certain medium requires access to tools that might not be available to everyone, or require a certain skillset that has to be learned. Also, producing a high quality movie still require a big array of assets such as actors, locations, lights or cameras. The production of animated movies in the form of Machinima is a viable alternative to a casual filmmaker.

Although tools that support the creation of these movies exist, they still require a certain skillset to be used. There is still a gap between the concepts involved in filmmaking, and machinima, thus hampering the progress of transmedia attempts in this field. There is the need for more specific tools that shorten the distance between this two realities, so that more people can have access to the fast production capabilities of machinima. This article will provide insight into the existing tools to produce Machinima, comparing and evaluating them under the Transmedia perspective. A description of our approach follows, arguing why we feel our system is more adequate to transmedia productions, followed by future work and how the tool will be expanded.

2 Related Work

As Machinima is a new kind of media, there are but a few efforts into developing tools that aid the production of these movies. Series such as Red vs. Blue from *Halo* or the first Machinima *Diary of a Camper* were not assisted by tools, but simple recordings of gameplay, sometimes aided by mods that would give better camera options or change some game rules to allow for more interactions. Nowadays, some in-game tools exist to support these mods, not requiring the user to change his game code. One good example is *WoW Machinima Tool*[1] for the World of Warcraft game client. Other games that show similar tools are *League of Legends*, *The Sims*, *Half Life 2*, *Minecraft* and *Eve Online.*

Regarding standalone tools, all of them provide control over lights and camera and the basic workflow: scripting the story, gathering the visual assets needed, casting the actors, preparing the location, and finally playing and recording the scene; but some brought up new features. *Machinima Studio* [2] allows one to get assets from more than just one game and use on the same production. *Xtranormal* [3]is a similar tool that adds features such as staring, gestures and facial expressions to the characters to the usual ones. *Matinee*[4] is the cinematic tool from *Unreal Development Kit*, mor focused on games. *Source Filmaker*[5] uses *Valve's Source Engine*[6] to perform similar tasks, but not integrated in a Game SDK as Matine. *Moviestorm* [7] is probably one of the best machinima tools on the scene, regarding workflow and ease-to use, Although being behind graphics-wise when compared to the previous two.

3 Mappets Overview

Some state of the art machinima tools described on Section 2 solve one of the problems described in Section 1, that is the high entry point for machinimists without gaming or modding experience, but they still lack the support for transmedia productions. After the evaluation of these tools and direct contact with members of BeActive [8], a company dedicated to Transmedia Storytelling, we

[1] http://pxr.dk/wmt/
[2] http://www.machinimadev.com/
[3] http://www.xtranormal.com/
[4] http://www.unrealengine.com/features/cinematics/
[5] http://www.sourcefilmmaker.com/
[6] http://source.valvesoftware.com/
[7] http://www.moviestorm.co.uk
[8] http://www.beactivemedia.com

gathered a small list of basic requirements that a Machinima tool that could be used on Transmedia production should have, besides the ones already referred in Section 2. Although some of these are supported by the best available tools (Matine and Moviestorm), none of the evaluated tools provide all of them. The new requirements for transmedia productions fulfilled by our tool are the following:

1. Capability to import an external script written in the most used scriptwriting tool worldwide, Final Draft [9]
2. Adaptability to different assets and sceneries, to support different stories.
3. Reusability of assets and possibly content into a different media like a game.
4. Expansibility of the tool to support new assets for subsequent productions.
5. Enabling the division of tasks among a filmmaking team, not concentrating all the tasks in one single user.

3.1 Architecture

We decided to develop a tool over Unity3D, one of the most popular Game Developing Kits nowadays. Not only we are able to develop a generic game to be used as a base for the produced machinima, the tool already fulfills some of the requirements by itself such as character importing, map or scene creation Above all, Unity3D allows us to develop plugins to it's existing editor, a feature not offered by any other Game Development Kit. Figure 1 shows the current architecture for the developed tool that was divided in three main components: The Asset Manager, Director and Player.

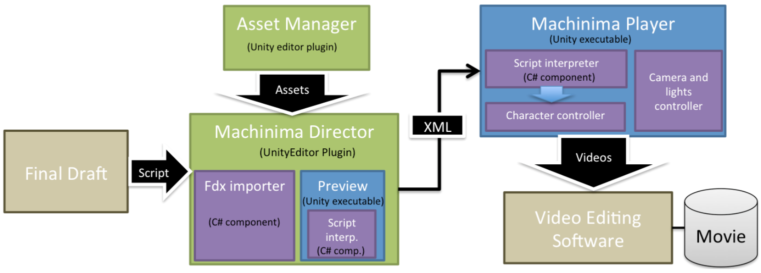


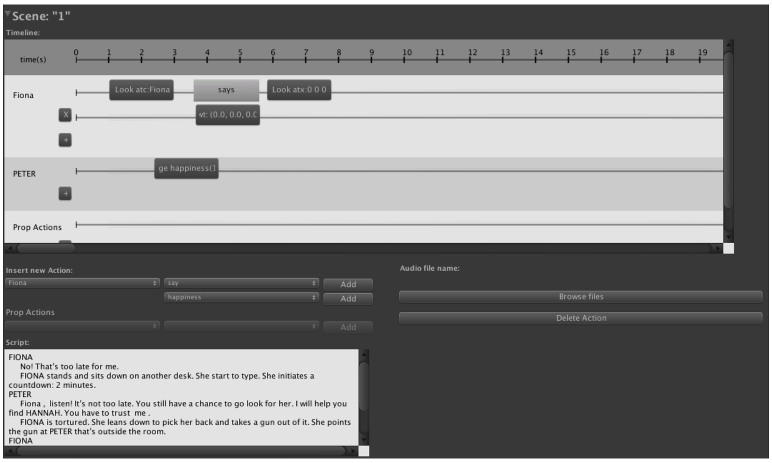
Fig. 1. Overview of our system. Final Draft and the video editing software are external tools.

In analogy to the real life, the assets used in a movie production are Actors, physical locations, and props .On the *Asset Manager* Actors are defined by combining a 3D model to character motion to define it's acting capabilities. By using SmartBody [10] as our character controller, we can remap an animation set used by one model into another, decoupling the playstyle from the looks of the character. The 3D locations can be created on unity or an external tool,

[9] http://www.finaldraft.com

and filled with the needed props for one specific scene. Props are also defined on the Asset manager as a combination of a 3D model, and metainformation for interaction with the actors. These assets are packed into a bundle so they can be imported on the director across different projects, maintaining the tool's modularity.

By importing a script wrote in *Final Draft* on the *Machinima Director* , the tool creates a list of characters and locations from the script, and divides the scenes defined on the script, with each scene containing a list of characters that play it, and a location assigned to it. Since dialog actions are annotated, those can be automatically placed on a timeline and assigned to the correct character. All the other actions that do not have a defined structure are manually translated through a simple actor/action interface (Figure 2), and placed on the same timeline, where their sequence and parameters can be defined. A small preview module is available through UnityEditor, where the assets and actions can be checked. An XML description of everything is then exported to be played on the *Machinima Player* to be recorded as a complete machinima. Camera settings such as focus, field of view and aperture can be controlled and saved on a timeline, and also light settings. The script interpreter module will read the XML action definition and forward it to SmartBody, the character controller, which tells the characters what to do at each moment.



FIONA
No! That's too late for me.
FIONA stands and sits down on another desk. She start to type. She initiates a countdown: 2 minutes.
PETER
Fiona , listen! It's not too late. You still have a chance to go look for her. I will help you find HANNAH. You have to trust me .
FIONA is tortured. She leans down to pick her back and takes a gun out of it. She points the gun at PETER that's outside the room.
FIONA

Fig. 2. Mappets plugin interface example. Window on the Director component where the actions are translated from the script to a timeline

3.2 Early Results

At this stage, our tool is on par with the most recent available options reffered on Section 2. Most of the basic requirements are already included in Unity3D, therefore also in our tool. Other functions such as lipsync, text-to-speech, emotions, can be found on SmartBody, our character controller. We currently do not have a character creator on our tool, but we believe importing any 3D model from Unity3D assets store or other source, and applying different animation sets to it results in a more natural and varied array of characters. Regarding visual quality, Unity3D is very powerful, but the quality of the produced machinima will most likely depend on the detail of the used assets. On the transmedia point of view, our tool has the upper hand among the contestants, making it incredibly faster to jump from a written script to the skeleton of a full movie. By being integrated in a gaming platform, code and content can be reused if a game is to be developed. And as a plugin to Unity Editor, a user experienced with the platform is also able to expand and adapt the tool to his own needs.

3.3 The role of the AI

Altough the current machinima tools are able to speed up and provide functions that facilitate the production of machinima, they are still a step behind real movie productions. The lack of social presence [11], is a problem seen in several machinima production nowadays. Trying to create it through manual instructions is not practical. By using intelligent agents as our actors, we aim to achieve the desired social presence, acting quality and diversity. Figure 3 shows the proposed change to Machinima Player, including two new components that aim to fulfill this.

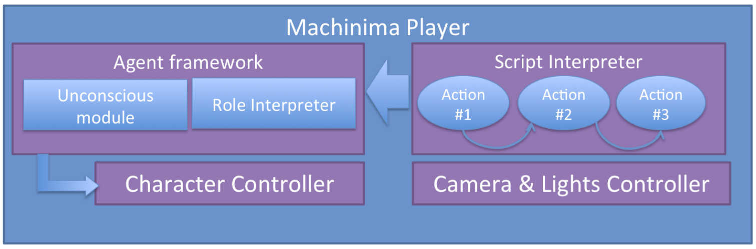


Fig. 3. Proposed changes on Machinima Player. The extended agent framework will provide unconscious behavior and a Role interpreter module.

The *Unconscious Module* (Figure 3) will include unscripted non-verbal communication such as reactions to other actor's presence and actions. The *Role Interpreter* (Figure 3) will add the diversity needed on machinima actors, allowing a director not only to define and change a playstyle for a specific character, but also will allow him to communicate and instruct these agents regarding their acting style, pacing, intensity, and other important aspects of a *Good acting*. We argue that specially on a transmedia setting, a film director will be more at ease, and will be able to employ his skills in a better fashion, if he is not burdened with keyframing and specifying action details.

4 Conclusion and Future Work

We presented a new tool for producing machinima in a transmedia environment. Our tool shortens the distance between a movie production and a Machinima by allowing a director to work with concepts used on a classical production, and a similar workflow. Even though it is still in an early stage, the tool is able to recreate an environment that looked familiar and easy to understand to the experts we are in contact with. Future work on this tool will follow the path described in Section 3.3, developing AI into our virtual actors, so the director can communicate with them as he does with human actors. We believe this will be the next big step taken by these tools, making the workflow and the results closer to real life productions.

Acknowledgments. This work was supported by national funds through FCT Fundação para a Ciência e a Tecnologia, under project PEst-OE/EEI/LA0021/2013 and the bilateral contract with BeActive Media. We also thank Cat Kutay, António Brisson, and Nuno Bernardo for cooperating in the development of this Tool.

References

1. Scolari, C.A.: Transmedia storytelling: Implicit consumers, narrative worlds, and branding in contemporary media production. International Journal of Communication **3**(4) (2009) 586–606
2. Berkeley, L.: Situating machinima in the new mediascape. Australian Journal of Emerging Technologies and Society **4**(2) (2006) 65–80
3. Lowood, H.E., Nitsche, M.: The Machinima Reader. MIT Press (2011)
4. Picard, M.: Machinima: Video game as an art form. Proc. CGSA 2006 (2006)
5. Nitsche, M.: Experiments in the use of game technology for pre-visualization. In: Proceedings of the 2008 Conference on Future Play: Research, Play, Share. Future Play '08, New York, NY, USA, ACM (2008) 160–165
6. Numminen, E., Wrenne, A.: Uncertainty reduction in software development by the use of a platform based development strategy. In: The Proceedings of the 3rd European Conference on Information Management and Evaluation: University of Gothenburg, Sweden, 17-18 Sept 2009, Academic Conf. Limited (2009) 356
7. Elson, D.K., Riedl, M.O.: A lightweight intelligent virtual cinematography system for machinima production. Defense Technical Information Center (2007)
8. Jhala, A., Rawls, C., Munilla, S., Young, R.M.: Longboard: A sketch based intelligent storyboarding tool for creating machinima. In: Proceedings of the Florida Artificial Intelligence Research Society Conference (FLAIRS). (2008)
9. Kirschner, F.: 4 toward a machinima studio. The Machinima Reader (2011) 53
10. Thiebaux, M., Marsella, S., Marshall, A.N., Kallmann, M.: Smartbody: behavior realization for embodied conversational agents. In: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 1. AAMAS '08, Richland, SC, International Foundation for Autonomous Agents and Multiagent Systems (2008) 151–158
11. Mennecke, B., Triplett, J., Hassall, L., Conde, Z.: Embodied social presence theory. In: System Sciences (HICSS), 2010 43rd Hawaii Intern. Conf. (2010) 1–10