Agents for Automated User Experience Testing

Pedro M. Fernandes
INESC-ID and Instituto Superior Técnico, Univ. de Lisboa
Lisbon, Portugal
pedro.miguel.rocha.fernandes@ist.utl.pt

Manuel Lopes
INESC-ID and Instituto Superior Técnico, Univ. de Lisboa
Lisbon, Portugal
manuel.lopes@tecnico.ulisboa.pt

Rui Prada
INESC-ID and Instituto Superior Técnico, Univ. de Lisboa
Lisbon, Portugal
rui.prada@tecnico.ulisboa.pt

Abstract—The automation of functional testing in software has allowed developers to continuously check for negative impacts on functionality throughout the iterative phases of development. This is not the case for User eXperience (UX), which has hitherto relied almost exclusively on testing with real users. User testing is a slow endeavour that can become a bottleneck for development of interactive systems. To address this problem, we here propose an agent based approach for automatic UX testing. We develop agents with basic problem solving skills and a core affect model, allowing us to model an artificial affective state as they traverse different levels of a game. Although this research is still at a primordial state, we believe the results here presented make a strong case for the use of intelligent agents endowed with affective computing models for automating UX testing.

Index Terms—user experience, software testing, artificial intelligence, agents

I. Introduction

Hitherto, User eXperience (UX) testing has mostly relied on real users interacting with the system under test, following a number of different approaches [1], [2]. When applied correctly, such methods allow us to gain a wealth of knowledge about the caveats of the system and take action in order to improve its UX.

Testing with users, however, is a slow endeavour. It is not usually viable to have a user testing session at each step of the iterative development process. Whereas today is common practice to run automated functionality tests each time a system suffers an alteration, the same cannot be currently done concerning UX.

In this paper, we will propose using intelligent agents endowed with an affective model to run automated UX tests. We do not defend such agents could, in the foreseeable future, completely replace testing with real users. What we propose is that agents could be used to maintain a focus on UX throughout the entire development process, running alongside automatic functional tests.

Our research objective is to understand if such testing agents could be a viable approach to partially automate UX testing. With this intent, we have developed an agent endowed with an internal core affect model [3]. This agent was then used to gain insights on the UX of different maps of a game (Sec. III). The results here presented aim to be a proof of concept, showing some of the information one could obtain using UX testing agents. We believe these primordial results make a strong claim for the usefulness of such agents. We further propose a number of ways on which these agents could be improved (Sec. VII).

This paper is organised as follows. In Sec. II we give a brief overview of previous research in the area. In Sec. III the testing environment is presented. In Sec. IV we describe the core architecture of the agent. The results are described in Sec. V and the conclusions on Sec.VI. Finally, in Sec. VII we discuss the results and propose a number of ways in which the presented UX testing agents could be improved.

II. Related Work

Autonomous agents have been previously endowed with emotional models and agents have been used for testing software. However, to the knowledge of the authors, both these things have never been done simultaneously in order to test UX. We believe this represents a research gap.

With regards to emotional agents, two reviews have been done, providing a wealth of information on different methods and approaches for creating agents endowed with artificial emotions [4], [5]. Most of these emotional agents were developed in order to be able to have more realistic interactions with users. They were therefore created to improve UX, but not to test it.

Playtesting agents have been developed that strive to play games in a similar fashion to real users. Designed to have some of the same limitations that human players have, these agents can have, for example, a field of view and a limited short-term memory. Such agents can then be used to play games and find problems and exploits that would normally require testing with real users to find. Stahlke et al. proposed a framework to create this type of agents, following heuristics based on human

behaviours, like exploration, hazard avoidance and aggression [6].

Playtesting agents have also been used to aid design. Holmgård et al. [7] developed agents with a range of different personas, that is, following different behaviours and with different objectives. These agents were then used to aid developers create maps for a 2D game. These agents allowed developers to predict how different players would traverse the map and make design decisions accordingly. The creation of playtesting agents has also been used to test board games [8], card games [9] and frameworks for the development of playtesting agents have been proposed [10], [11].

Outside the realm of games and focusing on the problem of automated UX testing, there is research in automatic Graphical User Interface (GUI) testing [12], [13] and automatic usability testing [14], [15]. Both these areas are of relevance to UX testing, but we were unable to find any work in them that attempted to do any real-time estimates as an agent interacted with a system. In the area of User Modelling, real-time estimates of the internal state of users is often done [16] and some works have even focused on modelling the internal emotional states of users [17], [18], but to out knowledge, none has done so with the intent of UX testing.

III. Test Case: Lab Recruits Game

The system under test for this paper will be a game called Lab Recruits[1]. Lab Recruits is a simple 3D game where the player must interact with objects in order to achieve a goal. The only actions the player can do is move or try to interact with objects. Objects with which the player can interact will henceforth be called **interactables**.

For the examples presented in this paper, the following objects were used:

- **Door with Button (relevant object):** A door that the player can open by interacting with the corresponding button. This object pair is represented in game by a door and a button connected by a wire.
- **Simple Button (irrelevant object):** A button that is not connected to anything. Even though the player can interact with it, such an interaction is not necessary for completing the player's objective. This object is represented in game by a sphere that can have different colours and which does not have a wire connecting it to anything.
- **Chair (the objective):** Finding this chair is, in all our examples, the final objective of the player. This object is represented in game by a black office chair.

Further ahead in the paper, four Lab Recruits maps will be introduced (Fig. 2a, 3a, 4a and 5a). All of those maps follow the same basic premise: the player spawns in a maze, which it must traverse in order to find a chair. To do so, the player might have to interact with certain buttons to open doors. Finding the chair is the ultimate goal of the player and when the chair is found, the game level ends.

IV. Agent Model

In order to be used for the UX testing of our Lab Recruits maps, the agents had to fulfil two main requirements: (a) be able to traverse the maze; and (b) record information relevant for UX assessment as they do so. Our approach to solving (a) is described in Sec. IV-A and our solution for (b) in Sec. IV-B.

A. Search and Traverse Algorithm

Here he will briefly describe the AI capabilities of the agents in order to give the reader an understanding of the agent's behaviour. Our goal was to have an agent that would behave in a similar fashion to a real user.

The agent has a field of view and cannot perceive what is behind a wall or further than a specific distance. Being endowed with a spatial memory, the agent creates an internal map recording all the locations and objects it has already found. This internal map is that which the agent uses for navigation.

The moment the agent spawns in a map, it only knows the location of that which it can directly perceive and knows which object it is trying to find: a chair. It also has the prior knowledge of how to open a door which is connected to a button. As the focus of our simulations was to test UX, it was counter productive to have the agent learn something that the grand majority of real users would already know how to do.

The agent search algorithm runs as follows:

1) The agent perceives the environment and adds to its internal map all the locations and objects it has found.
2) If the chair was found, the simulation ends.
3) If the chair was not found but a button that opens a door was found, the agent will move towards the button and interact with it in order to open the door. If more than one door opening button was found, the agent will randomly choose one to interact with.
4) If neither the chair nor a door opening button were found, the agent moves to the closest information limit of its internal map, striving to find more locations and objects. A location is considered an information limit if the agent does not have any information of what is beyond it. The agent considers walls hard limits and will not attempt to explore past them.

These steps are repeated in this order until the chair is found or all off the map is completely explored.

B. Affective Model

The ISO 9241-210:2019 defines UX as the "user's perceptions and responses that result from the use and/or anticipated use of a system, product or service.", further clarifying that "Users' perceptions and responses include the users' emotions, beliefs, preferences, perceptions, comfort, behaviours, and accomplishments that occur before, during and after use." [19].

Our approach is to endow the testing agents with an affective state and use that agents' internal state to infer properties of the expected UX.

Several theoretical models of emotion have been proposed [20]–[22], a number of which have already been used to

give agents artificial emotions [4], [5]. We decided to base our model on the Core Affect theory of emotions [3]. This theory defends that emotions are constructed from an initial affective state through processes like attribution and appraisal. This affective state is defined using two dimensions, Valence (positive/negative connotation) and Arousal (calm/exciting).

An affective state, by itself, is not enough to define an emotional state. But according to the Core Affect theory, it is the starting point of emotions, and as such will be the starting point of our research. In Sec.VII we will discuss how this approach could be built upon in order to characterise complete emotional states.

To use this model, we need to define how the agent's interaction with the environment will alter those affective dimensions (the appraisal rules). At this stage of the research, our main focus is understanding the feasibility of an agent based approach to automatically test UX. As such, on defining how the agent's interaction with the environment alters its artificial affective state, we decided to use simple rules. There is a lot of room for improvement over this rule based approach and some proposals will be made in Sec.VII.

Both the affective dimensions will be in the range $\{-5, 5\}$. When the agent spawns in the environment, both dimensions are neutral, that is, having a value of 0. From that moment on, the affective dimensions are thus calculated:

- **Valence:**
  - Whenever the agent is able to accomplish his objective (finding the chair) or intermediate sub-objectives (opening doors), the Valence dimension **increases** by 1.
  - If the agent does not accomplish any objective or sub-objective for 10 seconds, the Valence dimension **decreases** by 0.4.
- **Arousal:**
  - Whenever the agent finds a new interactable object (buttons and doors), the Arousal dimension **increases** by 1
  - If the agent doe not find any new interactable for 10 seconds, the Arousal dimension **decreases** by 0.4.

The values here used are not yet an accurate representation of the affective response of a real user. To be so, they would need to be experimentally tested, which they have not yet been at this point in the research. As it stands, we wish to understand if such a simple affective model could already provide meaningful UX insights when used to test a number of different scenarios. One of the next steps on the development of the model will be to define a more robust system for the affective state update.

V. RESULTS

On this section, we describe tests with four different Lab Recruits maps. For each, we will present the changes to the affective state of the aforedescribed agent (Sec. IV) as the map is traversed. Such changes will be shown both in function of the location of the agent and in function of time.
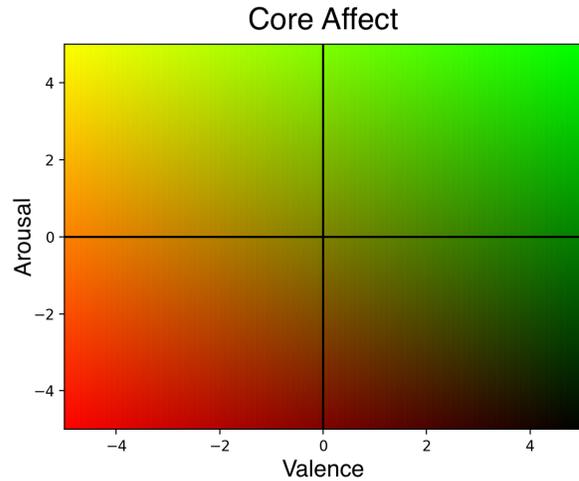

Fig. 1. The colour gradation in the two dimensional core affect space, used to represent the artificial affective state of the agent as it traverses the maps.

A. Map 1

The first Lab Recruits map we will explore is the simplest one (Fig. 2a). The agent must traverse a maze where it finds no interectables in order to reach a door. When the agent interacts with the button that opens that door, it finds the chair, which is its objective.

The evolution of the agent's affective state in function of time and space can be found on Fig. 2c and Fig. 2b, respectively.

The Valence and Arousal dimensions remain negative throughout the agent's traversal of the map. Both Arousal and Valence steadily decrease (Fig. 2c) until the agent finds the door and the button, where both dimensions finally increase (Fig. 2b).

B. Map 2

In the second map, the agent must traverse the same maze as it did in Map 1. However, this time the agent will find interactables on its way to its final objective. These interactables, being buttons not connected to anything, are not relevant to the completion of the agent's objective.

The evolution of the agent's affective state in function of time and space can be found on Fig. 3c and Fig. 3b, respectively.

The Valence dimension of the agent's affective state steadily decreases as it traverses the maze (Fig. 3c), but this time its Arousal dimension increases each time the agent finds an interactable (Fig. 3b).

C. Map 3

In Map 3, the agent once again finds itself in a maze. This time, to reach the chair, the agent will not only have to open the final door but 3 other doors that are located throughout the maze. The agent is unable to reach its objective unless the 4 doors are opened. Besides the doors and their corresponding buttons, the agent will find 2 other interactables in the maze.
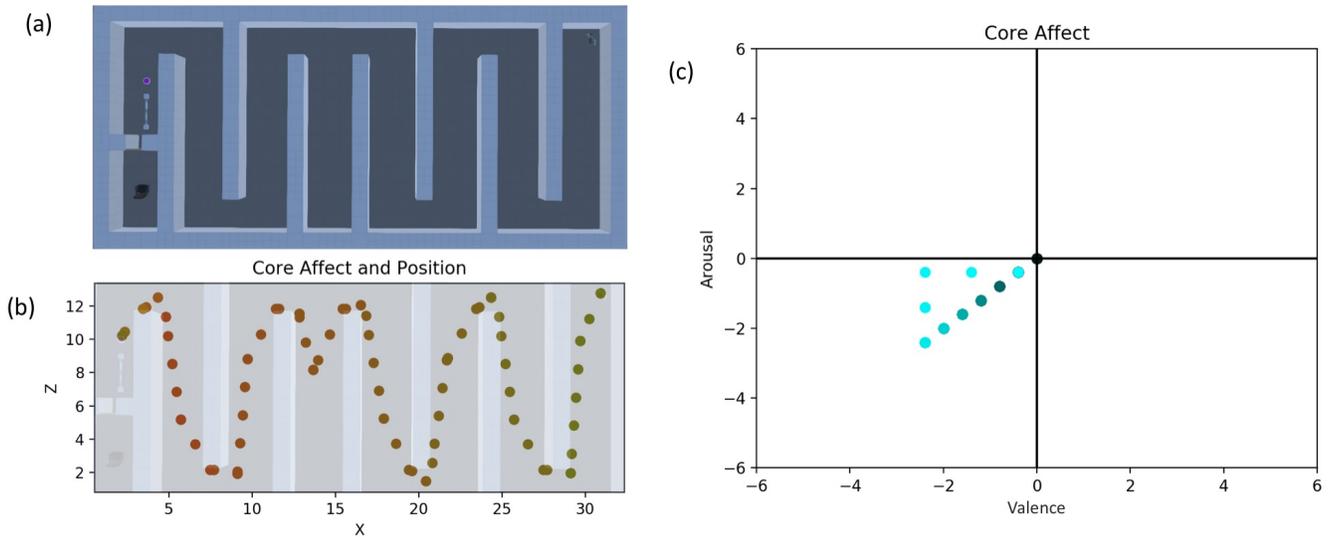
Fig. 2. The evolution of the agent's affective state in function of space **(b)** and time **(c)** as it traverses Map 1 **(a)**. The colours in **(c)** can be mapped to the 2-dimensional Core Affect space using Fig. 1. In **(c)**, the dots go from black to light blue as time passes. We can see that both affective dimensions turn gradually more negative over time until the agent finally finds the door and opens it, which leads to an increase in both the affective dimensions. The agent's affective state remains in the negative-arousal and negative-valence quadrant throughout the traversal.
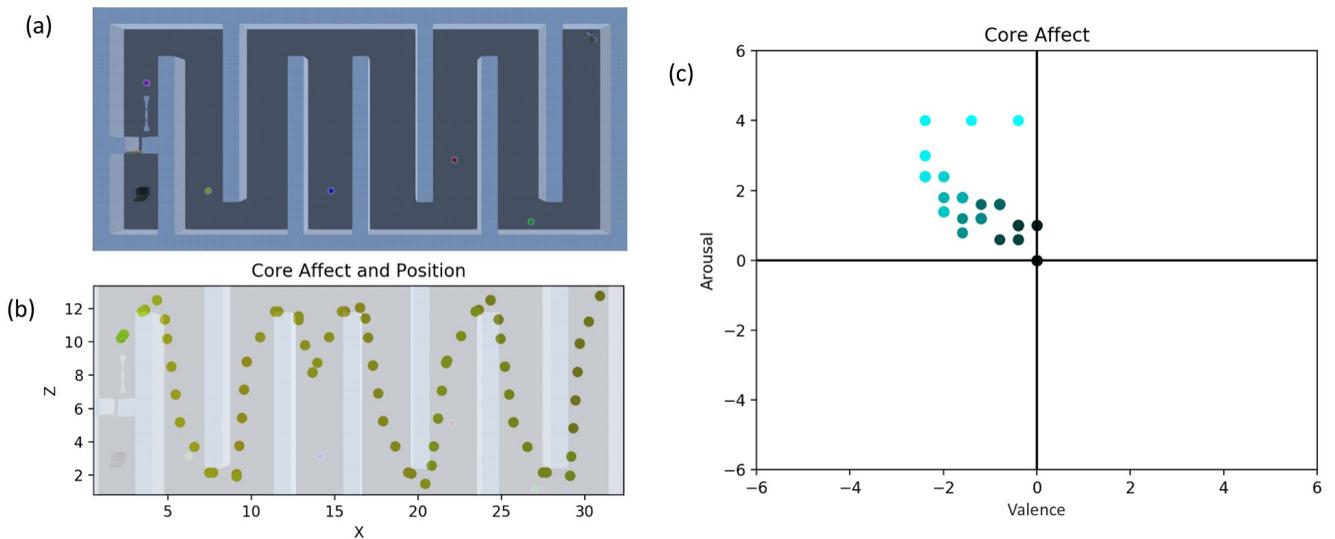

Fig. 3. The evolution of the agent's affective state in function of space **(b)** and time **(c)** as it traverses Map 2 **(a)**. The colours in **(b)** can be mapped to the 2-dimensional Core Affect space using Fig. 1. In **(c)**, the dots go from black to light blue as time passes. In this map, the arousal dimension increases each time the agent finds a new interactable. The valence dimension, however, steadily decreases until the agent finds the door and opens it, finding the chair. The agent's affective state remains in the positive-arousal and negative-valence quadrant throughout the traversal.

The evolution of the agent's affective state in function of time and space can be found on Fig. 4c and Fig. 4b, respectively.

Unlike the previous maps, both the Valence and Arousal affective dimensions remain positive throughout the agent's traversal of the map (Fig. 4c). The agent's Arousal increases whenever it sees an interactable and its Valence increases each time the agent successfully opens a door (Fig. 4b).

D. Map 4

The fourth map we will analyse is considerably different from the previous 3. This time, the agent is not in a maze but in a room with 28 doors. Each door has a button that opens it and behind only one of the doors, is the chair. The agent can see from the start all the doors and buttons but cannot see the chair before opening the right door.

In this map, the agent's affective experience is very different depending on how "lucky" the agent is. In the best case scenario, the agent will open the correct door on the first
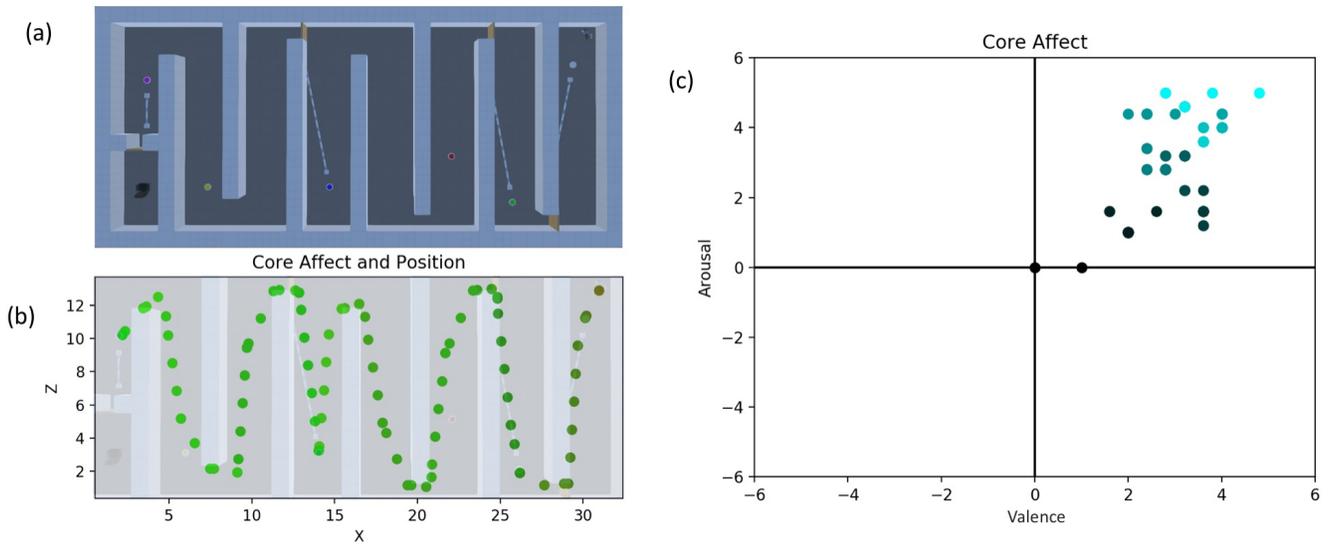
Fig. 4. The evolution of the agent's affective state in function of space **(b)** and time **(c)** as it traverses Map 3 **(a)**. The colours in **(b)** can be mapped to the 2-dimensional Core Affect space using Fig. 1. In **(c)**, the dots go from black to light blue as time passes. We can see that both affective dimensions increase as the agent traverses the map. Arousal increases whenever the agent finds a new interactable and valence increases whenever the agent is able to accomplish a goal or sub-goal (opening doors). The agent's affective state remains in the positive-arousal and positive-valence quadrant throughout the traversal.
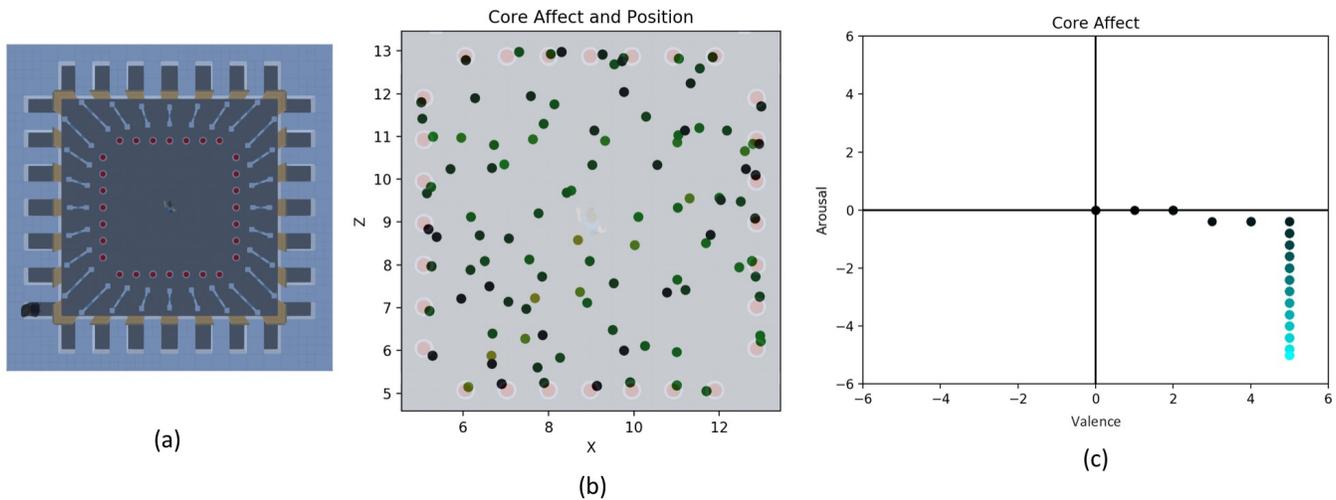

Fig. 5. The evolution of the agent's affective state in function of space **(b)** and time **(c)** as it traverses Map 4 **(a)** in the **worst case scenario**. The colours in **(b)** can be mapped to the 2-dimensional Core Affect space using Fig. 1. In **(c)**, the dots go from black to light blue as time passes. This traversal is a chaotic one, with the agent pressing all possible buttons before finally finding the chair. As the agent finds nothing new throughout the traversal, it's arousal dimension steadily decreases, whereas it's valence dimension increases as the agent is able to accomplish sub-objectives (opening doors).

attempt. On the worst case scenario, the agent will open 27 doors before finally opening the correct one. Because of this, we will here explore both the best and worst case scenarios.

1) Best Case Scenario: In the best case scenario, the first button that the agent activates opens the correct door, leading it to the chair.

In this scenario, the agent's traversal of this map is swift, as the agent must only move from its original position to that of the correct button. As a result, its Arousal dimension doesn't have time to suffer any changes. Its Valence dimension

increases as the agent is able to open the door and find the chair.

2) Worst Case Scenario: In the worst case scenario, the agent first opens at random 27 doors before finally opening the correct one, behind which it finds the chair.

The evolution of the agent's affective state for this scenario in function of time and space can be found on Fig. 5c and Fig. 5b, respectively.

Whenever the agent is able to successfully open a door, the agent's Valence dimension increases (Fig.5b). However,

since the agent could see all the buttons and doors since the beginning, there is nothing to increase the agent's Arousal, which steadily decreases with time (Fig 5c).

VI. Conclusion

Our agent had considerably different affective responses when traversing each of the 4 different maps that we have tested.

On Map 1 (Fig. 2), the absence of any interactables throughout the map traversal led the agent's affective state to be on the negative-arousal and negative-valence quadrant during the entirety of the simulation. On Map 2 (Fig. 3), interactables were scattered throughout the maze, being, however, not relevant to the completion of the agent's objective. This led the agent's affective state to remain on the positive-arousal and negative-valence quadrant. On Map 3 (Fig. 4), not only does the agent find interactables but also doors that it needs to open in order to accomplish its objective. Both these things make the agent's affective state remain on the positive-arousal and positive-valence quadrant. Finally, on Map 4 (Fig. 5), the nature of the scenario made it so that the agent would have very different traversals depending on which doors it chose to open. Because of this, we decided to study both the best and the worst case scenario. On the best case scenario, the agent quickly finds the chair, having an increase on the valence dimension of its affective state as a result. On the worst case scenario, the agent continuously opens doors that it had already found and that do not lead to its objective, increasing its valence dimension but making the agent's arousal steadily decrease. In this case, the agent's affective state remains in the negative-arousal and positive-valence quadrant.

We can thus see how different maps arouse different affective states on the test agent. Not only that, but the conclusions we can gather from the evolution of affective states are in accordance with what would be expected. It is not surprising that a map that consists of a monotonous maze will give a player neither arousal nor much valence whereas a maze with challenges that the player can solve and a number of objects to interact with will do the opposite. In maps of greater magnitude, which might have several sections with different densities of challenges and objects, it is not as trivial to understand how the affective state of a player will evolve. The information this approach provides can help developers to pinpoint exactly where changes need to be made in order to arouse in the players the desired affective states.

The very different results found in Map 4 (Fig 5a) for different traversals of the agent also shows that UX is not only scenario dependent. There will be variability of UX depending on the user's traversal of the system. Running the same scenario several times using stochastic agents or agents with different personas can help designers achieve UX robustness. A design goal could then be a specification of the percentage of agents that would have an UX that fit inside certain parameters.

This paper aims to be a proof of concept, showing that UX testing agents are able to provide information that can be highly useful for both testing and designing. The fact such agents can show UX related information at precise moments and locations also allows them to relay information that would be very difficult to obtain with real users unless physiological methods were employed. Enquiring a user about perceived experience while playing a game will inherently alter the user experience, but that is not the case with UX testing agents.

VII. Discussion

After seeing the examples here presented, the reader might have the following questions: "But how accurate is this information? How can I be sure the information the agents conveys does indeed reflect how a real user would feel and experience the system?". Those are very good questions, which we are currently working on to be able to answer. With this paper, our goal was to show how one could employ agents in order to test UX and explore the type of information such agents could provide. We believe our results show that agents could help automate UX testing and make it an integral part of the development process without having bottleneck concerns. In the following paragraphs, we will discuss ways in which this UX testing agents could be improved to be made more accurate and cover different kind of UX tests.

To ensure the results do indeed correlate with how users experience the system, user testing could be employed in order to fine tune the model to faithfully represent the affective changes users experience. This testing with real users could be done only once in order to attune the UX testing agents to the system under test, allowing the developed agents to be used automatically and without requiring further user testing for the remainder of the system's life-cycle.

This "tuning" process could be done, for example, using machine learning and user testing based on physiological methods [23], [24]. Users could be asked to interact with the system under test as both physiological measurements and game events are recorded. The game events, like player location and object interaction, could then be used as input for the machine learning model and an interpretation of the physiological measurements under the emotional model chosen could be the desired output. We could then fine tune the model parameters and the model itself to ensure they accurately represent how a real user experiences the system.

In order to have access to full emotional states instead of affective states, the model could be expanded to have an extra dimension, Dominance, and modified in order to be in accordance with the PAD model of emotions [20]. This model claims emotions can be characterised in a 3-dimensional state, the dimensions being Pleasure, Arousal and Dominance. The pleasure dimension of the PAD model corresponds to the valence dimension used for the core affect model.

The fact that different traversals of the same map lead to different affective results exposes the importance of developing agents display interaction traces that resemble human behaviour. If the agents make choices that a user would never do, then the UX information the agents convey might prove to be of no importance. It might also be of no interest to try

and program agents to exploit systems in order to find ways of interacting which lead to a negative UX, as the agent might, in the example of a game, decide to continuously walk against a wall and proceed to claim the game can be very uneventful. We thus defend that the development of UX testing agents will not only require well tuned models to assess UX, but also agents that behave similarly to human users.

Finally, we here present a preliminary approach to test a very simple game, but this approach could be modified to test a number of other systems. The objects used in the 4 scenarios here presented could be abstracted to represent other types of objects that are regularly found in games. The door with button can be abstracted as a NPC or interactable which, when interacted with, allows the player to progress towards its objective. The simple button can be abstracted as a NPC or other interactables that are present in games but which are not directly relevant to the completion of the player's objective. The chair can be abstracted as any item that a player must find or a specific location/state that must be reached.

Furthermore, this approach should not be interpreted as game-specific. Agents with a similar architecture could be used, for example, to test the interface of a store, with the valence and arousal dimensions being modelled based on the user finding an item of interest or being able to accomplish a successful purchase or using a coupon. We believe this approach could be used with any system where events that affect the core affect state of an user can be, to a certain degree, identified. This event identification could even be done automatically, using machine learning models and physiological measurements.

REFERENCES

[1] A. P. Vermeeren, E. L.-C. Law, V. Roto, M. Obrist, J. Hoonhout, and K. Väänänen-Vainio-Mattila, "User experience evaluation methods: current state and development needs," in *Proceedings of the 6th Nordic conference on human-computer interaction: Extending boundaries*, 2010, pp. 521–530.

[2] L. Rivero and T. Conte, "A systematic mapping study on research contributions on ux evaluation technologies," in *Proceedings of the XVI Brazilian Symposium on Human Factors in Computing Systems*, 2017, pp. 1–10.

[3] J. A. Russell, "Core affect and the psychological construction of emotion." *Psychological review*, vol. 110, no. 1, p. 145, 2003.

[4] T. Rumbell, J. Barnden, S. Denham, and T. Wennekers, "Emotions in autonomous agents: comparative analysis of mechanisms and functions," *Autonomous Agents and Multi-Agent Systems*, vol. 25, no. 1, pp. 1–45, 2012.

[5] Z. Kowalczuk and M. Czubenko, "Computational approaches to modeling artificial emotion–an overview of the proposed solutions," *Frontiers in Robotics and AI*, vol. 3, p. 21, 2016.

[6] S. Stahlke, A. Nova, and P. Mirza-Babaei, "Artificial playfulness: A tool for automated agent-based playtesting," in *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–6.

[7] C. Holmgard, M. C. Green, A. Liapis, and J. Togelius, "Automated playtesting with procedural personas with evolved heuristics," *IEEE Transactions on Games*, 2018.

[8] C. Guerrero-Romero, S. M. Lucas, and D. Perez-Liebana, "Using a team of general ai algorithms to assist game design and testing," in *2018 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, 2018, pp. 1–8.

[9] P. García-Sánchez, A. Tonda, A. M. Mora, G. Squillero, and J. J. Merelo, "Automated playtesting in collectible card games using evolutionary algorithms: A case study in hearthstone," *Knowledge-Based Systems*, vol. 153, pp. 133–146, 2018.

[10] I. Borovikov, Y. Zhao, A. Beirami, J. Harder, J. Kolen, J. Pestrak, J. Pinto, R. Pourabolghasem, H. Chaput, M. Sardari *et al.*, "Winning isn't everything: Training agents to playtest modern games," in *AAAI Workshop on Reinforcement Learning in Games*, 2019.

[11] J. Pfau, J. D. Smeddinck, and R. Malaka, "Automated game testing with icarus: Intelligent completion of adventure riddles via unsupervised solving," in *Extended Abstracts Publication of the Annual Symposium on Computer-Human Interaction in Play*, 2017, pp. 153–164.

[12] T. E. Vos, P. M. Kruse, N. Condori-Fernández, S. Bauersfeld, and J. Wegener, "Testar: Tool support for test automation at the user interface level," *International Journal of Information System Modeling and Design (IJISMD)*, vol. 6, no. 3, pp. 46–83, 2015.

[13] A. M. Memon, M. E. Pollack, and M. L. Soffa, "Automated test oracles for guis," *ACM SIGSOFT Software Engineering Notes*, vol. 25, no. 6, pp. 30–39, 2000.

[14] R. Bellamy, B. John, and S. Kogan, "Deploying cogtool: integrating quantitative usability assessment into real-world software development," in *Proceedings of the 33rd international conference on software engineering*, 2011, pp. 691–700.

[15] C. Katsanos, N. Karousos, N. Tselios, M. Xenos, and N. Avouris, "Klm form analyzer: automated evaluation of web form filling tasks using human performance models," in *IFIP Conference on Human-Computer Interaction*. Springer, 2013, pp. 530–537.

[16] M. C. Desmarais and R. S. d Baker, "A review of recent advances in learner and skill modeling in intelligent learning environments," *User Modeling and User-Adapted Interaction*, vol. 22, no. 1-2, pp. 9–38, 2012.

[17] C. Martinho, I. Machado, and A. Paiva, "A cognitive approach to affective user modeling," in *International Workshop on Affective Interactions*. Springer, 1999, pp. 64–75.

[18] C. Conati and H. Maclaren, "Empirically building and evaluating a probabilistic model of user affect," *User Modeling and User-Adapted Interaction*, vol. 19, no. 3, pp. 267–303, 2009.

[19] I. DIS, "9241-210: 2010. ergonomics of human system interaction-part 210: Human-centred design for interactive systems (formerly known as 13407)," *International Standardization Organization (ISO). Switzerland*, 2010.

[20] J. A. Russell and A. Mehrabian, "Evidence for a three-factor theory of emotions," *Journal of research in Personality*, vol. 11, no. 3, pp. 273–294, 1977.

[21] A. Ortony, G. L. Clore, and A. Collins, *The cognitive structure of emotions*. Cambridge university press, 1990.

[22] R. Plutchik, "A general psychoevolutionary theory of emotion," in *Theories of emotion*. Elsevier, 1980, pp. 3–33.

[23] M. Callejas-Cuervo, L. A. Martínez-Tejada, and A. C. Alarcón-Aldana, "Emotion recognition techniques using physiological signals and video games-systematic review," *Revista Facultad de Ingeniería*, vol. 26, no. 46, pp. 19–28, 2017.

[24] S.-E. Moon and J.-S. Lee, "Implicit analysis of perceptual multimedia experience based on physiological response: a review," *IEEE Transactions on Multimedia*, vol. 19, no. 2, pp. 340–353, 2016.