KEYWORDS Human-Computer Interaction; Usertesting; Playtesting; Game Development; Artificial Intelligence

Artificial Playfulness: A Tool for Automated Agent-Based Playtesting

Samantha Stahlke
Atiya Nova
Pejman Mirza-Babaei
University of Ontario Institute of Technology
Oshawa, Ontario, Canada
samantha.stahlke@uoit.ca,atiya.nova@uoit.net,pejman@uoit.ca

ABSTRACT

Usertesting is commonly employed in games user research (GUR) to understand the experience of players interacting with digital games. However, recruitment and testing with human users can be laborious and resource-intensive, particularly for independent developers. To help mitigate these obstacles, we are developing a framework for simulated testing sessions with agents driven by artificial intelligence (AI). Specifically, we aim to imitate the navigation of human players in a virtual world. By mimicking the tendency of users to wander, explore, become lost, and so on, these agents may be used to identify basic issues with a game's world and level design, enabling informed iteration earlier in the development process. Here, we detail our progress in developing a framework for configurable agent navigation and simple visualization of simulated data. Ultimately, we hope to provide a basis for the development of a tool for simulation-driven usability testing in games.

CHI'19 Extended Abstracts, May 4–9, 2019, Glasgow, Scotland UK

ACM ISBN 978-1-4503-5971-9/19/05.
https://doi.org/10.1145/3290607.3313039

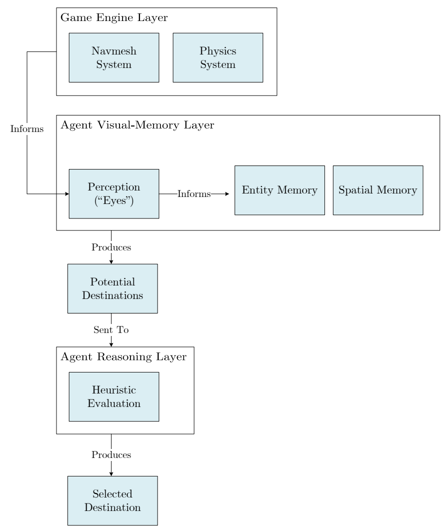

Figure 1: Information flow in the PathOS system. The "Game Engine" layer is external and specific to the development environment used. The perceptual system queries information from the game world, feeding it to the agent's memory model. Together, these subsystems produce a list of candidate destinations which are subject to heuristic evaluation before producing the agent's current goal. This goal is then sent to the engine's pathfinding utility to drive low-level locomotion.

1. INTRODUCTION

Video games are a complex form of media allowing for a rich variety of interactions as players engage with diverse mechanics, characters, and virtual worlds. GUR is a subfield of human-computer interaction (HCI) dedicated to understanding these interactions, and how they shape user experience (UX) in digital games [5]. A common approach applied in GUR is usertesting (or *playtesting*), in which researchers observe and analyze the behaviour of users interacting with game systems to ensure that actual player experience aligns with developer intentions [5]. Throughout the course of game production, multiple rounds of testing can be conducted to validate these questions, allowing developers to iteratively refine their design.

Usertesting is plagued by practical challenges, such as resource efficiency and the difficulty of recruiting participants reflective of a developer's target audience. These difficulties are especially pressing under the time constraints and resource limitations imposed by commercial projects, particularly for independent developers [5]. We are developing a framework to substitute agents driven by artificial intelligence (AI) for human participants in the early stages of game usertesting, to predict player navigation in the virtual world. This work builds on our previous exploration of the design and utility of such a system for agent-based playtesting [10]. By imitating how a human player might become lost, sidetracked, or fail to notice in-game objectives, designers may identify key obstacles to attaining the intended user experience. Additionally, AI agents may be configured to represent specific qualities of individual human players (e.g., aggression, curiosity, etc.) to better reflect a game's target audience. The goal of this utility is not to replace existing human-based evaluation, but rather to enable developers to pursue cost-effective, informed iteration of their designs earlier in the production process.

2. RELATED WORK

AI research has been entwined with game development for several decades, with both games serving as a testbed for the development of AI techniques. Traditionally, the focus of such work is to develop agents which display superior proficiency in their play decisions, rather than genuinely human-like behaviour (e.g., Google's AlphaGo [8]). The imitation of human behaviour in games has been explored as somewhat of a curiosity, and suggested as a means to develop more convincing and engaging in-game characters or opponents [7]. AI has also been applied in player modelling, the development of computational models to classify, describe, and predict user behaviour [13].

In game evaluation, simulation-driven testing has been explored as a means to validate the *playability* of game content (i.e., whether a level is possible to play). Agent-based approaches have been used to validate playability within level design tools, for instance, to verify both human- and procedurally-authored content [9]. AI has also been investigated as a means to develop automated quality assurance

Heuristic	Play Behaviours
Adrenaline	Risk-taking behaviour characterized by seeking moments of vertigo, environmental extremes, and challenges.
Aggression	Dominant behaviour characterized by seeking combat.
Caution	Conservative behaviour characterized by avoiding combat and prioritizing resources for self-preservation.
Completion	Achievement-oriented behaviour characterized by a need to complete in-game objectives and seek out collectibles.
Curiosity	Exploration-oriented behaviour characterized by a need to see everything the game has to offer.
Efficiency	Practically-oriented behaviour characterized by swift navigation to a game's mandatory objectives.

Table 1: In-development heuristics used by PathOS agents to evaluate potential destinations (current implementation is not yet final).

[1]https://unity3d.com/

(QA) processes for games (e.g., the Cicero framework [4]). AI-driven approaches have also been explored in usability and user experience evaluation. Tremblay et al., for instance, developed a pathfinding tool to predict player navigation in stealth-based games [12]. The resultant projected trajectories can be used by developers to verify whether a level's design supports the intended player path early in the iterative design process. Holmgård et al. explored the use of single-layer neural networks to simulate individual variations in gameplay style between different users [3]. Despite these advances, agent-driven playtesting remains relatively under-explored in the literature, further motivating our investigation of automated playtesting as a means to improve user experience.

3. PROTOTYPE DESIGN AND DEVELOPMENT

We are currently developing a framework for simulating human navigation in game environments, called PathOS, with Unity[1] game engine. In this paper, we describe the development of our configurable agents capable of navigating a virtual environment and present initial sample results for three different agents in a small testbed scenario.

3.1. Agent Perception and Memory

Since real users do not have access to a complete and correct model of a game's world, PathOS agents simulate the restrictions imposed by human perception. Agents maintain a camera serving as the player's "eyes", and commit entities to a simulated short-term memory, which is subject to degradation over time to mimic "forgetting". Our goal is to enhance this functionality to more accurately reflect the mechanics of visual short-term memory in humans (e.g., as outlined in [11]). By building an encoding of level topography in a tile-based "mental map", agents also maintain a form of spatial memory. This map is generated by querying Unity's navigation mesh (or *navmesh*), which maintains a physics-based representation of world geometry.

Since agents maintain an incomplete representation of the game world subject to information decay, they are capable of replicating navigation behaviours such as wandering, backtracking, or becoming "lost". This quality makes it possible for the system to more accurately estimate how real players will engage (or fail to engage) with a game's world. Information from an agent's visible entity set, entity memory, and spatial memory is used to generate a list of prospective destinations, which may be explicit objectives (i.e., entity locations) or suggestions for exploration (i.e., unobstructed directions for travel). A complete overview of the flow of information between agent perception, memory, reasoning, and game engine systems is presented in Figure 1.

3.2. Agent Reasoning and Locomotion

The logic used to select a goal from available destinations is based on a configurable model of player motivation, with dimensions selected based on a review of literature surrounding player psychology

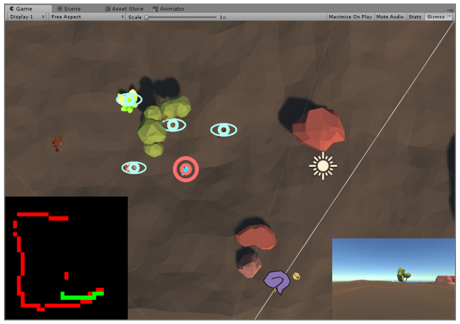

Figure 2: Simulation UI overlay in the PathOS framework. The agent's "mental map" is rendered in the lower left, and current POV is shown in the lower right. In-game entities are flagged as visible (cyan eye gizmo), in memory (purple brain gizmo), or targeted (red bullseye gizmo). All UI elements are toggleable and can be scaled to suit the viewer's needs.

and gameplay behaviour (e.g., [2, 6]). In PathOS agents, these qualities are represented as weighted heuristics used to evaluate the desirability of a given destination. For instance, the *Caution* heuristic prefers avoidance of enemies or environmental hazards, and is drawn to survival resources (e.g., health packs). A list of the currently developed heuristics with brief descriptions is provided in Table 1. To support the heuristic evaluation of potential destinations, we created a tagging system for developers to record the nature of in-game entities (e.g., "enemy", "survival resource" "POI", etc.) based on established game design patterns (e.g., [1]).

In evaluating a given destination, the agent will calculate a combined score based on entities that will be encountered at or en route to that destination, ultimately choosing the destination with the highest score. The agent's perceptual system is configured to re-process its surroundings every few frames, with a corresponding reassessment of available destinations, to simulate "on-the-fly" decision-making. Agents can be configured to represent particular player profiles by tuning the weighting of heuristics to reflect specific play-styles.

3.3. Data Collection Using Agent-Based Playtesting

Agents operate in Unity's Editor, allowing developers to manipulate their view of the scene as it is traversed. While the simulation is active, an overlay UI (see Figure 2) is displayed to assist designers in understanding agent reasoning at-a-glance, as well as allowing for debugging of unexpected behaviour. Before running the simulation, a logging system is activated to record an agent's trajectory over time, which can later be visualized in the Editor for reviewing in-world. We performed a simple case study to demonstrate data-collection using agent-based behavioural variation within three different agent profiles (aggression, curiosity, and completion). The results visualizing the pathways taken by agents with different simulated play-styles is presented in Figure 3. This system also supports basic visual customization, such as filtering and aggregation of multiple playtraces (the paths given in Figure 3 are screenshots showing this visualization).

4. DISCUSSION

Integration of agent-based testing into a commercial development cycle would vary significantly depending on the nature of a given project. For PathOS, we envision the framework as having the greatest utility early in the development process, to give developers a coarse indication of how players might experience the virtual world. After creating an initial level design, the framework could be deployed to identify likely navigation patterns of players within that design. Unintended or unexpected features of these patterns may then be used to identify and remedy issues related to world design. For instance, if a developer notes that an intended objective is not reached by the majority of agents, they can review a simulated playtrace to identify whether this is caused by poor visibility, a lack of points-of-interest in the surrounding area, or some other factor (e.g., a high density of hazards

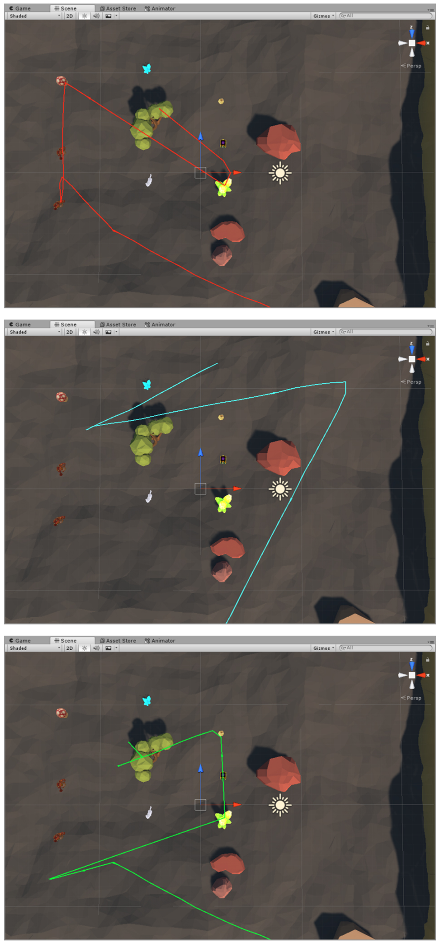

Figure 3: Paths taken by three different agents in the PathOS framework. Top: an agent tuned with high aggression and low caution immediately confronts "enemy" entities before moving on to game collectibles. Middle: an agent tuned with high curiosity explores the edges of the map to reveal more terrain. Bottom: an agent tuned to prioritize completion focuses on game collectibles.

discouraging cautious players from exploring a given region). Based on this insight, the design can then be iteratively refined and re-tested to validate the intended experience.

Identifying basic issues early in the development process prevents these issues from propagating to later testing with human users, where they might interfere with a researcher's ability to probe deeper questions relating to UX. For example, if the goal of a late-stage usertest is to assess players' experience with a game's narrative, but players miss interactions with key characters due to basic issues with level design, then the ability of the test to accurately investigate the core research question has been compromised. Testing early with artificial agents to identify these basic issues could help to prevent these situations. Furthermore, the pursuit of continuous, informed improvement from a very early stage of development may contribute to overall improved product quality and a better experience for the end user.

5. Limitations and Future Work

Though the PathOS framework is being developed with generality and accessibility in mind, the tool will not be automatically suitable for all games. The rich diversity of games available and under development spans an astonishing collection of genres, from 2D arcade-style shooters, to role-playing games, visual novels, and nearly every imaginable point in between. It is also difficult to make assertions regarding the commercial viability of PathOS without having conducted an evaluation of its utility. Thus our immediate next step will be to conduct an expert evaluation of the framework with members of the industry, followed by user studies with game developers to assess usability and gather feedback on any additional features which should be introduced. Additionally, we plan to conduct studies comparing the behaviour of human users in simple game scenarios with that of PathOS agents configured to represent users with the same behavioural profile.

Moreover, the current iteration of the framework is somewhat limited by its nature as an expert system; though this is advantageous in terms of model transparency and eliminates the need for training data, a machine learning system would be arguably better-suited to reproducing gameplay behaviour by learning from real players. However, the scale of gameplay data required for adequate training may prove difficult to obtain. Future work might explore the synthetic augmentation of data from small-scale playtests with random variation to address this challenge. Alternatively, such an endeavour might present a promising opportunity for collaboration between academic and industry researchers to operate on datasets obtained via remote telemetry from large-scale commercial games.

6. CONCLUSION

In this paper, we detailed our progress in the development of an agent-based game testing framework informed by existing research on player psychology and human cognition. Our goal is to reduce the burden of conducting usability tests with human participants and enable informed iterative design earlier in the development cycle. Our current prototype supports the simulation of player navigation through AI agents which can be configured to mimic diverse, customized user profiles. In future work, we plan to evaluate and extend this framework to improve its utility for developers, with the ultimate goal of releasing it as an open source tool to support game researchers and independent game creators.

ACKNOWLEDGMENTS

Samantha Stahlke would like to thank Svetlana Stahlke for her mentorship and support. This work has been supported by UOIT and the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] Staffan Bjork and Jussi Holopainen. 2005. *Patterns in Game Design.* Charles River Media.
[2] Benjamin Cowley and Darryl Charles. 2016. Behavlets: a method for practical player modelling using psychology-based player traits and domain specific features. *User Modeling and User-Adapted Interaction* 26, 2 (June 2016), 257–306. https://doi.org/10.1007/s11257-016-9170-1
[3] C. Holmgård, A. Liapis, J. Togelius, and G. N. Yannakakis. 2014. Evolving personas for player decision modeling. In *2014 IEEE Conference on Computational Intelligence and Games.* 1–8. https://doi.org/10.1109/CIG.2014.6932911
[4] T. Machado, D. Gopstein, A. Nealen, O. Nov, and J. Togelius. 2018. AI-Assisted Game Debugging with Cicero. In *2018 IEEE Congress on Evolutionary Computation (CEC).* 1–8. https://doi.org/10.1109/CEC.2018.8477829
[5] Pejman Mirza-Babaei, Naeem Moosajee, and Brandon Drenikow. 2016. Playtesting for Indie Studios. In *Proceedings of the 20th International Academic Mindtrek Conference (AcademicMindtrek '16).* ACM, New York, NY, USA, 366–374. https://doi.org/10.1145/2994310.2994364
[6] Lennart E. Nacke, Chris Bateman, and Regan L. Mandryk. 2014. BrainHex: A neurobiological gamer typology survey. *Entertainment Computing* 5, 1 (Jan. 2014), 55–62. https://doi.org/10.1016/j.entcom.2013.06.002
[7] N. Shaker, J. Togelius, G. N. Yannakakis, L. Poovanna, V. S. Ethiraj, S. J. Johansson, R. G. Reynolds, L. K. Heether, T. Schumann, and M. Gallagher. 2013. The turing test track of the 2012 Mario AI Championship: Entries and evaluation. In *2013 IEEE Conference on Computational Inteligence in Games (CIG).* 1–8. https://doi.org/10.1109/CIG.2013.6633634
[8] David Silver, Aja Huang, Chris J. Maddison, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529, 7587 (Jan. 2016), 484–489. https://doi.org/10.1038/nature16961
[9] Gillian Smith, Jim Whitehead, and Michael Mateas. 2010. Tanagra: A Mixed-initiative Level Design Tool. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games (FDG '10).* ACM, New York, NY, USA, 209–216. https://doi.org/10.1145/1822348.1822376
[10] Samantha N. Stahlke and Pejman Mirza-Babaei. 2018. Usertesting Without the User: Opportunities and Challenges of an AI-Driven Approach in Games User Research. *Comput. Entertain.* 16, 2 (April 2018), 9:1–9:18. https://doi.org/10.1145/3183568
[11] J. Jay Todd and René Marois. 2004. Capacity limit of visual short-term memory in human posterior parietal cortex. *Nature* 428, 6984 (April 2004), 751–754. https://doi.org/10.1038/nature02466
[12] Jonathan Tremblay, Pedro Andrade Torres, Nir Rikovitch, and Clark Verbrugge. 2013. An Exploration Tool for Predicting Stealthy Behaviour. In *Artificial Intelligence in the Game Design Process 2: Papers from the 2013 AIIDE Workshop.*
[13] Georgios N. Yannakakis, Pieter Spronck, Daniele Loiacono, and Elisabeth André. 2013. Player Modeling. In *Artificial and Computational Intelligence in Games*, Simon M. Lucas, Michael Mateas, Mike Preuss, Pieter Spronck, and Julian Togelius (Eds.). Dagstuhl Follow-Ups, Vol. 6. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 45–59. https://doi.org/10.4230/DFU.Vol6.12191.45