Usertesting Without the User: Opportunities and Challenges of an AI-Driven Approach in Games User Research

SAMANTHA N. STAHLKE and PEJMAN MIRZA-BABAEI, University of Ontario Institute of Technology

The use of human participants in game evaluation can be costly, time-consuming, and present challenges for constructing representative player samples. These challenges may be overcome by using computer-controlled agents in place of human users for certain stages of the usertesting process. This article explores opportunities and challenges in the use of behavioural modelling to create independent "user" agents driven by artificial intelligence (AI). We highlight the utility of imitating cognitive processes such as spatial reasoning, memory, and goal-oriented decision-making as a means to increase the viability of independent agents as a tool in usertesting. Specifically, we investigate the possible design and use of proxy AI "users" that mimic human navigational behaviour to assist in the evaluation of level designs. Ultimately, we propose that a configurable population of AI players can provide a data-rich supplement to current approaches in games user research.

CCS Concepts: • **Human-centered computing** → **HCI design and evaluation methods**; **User models**; User studies; • **Computing methodologies** → **Artificial intelligence**; **Model development and analysis**; *Interactive simulation*;

Additional Key Words and Phrases: Artificial intelligence, games user research, computer modeling

ACM Reference format:
Samantha N. Stahlke and Pejman Mirza-Babaei. 2018. Usertesting Without the User: Opportunities and Challenges of an AI-Driven Approach in Games User Research. *ACM Comput. Entertain.* 16, 2, Article 9 (April 2018), 18 pages.
https://doi.org/10.1145/3183568

1 INTRODUCTION

Games User Research (GUR) is an emerging field focused on applying user evaluation methods from psychology and Human-Computer Interaction (HCI) to the understanding, evaluation, and improvement of user experience (UX) in interactive media [20]. A fundamental component of GUR methodology is usertesting (sometimes referred to as playtesting), whereby user researchers aim to understand player behaviour, emotions, and experience by collecting and analyzing data from players interacting with a prototype or pre-released version of a game [8]. This data takes many different forms and can be collected from a number of sources, such as questionnaires, player interviews, video recordings of gameplay, metrics embedded into the application to detect in-game events, and physiological sensors capable of providing insights into changes in a player's emotional state. The key aim of usertesting is to compare actual player behaviour and experience

Authors' addresses: S. N. Stahlke and P. Mirza-Babaei, UXR Lab, University of Ontario Institute of Technology, 2000 Simcoe Street North, Oshawa, ON, L1H 7K4; emails: {samantha.stahlke, pejman}@uoit.ca.

https://doi.org/10.1145/3183568

with the experience intended by designers, with the goal of bringing the eventual game experience closer to the designers' vision.

Today, usertesting conducted with "human" users is a central component of GUR in both commercial and academic contexts [22]. The benefits of usertesting are well-known, as it allows designers to examine the interactions and experiences of individuals external to the development team. This process identifies potential issues with the game and provides developers with an opportunity to improve products before launch [19]. Since different users will often have differing experiences in their interaction with a given medium, the quantity and diversity of participants can be key factors in determining the value of a given usertesting endeavour. This is especially true for video games, as their evaluation is complicated by the unique nature of the medium; when contrasted with other software or forms of entertainment, games focus on experience rather than outcomes, impose vastly differing constraints on users, and offer more open and varied interactions [24]. Moreover, organizing usertest sessions in practice can be challenging and expensive, particularly for small development teams [21]. Organizing usertest sessions is a complex process, often involving multiple preparatory stages before data collection begins. These steps may include the formalization of usertest objectives, assessing the suitability of current game builds for evaluating these objectives, designing a test session, and administrative planning to ensure the availability of researchers, participants, and an acceptable testing environment [29].

A key objective of GUR is understanding the relationship between user action, gameplay, and player responses. A method of particular interest used in the analysis of gameplay is game metrics, whereby an application is extended with the ability to flag and record certain events or game data, such as player navigation, objective fulfillment, game deaths, and reward collection [17]. Such metrics can be used to search for patterns in the gameplay of large user populations, flag potential issues relating to difficulty and usability, or serve as a basis for determining areas of interest for further qualitative study. Although modern advances in game analytics and sensor technology have made the acquisition of game and player metrics more accessible, the difficulty of coordinating participants remains an open challenge for developers. As an additional challenge, usertesting is an inherently iterative process, often demanding that developers conduct multiple rounds of testing to achieve the desired experience in the final product. The necessity of iterative testing is driven by two key factors: first, to examine if changes made in response to previous testing have positively impacted player experience; second, to allow for early testing rounds focused on broad aspects of product quality (e.g., regarding usability, learning, and navigation) before evaluating more fine-grained aspects of the complete gameplay experience. Multiple rounds of testing, coupled with the difficulties associated with coordinating individual test sessions, can drive an escalation in the complexities and cost of conducting user tests throughout the game development cycle. These issues pose a key challenge in GUR, as many studios, particularly independent developers, may be unable to afford the resources associated with comprehensive usertesting.

We endeavour to answer this challenge by discussing the possibility of using artificial intelligence (AI) as an alternative to human users in GUR, while preserving the human factors that create value in usertesting. Specifically, we highlight opportunities and challenges in applying player modelling techniques to partially automate the evaluation of level design in video games, allowing for the collection of game navigation metrics from AI-driven agents designed to mimic the tendencies of human users. With this research, we aim to reduce the need to organize human participants in the early stages of testing, empowering developers to iterate and evaluate their designs more quickly and inexpensively, perhaps simulating thousands of users in the span of a few hours, rather than a few dozen users over the course of a few weeks. In this article, we propose a theoretical basis for a framework capable of simulating user behaviour by leveraging techniques in cognitive and player modelling, forming the basis for our future work in implementing the

framework for use by game developers and researchers. Our ultimate goal is to develop a tool capable of imitating a diverse range of player types that can be easily applied with minimal modification to any game involving spatial navigation. Such a tool may assist in the identification and evaluation of issues relating to level geometry, player disorientation, and the layout of game objectives before developers involve real players, allowing in-person sessions to focus on more complex and nuanced aspects of user experience.

2 BACKGROUND

Understanding the dichotomy between intended and actualized player experience is a key goal in the GUR field. As discussed in Section 1, this understanding is typically achieved by having a representative set of players interact with the game during the development process, providing developers with comments and gameplay data. However, collecting and analyzing gaming interactions can be a highly complex task. Data collection can demand meticulous manual work from the usertesting team or require the integration of utilities that are often game or platform-specific. Moreover, game developers must often work under non-disclosure agreements from publishers, which complicates the process of exposing the game to individuals external to the development team. Beyond these initial obstacles, the analysis of player behaviour and game interactions carries its own challenges in providing a comprehensive understanding of player experience.

2.1 Tools in GUR

As outlined in Section 1, GUR professionals have a wide array of research techniques at their disposal, including gameplay observation, interviews, questionnaires, and physiological measures, to name a few. It is often the case in GUR that researchers must comb through and analyze vast amounts of disparate data; furthermore, differing study designs and test objectives require different methodologies [15]. This means that mixed-method approaches are relatively commonplace, and thus, researchers are often left with multiple separate data sets that must be recombined. Each data type conforms to certain modalities and must be combined from a number of different sources, posing unique challenges in working with multimodal data [38].

In both industry and academia, software tools are essential for GUR to administer, collect, integrate, analyze, and report on games' and players' data. Examples of commercial tools with features useful for these purposes are screen/video capturing (e.g., ScreenFlow[1]), transcribing videos (e.g., Transana[2]), coding (e.g., Nvivo[3]), processing physiological data (e.g., Biograph Infiniti[4]), and administering questionnaires (e.g., SurveyMonkey[5]) [28]. Although off-the-shelf commercial tools such as these are often effective in their respective strengths, using them in combination is not always the most efficient means to employ mixed methods of data collection; having to maintain several data interfaces and constantly switch between each tool is error-prone and counter-productive [38]. Moreover, since off-the-shelf tools often aim to attract a wide audience beyond game developers, essential features needed for game evaluation may ultimately be missing. In response, many in-house tools have emerged in game development and other related domains. Researchers and practitioners have developed new tools to enable detailed collection and analysis of multimodal data, including physiological responses [18], self-reports [27], and in-game telemetry data [26].

Although various in-house tools have already been developed to assist game developers and researchers with data collection and analysis, the majority are focused on data collection

[1] https://www.telestream.net/screenflow/overview.htm.
[2] https://www.transana.com/.
[3] http://www.qsrinternational.com/nvivo-product.
[4] http://thoughttechnology.com/index.php/biograph-infiniti-software-upgrade.html.
[5] https://www.surveymonkey.com/.

alone, rather than automation of the usertesting process itself. To date, most automated testing approaches focus solely on identifying issues with basic usability elements, such as user interface (UI) responsiveness and browser compatibility in web applications (e.g., SmartBear,[6] Helium[7]). However, advancements in AI and player modelling mean that we may now collectively explore the possibilities of unexplored approaches, such as automated AI-driven usertesting in game development. To this end, we aim to discuss the opportunities and challenges of applying AI techniques and computer modelling to user experience evaluation in games.

2.2 Computer Modelling in Games

Computer modelling has been used to estimate aspects of player experience in video games by approximating dimensions of player emotion based on game metrics [35]. Most work in player modelling focuses on extrapolating human responses, such as frustration, confusion, or delight, from actions taken by human players in-game. In the realm of design, these techniques have been applied to enhance player experience, for example, by creating believable AI meant to seem more realistic, or more "human" [36]. Player modelling has also been applied in the creation of adaptive difficulty algorithms, by analyzing players' self-reported gaming experience and motivations coupled with performance analysis to dynamically adjust game challenge [37]. Player modelling and profiling techniques have also been leveraged in GUR, where such tactics have been used, for example, in supplementing the analysis of playtesting data [35]. However, such methods are more commonly applied outside the field of games research, in exploring the behaviour of users on the web. Data analysis techniques have also been applied in modelling online user behaviour. For instance, Vieira [32] investigated the use of deep learning algorithms to predict the buying habits of web users based on aggregations of existing eCommerce data. Adar et al. [1] developed models of web searching behaviour based on community activity regarding topics of popular interest, such as current events.

Recent advancements in AI have yielded improvements in the simulation of human behaviour and interaction. Convolutional neural networks, for example, have been trained to play arcade games such as Pong[8] and Space Invaders[9] with a proficiency level comparable to that of human players [14]. Game-playing AIs designed to resemble human behaviour in platforming games have been developed, with applications, for instance, in the evaluation of procedural level content [23]. Predictive player modelling is another interesting application of AI in GUR, whereby researchers attempt to predict certain patterns in player behaviour, such as play-session length or likelihood of user retention. Mahlmann et al. [13], for instance, used supervised learning algorithms trained using gameplay metrics (e.g., navigation data, player deaths, rewards collected) to predict player progression and total play-time. AI-assisted tools have also been developed for the simulation of 2D gameplay, as well as the review and evaluation of human or computer-simulated play sessions [12]. However, to our knowledge, the practical use of AI as standalone agents in general usertesting is a largely unexplored concept.

3 DESIGNING AN AI FRAMEWORK FOR USERTESTING

To investigate the utility of AI-driven agents in game usertesting, we explore the possibility of developing GUR-oriented frameworks informed by existing models of human cognition. Such frameworks could allow for the development of configurable, computer-controlled agents suitable as stand-ins for human participants in the early-stage evaluation of game and level design.

[6] https://smartbear.com/product/testcomplete/web-testing/.
[7] https://heliumhq.com/.
[8] Atari, 1972.
[9] Taito, 1978.

In particular, we examine the potential of AI "players" to mimic human navigation in game environments. To that end, we focus on the high-level design of customizable models for human memory, reasoning, and instinct as pertaining to spatial navigation. This section will begin by reviewing existing applications of computer modelling in games, before discussing techniques specific to simulating human navigation. Following this, we explore the potential design of an AI testing framework intended to drive agents capable of simulating human navigation in virtual worlds and games. Finally, we will investigate aspects of the proposed framework relating to simulated player profiles, memory, and decision-making.

3.1 Navigation Modelling

Navigational mechanisms and pathfinding algorithms have been the subject of research in human cognition [34], robotics [9], and game AI [36]. Human navigation is understood to depend on aspects of both environmental context (i.e., sensory cues) and cognitive mechanisms pertaining to memory, goal orientation, and interpretation of spatial information [34]. Furthermore, in virtual environments, human navigation ability with tasks relating to landmark and route knowledge is comparable to performance in the real world [31]. Therefore, we suggest that modelling approaches for AI-driven usertesting should focus on computational representations of human sensory perception and cognitive interpretation.

Navigation modelling has an established history of use for AI in games, with a number of pathfinding algorithms available for automating the movement of computer-controlled game agents. Such algorithms typically operate by calculating an optimal path based on all available spatial data [2]. Other work has focused on pathfinding inspired by human navigation, relying on the simulation of landmark recognition to navigate a graph network [33].

We propose a framework intended to function atop existing systems for the mathematical and physical evaluation of level geometry. Such systems are commonly referred to as navigational meshes or "navmeshes," which contain a mathematical representation of level terrain suitable for physics calculations and queries such as raycasting. Many commercial game engines, such as Unity[10] and Unreal Engine,[11] include this functionality as part of the core engine application program interface (API), providing a flexible basis for the programming of AI-driven navigational agents. Our proposed design consists of two key layers designed to work with the navigational systems present within a given game engine or development framework (see Figure 1). These two layers are the non-omniscient player model, analogous to a "brain" or decision-making machine, and an intermediary communicator, analogous to the sensory organs. The communicator is responsible for filtering information available to the player model from the navigational data (navmesh), based on, for example, player point-of-view visibility. The player model maintains the artificial player's memory, and makes navigational decisions based on information from memory, player characteristics (e.g., play style), and "sensory information" from the communicator. When combined, these layers form a navigational entity, or agent, intended to mimic the behaviour of a human user navigating in the game space. This agent will act based on a simple procedure meant to simulate the way in which players process information to make decisions (see Figure 2).

The core functionality of this design resides within the player model, or "brain," of the system. In the interest of more accurately imitating user behaviour, it is important that agents model the limitations of human users. While traditional navigation modelling focuses on the generation of optimal paths, a GUR-centric approach must consider fallible components of human perception and reasoning to simulate issues with player navigation, such as getting lost, missing key areas,

[10] https://unity3d.com/.
[11] https://www.unrealengine.com/.

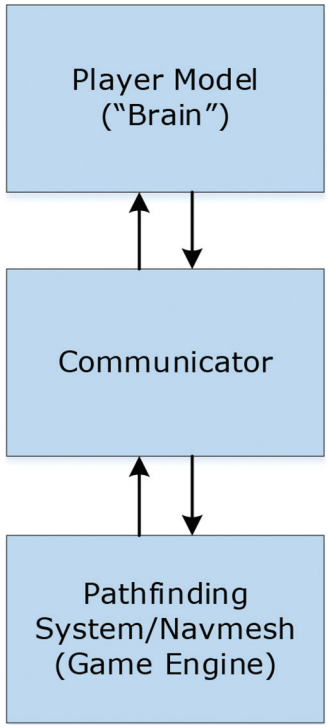

Fig. 1. Logical layers of an AI-driven player navigation agent.

or losing track of game objectives. To that end, we suggest that AI-driven usertesting approaches should focus primarily on modelling the processes and limitations of human memory and logical reasoning.

3.2 Human Memory Model

Our design objective in developing a player memory model is to generate a selective store of information that may improve agents' abilities to mimic the decision-making strategies of real human users, as compared to the decisions of omniscient agents. The simulation of human information processing and retrieval has been an area of research interest for several decades, with several proposed models comparing the process to data storage and retrieval in computing [10]. Recent AI-based models of human brain functionality have successfully replicated phenomena relating to the limits of human short-term memory [7], i.e., the ability to retain a certain volume of information obtained in quick succession. The selection of competing information for commitment to memory is understood to rely on the prioritization of data relevant to an individual's current goals [11]. Furthermore, remembered information can be susceptible to mutation over time, as conflicting stimuli may disrupt the accuracy of existing memories [4]. An accurate model of player memory should therefore account for these factors in determining an agent's ability to retain spatial and game information.

To mimic the aspects of human memory relevant to in-game navigation, agent memory should track both past actions (e.g., paces in a particular direction, turns, obstacle encounters) and spatial information (e.g., barriers, corridors visible from particular areas) to construct a simulated mental map of player surroundings. Selection of information for commitment to agents' long-term memory may be determined by a number of factors:

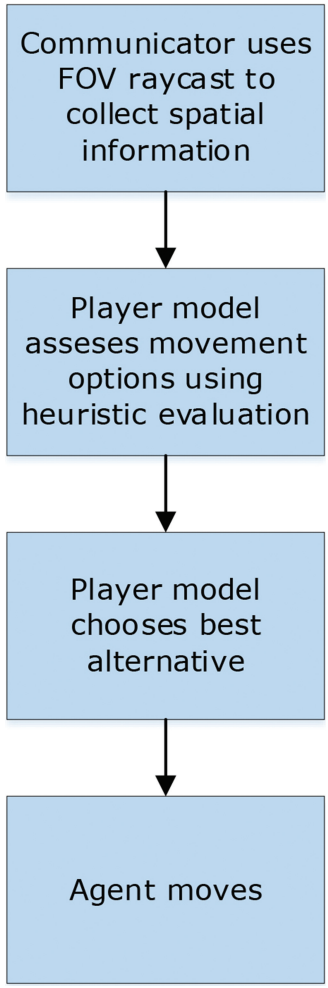

Fig. 2. Control flow overview for navigation decision-making of AI player models.

Information Volume. The volume of information processed by the agent at any time will depend on the complexity of the game environment and the speed at which the agent is travelling. The proportion of information committed to memory correlates inversely with agent speed and environmental complexity—i.e., an agent will remember a simple area passed through slowly with higher fidelity than a complex area passed through quickly.

Information Relevance. The relevance of a particular piece of information will be determined based on its potential to help the agent reach an in-game goal. For example, the position of a corridor in the direction of a known level objective will be assessed as more relevant than the position of a visibly dead end. Information with a higher computed relevance will be prioritized in the selection of memories for long-term storage.

Simulated Player Profile. To account for the improved proficiency and navigational abilities of more experienced players, agents with a higher simulated experience level will be capable of committing a larger proportion of accurate information to memory. Furthermore, known

discrepancies in navigational ability arising from demographic factors including age and gender [6] may be used to more accurately simulate diversity within a given population of AI "users."

Individual "memories" of a particular action or spatial feature may be defined by qualities such as the strength or accuracy of the impression made, which can fade over time depending on the agent's simulated experience level and the number of times that a relevant stimulus has been encountered. Whenever an agent is confronted with the opportunity to make a new navigational decision, such as adjusting movement speed or changing direction, only information available from the agent's viewpoint (i.e., visible surroundings) and committed memories should be considered. We expect that this may lead agents to make non-optimal decisions in unfamiliar territory when compared to an omniscient pathfinding algorithm, analogous to the process by which new players may become lost or 'wander' until confident in their surroundings.

3.3 Simulated Reasoning and Instincts

In any given game navigation scenario, players may have a number of factors to consider before making their next move, such as current objectives, hazards or enemies, and a desire to explore. Simulated agents should therefore evaluate available navigational alternatives based on heuristics such as alignment with game objectives (i.e., might this pathway lead to an in-game goal?), potential for danger (i.e., are enemies visible along this pathway?), potential for discovery (i.e., is this new territory?), and number of previous traversals. To calculate each alternative's heuristic scores, agent reasoning may consider patterns of action available in memory (e.g., having travelled in the same direction for several paces), information given about objectives (e.g., in-game compass indicating goal is to the north), and spatial cues (e.g., placement of walls). The evaluation and prioritization of heuristic values would then depend on a number of factors stemming from logical interpretation and simulated player attributes:

Goal Prioritization. A common technique used in guiding AI agents is Goal-Oriented Action Planning (GOAP), which has been used to enhance pathfinding mechanisms in both robotics [9] and game AI [36] by using known objectives to guide action and navigation. For our purposes, the goal-oriented heuristic score of a navigational alternative will depend on whether pursuing a given path brings agents spatially closer to one or more known in-game objectives. The logic of the player model will then dictate that pathways with a high score in the goal alignment heuristic should be prioritized for investigation. The weighting of this factor could depend on simulated play-style traits such as danger aversion or a desire to explore before moving on, which may override an agent's tendency to efficiently seek out in-game objectives.

Experience Level. Agents with a higher level of simulated experience may emulate the confidence and proficiency of more skilled players by reducing the importance of the danger heuristic, and disregarding paths that have been previously explored multiple times. Furthermore, agents with more experience will be compelled to travel faster, and spend less in-game time on the evaluation of their surroundings.

Play-Style. Agents will also prioritize potential alternatives based on simulated player type, demographics, and personality characteristics such as aggression and curiosity. As an example, an agent intended to stand in for an explorer-type player with natural curiosity will be more keen to explore areas with high potential for new discoveries, for example, corridors that open into multiple new pathways.

The reasoning aspect of our proposed design aims to account for how personality and past experience might influence navigational behaviour, by emulating, for instance, a cautious and inexperienced player's tendency to favour investigating safe, open spaces as opposed to potentially dangerous narrow corridors. Aspects of player aggression, curiosity, and other traits governing

play-style may be defined more simply by assigning preset values according to designer-defined player types based on classifications such as the HEXAD scale [30] or Bartle player types [3]. However, in a holistic gaming environment, players are confronted with a vast quantity of information beyond spatial cues, which may also be unique to any given game. Processing this additional information in a generalizable fashion will inevitably pose challenges in the design and implementation of a more complete decision-making model. Furthermore, an inability to account for the full impact of mood and emotions in player reasoning may prove to limit the depth of insights available from a navigation-centric approach.

3.4 Data Logging and Visualization

To increase the viability of AI-driven frameworks as usertesting tools, considerations must be made for the logging and visualization of rich navigational and behavioural data. Developing such systems for data collection and analytics may be informed by the design of existing tools created for traditional human usertesting. Such tools may integrate features pertaining to visualizing player paths, tagging in-game events, and recording users' emotional responses [[5, 19]. For an AI-based system, useful features may include, for example, the ability to review individual and average trajectories for groups of agents, identify hotspots of heavy traffic or getting lost, and provide logs of individual decisions made by the model. Decision logs may be timestamped and linked to specific points in "user" paths, allowing developers to identify, for example, the decisions that could lead to players becoming lost in a particular area. In doing so, the depth and richness of collected data may be augmented to provide researchers with a more complete record of simulated gameplay sessions. As the complexity of an AI agent's behaviour increases with the integration of additional interactions such as those described in Section 5.1, so too will the demands on any associated tools for data collection and analysis. This may complicate the ability of researchers to visualize and interpret potentially unwieldy data sets.

4 USE CASE: CONDUCTING USER RESEARCH WITH AI PLAYERS

Once a system for simulating the behaviour of AI players has been established, developers are left with the task of applying this framework in a testing environment. In this section, we present a theoretical use case for the proposed framework in the context of evaluating prototype level design for a first-person shooter (FPS) game. Consider the following hypothetical scenario, in which a development team has completed a game prototype for a basic FPS with a simple progression system, where players must traverse levels filled with potential hazards and enemies to complete in-game missions. Each level contains a series of objectives that players must reach in sequence to move forward successfully. Having prototyped several different level layouts, the designer wishes to coarsely evaluate their effectiveness before selecting and refining levels for further development. Their core objective is to identify issues such as players becoming lost or missing key in-game areas, by collecting information about players' traversal through various levels—data such as time spent in individual areas, order of objectives visited, avoidance of obstacles, and time taken to explore an entire level. An established GUR approach may involve conducting preliminary usertesting with human players, using game metrics to record traces of player pathways taken in-game and observing gameplay to look for unexpected behaviours. However, this process is time-consuming, costly, and cannot feasibly be re-run every time changes are made to the design. The AI-driven framework described in Section 3 is thus an attractive, minimally resource-intensive solution for such an assessment; the use of AI navigational agents can be used to generate navigational datasets, such as path visualizations, which are directly comparable in nature to that obtained from traditional metrics-based approaches.

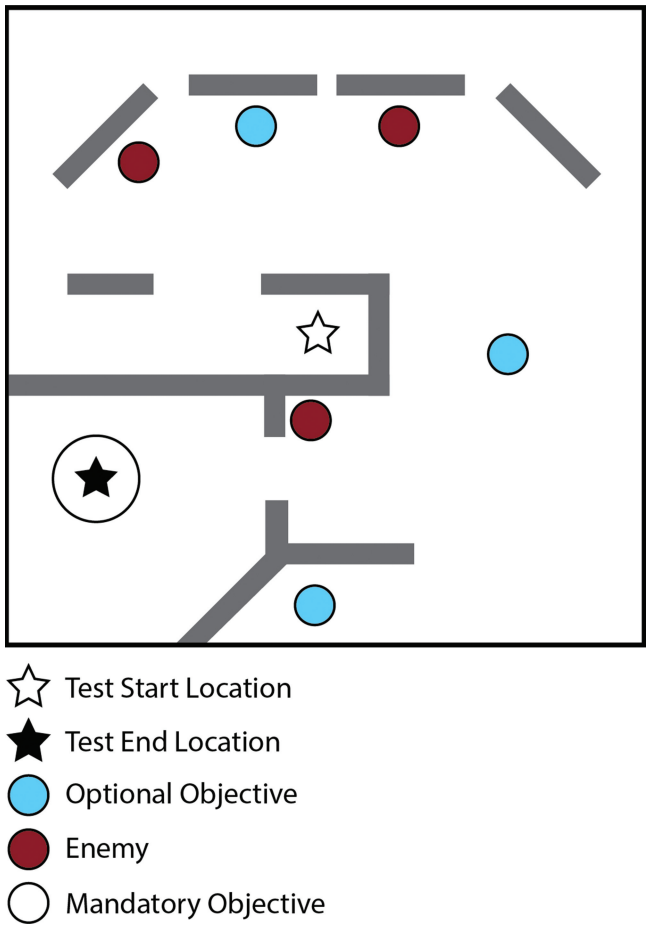

Fig. 3. Example level design showing areas of interest representing level objectives, enemies, and end goal.

4.1 Setting up the Testing Environment

Depending on the nature of the development environment, integration of the framework will demand varying levels of technical overhead. For our purposes, we will assume that the developer is working with a plugin-type implementation of the framework (see Section 5 for more details). As a starting point, the developer will need to supply the framework with information regarding player locomotion, such as movement speed and axes of movement. They may also specify a list of objects or obstacles to be classified as potentially dangerous (e.g., enemy characters or environmental hazards such as fire) or collectibles (e.g., power-ups, ammunition, or health packs). For each level to be tested, the developer may indicate one or more areas to be considered mandatory or optional player objectives (e.g., mission areas of interest, "drop points," and so on), and an end condition for each testing scenario (e.g., a simulated time limit or reaching a particular objective area). An example of regions of interest mapped to a simple level layout is given in Figure 3.

With key context-specific details defined for each level, the developer may move on to chart expected or ideal player paths through the level as shown in Figure 4, for later comparison with AI player trajectories. This exercise helps to clarify the intended course of player action, serving as a starting point for the evaluation of results. Charting the ideal player path may be considered

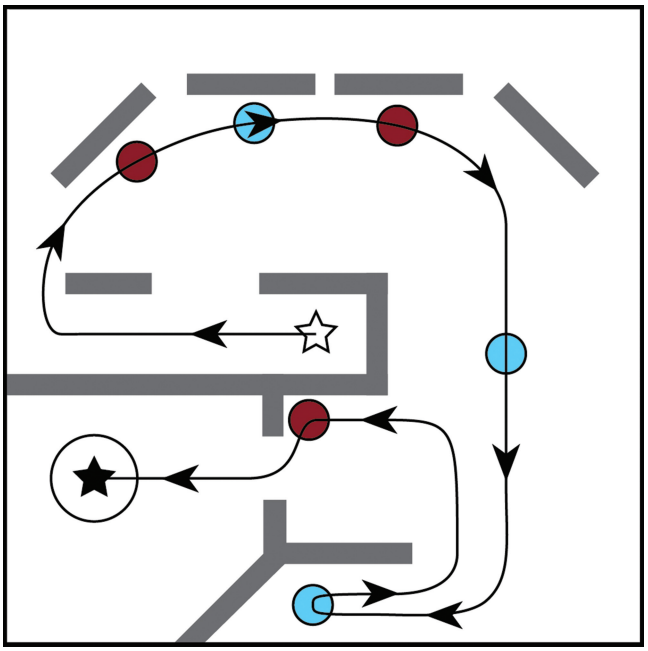

Fig. 4. Illustration of intended player path through the level. Players begin by seeking cover to gain an advantage on the first enemy before reaching all enemies and optional objectives in sequence until in sight of the final goal. Then, they deviate to reach a last optional objective before meeting a final enemy and completing the level.

analogous to techniques such as graphing intended player experience [5], providing a baseline from which deviations or anomalies can be easily identified. This may ultimately assist in the process of gleaning valuable insights from eventual results.

4.2 Defining the Player Population

Before evaluation can begin, the developer must define a population of "users" for testing, based on the game's target audience and the goals of a particular assessment. This phase is analogous to participant recruitment in traditional usertesting, whereby a representative population of players is enlisted to achieve a particular testing objective. Early stage AI-driven testing replaces this potentially time-consuming process with the ability to specify a custom user population fulfilling developer requirements for a given test. This population may be defined in a tabular or spreadsheet format specifying the number of users and the characteristics associated with each player entity (example data that may be used to define a user group is shown on the next page in Table 1). Based on this data, the framework will generate a series of player entities for testing, which the developer may then modify to redefine user groups, adjust player characteristics, or generate new users with similar characteristics to those already provided. From here, the developer proceeds to define the particulars of the testing schedule and data collection.

4.3 Orchestrating the Tests

After the definition of the testing environment and user population, the developer must configure parameters defining the operation of the testing procedure. This may include, for example, batching users into groups to be tested sequentially, configuring maximum processing time, and

Table 1. Sample Population of AI Players Defined for Testing

ID	Group	Gender	Age	Experience Level	Aggression Tendency	Exploration Tendency	Achievement Tendency
1	Novice	Male	17	Very Low	Very Low	High	Moderate
2	Novice	Female	18	Low	Moderate	Moderate	High
3	Veteran	Male	24	Very High	Moderate	Low	Very High
4	Veteran	Male	22	Very High	High	Low	Moderate
...							

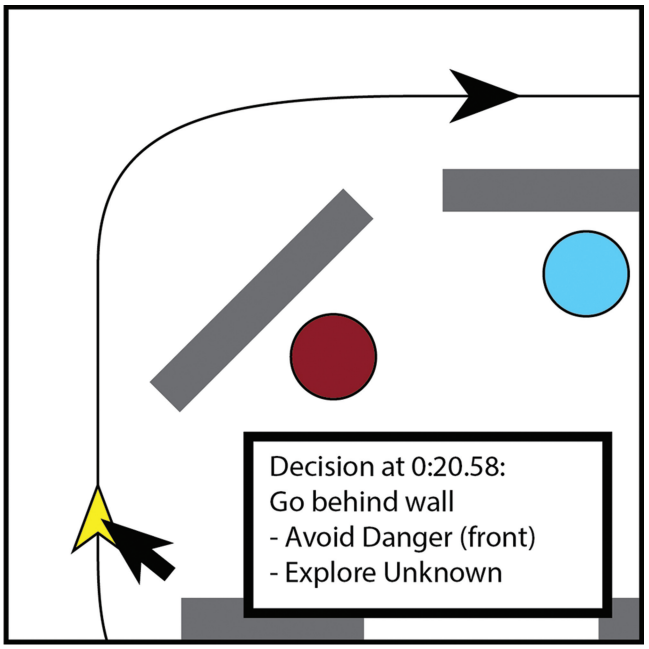

Fig. 5. Mock-up of contextual overlay showing decision logic underlying key movements in a user's path.

specifying a storage scheme for output data files. They may also configure settings for the generation of various result data logs:

Player Paths. For the logging of individual player paths, a sampling rate may be specified to increase or decrease the resolution of saved path data. Each path "sample" specifies a position, orientation, and simulated timestamp. This file may be read back into the framework to reconstruct a particular user's path, visualizing the trajectory in-scene or playing the scenario back in real-time for developer interpretation.

Decision Logs. To understand the logic underlying particular player actions, a verbose log of agent decisions may be saved detailing the reasoning behind each significant change in course. These decisions may be labelled with positional data and timestamps for cross-referencing with path data. Timestamp data may be used to automatically display relevant data from the decision log for any given point on a path visualized in-scene (see Figure 5 below).

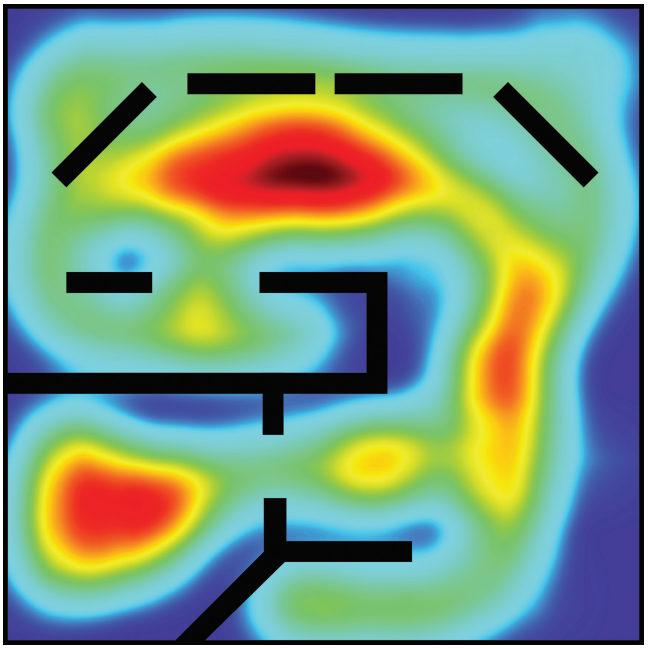

Fig. 6. Mock-up visualization of heatmap displaying player position across multiple trials for an early FPS level.

Heatmaps. For longer testing sessions or large player populations, various heatmaps may be generated from path records, displaying aggregate positioning data for individual players or groups of users. This may take the form of a static image or in-scene visualization displaying the relative amount of time spent in each unit of space for configurable levels of granularity (see Figure 6 below).

Once the procedure for preparing results has been established, testing may commence. The time taken to process any given group will depend largely on factors including level complexity and the number of users or testing rounds specified. After a testing session is complete, the developer is free to interpret the data, following a process aligned with data analysis in traditional usertesting.

4.4 Interpreting and Applying Results

For early-stage level evaluation, the developer is primarily concerned with glaring design issues such as players getting lost, missing key objectives, or finding parts of the level to be completely skippable or unnecessary. By analyzing the visualization of agent navigation and decision data, the developer may more easily identify these areas of concern, allowing for more iteration on level design before commencing later-stage testing focused on finer elements of the player experience. It is important to note here that the data obtained from AI agents—paths taken in-game, objective areas reached, and timestamps adjusted to "real-time" scale—is functionally identical to that gathered using standard game metrics. Therefore, the developer is free to analyze data-sets in the same way they might treat data gathered from actual players. Aggregate or individual paths viewed in-scene, combined with measurements of simulated time taken for level completion, can be used to estimate overall traversal difficulty and identify regions where players are prone to wandering or becoming lost. An illustrative example of potential pathways taken by two distinct agents is demonstrated in Figure 7. By contrasting actual agent trajectories with the designer's intended

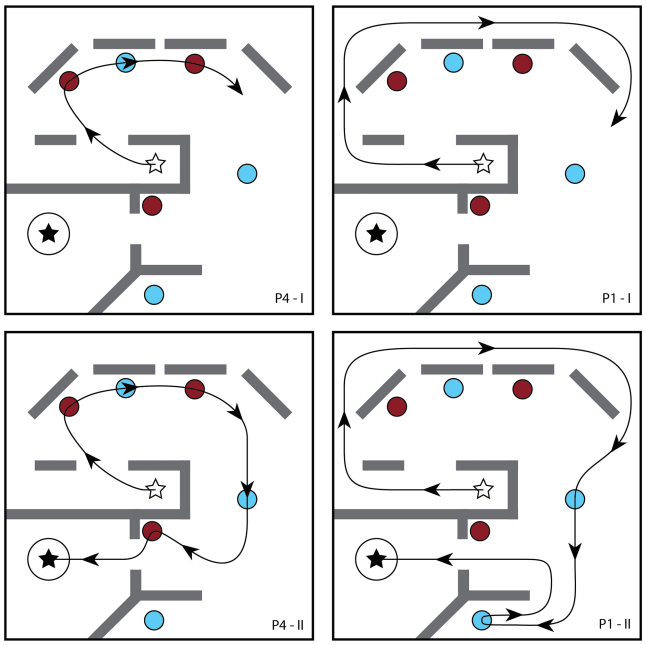

Fig. 7. Hypothetical trajectories for two different AI user agents (refer to Figure 4 for the intended player path). Left: Player 4 from Table 1, an expert with aggressive tendencies. At top, P4 skips cover, heading directly for the first two enemies. At bottom, they meet another optional goal before gaining sight of the final enemy, which they immediately engage. Finally, they head directly for the goal, neglecting to explore the alcove containing the final optional objective. Right: Player 1 from Table 1, a novice with low aggression and strong exploratory tendencies. At top, P1 keeps cover and travels behind walls to avoid enemies, missing the first optional objective. At bottom, they meet the second optional objective before exploring unknown territory to find the last optional goal, avoiding the last enemy before proceeding to level completion.

player path, discrepancies between designed and actualized player navigation can be identified. Cross-referencing this data with decision log information may provide insights as to the origin of such deviations, providing designers with a starting point for potential modifications. For example, if novice players avoid a particular area in an introductory level due to a relatively high concentration of possible hazards, then these dangers may be redistributed to encourage further safe exploration.

Heatmap data may also serve to flag areas of interest for further investigation. For instance, the developer may find that an area seemingly off the main path sees an unexpectedly high volume of traffic as players gain their bearings in the level. As a result, the area may serve to benefit from placing game resources (e.g., ammunition) or minor set-pieces, providing a small reward or guide for players along an otherwise barren path. By analyzing the timestamps associated with AI player paths, designers can also gain a better understanding of level pacing, informing future changes. These changes may be validated through further rounds of AI tests before proceeding to in-person tests to understand finer points of players' gameplay experience. Since subsequent rounds of AI testing may be easily conducted with an identical or modified user population, the developer may test several iterations on individual level designs based on this information, selecting the variations that promote ease of navigation and yield an average trajectory closer to the designer's intended experience.

5 LIMITATIONS AND FUTURE WORK

Since its inception, the field of AI in general has progressed significantly in simulating aspects of human cognition such as memory and action planning [7, 36]. While AI-driven testing is becoming more common in simple web and mobile applications, where tasks such as validation of interface elements can be automated using machine learning techniques [25], the majority of game-playing AIs are quite limited in their universality. As of this writing, technological and design limitations have largely prevented the establishment of generalizable AI-based GUR techniques, as a result of the complex nature of video games. Individual game-playing agents are most often proficient in specific game scenarios, rather than a spectrum of vastly different situations, though so-called "hyper-agents" have been used to improve the performance of game-playing AIs [16]. However, a central issue in the design of AI for GUR applications is that the desired result should imitate both skillful and faulty behaviour, accounting for vagaries such as perception, misinterpretation, and past user experience. This may prove to be a challenging factor in the development of more robust AI-based GUR tools.

5.1 Extending the AI Player Model

While our analysis so far has focused primarily on geometry-based navigation, it is important to consider ways in which this design may be extended to improve its utility as a comprehensive usertesting tool. This may encompass, for instance, the integration of gameplay mechanics (e.g., combat, collection, interacting with non-player characters (NPCs)) and improved models of human information processing. Such extensions could contribute to a more complete framework that is more broadly applicable to a wide range of games and GUR applications.

Vision Modelling. Sensory perception, particularly visual information, is of key importance in human navigational ability [34]. Furthermore, vision-based techniques have been used in robotics as a component of pathfinding and navigation systems [9]. Thus, an accurate player model may be further enhanced by attempting to model human visual perception using image recognition techniques. This system could potentially model factors driving player attention and focus, such as recognizing positive and negative space, the use of colour, in-game symbols, and shape-based cues. Such an extension may conceivably augment the realism of an agent's navigational behaviour and present opportunities to assess the visibility of game objectives and user interface elements.

Action Beyond Navigation. To make AI user agents more relevant throughout the testing process, consideration should be taken for actions apart from navigation in the game world. This may include, for example, combat, item collection, and NPC interaction. Combining these data with existing positional information will produce much richer datasets for designers to work with as part of the evaluation process. Since many of these actions may be highly specialized on a game-specific basis, attempting a generalized implementation of multiple mechanics for AI-driven testing will prove to be a uniquely challenging design opportunity. While the creation of an AI that is capable of universally interpreting a wide array of actions in the same way that a human player might is far beyond the scope of our current research, the integration of such logic beyond navigation will be a critical step in the eventual normalization of AI-based GUR.

5.2 Technical Implementation & Future Work

Our next step will be the development and evaluation of an initial prototype of an AI-driven usertesting framework based on existing research regarding player behaviour and human spatial awareness. For our initial prototype, we plan to develop the framework as a standalone plugin for an existing game engine. In the interest of this endeavour, we have chosen to work with Unity,[12]

[12] https://unity3d.com/.

a popular and freely available commercial game engine. We will be creating the AI testing framework, tentatively named *PathOS*, as a managed Unity plugin and asset package, designed to be integrated into any existing 3D game prototype with minimal overhead. Our intent is to create an open, customizable framework for level evaluation applicable to many different game genres, including first-person shooters, stealth games, platformers, and role-playing games. Ultimately, we hope to provide a lightweight early-stage testing solution to developers that demands minimal investment of development resources.

We plan to evaluate the effectiveness of framework prototypes through two main testing methods: comparison with traditional usertest results, and expert analysis. To determine whether the system is capable of imitating real players, we will test several level designs with human users and simulated users configured to mimic aspects of the chosen human participants, such as experience level and player type. The paths taken by humans and simulated users will be contrasted to investigate whether AI "users" can make decisions similar to those of their human counterparts. We will also gather expert designer and developer opinions on the proposed approach to examine the value of its findings in comparison with current testing approaches. During the later design phases and technical implementation of the framework, we intend to conduct semi-structured interviews with independent game developers and user research experts to assess their needs with respect to a semi-automated usertesting tool. We may face obstacles in our ability to test the effectiveness of the framework early-on as a result of its initial scope, which will be limited to 3D navigation, as well as potential considerations to be made for implementation-dependent or platform-specific features. To combat this, we intend to maximize the modularity and extendability of our final prototype implementation.

6 CONCLUSION

Current usertesting methods can present immense logistical and economic challenges, especially for independent developers. To help mitigate these challenges, AI-driven testing frameworks may provide a data-rich and resource-conscious alternative for game usertesting. AI techniques can thus improve the value, feasibility, and depth of the usertesting process by augmenting traditional approaches with computerized player models.

ACKNOWLEDGMENTS

The authors thank UOIT for their support in conducting this research. Samantha Stahlke would like to thank Svetlana Stahlke for her mentorship and advice in editing this manuscript.

REFERENCES

[1] E. Adar, D. S. Weld, B. N. Bershad, and S. S. Gribble. 2007. Why we search: Visualizing and predicting user behavior. In *Proceedings of the 16th International Conference on World Wide Web Pages (WWW'07)*, 161–167. DOI: 10.1145/1242572.1242595

[2] Z. A. Algfoor, M. S. Sunar, and H. Kolivand. 2015. A comprehensive study on pathfinding techniques for robotics and video games. *Int. J. Comput. Games Technol.* Article 736138 (2015). DOI: 10.1155/2015/736138

[3] R. Bartle. 1996. Hearts, clubs, diamonds, spades: Players who suit MUDs. Retrieved from http://mud.co.uk/richard/hcds.htm.

[4] M. J. Chadwick, R. S. Anjum, D. Kumaran, D. L. Schacter, H. J. Spiers, and D. Hassabis. 2016. Semantic representations in the temporal pole predict false memories. *Proc. Natl. Acad. Sci. U.S.A.* 113(36): 10180–10185. DOI: 10.1073/pnas.1610686113

[5] B. Drenikow and P. Mirza-Babaei. 2017. Vixen: Interactive visualization of gameplay experiences. In *Proceedings of the 12th International Conference on the Foundations of Digital Games (FDG'17)*. Article 3. DOI: 10.1145/3102071.3102089

[6] I. Driscoll, D. A. Hamilton, R. A. Yeo, W. M. Brooks, and R. J. Sutherland. 2004. Virtual navigation in humans: The impact of age, sex, and hormones on place learning. *Hormones Behav.* 47, 3, 326–335. DOI: 10.1016/j.yhbeh.2004.11.013

[7] C. Eliasmith, T. C. Stewart, X. Choo, T. Bekolay, T. DeWolf, Y. Tang, and D. Rasmussen. 2012. A large-scale model of the functioning brain. *Science* 338(6111): 1202–1205. DOI: 10.1126/science.1225266
[8] IGDA Games User Research SIG (n.d.) What is GUR? Retrieved from http://gamesuserresearchsig.org/what-is-gur/.
[9] C. Giovannangeli and P. Gaussier. 2008. Autonomous vision-based navigation: Goal-oriented action planning by transient states prediction, cognitive map building, and sensory-motor learning. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'08)*. DOI: 10.1109/IROS.2008.4650872
[10] D. L. Hintzman. 1984. MINERVA 2: A simulation model of human memory. *Behav. Res. Methods Instrum. Comput.* 16, 2, 96–101.
[11] B. A. Kuhl, N. M. Dudukovic, I. Kahn, and A. D. Wagner. 2007. Decreased demands on cognitive control reveal the neural processing benefits of forgetting. *Nature Neurosci.* 10, 908-914. DOI: 10.1038/nn1918
[12] T. Machado, A. Nealen, and J. Togelius. 2017. SeekWhence: A retrospective analysis tool for general game design. In *Proceedings of the 12th International Conference on the Foundations of Digital Games (FDG'17)*: Article 4. DOI: 10.1145/3102071.3102090
[13] T. Mahlmann, A. Drachen, J. Togelius, A. Canossa, and G. N. Yannakakis. 2010. Predicting player behavior in tomb raider: Underworld. In *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*. 178–185. DOI: 10.1109/ITW.2010.5593355
[14] V. Mnih, K. Kavukcuoglu, D. Silver et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 529–533H. DOI: 10.1038/nature14236
[15] J. McCarthy and P. Wright. 2004. Technology as experience. *Interactions* 11, 5, 42–43. DOI: 10.1145/1015530.1015549
[16] A. Mendes, J. Togelius, and A. Nealen. 2016. Hyper-heuristic general video game playing. 2016 In *IEEE Conference on Computational Intelligence and Games (CIG'16)*. DOI: 10.1109/CIG.2016.7860398
[17] G. McAllister, P. Mirza-Babaei, and J. Avent. 2013. Improving gameplay with game metrics and player metrics. In *Game Analytics*, M. Seif El-Nasr, A. Drachen, and A. Canossa (Eds.). Springer-Verlag London, London, 621–638
[18] P. Mirza-Babaei. 2013. Biometric storyboards: A games user research approach for improving qualitative evaluations of player experience. Ph.D. Dissertation. University of Sussex, United Kingdom.
[19] P. Mirza-Babaei, L. E. Nacke, J. Gregory, N. Collins, and G. Fitzpatrick. 2013. How does it play better? Exploring user testing and biometric storyboards in games user research. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1499–1508. DOI: 10.1145/2470654.2466200
[20] P. Mirza-Babaei, V. Zammitto, J. Niesenhaus, M. Sangin, and L. E. Nacke. 2013. Games user research: practice, methods, and applications. In *Proceedings of the 2013 CHI Conference on Human Factors in Computing Systems: Extended Abstracts (CHI'13)*. 3219–3222. DOI: 10.1145/2468356.2479651
[21] N. Moosajee and P. Mirza-Babaei. 2016. Games user research (GUR) for indie studios. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems: Extended Abstracts (CHI'16)*. 3159–3165. DOI: 10.1145/2851581.2892408
[22] L. E. Nacke, S. Engels, and P. Mirza-Babaei. 2015. Actionable inexpensive games user research. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems: Extended Abstracts*. 2461–246. DOI: 10.1145/2702613.2706681
[23] J. Ortega, N. Shaker, J. Togelius, and G. N. Yannakakis. 2013. Imitating human playing styles in Super Mario Bros. *Entertain. Comput.* 4, 2, 93–104. DOI: 10.1016/j.entcom.2012.10.001
[24] R. J. Pagulayan, K. Keeker, D. Wixon, R. L. Romero, and T. Fuller. 2003. User-centered design in games. In *The Human-computer Interaction Handbook*, J. A. Jacko and A. Sears (eds.). L. Erlbaum Associates Inc., Hillsdale, NJ, 883–906
[25] J. Renaudin. 2016. The role of artificial intelligence in testing: An interview with Jason Arbon. Retrieved from https://www.stickyminds.com/interview/role-artificial-intelligence-testing-interview-jason-arbon.
[26] M. Seif El-Nasr, A. Drachen, and A. Canossa. (eds.) 2013. *Game Analytics*. Springer-Verlag, London.
[27] C. T. Tan, T. W. Leong, and S. Shen. 2014. Combining think-aloud and physiological data to understand video game experiences. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'14)*. 381–390. DOI: 10.1145/2556288.2557326
[28] C. T. Tan, P. Mirza-Babaei, V. Zammitto, A. Canossa, G. Conley, and G. Wallner. 2015. Tool design jam: Designing tools for games user research. In *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play (CHIPLAY'15)*. 827–831. DOI: 10.1145/2793107.2810263
[29] D. Tisserand. (In Press) It's All About the Process. In *Games User Research*, A. Drachen, P. Mirza-Babaei, and L. E. Nacke (eds.). Oxford University Press, UK, 31–44.
[30] G. F. Tondello, R. R. Wehbe, L. Diamond, M. Busch, A. Marczewski, and L. E. Nacke. 2016. The gamification user types hexad scale. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play (CHIPLAY'16)*. 229–243. DOI: 10.1145/2967934.2968082
[31] I. J. M. van der Ham, A. M. E. Faber, M. Venselaar, M. J. van Kreveld, M. Loffler. 2015. Ecological validity of virtual environments to assess human navigation ability. *Front. Psychol. 2015*, 6. DOI: 10.3389/fpsyg.2015.00637

[32] A. Vieira. 2015. Predicting online user behaviour using deep learning algorithms. arXiv:1511.06247. Retrieved from https://arxiv.org/abs/1511.06247.

[33] M. Vijesh, S. Iyengar, S. M. Vijay Mahantesh, A. Ramesh, C. Pandurangan, and V. Madhavan. 2012. A navigation algorithm inspired by human navigation. In *Proceedings of the 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. 1309–1314.

[34] T. Wolbers and M. Hegarty. 2010. What determines our navigational abilities? *Trends Cogn. Sci.* 14, 3, 138–146. DOI: 10.1016/j.tics.2010.01.001

[35] G. N. Yannakakis, P. Spronck, D. Loiacono, and E. Andre. 2013. Player modeling. *Dagstuhl Follow-Ups* 6: 45–59.

[36] S. Yildrim and S. B. Stene 2008. A survey on the need and use of AI in game agents. *Proceedings of the 2008 Spring Simulation Multiconference (SpringSim'08)*. 124–131.

[37] C. Yun, P. Trevino, W. Holtkamp, and Z. Deng. 2010. PADS: Enhancing gaming experience using profile-based adaptive difficulty system. In *Proceedings of the 5th ACM SIGGRAPH Symposium on Video Games*. 31–36. DOI: 10.1145/1836135.1836140

[38] V. Zammitto, P. Mirza-Babaei, I. Livingston, M. Kobayashi, and L. E. Nacke. 2014. Player experience: mixed methods and reporting results. In *Proceedings of the 2014 CHI Conference on Human Factors in Computing Systems: Extended Abstracts (CHI'14)*. 147–150. DOI: 10.1145/2559206.2559239

Received January 2018; accepted January 2018