RESEARCH-ARTICLE

Charting the Uncharted with GUR: How AI Playtesting Can Supplement Expert Evaluation

ATIYA N NOVA, Ontario Tech University, Oshawa, ON, Canada

STEVIE CHERYL SANSALONE, Ontario Tech University, Oshawa, ON, Canada

RAQUEL ROBINSON, Ontario Tech University, Oshawa, ON, Canada

PEJMAN MIRZA-BABAEI, Ontario Tech University, Oshawa, ON, Canada

Published: 05 September 2022

FDG22: 17th International Conference on the Foundations of Digital Games
September 5 - 8, 2022
Athens, Greece

FDG '22: Proceedings of the 17th International Conference on the Foundations of Digital Games (September 2022)
https://doi.org/10.1145/3555858.3555880
ISBN: 9781450397957

Charting the Uncharted with GUR: How AI Playtesting Can Supplement Expert Evaluation

Atiya Nova
Ontario Tech University
Oshawa, Canada
atiya.nova@ontariotechu.net

Stevie C. F. Sansalone
Ontario Tech University
Oshawa, Canada
stevie.sansalone@ontariotechu.net

Raquel Robinson
Ontario Tech University
Oshawa, Canada
raquel.robinson@ontariotechu.net

Pejman Mirza-Babaei
Ontario Tech University
Oshawa, Canada
Pejman.Mirza-Babaei@ontariotechu.ca

ABSTRACT

Despite the advantages of using expert evaluation as a method within games user research (GUR) (i.e. provides stakeholders low cost, rapid feedback), it does not always accurately reflect the general player's experience. Testing the game out with real users (also called playtesting) helps bridge this gap by giving game developers an in-depth look into the player experience. However, playtesting is resource intensive and time consuming, making it difficult to implement within the tight time frames of industry game development. AI can help to mitigate some of these issues by providing an automated way to simulate player behaviour and experience. In this paper, we introduce a tool called PathOS+—a playtesting interface which uses AI playtesting data to help enhance expert evaluation. Results from a study conducted with expert participants shows how PathOS+ could contribute to game design and assist developers and researchers in conducting expert evaluations. This is an important contribution as it provides game user researchers and designers with a fast, low-cost and effective game evaluation approach which has the potential to make game evaluation more accessible to indie and smaller game studios.

CCS CONCEPTS

• **Computing methodologies** → **Intelligent agents**; • **Human-centered computing** → **User studies**.

KEYWORDS

artificial intelligence, expert evaluation, game development

ACM Reference Format:
Atiya Nova, Stevie C. F. Sansalone, Raquel Robinson, and Pejman Mirza-Babaei. 2022. Charting the Uncharted with GUR: How AI Playtesting Can Supplement Expert Evaluation. In *FDG '22: Proceedings of the 17th International Conference on the Foundations of Digital Games (FDG '22), September 5–8, 2022, Athens, Greece.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3555858.3555880

FDG '22, September 5–8, 2022, Athens, Greece

ACM ISBN 978-1-4503-9795-7/22/09…$15.00
https://doi.org/10.1145/3555858.3555880

1 INTRODUCTION

With the growing popularity and success of the digital game industry in recent years [15], there exists an ever-increasing need to ensure that games provide a fun and engaging experience for players. Game user researchers (GUR) help to fulfill this role, by better understanding methods and practices that contribute to the creation of engaging game experiences. The GUR discipline is primarily focused on better understanding player interactions through a variety of user research methods to support developers in creating fun and engaging experiences [12]. Of these methods, some are predictive, meaning there is a focus on early detection of potential design problems without the aid of player data, while others are experimental, focusing on data-driven analysis and playtesting.

One of the core predictive methods in GUR is called 'expert evaluation', in which domain experts evaluate a game in development, often with the aid of heuristic guidelines [22]. This evaluation process is easy to conduct, cost effective, and can be applied in different development phases which can help identify usability problems within the early stages of the development process. However, as there is often limited or no player data to support the evaluations, any conclusions drawn are subjective, solely based on evaluators' expertise and how they interpret design heuristics and guidelines. Alternatively, experimental methods like playtesting, which enables researchers to collect players' data as they play through a game in development [4], gives researchers an opportunity to base their evaluation on data directly captured from how players engage with a game. However, playtesting is not an accessible process as it can be resource intensive, time-consuming, and difficult to obtain user testers [20]. In our research, we are interested in augmenting the expert evaluation process by providing experts with simulated player data, generated by customizable artificial intelligence (AI) agents.

AI has been used to mitigate some of the aforementioned playtesting challenges. With AI it is possible to create intelligent agents that are able to play through games while mimicking human behaviour, hence lessening many of the time and resource costs [30]. The benefits of artificial playtesters could be extended to predictive

methods in order to overcome their limitations. For example, in expert evaluations, evaluators can use artificial playtesters to generate simulated player data to enhance their evaluation, while also retaining the low cost and ease associated with expert evaluation.

In this paper, we describe our expansion of an open-source AI Playtesting tool called PathOS [31]. PathOS is built with Unity[1] and allows developers to create and modify their own simulated player agents to playtest game levels (see Figure 1). We created a tool called PathOS+ with the goal to expand PathOS to support games user researchers performing expert evaluation using the data generated by autonomous playtesters. In the following sections, we report our implementation process and the results from evaluating PathOS+ with industry experts. Our results highlight how AI tools can assist user researchers in conducting expert evaluations, particularly in supporting both qualitative and quantitative analyses methods. This is an important contribution to the GUR field as it shows how AI-based user research tools can impact game development, particularly in providing research capabilities for small and indie developers that often have limited resources to conduct playtesting. By contributing to open-source tools and sharing our evaluation results, we help to democratize user research in the game industry and academia.

2 BACKGROUND

2.1 Expert Evaluation

Expert evaluation is the process of having a domain expert or team of experts analyze a developing work to detect potential problems with and propose possible improvements to the design [5, 17, 33]. In the context of GUR, this process refers to having domain experts analyze a developing game to identify potential usability issues to improve the quality of the product [36]. On its own, this approach is inexpensive and efficient as it wouldn't require a playtesting lab, nor involving players. The evaluation can be done by a small groups of experts, meaning the findings can often be returned quickly without the cost of time or other resources involved with a user study [23]. Expert evaluation has drawbacks however, with one of the core drawbacks being a lack of consistency in the results of the evaluations performed [36]. One method used to facilitate expert evaluations is heuristic analysis. Heuristics are sets of guidelines for helping designs conform to best practices in the industry and they are frequently used by games user researchers to provide support for their assessments [21, 23, 36]. Evaluators can then make their assessments and demonstrate how their appraisal of the design is supported by heuristic guidelines as evidence [27]. Many lists of heuristics exist, each with their own design goal [36]. For example, PLAY [9] heuristics focus on general usability guidelines for a wide variety of games, while GAP [10] heuristics focus on a game's approachability, or how easy it is for players to learn based on what the game provides through its tutorials and other teaching indicators. These standardized guidelines are an efficient and inexpensive tool for researchers, but because the application of them is dependent on the individualized experience of each evaluator, the inter-observer reliability between each assessment is generally weak [36].

[1]https://unity.com/

Research has been done to determine the optimal number of user researchers for an expert evaluation and what levels of expertise are required for most effectively evaluating a system, but while this helps for detecting more usability issues during expert evaluations, it still does not address the inconsistencies in reported problems and frequent lack of quantitative data [36]. User data is expensive to come by and many studios, particularly small or indie studios, cannot afford to run user tests to create this source of quantitative data. Additionally, the time it takes to put together a user study can make it difficult to fit into the strict schedule of a game's development cycle even when studios can afford to run user tests.

Digital tools have been investigated as a possible method for facilitating expert evaluations and past research has suggested that evaluators prefer using digital tools, but that this process can be hindered by a need to switch between the evaluation program and the game being evaluated [17]. In response to these problems, research has been done to investigate the role AI can play in helping to both facilitate expert evaluations and generate simulated user data as a cost-effective alternative source of user data.

2.2 AI for Analysis

The analytical capabilities of AI can help streamline tedious tasks. For instance, user researchers typically have to sift through large amounts of data. AI can help lessen the burden of going through this information with methods like game data mining. This was demonstrated with the work of Braun et al. [3]; they wanted to analyze the player data of *Overwatch* (Blizzard, 2016) and create visualizations and suggestions to help new players, who can feel overwhelmed by the sheer amount of knowledge required to play the game well. They used a method of clustering the players based on which character they chose to play as, and found correlations between that and high win rates. An analyst can then examine that data to make informed conclusions.

AI can also help in regards to player typology, which is a way of classifying player behaviours into distinct categories. Player typology is useful to GUR as a way to categorize and predict player behaviors. This can be done through player modeling, which uses AI or statistics to create models of player behavior. Drachen, Canossa and Yannakakis wanted to construct models of players of the game *Tomb Raider: Underworld* (also known as TRU) [11]. They gathered data from 1365 players through the game's engine with the EIDOS Metrix Suite and the XBox Live Web Service. Six features from the data were examined (which included ways of dying, total number of deaths, etc). They used an unsupervised learning approach that, once it was trained, was able to identify 4 player types with distinct behaviours (veterans, solvers, pacifists, runners). Identifying information like this can help designers ensure that elements of the game are not under-used or imbalanced.

The prediction of player behavior can also be used for dynamic difficulty adjustments or player retention. This was explored by Hawkins, Nesbitt, and Brown [16]. While noting that a player's performance ability in a game relies on their willingness to take risks, they used a particle filtering technique to try and create idealized models based off of player risk profiles. The filters are based on the Monte-Carlo Tree Search (MCTS) algorithm, and the number of particles can vary in order to create agents of different

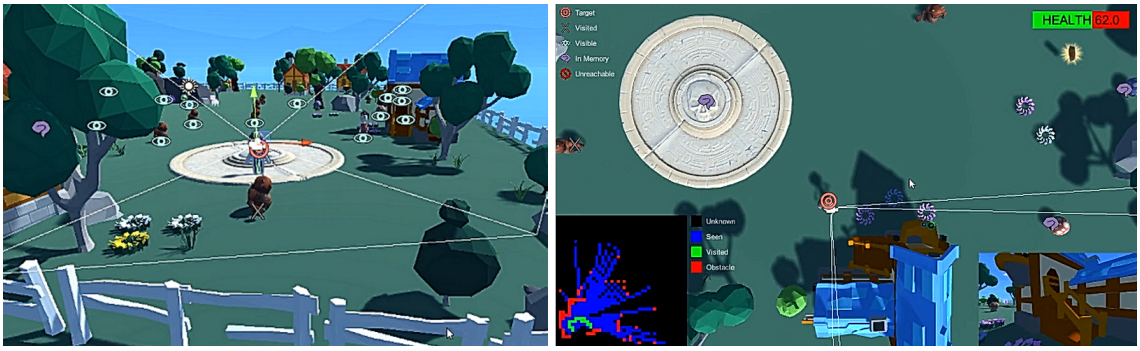

Figure 1: Overview of the tool. On the left is a scene view of the agent navigating the level, on the right is the in-game runtime interface.

performances. These models were able to dynamically adjust the difficulty of a game while a player was playing it. Similarly, Roohi et al. wanted to predict player engagement and game difficulty [28]. They used deep reinforcement learning (DRL) game-playing agents for player modeling to identify churn rates, which could then be used to infer certain things about the game (i.e. difficulty of a given level) and inform design decisions. They combined their DRL agents with MCTS and evaluated the algorithm on 168 levels of the free-to-play game *Angry Birds Dream Blast*. This method allows designers to obtain valuable information about the player without having to rely on techniques like questionnaires.

2.3 AI for Simulations

There are many ways that AI can be used to emulate players, such as player personas which simulate how players of different personalities play through games. Ariyurek, Surer, and Betin-Can wanted to expand the applicability of personas in order to make playtesting a more feasible process [1]. Original personas are rigid in what they can emulate in that they stick to specific goals or rewards. They proposed a reinforcement learning APF (alternate path finder) method in which the agent is encouraged to explore more expansively, and gets punished for visiting previously visited states. They describe this approach as "goal based personas", and these can change their personality type during a playthrough (for instance, they could start by focusing on defeating enemies, and then switch to collecting treasures). They evaluated and tested their game on GVG-AI and VisDoom environments. The results showed that the designer can use this method to see how different kinds of players can interact with their game, which can help them gain valuable insights.

Simulation testing allows developers to test the playability of their game. When it comes to physics-based puzzle games, for instance, it can be challenging to test the game's difficulty and determine whether the puzzles are possible to complete. For that reason Shaker, Shaker and Togelius created a tool called Ropossum [29] for the game *Cut the Rope*. It is a design tool that lets designers edit procedurally generated levels and use AI agents to solve them. The tool also makes suggested modifications so that the final design is playable. Deng and Fan similarly wanted to leverage AI to investigate the playability of games, but they used a different approach [8]. They used AI-based emotion sensors to detect what mood the player is in at different parts of a game, in the hopes that such information could help game developers provide a more enjoyable experience.

Aside from playability, AI can also be helpful when it comes to game balancing, a notoriously difficult task. For instance, Silva et al. were tasked with running playtests on pre-release builds of the *Sims Mobile* [7]. In order to bypass the slowness of the game–which relied on finger taps to progress–they took the base mechanics and parameters of the game and created a separate version that isolated those elements. They then used the A* algorithm for agent decision making and ran simulations. What would take human testers days of gameplay would only take their AI minutes, because they were able to run thousands of tests. This also allowed them to find significant imbalances within the gameplay that the game designers could then act on. Similarly, to improve and accelerate the playtesting process for *Hearthstone*, Garcia-Sanchez et al. looked into using an Evolutionary Algorithm (EA) to create decks that are played by AI [14]. They would then play some matches. A human would analyze the playthroughs in order to find imbalances. Their results demonstrated how the AI generated decks outperformed the human-designed ones, allowing designers to find optimal decks. They were also able to determine which cards were imbalanced or overpowered. These approaches show that not only can AI help to save time, but it can also be used to find imbalances in different types of games.

Lastly, an active area of research is how to make these agents behave more like humans. This includes making the agents imperfect. This is something that Khalifa et al. investigated for MCTS agents [18]. They made modifications to the algorithms so that the agents were more likely to pause, repeat actions, and other human-like behaviors. Their evaluations showed that they were successful in making the AI behave more like humans in certain games, but not others.

3 PATHOS TOOL

The original PathOS tool was built to aid the level design process, with some key objectives being improving accessibility and increasing generalizability by simulating the navigation of any 3D game. The tool was created for the Unity game engine due to its accessibility, as it is a popular engine for independent developers

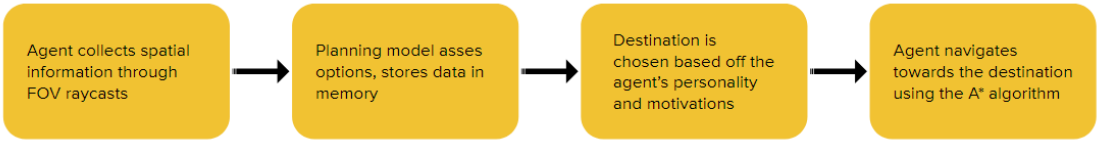

Figure 2: High level overview of how PathOS functions

to use. Written in C#, PathOS makes use of Unity's physics and NavMesh system. The NavMesh is what allows Unity's AI agents to traverse through level geometry. A high level overview of the AI functionality is provided in Figure 2.

In order to use the tool, all the user has to do is download the files and ensure that they are located within their Unity project. From there they can drag and drop the different PathOS elements—-dubbed "prefabs" within Unity—-into a game scene to begin using it. There is also a demo scene included in the project to give users an example of what a typical PathOS-supported scene is like. The entire project is free and open-source, and the original files can be found on Github. The following sections describe some of the key elements of PathOS.

3.1 Agent Behaviour

PathOS uses a cognitive architecture model [34], similar in style to the GOMS (goals, operators, methods, and selection rules) HCI model used to analyze user tasks [19]. The agents are able to do basic simulations of perception, memory, and decision-making.

3.1.1 Perception Model. Within Unity, agents have a point-of-view (POV) camera attached to them that simulates the player's in-game view. The agent is able to "see" in the game world via raycasting (line-of-sight rays are shot out from the POV camera). These rays then collide with in-game entities that are not occluded by level geometry, at which point the agent is "seeing" those entities.

3.1.2 Memory Model. The agent's memory model is split up into spatial memory and entity memory. Spatial memory is how the agent keeps track of level geometry, and is composed of the information gathered from the raycasts. This information is displayed to the user via a tile-based mini-map that shows up in the in-game interface.

Within the game level the user can mark objects as "entities" that represent different gameplay components—from enemies, to loot, to objectives. Entity memory is where records of these objects and their location are stored. How long the entity will stay in memory depends on how long they are visible for, as a way to replicate human memory. If the entity is only perceived for a short amount of time it gets stored in the agent's short term memory, in which there is limited space. If the agent perceives that entity for a longer amount of time, it gets transferred to long term memory, where it will be permanently remembered.

3.1.3 Decision Making. While the agent is traversing a level, it makes calculations to decide where it wants to go. For these calculations it determines desirable destinations to travel to, which could be entity destinations (in which the agent selects and travels to a specific in-game entity), or exploration destinations (in which the agent travels to a point in the level that is not tied to a particular entity). A scoring function is used to determine which destination the agent should go to (the destination with the highest score is what the agent uses as their target). This scoring function is run intermittently, making the agent capable of changing their mind and backtracking, in order to further resemble real players.

3.2 Interface

PathOS has its own unique interface elements that have been integrated into the Unity engine. This section goes over the interface in detail (see Figure 3).

3.2.1 Level Markup. The level markup allows users to mark objects within the level as entities. The user is shown a list of icons that represent each entity, they select the relevant icon, and then click an object within the scene for it to be registered as that entity (in this mode, the cursor is referred to as "the markup brush"). There is also a list within the interface that shows all the current entities within that level, that the user can then reference to modify the properties of a specific entity.

3.2.2 Runtime Interface. While an AI simulation is running several UI elements are shown within Unity's game window. If the agent is selected within Unity's hierarchy, it will show their first-person camera view, mental map, and which entities it has stored in its memory, among other things.

3.2.3 Behavior Customization. The interface supports the customization of agent personalities. Within the Unity inspector a slider is shown for each personality attribute (curiosity, aggression, etc.) for the selected agent, which the user can modify at will. They can also create personality "profiles", which are preset values that they can use to generate agents that fall within a specific personality type.

3.2.4 Batch Tests. A section of the interface is dedicated to running batch tests (in which multiple agents can be simulated at the same time, or one after the other). The user is given the ability to modify properties such as how many agents they want to run, what personality type they should have, and so on.

3.2.5 Visualization. Agent simulations can be recorded as logs, which can then be loaded back in as visualization data. This data can take the form of heatmaps, playtraces, and aggregates of agent interactions (which show up as circles in areas where the agent interacted with an entity). The user is given options to customize this data by doing things like selecting a specific time range, changing the colors, loading in multiple sets of data, etc.

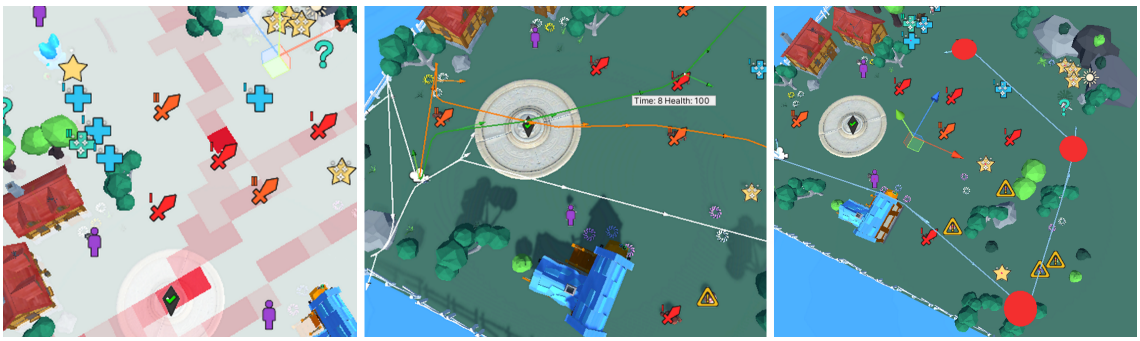

Figure 3: From left to right: heatmaps, play traces, entity interactions in PathOS

3.3 Additions for PathOS+

We built PathOS+ based off the prior open source PathOS tool developed by Stahlke et al. We prioritised our feature development based on the data collected in their study [31]. We revisited the transcripts of their interviews—through the lens of expert evaluation—in order to figure out what changes would have to be made to create PathOS+. In our previous paper, we provided in-depth details on how we decided on PathOS+ features [reference anonymized for review] by examining interview transcripts and coding them to determine which requested features would be the most applicable for expert evaluation, and improving the general analytical capabilities of the tool. Below we provide a brief summary of the features we developed for PathOS+

3.3.1 Usability features. Our first step was to address and improve the general usability issues with the tool. We particularly focused on the level markup feature to add the ability to mass tag objects as a certain entity. We added a feature to mass tag objects by dragging the markup brush in the scene. Whatever the markup brush interacts with will get set as that entity type, making the process of setting up levels to evaluate far simpler.

Another issue with the markup tool was how the brush would deselect the chosen entity type if the user were to select a key—something they would be doing often if they were performing basic actions, such as rotating the scene camera. We disabled this feature in order to further improve the ease of use of setting up levels.

We also made the timescale feature more prominent–by moving it to the top of its respective section–so that users can immediately take notice of it.

3.3.2 Streamlined Visualizations. The visualization features were an important part of PathOS. For our development, we focused on streamlining the use of these features in order to encourage users to make use of them, and to expand the tool's analysis capabilities.

In the original PathOS, in order to record logs a certain checkbox has to be checked, and the correct file directory has to be selected. These are deselected by default, and caused some confusion. To fix this issue, a lot of careful consideration and trial and error were put into how the interface was designed. Basic HCI principles for creating effective interfaces were followed [2]. Namely, care was taken to ensure that the user is not overburdened with information. Rather, they are only shown information as it was relevant. The positioning of elements were modified in order to be easier to follow, and the language was kept consistent. While this may seem like a minor consideration, ensuring the ease of use of the tool makes it more viable for a wider range of researchers to make use of it.

Another issue that we fixed was the lack of time-context for these playtraces. With the above considerations in mind, not only do the playtraces in PathOS+ show arrows indicating which direction the agent was facing at any given point, but when the user hovers over it with their cursor they can see the time (in seconds) at that specific point.

3.3.3 Resource System. Another limitation of PathOS was in its simulation capabilities—since it was purely focused on navigation, there were difficulties in envisioning games of certain genres. Our solution was to expand the sophistication of the AI and introduce new features to make it more robust. Particularly, we added new entity types for PathOS+. Instead of being only one entity type for enemies and health, different tiers were created to represent different levels of difficulty and effectiveness (i.e. a low level resource item would not heal as much as a high level resource). The user had the freedom to decide the values for these different tiers. The agent was given a variable for health, that would either increase or decrease based on their interactions. Their decision-making mostly stayed the same, except their personality values would shift if their health became low. For instance, they may become a little more cautious if they were close to dying. This was so that they were more likely to avoid enemy encounters in situations that were too dangerous, similar to how a real player would behave.

3.3.4 Centralized Window. Another issue with PathOS was the confusing layout and menu organization of the features. We created a centralized window where all the different PathOS+ features were located, instead of being separated by various game objects. This allows all the unique elements of the tool to be centralized and easily accessible. One of the tabs in this centralized window is dedicated to expert evaluation. This tab lets users create comments within the tool itself and assign a severity. They can also right-click gameobjects in the scene in order to attach that gameobject (and corresponding entity type) to a comment. The user can then export their findings into a formatted .CSV file that not only displays all their comments, but also organizes them into a table based on entity type and severity.

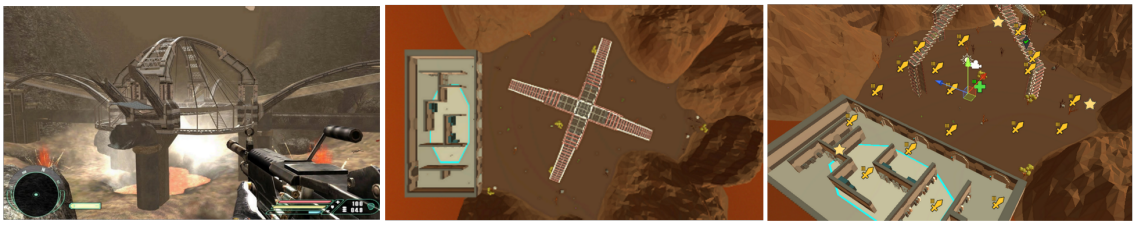

Figure 4: From left to right: screenshot of Far Cry [13], Birds-eye-view of scenario 1, the entities marked up in scenario 1

4 USER STUDY

In order to evaluate the effectiveness of PathOS+ as a tool for expert evaluation, we ran a user study with 8 expert users within the GUR field. The study was fully remote and divided into three parts: a pre-interview, a take-home expert evaluation task, and a post-interview. For the task, the participants were expected to conduct an expert evaluation on three game levels (referred to as "scenarios") that were created for this study. The key questions we wanted to assess with this study were:

- *Were participants able to identify the design problems within each level?*
- *Was PathOS+ instrumental in identifying these issues?*

4.1 Pre-Interview

The pre-interview was meant to help us learn more about the participant and their experience with game development and user research. We also used these sessions to do a demonstration of PathOS+ so as to help familiarize participants with the tool and answer any questions. The demonstration consisted of an explanation of the basic features of PathOS+ and how to run simulations, along with a tour of the PathOS+ window and all the tabs located within. The participants were explained their task and also shown how to export their evaluations into .CSV files.

4.2 Expert Evaluation Task and Level Setup

For the task we chose to base the three scenarios off of existing levels in commercially released games. We chose ones that were criticized by reviewers and players for various design flaws, in an attempt to recreate representations of those same flaws. All these scenarios were made within Unity, using various free assets in order to vaguely resemble the commercial games of which they're based.

4.2.1 Volcano, Far Cry (Ubisoft). The first entry in the *Far Cry* (Ubisoft) franchise came out in 2004, and has spawned numerous mainline entries and spinoffs. The games are first-person shooters in which the player is (typically) rampaging through tropical environments, and facing off with charismatic dictators. The first *Far Cry* was notable for multiple reasons, with one of those reasons being the infamous Volcano level.

In a GamesRadar article titled "Remember these 11 frustrating video game levels and try not to smash your controller" [32] Ryan Taljonick and Connor Sheridan say in regards to this level: *"Armed to the teeth, you step into the rim of an active volcano to fight an army of rocket spamming Hulks (which are stupidly hard to kill and are entirely out of place)... if you need to heal up, you can grab the level's only health pack–which, by the way, is on the opposite end of the arena."*

Based off of this article, along with numerous comments across message boards, we added the following design flaws in our recreation of this level for scenario 1 (two core flaws, and one miscellaneous flaw, in that order):

- The enemies are too strong, they do too much damage
- There is only one health pack in the level, making it easy to die
- The end goal is partially obscured, making it difficult for the player to find

Enemy entities were placed throughout the scenario and were marked as the highest tier of hazard enemies, which meant they did a lot of damage. Only one health pack was placed in the level, and the end goal was obscured behind a large structure (see Figure 4).

4.2.2 Blighttown, Dark Souls (FromSoftware). The *Dark Souls* (FromSoftware) series is known for its brutal difficulty and intriguing lore. They are third person action games where the player faces off against ghoulish monsters and overwhelming bosses alike. The first *Dark Souls* released in 2011 and one of its locations, named Blighttown, gained more notoriety than others.

In a Polygon article titled "Dark Souls Remastered guide: Blighttown Map" [24] Jeffrey Parkin says, *"Dark Souls Remastered's Blighttown is all kinds of awful. The enemies are equal parts dangerous, aggressive, and annoying. The ones that aren't breathing fire are probably trying to poison you. And it's not just the enemies — the design of Blighttown often feels like it's actively trying to kill you. It's full of narrow walkways, blind corners, and precipitous drops. Once you're past that, you'll enter a poisonous swamp."*

Based off of this article, along with numerous comments across message boards, we added the following design flaws in our recreation of this level for scenario 2 (two core flaws, and one miscellaneous flaw, in that order):

- There is too much poison, making it difficult to survive
- The enemies are difficult, making combat unmanageable
- There is a high level healing item right at the entrance, at a point where the player doesn't need it

We added weak enemies scattered throughout the level to represent the sequential damage that players would take from poison. To represent the difficult combat, enemy entities—ranging from easy enemies to bosses—were inserted into the level at numerous points. Alongside that, a high healing entity was placed right at the

Figure 5: From left to right: screenshot of Dark Souls [6], Birds-eye-view of scenario 2, the entities marked up in scenario 2

entrance, at a location where it would have no benefit to the player (see Figure 5). It is worth noting that in the *Dark Souls* games the bonfires can be used multiple times. However the healing item in Scenario 2 can only be used once, exacerbating the problem of its placement.

4.2.3 The Library, Halo (Bungie). *Halo: Combat Evolved* took the world by storm when it came out in 2001. As one of the XBox's launch titles it helped the console gain traction, and as a first-person shooter it helped inspire many of the shooters that came after it. While some laud this game as being one of the greatest of all time, one level in particular has drawn the ire of fans.

In a WhatCulture article titled "13 Terrible Levels In Otherwise Awesome Video Games" [25] Jack Pooley writes *"There's literally nothing more to [Halo's library level] than fighting the same enemies again and again as you ascend a building. It's pure lazy, copy-paste nonsense that drags on far, far too long and temporarily derails an otherwise magnificent game."* Based off of this article, along with numerous comments across message boards, we added the following design flaws in our recreation of this level for scenario 3 (two core flaws, and one miscellaneous flaw, in that order):

- The levels are really repetitive and "copy-paste", making it tedious
- The level itself is too long, which extends the tedium
- It's very hard to navigate, so players can get lost

In order to recreate this level a set of two rooms were created that were then copy-pasted multiple times. Each room had the same distribution and positioning of entities, including the exact same low-level enemies, and the exact same health. Not only was this level very long, but due to the repetitive nature and the number of open doorways in each room, it is disorienting to navigate (see Figure 6).

4.3 Post-Interview

The purpose of this interview was to go over the participant's evaluations of each scenario. They were given the opportunity to explain their findings, along with their general thought process. They were also asked questions about how PathOS+ factored into their evaluations, and their feedback on the tool itself. These interviews were recorded and then transcribed for analysis.

4.4 Participants

Table 1 outlines the participants who were recruited for the study. We initially recruited nine participants (N=9) to participate in the user study. Participants were recruited with relevant experience in game design or development (either through university or industry experience). One participant (P7) was not able to participate in the study beyond the initial interview—they had to drop out due to problems with their computer, leaving a final participant count of 8 for the user study (N=8). In addition, P8 completed the study, but their laptop struggled to run Unity. This led to issues with how well they were able to complete the study, and as such they were not able to run simulations for the second scenario. We still included their evaluations and general comments, however this might have affected the results.

Table 1: List of participants, their current occupation, previous UR experience, and previous unity experience

ID	Current Occupation	UX Experience	Unity Experience
P1	UX Coordinator	6-7 months	Experienced
P2	UX Grad Student	6 years	Minimal Experience
P3	Lead Designer, GUR instructor	3 years	Experienced
P4	Lead XR Developer (VR/AR)	1 year	Experienced
P5	User Research Moderator	1-2 years	Experienced
P6	User Research Moderator	3 years	No Experience
P7	NA (WITHDRAWN)		
P8	User Research Moderator	4.5 years	Minimal Experience
P9	UX Grad Student	2 years	Experienced

4.5 Data Analysis

We performed a thematic analysis on the expert evaluation task and post-interview transcripts. The pre-interview was not analyzed as it was meant to help us better understand the participants' prior experience for the purposes of discussing our findings.

In order to analyze the results of the study, two researchers individually went through the evaluation files and interview transcripts. The transcripts were divided by the interview questions. For each the researchers coded the comments and obtained relevant quotes. There were no predefined codes, rather this approach was exploratory in nature. Once they finished they then reconvened to discuss any discrepancies between their individual codes. Discrepancies were resolved through discussion until 100% agreement was reached. Overall approximately items were extracted from the interview data, almost all of which are seen in the next section, with only a couple omitted due to being outliers or only mentioned by 1 participant.

5 FINDINGS

For the following sections, we discuss our qualitative findings from post-interview transcripts. Particularly, if participants were able to identify the intended design issues (see Table 2 for reference),

Figure 6: From left to right: screenshot of Halo [26], Birds-eye-view of scenario 3, the entities marked up in scenario 3

their approach toward evaluation, usage of the tool, and finally any benefits they found from integrating the tool into their current evaluation practice.

5.1 Identification of Intended Issues

Table 2: Scenario, Intended Issues, and players that mentioned the intended issue

Level	Issues	Participants identified
Scenario 1	Enemies are too strong	7/8 (P1, P3, P4, P5, P6, P8, P9)
	Only one health pack in the level	4/8 (P1, P2, P4, P5)
	End goal is obscured	5/8 (P2, P3, P4, P5, P8)
Scenario 2	Too much "poison"	3/8 (P2, P3, P6)
	The combat is too tough	2/8 (P1, P5)
	Big health pack right at entrance	4/8 (P1, P2, P4, P5)
Scenario 3	The levels are repetitive	7/8 (P1, P2, P3, P4, P5, P6, P8)
	The level is too long	3/8 (P1, P3, P5)
	Hard to navigate	6/8 (P1, P3, P4, P5, P6, P8)

For the first scenario, participants noted that combat was too difficult (P1,P3,P4,P5,P6,P8,P9). Participants mentioned that it was a problem that there was only one health pack in the level (P1,P2,P4,P5), and that the end goal was too obscured (P2,P3,P4,P5,P8). Describing this concern, one participant explained that "*Having only the one health pack in this level could prove to be a problem since there are so many enemies*" (P5). There was also an unintentional problem that participants noted: in this level the agent spawns within a building, which opens up to a big external area. Participants mentioned that the agent had a hard time escaping this beginning area, frequently describing that "*[the AI] would get stuck in the hallways and not traverse outside of it*" (P4). This indoor area was also considered problematic for the placement of elite enemies with one participant noting that "*if you're starting in the indoor area, there's no way to exit it without really encountering an enemy*" (P5).

The second scenario had the fewest intended issues reported. Participants mentioned there was too much poison (P2,P3,P6), the combat was too tough (P1,P5), and it was problematic that the big health was right next to the entrance (P1,P2,P4,P5). In identifying these issues, participants described that "*there are too many environmental hazards that they the agents will just want to avoid at all costs*" (P3) and that "*There's like... the highest level three healing item at the start*" (P5). In relation to the challenge arising from combat, participants had mixed opinions with some identifying that "*the bosses in the second level [are clustered towards one area]. Perhaps it would be difficult for a player to get through*" (P1), while others commented that "*combat seems well balanced*" (P4). While participants were divided in their assessment of whether the combat difficulty was appropriate or overly challenging, participants tended to like the variety of enemies present in the level, with one participant stating that "*One thing is, that's definitely really nice in this map is that there is a lot of enemy varieties*" (P1). Interestingly, one of the participants stated, "*Yeah, I thought the second level... based on the mass amount of enemies in there... I thought everyone was just gonna die. [The agents] were okay, if anything, if I had to choose which one had the best balance for combat, it was definitely the second level...*" (P4). Notably, this contrast between what the participant expected in level balance based on their initial appraisal and what they observed from PathOS+ testing highlights one of the tool's core strengths: that PathOS+ provides a way to test assumptions and gain insights which may challenge the intention of evaluators.

For the third scenario, participants pointed out that the level was repetitive (P1, P2, P3, P4, P5, P6, P8), too long (P1, P3, P5), and hard to navigate (P1, P3, P5, P6, P8). The repetitiveness included the environment, enemy types, and enemy layouts (all of which were intended). P1 mentioned "*there's a chance that players can easily get lost within because they don't really have a sense of direction to know where they're supposed to go*" and P3 stated that "*The goal is too far away*". As one participant summarized it, "*The issues all kind of stemmed from the same problem in which there wasn't a lot of variety*" (P4). There were other unintended issues brought up as well, such as how this scenario did not provide much challenge with one participant describing that "*there didn't seem to be a lot of challenge, either to it. None of them came close to dying, ever*" (P4). This lack of challenge was attributed by participants to a number of factors including how "*The enemies that are on this map, they're all level one enemies...*" (P1) and that "*...There's a high level preservation item in each room. But I think that might make it too easy for players...*" (P1).

PathOS+ has directly factored in the identification of some core issues. This was particularly evident for Scenario 3, as participants stated that they only would have noticed the issues of this scenario by watching the agents. P5 stated "[In Scenario 3]... I probably wasn't thinking of like, this is also uniform... I'm like, there's actually just a simple direct path from like the start all the way till the end ...But once I saw the agents moving around, I saw them just going between the same rooms ... And that kind of made me think, the uniformity of these levels can actually prove to be an issue." Similarly P6 said, "The AI and his exploration of [Scenario 3] helped me to determine that, oh, there's actually far too many

doors at this level, because it's so easy to get in and out of every single door."

Table 3: High level overview of general insights from results

Insight	Participants
Started evaluations by watching specific agents	P2, P3, P4, P6, P8
Focused on qualitative data	P1, P2, P6, P9
Focused on quantitative data	P3, P5, P8
Mixed qualitative and quantitative	P4
Agents challenged perspective	P1, P2, P4, P5, P6, P8, P9
Thought it was cumbersome to switch between agents	P1, P5, P6, P8
Wanted more dynamic interactions	P2, P8
Used visualizations most often	P5, P8, P9
Used time-scaling most often	P1, P4, P6
Used agent batching most often	P2, P3, P4
Couldn't use time-scaling	P3, P5
Chose not to log agents	P1, P2, P3, P6
Did not use agent view	P4, P5
Exclusively used personality profiles	P3, P4
Exclusively used personality sliders	P8, P9
Found PathOS+ intuitive or easy to learn	P1, P4, P5, P8, P9
Had issues learning how to use PathOS+	P2, P3, P6
Thought runtime minimap was useful	P3, P4, P5, P6, P8, P9
Thought first-person agent view was useful	P4, P6, P9
Wanted clearer feedback	P1, P2, P6
Unable to see gizmos	P3, P6, P9
Liked right-clicking objects	P1, P4, P5, P6
Thought PathOS+ was time-efficient	P1, P6, P8, P9
Wanted AI improvements	P2, P4, P6, P8
Wanted usability improvements	P3, P4, P6
Wanted better onboarding	P1, P8, P9
Would use PathOS+ for personal projects	P1, P2, P4, P5, P6, P8, P9
Would use PathOS+ for professional project	P3

5.2 Usage of PathOS+

Participants used PathOS+ for gaining insights about the scenarios, based on feedback from the post-interviews.

When asked about their expert evaluation process, participants generally preferred to start their evaluations by watching one or more agents navigate the level (P2, P3, P4, P6, P8). For example, P6 stated, "*I just wanted to watch the agent do whatever he was doing first*". Participants would then guide their further evaluation of the level based on their observation of agent behaviours. While participants were split on whether they preferred observing single agents (P6), batches of agents (P3, P5, P8), or a mixture of both methods (P4), in general, participants tended to use at least some batching in their evaluations. Breaking this down further, half of the participants used PathOS+ to generate qualitative data by using tools such as the first-person agent camera to focus on player perspectives (P1, P2, P6, P9), and three of the participants focused on using PathOS+ for generating quantitative data through the batching, logging, and visualization features (P3, P5, P8). In contrast to these participants, P4 used PathOS+ to perform a comprehensive review. They started their evaluation of each scenario by generating qualitative data by focusing on player perspectives. They then moved into quantitative data analysis by using batches of agents to observe how consistently groups of agents with the same personality would act in a given scenario. "*I would start with a single agent go through, just like following in his footsteps ... And then if I had batching working, I would set out for of one type, and then I'd set up for another personality, so on and so forth, and just kind of see where they differed....*" (P4) Regardless of the chosen method, participants consistently reported that observing PathOS+ agents navigate each of the scenarios challenged their initial assumptions about them (P1, P2, P4, P5, P6, P8, P9). One way that agent behaviour frequently challenged the expectations of participants was with how agents navigated through the scenario and chose targets (P1, P5, P6, P8, P9). In evaluating the tool, participants described it as time efficient (P1, P9), credited it with helping them recognize the problem of repetitiveness in scenario three (P5, P6), and reported that using PathOS+ had provided them with insights they would not have obtained by playing the level themselves (P1, P3, P4, P8). Describing their experience of having expectations challenged by observing PathOS+ agents, one participant explained how they "*just assumed that [players just] go to the end. But now I understand that players actually want to explore each area and kind of interact with everything that's available to them*" (P1). This participant continued to explain how this differed from their personal approach in which "*I don't really deal with the optional goals or anything, I just kind of go straight to the end*" (P1).

While participants felt PathOS+ had provided them with new insights, they also had some issues with use of the tool and agent behaviour. Four of the participants mentioned issues with switching between agents and PathOS+ window tabs during simulations (P1, P5, P6, P8), and two participants also reported a desire for more dynamic interactions, such as moving enemies (P8), and interactions between agents and the environment (P2, P8). Of the existing features in PathOS+, participants reported frequent use of visualization tools (P5, P8, P9), agent batching (P2, P3, P4), and time-scaling (P1, P4, P6). All the participants who reported frequent use of visualization tools indicated that they liked using heatmaps. Additionally, both P3 and P5 indicated a desire to use time-scaling but were prevented by bugs with the tool which they were not aware of at the time. Attempting to use this feature, P5 described how they "*tried playing around with the timescale, but it didn't seem to do much*", while P3 indicated that their biggest barrier when attempting to use PathOS+ features was that they were "*running into I think too many issues with the setup*". In contrast to the preference and interest participants showed for these features, half of the participants indicated that they didn't choose to log player data for further analysis (P1, P2, P3, P6), and two participants didn't use the first-person agent camera view (P4, P5). Agent personalities, a key feature of PathOS+ for adjusting the decision-making process of agents, were used differently across participants. Some participants opted to use pre-set personality profiles exclusively (P3, P4), while others avoided profiles entirely and adjusted the individual agent traits manually (P8, P9). Improving these issues could make it easier for participants to identify design problems.

5.3 Feedback on PathOS+

Learning how to use PathOS+ effectively was generally considered straightforward with five participants describing it as intuitive or easy to learn (P1, P4, P5, P8, P9). P4 was able to learn the tool despite not accessing the manual during the study. Of the three who did not describe it this way (P2, P3, P6), one said the tool "*was a little finicky. Especially with all the setup*" (P3), while P2 and P6 said that there was a steep learning curve. However, P2 added that the tool would not be difficult to learn for users with experience using Unity. Overall, participants were able to use the interface of PathOS+ with all participants describing most of their experience favourably. UI elements which were particularly useful included the simulation mini map (P3, P4, P5, P6, P8, P9) and the first-person view agent camera (P4, P6, P9). However, three participants described that they would have liked more clearly displayed feedback for events such as characters taking damage (P1, P2, P6), and three participants were unable to see gizmos such as entity tags during simulations (P3, P6, P9).

Despite these points of friction, all participants indicated that they considered PathOS+ to be a useful tool for conducting expert evaluations and four participants particularly commended the feature of attaching comments to game objects by right-clicking (P1, P4, P5, P6). While all participants considered the tool useful, seven of the participants indicated elements of PathOS+ they would like to see improved. These elements included requests for improvements to AI (P2, P4, P6, P8) and usability (P3, P4, P6), as well as a wider range of options for the entity tagging system (P5). Additionally, three participants requested improvement to onboarding and learning experience of PathOS+ (P1, P8, P9). Specifically, P1 requested further documentation and a tutorial scenario, P8 requested tooltips for helpful reminders on the functionality of various menu elements, and P9 requested some way of better indicating how to toggle gizmos and a way to toggle them through the PathOS+ menu. Finally, seven of the participants indicated they would most likely use PathOS+ for some form of personal project (P1, P2, P4, P5, P6, P8, P9), while P3 suggested they would use it for commercially produced games in a compatible genre such as tactical strategy.

6 DISCUSSION

In this section, we discuss key benefits of PathOS+ based on our results, as well as how this feedback from participants will inform the future design and integration of the tool into real world contexts.

6.1 Identification of Intended Issues

We intentionally put design issues within each scenario in order to determine if PathOS+ could be used to identify them. The results indicated that participants were able to identify these issues and justify why they thought they were issues, all while using the data generated by PathOS+. This is important because it tackles one of the key drawbacks of expert evaluation, that it is difficult to justify findings due to a lack of player data. Using tools like PathOS+ means that evaluators can have evidence for any claims that they make, thus increasing their confidence in reporting issues.

6.2 Benefits of PathOS+

PathOS+ supports flexible styles of work. One of the core benefits of PathOS+ is that it can support either qualitative or quantitative data collection. We noticed that among the participants there was a clear distinction in how they approached the evaluation task. Some participants focused on the player perspective and primarily ran single tests to get qualitative data, which helped them ensure the game was properly tailored for their target audience.

Other participants focused on data generation and ran batch tests to obtain quantitative data. This was well suited for game balancing purposes, as it empowered participants to see the bigger picture and identify patterns in agent behaviour. These results suggest that PathOS+ is flexible enough to support both qualitative and quantitative approaches to data collection.

PathOS+ supports different levels of experience. PathOS+ was able to accommodate researchers with different levels of experience. This is evident with how P1 (who had the least experience) identified most of the core issues, while P5 (who had almost double the experience) identified the same amount of issues. This is important because it means that the tool is approachable to any user researcher, and even for the inexperienced it can still be an effective way to identify design problems.

PathOS+ saves time in the game development process. A comment that multiple participants brought up is that PathOS+ saves time. This is in accordance with a point described in the reviewed literature that saving time is one of the key benefits of AI testing. In turn, this helps solve a common issue with data analysis, that it takes too long or that the volume of player data is too large for humans to reasonably sift through [35]. The reality of game development is that playtests cannot be run all the time. It is a taxing process, and even recruiting the right participants for a playtest can be a Herculean task. PathOS+ allows playtests to be run frequently and from an early stage in the development process. This then means that any design-based problems can be noticed and rectified early, so more design iterations can occur.

Another notable observation is that many participants appreciated the expert evaluation tab and the features therein. This included right-clicking on in-game objects to add comments associated with that object, and both importing and exporting their findings as "comments". This preference for features which streamline the review process is in line with results from some of the reviewed literature that user researchers prefer digital tools because they help with efficiency [17]. Another point from this literature was that users found it annoying to swap between the game they were analyzing, and the program where they were doing their evaluation. PathOS+ allows reviews to be done in the same program as the game being developed which in turn solves this problem. The results of our study indicate that PathOS+ is a valuable tool for making expert evaluation more efficient.

6.3 Applications to Industry

PathOS+ and its use cases are especially applicable to independent game studios due to its accessibility (as it is open source), ease of setup, and intuitiveness to learn. The tool requires minimal setup in comparison to standard playtesting. For PathOS+, the files need to be downloaded and imported into a project, the assets

dragged and dropped into the scene, and then the scene marked up accordingly. This means an indie studio could use this tool at any stage of the development process in order to quickly run many tests and gain quick insights into their game. The tool also makes up for the lack of human feedback through the emulation of different personality types. Developers can quickly run tests to see how different types of players (e.g. aggressive, to cautious, to completionist player types) engage with their levels. The flexibility of PathOS+ allows the nuances of these personalities to be tweaked within the tool to reflect the particular player type the developer wants to test. Overall, these aspects can potentially help developers better and more easily make important design decisions.

Another key application of PathOS+ is large-scale testing. The number of agents that can simultaneously navigate a level during a batch test is currently capped at 8, however this amount can be increased to suit the developer's needs. This feature can help developers quickly run dozens of tests without the need to worry about recruiting a lot of human testers, which can be time consuming and costly. These batch tests can then generate lots of data in a short time-frame, saving the development team time and money. By looking at the ensuing heatmaps from these tests, for instance, developers may gain insights such as what parts of the level the agents were spending the most time on and adjust the design accordingly. This process can also help with difficult issues such as game balancing. This was witnessed during the study, for some of the intentional design issues incorporated into the scenarios. The first scenario, for instance, had an issue where the combat was too difficult. Almost all the participants were able to identify and document this issue. Within a game development team, after a user researcher identifies this problem, the game developer can adjust the combat in their game in order to balance it properly. A tool such as this could also hypothetically be used to guide more targeted human testing, in which the AI could be used to identify general problem areas, from which more focused tests could be run with human participants, thereby saving time.

The expert evaluation tab not only empowers user researchers to record their evaluations within a given game level, but it also allows them to seamlessly share their findings with other developers on the team. Evaluations can be exported and then sent to other team members, who can then import it into their own project. The benefit of this is that not only will they be able to see the general comments the user researcher would have made, but they also would see comments tied to specific game objects within the scene. This can help facilitate communication and improve the clarity of identified design issues.

6.4 Limitations and Future Work

One possible reason for why more of the issues were not detected by participants could be due to some of the caveats we identified. The AI lacking dynamism meant that it was harder for participants to visualize or fully understand certain interactions. The usability issues (and subsequent bugs) made it difficult for some participants to use the tool. These issues will be the focus of future work to improve the usability of this tool. Another potential reason that the participants were not able to find all the issues is that this was a simulated study. The participants were not working on a real game, and they did not have access to designers they could consult. In a practical setting, participants would have talked to developers and referenced design guidelines. All of these things were absent in our study, which is important to keep in consideration. Additionally, while the tool is currently open-source, modifying parts of it may seem daunting to users. We could improve the interface to allow users to add things like their own entities and resources, along with making the process of creating personality types within the tool more robust. Improving these open-source aspects of the tool could help make it more generalizable.

7 CONCLUSION

As a predictive methodology, the main benefits of expert evaluation are that it is cost-effective, easy to conduct, and can help developers gain valuable insights early on. The downside is that it may not accurately capture the issues with the player experience. Playtesting, on the other hand, is a costly and resource intensive process. But it gives designers an accurate glimpse into what the player experience is like. With AI testers, these two techniques can be merged to take full advantage of the benefits, without being weighed down by the detriments.

We developed our tool (PathOS+) based on an open-source AI playtesting tool for the Unity game engine. Our goal is for PathOS+ to be used as a tool to enhance expert evaluation. We added features to make it a more suitable tool for analytical purposes and made some usability improvements to the tool, added a simple resource and combat system, streamlined the visualization process, and created a centralized window where all the features are located, including a tab specifically to conduct and export expert evaluations. These changes were incorporated to not only make the tool easier to use, but to make it easier for users to analyze game data and share their findings.

We ran a study to determine how beneficial PathOS+ was for expert evaluation. Three unique scenarios were created in Unity based off of real commercial games, and they had intentional design flaws within them. We recruited participants with varying degrees of user research experience to conduct evaluations on all these scenarios. When they were done with their evaluations they had to export their findings via the tool, and have an interview with one of the researchers to discuss their findings and overall experience with the tool.

Overall, participants were able to identify most of the design flaws. They all stated that the tool was useful for expert evaluation (such as being able to right click entities to add comments about them, and import/export findings). Participants mentioned that the tool was intuitive to learn (despite the different levels of experience) and that it helped them save time. Some of the participants said that they were only able to identify some of the design flaws within the levels due to the AI. We also determined two core approaches that participants took when it came to conducting their evaluations–player perspective (prioritizing single AI tests) and data generation (prioritizing batch tests), with one participant going with a hybrid approach.

Overall, this paper shows how AI-based user research tools can impact game development, particularly in providing research capabilities for small and indie developers that often have limited

resources to conduct playtesting. This is an important contribution for GUR as it enables more developers to benefit from user-centred approaches in their game development process.

ACKNOWLEDGMENTS

This work has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC DiscoveryRGPIN-2021-03500 and NSERC CREATE SWaGUR CREATE 479724-2016).

REFERENCES

[1] Sinan Ariyurek, Elif Sürer, and Aysu Betin Can. 2021. Playtesting: What is Beyond Personas. *ArXiv* abs/2107.11965 (2021).

[2] Adream Blair-Early and Mike Zender. 2008. User Interface Design Principles for Interaction Design. *Design Issues* 24 (2008), 85–107.

[3] Peter Braun, Alfredo Cuzzocrea, Timothy D. Keding, Carson K. Leung, Adam G.M. Padzor, and Dell Sayson. 2017. Game Data Mining: Clustering and Visualization of Online Game Data in Cyber-Physical Worlds. *Procedia Computer Science* 112 (2017), 2259–2268. https://doi.org/10.1016/j.procs.2017.08.141 Knowledge-Based and Intelligent Information and Engineering Systems: Proceedings of the 21st International Conference, KES-20176-8 September 2017, Marseille, France.

[4] Judeth Oden Choi, Jodi Forlizzi, Michael Christel, Rachel Moeller, MacKenzie Bates, and Jessica Hammer. 2016. Playtesting with a Purpose. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play* (Austin, Texas, USA) *(CHI PLAY '16)*. Association for Computing Machinery, New York, NY, USA, 254–265. https://doi.org/10.1145/2967934.2968103

[5] Gilbert Cockton and Alan Woolrych. 2001. Understanding Inspection Methods: Lessons from an Assessment of Heuristic Evaluation. (01 2001). https://doi.org/10.1007/978-1-4471-0353-0_11

[6] Dan Curtis. 2018. Blighttown - Dark Souls Wiki Guide. https://www.ign.com/wikis/dark-souls/Blighttown

[7] Fernando de Mesentier Silva, Igor Borovikov, John F. Kolen, Navid Aghdaie, and Kazi A. Zaman. 2018. Exploring Gameplay With AI Agents. In *AIIDE*.

[8] Shuh-Yeuan Deng and Kuo-Kuang Fan. 2021. Evaluation system for game playability using emotion sensor based on ai. *Sensors and Materials* 33, 9 (2021). https://doi.org/10.18494/sam.2021.3479

[9] Heather Desurvire and Charlotte Wiberg. 2009. Game Usability Heuristics (PLAY) for Evaluating and Designing Better Games: The Next Iteration. In *Online Communities and Social Computing*, A. Ant Ozok and Panayiotis Zaphiris (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 557–566.

[10] Heather Desurvire and Charlotte Wiberg. 2010. *User Experience Design for Inexperienced Gamers: GAP – Game Approachability Principles*. 131–147. https://doi.org/10.1007/978-1-84882-963-3_8

[11] Anders Drachen, Alessandro Canossa, and Georgios N. Yannakakis. 2009. Player modeling using self-organization in Tomb Raider: Underworld. *2009 IEEE Symposium on Computational Intelligence and Games* (2009), 1–8.

[12] Anders Drachen, Pejman Mirza-Babaei, and Lennart Nacke. 2018. *Games User Research*. Oxford University Press, Inc., USA.

[13] GamingRevenant. 2018. Far cry 1: Walkthrough - volcano [level 20] (realistic mode) 4K UHD - 60fps max settings. https://www.youtube.com/watch?v=1fD7xcp7KuU&ab_channel=GamingRevenant

[14] Pablo Garcia-Sanchez, Alberto TONDA, Antonio Mora, Giovanni Squillero, and Juan Julian Merelo. 2018. Automated playtesting in collectible card games using evolutionary algorithms: A case study in hearthstone. *Knowledge-Based Systems* 153 (Aug. 2018), 133–146. https://doi.org/10.1016/j.knosys.2018.04.030

[15] Ben Gilbert. 2020. Video-game industry revenues grew so much during the pandemic that they reportedly exceeded sports and film combined. *Business Insider* (2020). https://www.businessinsider.com/video-game-industry-revenues-exceed-sports-and-film-combined-idc-2020-12

[16] Guy Hawkins, Keith Nesbitt, and Scott Brown. 2012. Dynamic Difficulty Balancing for Cautious Players and Risk Takers. *International Journal of Computer Games Technology* 2012 (06 2012). https://doi.org/10.1155/2012/625476

[17] Ebba Thora Hvannberg, Effie Lai-Chong Law, and Marta Kristín Lárusdóttir. 2007. Heuristic evaluation: Comparing ways of finding and reporting usability problems. *Interacting with Computers* 19, 2 (2007), 225–240. https://doi.org/10.1016/j.intcom.2006.10.001 HCI Issues in Computer Games.

[18] Ahmed Khalifa, Aaron Isaksen, Julian Togelius, and Andy Nealen. 2016. Modifying MCTS for Human-like General Video Game Playing. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* (New York, New York, USA) *(IJCAI'16)*. AAAI Press, 2514–2520.

[19] David E. Kieras. 2003. GOMS Models for Task Analysis.

[20] Pejman Mirza-Babaei, Naeem Moosajee, and Brandon Drenikow. 2016. Playtesting for Indie Studios. In *Proceedings of the 20th International Academic Mindtrek Conference* (Tampere, Finland) *(AcademicMindtrek '16)*. Association for Computing Machinery, New York, NY, USA, 366–374. https://doi.org/10.1145/2994310.2994364

[21] Jakob Nielsen and Robert L. Mack. 1994. *Usability Inspection Methods*. Wiley, New York.

[22] Jakob Nielsen and Rolf Molich. 1990. Heuristic Evaluation of User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Seattle, Washington, USA) *(CHI '90)*. Association for Computing Machinery, New York, NY, USA, 249–256. https://doi.org/10.1145/97243.97281

[23] Atiya N Nova, Stevie Cheryl Francesca Sansalone, and Pejman Mirza-Babaei. 2021. PathOS+: A New Realm in Expert Evaluation. In *Extended Abstracts of the 2021 Annual Symposium on Computer-Human Interaction in Play* (Virtual Event, Austria) *(CHI PLAY '21)*. Association for Computing Machinery, New York, NY, USA, 122–127. https://doi.org/10.1145/3450337.3483495

[24] Jeffrey Parkin. 2018. Dark souls remastered guide: Blighttown Map. https://www.polygon.com/dark-souls-remastered-guide/2018/7/2/17478910/blighttown-map-items-npc

[25] Jack Pooley. 2017. 13 terrible levels in otherwise awesome video games. https://whatculture.com/gaming/13-terrible-levels-in-otherwise-awesome-video-games

[26] Jack Pooley. 2021. 10 convoluted video game levels everyone got lost in. https://whatculture.com/gaming/10-convoluted-video-game-levels-everyone-got-lost-in?page=9

[27] Mikko Rajanen and Dorina Rajanen. 2018. Heuristic evaluation in game and gamification development. In *GamiFIN*.

[28] Shaghayegh Roohi, Christian Guckelsberger, Asko Relas, Henri Heiskanen, Jari Takatalo, and Perttu Hämäläinen. 2021. Predicting Game Difficulty and Engagement Using AI Players. *Proc. ACM Hum.-Comput. Interact.* 5, CHI PLAY, Article 231 (oct 2021), 17 pages. https://doi.org/10.1145/3474658

[29] Noor Shaker, Mohammad Hossein Shaker, and Julian Togelius. 2013. Ropossum: An Authoring Tool for Designing, Optimizing and Solving Cut the Rope Levels. In *AIIDE*.

[30] Samantha . Stahlke, Atiya Nova, and Pejman Mirza-Babaei. 2019. Artificial Playfulness: A Tool for Automated Agent-Based Playtesting. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI EA '19)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3290607.3313039

[31] Samantha Stahlke, Atiya Nova, and Pejman Mirza-Babaei. 2020. *Artificial Players in the Design Process: Developing an Automated Testing Tool for Game Level and World Design*. Association for Computing Machinery, New York, NY, USA, 267–280. https://doi.org/10.1145/3410404.3414249

[32] Ryan Taljonick and Connor Sheridan. 2017. Remember these 11 frustrating video game levels and try not to smash your controller. https://www.gamesradar.com/frustrating-levels-nearly-made-us-break-our-controllers/

[33] Wei-Siong Tan, Dahai Liu, and R. Bishu. 2009. Web evaluation: heuristic evaluation vs. user testing. *International Journal of Industrial Ergonomics* 39 (2009), 621–627.

[34] Kristinn Thórisson and Helgi Helgason. 2012. Cognitive Architectures and Autonomy: A Comparative Review. *Journal of Artificial General Intelligence* 3 (01 2012), 1–30. https://doi.org/10.2478/v10229-011-0015-3

[35] Günter Wallner and Simone Kriglstein. 2013. Visualization-based analysis of gameplay data - A review of literature. *Entertain. Comput.* 4 (2013), 143–155.

[36] Gareth R. White, Pejman Mirza-babaei, Graham McAllister, and Judith Good. 2011. Weak Inter-Rater Reliability in Heuristic Evaluation of Video Games. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI EA '11)*. Association for Computing Machinery, New York, NY, USA, 1441–1446. https://doi.org/10.1145/1979742.1979788