Evaluation of a Multi-Agent 'Human-in-the-Loop' Game Design System

JAN KRUSE

Mediadesign University of Applied Sciences, j.kruse@mediadesign.de

ANDY M. CONNOR

Auckland University of Technology, andrew.connor@aut.ac.nz

STEFAN MARKS

Auckland University of Technology, stefan.marks.ac@gmail.com

Designing games is a complicated and time-consuming process, where developing new levels for existing games can take weeks. Procedural content generation offers the potential to shorten this timeframe, however automated design tools are not adopted widely in the game industry. This paper presents an expert evaluation of a human-in-the-loop generative design approach for commercial game maps that incorporates multiple computational agents. The evaluation aims to gauge the extent to which such an approach could support and be accepted by human game designers, and to determine whether the computational agents improve the overall design. To evaluate the approach, eleven game designers utilized the approach to design game levels with the computational agents both active and inactive. Eye tracking, observational and think aloud data was collected to determine whether designers favored levels suggested by the computational agents. This data was triangulated with qualitative data from semi-structured interviews that were used to gather overall opinions of the approach. The eye tracking data indicates that the participating game level designers showed a clear preference for levels suggested by the computational agents, however expert designers in particular appeared to reject the idea that the computational agents are helpful. The perception of computational tools not being useful needs to be addressed if procedural content generation approaches are to fulfill their potential for the game industry.

CCS CONCEPTS • Computing methodologies~Artificial intelligence~Distributed artificial intelligence~Multi-agent systems • Theory of computation~Design and analysis of algorithms~Mathematical optimization~Discrete optimization~Optimization with randomized search heuristics~Evolutionary algorithms • Applied computing~Computers in other domains~Personal computers and PC applications~Computer games

Additional Keywords and Phrases: Human-based computation, autonomous agents, evolutionary computation, multi-agent systems, procedural content generation, user evaluation.

1 INTRODUCTION

Designing game content is a complicated and time consuming process [28] with one game designer suggesting that a complete game map can take around five weeks to design [59]. Poorly designed game maps can quickly result in players losing interest [46], therefore designing quality game maps is critical in order to develop a successful game. Whilst automatic generation of game maps is a topic of interest to the research community [8,38,40], the design of commercial game maps continues to generally be a manual process that does not fully exploit the potential for automated approaches. This paper presents an expert evaluation of a human-in-the-loop generative design approach that is intended to procedurally generate game maps. The purpose of the system is to supplement a human designer rather than replace them, and as such it is implemented around a cognitive model of game designers that allows autonomous agents to evaluate and present game maps to the human designer.

2160-6455/2022/1-ART1 $15.00
https://doi.org/10.1145/3531009

Human-in-the-loop systems allow the designer to make decisions related to the design process whilst still offering the potential for automation to speed up the process. The design approach under consideration in this paper utilizes a multi-agent approach that extends an underlying interactive genetic algorithm to better support the designer by the addition of computational agents that make recommendations to the designer. The inclusion of the additional agents has been shown to be effective in improving the quality of game maps [37], however the adoption of such systems in practice depends very much on whether game designers perceive value in the tools. The overall goal of this research is therefore to evaluate how the multi-agent system is utilized and received by game designers.

Implicit within this goal are three intermediate goals: 1) do the computational agents within the system propose potential solutions that are of interest to the designers? 2) does the system allow the designers to create playable levels? 3) How do the designers react to the whole system? This research utilizes a combination of quantitative data from both software telemetry and eye tracking that can be triangulated with qualitative data collected using a think-aloud protocol and semi-structured interviews. This triangulation allows both behavior and perceptions to be analyzed and compared.

This paper is structured as follows. Section 2 outlines the background context and existing work in this area. An overview of the human-in-the-loop design approach is given in Section 3 along with details related to the game concept used in this study. Section 4 describes the method used to design and conduct the evaluation and Section 5 presents the results from both the eye tracking and the interviews. These results are discussed in Section 6 and the conclusion is given in Section 7.

2 BACKGROUND AND RELATED WORK

The research outlined in this paper is focused on the evaluation of a multi-agent system for generative game map design and as such sits in the context of procedural content generation.

2.1 Procedural Content Generation

Procedural content generation (PCG) in computer games can be defined as the automatic creation of game content using algorithms [64]. Other definitions are more liberal and define procedural content generation as game content creation with limited or no user input at all [58]. This research uses the latter definition and understands PCG as game content creation with user interaction, where the PCG system is intended to reduce the cognitive load on the game designer and assist the design process by presenting novel solutions that align with the designers' intent.

A complete review of PCG as a field of study is beyond the scope of this paper. However, it is worth noting that PCG has been used to create an array of game assets and content, such as maps, adventures, characters, weapons, planets, plants and histories [16]. A wide variety of techniques have been utilized to create such content, including methods from artificial intelligence (AI) or computational intelligence (CI), such as evolutionary computation and constraint satisfaction [62,67].

Much of the research in procedural content generation focuses on techniques that effectively replace the human designer. Approaches that are based on metaheuristic search algorithms require the formulation of a fitness function, a mathematical equation that calculates the quality (or fitness) of a candidate as a numeric value. However, some approaches have been developed with the intention of supporting or complementing the designer. Several studies have investigated the use of interactive

evolutionary computation [7,9,25,46,50], often in cases where the design intent cannot be easily formulated as a fitness function. These are specific examples of human-computer co-creation, a much broader area of research that can be seen through a number of different modes of interaction [31]. In the area of computer games, this type of cooperation has been termed mixed-initiative content creation [41], and generally is an attempt to combine the advantages of procedural generation and human creation of content. A number of researchers have adopted this approach [4,42] with the goal of ensuring that the developed systems are usable, interpretable and have an impact on real users [68].

Including a human designer in the loop is one way of bypassing the challenges of evaluating content, however the main issue with this approach is that of human fatigue [49]. To fully realize the potential of human-computer co-creation, there is still a need to be able to automatically evaluate the potential usefulness of content to enable the computer to work effectively.

2.2 Procedural Content Evaluation

Procedural content evaluation is the field of research that is concerned with the quality and assessment of procedurally generated content. It seeks to quantify how well a particular game element fulfils its purpose [57]. In this context, a game element does not have to be an asset, such as a three-dimensional model, an image (sprite), or a sound. It can also be a game mechanic, a narrative, or a game map. Creating procedural content can be undertaken using a number of approaches, with even random methods having a certain appeal [13] and more sophisticated methods being able to provide complex gameplay experiences. However, assessing whether a particular asset, mechanic, or full game is of good quality or not, is still subject to investigation.

Cook et al. [14] created a computational creative program called ANGELINA which seeks to generate games that are then assessed by players for instance through online comments. While it considers a designer perspective by applying general design principles, it does not model designer feedback into the system. ANGELINA's strength seems to be the ability to generate content and full games based on simple and sparse themes.

Several researchers are actively working on solutions that allow for a computational evaluation of content. Investigations have been conducted into assessing 2D rogue-like games [44], aesthetics for aiding computational creativity [20], and assessing platform games by using Super Mario as a case study [60] to name a few. Other works have looked at creating and benchmarking 3D First Person Shooter (FPS) games [22] or creating a framework for generation and evaluation based on player perspectives [43]. We believe that these attempts are important stepping stones toward computational domain expertise that is able to complex game content, such as FPS levels. However, there remains significant work to be done in this area.

A number of researchers argue in favor of generic methods for evaluating content such as game levels [44]. However, experts in game level design have highly domain-specific skills, and a major part of their ability to assess the quality of a level and give a founded statement about playability or other metrics about a game level, is based on this domain specificity. The research in this paper attempts to capture such domain specificity through the development of autonomous agents based on designer cognitive models.

2.3 Evaluating Procedural Content Generation

Procedural Content Generation (PCG) has garnered considerable interest in the academic research community, however much of the research in this area is focused on the technical feasibility of the various approaches and not the extent to which the generated game content is received by either game players or designers. Only a small number of studies have considered whether generated content produced playable maps or engaging content [12,21,30,52,53]. Several studies take different approaches to using player input to evaluate procedurally generated content. For example, Togelius, De Nardi and Lucas [63] use human players to train an artificial neural network that is later used as an agent to evaluate procedurally generated race tracks. An alternative approach is used by Hastings, Guha and Stanley [25] who use player data related to how often a weapon is fired in order to determine its fitness whilst generating alternative weapons. Similarly, Cardamone, Yannakakis, Togelius and Lanzi [8] utilize player behavior to infer the fitness of game levels. However, the vast majority of research in this area seems to focus on validating the output of a system in relation to its internal representations as opposed to verifying if the output is useful in a playable context.

In a similar vein, there is little research that investigates how procedurally generated content or PCG systems are viewed by game designers. Recent studies that focus on mixed-initiative creation have included game designers in the evaluation of systems [3], however the majority of studies do not. Even with interactive design approaches, many studies utilize "users" that may not be designers to drive the system [1,7,55] or simply do not include any form of user study at all. For example, Diaz-Furlong and González-Cosío describe a system that is developed to allow designers to set a difficulty level, however it is evaluated without any game designer input [18]. Similarly, Ølsted, Ma and Risi present an interactive design approach and evaluate its levels using players but there is little or no input from game designers into the design process or the evaluation [46].

A number of approaches are emerging that may allow game levels to be developed with a lower degree of designer input. For example, reinforcement learning has been used to train level-designing agents [33] or recombination approaches that utilize existing games to build new ones [24]. However, there is a growing interest in seeing how PCG tools and content are accepted, and an understanding of how players and designers can be involved [15] and still a pressing need evaluating how designers perceive and may use automated tools.

3 SYSTEM OVERVIEW

3.1 Game Context

Action games, and specifically First Person Shooter (FPS) games, have been one of the fastest growing and most successful genres in the computer games industry [11,51], with a recent study focused on the Steam database indicating that 45% of downloads from the platform were action games [65]. The research in this paper is therefore focused on designing maps for an FPS game. Whilst the game itself is not entirely relevant to the evaluation of the multi-agent design system, it is worth noting that it is a multiplayer FPS involving a simplified conquest-type (also known as "capture the flag") game mode with a single flag point. This mode of gameplay produces several issues that need to be considered during the design process. For example, if the single flag is located too close to a team's initial location at the start of game

play, referred to as the spawn point, then this may result in an inequitable player experience. These constraints have been embedded into the metrics used in the evaluation of the game levels, but it is worth noting that this would have to be adjusted for the system to be applied to different modes of game play, such as a defuse the bomb or dual flag conquest.

The maps can be considered similar to a number of commercial games, such as Insurgency, Insurgency: Sandstorm and Counter-Strike: Global Offensive. The game does not include any non-player characters and is set in an urban environment. This environment is in a deserted state that suggests the occupants of these places had only recently left, rather than fully post-apocalyptic. The streets are only slightly cluttered from the aftermath of whatever caused the population to leave, and buildings are intact, though not internally navigable in the game.

The intention of the research is to produce games that could be considered a minimum viable product, incorporating a small number of core game design elements that are common in conventional FPS map designs, and are directly related to game design elements proposed by Järvinen [29]. How the players interact in the game environment is influenced by the use of such elements, and the ones used in this system support the intended gameplay and player experience. Specifically, these elements are: the basic terrain, streets, buildings, shipping containers, and the flag. When laid out on a map, these elements have the potential to produce a player experience that is engaging as they allow variability in factors such as the degree of cover, line of sight and the presence of choke points, which are locations where close proximity is forced on the players.

The focus on producing a minimum viable product simplifies the game design process to better allow the impact of the computational agents to be determined and therefore better support the research goal of how the multi-agent system is utilized and received by the game designers. This system and the function of the computational agents is described in the next section.

3.2 Multi-Agent System

The game map design prototype in this study is loosely based on the human-based genetic algorithm framework [35] in which a human user acts as one of the agents integral to the genetic algorithm. Genetic algorithms are evolutionary metaheuristic algorithms based on models of population genetics and the principals of natural selection and survival of the fittest. A population of individuals are created and evolve through a number of generations where the better candidate solutions are more likely to pass on their heritable traits to the next generation.

The intent of the system is to augment the abilities of map designers and allow designers to explore different options through the human-based genetic algorithm implementation that includes two additional computational agents. These agents are responsible for map analysis, applying design knowledge extracted from human expert statements about their process and creating diversity among the potential candidate solutions in each population of the genetic algorithm. These agents work in parallel with a human map designer to create a minimal but playable FPS “capture-the-flag”-style game. The overall architecture of the system is shown in Figure 1.

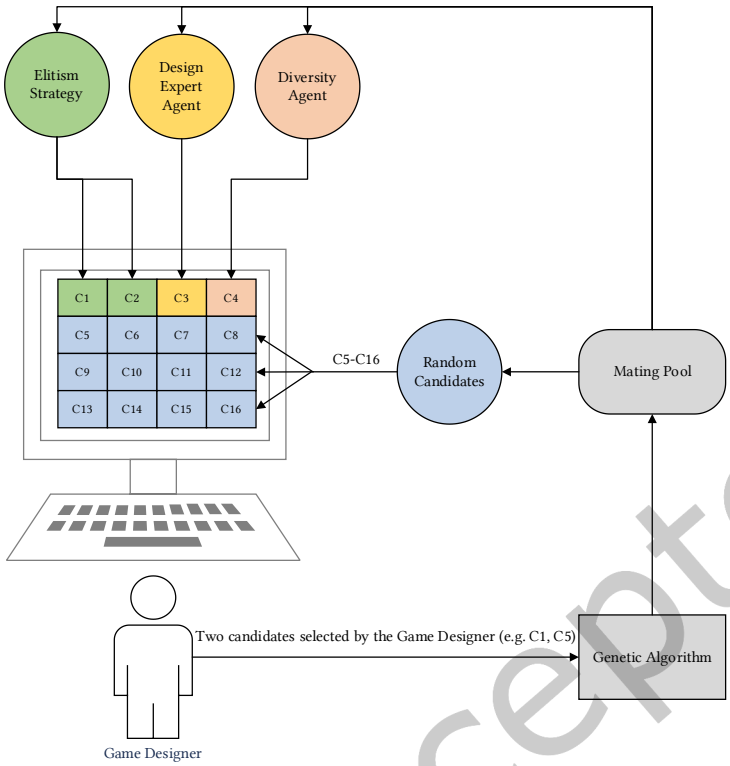

Figure 1: Schematic overview of the multi-agent game map design prototype

Unlike with traditional genetic algorithms, there is no mathematical fitness function embedded in the system that evaluates candidates for the next mating pool. Instead, it uses the output of several agents to evaluate the candidate pool and individual candidates. For each generation, only 16 candidates from a mating pool of 1000 are presented to the designer to reduce the cognitive load on reviewing a large number of candidates. From these 16 candidates, the designer selects two parents for further breeding. The system is elitist in nature, so as a result the two candidates selected as parents by the designer are carried forward into the next generation of candidates. The computational agents propose two further candidates from the mating pool based on several design heuristics, and the remaining 12 are extracted randomly from the mating pool and presented to the designer.

The candidates are presented to the user graphically in the grid format shown in Figure 1. The candidates are numbered from C1 to C16, starting from the top left-hand corner. C1 and C2 are the first two cells in the top row and would always be the candidates carried forward through the elitist strategy. C3 and C4 are the remaining cells on the first row and would be the candidates proposed by design expert and diversity agents when active. Each cell contains a 2D representation of a candidate game level with the location of the starting points ("spawn points"), flag, buildings and shipping containers. A sample of this representation is show in Figure 2. When a level is selected as a result of being clicked, a simplified 3D

render of the level is presented to the designer which allows them to rotate and view the level from different angles.

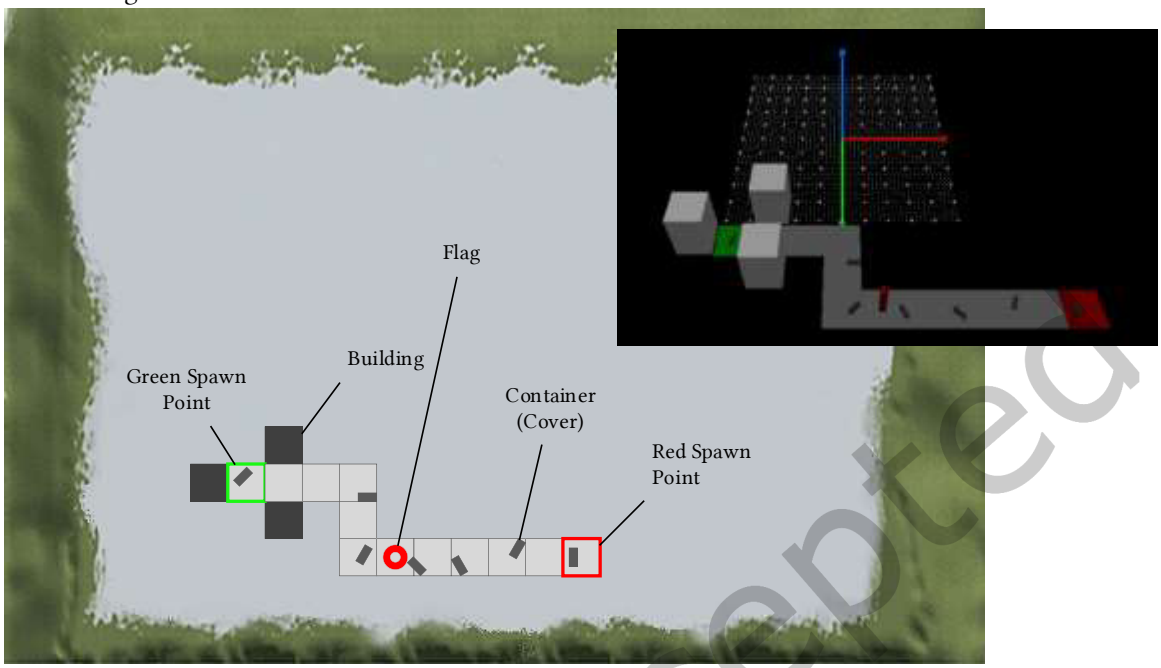

Figure 2: Single candidate level with annotations added to identify game elements and associated simple 3D render

In practical terms, the designer interacts with the system by viewing candidate solutions on the screen and selecting two solutions from each generation. This process is repeated over several generations until the designer has produced a map that they believe is of good quality. There is no option for the designers to articulate or define specific design goals into the system. Instead, they influence the system by selecting candidates that have favorable characteristics. In essence, these characteristics can be considered as heritable traits. Natural selection alters the populations over time by propagating these heritable traits through the modelling of the genetic processes. If a designer has an interest in a particular characteristic, say a map with many buildings, by continually selecting candidates with the most buildings in each generation then this would tend to produce candidates in subsequent generations with higher numbers of buildings.

The inner workings of both the design expert agent and the diversity agent have been described in previous work [24,25]. The design expert agent (DEA) is based on ideas proposed by Zhu et al. [53], and models a human map designer, and applies high-level, abstract concepts to FPS maps to evaluate their quality. This is done utilizing three visibility metrics and seven distance metrics determined from the game map as shown in Table 1. These metrics attempt to encapsulate the design constraints imposed by the mode of gameplay, however these are not implemented as hard constraints which allows the DEA to propose solutions that may violate one or more accepted design constraints that exhibit other favorable characteristics.

Table 1: Game map quality metrics

Visibility Metrics	Distance Metrics
Visible obstacles in the direction of the flag from the green spawn point (M1)	Distance from green spawn point to first visible obstacle (M4)
Visible obstacles in the direction of the flag from the red spawn point (M2)	Distance from red spawn point to first visible obstacle (M5)
Visible obstacles directly between the green and red spawn points (M3)	Distance from flag to nearest obstacle in the direction of the red spawn point (M6)
	Distance from flag to nearest obstacle in the direction of the green spawn point (M7)
	Distance between the two spawn points (M8)
	Distance between red spawn point and flag (M9)
	Distance between green spawn point and flag (M10)

The design expert agent runs independently in the system and uses digital differential analysis [45] to find intersections with solid obstacles from its current position. In addition, it tests metrics pertaining to the relationship between both spawn points and the location of the flagpole and assesses the entire breeding pool in preparation for candidate selection. The diversity agent utilizes the same knowledge as the design expert agent but works to ensure that the two proposed candidates are sufficiently different to ensure that premature convergence does not occur. The diversity agent identifies candidates that are ranked highly in the mating pool but have one of the above measurements significantly different from those proposed by the DEA. This could be levels where, for example, the path length may be very different, or one team may have much less cover than in the strongest candidate. All other parameters are kept at a high ranking to ensure that the candidates proposed are generally good levels.

To determine whether the computational agents within the system propose candidates that are of interest to the designers, the system has been designed so that the agents can either be active or inactive. If the computational agents are not active and the user is simply employed to control the genetic algorithm by selecting the candidates for breeding, the resulting system will simply be an interactive genetic algorithm [61], which serves as a baseline in our evaluation of the system. Previous work has shown that when the agents are active that the quality of the mating pool increases more quickly than when the agents are inactive [37]. This work focused on whether designers select these proposed candidates and utilize them in the design process, and this is discussed in more detail in the next section.

4 METHOD

The research design employed in this study is at its core an evaluation of the multi-agent game design prototype with participating game designers. This evaluation of the prototype tool was conducted with the goal of gaining a basic understanding of the perceived usefulness of the multi-agent system, and to gather initial feedback from professional game designers to determine how designers would receive and utilize such a system.

4.1 Participants

In total, 11 game designers were recruited for this study that covered a diverse range in terms of gender, cultural background, age, gameplay preference, and game design experience. Each designer completed a pre-participation questionnaire to capture detailed insight in terms of their computer game play preferences, their level of experience designing games, and their capacity to create FPS game levels in particular. In addition, the questionnaire captured basic demographic data to review whether there was any bias in terms of gender and age. Details of the participants' general gaming experience are given in Table 2, and Table 3 provides an overview of their game design experience.

Table 2: Participant demographics and gaming experience

Participant	Age Group	Gender	# Years Gaming	Gaming Platforms	Session Duration	Frequency	Current Games Played	First Person Familiarity
1	30-49	Male	6-10	PC, Mobile, Wii	15-30 minutes	Daily	Settlers of Catan	4
2	18-29	Male	1-5	PS3, PC, Mobile	2-3 hours	Weekly	N++ / UnderRail	4
3	30-49	Female	6-10	PS4, PC	1-2 hours	Monthly	Call of Duty	3
4	30-49	Female	1-5	PC, Mobile	15-30 minutes	Weekly	narrative	2
5	18-29	Male	1-5	ALL	2-3 hours	Daily	PUBG, Overwatch, Diablo	4
6	50+	Male	11+	PC, Xbox, Mobile	2-3 hours	Weekly	Rise of Tomb Raider, Deadlight, Return to Castle Wolfenstein	4
7	18-29	Female	1-5	PC, Xbox, PS3, PS4	30-60 minutes	Daily	The Sims 4	4
8	18-29	Male	6-10	PC, Xbox, PS3, PS4	1-2 hours	Weekly	PUBG, Call of Duty	4
9	30-49	Male	6-10	PC, Mobile	2-3 hours	Weekly	No Man's Sky	2
10	30-49	Male	6-10	PC, Mobile, Xbox	30-60 minutes	Daily	Path of Exile	3
11	30-49	Female	6-10	PC, Wii	2-3 hours	Daily	Super Mario / Switch games	3

The pre-participation questionnaire captured three age groups reflecting young adults, middle-aged designers, and older participants, who were mostly born before the dawn of home computer games (ages 18-29, 30-49, 50+). Except for one participant indicating that he belonged to the last group (aged 50+), the remaining participants were evenly distributed across the first two groups, with four in the first group (ages 18-29) and six in the second group (ages 30-49). The gender balance was roughly two thirds in favor of male participants, which possibly reflects a gender bias in the game industry. The participants in the study can be characterized as being reasonably active gamers based on the length and frequency of their gaming sessions. Each participant reported their familiarity with first person games using a 4-point Likert scale (1=None at all, 4= A lot), and most of the participants would be considered as being familiar with this type of gameplay.

Table 3: Participant game design experience

Participant	# Years Designing	Experience Level	FPS Experience Level	No. Levels Designed	Automated Tool Use
1	6-10	3	2	3-10	Semi-Automated
2	1-5	3	2	11+	Fully Automated
3	6-10	2	1	3-10	None
4	1-5	2	1	1-2	Semi-Automated
5	1-5	3	3	1-2	None
6	11+	3	3	11+	None
7	1-5	2	2	3-10	None
8	6-10	3	1	3-10	None
9	6-10	2	2	1-2	None
10	6-10	2	2	3-10	Semi-Automated
11	6-10	3	1	11+	None

Considering job requirements indicated by advertisements in the game design industry, it can be said that this study was conducted with game designers mostly at senior level (5+ years of experience) and a few at junior level (1-3 years).The least experienced designer had roughly two years of game design experience, whereas the most experienced game designer had been active in the industry for more than 21 years, and had worked on Counter Strike: Condition Zero, among other released AAA titles, which made their experience highly relevant to this study. Participants were also asked to self-report their game design experience level using a 4-point Likert scale ranging from Novice (1) through to Expert (4). As can be seen in Table 3, most participants would consider themselves to be Intermediate with the rankings being either a 2 or a 3 on the Likert scale. Similarly, participants were asked to indicate their level of experience for designing FPS games on the same 4-point Likert scale and the number of such game maps they had designed, as well as whether they had previous used any type of design automation tools.

4.2 Research Design

This research was based on a two-stage design, with the first stage being an observation of the interaction of designers with the system and the second being based on follow up interviews. This is a mixed-methods study with both qualitative and quantitative data collection in the observation stage. The overall design of the study is shown in Figure 3.

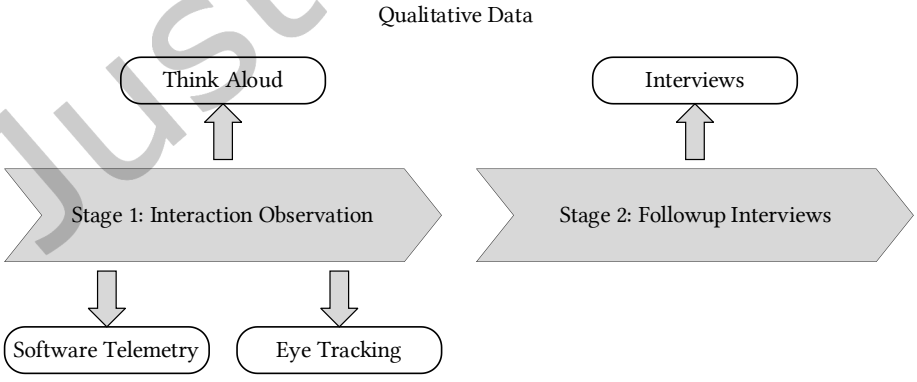

Figure 3: Overall research design

The interaction observation was conducted with the participants using the game map design tool. During this stage, participants were asked to use the prototype several times ("runs") until a candidate was created that was deemed a playable and potentially enjoyable level. First, the game designers were introduced to the prototype tools, and features and limitations of the system were explained. The designers were made aware that the computational agents would be active in some of the tests, however, to not bias their behavior, they were never informed when the agents were active.

The participants were then asked to conduct their level design test by having them select the candidates they wanted to be the parent candidates for the next generation of the genetic algorithm, to think aloud whilst doing so, and to indicate when they had achieved a level that they would be happy with. In addition to the think aloud protocol, software telemetry and eye tracking data was collected during each run to facilitate triangulation of the data to gain a deeper insight to how the tools were used by the designers.

During the evaluation, half of these runs relied only on the human designer and had no other agents active, while the other half of the runs had all computational agents active to support the human designer. Using no agents and letting the designer essentially conduct the entire selection process, without any suggestions and interference by computational agents created a baseline for comparing these runs with the full system being active.

The second stage of this research involved semi-structured interviews with the designers intended to explore their reflections on using the game design prototype. This interview was designed to capture any thoughts, ideas, and views about the prototype and the design process that could not be captured by any other means.

4.3 Data Collection and Analysis

Each of the data collection methods shown in Figure 3 are introduced in more detail in this section, along with a summary of the data analysis methods.

4.3.1 Software Telemetry.

A wide range of software telemetry data was collected during all runs during the observation stage, which included key presses and mouse clicks in addition to recording the current state of the system. The latter included the fitness and ranking of candidate solutions for all generations that allows the change of overall population fitness to be determined over the course of a run. The mouse click data captured which candidates were being selected as parents by the designer from the options presented to them in the grid shown in Figure 1. This allowed the frequency of selection by the designer to be determined for each cell in the grid and facilitated comparison for when the computational agents were both active and inactive, which effectively determines to what extent the designer is utilizing the candidates proposed by the agents.

4.3.2 Eye Tracking.

Eye tracking is a quantitative method for capturing gaze data from participants. It uses infrared cameras to capture the position of the pupils to allow a very accurate estimate of what the participant looked at on the screen [19]. This data can be considered a good proxy for the visual attention of the user [23], which provides cues to what the participant is actually interested in. Eye tracking has been used to minimize

human fatigue in the context of interactive evolutionary computation [48], however in this research the intent is simply to identify to what extent the designer is looking at the different map candidates.

While the mouse click data indicates the actual decisions made by the designer, the eye tracking data provides continuous insight into what the designer physically looked at while forming the basis for these decisions. Instead of a snapshot of a decision event it adds a temporal component that captures the time during which the decisions had been formed, rather than the result of this decision-making process.

There are several ways to visualize the tracking positions, the most common being the use of heatmaps. These (often colour-indexed) representations show a stronger colour and density where gaze had been focused most of the time, and a weaker colour and lower density for where the user looked at less. The heatmaps can be superimposed onto the original screen material to visualize the gaze pattern of the user in context.

4.3.3 Think Aloud.

The participants were encouraged to think aloud during their attempts to design maps using the system. According to Charters [10], "Think-aloud is a research method in which participants speak aloud any words in their mind as they complete a task", which can be traced back to the concept of 'inner speech', originally noted in 1962 by Vygotskiĭ [66]. The premise is that words and phrases captured through the think-aloud approach serve as close representations of inner speech, and are often expressed as incomplete sentences, but as a nonetheless useful reflection of a person's thought processes.

Data collected via the think-aloud method was coded based on a selective, reduced data collection approach. The coding process was conducted in NVivo using the three stage process described by Saldana [54]. The pre-coding stage is essential for gaining an initial understanding of the data. It serves to form an understanding of phrases that are common, and themes that can be identified across all responses.

The second stage of analysis seeks to assign labels to nodes. In this study, interview sub-questions were used to develop a-priori initial labels that were revised and updated throughout the coding process. The final stage is the post-coding stage, which drives how findings are presented. There are various ways in which to approach post-coding, including project maps, concept maps and charts, or by presenting the findings as closely related to the wording used by participants [54]. The latter method was chosen for the simple reason that the participants are experts in the field of game design, which resulted in a fairly consistent choice of terms.

4.3.4 Semi-Structured Interviews

The interview questions were intentionally designed to allow the participants room to elaborate and talk about their experience. As previously noted, the designers were not aware during the tests whether the computational agents were active or not. In addition to general usability, a key aspect of the interviews was to try and determine whether the designers had perceived any difference between their runs that could be attributed to the agents. As a semi-structured interview was undertaken, the guiding questions shown in Table 4 were used as a basis to ensure that consistency was achieved between interviews. However, the interviews were allowed to extend beyond these questions based on the responses provided by the designers.

Table 4: Interview questions

Question #	Question Text
1	You have just completed a design experiment using a semi-automated level design tool.
	a. How would you describe the overall experience?
	b. How difficult was it to use the tool?
	c. Do you think that you have achieved the levels that you set out to create at the start of the experiment?
2	You have completed several levels.
	a. Did you find it easy to make very different levels?
	b. Did you notice a difference between the different design tests?
	c. Do you think that you created any playable levels, if so, which ones?
	d. Do you think that you made any un-playable levels, and if so, which ones?
	e. Regarding 2.c. and 2.d., why do you think that is the case?
3	These tools use autonomous agents that are small smart programs that attempt to understand your preferences during the design process. These agents were only active in some of the tests.
	a. Did you notice a difference?
	b. Do you think autonomous agents are a useful addition to this tool?
4	What would you change regarding the usability of the tool?
5	Did you like the fact that you had only simple choices available to you, e.g., just being able to choose two 'parent' levels to create new levels?
6	What would you change regarding the responsiveness of the tool with regards to your design goals?
7	Do you think that a tool using semi-automated design processes is useful or an obstacle to game level design?
8	How do you envision the ideal semi-automated design support tool?

The qualitative data from the interviews was subject to the same coding and analysis protocols as the think aloud data.

5 RESULTS

The results of the two stages outlined in the method section are presented here starting with an overview of the number of scale and scope of the results. In total, 31 runs were completed with all designers completing at least one run with the agents active and one run without. Of the total number of runs, 15 were completed with agents active and 16 without. There was a high degree of variability in the number of generations this took, with the shortest run being just 12 generations and the longest 76. Each designer conducted at least one attempt to design a level both with and without the agents active.

The following sections outline the results relating to the software telemetry, eye tracking, think aloud and interview data collection. The quantitative data from the software telemetry and eye tracking essentially describe the objective behavior of the designers whilst the qualitative data from the think aloud and interviews focus more on their subjective beliefs and perceptions of the system.

5.1 Software Telemetry Data.

Figure 3 shows mouse click data for the evaluation of the system when the agents are not active, where the first candidate corresponds to the cell C1 in Figure 1, the second to C2 and so on. The colors used in the bar chart are consistent with those in the cells in Figure 1 to provide an additional visual reference. The selection frequency is biased towards the elitist parents carried forward from the previous generation, where the candidates selected by the designer are maintained in the generation and presented in cells C1 and C2. Nearly one third (27.1%) of all selections made were the elitist parents. In many cases, the designers would have selected a single parent from the previous generation to attempt to retain desirable

characteristics and trying to introduce a new parent from the random candidates to introduce new features.

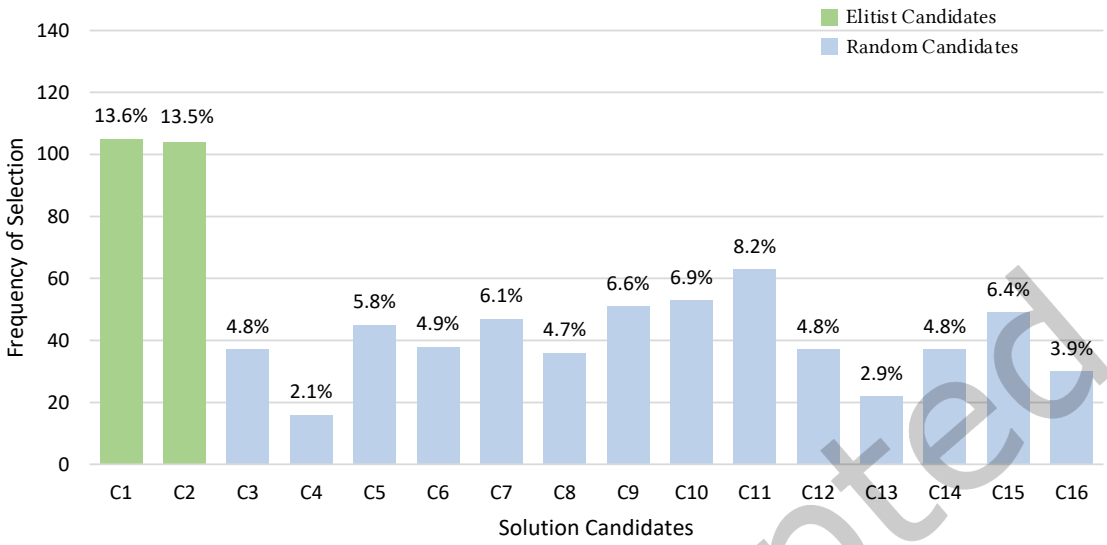

Figure 4: Frequency of candidate selection without agents active

Figure 4 presents the frequency of candidate selection from the mouse click data for the evaluation of the system when the agents were active. Whilst the elitist parents are still selected the most by the designers, there is a distinct increase in the number of times that C3 and C4 are selected. These cells contain the candidates proposed by the agents, again using the same colors as with Figure 1.

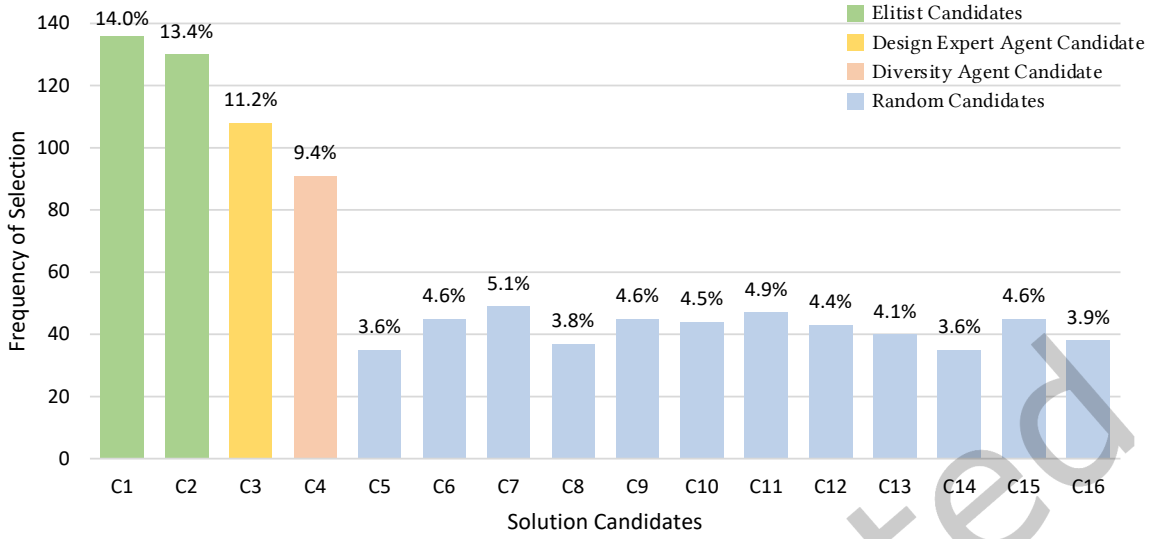

Figure 5: Frequency of candidate selection with agents active

With the different length and number of runs, there have been a different total number of selections. In percentage terms, the selection of elitist parents is similar to when the agents are not active (27.4%). The candidates recommended by the agents account for just over one fifth of all the selections (20.6%) and are therefore reducing the number of selections made from the random candidates. Observation of the combination of selections made by the designer confirms that in many cases the participant selected a combination of an elitist parent and a candidate proposed by the agent in an attempt to direct the process towards potentially novel outcomes.

5.2 Eye Tracking Data.

An eye-tracking heatmap was produced for all 31 evaluation runs, however for brevity only a subset of representative examples are included in these results. Six examples were considered enough to show the variation and diversity across all of the runs. Figure 6 shows six examples of eye tracking data with an active multi-agent system where the grid corresponds to the way that candidate solutions are presented to the designer as presented in Figure 1. In each image in Figure 6, the top row contains two candidates brought into the generation through the elitist strategy in the most left two cells (C1, C2) and candidates suggested by the design expert and diversity agents in the two rightmost cells (C3, C4).

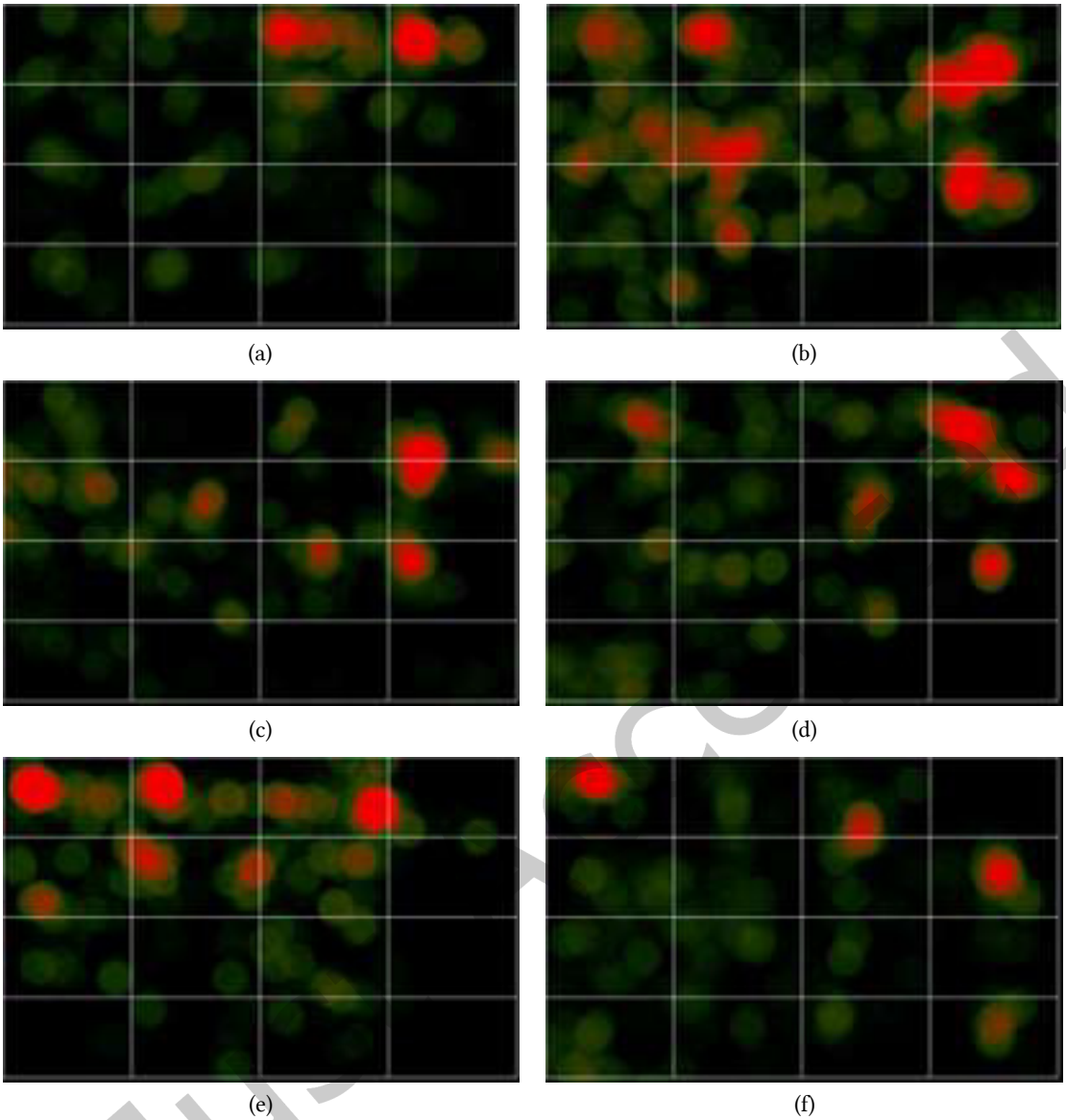

Figure 6: Eye tracking heatmap with agents active

Each of these runs are relatively long, which results in a denser heat map overall. In Figure 5(a), the top left cells, C1 and C2, show light heat which indicates very moderate interest on the part of the user, as does C7. C3 and C4, on the other hand, show very intense heat, an indication that the user had an elevated interest in these candidates.

Figure 5(b) demonstrates a very different pattern, with several tiles showing a strong signature which is the result of being the longest run of the results. A number of tiles have large red areas, for example, C2, C4, C6, C12. This would suggest that the designer is not using the agent recommendations as much.

Interestingly, data from the talk aloud protocol shows that the participant was looking for "*something different*" and want to "*push the algorithm into a different direction*" which explains their interest in the random candidates.

The pattern in Figure 5(c) shows a strong marker on C4, but also moderate to strong indications of some random candidates. This suggests that the designer had some interest in the candidates presented by the agents. A similar pattern is shown in Figure 5(d), though here the designer is also showing some interest in the elitist candidates. This interest is more apparent in the run shown in Figure 5(e) where the designer is very interested in the elitist candidates, as well as those presented by the agents and some other random candidates. It is possible that the bias towards the top left is an indication of normal reading of text, scanning from left to right and down the screen. A similar pattern is shown in Figure 5(f) where the designer looked a lot at one of the elitist parents and potentially one of the ones proposed by the agents.

Whilst there is a general pattern here towards the upper row, the extent to which the designers are utilizing the candidates proposed by the agents can only be evaluated through comparison with runs where the agents were not active. Figure 7 shows six such runs without the agents active. In contrast to Figure 6 the top rightmost cells now contain random candidates and not those proposed by the diversity and design expert agents.

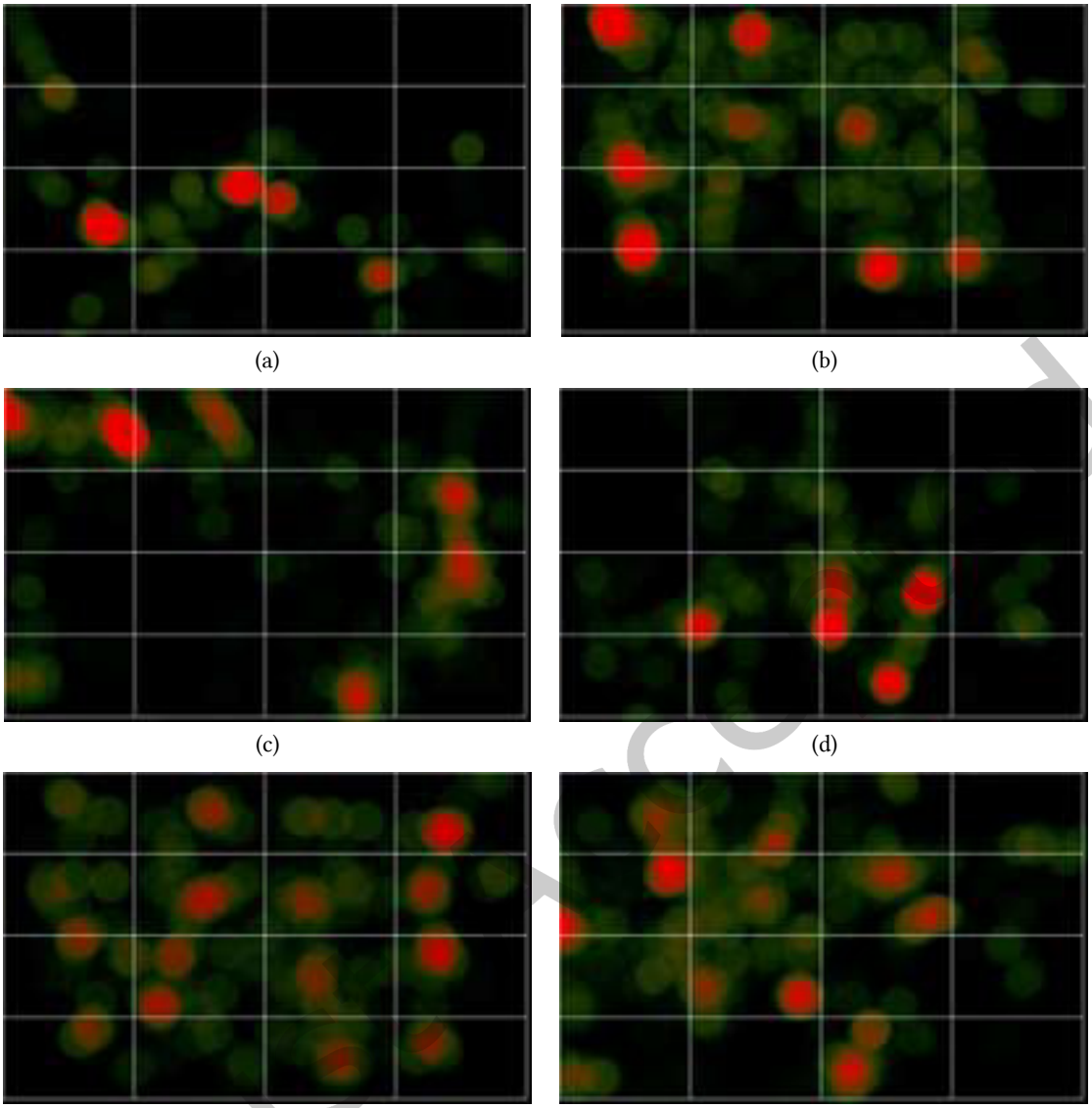

Figure 7: Eye tracking heatmap with agents inactive

In comparison with the results presented in Figure 5, there is a much broader interest in the candidates presented to the designer. In some cases, this still includes the elitist parents suggested in C1 and C2, however the random candidates in C5-C16 are receiving considerably more attention as a result of the agents not being active in comparison to the interest in C3 and C4 as seen in Figure 6.

5.3 Think Aloud Data.

For all 31 of the runs undertaken, all spoken comments were noted in addition to observations of non-verbal frustration and enjoyment. All such comments and non-verbal indications were related to a given generation of the run, and where possible a specific candidate solution that had been presented to the designer. The focus of this paper is an analysis of the spoken comments. Some designers provided a high level of commentary, others were less verbose.

During the coding of the comments two themes emerged. The first of these related to the goal that the designers were hoping to achieve, the second were comments that related to the computational agents. Following the labelling of the data, various comments that can be considered as representative of the themes have been selected to illuminate the issues being commented on by the designers.

5.3.1 Stated Design Goals.

Some participants stated an explicit design goal, sometimes involving several player experience goals. For example, one participant stated that they "*really [want] a small level where the teams are immediately facing each other*" in order to provoke a very fast-paced game, where players do not survive very long. This was different to most design goals, but reflects the style of some small scale, fast-paced FPS levels from Battlefield and Call of Duty.

Another game level designer aimed to create a "*street that is long and winding*" so that "*there are many corners for cover*". This implies the assumption that most corners will have either buildings or street elements that offer shipping containers as elements for cover, which will be a second design goal for the same level. The concept of cover came up often, with several designers suggesting at the start of their design process that "*players should have as much cover as possible, so more buildings are needed*".

Some goals were more abstract, such as "*I just want to make something silly, not your typical map*". Clarification of this intent led to the designer stating that the level should have large open fields without much cover, and a long and straight street to make the map very linear. While this may go against best practice for FPS level design, the participant was not only able to achieve this design goal, but also felt that the final result would be "*an interesting addition to a map package that contains more traditional map designs. It would be very quick games though*".

Another similar goal, achieved using a different approach, was announced as "aiming to create a 'mean' level, where both spawns are close to each other, and both are far away from the flag". This kind of design will "*probably introduce a very strategic approach. You [the player] can't just rush in otherwise you die quickly*". Here it can be seen that some design goals were based on assumptions that can ultimately only be verified by playtesting.

Some participants were quite specific about typical elements that create mechanics similar to many popular games, and that are in line with what expert designers suggest for FPS levels. For example, one participant stated that they "*only want a building if it provides cover at a choke point. Otherwise no buildings, just containers*".

5.3.2 Computational Agents.

Several comments were made that relate to the support provided by the computational agents, and in other cases, comments on their absence. These comments can be categorized into two groups, one reflecting the

agents helping the designer, and the other related to the agents being inactive. The participants were made aware that the agents would be active and inactive at times, but they did not know when this was the case. Thus, participants were not deceived, but had no knowledge about the state of any particular run.

When the computational agents were active, comments such as "*Oh, it understands me*" or "*It feels like every new generation is much better than the previous*" were made. In particular, when participants looked at and also pointed to the two agent candidates in C3 and C4, comments such as "*There are a couple of good levels here*", "*Oh, that is a good one*" and "*I like this one*" were made. A more generic "*The top row is generally the most interesting*" was consistent with both elitist parents and two computational agent suggestions in C1 to C4. These comments were found to be consistent with the eye tracking data, which is presented in section 5.2.

Statements in stark contrast to those mentioned above were made in runs where the computational agents were inactive. This was, again, reasonably consistent with the eye tracking data as per the following remarks: "*It is a bit random, there was a better solution in the previous generation I think*", "*It just does not improve enough from selection to selection*", "*The tool does the same thing over and over. There is not enough change*", "*These candidates are all very random*" and "*I am not getting any closer to what I want*". These participant statements reflect the common theme dominant in non-agent runs. Some mild frustration was observed in particular when it came to particular design goals, such as "*Trying to give the red team some cover, but that is not happening*". In other words, the designer was unable to pursue a specific design goal in this particular case.

This can not necessarily be ascribed to the impact of the multi-agent system (or in these cases, the lack of a multi-agent system); rather, it may simply have been normal frustration and fatigue, which is to be expected in a pure interactive genetic algorithm, which can take a large number of runs to eventually converge on a set goal [39]. Through indirect interaction with the artefact that is being designed (in this case, FPS game levels), some designers seem to experience a sense of loss of control, simply because the genetic algorithm often takes multiple runs to modify a simple change, which could have been manually altered very quickly. However, this is only true for some specific changes. The genetic algorithm is in principle very fast when it comes to variations and idea generation. It is simply the perception of the user that makes it appear to take longer, due to a lack of direct interaction with the artefact.

During each use of the system, the participants were observed to help identify the context of their think aloud commentary. In general, participants seemed to favor the top right corner of the screen (where the agent suggestions were located) significantly when the agents were active. This observation appears to be supported by some think-out-loud comments made by the designers, specifically and unequivocally when the agents were active. It can be said that the response to the multi-agent system, based on observations and think-out-loud recordings, was overall very positive. In contrast, participants seemed to randomly select candidates when the agents were not active. This behavior was noted several times in the observation protocols and suggests that the use of the system when the agents were inactive was less directed. However, there was also a single case in which the active multi-agent system did not perform according to the very straightforward design goal of "*some cover and more bends in the pathway*" at all. This was stated as an issue by the participant, but is likely an isolated case where the algorithm converged into a local minimum without any means to escape.

5.4 Interview Responses

This section presents the qualitative data of the evaluation of the game level design prototype from the interviews. The interviews were focused on a number of areas, including the usability of the prototype, the effectiveness of the computational agents and feature suggestions.

5.4.1 Usability and User Experience

It is important to note that the overall user experience includes the multi-agent system; therefore, to some extent, the responses reflect the user experience related to immediate interactions such as user interface, and mouse and keyboard, but also include indirect critiques of the underlying genetic algorithm and computational agents. The overall experience was described as "*intuitive*", "*encouraging with a lot of potential*", "*offers possibilities*", "*creative*" and "*enjoyable*". The designers also unanimously stated that the tool was easy to use.

Response to questions about whether the system allowed them to achieve their design intent were generally positive, with comments that it was "*easy*" or "*possible*". However, one participant stated they took a "*playful approach*" and therefore did not set any initial design goals, but rather set out to see what emerged. Even participants who were extremely reserved in their assessment of the computational agents, and believed that the tool would not be very useful to them, still found the overall experience positive and the tool "*fun to use*".

Furthermore, participants stated that they could easily create different variations of the same map and liked the possibility of discovering new ideas very quickly. However, one participant criticized the system because they felt it reacted very aggressively when changing selections. This would not be unexpected considering the parameterization of the genetic algorithm.

5.4.2 Computational Agents

The responses to the multi-agent system were quite mixed. It was interesting to observe that, when asked about the agents, designers who claimed up to 10 years of experience showed a positive tendency in their responses whereas designers with more than 10 years of experience made significantly fewer encouraging statements. Long-standing experts thought that they did "*not need any algorithm to support [them]*" when conducting the trial. They felt that they could find "*good choices*" on their own, and that the previous selections in C1 and C2, as well as the suggestions made by the agents in C3 and C4, were not needed, and that they "*selected different levels than those at the top anyway*". Designers with less than 10 years of experience were more enthusiastic and thought that "*the agents made good suggestions*", but these designers also realized that this was not true for all runs and could clearly identify those in which the agents were inactive. Participants responded that some runs took a long time, as "*no good levels were presented by the system for many generations*". Designers seemed to recognize that the system was not always performing as they had hoped, and from observation protocols, it is clear that these runs were without agent support. For example, one participant stated that the multi-agent system offers a "*playful approach*" and that it is "*easy to find some interesting suggestions*", even though "*some did not turn out very good*", which was ascribed to a "*lack of options by the algorithm*". The comment (made in a case where the agents were inactive) makes sense in light of five runs having been conducted by this individual, of which two did not have the agent system active. The observation protocol confirms that the two sessions without

agents were perceived as slow and difficult. This participant also made a very interesting comment about "*best design practice*", which an algorithm should implement. The comment included a suggestion to look at existing maps "*like Dust2 and analyze classic elements and suggest them to the designer*". The heuristics that were implemented in the DEA were derived as a result of a qualitative analysis of expert game designer opinions available in the public domain, including from the designer that developed the Dust2 map, so whilst the agents appear to be useful there is perhaps scope to provide more concrete guidance based on specific map features during the design process.

Finally, five participants indicated that the system "*offers interesting alternatives*" and "*rather unexpected*" alternatives in the top left corner, where the diversity agent seemed to have offered a solution that was different to the designer expert agent, and both candidates were considered compelling.

5.4.3 Feature Suggestions

Throughout the interviews, several of the designers made suggestions on how the tool could be improved. Those suggestions formed three categories: improved visual aids, workflow improvements, and specific game level design functionalities.

In terms of visual aids, participants indicated that the visual guides were not sufficient for judging the map and suggested that it would be helpful to see the grid that is being used to place elements such as streets, buildings, or spawn points. Such a guide would provide a quick way to evaluate, for example, path length, distance of spawn points to the flag pole, and other similar metrics that participants use to assess the quality of a candidate, in order to make a decision for or against it. A second aid that has been requested multiple times is a debug overlay that shows what the agents do, and how they assess the levels. This was precluded from the prototype design as a necessity to hide when the agents were active, however there is a good argument that a designer needs to understand the agent system they are working with in order to maximize its benefits to the design process [56] and as such this feature suggestion might improve the overall reception of the system.

Workflow improvements capture a range of general suggestions that are related to the workflow implemented in the prototype, rather than suggestions specific to game level design. The most common improvement asked for was an undo feature, which is technically possible given that the software telemetry data included all the population data for each generation. There could be some conceptual arguments against this in terms of how natural selection is the core of a genetic algorithm approach and evolution is a progression, however most digital tools offer virtually limitless undo steps and so this could be a convenience for the designers that can save them time overall whilst also exploring the implications of pushing the algorithm into a different direction. The ability to exclude certain candidates from future breeding is another workflow improvement that has been requested several times which may also lead to an even faster convergence to a final solution.

Feature suggestions that relate to the specific game level design functionalities are the most useful to this research, as they provide the greatest insight as to how game designers receive the tool as a whole system and how they wish to work. Two main themes emerged from the interviews that related to wanting to set constraints prior to initiating the genetic algorithm, and second, constraints that users add while conducting the selections for the algorithm.

Several designers asked for additional manual control of the level design process. Two participants suggested the introduction of design constraints, which advanced users could define prior to using the actual design tool. Possible constraints included the length of the path, how much branching occurs in a path, where the spawn points are, and so on. Certainly, many of these elements could be set as either hard or soft constraints, however this changes how the tool would be used and it becomes more directed rather than exploratory.

Two participants also suggested that it would be helpful to be able to lock spawn points at any point in any run so that both spawn points remain in their current grid location, and the system evolves only the other features such as path length, path curvature, and obstacles through the genetic algorithm. Similarly, it was suggested that the manual selection or deselection of buildings and containers would be useful.

6 DISCUSSION

The overall goal of this research was to evaluate how the multi-agent system is utilized and received by game designers, with specific focus on whether the computational agents within the system propose candidates that are of interest to the designers, how the designers react to the whole system, and whether it allows them to design playable maps. The results are briefly summarized, and the implications of the results discussed in the following sections.

6.1 Summary of Results

Both the software telemetry data and the eye tracking data collected in this study show that the computational agents propose candidate solutions that are of interest to the designers. When the agents were inactive, the designers tended to look more broadly across the 16 candidate solutions and tended to select the elitist solutions from the previous generation. When the agents were active, the candidate solutions proposed by the agents were selected with a frequency that was comparable with the elitist solutions.

The qualitative data collected through the think aloud protocol would tend to support the effectiveness of the agents. However, the interviews indicated that expert designers appeared to reject the idea that the computational agents were helpful, even when shown evidence to support they were helpful. Expert game level designers seemed to believe that they needed to rely on their expertise more than 'metrics and algorithms'. However, less experienced designers tended to be more positive in their reception to the system and the computational agents. Generally, the system was considered to be usable but still able to be improved with the addition of new features.

6.2 Insight and Implications

The results in themselves make clear statements around the effectiveness of the computational agents and how the system is received by the designers. However, it is the triangulation of the data and comparison that provides the most insight. In particular, there is clear contradiction between how expert designers behave and what they believe. The two most senior designers that took part in this study both strongly rejected the idea that the agents were of any value, and confirmed they would only follow their own experience. They were also not able to tell the difference between any of the runs. Interestingly, both of these designers made heavy use of the agent-based suggestions, and looked at them much more frequently

than any of the other candidates. This may have been simply because the agents were proposing candidates that aligned with the design goals for these designers, a potential endorsement for the design of the agents. Certainly, expert designers actions are "high-level mechanisms based on heuristics and assumptions learned from professional experience" [6:163] and the computational agents are based on similar heuristics. However, it is also possible that the designers have a certain mistrust of computational support tools. Bernal, Haymaker and Eastman [6] indicate the computational support for designers needs better representation of the tacit aspects of domain knowledge as well as being designer-centric as opposed to design-centric. Whilst the system described in this paper is based on tacit knowledge, that representation is hidden from the designers. Similarly, the system was designed around the game itself and not necessarily the way that game designers choose to work. Both factors may have influenced the level of trust that the designers have in the approach. Trust has been identified as an important variable that links automated systems, their use, and consequent performance [34] and as such it is possible that no matter how well a system performs that if there is no trust in the system then it may never be truly accepted.

In addition to trust, an aspect that needs to be considered here is that of control. A limitation of interactive genetic algorithms is that there is no way of specifying a particular goal and no means to control the process to achieve that goal other than selecting solutions to try and reinforce their characteristics in the next generation. For expert designers, this could be a step too far away from the certainty of their established ways of working. Some authors suggest that these established ways of working are best considered as habits, and have a strong influence on the perceived usefulness and usability of systems, and that the "uncertainty involved in exploring new ways of working may be less attractive than following a familiar though somewhat cumbersome routine" [27:576]. The fact that several designers indicated that they wanted more control over the process suggests that there is a degree of discomfort in working with tools that rely on the emergence of solutions.

This raises several questions about the purpose of design support tools and how they are best integrated into the workflow of game designers. When genetic algorithms are used in design support tools they often are framed as 'novelty generators', often based on the view of Bentley that evolution is best viewed as continuous novelty generator and not as an optimizer [5]. This would certainly be in line with how several of the designers used the tool in this study, particularly those that adopted a more playful and exploratory approach. However, other designers were more focused on reaching their particular goal and expected to be able to shape and manage the process of reaching that goal. Both approaches could be implemented in the same system by incorporating a set of variable constraints that could be chosen to be applied, as well as offering up the ability to change the parameterization of the generic algorithm and the computational agents. The latter could include, for example, allowing the user to choose different mutation strategies that were intended to produce either incremental changes that might improve the solutions in play or more radical mutations to promote the inclusion of more novel candidates. However, such changes are based on assumptions of what game designers want and how they choose to work and may suffer from the same challenges in terms of acceptance and adoption.

6.3 Contribution

There are two main contributions that arise from this research. On an implementation level, it has been shown that the computational agents appear to have suggested levels that the participants responded

positively to. This research shows that a simple approach such as using heuristics rather than more sophisticated, deep neural network-based machine learners, can create an effective cognitive model of expert designers. In terms of one of the goals of this research, namely whether the computational agents within the system propose potential candidates that are of interest to the designers, then the quantitative data would support that this has been achieved. There is also evidence within the qualitative data that, in general, the use of the agents assists the designers in the production of playable levels.

The second contribution arises from the interpretation of the contradiction between the quantitative and qualitative data. In the introduction to the paper, it was suggested that tool adoption is only likely to be successful if the designers felt that there was value in the use of the system. This needs to be linked with the observation in the background section of this work that it is common in work of this type that generated game content is not evaluated using either game players or designers. Whilst this study isn't focused on the game content, but the system of producing it, the use of designers to evaluate the system has shown a certain level of rejection of the methods, particularly by more experienced game designers. Here it would seem that the goal of this research to understand how designers react to the system gives rise to the potential of a different way of working. A significant proportion of the research in the area of procedural content generation is undertaken by researchers in academic environments making assumptions about what game designers want and how they work. The second contribution of this work is therefore the suggestion that for procedural content generation research to find greater acceptance then there is an opportunity to consider when designers become involved in the process. The potential for considering how game designers work, what they need to support their work, and the co-creation of systems to do so is considered in section 6.5.

6.4 Limitations

One of the main limitations of this study is the number of designers that participated. There is growing interest in how qualitative studies can address issues of generalizability [47] and this perceived need is acknowledged in this paper, albeit not directly addressed. In terms of the participants, there is high degree of diversity in their makeup in terms of gender, age, experience, and gameplay preferences. However, when viewed in terms of whether the participants are representative of the complete population of game designers then the conclusion would have to be that the representation, whilst diverse, is certainly sparse. That would bring in to question the generalizability of this study and it is acknowledged that a different set of designers would likely have completely different responses to the system. Certainly, in the process of coding and labelling the qualitative data there was no evidence of data saturation [26] in the think-aloud or interview data. Whilst no formal method was used in evaluating the degree of saturation, again a limitation of this work, it would seem that the sample size is not sufficient for it to be considered anything other than exploratory in nature.

Similarly with the quantitative data, the analysis of the data is limited as a result of the number of participants. The frequency of selection of candidate solutions is a macro-level measure of how the participants made selections and used the proposals of the computational agents. A more fine-grained inspection of the data was undertaken, looking at individual designers on individual runs, but the size of the dataset makes it difficult to determine anything from this with any certainty. A larger study, with

either more participants or more runs, would facilitate a statistical analysis to be undertaken. This could be aimed towards understanding difference in behavior between expert designers and less experienced ones.

6.5 Future Work

There are two possible directions for future work to take, that align with the two contributions of this paper. The first focuses on further developing the system on the assumption that with more features and better visibility that it could be better received by game designers, or at least focus on adoption by less experienced designers. The second direction is based on the idea that designers need to be involved in the design of the system, not just the evaluation.

The various feature suggestions proposed by the designers involved in this study could easily be incorporated into the system. It would be interesting to see whether providing more visibility of the workings and actions of the agents would lead to a better reception with expert designers. Similarly, it would be interesting to gauge the impact both in terms of reception and performance if the introduction of more control through the ability to include or add constraints. The extension of the system to include more agents is one way that could be considered as balancing the tradeoff between novelty generation and goal achievement. The introduction of a user-preference agent [36] has always been considered beneficial in terms of reducing designer fatigue and speeding up the design process. However, there is potential for introducing the agent as a means to establish a better value proposition with expert designers. Would the idea of a system that learns their specific preferences be more appealing given that it attempts to encapsulate the tacit experience of the designer through their selections? An interesting idea would be to conduct controlled studies to determine this, potentially even using a similar design to that of Denisova and Cairns who examined whether the placebo effect is experienced in games [17].

As well as developing the system further, there is an opportunity to consider how it is evaluated. As well as a limited number of designers used in evaluating game content, there are also few studies in the area of procedural content generation that utilize actual player evaluations [12]. Extending the scope of the game design output to beyond a minimum viable product and using players to evaluate the quality of the game maps is valuable in its own right. However, this also opens up further possibilities. For example, if the system is considered a novelty generator, then player feedback on 'alpha release' of game maps could be collected and fed into a player preference agent that then could be used to improve the design process by proposing levels to the designer that are similar to, but different enough, from maps that have already been established as popular. It is worth acknowledging that extending the game complexity may also impact the extent to which game designers perceive the value of the system.

An alternative, and not mutually exclusive, direction would be to reconsider how game designers work and what they need in a game design support tool. In other field of design there have been various studies that have examined the difference between how novice and expert designers approach design [2,32]. Potentially a study like this may provide further insight as to why expert designers seem to be less responsive to the system described in this study, and indeed may also lead to a better understanding of what is required in a design support tool. Ultimately, the most value in terms of ensuring the success of a game design support tool may be realized through a formal co-creation process where game designers are involved in the design of the tool itself.

7 CONCLUSION

The goal of this research was to evaluate how a multi-agent design support system is utilized and received by game designers and this was broken down into three intermediate goals: 1) do the computational agents within the system propose potential solutions that are of interest to the designers? 2) does the system allow the designers to create playable levels? 3) how do the designers react to the whole system?

In terms of the first of these intermediate goals, the quantitative and qualitative data collected indicates that designers of different levels of expertise both examine and use the candidate solutions proposed by the computational agents during the design evolution process. Generally, designers found the system easy to use and managed to achieve their goals whilst using the system. However, some designers indicated that they felt that they did not benefit from the use of the system. This opens up questions as to when and how game designers should be incorporated into the design and evaluation of design support tools, and also to develop a better understanding of how designers of different levels of experience work and how they would adopt such tools in practice.

REFERENCES

[1] Chad Adams, Hirav Parekh, and Sushil J Louis. 2017. Procedural level design using an interactive cellular automata genetic algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ACM, Berlin, Germany, 85–86.

[2] Saeema Ahmed, Ken M Wallace, and Lucienne T Blessing. 2003. Understanding the differences between how novice and experienced designers approach design tasks. *Res. Eng. Des.* 14, 1 (2003), 1–11.

[3] Alberto Alvarez, Steve Dahlskog, Jose Font, Johan Holmberg, Chelsi Nolasco, and Axel Österman. 2018. Fostering creativity in the mixed-initiative evolutionary dungeon designer. In *Proceedings of the 13th International Conference on the Foundations of Digital Games*, ACM, Malmö, Sweden, 1–8.

[4] Alexander Baldwin, Steve Dahlskog, Jose M Font, and Johan Holmberg. 2017. Mixed-initiative procedural generation of dungeons using game design patterns. In *2017 Conference on Computational Intelligence and Games*, IEEE, New York, USA, 25–32.

[5] Peter Bentley. 1999. Aspects of evolutionary design by computers. In *Advances in Soft Computing*, R. Roy, T. Furuhashi and P.K. Chawdhry (eds.). Springer, London, UK, 99–118.

[6] Marcelo Bernal, John R Haymaker, and Charles Eastman. 2015. On the role of computational support for designers in action. *Des. Stud.* 41, Part B (2015), 163–182.

[7] L. Cardamone, D. Loiacono, and P.L. Lanzi. 2011. Interactive Evolution for the Procedural Generation of Tracks in a High-End Racing Game. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, ACM, Dublin, Ireland, 395–402.

[8] L. Cardamone, G.N. Yannakakis, J. Togelius, and P.L. Lanzi. 2001. Evolving interesting maps for a first person shooter. In *European Conference on the Applications of Evolutionary Computation*, Springer, Como, Italy, 63–72.

[9] Luigi Cardamone, Pier Luca Lanzi, and Daniele Loiacono. 2015. TrackGen: An interactive track generator for TORCS and Speed-Dreams. *Appl. Soft Comput.* 28, (2015), 550–558.

[10] Elizabeth Charters. 2003. The use of think-aloud methods in qualitative research an introduction to think-aloud methods. *Brock Educ. J.* 12, 2 (2003), 68–82.

[11] F. Cifaldi. 2006. Analysts: FPS 'Most Attractive' Genre for Publishers. Retrieved from http://www.gamasutra.com/php-bin/news_index.php?story=8241

[12] A.M. Connor, T.J. Greig, and J. Kruse. 2017. Evaluating the Impact of Procedurally Generated Content on Game Immersion. *Comput. Games J.* 6, 4 (2017), 209–225.

[13] A.M. Connor, T.J. Grieg, and J. Kruse. 2018. Evolutionary generation of game levels". *EAI Trans. Creat. Technol.* 5, 15 (2018), e4.

[14] Michael Cook, Simon Colton, and Jeremy Gow. 2017. The ANGELINA Videogame Design System—Part II. *IEEE Trans. Comput. Intell. AI Games* 9, 3 (2017), 254–266.

[15] Rui Craveirinha, Nuno Barreto, and Licinio Roque. 2016. Towards a Taxonomy for the Clarification of PCG Actors' Roles. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, ACM, Austin, Texas, USA, 244–253.

[16] Daniel Michelon De Carli, Fernando Bevilacqua, Cesar Tadeu Pozzer, and Marcos Cordeiro d'Ornellas. 2011. A Survey of Procedural Content Generation Techniques Suitable to Game Development. In *2011 Brazilian Symposium on Games and Digital Entertainment*, IEEE Computer Society, Salvador, Brazil, 26–35.

[17] Alena Denisova and Paul Cairns. 2015. The placebo effect in digital games: Phantom perception of adaptive artificial intelligence. In *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*, ACM, London, UK, 23–33.

[18] Adrian Diaz-Furlong Hector and Luisa Solis-Gonzalez Cosio Ana. 2013. An approach to level design using procedural content generation and difficulty curves. In *2013 Conference on Computational Intelligence in Games*, IEEE, Niagara Falls, Ontario, Canada, 1–8.

[19] Andrew T Duchowski. 2017. *Eye tracking methodology: Theory and practice*. Springer, London, UK.

[20] Philip Galanter. 2012. Computational aesthetic evaluation: Past and future. In *Computers and Creativity*, J. McCormack and M. d'Inverno (eds.). Springer, Berlin, 255–293.

[21] Riccardo Galdieri, Alessandro Longobardi, Michele De Bonis, and Marcello Carrozzino. 2021. Users' Evaluation of Procedurally Generated Game Levels. In *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, Springer, Online Conference, 44–52.

[22] Edoardo Giacomello, Pier Luca Lanzi, and Daniele Loiacono. 2018. Doom level generation using generative adversarial networks. In *2018 Games, Entertainment, Media Conference (GEM)*, IEEE, Galway, Ireland, 316–323.

[23] Joseph H Goldberg and Xerxes P Kotval. 1999. Computer interface evaluation using eye movements: methods and constructs. *Int. J. Ind. Ergon.* 24, 6 (1999), 631–645.

[24] Matthew Guzdial and Mark Riedl. 2018. Automated game design via conceptual expansion. In *Proceedings of the 14th Conference on Artificial Intelligence and Interactive Digital Entertainment*, AAAI, Edmonton, Alberta, Canada, 31–37.

[25] Erin Jonathan Hastings, Ratan K Guha, and Kenneth O Stanley. 2009. Automatic content generation in the galactic arms race video game. *IEEE Trans. Comput. Intell. AI Games* 1, 4 (2009), 245–263.

[26] Monique M Hennink and Bonnie N Kaiser. 2020. Saturation in qualitative research. In *SAGE Research Methods Foundations*, Paul Atkinson, Sara Delamont, Alexandru Cernat, Joseph W. Sakshaug and Richard A. Williams (eds.). SAGE Publications Limited, Thousand Oaks, California, United States. Retrieved from https://methods.sagepub.com/foundations/saturation-in-qualitative-research

[27] Morten Hertzum. 2010. Images of usability. *Intl J. Human–Computer Interact.* 26, 6 (2010), 567–600.

[28] K. Hullett and J. Whitehead. 2010. Design patterns in FPS levels. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, ACM, Monterey, California, 78–85.

[29] Aki Järvinen. 2009. *Games without frontiers: Methods for game studies and design*. VDM Verlag, Saarbrücken, Germany.

[30] Ming Jiang and Li Zhang. 2021. An Interactive Evolution Strategy based Deep Convolutional Generative Adversarial Network for 2D Video Game Level Procedural Content Generation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, IEEE, Online Conference, 1–6.

[31] Anna Kantosalo and Hannu Toivonen. 2016. Modes for creative human-computer collaboration: Alternating and task-divided co-creativity. In *Proceedings of the 7th International Conference on Computational Creativity*, Association for Computational Creativity, Paris, France, 77–84.

[32] Manolya Kavakli and John S Gero. 2002. The structure of concurrent cognitive actions: a case study on novice and expert designers. *Des. Stud.* 23, 1 (2002), 25–40.

[33] Ahmed Khalifa, Philip Bontrager, Sam Earle, and Julian Togelius. 2020. Pcgrl: Procedural content generation via reinforcement learning. In *Proceedings of the 16th Conference on Artificial Intelligence and Interactive Digital Entertainment*, AAAI, Online Conference, 95–101.

[34] Mohammad T Khasawneh, Shannon R Bowling, Xiaochun Jiang, Anand K Gramopadhye, and Brian J Melloy. 2003. A model for predicting human trust in automated systems. In *Proceedings of the 8th Annual International Conference on Industrial Engineering – Theory, Applications and Practice*, Citeseer, Las Vegas, Nevada, USA, 216–222.

[35] A. Kosorukoff. 2001. Human based genetic algorithm. In *2001 International Conference on Systems, Man, and Cybernetics*, IEEE, Tucson, Arizona, USA, 3464–3469.

[36] J. Kruse and A.M. Connor. 2015. Multi-agent evolutionary systems for the generation of complex virtual worlds. *EAI Endorsed Trans. Creat. Technol.* 2, 5 (2015), e5.

[37] J. Kruse, A.M. Connor, and S. Marks. 2021. An Interactive Multi-Agent System for Game Design. *Comput. Games J.* 10, (2021), 41–63.

[38] J. Kruse, R. Sosa, and A. M. Connor. 2016. Procedural urban environments for FPS games. In *Proceedings of the Australasian Computer Science Week Multiconference*, ACM, Canberra, Australia, 1–5.

[39] Jan Kruse. 2014. Interactive evolutionary computation in design applications for virtual worlds. Thesis. Auckland University of Technology. Retrieved June 25, 2015 from http://aut.researchgateway.ac.nz/handle/10292/8593

[40] P.L. Lanzi, D. Lomacono, and R. Stucchi. 2014. Evolving maps for match balancing in first person shooters. In *2014 Conference Computational Intelligence and Games*, IEEE, Dortmund, Germany, 1–8.

[41] Antonios Liapis, Gillian Smith, and Noor Shaker. 2016. Mixed-initiative content creation. In *Procedural Content Generation in Games*, Noor haker, Togelius and Mark J. Nelson (eds.). Springer, Cham, Switzerland, 195–214.

[42] Antonios Liapis, Georgios N Yannakakis, Constantine Alexopoulos, and Phil Lopes. 2016. Can computers foster human users' creativity? Theory and praxis of mixed-initiative co-creativity. *Digit. Cult. Educ.* 8, 2 (2016), 136–153.

[43] Antonios Liapis, Georgios N Yannakakis, and Julian Togelius. 2013. Sentient sketchbook: computer-assisted game level authoring. In *Proceedings of the 8th International Conference on the Foundations of Digital Games*, ACM, Chania, Crete, Greece, 213–220.

[44] Antonios Liapis, Georgios Yannakakis, and Julian Togelius. 2013. Towards a generic method of evaluating game levels. In *Proceedings of the 9th Conference on Artificial Intelligence and Interactive Digital Entertainment*, AAAI, Boston, Massachusetts, USA, 30–36.

[45] Ken Museth. 2014. Hierarchical digital differential analyzer for efficient ray-marching in OpenVDB. In *ACM SIGGRAPH 2014 Talks*. ACM, Vancouver, British Coloumbia, Canada, 1–1.

[46] P. T. Ølsted, B. Ma, and S. Risi. 2015. Interactive evolution of levels for a competitive multiplayer FPS. In *2015 Congress on Evolutionary Computation*, IEEE, Sendai, Japan, 1527–1534.

[47] Lisa M Osbeck and Stephen L Antczak. 2021. Generalizability and qualitative research: A new look at an ongoing controversy. *Qual. Psychol.* 8, 1 (2021), 62.

[48] Denis Pallez, Philippe Collard, Thierry Baccino, and Laurent Dumercy. 2007. Eye-Tracking Evolutionary Algorithm to minimize user fatigue in IEC applied to Interactive One-Max problem. In *Proceedings of the 9th Annual Conference Companion on Genetic and Evolutionary Computation*, ACM, London, UK, 2883–2886.

[49] Yan Pei and Hideyuki Takagi. 2018. Research progress survey on interactive evolutionary computation. *J. Ambient Intell. Humaniz. Comput.* (2018), 1–14.

[50] Michele Pirovano, Renato Mainetti, and Daniele Loiacono. 2015. Volcano: An interactive sword generator. In *2015 Games Entertainment Media Conference*, IEEE, Toronto, Ontario, Canada, 1–8.

[51] Alexandru Predescu and Mariana Mocanu. 2020. A data driven survey of video games. In *12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, IEEE, Bucharest, Romania, 1–6.

[52] Willian MP Reis, Levi HS Lelis, and others. 2015. Human computation for procedural content generation in platform games. In *2015 Conference on Computational Intelligence and Games*, IEEE, Tainan, Taiwan, 99–106.

[53] Luiz Rodrigues and Jacques Brancher. 2019. Playing an educational game featuring procedural content generation: Which attributes impact players' curiosity? *RENOTE* 17, 1 (2019), 254–263.

[54] Johnny Saldaña. 2016. *The coding manual for qualitative researchers* (2nd ed.). Sage.

[55] Jacob Schrum, Jake Gutierrez, Vanessa Volz, Jialin Liu, Simon Lucas, and Sebastian Risi. 2020. Interactive evolution and exploration within latent level-design space of generative adversarial networks. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, ACM, Online Conference, 148–156.

[56] Stefan Seidel, Nicholas Berente, Aron Lindberg, Kalle Lyytinen, and Jeffrey V Nickerson. 2018. Autonomous tools and design: a triple-loop approach to human-machine learning. *Commun. ACM* 62, 1 (2018), 50–57.

[57] Noor Shaker, Gillian Smith, and Georgios N Yannakakis. 2016. Evaluating content generators. In *Procedural Content Generation in Games*, Noor Shaker, Julian Togelius and Mark J Nelson (eds.). Springer, Cham, Switzerland, 215–224.

[58] Tanya Short and Tarn Adams. 2017. *Procedural generation in game design*. CRC Press, Boca Ration, Florida, USA.

[59] Mike Stout. 2016. A Beginner's Guide to Designing Video Game Levels. Retrieved from https://gamedevelopment.tutsplus.com/tutorials/a-beginners-guide-to-designing-video-game-levels--cms-25662

[60] Adam Summerville, Julian RH Mariño, Sam Snodgrass, Santiago Ontañón, and Levi HS Lelis. 2017. Understanding Mario: An evaluation of design metrics for platformers. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*, ACM, Hyannis, Massachusetts, USA, 1–10.

[61] H. Takagi. 2001. Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. *Proc. IEEE* 89, 9 (2001), 1275–1296.

[62] J. Togelius, Mike Preuss, Nicola Beume, Simon Wessing, Johan Hagelbäck, Georgios N Yannakakis, and Corrado Grappiolo. 2013. Controllable procedural map generation via multiobjective evolution. *Genet. Program. Evolvable Mach.* 14, 2 (2013), 245–277.

[63] Julian Togelius, Renzo De Nardi, and Simon M Lucas. 2006. Making racing fun through player modeling and track evolution. In *Proceedings of the 2006 Workshop on Adaptive Approaches for Optimizing Player Satisfaction in Computer and Physical Games*, SAB, Rome, Italy.

[64] Julian Togelius, Emil Kastbjerg, David Schedl, and Georgios N. Yannakakis. 2011. What is procedural content generation? Mario on the borderline. In *Proceedings of the 2nd International Workshop on Procedural Content Generation in Games*, ACM, New York, NY, USA, 1–6.

[65] Eric J Toy, Jaya VHH Kummaragunta, and Jin Soung Yoo. 2018. Large-scale cross-country analysis of steam popularity. In *2018 International Conference on Computational Science and Computational Intelligence*, IEEE, Las Vegas, Nevada, USA, 1054–1058.

[66] Vygotskiĭ. 2012. *Thought and language*. MIT Press, Massachusetts, PA, USA.

[67] Georgios N Yannakakis and J. Togelius. 2011. Experience-driven procedural content generation. *IEEE Trans. Affect. Comput.* 2, 3 (2011), 147–161.

[68] Jichen Zhu, Antonios Liapis, Sebastian Risi, Rafael Bidarra, and G Michael Youngblood. 2018. Explainable AI for designers: A human-centered perspective on mixed-initiative co-creation. In *2018 Conference on Computational Intelligence and Games*, IEEE, Maastricht, the Netherlands, 1–8.