INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Tips for Tips - A Methodical Approach Towards Better Hint Systems

André Filipe Gonçalves Silva

Masters in Computer Engineering

Supervisor:
Dr. Pedro Santana, Assistant Professor
Iscte - University Institute of Lisbon

November, 2021

TECHNOLOGY
AND ARCHITECTURE

Tips for Tips - A Methodical Approach Towards Better Hint Systems

André Filipe Gonçalves Silva

Masters in Computer Engineering

Supervisor:
Dr. Pedro Santana, Assistant Professor
Iscte - University Institute of Lisbon

November, 2021

"Remember that right here in this moment is all you are guaranteed, and the fact that you are living is what life is all about. So live your life to the fullest, according to your happiness and the betterment of all"

— Sir Robert Bryson Hall II

Acknowledgment

First and foremost I would like to thank my girlfriend, Mariana, for all the help with playtesting, finding bugs, and proofreading this document. Your help was invaluable! I hope in the future I will be able to help you with your Data Science degree even more than you helped me with this dissertation. I also want to thank my parents, Mário and Lurdes, my brother, Mário Sérgio, my uncle and aunt, Fernando and Luísa for their exceptionally strong support throughout the development of this dissertation, and for constantly teaching me all the things I need to succeed in life. Thank you Pedro Santana, my supervisor, for your patience, hours of discussing new and exciting games, acceptance of my strange schedules and constant tiredness from working and studying. Thank you José Cavalheiro, my manager, for helping me learn Python, which was invaluable to perform statistical analysis for this dissertation. For your constant availability to discuss best programming practices, evangelizing me on reading technical books, and understanding the difficulties of working and studying in parallel. Finally, thank you to the playtesters who took over an hour of their day to participate in the experiment. Without your participation this would not have been possible.

Resumo

Estudos existentes em assistência a jogadores de videojogos focam-se em qual a melhor forma de fazer o design de dicas e qual o seu potencial impacto na experiência dos jogadores. No entanto, tanto quanto temos conhecimento, não existe investigação sobre o impacto de diferentes tipos de sistemas de seleção de dicas na experiência dos jogadores. Esta dissertação propõe uma framework conceptual para o design e comparação de sistemas de dicas como uma peça fundamental do design centrado no utilizador. Esta framework conceptual foi aplicada ao desenvolvimento de ”Island”, um jogo de exploração em primeira pessoa, baseado em conhecimento, juntamente com duas abordagens de seleção de dicas: seleção aleatória ou uma ordenação por nível de quão ocultas estas são. Foram realizados testes com participantes e, posteriormente, análises estatísticas que revelaram existir uma relação entre o sistema de seleção de dicas e parâmetros do jogo, tais como a percentagem de vitórias, que validam a nossa framework conceptual. Estes resultados podem ter um impacto significativo no estudo da adaptação da experiência do jogador através da forma como a seleção de dicas é realizada, não só para jogos para divertimento, mas também para jogos sérios e educativos.

Palavras-chave: Videojogos; Systemas de Dicas; Jogos Afetivos; Interação Pessoa-Computador.

Abstract

Previous research on video game player assistance has focused on how to best design hints and their potential impacts on player experience; however, to the best of our knowledge, no research has been done on the impact of different hint selection systems on player experience. This dissertation proposes a conceptual framework for designing and comparing hint systems as a core piece of user centered design. This conceptual framework was applied to the development of ”Island”, a first person knowledge-based exploration game, along with two approaches for hint selection: randomly selecting hints or ordering hints by their covertness. Playtests were performed with 20 different participants and a statistical analysis was performed which showed that there is a relationship between the hint selection system and gameplay features such as win rate, which validates our conceptual framework. We believe our findings have the potential to impact the study of player experience adaptation through hint selection, not only for games for fun, but also serious and educational games.

Keywords: Videogames; Hint Systems; Affective Gaming; Human-Computer Interaction.

Contents

Acknowledgment i

Resumo iii

Abstract v

List of Figures ix

List of Tables xi

Chapter 1. Introduction 1

1.1. Context 1

1.2. Motivation 1

1.3. Research Questions 3

1.4. Objectives 4

1.5. Research Methodology 4

Chapter 2. Literature Review 7

2.1. Game Design 7

2.2. Affective Gaming 8

2.3. Flow 10

2.4. Dynamic Difficulty Adjustment 12

2.5. Hint Systems 15

2.6. Measuring Player Experience 18

2.7. Contribution 20

Chapter 3. Conceptual Framework 21

3.1. Player Influence Vectors 21

3.2. Workflow 26

Chapter 4. Game Design and Development 29
4.1. The Game 29
4.2. Development 40
Chapter 5. Evaluation 47
5.1. Playtests 47
5.2. Demographics 49
5.3. Gameplay Features 50
5.4. Game Experience Questionnaire 54
5.5. Informal Post-Game Discussions 56
5.6. Results Discussion 57
Chapter 6. Conclusions and Future Work 59
6.1. Conclusions 59
6.2. Limitations and Future Work 60
References 63
Appendices 71
Appendix A. Experience Questionnaire 71

List of Figures

1 The map of the Outer Wilds. 2

2 An interest curve for an entertainment experience. 8

3 A typical representation of the affective loop. 9

4 The flow channel model. 11

5 Experience Fluctuation Model. 12

6 Arroyo *et al*.'s classification of hints into four categories. 17

7 Our proposed bisection of the Experience Fluctuation Model. 22

8 Our proposed bisection of the Experience Fluctuation Model showing the areas of application of DDA algorithms and hint systems. 23

9 Player influence vectors based on task-relatedness and solution/logic-relatedness. 24

10 Goal vector calculated from an estimation of the current player emotion and our target emotion. 25

11 Firefly hint in low fidelity compared to high fidelity. 26

12 A screenshot of the game with no UI elements overlaid. 30

13 Screenshots of the score gauge. 31

14 Objective Selection flowchart. 32

15 An overview of the map for ”Island”. 33

16 Footprints hint when searching for an Axe. 35

17 Fireflies hint when searching for a flashlight. 36

18 Riddle hint when searching for a Backpack. 36

19 Photo hint when searching for a knife. 37

20 Player influence vectors for the hints in ”Island”. 38

21 A pie chart of game engines used for 5400 of games made for the 2020 GMTK Game Jam [49]. Unreal Engine falls into the category ”Other”. 41

22 A components diagram for ”Island” 43

23 Histogram of participants’ average weekly time playing videogames in the past three months. 50

24 Percentage of long item objectives for each item when using a random hint selection (RH) and non-random hint selection (NRH). 51

25 Time spent idle and moving for random hint selection and non-random hint selection. 53

List of Tables

1 Flow manifestation in games. 13

2 Game objectives and corresponding spawn logic. 32

3 Game objectives and corresponding riddles. 38

4 Comparison between game engines. 41

5 Participant self-reported experience level and types of hint selection used in playtesting 49

6 Average rounds played per hint selection type. 50

7 Percentage of long item objectives for each item (Mean ± Std. Dev.). 52

8 Game Experience Questionnaire Core Module Results for N = 10 (Mean ± Std. Dev.). 55

9 Game Experience Questionnaire Post-Game Module Results for N = 10 (Mean ± Std. Dev.). 55

10 Game Experience Questionnaire Core Module Wilcoxon rank sum test. 56

11 Game Experience Questionnaire Post-Game Module Wilcoxon rank sum test. 56

CHAPTER 1

Introduction

1.1. Context

Videogames present an interesting case of interactive media because they influence players' susceptibility to new ideas [1] while keeping them in control of their own actions. This characteristic of videogames means they are a powerful tool to try and increase the player's knowledge while simultaneously letting the player have fun.

Previous research has shown that games that give the player freedom to experiment could be more effective at teaching than those that restrict freedom to focus on specific points through a tutorial [2]; however, more research is needed on how to facilitate this experimentation while making sure players effectively learn a game's mechanics [2]. Exploration games, especially those in which players have full control over where and when to act, are particularly interesting because of the increased feeling of being in control.

Since all players are different, some might need more time to understand and learn a certain mechanic. Others might have considerable of experience with a game genre and need no guidance at all. A game that gives a large amount of guidance might cause players to feel bored, for example, when a tutorial is too long. In contrast, players might feel lost, if there is not enough guidance. To illustrate these points, we looked at two games that tackle the teaching of mechanics in contrasting ways.

1.2. Motivation

The videogame industry has tackled the problem of teaching a game's mechanics to players in many different approaches. This is the case for both entertaining and educational games.

Valve's Half-Life 2 [3] is a First Person Shooter (FPS) that utilizes subtle design elements to teach the game's mechanics without use of traditional tutorial cues. An example of this is in the game's famous Ravenholm scene. Upon entering a house in Ravenholm, the player is presented with a dead zombie sliced in half with a saw blade. To progress, players must remove the saw blades that are stuck in a door frame, blocking a doorway. Upon removing a saw blade, the game is designed in such a way that the player will naturally be looking at another door, from which an alive zombie will appear. Since a saw blade was picked up to go through the door, players will already be equipped with a saw blade and instinctively shoot it at the zombie, learning in the process what the most effective weapon against them is.

A game that approaches teaching players in a completely opposite manner is Outer Wilds [4], an exploration game set in space which rewards player curiosity. The game only moves forward if the players choose to explore the different available areas, many of which will provide no reward other than learning more about the universe. The solar system where players spawn is organized in such a way that each of the planets, or other points of interest, teaches a different core mechanic. The many different points of interest can be seen in the map, presented in Figure 1, which is from a poster on the players' ships. Since the players are completely free to choose their path, they can choose to skip the starting area in its entirety, and unwittingly progress towards puzzles they do not yet have the skills or knowledge to solve.

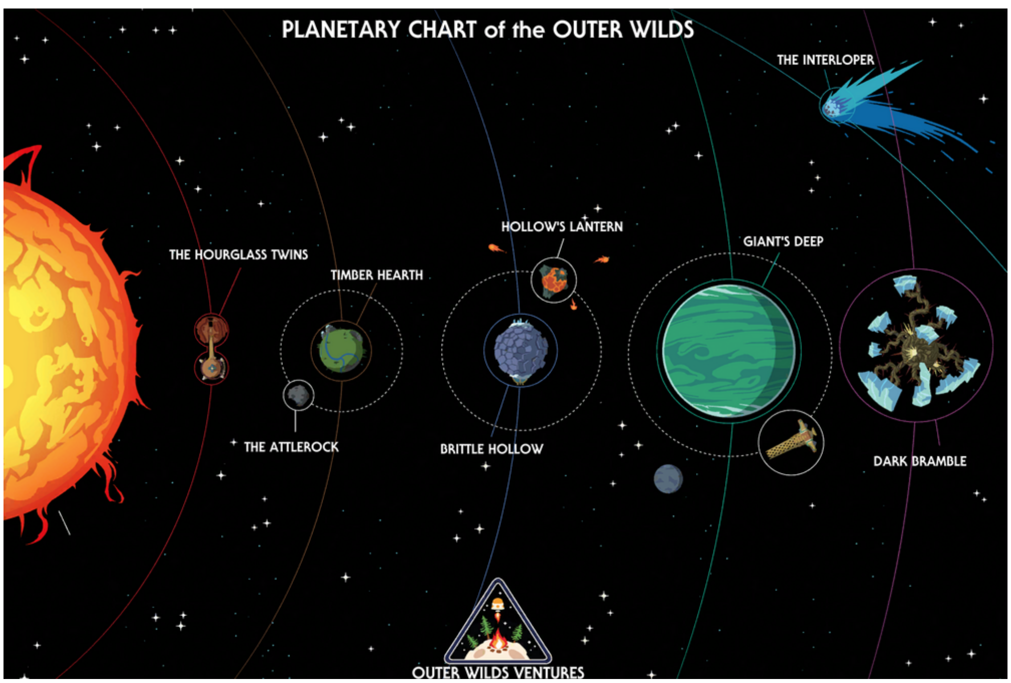

Figure 1. The map of the Outer Wilds [5]. Players are allowed to freely explore the solar system, and can unwittingly skip the starting area and go towards an area they do not yet have the knowledge to explore.

In Half-Life 2 and Outer Wilds, we see two distinct ways of handling the task of teaching players new mechanics. Half-Life 2 creates situations where players learn the games mechanics intuitively and passively, through a type of scaffolding that supports the player's learning, whereas Outer Wilds allows the players to make mistakes and skip content entirely, while adding much more freedom to explore. Although both systems have advantages, we can pinpoint two clear problems. To create situations like the one described above in Half-Life, a game designer will most likely have to iterate on the scene multiple times in order to make each scene work well as a teaching tool and, conversely, these scenes require much more development time compared to a classic tutorial. Outer Wilds, in contrast, does not provide players any mandatory information and makes them depend on their curiosity for guidance. Since each player will be different, the game might immediately push away those who prefer a more explicit tutorial.

If Outer Wilds could adapt its content in such a way that players would be provided hints before they got frustrated due to being stuck by lacking some critical piece of information, we could potentially have an experience closer to that of Half-Life's carefully designed tutorials, with smaller development time. For this purpose, designing a hint system methodically and iteratively and as a central piece of user-centered design should be advantageous, not only due to the potential improvements to player experience for games for fun, but also in serious and educational games.

1.3. Research Questions

From the points presented in Section 1.2, it is clear that research in hint selection systems is novel in that they are not widely used in the video game industry. This dissertation aims to answer four research questions:

(1) Does designing hint systems in an iterative and methodical fashion help in finding solutions that give game designers control game parameters?
(2) Does designing hint systems in an iterative and methodical fashion help in finding solutions that improve player experience?
(3) Does altering the hint selection strategy impact game parameters?
(4) Does altering the hint selection strategy impact player experience?

To try and answer these questions, we herein propose a conceptual framework to guide this dissertation, along with future research regarding hint selection systems. We applied the framework to the implementation of a game with two different iterations of a hint system, and compared these, to mimic how a game developer could potentially iterate on their hint system design. A first iteration selects hints randomly from a pool of possible hints, whereas a potential improvement that a game developer could make would be to provide hints ordered from the most covert (hidden) to overt (visible). These two systems, and as such, the iterative process proposed by our conceptual framework, were tested with people with video game experience, which will help in answering our research questions.

1.4. Objectives

One of the core goals behind this dissertation is to analyze if it is advantageous to design hint systems methodically and iteratively. A second goal is to explore the effects of differing hint selection systems on gameplay features and player experience. Another goal is the design of a conceptual framework which encompasses the iterative process inherent to game design, and puts hints and hint selection systems as a core piece of user-centered design. Finally, it is also our goal to apply the framework to the development of a game which serves as a benchmark for future studies, along with validation with testers and the analysis of results to understand if there is a statistical significance between each of the hint systems, through player self-reported experience questionnaires and data extracted from the playtesters gameplay.

The results from the application of this conceptual framework to a game are an example of how the application of an iterative and methodical workflow is beneficial in comparing different hint systems, and the developed game serves as a baseline for future studies to compare results to.

1.5. Research Methodology

The development of this project will follow the Design Science Research Methodology (DSR). DSR defines a process for conducting research in information systems [6]. This process is split into five steps:

(1) Problem Awareness;
(2) Solution Suggestion;
(3) Development;
(4) Evaluation;
(5) Conclusion.

Our dissertation presents the application of each of these items. In Chapter 1, the problem and our motivation was presented, corresponding to Step (1), a literature review was also conducted and is presented in Chapter 2. Our proposed solution is presented in Chapters 3 and 4, which correspond to Step (2). Section 4.2 presents the created game and tests which corresponds to the Development step of the DSR, or Step (3). Results and corresponding evaluation are presented in Chapter 5, which fulfills the Evaluation step, or Step (4). Finally, Chapter 6 contains conclusions regarding the executed research, along with suggested improvements and future work, fulfilling the DSR methodology's final step, Step (5).

CHAPTER 2

Literature Review

To better understand the various components required to develop hint systems, and analyze and compare them, a review of existing research on related topics must be performed. To start, research on game design was conducted.

2.1. Game Design

The purpose of a game is to create a memorable and impactful experience upon a player. This can have many different appearances for different players, but like with every discipline, it is important to look at related areas for guidance on how to approach a problem. First, however, we must define what it means to design a game.

2.1.1. Definitions

It is important to define a game, and the act of game design. Costikyan [7] defines a game as ”an interactive structure of endogenous meaning that requires players to struggle toward a goal”. This definition states that the player must interact with the game in order for it to be a game. Additionally, there is some form of goal that the player must progress towards through hardship, meaning it must not be a clear path towards said goal and it must instead be challenging. Finally, a game’s content must only have some value within the game, which is to say, anything that is a part of the game should serve no purpose other than the one it serves inside the game. Score should only serve as a measure of progress in the game, and not for some real world value, which is the case in casinos, whose chips are a representation of real world currency [8].

Schell [8] defines game design as the design of an experience. An experience is how the players perceive the game, and is a completely subjective and ultimately unsharable perception. Although we can use different means to predict how most players will feel while playing a game, in the end an experience is unique to the players and can never be fully recounted. One of the methods to potentially improve the impact of a game is by thinking about the game’s interest curves.

2.1.2. Interest Curves

Schell [8] defined Interest Curves as a series of moments which have different intensities, and as such can be charted on a graph as a curve. Figure 2 shows an example of an interest curve.

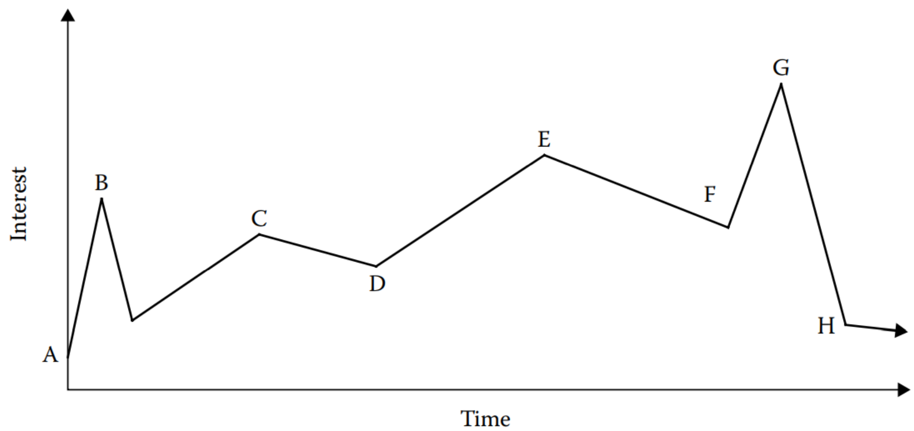

Figure 2. An interest curve for an entertainment experience. Extracted from [8].

A game is composed of moments with different levels of intensities. There might be moments of high intensity where the players are almost overwhelmed by the number of enemies on screen but manage to survive, and other moments where a portion of the story is told by a character, and the players can simply listen. If a game was full of only high intensity moments, players would quickly get exhausted, and a climax would not be felt as one, whereas if a game only contained slow moments, with story elements which in and of themselves are also of low intensity, then the players could get bored.

Kim *et al*. [9] designed levels for ”Super Mario Maker 2” with and without interest curves and showed that player immersion and satisfaction increased when playing on levels designed with the application of interest curves. Ouellette *et al*. [10] presented a process of iterative analysis of educational game interest curves, which was shown to increase player enjoyment and skill retention. Another way to potentially increase player enjoyment is through affective gaming.

2.2. Affective Gaming

Typically, a game is played by a player interacting with their system, through an interface, such as a mouse and keyboard or a controller. The game can be the same for every player, or it can even be procedurally generated [11], where the maps, items or even story change with each iteration. However, there are few games that change their content by detecting the player's affective state, the feeling, emotion or mood that the player might be experiencing at any given time.

Affective Gaming's [12] main goal is to fulfil the affective loop. The affective loop consists of four stages. The player of a videogame expresses some emotion, which a system detects. Then, the game is adapted to guide the player towards the desired emotion. This adaptation will have some impact on the player experience which in turn impacts player emotion. This loop can be seen in Figure 3.

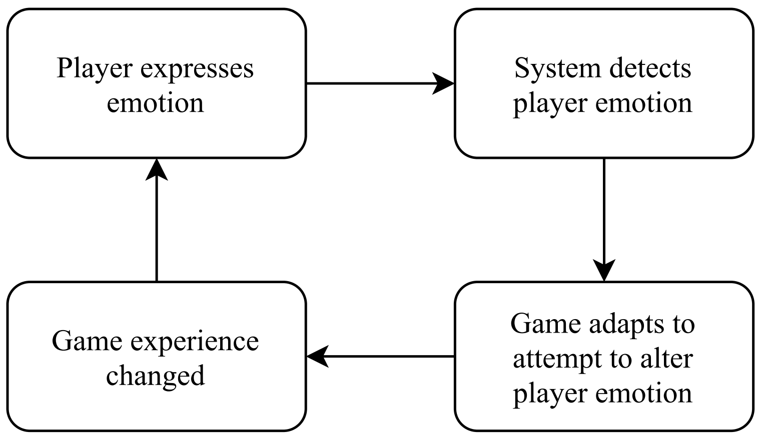

Figure 3. A typical representation of the affective loop. Adapted from [13].

There have been multiple attempts at measuring the affective state of a player. Sykes and Brown [14] found that they could determine a player's arousal by the pressure with which they pressed buttons on a gamepad. Martínez *et al*. [15], Drachen *et al*. [16] and Darragh *et al*. [17] all utilized heart rate sensors and electrodermal activity monitors to predict player affective states across games of different genres and with different game mechanics. Zheng *et al*. [18] were able to predict player's emotion through pupil dilation levels. Pedersen *et al*. [19] were able to match gameplay features, such as time spend idling, to specific player's emotions.

Dekker and Champion [20] successfully improved player experience for those who like to play horror games. Liu *et al*. [21] adapted the difficulty of a game dynamically by measuring player affect through physiological features. Yannakakis *et al*. [22] showed that adapting camera viewpoint based on biological features impacted the psychophysiological state of the player.

A common goal found throughout research papers when talking about affective games is that of creating an optimal experience, often referred to as being in a state of Flow.

2.3. Flow

When an individual partakes in an activity for purely intrinsic purposes, meaning there are no external influences such as monetary rewards or threat of punishment, and only internal influences, such as playing a game for the experience, there is a chance of the individual entering a state of flow.

Flow is a concept first introduced by Csíkszentmihályi [23] in 1990. It is a state of deep concentration on the task at hand, and was found to be isomorphically present in people from different cultures, genders, and backgrounds.

A set of prerequisites must be met in order to feel flow, perhaps most interestingly, the match between the individuals perception of the challenge of the task they are undertaking, and the perception they have of their own skills in regards to completing said challenge. Figure 4 shows the flow channel model initially proposed by Csíkszentmihályi. When an individual's skills match with their perception of the difficulty of a task, they can enter the flow channel, and experience the state of flow. Conversely, if the difference between the perception of the challenge of the task, and of the individual's own skill level regarding the task is of a significant magnitude, flow cannot be reached.

A disparity was found in the existing flow model. When individuals encountered a task whose challenge and skills were low, they did not experience flow, but instead entered a state of apathy. Massimini and Carli [24] proposed a new model that divides the plane into eight parts, which can be seen in Figure 5. This model features eight different affective states: worry, apathy, boredom, relaxation, control, anxiety, arousal, and flow.

The experience fluctuation model [24] presents an interesting parallel with videogames, which inherently require focus, and advance in complexity as familiarity and proficiency increase [25]. A videogame is usually designed in such a way that the players will learn enough to complete the challenges that are proposed to them, and the challenges keep increasing in difficulty. This keeps a symmetry between the player's perception of their skills, and the challenge of the tasks proposed.

To reach flow, however, there are more requirements than simply having a balance between challenges and skills. Csíkszentmihályi [23] stated that those that have experienced flow mentioned

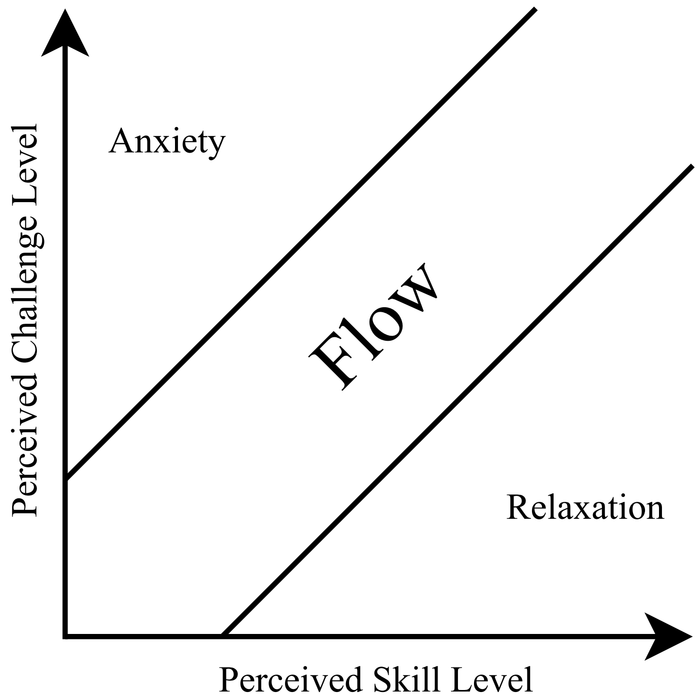

Figure 4. The flow channel model, adapted from [23]. When an individual's perceived skill level is above the challenge they perceive for the task, they would be relaxed. In contrast, when the challenge is higher than the individual's perceived skill, they would feel anxiety. Finally, if there is a balance between skill and challenge, flow can be achieved.

at least one, or all of the following. A flow state was reached when individuals partook in a task they had a chance of completing. Secondly, the individuals were able to concentrate on the task. This concentration comes from the third and fourth element of flow, the task having clear goals and providing immediate feedback. The fifth element of flow is that individuals have a deep but effortless involvement on the task. Sixth, an enjoyable experience allows individuals to exercise a sense of control over their actions. Seventh, individuals stop being concerned with themselves, however, after the flow experience is over, the sense of self is much stronger than before. Finally, the eighth element of flow is that there is a change in the sense of duration of time.

Jones [26] presents a comprehensive table of the elements of flow and how they map to their manifestation in games. This mapping can be seen in Table 1. Jones also states that although the visual, musical and sound quality of the game might provide pleasure, these do not necessarily make a good game [26]. What is important is not that the visuals are of the highest quality, but that they

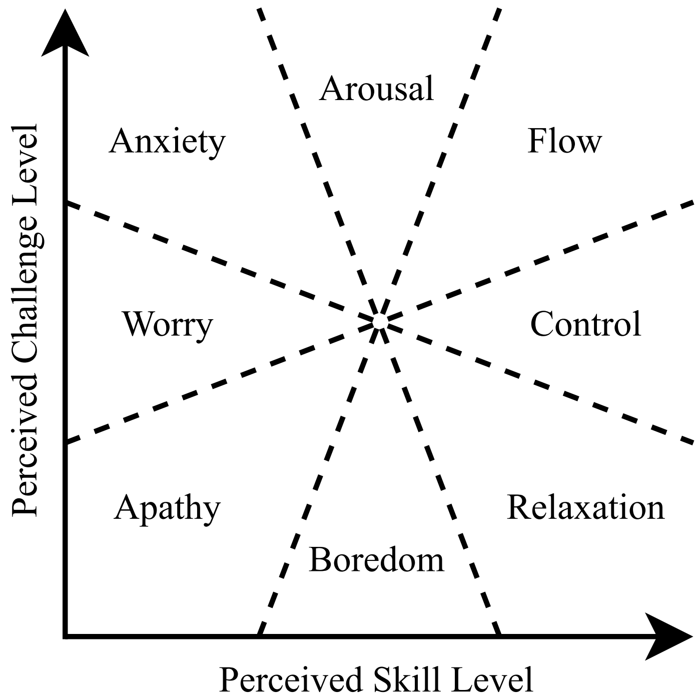

Figure 5. Experience Fluctuation Model, adapted from [24]. Depending on the individual's current perception of their skill, and of the challenge of the task they face, they could be in any of the eight sections of the graph, and would most likely be in the corresponding affective state.

are consistent with the remaining elements of the game. Additionally, deeper problems are much more likely to provide a feeling of flow than easy problems [26]. This means that managing the difficulty of the problem and how the player perceives it is essential to helping players reach flow. The following sections present how the problem of dynamically adjusting difficulty and providing hints to players is tackled in academia and in the industry.

2.4. Dynamic Difficulty Adjustment

Dynamic Difficulty Adjustment (DDA) adapts the difficulty of an interactive application to match to the user skills or to attempt to invoke an affective state on a user. Afergan *et al*. have shown that a DDA algorithm applied to UAV operators decreased failure rate by 35% [27], which showcases the applicability of DDA algorithms towards learning systems for training applications.

In the case of video games, some algorithms adapt the number of enemies [28], spawn rate [29] or game mechanics [30] to increase player success. Although player success increases, if the player

Table 1. Flow manifestation in games, adapted from [26].

Element of Flow	Example of manifestation in a game
1. Task that can be completed	The use of levels in games provides small sections that lead to the completion of the entire task.
2. Ability to concentrate on task	Creation of convincing worlds that draw users in. The dungeons and labyrinths in Doom II help suspend your belief systems for a time.
3. Task has clear goals	Survival, collection of points, gathering of objects and artefacts, solving the puzzle.
4. Task provides immediate feedback	Shoot people and they die. Find a clue, and you can put it in your bag.
5. Deep but effortless involvement (losing awareness of worry and frustration of everyday)	The creation of environments far removed from what we know to be real helps suspend belief systems and takes us away from the ordinary.
6. Exercising a sense of control over their actions	Mastering controls of the game, such as a mouse movement or keyboard combinations.
7. Concern for self disappears during flow, but sense of self is stronger after flow activity	Many games provide for an environment that is a simulation of life and death. One can cheat death and not really die. People stay up all night to play these games. It is the creation of an integration of representation, problem, and control over systems that promotes this.
8. Sense of duration of time is altered	Years can be played out in hours; battles can be conducted in minutes. The key point is that people stay up all night playing these games.

understands they are being helped by the system, they might feel that since they require assistance, their skill level is lower than they initially thought. This means there could be a potential negative correlation between a DDA algorithm and player perceived skills.

In the video games industry, there is a famous example of DDA applied to video games: The Director from Left 4 Dead. Left 4 Dead [31] and Left 4 Dead 2 [32] adapt not only the games difficulty, but the way the whole game loop is organized. The game is both singleplayer and multiplayer and pits four human or AI controlled characters against hordes of zombies. These zombies do not have defined spawn points, but instead The Director, a component in the game that uses a combination of rules and information regarding the game state, to place enemies and resources depending on the players current situation, status, skill, and location [33].

Generally, The Director spawns enemies out of sight of the players. There are four different enemy classifications, each with its own spawn rules. Wanderers, the most common zombie type, are usually spawned in front of the players, and move around until they come across a player character. In contrast, hordes of zombies, the second enemy type, will instead be spawned to the side or behind players. The final two enemy types are special zombies, which have their own individual spawn mechanics. They have assigned spawn zones, which are on the main path throughout the map and are static, meaning multiple playthroughs on the same map will have special zombies appear in the same positions. This presents a mix of carefully picked special zombie spawn locations, and the possibility of change between playthroughs caused by The Director.

While killing enemies, players increase their intensity value, a measure of how much pressure is currently on the player at that moment in the game. This pressure is what The Director uses to control game phases. The game is split into three main cyclical phases. Phase one is the build up phase, where the enemies are spawned regularly and player intensity increases as they kill, get injured or incapacitated. When the players reach full intensity, The Director enters phase two, Peak. In the Peak phase, enemies stop spawning, allowing the players to finish killing all enemies on the map. Finally, the third phase is the Relax phase. In this phase players have a limited time to use healing items, rest, reload and change weapon loadouts. If the players move more than a short distance from their position on the map when the relax phase started, this timer is shortened. After the relax phase is over, the game starts spawning enemies again, and enters a new build up phase, completing the cycle.

This three phased approach creates a dynamic roller coaster effect, where player arousal gradually increases, until a peak, where they are then given time to rest. This helps keep players on edge, while also giving them time to breathe and not get exhausted. This is an industry application of the Interest Curves discussed in Section 2.1.2, where player intensity can be mapped to a position on the interest curve. When the players are in a buildup phase, their intensity value in game increases until it reaches peak phase. Here, the intensity stops increasing, corresponding to the decrease in the interest curves. Finally, when the game is in a relax phase, the interest curve is at one of the local minima.

The Director also uses the players' locations to adapt the game. If players are killing enemies while being far from them, for example when using a long-ranged weapon, their associated intensity level increases much more gradually than when they are close, using for example melee attacks. Additionally, if a players wanders too far from their team, The Director guides the next attack onto the individual player, forcing the team to stay together or lose a member.

All of these methods put together create a dynamic game, that is influenced by not only individual player skill, but also through the team's skill as a group. This ensures the game is different between runs, as a player trying a different approach, such as going alone to find healing items, can greatly change the course of the playthrough.

Although dynamically adjusting the difficulty of a game can help players with not getting frustrated at the game, and subsequently giving them more time to learn the game's mechanics, another way to assist players when they are having a hard time with their game is through hints, which provide some information regarding how the player can complete their current task.

2.5. Hint Systems

The task of providing hints to players is not a simple one. Hints are often perceived as the game assuming the player needs help, which can impact player experience negatively.

Perhaps unintuitively, Schell [8] suggests that what matters most in a puzzle is the moment where players see the answer, even if they did not find it by themselves. This is a core concept behind providing hints for players. It is better if the player knows how to progress in a given situation, than if the player is stuck, gets frustrated, and decides to stop playing. This was also shown by Wang [34], which provided concrete and abstract hints to players and found that concrete hints are perceived as more helpful than abstract hints, and that abstract hints can even be perceived as worse than no hints at all. However, in Wang's study, players could request only one hint to solve a puzzle, which could be related as to why a more concrete hint would be perceived as being much more helpful. If players could get multiple hints, and if hints were provided, not requested, there could be a change in the perception regarding different hint types.

Arroyo *et al*. [35] classified hints based on their symbolism, similarly to Wang [34], and on their interactivity, as can be seen in Figure 6. A hint that is highly interactive is one where the user learns by doing. In contrast, low interactivity is when one learns by being told. A highly symbolic hint is one that does not make a connection with real life objects, whereas a low symbolic or concrete hint does. The researchers found that boys showed preference for non-interactive hints, in contrast with girls who preferred highly interactive hints. Additionally, the researchers found a link between symbolism and cognitive ability. Students with lower cognitive ability preferred hints that are less symbolic. Conversely, students with higher cognitive ability showed preference for highly symbolic hints. Wang [34] performed a part of their study with students from the University of Illinois, which skewed the results towards young participants with high cognitive ability, who showed a preference for more concrete hints. This shows a contrast with Arroyo *et al*.'s findings, where students with a higher cognitive ability preferred more symbolic hints.

Wauck and Fu [36] have shown that hints given to players with less gaming experience might suffer from a placebo effect, meaning that the act of providing a hint could be sufficient to increase player experience, even though the way the player acted in game did not change.

In the case of educational games, games with the purpose of teaching players a new skill, O'Rourke *et al*. [37] have shown that providing hints to players might have a negative impact on performance when compared to a no-hint situation. However, players that earned their hints showed more persistence, which indicates the way hint selection is performed can impact player experience.

Gilleade and Dix [38] propose three affective game design heuristics: assist me, challenge me, emote me. The assist me heuristic states that a videogame can measure player frustration through physiological signals and adapt before frustration levels get too high. The challenge me heuristic states that player engagement can be measured through arousal level, and this measurement can be used towards dynamically altering the difficulty of the game, to better suit the individual players. Finally, the emote me heuristic states that by measuring the current emotional state of the player, the game can then adapt to guide the player to its desired emotion.

These studies show evidence of hints impacting player experience, however, some studies seem to show contrasting results such as that of Wang [34] and Arroyo *et al*. [35]. This could be evidence of the need of further analysis regarding hints and hint selection systems, for which this dissertation

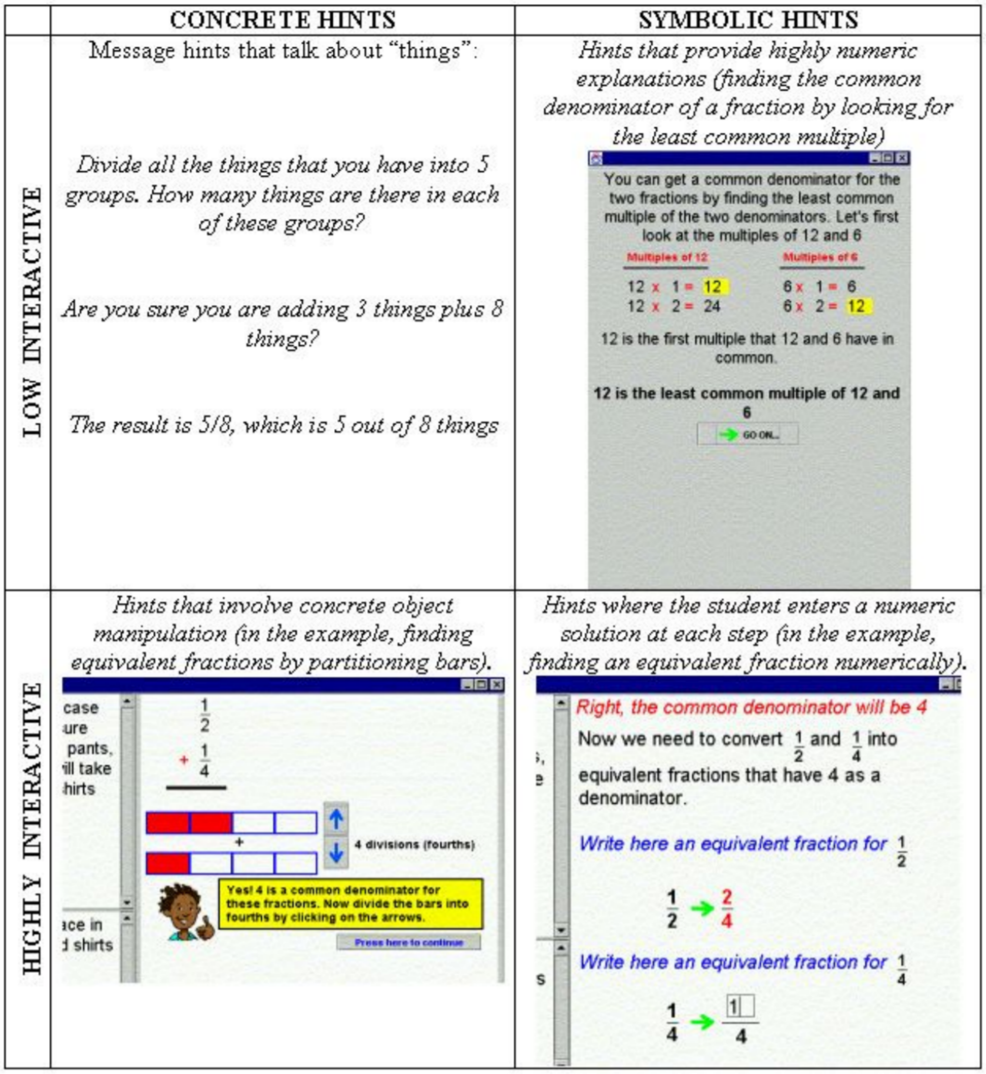

Figure 6. Arroyo *et al.*'s classification of hints into four categories [35].

contributes with a conceptual framework which will in with further research by providing common definitions and guidelines.

To know the impact of dynamically adjusting difficulty and providing hints to players we must have a way to measure player experience. For that it is useful to know characteristics of the target audience. Some characteristics and a questionnaire used for measuring player experience will be discussed in the following section.

2.6. Measuring Player Experience

Players come from many different cultures, have different ages, enjoy different activities, have different capabilities and, in general, have different backgrounds. With so many things that can differ between players, creating a game, a DDA algorithm, or even a hint system that can match all the players' expectations can seem like a herculean task. To combat the potential variability between player characteristics, we can attempt to classify players by grouping them based on their age and then we analyze an questionnaire widely used in academia.

2.6.1. Age ranges

One of the most common ways to group people is by their age. As individuals get older, their characteristics change. Younger players tend to have more free time, in contrast with older players; however, the latter have more purchasing capabilities, and, thus, more possibilities to try different experiences. Schell [8] suggests nine different groupings:

- 0-3, infants/toddlers: games are often too complex for them, and they show more interest in toys. Direct interfaces like touchscreens are fascinating, whereas abstract interfaces like a controller are too complex;
- 4-6, preschoolers: start showing interest in simple games, and are often played with parents;
- 7-9, kids: start making their own decisions on which games they want to play since they have entered school and are generally able to read and solve hard problems;
- 10-13, preteens: children start to get quite passionate about their games and are going through a period of intense neurological growth;
- 13-18, teens: very interested in experimenting with new activities. Some of those activities can involve games. There is a divergence between male and female interests;
- 18-24, young adults: the first "adult" age grouping. Adults, in general, play less than children do. They have established certain tastes and have both time and money to apply to games;
- 25-35, twenties and thirties: more adults start to form families and have less time to spend on games. Those whom playing games is their primary hobby purchase a lot of games and are quite vocal about what they do and do not like;

- 35-50, thirties and forties: most adults in this grouping are very busy and have much less time to spend playing games. They decide what to buy for their children and look for opportunities of games the family can play together;
- 50+, fifties and up: will soon retire and tend to have a lot of time on their hands since their children have moved out. Some return to games, but their physical capabilities have changed. Any games involving precise interactions can become frustrating as they might not see or be able to interact with games like they used to.

Schell suggests that the transitions between each age group are especially interesting as they often match with periods of mental or familial developments. All of the properties of each age group and transition period should be taken into account when designing hints and hint systems. For example, a hint system for a teenager is different than that of one for a working adult with children. The former might prefer a slower pacing on their hint orchestration, to give them more time to think and solve problems on their own when compared to the latter, which has much less free time, and as such might prefer more frequent, obvious hints.

To analyze player experience, it is useful to have a questionnaire that can be applied after playing through a game, and can be used to calculate different measures on player experience.

2.6.2. Experience Measurement Questionnaires

Most of the studies presented in Chapter 2 utilize their proposed methods as predictors of self-reported player affect. Players answer a questionnaire, which provides measurements for common affective states, such as positive affect, negative affect, frustration, flow, challenge, immersion and competence. Then a statistical analysis on the data is performed, comparing two or more different versions of their game, for example, with or without a particular affective adaptation. The most common questionnaire provided to players is the Game Experience Questionnaire (GEQ).

The Game Experience Questionnaire outputs seven different measures of player experience: Sensory and Imaginative Immersion, Flow, Competence, Positive Affect, Negative Affect, Tension and Challenge. Sensory and Imaginative Immersion is a measure of how connected the players felt with the game. Flow refers to the change of the players' perception of time. Competence is a measure of how the players felt they performed in achieving the goals set by the game. Positive and negative affect represent positive and negative emotions, respectively. Tension refers to the feelings of annoyance, frustration, or irritation. Finally, challenge represents the level to which the players found the game to be difficult.

The GEQ's questionnaire items were shown to have significant overlap between individual questions [39]. The challenge, tension and negative affect factors were also found to have significant overlap [39]. However, this study was performed with games which the players were familiar with, potentially skewing the results. For the purposes of our study, players will be playing a game they have not seen before, and so, these results are less relevant. Additionally, the GEQ is still the most widely used questionnaire of player experience, having been applied to compare player experience using different input methods [40], across differing genres [41], in virtual learning environments [42], to measure the impact of music and sound effects on video games [43] and to measure the influence of biofeedback in first person shooter games [44]. This places it as one of the best options to use, as it is widely utilized across research, and, as such, easily recognize and understood, and, consequently, is applied in this dissertation.

2.7. Contribution

From the review of existing literature, we found a gap in studies regarding the impact of differing hint selection systems on player experience and gameplay features. By implementing and selecting two different hint selection systems, one which follows a strategy of choosing more covert hints first, and one which does not have a strategy and chooses hints randomly from all possible hints, and applying statistical analysis to gather findings regarding if different systems impact player experiences and gameplay features, we are presenting an introductory study in regards to hint system impact on player experience and gameplay adaptation.

We also propose a conceptual framework which can help those who wish to perform further research in the fields of hints and hint selection systems, and those who wish to implement hint systems in videogames, by providing common language and guidelines. A game, and two different hint selection systems were developed following our conceptual framework and as such serve as validation of its applicability.

CHAPTER 3

Conceptual Framework

As was presented in Sections 2.4 and 2.5 of the Literature Review, there are two main pillars regarding assisting players when they are struggling to progress through a game: Dynamic Difficulty Adjustment and Hint Systems. Although the conceptual framework we propose embodies both pillars, the focus of this research will be on Hint Systems.

This conceptual framework places hints and hint systems as a piece of user-centered design. Hints must be though of as components that can impact the user, or in our case, player's experience, and must be present throughout the design process to ensure they are effective at guiding and teaching players. We look at hints in terms of being oriented to the solution of a given task, or towards the logic behind the task, and in terms of being related or unrelated to the task at hand. Each of the possible combinations of these terms defines a type of hint, and will have an associated vector which we consider as a guideline for how a hint of that type influences player experience.

3.1. Player Influence Vectors

If we consider the experience fluctuation model seen in Figure 5, we can see that it depends on the players perception of the challenge of the task at hand, and the players perception of their own skills. By bisecting the graph across a 45º angle, the graph is split into two sections: one where the players perceived challenge about a given task is superior to how the players see their own skill levels, and another where the players believe themselves to be more skillful than the task at hand. The bisected graph can be seen in Figure 7.

Keeping in mind the goal of reaching a state of flow, we designed our conceptual framework that determines the type of assistance to be given in each of Figure 7's sections. For the top-left case, when the players feel that a task is more difficult than their own skills, we will assist the player by providing hints. Conversely, for the second case, when the players feel they are too skilled for their current task, we would dynamically adjust the difficulty of the task. The heuristics proposed by Gilleade and Dix [38] seem to match up with this model. The *assist me* heuristic suggests that

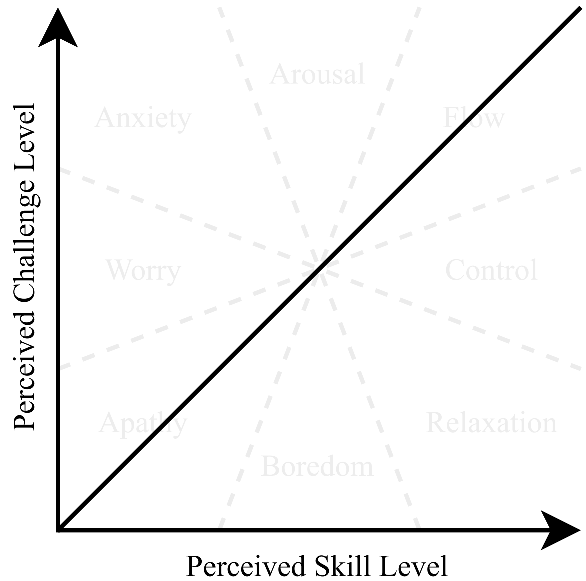

Figure 7. Our proposed bisection of the Experience Fluctuation Model.

players should be assisted before frustration levels become too high. The frustration affective states on the experience fluctuation model reside exactly on the top-left portion of the chart. In contrast, the bottom-right part of the graph contains affective states that map to low arousal levels. The *challenge me* heuristic states that the affective state can be used to alter the game's difficulty to better suit the players preferences. The bisected graph with corresponding applications can be seen in Figure 8.

If we analyze hints depending on how related to the task at hand they are, and if they are oriented to the solution or to the logic behind the task, we can infer that a hint that is unrelated to the task would increase the perceived challenge, since the apparent number of things the player needs to do to complete the challenge seems to increase. However, a hint that is task-related would feel like a part of the solution of the current problem, hence reducing the perceived challenge. When providing logic-oriented hints, in the long term, the players' knowledge about similar challenges increases, hence potentially increasing their perceived skill level greatly. When providing solution-oriented hints, the learning impact might be smaller, as the players were not guided through the logic behind the solution.

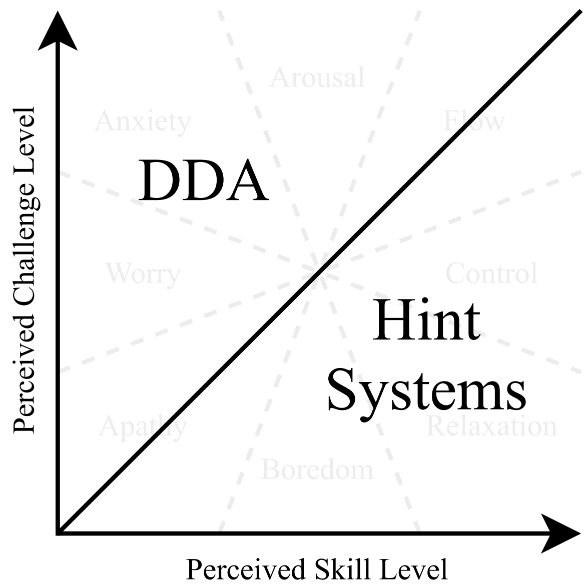

Figure 8. Our proposed bisection of the Experience Fluctuation Model showing the areas of application of DDA algorithms and hint systems.

An overview of how our conceptual model previews each type of hint impacts player experience can be seen in Figure 9. The horizontal axis represents task relatedness, and the vertical axis represents solution/logic-relatedness. The vectors, henceforth dubbed Player Influence Vectors, represent the potential direction and magnitude of the influence of a certain hint, for example, a logic-oriented task-unrelated hint would move player emotion towards the top-right corner of the experience fluctuation graph, closer to a flow state.

For examples of each type of hint, we will use Klei's Don't Starve [45] and ideate hints following our workflow using the game as a base. Don't Starve is an action-adventure survival game where the main goal is to survive as long as possible. When players are hungry, their task should be to find food. An example of a Solution-oriented, Task-unrelated hint towards finding something the character can eat would be the use of a moving object that is not directly related to food, such as a tumbleweed, to guide the player to a location where there is food. A Solution-oriented, Task-related hint could be a rabbit running across the players screen, leading the player to the location of a carrot. A Logic-oriented, Task-related hint would be seeing a rabbit run across the screen, however, after chasing the rabbit the players would reach the burrow which already has an existing trap over it, trapping the rabbit and showing players they can catch food using rabbit traps. Finally,

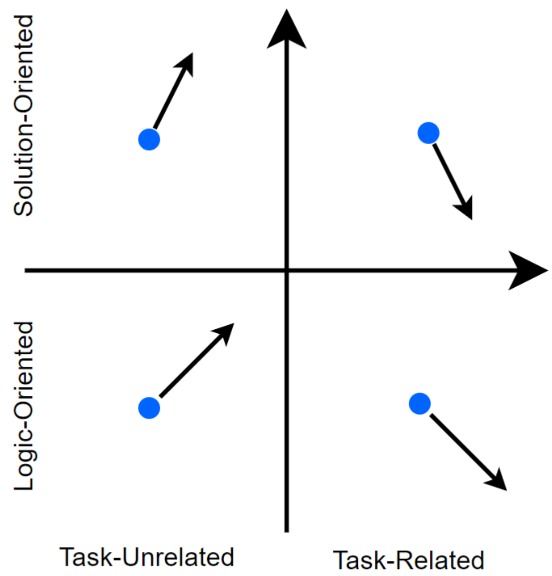

Figure 9. Player influence vectors based on task-relatedness and solution/logic-relatedness.

a Logic-oriented, task-unrelated hint would be the player being told they need to build a bed-roll to heal, perhaps through an in-game note or quest. Since bed-rolls require grass to craft, and rabbit burrows usually spawn near grass, when players search for grassy areas they might understand the underlying logic behind rabbit burrow locations.

By observing the proposed Player Influence Vectors, we can conclude that we can only guide the player from the top-left section of the flow diagram to the bottom right. No matter the combination of hints provided, we can either move closer towards the top right, while staying inside the top-left section, or move towards the bottom right, closer to DDA algorithms' area of influence. We propose these vectors be considered to determine what kind of hint to give to a player when aiming to reach a flow state.

To determine the best type of hint to provide when a player is in a given emotional state, we take the distance vector between the estimated player emotional state and the emotional state we want our player to feel at that given time. This goal vector, $\vec{\mathbf{g}}$, can be seen in Figure 10.

After the goal vector has been calculated, we can use the dot product to find the vector which more quickly guides the player to our goal emotion. Equation 3.1 shows the mathematical representation of the selection of the best hint from our pool of possible hints, where $h_{\vec{g}}$ is the selected hint

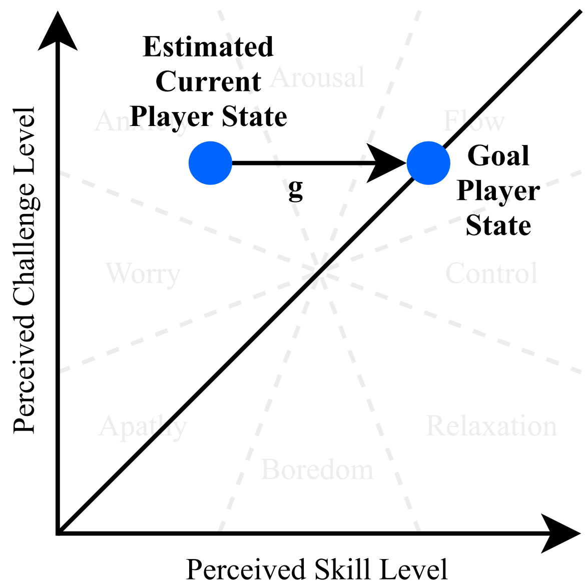

Figure 10. Goal vector calculated from an estimation of the current player emotion and our target emotion.

for the given goal vector, $\vec{\mathbf{g}}$ is the goal vector, h is a hint, H is the hint pool and $I(h)$ is the function which returns the player influence vector for the given hint. The potential influence vectors can be seen in Figure 9, and, each hint developed for a game would have an associated influence vector, which would be returned by $I(h)$.

$$h_{\vec{\mathbf{g}}} = \arg\max_{h \in H}(I(h) \cdot \vec{\mathbf{g}}) \tag{3.1}$$

Equation 3.1 was used in this dissertation as a guideline while developing our hint system and was not implemented. Further research could tackle the application of this equation as part of an implementation of the framework as a whole.

This selection system presents challenges which should be investigated further. To begin with, how do we know where the players are located on the experience fluctuation model at any given time, to be able to then guide them towards the goal emotion. Secondly, the proposed player influence vectors, which are based on intuition and heuristics, require further validation. Furthermore, research should also consider the blending of player influence vectors, as a hint might not be binary in how logic-oriented or task-related it is. Additionally, this conceptual framework only discusses the topic of which hint to provide to a player, without tackling the problem of when to provide hints.

3.2. Workflow

Considering the potential complexity behind hint systems, originated from the potential irregularity of the various variables behind a game, it seems advantageous to follow a methodical workflow, aimed at consistency, iteration and replicability. Since there are different foci during the various stages of development, the hint system's design should also develop throughout the various stages. An example of this could be that at a very early stage, most of the game will exist only as low-fidelity representation. The map will be represented by shapes or objects with no textures, to simply block out what the potential level design could be. Hints that are dependent on visual noise for disguise will be much more visible in the block out of a level than in the final high-fidelity version. An example for this can be seen in Figure 11, where a hint consisting of yellow particles to guide the player, is much less visible in the final high-fidelity version than on an initial low-fidelity version of ”Island”. This shows that including hints early on in the game design process is not enough, and that they must also be iterated upon in parallel with the rest of the game.

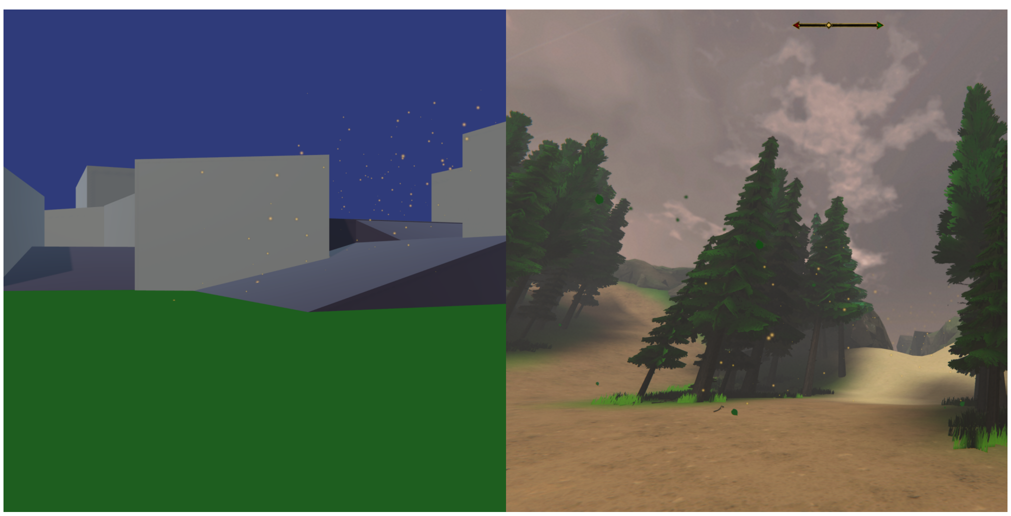

Figure 11. Firefly hint in low fidelity compared to high fidelity. Small yellow particles guide the players towards the location they need to go. In both images, the particles lead towards the right path.

The four steps behind the our proposed workflow are:

(1) Implement hint systems;

(2) Select a Player Experience Questionnaire;

(3) Playtest with questionnaire application and gathering of gameplay metrics;

(4) Analyze and evaluate results. If needed, return to (1).

Step (1) corresponds to the implementation of hint systems. The hint systems do not need to be complete, since they can be only prototypes that have only a portion of the planned hints.

For step (2), the goal is to analyze the various existing questionnaires of player experience, compare them, and select the one believed to be the most appropriate for the implemented game and hint systems.

Finally, the game must be played by playtesters within the target audience (3), ideally in quiet environment that fosters concentration. From the game, metrics regarding gameplay should be gathered for later analysis, such as number of wins, losses, time moving or others, depending on the genre of the game. Immediately after finishing the game, the selected experience questionnaire should be applied.

From the various playtests a number of results will be gathered and must be analyzed statistically to ensure the findings are correctly interpreted (4). This interpretation could point towards a superior hint system, from which a new iteration should be developed and the same workflow applied to then compare with the superior one from the previous iteration.

As it is likely that a game implementation changes in between iterations of hint systems, it is important to note that a different game implementation could potentially also impact player experience. This means that the results from a previous iteration of a hint system can no longer be compared to results gathered in the new game version. The results for the previous iteration must be gathered again by re-applying the proposed methodical workflow, for a fair comparison between iterations.

CHAPTER 4

Game Design and Development

Inspired by our proposed conceptual framework, which guides the though process behind developing and comparing hint systems, we developed an exploration game where players must find lost objects to progress, to study the impact between two different hint systems. Each level, the players must search for the wanted item and build up knowledge of the map and its features. If the player fails to find the objective item, and fails the level, the game resets the player back to the spawn position, and they must then repeat the objective until they are successful. This way, we could benchmark how the player performed when assisted by different hint systems, and made use of a post-game questionnaire to gauge how much each system impacted the player's enjoyment.

4.1. The Game

"Island" is a first-person exploration game, where exploration is guided by the need to find specific objects to survive, within a certain time-frame. It is implemented in the Unity game engine, and features keyboard and mouse input. A screenshot of the game, from the point of view of a player, can be seen in Figure 12.

After launching the game, players are presented with the main menu, where they are briefed on the experiment, and can choose to start the game whenever they feel prepared. After the start button is pressed, initial lore is shown to the player, which introduces the game's story and objective. Afterwards, the first of multiple possible repeatable items the player needs to find is introduced by some text about why the game's character needs it. For example, when looking for an axe, the game tells the player "It's going to get cold at night. I'll need to collect some firewood. An axe would make that job much easier for me and, if needed, I can use it to defend myself against any hostile wildlife". This text not only tells the players what item they need to find, but also the story on why the main character needs the item.

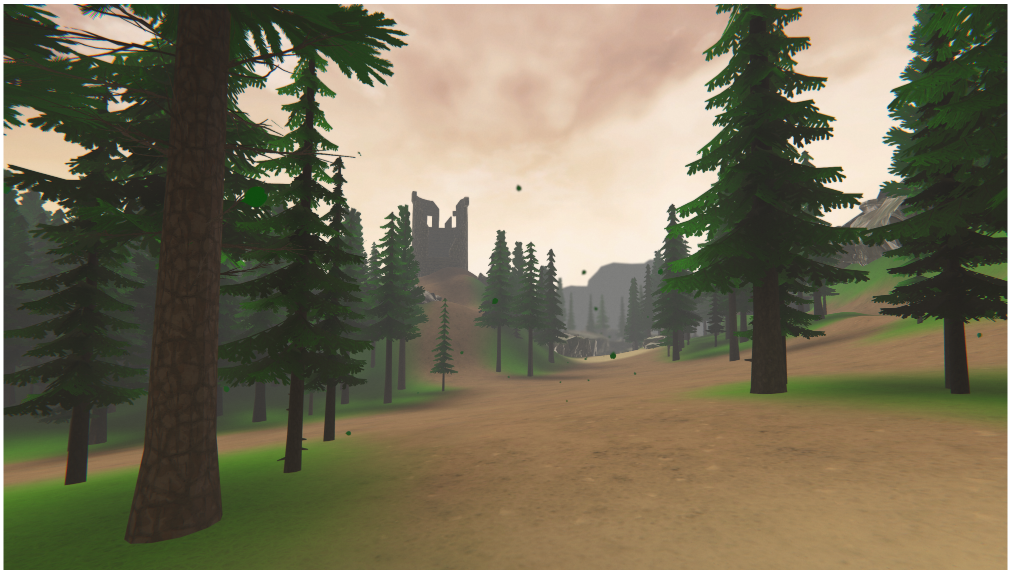

Figure 12. A screenshot of the game with no UI elements overlaid.

The player is then given control of the character within the game world, and can see a Heads Up Display (HUD) which indicates the amount of time left before the current level is failed in the top left corner, and an indication of the current item the player is searching for in the top right corner.

4.1.1. Mechanics

The game success or failure depends on a gauge that goes up when the player successfully picks up the required item, and goes down when the player fails to pick up the item. The gauge is capped at positive or negative twelve score points, meaning the game ends when reaching a cap. This value was found empirically through preliminary playtesting with five players, and was chosen because it presented a good balance between allowing the players to fail early in the game, due to lack of knowledge, and not being too hard to complete after the players understood the logic behind the items. Since score is a gauge, meaning a value that can both go up and down, up to a cap, this means the player can lose eleven rounds in a row, bringing their score to negative 11, and afterwards win twenty three rounds and still win the game, due to reaching a score of positive 12, allowing for the feeling of making a ”comeback” from an apparent obvious loss. The gauge UI can be seen in Figure 13.

When the player wins a round, meaning the object was found within the time limit, the game picks a new objective for the player. However, if the player does not find the object in the allotted

Figure 13. A screenshot of the score gauge where the score is 0 (top) and another, when the player has lost multiple rounds (bottom). The yellow handle moved to the left, closer to the end of the gauge.

time, and consequently loses the round, the objective remains the same, and in the same location, which means the player is not progressing. In both cases, at the start of a round the player is respawned back at the spawn position.

After the game ends, the player is presented with an end screen containing some final lore and a continue button. After pressing the continue button, a screen with raw JSON and a copy button is presented to the player. This JSON contains the extracted gameplay features, which are explained in Section 4.1.5 and are used for step (3) of our proposed workflow, which indicates that we must gather data regarding the gameplay.

4.1.2. Game Objectives

There are six different objective items that can be chosen for the player to find. These are the Axe, Knife, Backpack, Flashlight, Water bottles and the Map. Each item can spawn in only one biome. The possible biomes are beach, forest, and mountain, and are explained in Section 4.1.3. Within each biome items have a more specific spawn logic. An Axe, for example, can only be found stuck in, or around, old tree stumps. This logic was created to allow for players to create a gradual mental map. At first, a player might associate an item with its biome, and only after finding the item in a few different locations they might understand the second logic. An overview of the objectives and their corresponding spawn logic can be seen in Table 2.

The selection of the objective item is performed by extracting a random item from a pool of possible items. When the pool is empty, it is reset, hence guaranteeing every item will be selected

Table 2. Game objectives and corresponding spawn logic.

Objective	Biome	Spawn Logic
Axe	Forest	Stuck in, or around tree stumps
Knife	Forest	On or near the tables
Flashlight	Beach	On the beach
Backpack	Beach	On the beach, near rocks
Map	Mountain	Stuck to tower structures
Water Bottles	Mountain	Near tower structures

before repeating. However, this does not stop the consecutive selection of the same item when the pool is reset. Figure 14 shows a flowchart for the objective selection.

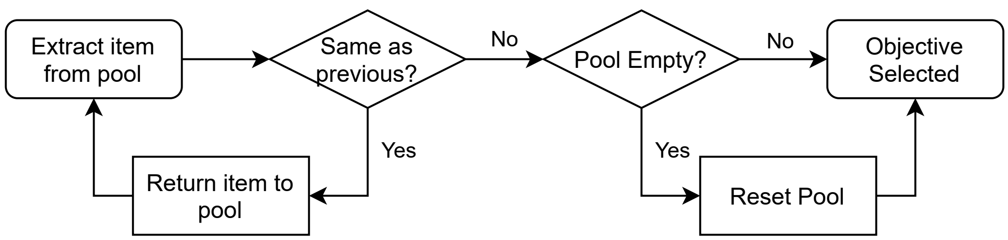

Figure 14. Objective Selection flowchart. When an item is drawn from the pool, it is compared to the previous item. If it is the same, it is placed back in the pool. If it is not, then the objective was selected. Whenever the pool is empty, it is filled with items again

To avoid repeating the same item twice in a row, a record of the previous objective is kept, and when the newly picked objective matches with the previous level's objective, the current objective is returned to the pool and a new iteration of this algorithm is executed. This is done until the new objective is different than the previous level's objective.

After the objective is chosen, its position is then selected from four possible positions within its biome. A similar algorithm to the one used for objectives ensures that after an object repeats, its position is not the same as the previous time the objective was chosen.

4.1.3. Objective Spawn Positions

To minimize variability between test sessions, the game's map is static, and item spawn positions are picked from a set of possible positions randomly. An overview of the game's map and potential spawn positions can be seen in Figure 15.

Figure 15. An overview of the map for ”Island”. Circles identify the different biomes. Numbers 1 and 5 identify the beach biome. 2 and 4 identify the mountains biome. Finally, 3 and 6 mark the forest biome

The potential spawn positions are split into biomes. Biomes are a collection of fauna and flora in a given area. In videogames, they commonly refer to collections of elements, be they fauna, flora, enemies, items or other objects. In ”Island” there are three possible biomes: beach, mountains, and forest.

The beach biome is represented on Figure 15 by numbers 1 and 5. It is the only biome close to water, featuring water sounds effects, and always features a sand path leading up to it. Additionally it is the only biome that has palm trees. Numbers 2 and 4 identify the mountains biome, whose main characteristic is being the only biome to feature a significant height elevation, an old stone tower meant to indicate this biome was usually used as a vantage point, and has wind sound effects. Finally, numbers 3 and 6 represent the forest biome. Here the player will be surrounded by trees and find old cottages. Additionally there are old tree stumps, representing a location where tree felling has occurred. Biomes were created to be visually distinct to reduce player memory load and assist in recognizing the logic behind each item.

4.1.4. Hints and Hint Selection

To assist players when they are having a difficult time trying to complete objectives, a lot of games present some form of assistance, to avoid their players feeling frustrated with the game and deciding not to continue playing.

This presents many challenges. For one, players have varying levels of experience. Some more experienced players might not want hints at all, while less experienced players might need hints to be able to progress. As discussed in Section 2.6.1, the players' age can also have a strong impact on their preferences. Younger players have more free time to spend learning about the game without receiving as many hints whereas an older player, with a family and job might have much less free time to spend on a game which takes a long time to learn without any assistance.

With all of these issues in mind, it is evident that designing effective hints, and an effective hint system, especially one for a broad target audience, is no trivial task. We opted to design hints with varying degrees of covertness, meaning the degree of visibility, or how evident the hint is, varied. This would mean that some hints might be seen only by players that require these hints, or ideally, might not even be interpreted as hints at all.

"Island" has four different hints available, with varying degrees of covertness, and appealing to different kinds of players. Four hints were created and ordered according to their covertness:

(1) Footprints - The character would leave footprints where they had already been, ideal for players with lower spacial awareness to be able to more easily identify which zones that have already been explored. Since this hint is heavily integrated with the map and the character, we classified it as the most covert (Figure 16);
(2) Fireflies - Blinking yellow particles indicate the way to the zone where the item is located. Although this hint can give the entirety of the solution, the fact that it is represented by yellow particles which blend in with the brown and yellow tones of the map, and with the other falling leaf particles, makes this hint also very covert. Since it is located more at eye-level, and have no direct connection with the in game character or story, we classified it as less covert than the footprints(Figure 17);

(3) Riddle - A textual riddle which indicates in which biome the player might find the items. Since this hint is presented through a UI element, it is very visible, making it one of the least covert hints(Figure 18);

(4) Photo - A picture of the item with landmarks visible. The photo was classified as the least covert hint due to the fact that it has a bigger UI change when it appears, compared to the previous riddle hint, and due to it providing enough information for the player to immediately know where the item is located(Figure 19).

Figure 16. Footprints hint when searching for an Axe. Note the brown footprints leading up the pathway. This indicates to the player that this area has already been explored.

The hints fall on different quadrants of the player influence vector model as can be seen in Figure 20. The footprints hint is considered task-unrelated because it is not related to any specific objective, and logic-oriented because it helps players understand where they have already been, and as such increase their knowledge of the island, which will help them understand the logic behind each item. The Fireflies hint is task-unrelated because there is no direct association between fireflies and any of the objectives. It guides player attention through movement and contrasting colours. It is solution-oriented because it guides players to the location of the objectives. The riddle hint is task-related because it always mentions the objective in some way, and logic-oriented because it helps

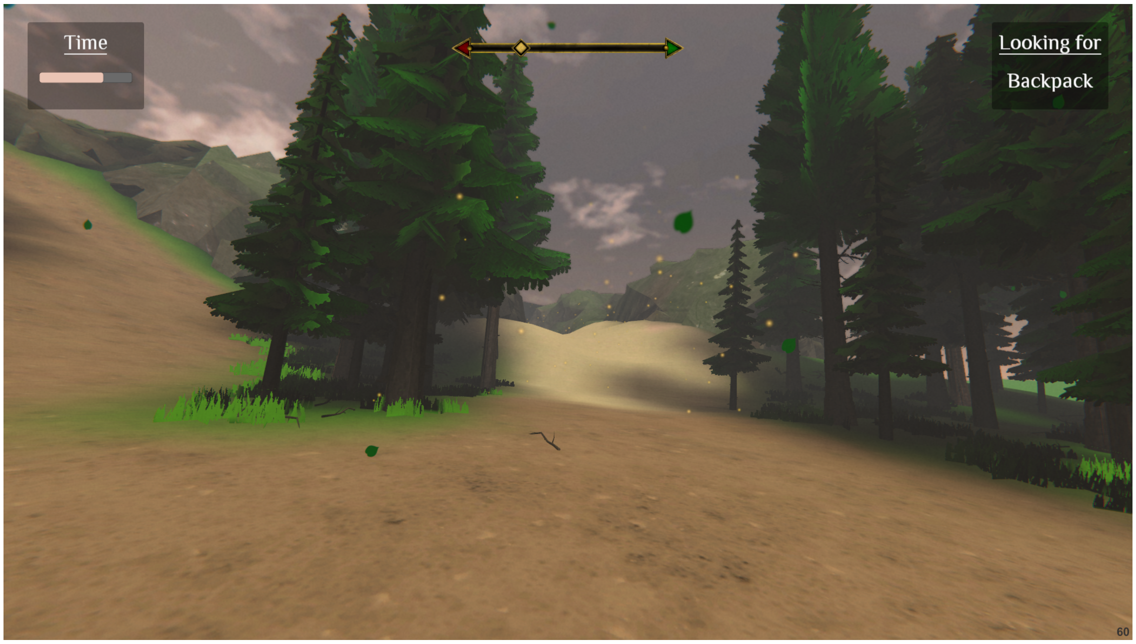

Figure 17. Fireflies hint when searching for a flashlight. Note the yellow particles leading up to the sandy area, the biome where the flashlight spawns.

Figure 18. Riddle hint when searching for a Backpack. The hint is visible at the top of the screen, below the score gauge.

players understand the underlying logic behind each item. Finally, the photo hint is task-related as it refers to each of the objectives, and solution-oriented as it shows their exact locations.

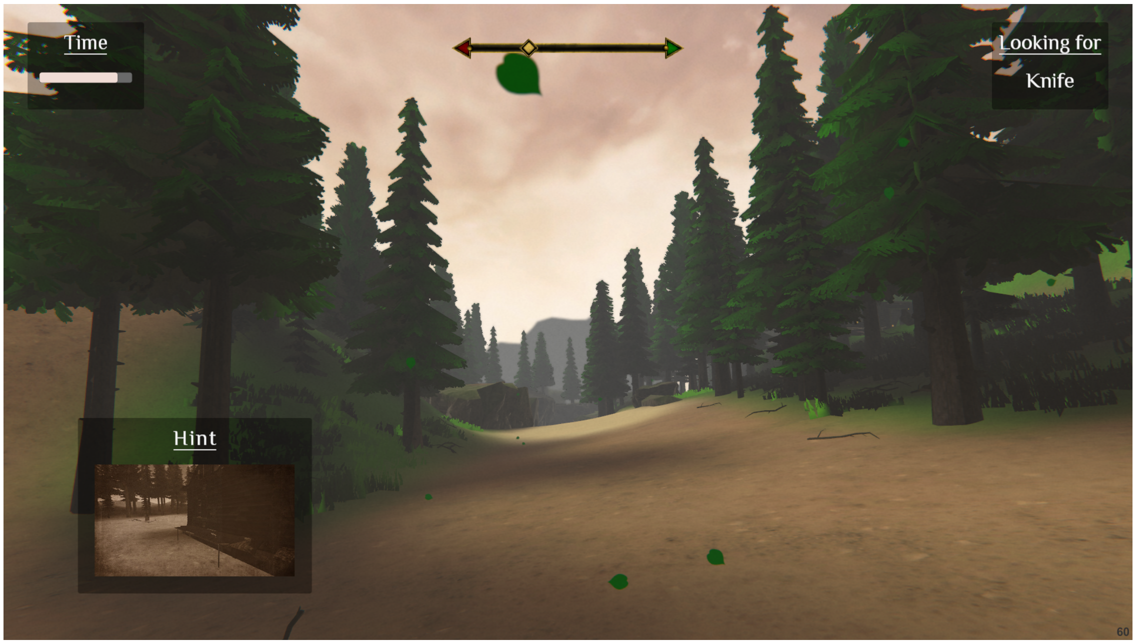

Figure 19. Photo hint when searching for a knife. The hint is visible on the bottom left of the screen.

Regarding the selection of the hints, we implemented two different systems. The first system chooses a random hint whenever the player loses a level. There is no restriction to which hint is chosen and, as such, the game can pick the same hint type throughout the entirety of the game. This serves as a baseline to compare other hint systems to. The second system chooses the most covert hint after the player loses the first time for an item. If the player loses again, it moves down the covertness level, and keeps doing this until it reaches the least covert hint, or resets if the player wins the round. This decision was made due to finding conflicting studies regarding player preference of hint symbolism. To ensure all players can get the types of hints they prefer, we provide the most covert hints first, and the most overt hints last.

The type of hint selection is handled by a flag, controlled by a checkbox that is visible when players start the game, which chooses between random hint selection, and the covert to overt hint selection system. Players were instructed on whether to select the checkbox, but not what the checkbox is for.

The riddle hints were all designed in such a way that they would help the player in finding the logic behind the item and not its precise location. For instance, the hint for the Map tells the player that a map is usually used from a high position, which makes sense as it is where one would have

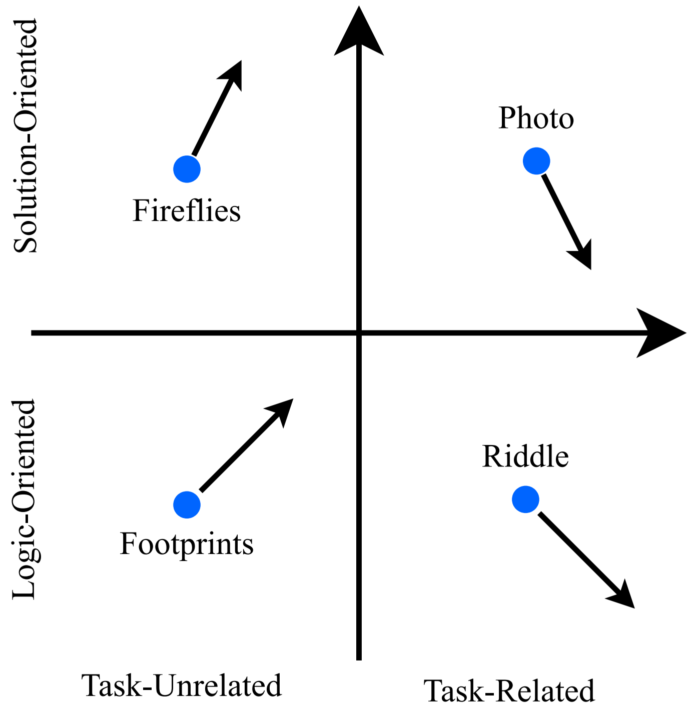

Figure 20. Player influence vectors for the hints in ”Island”.

the most visibility of surrounding interest points. This primes the player to understand the logic behind the item, without explicitly saying that the maps are always at the top of the mountains. The riddle for each item can be seen in Table 3.

Table 3. Game objectives and corresponding riddles.

Objective	Riddle
Axe	Firewood is used for keeping fireplaces lit
Knife	Skinning animals makes a mess. I wouldn’t do it inside the house
Flashlight	You tried using one to signal passing ships
Backpack	I dropped mine when I arrived and got off the ship
Map	A map is best used from a high point
Water Bottles	You remember having them in a windy location

4.1.5. Gameplay Features

In order to compare results between players, it would be useful to extract gameplay features regarding each player's game session. With this in mind, a number of gameplay features were extracted:

- UUID - A unique id per player that was used to match the extracted features to their questionnaire responses;
- Round Time Moving - The amount of time in seconds the player spent moving in a given round;
- Round Time Idle - The amount of time in seconds the player spent idle in a given round;
- Round Length - The duration of the round in seconds;
- Round Objective - The objective for the current round;
- Round Won - True if the round was won, false otherwise;
- Using Random Hints - If the player in question was using a random hint selection.

These features were chosen to help characterize the game and the testers. By analyzing the gameplay features and the tester demographics we can not only compare results between random hint selection and an covert to overt hint selection, but also view how difficult each objective is and compare more experienced players with less experienced players.

These analytics were stored in a dictionary and later converted to JSON for playtesters to paste into the questionnaire presented in Section 5.1. We chose to utilize JSON for easier parsing later when we would do data analysis. Another encoding could have been used such as XML to achieve the same results. An example of the gameplay features encoded in JSON format can be seen below:


```
{
    "uuid": "b1c7777d-d619-4c85-aff2-6e49681dd24f",
    "roundCounter": 31,
    "timeAnalytics": {
        "0":{"total_time_moving":22.4996147,"total_time_idle":4.158269},
        "1":{"total_time_moving":15.1245308,"total_time_idle":2.75829458},
        ...
        "30":{"total_time_moving":13.5916491,"total_time_idle":1.09999955}},
    "winLossCounter": {
```

```
        "win":22,
        "lose":10
    },
    "itemCounter": {
        "Flashlight":3,
        "Map":4,
        "Backpack":4,
        "Knife":3,
        "Water Bottles":4,
        "Axe":4
    },
    "randomHints": true
}
```


The ”uuid” key contains a unique 128 bit id which is used to facilitate pairing the player questionnaire with their demographics, and referring to individual playtests when analyzing data. the ”roundCounter” key contains the number of rounds played during the playtest, in this case the playtest lasted for 31 rounds. The ”timeAnalytics” key contains a mapping between a round number, and how much time players spent moving, or standing still. For this player, we can see they spent approximately 4.16 seconds idling in the first round, and only 1.10 seconds idling in the last round. The next key is the ”winLossCounter” which contains the number of rounds the player won and lost. Then, the ”itemCounter” key indicates how many of each objective the player had to complete. Note that the sum of all the number of objectives will at best be the same as the round counter, since an objective can have multiple rounds if the player does not complete the objective on the first try. Finally, the ”randomHints” key contains a boolean indicating if the game was played with a random hint selection or a covert-overt hint selection.

4.2. Development

An initial analysis of two commercial and one open source game engines was performed to find the one most suitable for the development of a 3D exploration game. The game engine should be free for educational use, have sufficient documentation and information online and have an asset library for finding 3D models and scripts for quick prototyping and development. The results of this comparison can be seen in Table 4.

Table 4. Comparison between game engines.

	Unity [46]	Unreal Engine [47]	Godot [48]
Language	C#	C++	GDScript
Open Source	No	No	Yes
Information Online	High	Low	Medium
Pricing	Free	Free	Free
Asset Library	Yes	Yes	Yes

Unity, Unreal and Godot are all free for educational use and feature an asset library which contains assets ideal for initial prototyping. A search for ”<Game Engine> Tutorial” on Google retrieved 118 million hits for Unity, 35.5 million hits for the Unreal Engine, and 1.21 million hits for Godot. This smaller amount can be explained due to Godot being the most recent engine of the three. Additionally, Unity is the engine most used by indie game developers, as can be seen in the submission results for the GMTK Game Jam [49] in Figure 21.

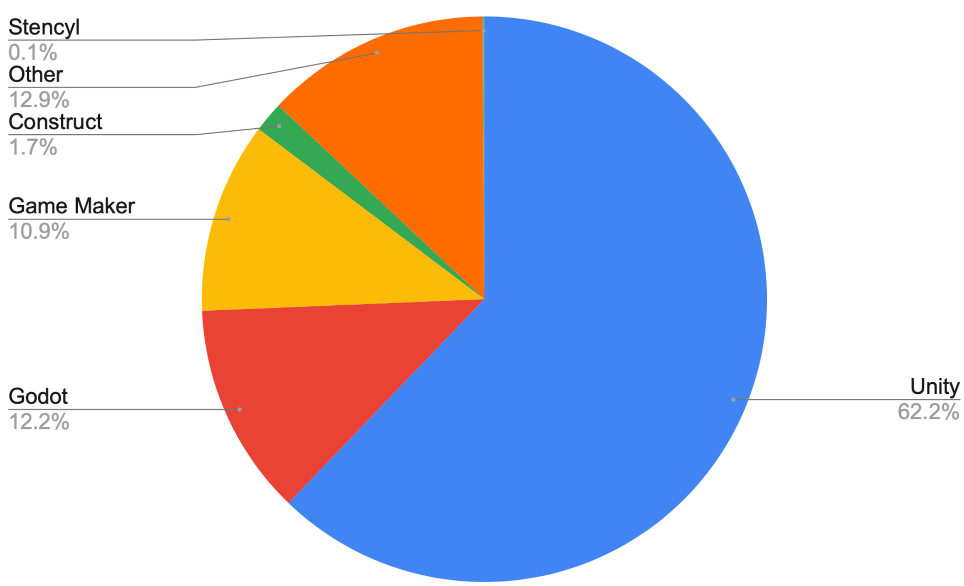

Figure 21. A pie chart of game engines used for 5400 of games made for the 2020 GMTK Game Jam [49]. Unreal Engine falls into the category ”Other”.

The decision was made to make the implementation using the Unity game engine and the C# programming language. This choice was made mainly due to the amount of information available online, and the fact that it is the engine most used by indie game developers.

The game implementation is divided into two main parts. The first is the world, which includes the terrain, details, ambient sounds and light sources. The second is the game components, which include the main game orchestrator, hint selector and the lore and UI managers. The two most important components, responsible for objective orchestration and hint selection, were discussed in further detail in Sections 4.1.2 and 4.1.4 respectively.

4.2.1. Inter-Component Communication

As was discussed in the Theoretical Framework in Section 3, iteration is a core concept behind game development, especially during early development due to the need to balance the game. The game must be balanced in such a way that most players find it challenging, for our hint system to be useful. With iteration in mind, special care was taken to avoid one of the most common pitfalls in software and game development: tight-coupling.

Tight coupling is when a system depends directly on other systems, meaning changes to one potentially causes others to break. In contrast, loose coupling is designing a system while reducing the amount of connections between different parts of a program, and keeping the existing connections small, simple, direct and flexible [50]. This way, if a change in a component is executed, this change would impact, at worst, the reduced number of connected systems and, at best, since the connections are small and simple, no connected systems at all.

We decided to look at the individual internal game components, which are responsible for orchestrating the game logic, as black boxes which communicate through a publish-subscribe messaging pattern. They read from zero or more input channels, commonly known as topics, and can write to zero or more output channels.

By using this messaging pattern, each component can expose a public API to an unknown number of subscribers. Each component only depends on messages appearing on certain topics to be able to function, meaning that components are agnostic to who published the messages. This way, changes to a component which publishes messages to a topic would not break existing subscribers as long as the messages are still published to said topic.

A diagram representing the key components of the main game orchestration can be seen in Figure 22.

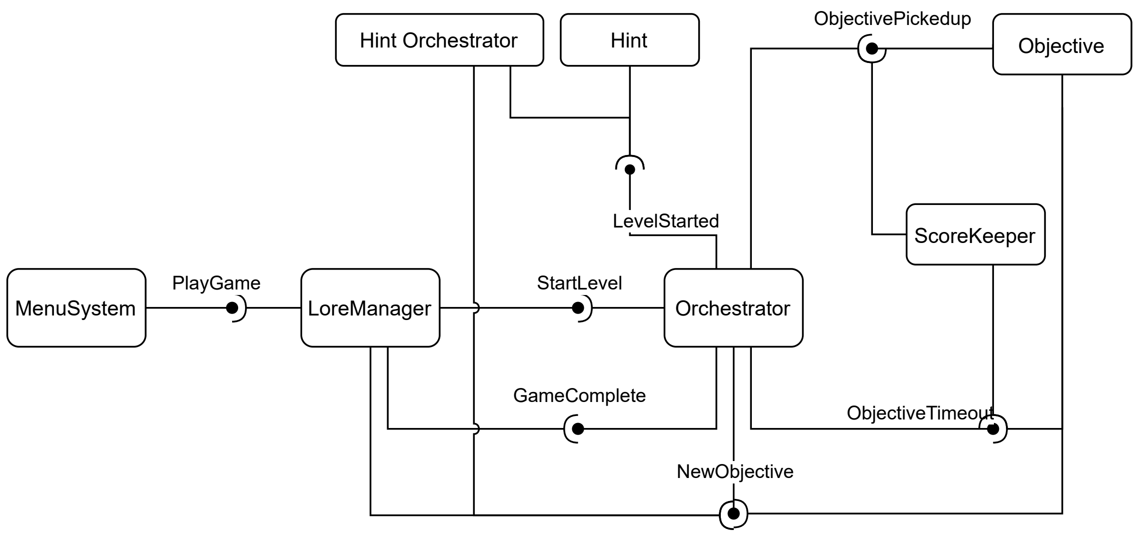

Figure 22. A components diagram for "Island"

The diagram uses ball and socket notation. The ball represents a provided interface, whereas the socket represents the requirement of an interface. For example, for the LoreManager to function, the MenuSystem must abide by its established interface and publish a message onto the PlayGame topic. Similarly, the ScoreKeeper listens to the ObjectivePickedup topic to increase the score, and to the ObjectiveTimeout topic to decrease the score. It does not need to know what component is publishing the messages, just that when one is received, it needs to increase or decrease the score.

We implemented a publish-subscribe messaging system based on C#'s delegates. A delegate is a pointer to a method and optionally an object to call the method on. Delegates are often used to implement callbacks. The following code was implemented by us for managing message distribution. When a component is interested in receiving messages from a certain topic, it calls the subscriber creation method for our messages broker:

```
Broker#Subscriber(topic, Action<BaseMessage> callback, GameObject caller)
```

The Broker class is responsible for handling all references to publishers and subscribers. By calling `Broker.Subscriber()`, a callback is registered onto a dictionary, with the key being the given `topic`, and the values being list of given callbacks, with the appended `callback`. The callback is of type `Action`, an encapsulation of a delegate that takes a single parameter of type `BaseMessage` and does not return a value.

A `BaseMessage` is a baseclass for all messages to allow for polymorphism when handling messages. A message publisher can define a subtype of `BaseMessage` to carry additional information with the published message. For example, the `Orchestrator` component publishes messages of the `NewObjectiveMessage` type onto the `NewObjetive` topic, which contain the new objective index and its name. When a component needs to create a publisher it calls:

```
Broker#Publisher()
```

This method registers the publisher internally on the Broker, and returns a `Publisher` instance for the caller to use. Whenever a caller wants to publish a message they can call:

```
Publisher#Publish(BaseMessage message)
```

This makes the publisher loop through all subscribers registered in the Broker and calls the registered callbacks with the given `message` argument. This way the Broker handles all references and avoids the need for coupling between components.

4.2.2. Builds

Unity supports building for the Windows, Mac and Linux Operating Systems, Android and iOS, Web and consoles. For the purposes of this experiment, two different builds of the game were created, one targeting the web, for ease of access, and one targeting Windows for maximizing graphical quality and framerate.

The Web build was created to facilitate distribution. By allowing anyone with a computer to click a link and start playing the game, we could expand our potential player base and get more information. After testing the build with one of our playtesters, this build ran at a much lower quality and with much lower framerate, with occasional freezes.

Although Unity supports building for multiple platforms, this shows that getting a game running on more than one is not as easy as running the build process for each. To get each of the various builds running with the same framerate and quality would require iteration time. Due to the inferior game experience, the Web build was dropped in favour of the Windows build.

The Windows build features higher visual quality and framerate. The game can be downloaded through itch.io[1], has a size of 295MB when compressed, and 579MB when not compressed.

[1]Link to the game's page on itch.io

CHAPTER 5

Evaluation

This chapter describes the set of tests that were carried out and presents and describes the main results. Results are split into three sections. Section 5.2 presents the results regarding the demographic questions. Section 5.3 presents the extracted gameplay metrics from the various playtests, for players with random hint selection and covert to overt hint selection. Finally, Section 5.4 presents the results and statistical analysis of the game experience questionnaires run at the end of each playtest.

Statistical analysis was performed using Wilcoxon Rank Sums test due to it being a nonparametric test of the null hypothesis, avoiding making assumptions about the distribution of the observed data. Additionally, when the data was shown to be statistically significant, effect size was calculated using Hedges's g, which is adequate for small sample sizes (N = 20).

5.1. Playtests

Due to the Covid-19 pandemic, to allow for a safe space where playtesters were free from worry of contamination, the tests were performed in a remote-only environment through Zoom [51] or Discord [52]. An effort was made to request that tests be performed in a calm and quiet environment, in a place where testers would often play games at home. A playtest was divided into five stages:

(1) First contact - Potential playtesters were contacted through social media and asked if they would be interested in participating in a study regarding videogames;
(2) Initial briefing and demographic questions - Playtesters were briefed on the controls, rules, objectives and mechanic of the game;
(3) Playing the game - The playtesters played the game to completion, while sharing their screen;
(4) Experience Questionnaire - The players filled out the Experience Questionnaire;
(5) Informal discussion - An informal discussion regarding the game, how the playtesters felt about each objective, hint, and situations that arose during their playtime.

The potential participants were contacted through social media (1) and given information regarding the study, specifically that it was for a Masters Degree, related to videogames, and that it would take on average between 45 and 60 minutes. Those interested were asked to pick a time where they could be in a calm environment, where they would not be interrupted for the duration of the experiment.

The experiment started with players being presented with a link to an online form to fill demographic questions. The questions asked in the form are available in Appendix A and ask about the playtesters age, gender, experience with videogames, favourite videogame genres and fluency in English. After filling out the demographic questions, players were directed to the itch.io link where they would download the game. While the game was downloading, they were informed (2) of the games controls, rules and objectives. Players were also informed the aim of the study was not to evaluate their performance at the game, but the game itself, and that the study would be anonymous. This was done in an effort to reduce social desirability bias. Additionally, players were not told that the main focus of the study was regarding the hint systems. The download of the game always finished before the explanation was complete.

After the initial briefing was done and the players had the game prepared in their computers, they were asked to share their screen and launch the game. After the game was started (3), half the players were informed that they should select a checkbox labeled with ”RHS”, an acronym for ”Random Hint Selection” which was used as to not give clues to the playtesters of the content of the study they were participating in. The checkbox chooses between random hint selection and a covertness based hint selection.

After the game was over, the gameplay features discussed in Section 4.1.5 were presented on the screen in JSON format with a Copy button, to allow for playtesters to paste the analytics into the questionnaire. Finally, players were asked to fill an Experience Questionnaire, which contains a version of the Game Experience Questionnaire (4) that features two of the three core modules proposed by IJsselsteijn *et al*. [53], namely the core module and the post-game module. The social presence module was not used because the game features no in-game characters or social interactions. The Game Experience Questionnaire (GEQ) was chosen due to its emphasis on Flow as component, and its wide use within the study of videogame experience, as shown in Section 2.6.2.

5.2. Demographics

A total of 20 participants participated in the experiment. During development, a total of 5 test sessions were performed as formative tests to assess the difficulty of the game and perform changes accordingly, to ensure the game is balanced. After the game was considered balanced and completed, 20 summative tests were used to calculate the values shown in Sections 5.3 and 5.4.

Participant ages ranged from 18 to 25 years old, with mean age being 21.00 ± 1.73. There were 3 participants that chose ”Female” as the gender they most identified as. A total of 16 participants classified themselves as ”Male”. One participant preferred to stay unidentified and no participants chose the ”Fill in the blank” option.

In regards to their perception on their experience as videogame players, 13 participants classified themselves as ”Very experienced” and 7 as ”Somewhat experienced”. No one classified themselves as ”Not experienced at all”. Out of the participants that classified themselves as ”Somewhat experienced”, 2 played with a random hint selection and 5 with a covert to overt hint selection. Of the 13 very experienced participants, 8 played with a random hint selection, and 5 with a covert-overt hint selection. This distribution of player experience could not be controlled because we did not know the player self-reported experience until after the participants finished playing the game and answering the questionnaire. The player experience and types of hint selection used can be seen in Table 5.

Table 5. Participant self-reported experience level and types of hint selection used in playtesting

	Total	Random Hint Selection	Covert-Overt Hint Selection
Very Experienced	13	8	5
Somewhat Experienced	7	2	5

Participants were also asked to pick on average how long they spent playing video games per week in the last 3 months. One player stated that, due to recently starting a new job, their average playtime would be 0 hours, but that in the past they would play very regularly. The histogram of the participants answers can be seen in Figure 23.

If we consider that the participant that chose the 0 hours option used to play very regularly, we can observe that all our participants play videogames regularly, with 12 participants playing more

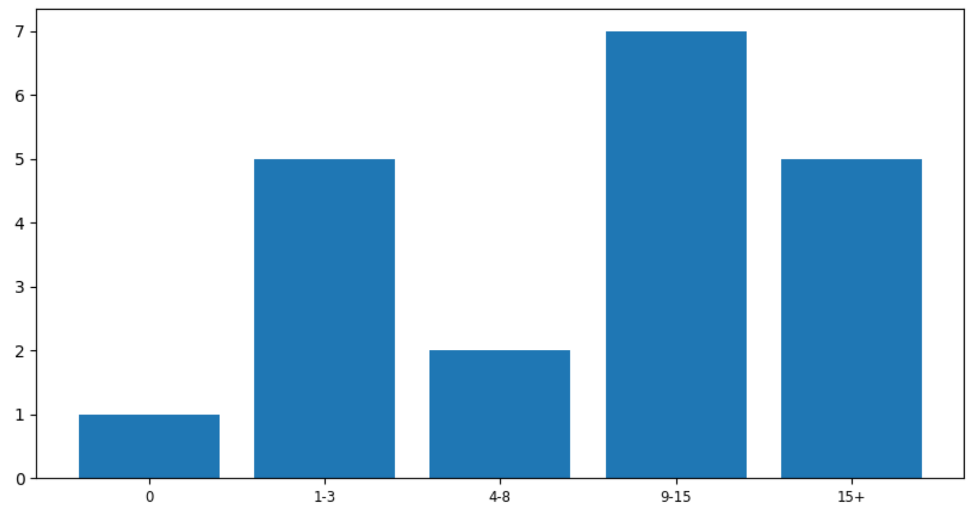

Figure 23. Histogram of participants' average weekly time playing videogames in the past three months.

than 9 hours per week. This indicates that all our participants have a degree of experience with videogames, with 12 participants having constant large play sessions every week.

5.3. Gameplay Features

A playthrough lasted on average 32.85 ± 7.78 rounds. Each round lasted on average 44.53 ± 25.52 seconds, which means each game took on average 24.38 minutes excluding times spent reading lore between rounds. For participants with a random hint selection, the mean number of rounds was 29.3 ± 6.43. In contrast, participants with a covert to overt hint selection played on average for 36.4 ± 7.65 rounds. This can be seen in Table 6.

Table 6. Average rounds played per hint selection type.

Hint Selection Type	Average	Standard Deviation
Both	32.85	7.78
Random Hint	29.30	6.43
Covert-Overt	36.40	7.65

The data is statistically significant within a 90% confidence level (p = .059) and shows a large effect size (g = 0.962). The difference in means implies that a random hint selection causes the games to be shorter, which we can attribute to the higher frequency of highly overt, more noticeable hints. More overt hints mean a player gets more information faster, hence finishing the game faster. This means that simply altering a hint selection system can have an impact on gameplay features.

Regarding the difficulty behind each item, we looked at the number of objectives for which the player required at least one hint over total number of objectives. With this ratio we are effectively analyzing the difficulty playtesters have with each item. We define an objective as a series of rounds that requested the participant to find the same item. The following sequence presents the items from a hypothetical playtest:

`Axe, Axe, Axe, Map, Knife, Flashlight, Backpack, Backpack, Water Bottles.`

The sequences for this playtest would be "`Axe, Axe, Axe`", "`Map`", "`Knife`", "`Flashlight`", "`Backpack, Backpack`" and "`Water Bottles`". From these, the Axe and Backpack objectives are the ones for which the player required at least one hint, because they lasted more than one round. The percentage of objectives with hints for this playtest would be the number of objectives with hints over the number of all objectives. This would equate to `2/6`, which is approximately 33.33%. Figure 24 shows the average percentage of long objectives for players using random hint selection, labeled as ”RH”, and for players using covert to overt hint selection, labeled as ”NRH”, which stands for non-random hints. The same data can be seen on Table 7 with corresponding standard deviations.

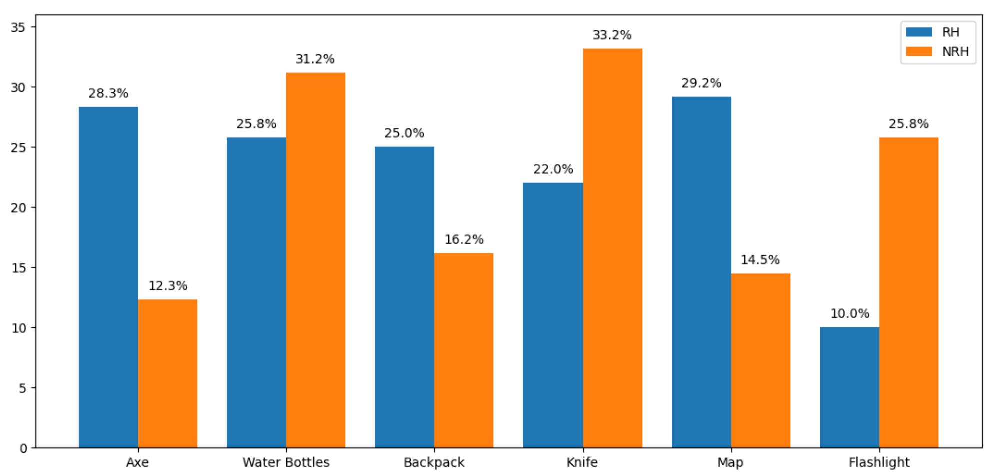

Figure 24. Percentage of long item objectives for each item when using a random hint selection (RH) and non-random hint selection (NRH).

Table 7. Percentage of long item objectives for each item (Mean ± Std. Dev.).

	Axe	Water Bottles	Backpack	Knife	Map	Flashlight
RH	28.3% ± 12.5%	25.8% ± 9.6	25.0% ± 22.7	22.0% ± 22.0	29.2% ± 17.5	10.0% ± 17.1
NRH	12.3% ± 13.5	31.2% ± 13.4	16.2% ± 12.5	33.2% ± 18.7	14.5% ± 15.8	25.8% ± 12.2

The Axe (p = .014, g = 1.176), Backpack (p = .625) and Map (p = .473) all had a higher sample mean of long objectives when played with a random hint selection system. The Water Bottles (p = .650), Knife (p = .345) and Flashlight (p = .364) had a higher sample mean percentage of long objectives in a covert to overt hint selection.

We have statistical evidence that, for one of the items, the Axe, there is a superior level of difficulty when applying a random hint selection compared to a covert-overt hint selection. For the remaining items, although the sample mean was higher, the statistical test does not guarantee that after repeating the process we would achieve the same results.

The Axe presented statistical significance (p = .014) within a 90% confidence level and a large effect size (g = 1.176) regarding the percentage of objectives that required hints between each type of hint selection. We believe this to be potentially attributed to multiple factors. In first place, the axe is always the initial objective when players start the game. When using a random hint selection, the players will most likely get a highly overt and concrete hint such as the riddle or photo, which tells the player most information required to find the objective, potentially leading the players to it without requiring that they understand the logic behind the item. This could cause players to need more hints later in the game, because they did not build enough knowledge that could be used later on. In contrast, a covert-overt hint selection always makes players wait until they have failed twice before providing a riddle, and then another failure to receive the photo hint. This means players have a total of three covert rounds, one with no hint, one with footsteps and one with the fireflies hint, to explore the map and as such, are compelled to learn the underlying logic, useful in subsequent rounds.

The axe is also the item that has most logical relation with its context in game. An axe is always found stuck in or around tree stumps. If players that used a covert-overt hint selection system found the item before the hint system started giving more overt hints, they could potentially explore more of the map and understand the logic of the item by themselves, making them require less hints in the future. This could be helpful when designing educational games. By giving players enough time and freedom to explore, minimizing the number of hints that the players perceive as being assistance, we can make the players feel empowered and possibly learn more.

The remaining means of objectives that required hints did not present statistical significance, hence it is harder to make conclusions regarding their results.

Participants that played with a random hint selection system had an average win rate of 71.87% ± 6.37% of rounds played, whereas a covert to overt hint selection had an average win rate of 67.27% ± 4.19% of rounds. This difference was found to be statistically significant within a 95% confidence level, and with a large effect size ($p = .049$, $g = 0.817$), showing that a change between these hint selection systems impacts gameplay. If a designer were to alter a hint selection system from random selection to covert-overt, they could potentially be impacting game difficulty and length.

Regarding the time participants spent moving, Figure 25 shows the average times idling or moving per hint selection system. Although both the idle and moving times for non random hint selection appear to be higher, the difference is not statistically significant for the idle times ($p = .999$) or the moving times ($p = 0.364$). With this we can conclude that there is no significant difference between the times players spend moving or idling when comparing a random hint selection system or a covert-overt hint selection system for ”Island”.

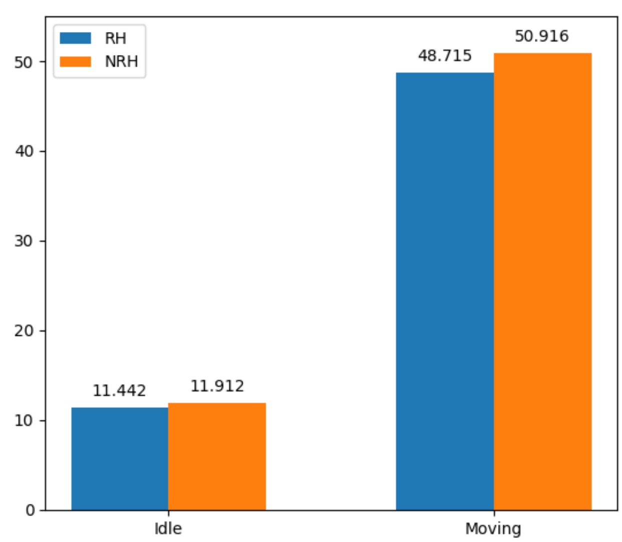

Figure 25. Time spent idle and moving for random hint selection and non-random hint selection.

This could give us a clue as to the variability shown in the difficulty data. Since there is no significant difference in times spent moving or idle between random hint selection and covert-overt hint selection, we can postulate that all players spent most of the game attempting to brute force a solution instead of stopping to think and internalize the hints the game was providing. Perhaps by implementing a system which slows down the player, such as cutscenes or textual story to be read. Outer Wilds presents moments where the players are controlling the character and can interact with the environment, and other moments where the players are controlling their spaceship and are forced to slow down and have time to think about what is happening. The main conclusions drawn from the data are:

(1) A covert-overt hint selection system increases game length greatly when compared to the baseline random hint selection system (p = .059, g = 0.962). This shows that we can influence gameplay by altering hint selection;
(2) For the axe, we found that using a covert-overt hint selection system increased player skill greatly when compared to the baseline (p = .014, g = 1.176), showing that we can increase player skill by altering our selection of hints;
(3) A covert-overt hint selection system caused players to lose many more rounds when compared to the baseline (p = .049, g = 0.817), which shows we can influence gameplay by altering hint selection.

5.4. Game Experience Questionnaire

The average results of the Game Experience Questionnaire from the 20 participants for both random hint selection and non random hint selection can be seen in Table 8 for the GEQ Core Module and Table 9 for the GEQ Post-Game module.

The core module results represent player experience while playing the game. For the competence, immersion, flow and positive affect components, the averages are higher in the random hint selection system when compared to the covert-overt hint selection system. The challenge and negative affect components are higher in the covert-overt system, which is expected as the games took longer and players lost more rounds than with the random hint selection system. The tension/annoyance component is very similar in both selection systems.

Table 8. Game Experience Questionnaire Core Module Results for N = 10 (Mean ± Std. Dev.).

Component	Random Hint Selection	Covert-Overt Hint Selection
Competence	2.68 ± 0.42	2.14 ± 0.89
Sensory and Imaginative Immersion	2.23 ± 0.47	2.05 ± 0.80
Flow	2.58 ± 0.73	2.26 ± 1.20
Tension/Annoyance	0.63 ± 0.76	0.60 ± 0.70
Challenge	1.74 ± 0.63	1.88 ± 0.91
Negative affect	0.45 ± 0.50	0.70 ± 0.64
Positive affect	2.84 ± 0.47	2.66 ± 0.71

Regarding how participants felt after they finished playing the game, Table 9 shows the results of the post-game module. Both of the positive components (Positive affect and returning to reality) show superior values when playing with a random hint selection system. The negative experience component shows a higher value in the covert-overt selection system. Similarly, the tiredness component shows the same results, which is expected since the games took on average 7.1 more rounds with a covert-overt selection system.

Table 9. Game Experience Questionnaire Post-Game Module Results for N = 10 (Mean ± Std. Dev.).

Component	Random Hint Selection	Covert-Overt Hint Selection
Positive Experience	1.87 ± 0.67	2.00 ± 1.04
Negative experience	0.10 ± 0.21	0.22 ± 0.18
Tiredness	0.20 ± 0.35	0.60 ± 1.07
Returning to Reality	0.73 ± 0.41	0.50 ± 0.36

Tables 10 and 11 show that the values do not present statistical significance within a 95% confidence level. This means that even when the sample mean is superior, we cannot guarantee that repeating the process would achieve the same results. However, an interesting characteristic of our data is that in Table 8, the covert-overt hint selection data presents a much higher dispersion (higher standard deviation) than when using a random hint selection system. This could suggest that the presence of a random selection strategy homogenized the data, and, as such, could corroborate our previous results that altering the hint selection system can impact gameplay features and in this case, player experience. Regardless, a higher sample size would be required to confirm these hypotheses.

Table 10. Game Experience Questionnaire Core Module Wilcoxon rank sum test.

Component	Statistic	P-Value
Competence	1.474	0.140
Sensory and Imaginative Immersion	0.567	0.571
Flow	0.189	0.850
Tension/Annoyance	0.038	0.970
Challenge	-0.340	0.734
Negative affect	-0.869	0.385
Positive affect	0.529	0.597

Table 11. Game Experience Questionnaire Post-Game Module Wilcoxon rank sum test.

Component	Statistic	P-Value
Positive Experience	-0.227	0.821
Negative experience	-1.814	0.070
Tiredness	-0.302	0.762
Returning to Reality	1.474	0.140

5.5. Informal Post-Game Discussions

After the participants finished playing the game and answering the player experience questionnaire, there was an informal discussion regarding how the participants felt about the game.

All participants felt the visual quality of the game was good, with one player making a comparison to Dark Souls. Some participants mentioned that the trees waving in the wind and the water waves, along with their corresponding wind and water splash sound effects added to the atmosphere greatly. Regarding the sound effects, the effect that the game played when an item was successfully picked up was said to be ”satisfying”, ”good” and ”fun”, whereas the one played when time was running out made the participants feel ”pressured” and ”stressed”, which was the intended effect. One of the players stated that the timeout sound effect reminded them of Outer Wilds which is particularly interesting since the sound effect was inspired by it. There was also a screen fading to black effect when participants were running out of time, which made players feel the same emotions as the timeout sound effect. Two participants mentioned the game could be adapted to become a horror game since the ambience was very good.

Regarding the controls for the game, multiple participants mentioned the sensitivity being too high as a flaw. Most of these participants were part of the group that classified themselves as ”Somewhat experienced”. This could have been fixed by implementing a sensitivity slider to allow participants to customize their controls. A large portion of more experienced participants mentioned their expectation of being able to jump, since most first person games allow it. Although this could have been implemented, the decision to not allow participants to jump was made to keep controls as simple as possible.

Finally, regarding the hints, some participants mentioned not noticing the riddle and photo hints appear on screen. We attribute this to the participants being focused on searching for an item, which causes them to focus their attention on the center of the screen, potentially missing information that appears on the edges. This was the case for both the riddle hint, which appeared at the top of the screen, and the photo hint, which appeared on the bottom left of the screen. This situation could have been minimized by adding a sound effect that is specific to receiving a hint, to allow participants who are focused to get an additional cue regarding the receiving of hints. All participants thought that the footsteps were always present throughout the game, and as such, did not associate them with the game assisting them with a hint. This could be due to the footsteps being diegetically part of the game, causing the participants to assume that footsteps always exist within the game world and narrative.

5.6. Results Discussion

The general informal feedback on the game was very positive. All players said that they had fun and were very interested when the topic of the research was explained to them after the informal discussion. The game visuals and sound effects were also praised and compared to games, such as Dark Souls and Outer Wilds, which have been critically acclaimed in the industry.

Our results show that there is a large ($g > 0.8$) statistical difference between random hint selection and covert-overt hint selection on the topic of gameplay features within a 90% confidence level. The players had more difficulty when finding the Axe in a random hint selection mode than in cover-overt hint selection. In regards to game length, we found that a game lasted on average 7.1 more rounds when playing with a covert-overt hint selection system when compared to the random hint selection.

These two facts show that hints and hint selection should be considered during the game design loop, as they could potentially impact the gameplay greatly, and thus validate our conceptual framework model. Although we can not confirm that the player experience was impacted, the variability of the standard deviations of the experience questionnaire results could imply that it is, and that we are lacking in sample size. Our results show that there is a large potential for future research on adapting gameplay features by selecting different kinds of hint systems.

CHAPTER 6

Conclusions and Future Work

6.1. Conclusions

The industry has tackled the teaching of mechanics to players in multiple approaches. Some games follow a more design-heavy method, which introduces mechanics and information to players by creating carefully crafted experiences which take a long time to implement. In contrast, other games follow an approach of letting players explore freely, which could backfire when players feel lost. To find a middle ground between these two approaches, we propose the use of hints and DDA algorithms while considering player experience as a critical factor.

We presented a conceptual framework which served as a thought guide towards placing hints and hint systems as a critical component of user centered design. To apply this conceptual framework we conceptualized two potential hint selection systems and four different hints, and developed "Island", our first-person exploration game, which served as a benchmark to study our conceptual framework, and as a basis for future research.

The literature review showed a lack of studies in regards to how different hint selections systems impact player experience. To our knowledge this is the first study that specifically targets the means by which hints should be selected as a potential way to influence player experience.

From the extracted gameplay data and questionnaires of player experience, we can conclude that the addition of a hint selection strategy did not ruin the experience for the player, when compared with a lack of strategy, represented by random hint selection. Additionally, the gameplay features extracted from players using a covert-over hint selection were different than the ones observed from those using a random hint selection system. This means we can confirm that using different hint selection systems impacts gameplay. Conversely, we did not find statistical significance that a different hint selection system alters player experience. Similarly, we did not find statistical significance that a covert-overt hint selection system improves player experience. However, this does not mean that changes in how hints are selected do not impact player experience, as this is dependent on the game, and possible ”Island” did not make this impact clear.

We were able to answer two of our research questions, presented in Section 1.3. First, we can confirm that designing hint systems in an iterative and methodical fashion helped us find solutions that let us control game parameters (H1). By comparing two iterations of a hint system, we can analyze each one and understand how they influence game parameters. Secondly, we found that on overt-covert hint selection increased game length, which means that the act of altering the hint selection system impacted a parameter of our game (H3). This presents important data towards designing impactful hint systems. Designers can use a methodical workflow to find different selection methods that alter gameplay features as they feel is necessary.

We found no statistical evidence that corroborates that our methodical approach to designing hint systems helped us find solutions that improved player experience (H2), however, the higher dispersion in our covert-overt hint selection system data could indicate that there is an impact to player experience, but that we do not have the required sample size to visualize the statistical evidence of this impact. In like manner, altering the way hints are selected did not seem to present statistical evidence of an impact on player experience (H4). This is not to say that altering the way hints are selected will never impact player experience. In fact, we can postulate that different hint systems will have different impacts on game parameters, and that those changes to game parameters could have an impact on player experience. This impact might not be visible for ”Island”, with a sample size of 20 tests, 10 for each hint selection system.

6.2. Limitations and Future Work

Due to the remote setting, finding participants that were interested in the study proved difficult. A game session averaged 45 minutes to an hour in length, which for some potential playtesters was too long, which caused them not to participate. For a future study, an effort should be made to either reduce the duration of each session, or develop the game in such a way that players can play two different levels, one with each hint system, to gather more data.

To be able to perform this study in a within-subjects approach, the game should feature two different levels, each with their own map and items. This would ensure players carry as little knowledge as possible between levels. Additionally, the order in which players play each level should be randomized to minimize learning across the two different conditions. A within-subjects design would allow the study to be more statistically powerful for the same sample size, but with knowledge-based exploration games a high degree of care must be taken to ensure a minimization of the sharing of experience between levels.

An effort was made to ensure a calm environment for testing, when performing tests remotely it proved very hard to control all variables, particularly, one playtest had to be discarded because the player's computer ran at an average frames per second of approximately 20, which heavily impacted player experience. To solve this, future studies should initially guarantee playtesters can run the game at an acceptable framerate and, in order to control more variables, perform in-person tests with the same computer and ideally in the same room conditions.

To ensure an equal spread of experience levels on each hint selection system, demographic questions should be asked before playtesters participate in the experiment, and the attribution of the hint selection type performed accordingly. This would guarantee a minimization of external variability between our samples.

The conceptual framework needs further development and validation regarding the Prototype Influence Vectors. Each vector should be validated statistically which would validate the hypothesis that we should use hint systems for the top-left portion of the experience fluctuation model, when aiming to reach flow. Additionally, different hint selection models should be considered, to further validate the conceptual framework.

The results seen from the application of the conceptual framework to "Island" should be used as a benchmark to guide future studies regarding the application of the framework to other types of games, namely of different genres and on serious and educational games.

References

[1] Y. J. Sah, 'How does interactivity in a video game render people susceptible? a mediating role of concrete-level action identity in immersive media,' *Cyberpsychology: J. of Psychosocial Res. on Cyberspace*, vol. 13, no. 4, Nov. 2019. doi: 10.5817/CP2019-4-4.

[2] E. Andersen, E. O'Rourke, Y.-E. Liu, R. Snider, J. Lowdermilk, D. Truong, S. Cooper and Z. Popovic, 'The impact of tutorials on games of varying complexity,' in *Proc. SIGCHI Conf. Human Factors in Computing Systems*, ser. CHI '12, Austin, Texas, USA, 2012, pp. 59–68. doi: 10.1145/2207676.2207687.

[3] Valve Corporation, *Half-Life 2*, (2004). Accessed Sep. 19, 2021. [Online]. Available: https://store.steampowered.com/app/220/HalfLife_2/.

[4] Mobius Digital, *Outer Wilds*, (2019). Accessed Sep. 19, 2021. [Online]. Available: https://www.mobiusdigitalgames.com/outer-wilds.html.

[5] ——, *Planetary chart of the outer wilds*, (2019). Accessed Nov. 14, 2021. [Online]. Available: https://outerwilds.fandom.com/wiki/Solar_System.

[6] K. Peffers, T. Tuunanen, M. A. Rothenberger and S. Chatterjee, 'A design science research methodology for information systems research,' *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007. doi: 10.2753/MIS0742-1222240302.

[7] G. Costikyan, 'I Have No Words but I Must Design: Toward a Critical Vocabulary for Games,' in *Proc. Comput. Games Digit. Cultures Conf.*, pp. 9–33.

[8] J. Schell, *The Art of Game Design: A Book of Lenses*, 2nd ed. Morgan Kaufmann Publishers Inc., 2008.

[9] S. Kim, K. Park and T. Kihl, 'Level design of platform games using interest curves,' *Int. J. Eng. Res. Technol.*, vol. 13, no. 12, pp. 5176–5181, 2020.

[10] M. Ouellette, L. Breeding and C. Clark, 'Using applied cognitive load theory and difficulty analysis for educational game design for understanding and transference of literacy skills in

adults,' in *Proc. 14th Int. Conf. Found. of Digit. Games*, ser. FDG '19, San Luis Obispo, CA, USA: Association for Computing Machinery, 2019, pp. 1–11. doi: 10.1145/3337722.3337725. [Online]. Available: https://doi.org/10.1145/3337722.3337725.

[11] G. N. Yannakakis and J. Togelius, *Artificial Intelligence and Games*, 1st ed. Springer, 2018.

[12] I. Kotsia, S. Zafeiriou and S. Fotopoulos, 'Affective gaming: A comprehensive survey,' in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops*, Portland, OR, USA, 2013, pp. 663–670. doi: 10.1109/CVPRW.2013.100.

[13] P. Sundström, 'Exploring the affective loop,' Ph.D. dissertation, KTH Royal Inst. of Technol., Stockholm, 2005.

[14] J. Sykes and S. Brown, 'Affective gaming: Measuring emotion through the gamepad,' in *Extended Abstracts 2003 Conf. Human Factors Comput. Syst.*, 2003, pp. 732–733. doi: 10.1145/765891.765957.

[15] H. Martínez, M. Garbarino and G. Yannakakis, 'Generic physiological features as predictors of player experience,' in *Proc. 4th Int. Conf. on Affec. Comput. and Intell. Interact.*, 2011, pp. 267–276. doi: 10.1007/978-3-642-24600-5_30.

[16] A. Drachen, G. Yannakakis, L. Nacke and A. Pedersen, 'Correlation between heart rate, electrodermal activity and player experience in first-person shooter games,' in *5th ACM SIGGRAPH*, 2010, pp. 49–54. doi: 10.1145/1836135.1836143.

[17] D. Egan, S. Brennan, J. Barrett, Y. Qiao, C. Timmerer and N. Murray, 'An evaluation of heart rate and electrodermal activity as an objective QoE evaluation method for immersive virtual reality environments,' in *2016 Eighth Int. Conf. Qual. Multimedia Experience (QoMEX)*, 2016, pp. 1–6. doi: 10.1109/QoMEX.2016.7498964.

[18] L. J. Zheng, J. Mountstephens and J. Teo, 'Four-class emotion classification in virtual reality using pupillometry,' *J. of Big Data*, vol. 7, no. 1, pp. 1–9, Jul. 2020. doi: 10.1186/s40537-020-00322-9.

[19] C. Pedersen, J. Togelius and G. Yannakakis, 'Modeling player experience for content creation,' *IEEE Trans. Comput. Intell. AI Games*, vol. 2, no. 1, pp. 54–67, Apr. 2010. doi: 10.1109/TCIAIG.2010.2043950.

[20] A. Dekker and E. Champion, ‘Please biofeed the zombies: Enhancing the gameplay and display of a horror game using biofeedback,’ in *DiGRA’07-Proc. 2007 DiGRA Int. Conf.: Situated Play*, 2007, pp. 550–558. doi: 10.25917/5d1443e8af4a0.

[21] C. Liu, P. Agrawal, N. Sarkar and S. Chen, ‘Dynamic difficulty adjustment in computer games through real-time anxiety-based affective feedback,’ *Int. J. of Human-Computer Interact.*, vol. 25, no. 6, pp. 506–529, Aug. 2009. doi: 10.1080/10447310902963944.

[22] G. N. Yannakakis, H. P. Martínez and A. Jhala, ‘Towards affective camera control in games,’ *User Model. User-Adapted Interact.*, vol. 20, no. 4, pp. 313–340, Oct. 2010. doi: 10.1007/s11257-010-9078-0.

[23] M. Csikszentmihalyi, *Flow: The psychology of optimal experience*. Harper & Row, 1990.

[24] F. Massimini and M. Carli, ‘The systematic assessment of flow in daily experience,’ in *Optimal experience: Psychological studies of flow in consciousness*, M. Csikszentmihalyi and I. S. Csikszentmihalyi, Eds., Cambridge University Press, 1998, ch. 16, pp. 266–287.

[25] B. Cowley, D. Charles, M. Black and R. Hickey, ‘Toward an understanding of flow in video games,’ *Comput. Entertainment*, vol. 6, no. 2, pp. 1–27, Jul. 2008. doi: 10.1145/1371216.1371223.

[26] M. G. Jones, ‘Creating electronic learning environments: Games, flow, and the user interface,’ in *Proc. Sel. Res. Develop. Presentations Nat. Conv. Assoc. Educational Commun. Technol.*, 1998.

[27] D. Afergan, E. Peck, E. Solovey, A. Jenkins, S. Hincks, E. Brown, R. Chang and R. Jacob, ‘Dynamic difficulty using brain metrics of workload,’ in *Proc. ACM CHI 2014 Human Factors in Comput. Syst. Conf.*, 2014, pp. 3797–3806. doi: 10.1145/2556288.2557230.

[28] R. Sutoyo, D. Winata, K. Oliviani and D. M. Supriyadi, ‘Dynamic difficulty adjustment in tower defence,’ *Procedia Computer Science*, vol. 59, pp. 435–444, 2015, International Conference on Computer Science and Computational Intelligence (ICCSCI 2015). doi: 10.1016/j.procs.2015.07.563.

[29] D. Ang and A. Mitchell, ‘Comparing effects of dynamic difficulty adjustment systems on video game experience,’ in *Proc. Annual Symp. Computer-Human Interact. Play*, New York, NY, USA, 2017, pp. 317–327. doi: 10.1145/3116595.3116623.

[30] C. Liu, P. Agrawal, N. Sarkar and S. Chen, 'Dynamic difficulty adjustment in computer games through real-time anxiety-based affective feedback,' *Int. J. Human–Computer Interact.*, vol. 25, no. 6, pp. 506–529, Aug. 2009. doi: 10.1080/10447310902963944.

[31] Valve Corporation, *Left 4 Dead*, (2008). Accessed Nov. 17, 2021. [Online]. Available: https://store.steampowered.com/app/500/Left_4_Dead/.

[32] ——, *Left 4 Dead 2*, (2009). Accessed Nov. 17, 2021. [Online]. Available: https://store.steampowered.com/app/550/Left_4_Dead_2/.

[33] *The Director*, Left 4 Dead Wiki. [Online]. Available: https://left4dead.fandom.com/wiki/The_Director (visited on 17/11/2021).

[34] R. Wang, 'Designing effective hints systems,' M.S. thesis, Comp. Sci., Univ. of Illinois, Urbana, 2017.

[35] I. Arroyo, J. E. Beck, B. P. Woolf, C. R. Beal and K. Schultz, 'Macroadapting animalwatch to gender and cognitive differences with respect to hint interactivity and symbolism,' in *Proc. Int. Conf. Intell. Tutoring Syst.*, 2000, pp. 574–583. doi: 10.1007/3-540-45108-0_61.

[36] H. Wauck and W. Fu, 'A data-driven, multidimensional approach to hint design in video games,' in *Proc. 22nd Int. Conf. Intell. User Interfaces*, 2017, pp. 137–147. doi: 10.1145/3025171.3025224.

[37] E. O'Rourke, C. Ballweber and Z. Popovií, 'Hint systems may negatively impact performance in educational games,' in *Proc. first ACM Conf on Learn. @ Scale*, 2014, pp. 51–60. doi: 10.1145/2556325.2566248.

[38] K. Gilleade, A. Dix and J. Allanson, 'Affective videogames and modes of affective gaming: Assist me, challenge me, emote me,' in *Proc. DiGRA 2005: Changing Views–Worlds Play*, 2005.

[39] D. Johnson, M. J. Gardner and R. Perry, 'Validation of two game experience scales: The player experience of need satisfaction (PENS) and game experience questionnaire (GEQ),' *Int. J. Human-Computer Stud.*, vol. 118, pp. 38–46, Oct. 2018. doi: 10.1016/j.ijhcs.2018.05.003.

[40] K. M. Gerling, M. Klauser and J. Niesenhaus, 'Measuring the impact of game controllers on player experience in fps games,' in *Proc. 15th Int. Academic MindTrek Conf.: Envisioning Future Media Environ.*, 2011, pp. 83–86. doi: 10.1145/2181037.2181052.

[41] D. Johnson, L. E. Nacke and P. Wyeth, ‘All about that base: Differing player experiences in video game genres and the unique case of moba games,’ in *Proc. 33rd Annu. ACM Conf. Human Factors Comput. Syst.*, 2015, pp. 2265–2274. doi: 10.1145/2702123.2702447.

[42] D. Janßen, C. Tummel, A. Richert and I. Isenhardt, ‘Towards measuring user experience, activation and task performance in immersive virtual learning environments for students,’ in *Int. Conf. Immersive Learn.*, 2016, pp. 45–58. doi: 10.1007/978-3-319-41769-1_4.

[43] L. E. Nacke, M. N. Grimshaw and C. A. Lindley, ‘More than a feeling: Measurement of sonic user experience and psychophysiology in a first-person shooter game,’ *Interacting Comput.*, vol. 22, no. 5, pp. 336–343, Sep. 2010. doi: 10.1016/j.intcom.2010.04.005.

[44] K. Kuikkaniemi, T. Laitinen, M. Turpeinen, T. Saari, I. Kosunen and N. Ravaja, ‘The influence of implicit and explicit biofeedback in first-person shooter games,’ in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2010, pp. 859–868. doi: 10.1145/1753326.1753453.

[45] Klei Entertainment, *Don't Starve*, (2013). Accessed Nov. 24, 2021. [Online]. Available: https://www.klei.com/games/dont-starve.

[46] Unity Technologies, *Unity*, (2005). Accessed Sep. 19, 2021. [Online]. Available: https://unity.com/.

[47] Epic Games, *Unreal engine*, (1998). Accessed Nov. 10, 2021. [Online]. Available: https://www.unrealengine.com/.

[48] *Godot*, (2014). Accessed Nov. 10, 2021. [Online]. Available: https://godotengine.org/.

[49] Game Maker's Toolkit [@gamemakerstk], I asked the developers of 5,400 game jam games what they used to make their game. Here's the results, Twitter, Jul. 25, 2020. Accessed Nov. 30, 2021. Available: https://twitter.com/gamemakerstk/status/1287115875624071171.

[50] S. McConnell, *Code Complete*, 2nd ed. Microsoft Press, 2004.

[51] Z. V. Communications, *Zoom*, (2012). Accessed Sep. 19, 2021. [Online]. Available: https://zoom.us/.

[52] I. Discord, *Discord*, (2015). Accessed Sep. 19, 2021. [Online]. Available: https://discord.com/.

[53] W. IJsselsteijn, Y. de Kort and K. Poels, *The Game Experience Questionnaire*. Technische Universiteit Eindhoven, 2013.

Appendices

APPENDIX A

Experience Questionnaire

A.0.1. Demographics

First need to gather some basic information.

What gender do you most identify as?

What is your age?

Please classify your experience as a video-game player

On average, how many hours per week do you play videogames?

A.0.2. Part I

Please indicate how you felt while playing the game for each of the items, on the following scale:

0 - Not at all 1 - Slightly 2 - Moderately 3 - Fairly 4 - Extremely

I felt content

I felt skillful

I was interested in the game's story

I thought it was fun

I was fully occupied with the game

I felt happy

It gave me a bad mood

I thought about other things

I found it tiresome

I felt competent

I thought it was hard

It was aesthetically pleasing

I forgot everything around me

I felt good

I was good at it

I felt bored

I felt successful

I felt imaginative

I felt that I could explore things

I enjoyed it

I was fast at reaching the game’s targets

I felt annoyed

I felt pressured

I felt irritable

I lost track of time

I felt challenged

I found it impressive

I was deeply concentrated in the game

I felt frustrated

It felt like a rich experience

I lost connection with the outside world

I felt time pressure

I had to put a lot of effort into it

A.0.3. Part II

Please indicate how you felt after you finished playing the game for each of the items, on the following scale:

0 - Not at all 1 - Slightly 2 - Moderately 3 - Fairly 4 - Extremely

I felt revived

I felt bad

I found it hard to get back to reality

I felt guilty

It felt like a victory

I found it a waste of time

I felt energised

I felt satisfied

I felt disoriented

I felt exhausted

I felt that I could have done more useful things

I felt powerful

I felt weary

I felt regret

I felt ashamed

I felt proud

I had a sense that I had returned from a journey

A.0.4. Part III

If you can, please indicate the logic behind each of the items present in the game.

Could you explain the logic behind the locations of the axes?

Could you explain the logic behind the locations of the water bottles?

Could you explain the logic behind the locations of the maps?

Could you explain the logic behind the locations of the knives?

Could you explain the logic behind the locations of the backpacks?

Could you explain the logic behind the locations of the flashlights?

A.0.5. Part IV

Sometimes the game provides hints to players when a new rounds has started.

Did you receive any hints, and if so, which types of hints did you receive?