Procedural challenge generation guided by player choice in video games

Gonçalo Alexandre Dias e Silva Marques

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Prof. Carlos António Roque Martinho

Examination Committee

Chairperson: Prof. Nuno João Neves Mamede
Supervisor: Prof. Carlos António Roque Martinho
Member of the Committee: Prof. Maria Beatriz Duarte Pereira do Carmo

January 2021

Acknowledgments

I would like to thank my dissertation supervisor, professor Carlos Martinho, for his wisdom, tireless help and teachings throughout the whole process, without whom this would not have been possible.

I would like to thank my parents for all the support and encouragement they have given me throughout the course, as well as throughout the rest of my life.

I want to thank my girlfriend very much, who has been a constant source of encouragement and love that has really helped me complete this journey.

Lastly, I would like to thank all my colleagues that have helped me and ”fought” alongside me throughout the course, without whom completing it would not have been possible.

Abstract

This work analyzes the feasibility of using procedural generation to create challenges in a video game based on the player's choices, such as weapon choice, and to compare that approach to one based on the player's skill as well as one based on generating content randomly. Few games have attempted to procedurally generate ways for the player to progress through the game, by generating challenges that keep the player learning new ways to use the existing mechanics. This work attempts to expand upon those concepts by three different ways of tailoring content to the player. We built a video game that generates content procedurally using the 3 aforementioned approaches and had several users test 3 different versions of the game, one for each approach. Our results suggest that, in this particular implementation, players preferred playing the random approach to the approaches with content procedurally generated, which leads us to believe that more work needs to be done to better understand how player adaptation needs to be implemented to improve play experience.

Keywords

Procedural Content Generation; Adaptive Content Generation; Challenges; Player Choice; Player Skill

Resumo

Este trabalho analisa a possibilidade de usar geração procedimental de conteúdo para criar desafios num video jogo baseados nas escolhas do jogador, como a sua escolha de armas a usar, e comparar essa abordagem com uma baseada na habilidade do jogador e ainda com outra que gera conteúdo aleatoriamente. Poucos jogos tentaram gerar procedimentalmente maneiras novas de o jogador progredir pelo jogo, ao gerar desafios que permitem ao jogador continuar a aprender novas maneiras de usar as mecânicas de jogo existentes. Este trabalho vai tentar expandir sobre esses conceitos ao comparar duas maneiras diferentes de adaptar conteúdo ao jogador. Construímos um video jogo que gera conteúdo procedimentalmente de acordo com as 3 abordagens mencionadas e pedimos aos nossos utilizadores para testarem as 3 diferentes versões resultantes do nosso jogo. Os nossos resultados sugerem que, especificamente nesta implementação, os jogadores gostaram mais da versão aleatória do jogo do que das versões com conteúdo adaptado, o que nos levou a crer que mais trabalho precisa de ser feito nos modelos não-aleatórios para compreendermos melhor como a adaptação ao jogador precisa de ser implementada para melhorar a experiência de jogo.

Palavras Chave

Geração Procedimental de Conteúdo; Geração Adaptativa de Conteúdo; Desafios; Escolhas do jogador; Habilidade do Jogador;

Contents

Acronyms

PCG Procedural Content Generation

MGSV “Metal Gear Solid V”

GEQ “Game Experience Questionnaire”

1

Introduction

Contents

1.1 Motivation

In this day and age, technology users expect to have content automatically curated to them, based on their preferences and consumption history. YouTube[1] recommends videos based on to the ones we are currently watching and have watched in the past while Spotify[2] plays us songs similar to the ones we liked. This is done in order to introduce new content that might be appealing to the user, which leads to increased engagement and consumption of the product or service.

In video games, Procedural Content Generation (PCG) is often used to generate content with the goal of increasing variety and therefore replayability and engagement. However, video games are very complex and generating a vast amount of engaging content can be challenging to do in a procedural manner. In fact, very rarely, if ever, can video games procedurally generate content that is as engaging as hand-crafted content. For example: No Man's Sky[3], one of the most well known commercial games relying heavily on PCG, was criticized at launch by some players for generating content that, although vast, was also repetitive and unappealing[4].

1.2 Research Question

Previous bodies of work have attempted to test the impact of procedurally generating content based on the player's skill, and concluded that it positively impacts the gameplay experience as opposed to not adapting content at all (see"Skill-based Progression Model for Smash Time" [1, 2] by João Catarino, expanded in a later section, and "Holiday Knight: a Videogame with Skill-based Challenge Generation" [3] by João Pardal, expanded in a later section). We now wish to take the next step in that direction by comparing this approach to a different one: adapting content to the player's progression choice. This could be achieved, for instance, by changing the challenges given to the player based on the weapons they prefer to use, having the player face enemies that are more challenging with the player's preferred weapon and offering the player upgrades or variations of their preferred weapon type.

Could it be that, between adapting content to the player's skill and adapting it to the player's progression choice, one of these approaches will have a more positive impact on the player experience? If so, which one and why? In order to answer this question, we will be developing a simple game that procedurally generates challenges in 2 different ways: by adapting those challenges to the player's skill, and by adapting them to the player's choices.

[1] https://www.youtube.com (last accessed in 12/2019)
[2] https://www.spotify.com (last accessed in 12/2019)
[3] No Man's Sky (PlayStation 4, PC, 2016; Xbox One, 2018), an action-adventure survival game developed and published by Hello Games.
[4] https://en.wikipedia.org/wiki/No_Man%27s_Sky#Reception

1.3 Expected Contribution

This body of work aims at: reviewing current literature on the topic of PCG, in order to offer further comprehension on the topic; adapting an existing video game for PCG; the development of both a skill-based PCG algorithm and a choice-based PCG algorithm in a video game, based on the provided literature; measuring the player experience of users while playing both versions of the video game, one that uses the skill-based PCG algorithm and another that uses the choice-based PCG algorithm; and comparing both approaches through statistical analysis.

1.4 Organization of the Document

This thesis is is organized as follows:

- Chapter 1 describes our motivation and goal for this work.
- In chapter 2 we will be outlining work made by other authors that are related to this work, including ones that this work is based upon.
- In chapter 3 we will describe our implementation of a video game in order to answer our research question.
- In chapter 4, we will describe the procedure we followed in order to collect results that helped us answer our research question, as well as report those results.
- To conclude, Chapter 5 will describe how our results helped us answer our research question.

2

Related Work

Contents

In this chapter, we introduce some concepts relevant to PCG and challenges in games, how commercial applications approach these topics and previous work we will base our approach on.

2.1 Procedural Content Generation

According to Togelius et al [4], PCG is the algorithmic creation of game content with limited or indirect user input. PCG methods are developed and used for a number of different reasons in games, including saving development time and costs, increasing replayability and allowing for adaptive games [5,6]. Some examples of commercial video games that apply PCG are "No Man's Sky" and "Minecraft"[1]. Both of these games were commercially successful and are able to generate vast amounts of content for the player to consume, despite having been made with a very small team of developers.

In academia, PCG was also used for the following bodies of work, that we consider relevant for our approach:

- In "Multi-dimensional Player Skill Progression Modelling for Procedural Content Generation" [7], Bicho and Martinho review previous skill-based PCG work and use it to generate content for a game that adapts to the player's skill, guided by a multi-dimensional skill model. The video game "Go GO Hexahedron" was developed in order to implement this skill progression model and test their hypothesis with users: whether modelling the evolution of multiple dimensions of a same challenge while the game is played helps creating a better game experience for the player. Their results concluded, among other things, that players have consistent and specific preferences regarding how difficulty should evolve over the course of a game, which should be taken into account when designing the game's progression.

- In "Polymorph: A Model for Dynamic Level Generation" [8], Smith et al. used an existing 2D platformer level generator based on player action-rhythm, which receives a set of player actions and generates levels that can be completed with that set of actions (Fig. 2.1), to manipulate a continuous level as the player progresses through it while taking into account the player's performance. This causes different players playing through the same level, which might start similarly, to not only experience different variations of that level, but variations that take into account their performance.

- In "Skill-Based Progression Model for Smash Time" Catarino and Martinho developed a skill-based progression model that attempts to address the problem of players with different skill levels tackling game content at the same pace, when in reality they might feel a different level of difficulty while facing the exact same challenges in a game. The hypothesis tested was whether generating

[1] Minecraft (Windows, macOS, Linux, Android, iOS, 2011; XBbox 360, 2012; PlayStation 3, 2013; PlayStation 4, PlayStation Vita, Xbox One, 2014; Wii U, 2015; New 3DS, Switch, 2017), a survival adventure game developed and published by Mojang, later bought by Microsoft.

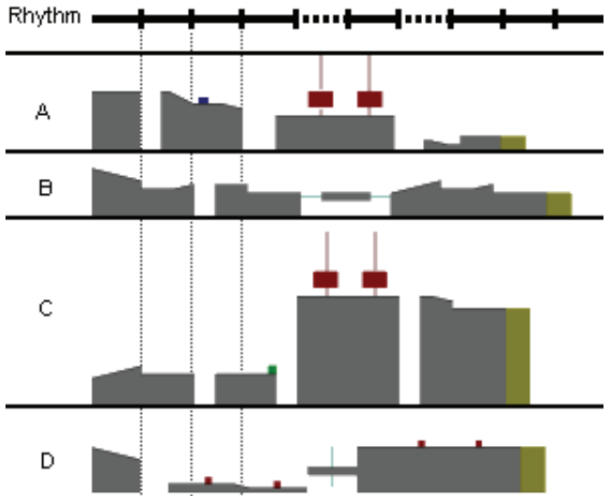

Figure 2.1: Examples of levels generated with the player action-rhythm level generator.

challenges more suitable to a player's skill level would create a more engaging experience. In order to test this hypothesis, an existing video game, "Smash Time", was modified. After testing with users, they concluded that a skill-based progression model was able to increase not only the duration of play sessions, but also the number of play sessions per player.

- In "Holiday Knight: a Videogame with Skill-Based Challenge Generation" Pardal and Martinho built upon Catarino's work on skill-based challenge generation by testing the same hypothesis on a different type of game. They concluded that, although the number of length of play sessions remained the same, players felt a more homogeneous level of challenge when playing a version of the game that used the aforementioned skill-based progression model.

Adaptive Procedural Content Generation is a subtype of PCG that focuses on adapting the content generation to the player's behavior [4]. This is the main approach we have used for this work.

2.2 Challenge-Reward Dichotomy

When playing a video game, it is a common expectation that the player will receive a reward after completing a challenge. According to Wang and Sun [9], that reward can come in many forms, such as increased score and experience points, new in-game content, progression of the story (possibly through cut-scenes), or in-game items.

Wang and Sun also note the following:

- Rewards in video games, although mostly extrinsic[2], can provide intrinsic rewards[3] to the player. If an extrinsic reward (e.g. an achievement, an in-game item) is given to the player for mastering a specific mechanic or completing a level in a different way, it will motivate the player to learn more about the game's depth and possibilities, contributing to an intrinsically rewarding experience.
- There should be a balance between the effort/time spent trying to receive a reward and the value of the reward. If a player spends a long time trying to get a reward and then finally receives it, only to find out that it is much less valuable than what was expected, the player will be frustrated, which will increase the chances that he abandons the play session and, potentially, the game entirely.

Our takeaways from these points are that, with Adaptive PCG, we can achieve the following:

- Adapt content to the player's choices by trying to guide the player towards experimenting with different gameplay strategies, in order to create a more rewarding experience for the player.
- Measure the player's effort toward achieving an award: matching a reward's value with the player's effort to receive that reward is important, in order to make sure the player doesn't feel frustrated.
- Generate rewards tailored for each player's preferences: given that different players will have different preferences, some games run the risk of rewarding the player with an item that, although valuable, has little value to some players, given their preferences (e.g. a player whose character is of class Warrior receiving a very powerful staff, directed towards the Wizard class, isn't very relevant). Adaptive PCG can help generate rewards tailored for each player's preferences to make sure this risk is minimized.

2.3 Adaptive Procedural Content Generation in Games

We will now take a look at several games, most of which are well-known and were commercially successful, that have adopted either a skill-based or choice-based PCG approach.

Adapting content to the player can be a challenging task, especially when that content is hand-made and incurs significant costs. Regardless, several games have employed such an approach.

Here are a few examples of games that adapt content to the player's choices:

- In "Metal Gear Solid V" (MGSV)[4], you control a secret agent trying to infiltrate enemy facilities. You have access to lethal and non-lethal weapons and gadgets that you can use to dispose of or dis-

[2] Extrinsic reward: tangible or physically given to you for accomplishing something. It is a tangible recognition of one's endeavor: a certificate of accomplishment, a trophy or medal for winning a race, a monetary reward for doing your job. Because extrinsic rewards are tangible, they are usually given to the person doing the activity; as such, they are typically not from within the person.

[3] Intrinsic reward: an intangible award of recognition, a sense of achievement, or a conscious satisfaction: the knowledge that you did something right, or you helped someone and made their day better. Because intrinsic rewards are intangible, they usually arise from within the person who is doing the activity or behavior.

[4] Metal Gear Solid V: The Phantom Pain (PlayStation 3, PlayStation 4, Xbox 360, XBox One, PC, 2015), a stealth-action game developed by Kojima Productions and published by Konami.

tract enemies. The game reacts to the player's play-style by modifying the challenges accordingly, e.g.:

  – If the player has a preference towards infiltrating bases during the night, the game will add more enemies with night-vision goggles, who will have an improved vision cone compared to other enemies;
  – If the player tries to dispose of enemies by shooting at them in the head, most enemies will start wearing helmets which nullify the first shot of the player.

- In Phoenix Point[5], the player controls a small group of soldiers trying to defeat alien enemies. These enemies will have random mutations that change the way they behave and attack. Mutations that do well against the player's squad will be saved for future attacks and ones that don't will stop being used[6].

Both these games take the player's choices as input, and use them to make the game more challenging. In "Multi-dimensional Player Skill Progression Modelling for Procedural Content Generation" [7], Bicho's results suggested that allowing the player to use 2 different strategies while making the preferred one more difficult to use over time doesn't necessarily incentivize the player to change their strategy. Our adaptive content generation algorithm is based on an approach similar to MGSV's choice-based approach, where we adapted the challenges to the player's equipment choices, in order to make the game more or less challenging as needed.

Examples of commercial games that adapt content to the player's skill level are:

- "Left 4 Dead"[7], a co-op zombie survival game, features a dynamic content generation system that spawns different quantities of enemies at different locations based on the players' individual and collective performance.
- "Crash Bandicoot"[8], a 3rd person platformer game, slows down obstacles and offers extra hit points and checkpoints if the player isn't performing well.
- "Resident Evil 4"[9], a third-person survival horror shooter, grades the player's performance throughout the game and adapts the enemies' aggression and damage dealt to the player accordingly (i.e. if the player performs poorly, the game lessens the damage dealt to and the overall aggression toward the player, and does the opposite when the player performs well).

[5] Phoenix Point (PC, macOS, 2019) a turn-based strategy game developed and published by Snapshot Games.
[6] https://www.pcgamesn.com/phoenix-point/phoenix-point-ai
[7] Left 4 Dead (PC, OS X, Xbox 360, 2008), a co-op zombie survival game developed and published by Valve.
[8] Crash Bandicoot (PlayStation 1, 1996), a 3rd person platformer game developed by Naughty Dog and published by Sony Computer Entertainment.
[9] Resident Evil 4 (GameCube, PlayStation 2, 2005; Windows, Wii, 2007; iOS, 2009; PlayStation 3, Xbox 360, 2011; Android, 2013; PlayStation 4, Xbox One, 2016; Switch, 2019) a third-person survival horror shooter game developed and published by Capcom.

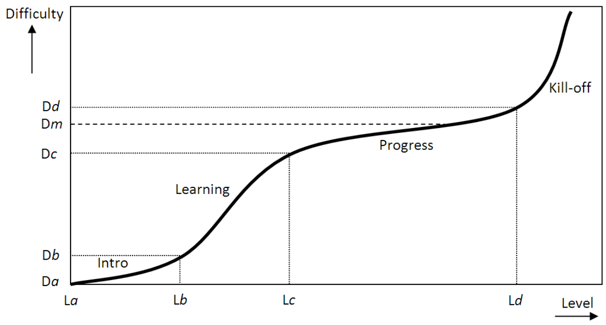

Figure 2.2: An example of a difficulty curve

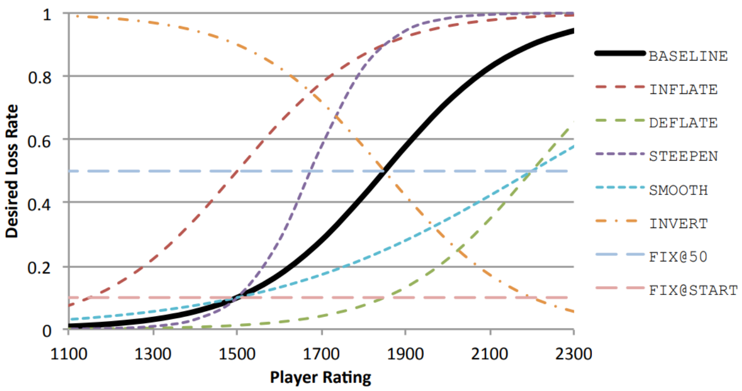

Figure 2.3: The curves used to analyze how they impact the gameplay experience.

2.4 Difficulty in games

This work uses a player performance curve, which works similarly as a difficulty curve, in order to specify the desired level of challenge to the player (this distinction is explained in section 3.2.1). Difficulty curves are a way to express how a game's difficulty varies over its lifetime (Fig. 2.2).

In work done by Aponte et al. [10], a game's difficulty curve is defined as the gradual change of the player's probability to lose over time. Additionally, they propose a way to identify which of the main abilities/actions that the player can use in a game will influence the difficulty more, by measuring how likely an (artificial) player is to win a challenge with different levels of mastery of each of these actions.

In Sarkar and Cooper's [11] work related to difficulty curves, they attempted to modify a game's existing difficulty curve by replacing it with simple cubed functions (Fig. 2.3) and concluded that modifying a game's difficulty curve can significantly impact the gameplay experience and, depending on which curve you use and which game you use them in, increase player engagement.

Horn et al. [12] attempted to model player learning in a puzzle game through AI-assisted analysis: a set of 4 artificial players was created, each using different algorithms while attempting to solve a set of puzzles. Those puzzles' difficulties were then classified by marking which artificial players were able to solve them, where puzzles that could only be solved by the most complex algorithm were labeled the most difficult, and puzzles that could be solved by all the algorithms were labeled the easiest. The hypothesis tested was whether these artificial players captured the difficulty that human players face when playing through this puzzle game. Indeed they did: it was concluded that classifying the puzzles' difficulties by labelling which algorithms could solve them correlated with the difficulty that players felt when solving those same puzzles (the more failed attempts in a puzzle, the greater the puzzle's difficulty for the players). This shows that classifying a game's difficulty through artificial players (at least in specific types of games) is indeed feasible.

Allart et al. [13] attempted to analyze how the difficulty of a game influences the player's motivation over time. They concluded that, as they progress through the game, players tend to prefer higher levels of difficulty (the player's probability to lose), ideally under a value of 50%. However, in one of the two games tested, players generally wanted a stable difficulty level for the first few hours of the game, which the authors claim is due to self-efficacy theory (at the start of the game, frequent failure can harm the player's belief of success, so he'd rather not lose very often during that time). Given that both games they tested had different ways of attaining success (one of the games relies mostly on the player's skill and the other mostly on the avatar's strength), this desire of a more stable difficulty during the first few hours of the game was only noted in the game which relied more on the player's skill: failing frequently in a game that relies on player skill is much more impactful on motivation that failing in a game that relies mostly on the strength of the player's avatar.

2.5 Measuring Game Experience

In order to answer our research question, which PCG approach leads to a more pleasant player experience: content generation based on the player's skill or the player's choice, we had to measure the gameplay experience of the users playing-testing our game. The most common approach is to ask users to fill out a questionnaire regarding their overall play experience after completing the activity. We researched relevant questionnaires for the players to complete during the testing phase, and selected the "Game Experience Questionnaire" (GEQ):

The GEQ [14] has been widely used in similar works that aim to measure the player's experience [15–17]. The dimensions it measures are Immersion, Flow, Competence, Positive Affect, Negative Affect, Tension and Challenge. We have used all but the Immersion dimension for our questionnaire:

- Flow, Challenge, Competence - the concept of flow, coined by psychologist M. Csikszentmihalyi [18], is a core concept in the realm of video games. Flow is a mental state of heightened focus and immersion in an activity, a sense that one's skills are adequate to cope with the challenges at hand. Concentration is so intense that there is no attention left to think about anything irrelevant, self-consciousness disappears momentarily and the sense of time becomes distorted. Flow is a central concept to whether the gameplay experience is pleasant to the player. The two most important dimensions of flow are Challenge and Skill (which in this questionnaire are equivalent to the Challenge and Competence dimensions respectively). We have therefore used the Dimensions of Flow, Challenge and Competence from the GEQ.

- Tension - this dimension evaluates whether the player is feeling frustrated or annoyed during the play experience. It directly contributes to and might be a negative influence on the overall experience, which makes it important for us to measure it.

- Positive/Negative Affect - this dimension is designed to evaluate the positive and negative emotions felt by the player throughout the experience, which will be useful in this work.

In conclusion, the main dimensions we have measured are: Flow, Challenge, Competence, Positive Affect, Negative Affect and Tension.

In order to support the evaluation of our hypothesis, we asked our users to fill out a questionnaire, comprised of questions from the GEQ.

2.6 Previous skill-based adaptation work

This section describes previous bodies of work in the realm of skill-based procedural generation, namely Catarino's [1, 2] and Pardal's [3] work on skill-based progression modelling, described briefly in the following sections 2.6.1 and 2.6.2. This work will be based on concepts and practical work developed by these authors, which we will be describing in greater detail.

2.6.1 Skill-Based Progression Model for Smash Time

Catarino and Martinho [1, 2] developed a skill-based progression model that attempts to address the problem of players with different skill levels tackling game content at the same pace, when in reality they might feel a different level of difficulty while facing the exact same challenges in a game. The hypothesis tested was whether generating challenges more suitable to a player's skill level would create a more engaging experience, which would increase play-time and enjoyment.

In order to test the hypothesis, an existing mobile game, "Smash Time" (Fig. 2.4), was modified. Smash Time has fast gameplay mechanics that result from the combination of elements from classic games like "Whack-a-Mole"[10] and "Space Invaders"[11], mixed with puzzle mechanics. Smash Time's

[10] Whack-a-Mole, a popular arcade game that involves hitting plastic moles with a large, soft mallet, made in 1976 by Creative Engineering, Inc.

[11] Space Invaders (Arcade, Atari 2600, Atari 5200, Atari 8-bit, MSX, 1978), one of the first side-scrolling shooter games, developed and published by Taito.

Figure 2.4: A screenshot taken from Smash Time

characters are enemies, animals and heroes, that coexist in the same world. Enemies enter the screen from the top and both sides and try to attack the hero, at the bottom of the screen, and the animals that are trying to escape from them. The player's goal is to smash the enemies and clear all the incoming waves. To smash an enemy, the player must tap it with his finger.

Smash Time's modified game cycle, which incorporates the progression model developed by Catarino and Martinho (Fig. 2.5), is comprised of the following steps:

1. Generate a new challenge (game content) to present to the player using:
   - the Player Performance Predictive System from the Player Performance Model;
   - the Content Variety Data from the Content Variety Model;
   - the Challenge Library.
2. Register the player response dealing with the obstacles that compose the generated challenge;
3. Analyze the player performance through the recorded player actions relative to the generated challenge;
4. Register the player performance data in the Player Performance Model;
5. Predict the player's performance in the Player Performance Predictive System;
6. Register the challenge variety data in the Content Variety Model;
7. Back to step 1.

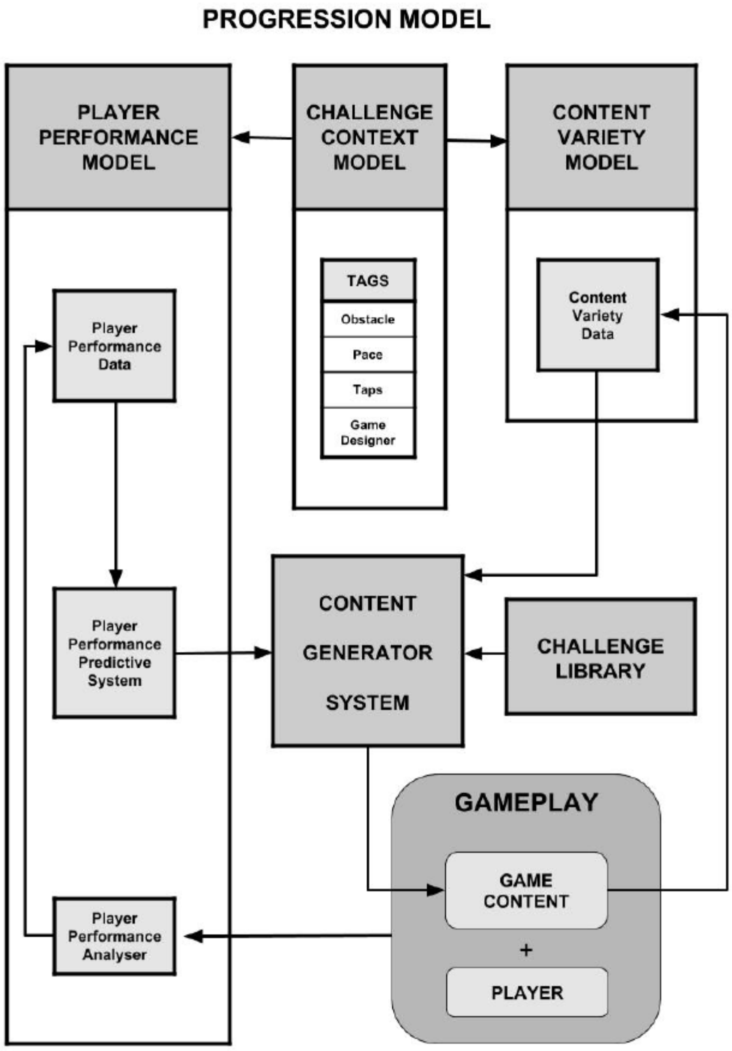

Figure 2.5: Progression model adapted from Catarino's work.

The progression model uses tags, assigned both by the game designer and the progression model, to describe and classify the challenges and obstacles of the game by assigning them a performance value and a variety value. The Content Generation Model uses both the Player Performance Model and the Content Variety Model to generate engaging and challenging game content: the Player Performance Model uses a Player Performance Curve, which indicates the evolution of the desired performance of the player throughout the game session and the Content Variety Model uses a Content Variety Curve, which indicates the evolution of the desired variety of the content throughout the game session.

In order to generate a challenge to the player, the progression model will randomly generate 50 rooms, which contain enemies taken from the Challenge Library, and calculate each of their utility value (i.e. how appropriate they are for the player at the current point in his play session) using a heuristic function that combines a room's predicted player performance value with its content variety value. The best player performance value is the one closest to the current value of the Player Performance Curve, and the best content variety value is the one closest to the current value of the Content Variety Curve.

Catarino conducted several informal preliminary testing sessions with users, where they played the version of the game that uses the aforementioned progression model, in order to produce bootstrap values for the player performance that he could use in the final evaluation.

The final evaluation of Catarino's hypothesis was made by having 2 groups of 16 players, one that tried the normal version of Smash Time, which increased difficulty equally across player skill levels, and one that tried the version with the aforementioned skill-based model, which would adapt the challenges to the player's skill level. The conclusion was that a skill-based progression model was able to increase not only the duration of play sessions, but also the number of play sessions per player. It was also concluded that this model has the potential to increase player immersion and, consequently, create more engaging gameplay experiences. All these positively contribute to the extension of a game's overall lifetime and revenue.

2.6.2 Skill-Based Challenge Generation in Holiday Knight

Pardal and Martinho [3] built upon Catarino and Martinho's work on skill-based challenge generation by testing Catarino's hypothesis, how impactful it is to adapt the game's challenges to the player's skill as opposed to not adapting content at all, on a different style of game.

In order to test Catarino's hypothesis with a different type of game, a new video game, Holiday Knight (Fig. 2.6), was developed. Holiday Knight is heavily inspired by games like "Enter the Gungeon"[12] and "Binding of Isaac"[13] where the main character fights enemies by shooting at them while at the same time

[12] Enter The Gungeon (Windows, OS X, Linux, PlayStation 4, 2016; Xbox One, Switch, 2017), a bullet-hell roguelike game developed by Dodge Roll and published by Devolver Digital.

[13] Binding Of Isaac (Windows, OS X, Linux, 2011), an action roguelike game developed and published by Edmund McMillen and Florian Himsl.

Figure 2.6: A screenshot taken from Holiday Knight

dodging their attacks (also known as “Shoot’em Up”[14] Games).

In Holiday Knight, the player will go from room to room, defeating all the enemies in each room. When all the enemies have been defeated, the player can progress to the next room and repeat the process. When all the rooms have been cleared of enemies, the game ends. If the player dies during the process, the game also ends.

In order to test the aforementioned hypothesis, Pardal had a preliminary evaluation with the objective of gathering bootstrap values for each tag’s performance from players, relating to the obstacles of the game, to be used in the model as default values. He then had a group of users play an adaptive version of the game, which would update the player’s performance values according to his performance, and another group of users play a static version of the game, which wouldn’t update the player’s performance values and instead use the default values calculated in the preliminary evaluation. Both these versions would adapt the challenges to the performance values, but only one would update those performance values throughout the play session. It was concluded that, while the adaptive version showed no significant differences over the static version regarding the duration of the play session or the number of play sessions, players who tried the adaptive version reported feeling a more homogeneous level of challenge than the ones who tried the static version.

This work was built upon a modified version of Holiday Knight.

[14] Shoot’em Up is a video game sub-genre of the shooter genre, where the player battles a number of enemies by shooting at them and dodging their fire, relying primarily on the player’s reaction time.

2.7 Concluding remarks

Some commercial AAA[15] games have found ways of effectively adapting content to the player using both skill-based content generation and choice-based content generation, which leads us to believe that they are viable for mass market games.

Previous work from Catarino, Pardal and Martinho have concluded that adapting content to the player's skill level, as opposed to not adapting content at all, can positively impact the player's experience.

Given that both the skill-based and choice-based approach can be further explored, we intended to compare how effective they are within the context of a same game, a modified version of "Holiday Knight" tailored for this purpose.

[15] AAA is a term used to refer to games of high-caliber. They usually have budgets in the multiple millions of dollars and are made with teams of many people, sometimes over one thousand.

3

Solution

Contents

What follows is a description of the solution used to implement a content generation model based on the player's choice, which will be compared to the previous approach based on the player's skill. It should be noted that, given the fact that the game to be used for testing has changed compared to the existing game, "Holiday Knight", the existing skill-based progression model had to be adapted in order to support the newly proposed gameplay mechanics and minimize the differences between both content generation models.

3.1 Approach

This work's research question, which PCG approach leads to a more pleasant player experience: content generation based on the player's skill or the player's choice, was tested using a modified version of the "Holiday Knight" video game described previously due to the fact that its gameplay and progression model have already been developed and tested in such a way that it can be built upon. The original version of the game was made from scratch using a 3rd party game engine[1] called Unity[2], which we have also used in order to extend and modify "Holiday Knight".

The modified version of Holiday Knight has the following gameplay progression:

1. Similarly to the original version, the player must go from room to room, defeating all the enemies in each room, in order to progress to the next. This process will repeat until the player either reaches the last room or loses. In this version of the game, there are 15 rooms in total and when the player defeats all the enemies in the last room they win that playthrough.

2. The player can pick up weapons, which can be either assault rifles, shotguns or sniper rifles, that can be switched with other weapons that the player finds. Assault rifles have a 3-round burst fire and "2D Recoil" (the weapon rotates around the player as he shoots with it), shotguns fire two rounds at once in 2 different directions and sniper rifles are precise but have a slower rate of fire. Each weapon has an element associated with it that will cause different effects when hitting enemies, based on which element it is:

   - Fire - The enemy causes a burst of fire that damages enemies around it
   - Ice - The enemy freezes (i.e. doesn't move or attack) for a few seconds
   - Poison - The enemy's bullets will travel at half the speed

   These effects are applied to the enemies once they have suffered a certain amount of damage from the player's weapon.

[1] Game engines are software programs that allow users to combine several multimedia files, such as 3D models, images and audio files, using source code, in order to create a video game.

[2] https://unity.com/

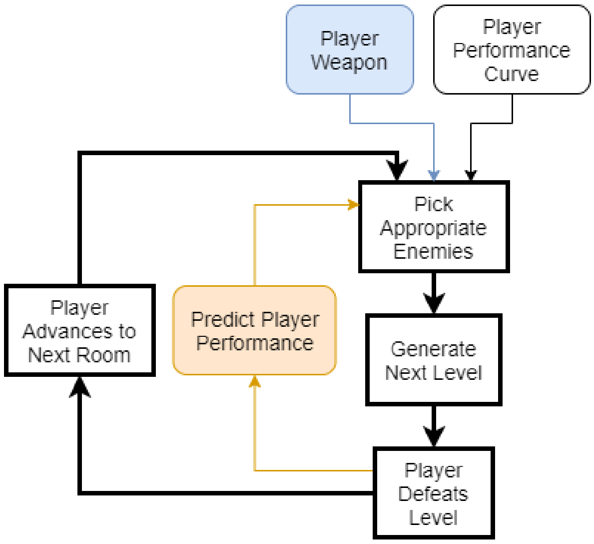

Figure 3.1: A diagram of both the skill-based and choice-based progression model.

3. Enemies have different levels of resistance against the effects suffered from the player's weapons. Some take less damage against fire, some already move slowly and hence are less affected by being frozen and some remain poisoned for a smaller period of time.

4. Each room also has a weapon placed in the corner, in order to allow the player to switch weapons throughout the playthrough.

3.2 Architecture

In order to support our modifications to the "Holiday Knight" game, we had to adapt the existing skill-based progression model. This section will outline the differences between that model and the new choice-based progression model. The conceptual differences between both models are shown in Fig 3.1, where white boxes are the common components between them, orange boxes are components related only to the skill-based progression model and blue boxes are components related only to the choice-based progression model.

The adapted skill-based model's event flow will work as following:

1. Using the Player Performance Curve and the Player Performance Prediction, several levels (rooms) will be randomly generated, and only the room with the best utility will be chosen as the player's next challenge, based on its enemies (further explained in section 3.2.1);

2. When the player defeats all the enemies in that room, the player's performance will be analyzed and his performance for the next level will be predicted based on previous values;

3. When the player advances to the next room, the cycle goes back to step 1;

The new choice-based model's event flow will work as following:

1. Using the Player Performance Curve and the player's current weapon, several levels (rooms) will be randomly generated, and only the room with the best utility will be chosen as the player's next challenge, based on its enemies (further explained in section 3.2.1);
2. When the player advances to the next room, the cycle goes back to step 1;

3.2.1 Progression models

In the following paragraphs, we will use the term ”utility” as a way to describe how useful a certain component of the model is relative to the Player Performance curve's value in the current level. The enemies' utilities will mean how much their difficulty matches the current value of the Player Performance Curve. Performance is inversely proportional to difficulty, which means that a lower performance value will imply a higher difficulty and a higher performance value will imply a lower difficulty.

Both models will take the Player Performance Curve as input. This curve's purpose is to allow the game designer to specify the desired player performance throughout the levels of the game. It works similarly to a Difficulty Curve, but reversed: when the difficulty increases, the value on a Difficulty curve will go up, while the value on a Player Performance curve will go down.

The adapted skill-based model takes the player's performance into account, measured by the amount of time the player takes to defeat all the enemies in the room (the time that the player takes to defeat each enemy only starts counting when the player damages that enemy for the first time). This performance value is then used to predict the player's performance in the next level, and pick the most appropriate enemies according to that predicted performance value. It is important to note that the player performance values are guided by tags defined by the game designer (e.g. tags that define that enemy type like “Swampy”, “IceZombie”, etc.) and tags updated by the progression model (e.g. informative tags like “ShotsToKill”, etc.). All enemies have tags like these, in order to guide the procedural generation of content throughout the game.

Unlike the skill-based model, the choice-based model will take into account **not** the player's performance, but the player's equipment instead, which in the case of “Holiday Knight” boils down simply to the player's current weapon. As mentioned above, the player is able to pick up weapons, and their elemental modifiers will be taken into account when generating enemies for each room, as well as the type of weapon the player is currently holding.

Each time the player defeats all the enemies in a room and advances to the next room, the game randomly generates many rooms and picks the best one based on its utility (the higher the utility, the more appropriate that room is for the player). The choice-based progression model uses the player's equipment to pick the best room. Given that a room's content is comprised solely of enemies, its utility can be boiled down to its enemies' utilities, which are calculated by matching an enemy's difficulty value against the current value of the Player Performance Curve.

Because some rooms may have a weapon inside them (which isn't the weapon the player is carrying), in that case, the enemies' utilities take into account the average difficulties between the player's current weapon and the weapon that is already inside the room.

The enemies' difficulty values are based on the following:

- Player's weapon element - as mentioned above, all weapons have an element associated with them which dictate the type of effects the enemy will suffer. All enemies have different resistances to these effects:
  - Fire - Some enemies take less damage against fire while others take more damage.
  - Ice - Some enemies move slowly and also remain still for longer periods of time and are hence less affected by being frozen. Enemies that move very frequently are more affected by being frozen.
  - Poison - Some enemies are more resistant to poison and hence remain poisoned for smaller periods of time, while others are less resistant and remain poisoned for longer periods of time.
- Player's weapon type - enemies take different amounts of damage depending on their size and the weapon damaging them. Enemies with a large size take more damage from some weapons while enemies with a smaller size take more damage from different weapons. In table 3.1, percentages refer to how much of the normal damage those enemies take in each situation (i.e. 130% means that enemy takes 130% of the normal amount of damage).

Table 3.1: Enemies' damage taken by weapons

	Sniper	Assault Rifle	Shotgun
Small Enemies	130%	100%	70%
Medium Enemies	100/70%	130%	100/70%
Large Enemies	70%	100%	130%

- Quantity - each room has between 2 and 4 enemies, and the more enemies a room has, the harder that room will be to complete.

As was mentioned in the previous paragraph, enemies with a high difficulty value are only chosen when the Player Performance curve value is low while enemies with a low difficulty value are only chosen when the Player Performance curve value is high. For instance: at the beginning of the game, the Player Performance curve's value will usually be high (in our approach, it was), which means that the game will choose enemies that are more fragile against the player's weapon element and easier to defeat with the player's weapon type. However, as the player's desired performance goes down, it will eventually reach a low enough value where the game will spawn enemies in greater quantity that will be more resistant to the player's weapon element and type.

In order to have a reference point to compare the skill-based and choice-based progression models against, we added a 3rd, random progression model. This progression model will choose both the type and the number of enemies (between 2 and 4) in each room completely randomly, without taking into account the player's performance or their equipment.

4

Evaluation

Contents

In order to answer our research question, which PCG approach leads to a more pleasant player experience: content generation based on the player's skill or the player's choice, we implemented the two aforementioned progression models (skill-based and choice-based) in the new version of Holiday Knight and then had players test those versions of the game.

In this chapter, we will describe all the stages of evaluation that we did for this work. Before we began the Final Evaluation, we did two Preliminary Evaluations. Their description will be in the following sections.

4.1 Preliminary Evaluations

The first preliminary evaluation was done to test the overall structure of the activity, where we used the skill-based and choice-based progression models. This evaluation was made with a total of 4 users. We intended to know the following:

- Whether the users that did the activity had any questions about it
- That there was nothing stopping them from starting or concluding the activity
- That they enjoyed the game enough to be engaged with it
- Whether there were distinct signs of their experience that differed along with the progression model assigned to them.

Because the gameplay experience would be similar to that of the original game, we decided to keep the bootstrap values for the enemies' performance in the skill-based progression model as-is, which were taken from Pardal's original preliminary evaluation.

Users started out by answering a short survey, where they were asked a few questions related to demographics, such as their age and how often they play video games. At the beginning of the survey, they were also told that the only difference between the two progression models they tested were how the enemies were chosen for each room. The full survey can be found in appendix A. After answering these questions, they were asked to launch the game.

When players launched the game, a Player ID was automatically attributed, as well as which of the two progression models (choice-based or skill-based) they would be assigned first. In order to manage both these properties, we set up a Node.js server on Microsoft's Azure cloud platform, which would keep track of the last issued Player ID and progression model. Whenever a new Player ID was requested, which was done automatically when the game started, in the form of an HTTP Get request, the server's internal Player ID would be incremented and it would respond to the request with that new value. As players would request Player IDs, this number would be incremented in order to make sure that all Player

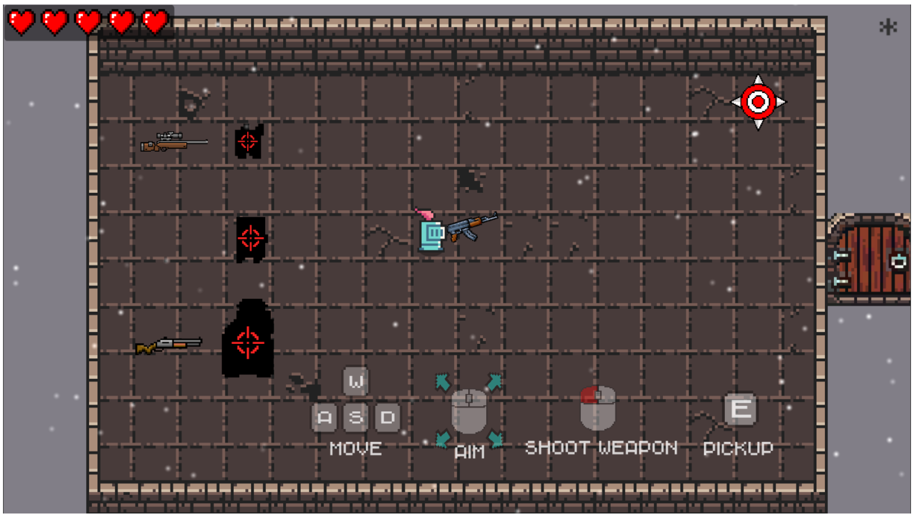

Figure 4.1: The First Tutorial Room

IDs were unique. The assignment of the progression model would work similarly, incrementing the value of the progression model (0 for skill-based and 1 for choice-based) every time a new progression model was requested, but cycling back to 0 once the number would surpass the last progression model, 1. The request for the progression model was done right after the request for the Player ID, also in the form of an HTTP Get request.

Each player went through a short tutorial where they were able to experiment with the game's mechanics at their own pace. The tutorial was comprised of two rooms: the first room, shown in Fig. 4.1, would simply have the three different weapons without any elemental effects, one of which the players needed to pick up and use to shoot a target and advance to the second room; in the second room, shown in Fig. 4.2, they could choose between one of the three weapon types (assault rifle (A), sniper rifle (B) and shotgun (C)) and switch between different elemental effects (fire (1), ice (2) and poison (3)). They could also shoot 3 different targets in the level, which would allow them to either spawn a random enemy (Z), spawn an enemy of the same type as the last enemy (Y) or start their first playthrough (X).

Each player had to go through 2 playthroughs of the game, one using the skill-based progression model and another using the choice-based progression model. Each playthrough would consist of 15 rooms. Players would have to complete a minimum of 8 rooms in total, for each of the progression models, possibly over more than 1 playthrough, in order to progress to their next playthrough. They could also do as many playthroughs as they wanted.

Once they completed their first playthrough, the game would ask them to go back to the survey, where they would answer some questions related to their gameplay experience. These questions were the ones from the GEQ's Core section, and can be found in appendix A.

Once they answered these questions, players would go back to the game and start their second

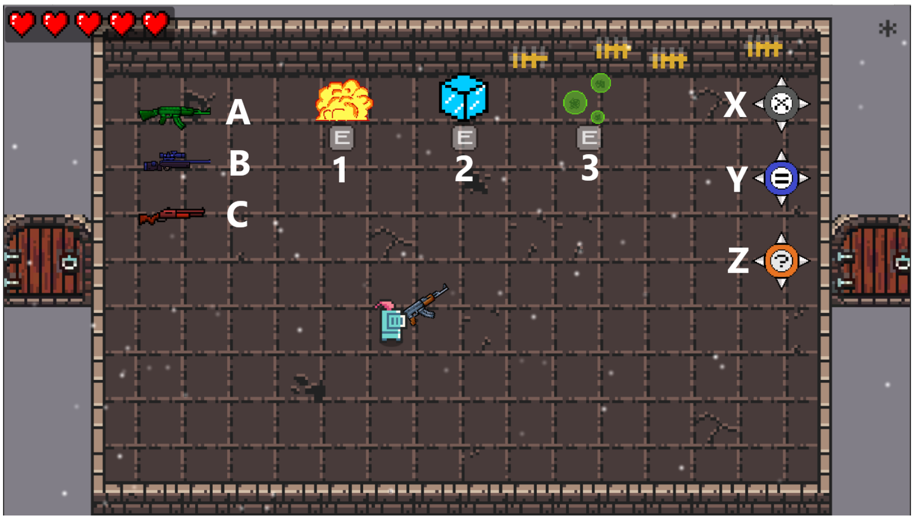

Figure 4.2: The Second Tutorial Room

playthrough, whose only difference was the progression model used to generate the rooms in the playthrough.

After they completed their second playthrough, they would again be asked to go back to the survey in order to answer the same questions from GEQ's Core section, this time about their second playthrough. After answering these questions, they would be given the option to provide us their email in order for us to contact them regarding the activity, if needed. And that concluded the activity.

The questions present in the GEQ are presented as a Likert scale with 5 different values ranging from "Not at all" to "Extremely", which are then converted to a number between 0 ("Not at all") and 4 ("Extremely"). By adding the numbers of all the questions related to each dimension, we then have a score associated with each dimension for each person that answered the survey.

These are the stats that we tracked during each playthrough for this study, which were sent to a Node.js server after the player completed his playthrough:

- The specific types of enemies defeated, in which room they were defeated and how long it took the player to defeat them;
- Which progression model was used in that playthrough;
- How much time the player spent in the tutorial room, before starting his first playthrough;
- How much time the player spent in his playthrough;
- How many rooms the player was able to complete in each of his playthroughs;
- The weapon the player used to defeat the last enemy in each room;
- How much health the player had left after completing each room.

These stats were tracked both in the preliminary and final evaluations. We will be reporting the mean and median of some of these stats in the following section, in the context of the final evaluation. In order for us to receive these stats, the game would automatically send the gathered stats to the aforementioned Node.js server, in the form of an HTTP Post request, after the player finished his playthrough.

After doing this activity, we realized that it would be best if we also had a third progression model, that would simply generate enemies randomly, to serve as a baseline. This caused us to do several modifications to the game used for the activity:

- Because we decided to have a 3rd progression model, we thought the best alternative would be not to have each player play all 3 progression models, but rather to have each player play only 1 of them. With this in mind, players would now play only 1 of the progression models, which would be picked sequentially every time a new player would start playing the game.
- Because players would now only do 1 playthrough, we raised the number of rooms to beat the playthrough to 20 and also the minimum rooms needed in order to finish the activity, in total, to 20. Based on the preliminary evaluation, this would be a good balance between allowing the player to have enough time to experience the game as well as not having an activity that would be too long and potentially become annoying.

After making these modifications, we began the second preliminary evaluation, in which participated a total of 6 players. After doing so, we had mixed results about which progression model players had a more positive experience with, however, we believed that there was mostly a preference toward one of the non-random progression models compared to the random progression model: the lowest value for the competence, flow, challenge and positive affect dimensions, as well as the highest value for the tension/annoyance and negative affect dimensions, were all reported by either one of the two players that had played the random progression model.

Table 4.1: Gameplay Experience statistics for players who were assigned the skill-based progression model in the second preliminary evaluation.

	Competence	Flow	Tension/Anoyance	Challenge	Negative Affect	Positive Affect
Median	13	16.5	2	12.5	0	10
Maximum	15	17	4	13	0	11
Minimum	11	16	0	12	0	9

Table 4.2: Gameplay Experience statistics for players who were assigned the choice-based progression model in the second preliminary evaluation.

	Competence	Flow	Tension/Anoyance	Challenge	Negative Affect	Positive Affect
Median	9	13	1.5	10.5	0.5	8
Maximum	12	15	3	13	1	8
Minimum	6	11	0	8	0	8

Table 4.3: Gameplay Experience statistics for players who were assigned the random progression model in the second preliminary evaluation.

	Competence	Flow	Tension/Anoyance	Challenge	Negative Affect	Positive Affect
Median	8.5	13	4.5	6	1.5	6.5
Maximum	14	18	9	8	2	9
Minimum	3	8	0	4	1	14

These results led us to believe that, although there wasn't a clear preference between the two non-random progression models, that there was a higher preference for those compared to the random progression model. As such, it made sense to proceed with the final evaluation.

4.2 Final Evaluation

For the final evaluation, we kept the modifications done for the second preliminary evaluation. Our goal was to gather 20 results per progression model, which made a total of 60 results. We ended up gathering 20 complete results for both the skill and choice-based progression models and 25 for the random progression model. Some of the results we gathered weren't complete, due to the fact that some players didn't finish their playthrough of the game before they submitted their survey. An important thing to note is that there is a difference between finishing the game, which was optional and involved completing 20 rooms without being defeated, and completing their playthrough, which was necessary and involved completing 20 rooms in total. The main thing to note about these results is that we know that these players played the game and answered the survey, we just couldn't gather any of their gameplay stats. In this chapter, we will refer to these incomplete results as players who didn't submit their gameplay data.

These are the stats that we tracked during each playthrough for this study, which were sent to a Node.js server after the player completes his playthrough. The numbers reported alongside them are relative to the final evaluation, not the preliminary evaluations:

- The specific types of enemies defeated, in which room they were defeated and how long it took the player to defeat them;
- Which progression model was used in that playthrough: 22 players used the skill-based progression model, 23 players used the choice-based progression model and 34 used the random progression model. The reason for the discrepancy between the number of players that played the non-random progression model and the random progression model was mainly due to technical constraints/shortcomings of our Node.js server;
- How much time the player spent in the tutorial room, before starting his first playthrough: if ignoring

one considerable outlier, the mean time was 103.97 $\pm$ 52.95 seconds and the median time was 97.50 seconds (max: 277; min: 10);

- How much time the player spent in his playthrough: the mean time was 573.06 $\pm$ 115.26 seconds and the median time was 562 seconds (max: 964; min: 376);
- How many rooms the player was able to complete in each of his playthroughs: the mean amount was 28 $\pm$ 6.12 rooms and the median amount was 26.5 rooms (max: 40; min: 20);
- The weapon the player used to defeat the last enemy in each room: 53% of players used the Ice Rifle more than any other weapon;
- How much health the player had left after completing each room.

Keep in mind that all the tests were done remotely, where each user would independently answer the survey and play the game on their own machine, without any supervision on our side. In the following paragraphs we will be doing an analysis of the results we obtained.

4.2.1 Demographics

Here are some demographic results that we obtained during the experiment:

- 87% of the players were male (Fig. 4.3) ;
- Only 2.5% of the players didn't play video games (Fig. 4.3);
- 62% of the players were familiar with and enjoyed games similar to Holiday Knight, where the player has to focus on shooting at several enemies simultaneously while dodging their fire (Fig. 4.4);
- 17% of the players were able to finish the game, which in the case of Holiday Knight means completing the 20 rooms without being defeated (Fig. 4.4).

4.2.2 Results

Due to the fact that, when testing different subsets of the data, not all of the 6 dimensions of the GEQ followed a normal distribution, we did a combination of parametric and non-parametric tests, verifying the normality of said subsets whenever applicable.

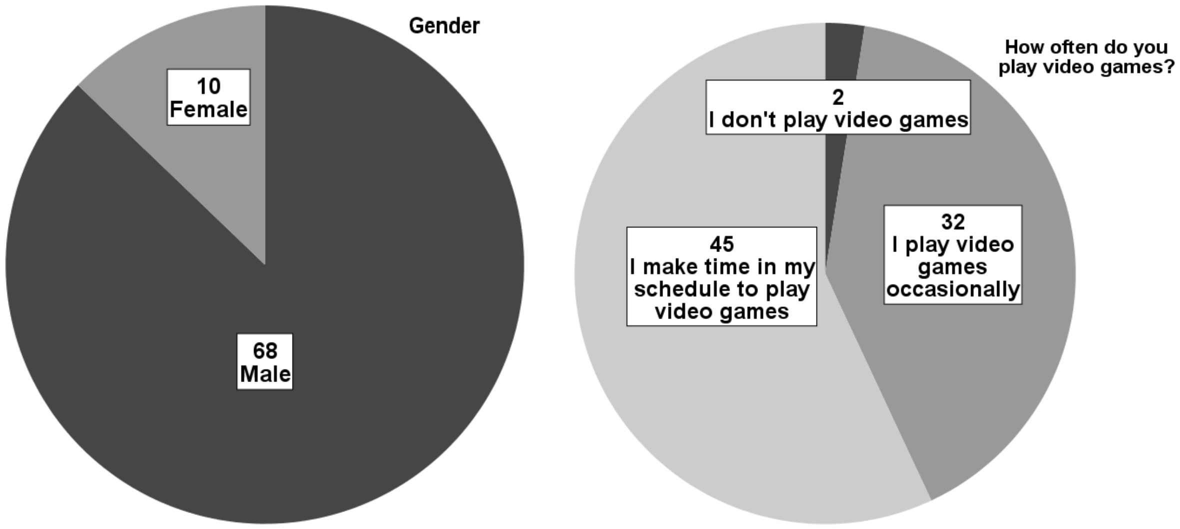

Figure 4.3

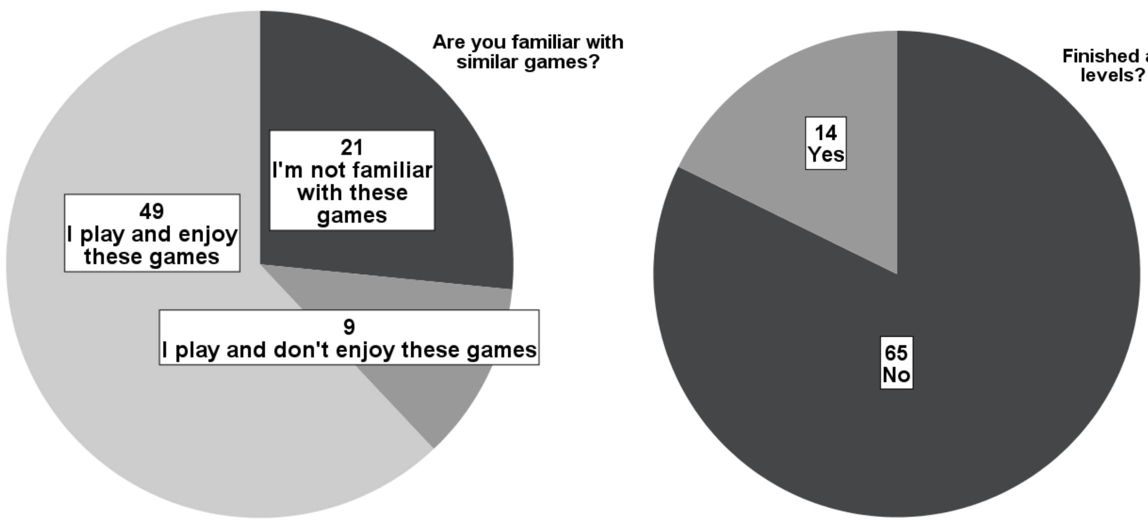

Figure 4.4

4.2.2.A Progression Models

A – Positive Affect When we put the data through a parametric One-Way ANOVA test, a statistically significant difference was found between the 3 progression models in the Positive Affect dimension. When only taking into account players who considered themselves familiar with similar games, a statistically significant difference $(F(2,46) = 5.295, p = 0.009)$ was found.

When we put the same data through a parametric Independent t-test, only taking into account players who considered themselves familiar with similar games, a statistically significant difference $(t(37) = -3.102, p = 0.004)$ was found where the players who played the random progression model felt more Positive Affect (17.58 ± 4.0) than those who played the choice-based progression model (13.15 ± 4.5).

B – Tension/Annoyance When we put the data through a parametric independent t-test, only taking into account players who had completed more than the median amount of rooms, a statistically significant difference $(t(17) = 2.588, p = 0.019)$ was found where the players who played the skill-based progression model felt more tension/annoyance (7.25) than those who played the random progression model (4.91).

C – Competence When we put the data through a non-parametric Kruskal-Wallis H test, only taking into account players who considered themselves familiar with similar games, a statistically significant difference $(X^2(2) = 6.909, p = 0.032)$ was found between the 3 progression models in the Competence dimension.

When we put the same data through a non-parametric Mann-Whitney U test, a statistically significant difference $(U = 13, p = 0.010)$ was found where the players who played the random progression model felt more competent (12.82) than those who played the skill-based progression model (6.13).

D – Remarks These results suggest that players preferred the random progression model over both the skill and choice-based progression models, and that, at best, they had a small preference over the skill-based progression model compared to the choice-based progression model.

A possible explanation for these results is that the difference in enemy quantity and type will be less pronounced in the non-random progression models, given the gradual increase in difficulty, which may cause the content to feel rather similar from room to room. Another reason may be that the loot presented to the player isn't of enough value to serve as a strong foundation for the choice-based progression model.

4.2.2.B Ice Rifle

A – Positive Affect When we put the data through a parametric Independent t-test, a statistically significant difference ($t(63) = 2.779, p = 0.007$) was found where players who didn't use the Ice Rifle the most felt more positive affect (17.57 ± 3.5) than those who did (14.80 ± 4.4).

B – Tension/Annoyance & Negative Affect When we put the data through a non-parametric Mann-Whitney U, a statistically significant difference ($U = 355, p = 0.024$) was found where players who used the Ice Rifle the most felt more tension/annoyance (37.86) than those who didn't (27.33). The same test found a statistically significant difference ($U = 303.5, p = 0.003$) where players who used the Ice Rifle the most felt more negative affect (39.33) than those who didn't (25.62).

C – Remarks This suggests that players, in general, didn't enjoy using the Ice Rifle, which may be due to the fact that it plays differently from the other weapon types, particularly the "2D recoil" and burst-fire features.

There are a total of 9 different combinations of weapons in this version of Holiday Knight (3 weapon types and 3 weapons elements). More than half the players used the Ice Rifle more than any other weapon throughout their playthrough. The reason for this could be due to the fact that the Ice Rifle is the closest weapon to player when the playthrough starts or that the game did not motivate the player enough to switch weapons.

4.2.3 Discussion

After doing the first preliminary evaluation, where we used only the skill-based and choice-based progression models, we decided to add a new random progression model that would pick enemies randomly, in order to have a baseline to compare both the other progression models with. The results we found indicated that players preferred the random progression model over either of the non-random progression models, which was not something we expected, and likely means that a few more steps need to be taken in regards to the other progression models' implementation before we have a strong foundation upon which to answer our research question, which PCG approach leads to a more pleasant player experience: content generation based on the player's skill or the player's choice. Additionally, there was no strong indication about the players' preference over either one of the non-random progression models over the other, which was the main goal of this work.

A possibility as to why players preferred the random progression model, as we mentioned in the previous section, is that the content variety was more pronounced in the random progression models than in the other two progression models, due to the fact that their increase in difficulty was very gradual. We had removed the Content Variety model originally present in Holiday Knight in order to minimize the differences between the skill-based and choice-based progression models, given that the choice-based progression model did not make use of the Content Variety model. This is likely something that should be reconsidered in future work, where the Content Variety model should be restored to the skill-based progression model, and possibly added to the choice-based progression model as well, given that it may be an important aspect of the gameplay experience for this experiment.

Another possibility is that the items' (in this case, the weapons) value to the player should be more thoroughly tested, given that the choice-based progression model heavily relies on that aspect of the game. In other words, there needs to be a certain degree of confidence that all the content presented to the player has value in certain gameplay scenarios. This could be achieved with a greater focus on playtesting and balancing. Although we did do these steps, we should have focused more on the players' use of the different gameplay elements.

Our survey allowed players to add comments about their experience playing the game, and some players took the opportunity to give their thoughts as to which weapons they believed were more or less valuable/powerful to them. There was a variety of different weapons that players felt were most valuable to them, which leads us to believe that there wasn't a specific weapon that was clearly more valuable/powerful than the other weapons. We consider this a good thing because, if there was a universal unanimity between players as to which weapon was more powerful, that would mean that that weapon would probably be unbalanced compared to all the others. Because this unanimity wasn't present, we believe the weapons to be relatively well-balanced in general.

When it comes to the heavy use of the Ice Rifle over any other weapon (more than half the players used the Ice Rifle more than any other weapon), we believe that this was due to the fact that the Ice Rifle is the closest weapon to the player when he starts his playthrough, which may have created a bias toward that weapon in particular. In future work, we suggest that the weapon types and elements that players have access to at the beginning of the game should be randomized, so that the bias that players might have in choosing the closest weapon doesn't lead to an increased use of one weapon in specific.

It's possible that the amount of time that players played Holiday Knight wasn't enough to properly analyze the efficacy of a progression model in the context of a game (the mean time a user spent playing was around 11 minutes and a half). We believe that the amount of content we asked the players to complete before they could finish the activity was a balance between asking them to play for too long, to the point where they could be bored with the activity, and asking them to play for too little time, to the point that our results would not be considered meaningful. It is possible that this balance isn't the most appropriate for the player to properly feel the content's progression and that we can ask players to play for longer, however, this may lead to us only being able to measure the impact with participants who don't mind playing for an extended period of time.

It's also possible that the disparity between the number of players who played the random progression

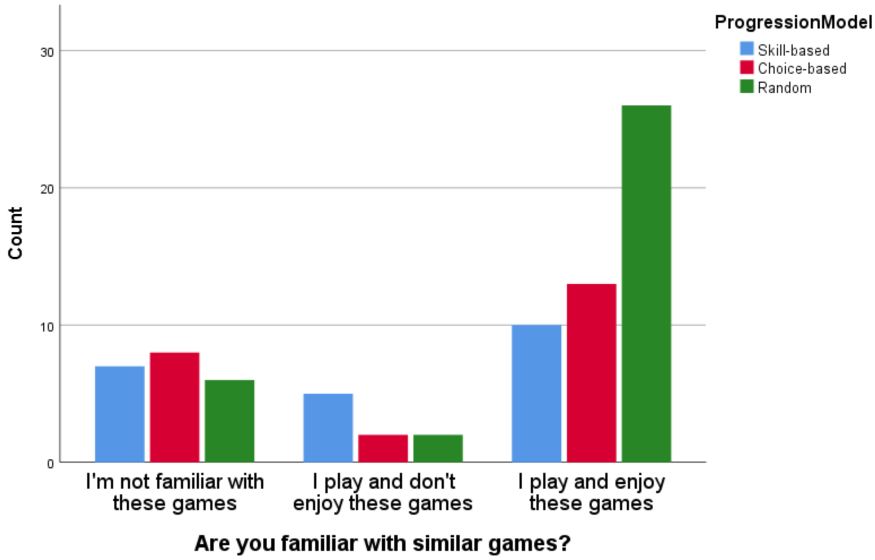

Figure 4.5

model compared to the players who played the non-random progression models may have influenced the results in one way or another. When separating players between three groups, one for each progression model, we found that the players who played the random progression model were much more likely to be familiar with and enjoy games similar to Holiday Knight, as is shown in Fig. 4.5.

Due to the COVID-19 pandemic, doing offline experiments with players in an office or lab was not feasible, so we resorted to doing online, unsupervised experiments, so that we could more easily achieve the amount of results we wanted, which caused, among other things, players who didn't complete the game the way we had intended. A specific case we encountered was that of a player who spent over 10 hours in the tutorial, but then spent very close to the mean amount of time completing the playthrough, which may have been due to the fact that he left the game running for many hours in the tutorial while doing something else before returning and completing the experiment.

5

Conclusion

This work's goal was answering the following research question: which PCG approach leads to a more pleasant player experience: content generation based on the player's skill or the player's choice. In order to answer this question, we adapted an existing game, Holiday Knight, and asked players to play it while measuring their gameplay experience in different versions of the game. The original version of Holiday Knight contained only a skill-based progression model, to which we added a new choice-based progression model. The way in which both of these differed was simply in the way in which they chose which enemies to present to the player at any given time: the skill-based progression model would choose these enemies based on the player's past performance facing them, while the choice-based progression model would choose them based on the weapon the player was currently carrying. In order to have a baseline progression model to compare them to, we added a third progression model that would simply choose both the amount of enemies and the individual enemies at random for the player to face.

In the experiments we did, we were not able to arrive to a definite conclusion regarding our research question. There was a slight preference, at best, toward the skill-based progression model compared to the choice-based progression model, however, there was a clear preference for the random progression model when compared to the other progression models. Several factors may have contributed to this outcome, and we've presented a set of guidelines to help with future follow-up work.

In future work, a greater emphasis should be put on making sure the content presented to the player is varied and that the loot presented is of value to the player. We had decided to remove the Content Variety model from Pardal's original version of Holiday Knight, in order to have the skill and choice-based progression models differ as little as possible, but it is likely that this will have to be reconsidered for future work.

In future work, we suggest that the weapon types and elements that players have access to at the beginning of the game should be randomized, so that the bias the players may have in choosing the closest weapon to them doesn't affect the use of any one weapon combination in specific over the others.

The COVID-19 pandemic limited our possibilities when it came to offline experiments with testers, so we had to come up with an alternative way of doing those experiments. We chose to do offline, deferred, unsupervised experiments, given that players shouldn't need any guidance during the process, which caused us to have difficulty interpreting some outliers in our dataset, as well as not having received the complete results from every player. In the future, similar experiments should be more cautious about this approach, and consider doing online, supervised experiments instead, although that may slow the process of gathering results considerably.

Bibliography

[1] J. Catarino and C. Martinho, "Skill-based progression model for smash time," in *Proceedings of IEEE Conference on Games 2019*, 2018.

[2] J. Catarino and Martinho, "Skill-based progression model for smash time," Master's thesis, Instituto Superior Técnico, 2018.

[3] J. Pardal, "Holiday knight: a videogame with skill-based challenge generation," Master's thesis, Instituto Superior Técnico, 2019.

[4] J. Togelius, N. Shaker, and M. J. Nelson, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2016.

[5] M. Hendrikx, S. Meijer, J. V. D. Velden, and A. Iosup, "Procedural content generation for games: A survey," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 9, no. 1, 2013. [Online]. Available: https://doi.org/10.1145/2422956.2422957

[6] G. Smith, E. Gan, A. Othenin-Girard, and J. Whitehead, "Pcg-based game design: Enabling new play experiences through procedural content generation," in *Proceedings of the 2Nd International Workshop on Procedural Content Generation in Games*. ACM, 2011. [Online]. Available: http://doi.acm.org/10.1145/2000919.2000926

[7] F. Bicho and C. Martinho, "Multi-dimensional player skill progression modelling for procedural content generation," in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, 2018.

[8] M. Jennings-Teats, G. Smith, and N. Wardrip-Fruin, "Polymorph: A model for dynamic level generation," in *Proceedings of the Sixth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, ser. AIIDE'10. AAAI Press, 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=3014666.3014693

[9] H. Wang and C.-T. Sun, “Game reward systems: Gaming experiences and social meanings,” in *Proceedings of the 2011 DiGRA International Conference: Think Design Play*, 2011. [Online]. Available: http://www.digra.org/wp-content/uploads/digital-library/11310.20247.pdf

[10] M.-V. Aponte, G. Levieux, and S. Natkin, “Measuring the level of difficulty in single player video games,” *Entertainment Computing*, vol. 2, 2009.

[11] A. Sarkar and S. Cooper, “Transforming game difficulty curves using function composition,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 2019. [Online]. Available: http://doi.acm.org/10.1145/3290605.3300781

[12] B. Horn, A. Hoover, Y. Folajimi, J. Barnes, C. Harteveld, and G. Smith, “Ai-assisted analysis of player strategy across level progressions in a puzzle game,” in *Proceedings of the 12th International Conference on the Foundations of Digital Games*, ser. FDG ’17. ACM, 2017. [Online]. Available: http://doi.acm.org/10.1145/3102071.3102083

[13] T. Allart, G. Levieux, M. Pierfitte, A. Guilloux, and S. Natkin, “Difficulty influence on motivation over time in video games using survival analysis,” in *FDG ’17*, 2017.

[14] Y. d. K. W.A. IJsselsteijn and K. Poels, *The Game Experience Questionnaire*. Technische Universiteit Eindhoven, 2013.

[15] 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 *Immersive Learning Research Network*. Springer International Publishing, 2016.

[16] 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 *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 2015. [Online]. Available: https://doi.org/10.1145/2702123.2702447

[17] K. M. Gerling, M. Klauser, and J. Niesenhaus, “Measuring the impact of game controllers on player experience in fps games,” in *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*. Association for Computing Machinery, 2011. [Online]. Available: https://doi.org/10.1145/2181037.2181052

[18] M. Csikszentmihalyi and S. Csikszentmihalyi, *Optimal Experience: Psychological Studies of Flow in Consciousness*. Cambridge University Press, 1988.

A

Post-Game Game Experience Questionnaire

Holiday Knight Player Questionnaire

Thank you for taking part in this study!

I have developed 3 versions of the game Holiday Knight for my Master's degree, and would like your opinion on one of them.
The difference between the 3 versions of the game is only the way in which the enemies are chosen throughout the game.

After answering a few questions about yourself, I would kindly ask you to play the game and answer a short questionnaire. The whole experiment should last around 15-20 minutes, depending on how long you play.

I want to inform you that your participation is completely voluntary and you can stop the activity at any moment. If you have any questions, you can send an email to goncalo.alex.marx@gmail.com.
You will not be identified in any phase of this activity and individual results will not be shared with anyone.

By continuing this survey you're consenting to the points in the previous paragraph.
* Required

1. How old are you? *

____

2. What is your gender? *

Mark only one oval.

- ( ) Female
- ( ) Male
- ( ) Other: ______________________________

3. How often do you play videogames? *

Mark only one oval.

- [ ] I make some time in my schedule to play video games.
- [ ] I play video games occasionally when the opportunity presents itself.
- [ ] I do not play video games.

4. Are you familiar with games in which the protagonist combats a large number of enemies by shooting at them while dodging their fire, such as Enter the Gungeon, Binding of Isaac, Cuphead or Nuclear Throne? *

Mark only one oval.

- [ ] I enjoy these games and have played/watched others play them multiple times.
- [ ] I played/watched others play them enough to understand I do not appreciate them.
- [ ] I am not familiar with these games and/or have no formed opinion on them.

Your Player ID

Now that you have told us a bit about you, please download Holiday Knight at this link:

https://goncasmage.itch.io/holiday-knight
Password: holidayknight

When you launch the game, you can start playing the tutorial where you will have the change to get acquainted with the game's mechanics. After you feel confident with the game's mechanics, start playing the game. To ensure you have enough experience with the game, we ask you to beat at least 20 rooms in total.

After you've completed your playthrough, a Player ID will be generated, which you'll need in order to proceed with this activity.

5. Please insert your Player ID *

____

Playing the game

Now that you have launched the game, you can start playing the tutorial where you will get acquainted with the game's mechanics. After you feel confident with the game's mechanics, start playing the game. To ensure you have enough experience with the game, we ask you to play at least 20 rooms. When you are done, you will be asked to return here and fill a short survey.

Experience

Before answering these questions, make sure you've completed the tutorial and finished your playthrough of the game.

6. Please indicate how you felt while playing the game. *

Mark only one oval per row.

	not at all	slightly	moderately	fairly	extremely
I felt content					
I felt skilful					
I thought it was fun					
I was fully ocuppied 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					
I forgot everything around me					
I felt good					
I was good at it					
I felt bored					
I felt successful					
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 was deeply concentrated in the game					
I felt frustrated					
I lost connection with the outside world					
I felt time pressure					
I had to put a lot of effort into it					

7. If you have any comments about your playthrough, positive or negative, please tell us about them.

Thank you!

Thank you so much for participating in this survey! We really appreciate it and hope you've enjoyed playing the game.

8. If you're willing to let us contact you further about this activity, could you give us your email address?