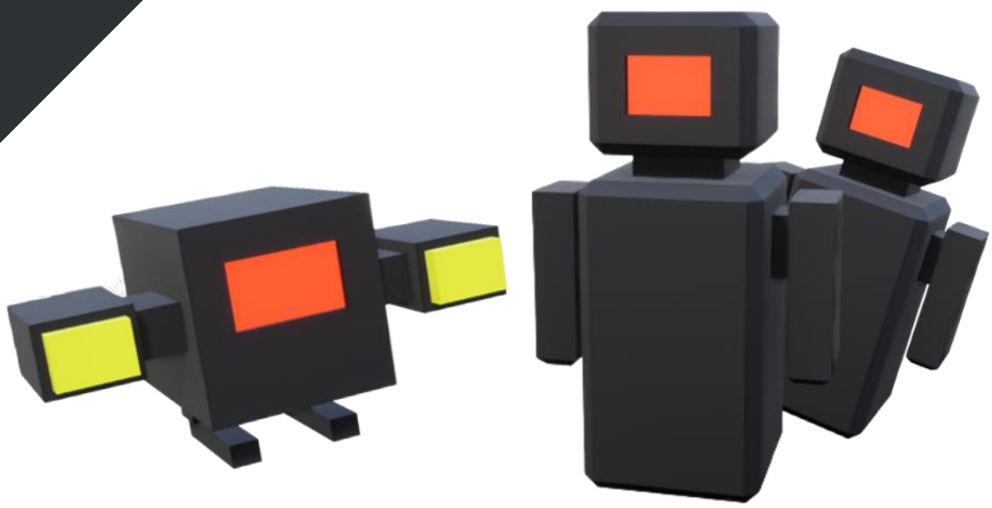

Replayable Looter Shooter

BA (Hons) Game Design and Production

Liam Rickman – 1902527

School of Design and Informatics
Abertay University (May 2023)

Table of Contents

List of Figures

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor Fraser Simpson for all their help and guidance. Their feedback, support, and expertise throughout the project has been invaluable while developing this project.

I am thankful for the lecturers from the School of Design and Informatics at Abertay University who provided further support and guidance. I am also grateful for the resources provided by Abertay which were vital for developing the project.

Finally, I would like to thank my family, friends, and peers for all their support. Their feedback and encouragement has been invaluable to the project and keeping me motivated.

Abstract

Improving replayability in games is important for any game developer wanting to enhance the longevity and enjoyment of their game. This project investigates replayability and how game designers can improve replayability through their design. It was inspired by other replayable games *Deep Rock Galactic* (Ghost Ship Games. 2018) and *Doom Eternal* (id Software. 2020). The outcome of the project is a first-person looter shooter prototype developed in Unreal Engine 5, featuring replayable game design techniques.

The project started with an extensive literature review which explored core loops, mechanic design, and the use of randomness in games to create a knowledge base of practices used today. Additionally, game analysis provided insight into current techniques both successful and unsuccessful used in games. To build on this knowledge, an iterative process was used to create the game prototype, using playtests and user feedback to inform development. Finally, the prototype and development were evaluated through a postmortem to discuss its successes and areas for improvement.

The findings emphasize the importance of motivation and player-led goals in enhancing replayability. Intrinsic motivation and self-determination theory were particularly powerful for motivating players to replay games. A strong core loop foundation is crucial for creating an engaging experience and works best when supported with additional gameplay and feedback loops. Appropriate implementation of randomness and procedural generation was beneficial in creating infinite variations of content which can prevent repetition and delay boredom. While these features can help boost replayability, further investigation would be required on a per-game basis due to the differences between games developed.

1. Introduction

1.1. Topic

This project investigates replayability in games through a first-person looter shooter prototype featuring the core loop and one mission type. I was inspired by designers who make highly replayable games with satisfying core loops. I aim to improve my skills in gameplay and mechanic design, but may explore level design, UI/UX design, and other areas.

Replayability or replay value is the perceived value a game has of being played multiple times (Spacey. 2016). This translates well for games like *Super Mario Bros* (Nintendo. 2012), featuring a clear end goal – saving Princess Peach. More open-ended games like *Deep Rock Galactic* (Ghost Ship Games. 2018) can have replay value by allowing players to replay the core loop repeatedly, with new experiences each time.

1.2. Questions, Aims, and Objectives

1.2.1. Research Question

I aim to answer the following:

- How can game designers improve replayability in video games?

1.2.2. Aims

To answer this, I aim to:

- Define replayability.
- Examine replayability factors (including):
    - Core loops
    - Mechanics
    - Randomness and procedural generation
- Establish best practices for replayability including:
    - Core loops
    - Mechanics
    - Randomness and procedural generation
    - Iterative design process
    - Playtesting and user feedback

1.2.3. Objectives

This will be achieved by:

1. Analysing replayable gameplay and mechanic design in literature and existing games.
2. Designing and iterating on a game prototype using replayable game design methods.
3. Evaluating and discussing the production process through a postmortem.

1.3. Methods

1.3.1. Literature Review

The review will build up a foundation of existing knowledge of replayable game design. I will focus on gameplay and mechanic design but may include other areas. This research will be put into practice to help design and develop my practical prototype.

1.3.2. Game Analysis

Various games will be analysed to discover what improves or harms replayability in current games. This will reinforce my understanding of existing practices in games today and develop my prototype. Games will be analysed for features including mechanics, gameplay, UI/UX, and core loops.

1.3.3. Practical Research

I will create a prototype using an iterative approach, with small sprints to break up development. Sprints will be flexible to allow for adjustment as needed. These sprints include research, planning, preproduction, production, playtesting, and analysis stages. I will conduct primarily practical work to create playable prototypes allowing for more dynamic playtesting and feedback.

1.3.4. Postmortem

The postmortem evaluates the development of the prototype, breaking down the steps to bring the project from concept to completion. I will discuss different iterations of the project, evaluating what went well, what went wrong, and what could have been improved.

2. Research Context

2.1. Motivation and Goals

2.1.1. Intrinsic vs. Extrinsic Motivation

2.1.1.1. Intrinsic Motivation
Two motivational theories are frequently used to motivate people inside and outside of games (Reiss. 2012). Firstly, intrinsic motivation is described by Reiss as motivation where a person would do something for their own reasons or enjoyment, without external motivators. It is usually the strongest method to motivate people and keep engagement high (Eng. 2019). Eng states that intrinsic motivation works best for areas someone may already have an interest in, such as solving puzzles because they enjoy the challenge.

2.1.1.2. Extrinsic Motivation
Extrinsic motivation is when players are given incentives to complete an activity, such as earning money after completion (Reiss. 2012). This is best used to motivate people to try something new, such as learning a new topic in school (Cherry. 2022). Cherry recommends avoiding combining intrinsic motivation and extrinsic motivation for the same area as this can lead to overcompensation. She says this is common when adding extrinsic motivation to existing intrinsic motivation and often decreases the impact of intrinsic motivation. As intrinsic is a more powerful motivation, this can detrimentally impact the overall motivation provided (Eng. 2019).

2.1.2. Self-Determination Theory

A mini theory of self-determination theory is Cognitive Evaluation Theory (CET) which discusses people doing things for their own reasons – a core feature of intrinsic motivation (SelfDeterminationTheory. No date). CET has three core needs: competence, autonomy, and relatedness (Rigby and Ryan. 2011). Games that satisfy these needs can help boost a player's sense of immersion and motivation which could boost replayability (Przybylski, Rigby, and Ryan. 2010).

2.1.2.1. Need for Competence
Competence refers to a person's urge to improve or progress (Burke. 2014). Players seek to learn and improve themselves in areas of interest (Rigby and Ryan. 2011). This is common in games like *Elden Ring* (FromSoftware. 2022) which has lots of skill expression and chances to show off what has been learned. Competence is also seen in competitive games such as *Counter-Strike: Global Offensive* (Valve. 2012). The competitive nature of the game invites players to improve their skills and master the game, displaying their competence. It can also be met through games like *Satisfactory* (Coffee Stain Studios. 2019), where the progression system allows players to learn about the game's features and mechanics gradually, to build up their competence.

2.1.2.2. Need for Autonomy
Games are generally played voluntarily, with players choosing which games they play and why (Bartle. 2004). Autonomy is similar, being the desire to make choices and decisions without being directed every step of the way (Burke. 2014). Players

don't want to be controlled and would rather take actions of their own volition (Rigby and Ryan. 2011). This allows players to feel in control and have their decisions impact the game. *Civilization VI* (Firaxis Games. 2016) has high autonomy with players constantly making decisions. These often have short and long-term impacts on how the game plays out for them and other players. *The Walking Dead* (Telltale Games. 2012) appears to have high autonomy, however many choices lead to the same outcome, including which characters die. Giving players different choices, avoiding limiting choices, and more importantly, making choices that have a real impact on the game can help fulfil the need for autonomy (Rigby and Ryan. 2011).

2.1.2.3. Need for Relatedness

Relatedness is the player's connection to the game, helping players feel immersed and have a sense they are in the game world (Ryan, Rigby and Przybylski. 2006). Players want to feel their actions have a reason and purpose behind them (Burke. 2014). In singleplayer games, the in-game characters often fill this need as a means of social interaction (Przybylski, Rigby, and Ryan. 2010). They mention that in multiplayer games, other players can fill this need, often more successfully. Games such as *Destiny 2* (Bungie. 2017) and *World of Warcraft* (Blizzard Entertainment. 2004) use social systems like clans to connect people and reward them for doing so.

2.1.3. Player Goals

Player goals are what the player is aiming to do and are fundamental to giving players a sense of progression and accomplishment (Weitze, C, L. 2014). Without goals, players have no direction which can make it difficult to stay engaged and in turn, replay a game (Macklin and Sharp. 2016). *Super Mario Bros* (Nintendo. 1985) has an overarching narrative goal to rescue the princess, but also smaller goals such as stomping on Goombas, collecting coins, or moving to the right (Debus, Zagal and Cardona-Rivera. 2020). Implementing short-term and long-term goals keep engagement high and offer something players can work towards at all stages of the game (Game Maker's Toolkit. 2018a). Long-term goals can include player levels or achievements and are worked towards over multiple play sessions.

Some games rely on player-driven goals to drive gameplay. Sandbox games like *Minecraft* (Mojang Studios. 2011) commonly feature player-driven goals, relying on player input to decide many of the goals for the game. This is supported by short-term goals like killing enemies and eating food, however the player can decide how they do this. Players can choose long-term goals to achieve, which combined with intrinsic motivation, encourages players to do things for their own benefit and enjoyment.

2.2. Game Loops

2.2.1. Gameplay Loops

Gameplay loops are a series of repeated mechanics the player uses constantly while playing the game (Despain. 2012). Despain states that games have a core gameplay loop that defines the fundamental mechanics of the game. This varies in length, but as it is the lowest level loop, it is usually short to only cover the basics (Lovato. 2017). Core loops can be simple like walk, attack, collect, but could be more detailed such as entering rooms, killing enemies, and gathering loot. The core loop should then be supported by other loops to build up the gameplay. Ideally, each play session incorporates the core loop at least once, so it's important to decide the session length early on (Millard. 2019). Longer games like *Factorio* (Wube Software. 2020) or *Cities: Skylines* (Colossal Order. 2015), where the core loops can take hours and multiple sessions.

2.2.2. Feedback Loops

2.2.2.1. Positive Loops

Feedback loops help support gameplay loops and provide feedback to players relative to their actions. Positive loops are more common, used to reinforce successes with more successes and punish failures with more failures (Despain. 2012). Badly designed positive loops can lead to poor game balance, especially in multiplayer or competitive games. In *Call of Duty: Modern Warfare II* (Infinity Ward. 2022), getting kills awards killstreaks which can be used to get more kills, leading to a snowball effect. This is beneficial for those in the lead, but detrimental for weaker players who could get left behind (Game Maker's Toolkit. 2018b). He mentions that positive loops can help avoid stalemates in games like *Titanfall 2* (Respawn Entertainment. 2016) preventing games from lasting forever.

2.2.2.2. Negative Loops

Negative loops do the opposite by punishing successes and rewarding failures (Despain. 2012). This works well in casual or party games when a balanced playing field is desired to give everyone a good time. The item distribution in *Mario Kart 8* (Nintendo. 2014) is a negative loop where players in the lead get weaker items, making it harder to stay ahead. Lower-placed players receive stronger items to offer an advantage, keeping the game competitive (Game Maker's Toolkit. 2018b). However, negative loops can feel unfair, and players may feel like they are rewarded for failure and lose motivation to perform well. (Despain. 2012).

2.2.2.3. Combined Loops

Positive and negative loops can work well when combined and may provide the best experience (Game Maker's Toolkit. 2018b). They could provide rewards that don't directly impact the gameplay, avoiding upsetting the game balance (Despain. 2012). *Valorant* (Riot Games. 2020), combines loops in their economy system. Positive loops reward winning players by giving them more money than the losing team. To counteract a long-term snowball effect, a negative loop gradually gives the losing team more money if they lose multiple rounds.

2.3. Randomness and Procedural Generation

2.3.1. Randomness

Randomness adds variety to games which can improve replayability by creating infinitely more scenarios than manually designed ones. This allows players to replay the core features repeatedly but with differences each time. Randomness can stop games from feeling predictable and allow the player to master core mechanics to adapt to any scenario instead of memorising one scenario (Game Maker's Toolkit. 2020). A little randomness can keep a game unpredictable, however if it relies on it, randomness can take away a player's agency as there is a limit to success at random number generators (Design Doc. 2020). This takes away their incentive to play again, harming replayability. *PUBG: Battlegrounds* (PUBG Studios. 2017) uses randomness in its mechanics to decide where players land, what loot they find, and where they play on the map (Gałka and Strzeleckilt. 2021). This gives newer players a chance to keep up with skilled players if they get good loot and positioning. Randomness can also boost the impact of rare drops or encounters, especially if the player knows how rare they are (Game Maker's Toolkit. 2020).

2.3.2. Input vs. Output Randomness

2.3.2.1. Input Randomness

There are two types of randomness, the first is input randomness where the random element is used before a player makes a decision (Burgen. 2018). This includes procedural level generation or drawing a hand of cards where players can make decisions after the random element is revealed (Game Maker's Toolkit. 2020). It's important for games featuring randomness to have meaning behind it and allow players to react to the random elements, giving them more control (Burgen. 2018).

2.3.2.2. Output Randomness

Output randomness is where the random element comes in after players make decisions (Burgen. 2018). This is common in games such as *RISK: Global Domination* (SMG Studio. 2015) where you move pieces and then the randomness comes in via dice roll. Output randomness usually plays a more significant role in a player's anger and frustration towards randomness as they can't react to the randomness which can feel unfair. (Zhang et al. 2021). Because of this, input randomness is usually preferred over output randomness to give players the chance to impact the randomness and reduce any annoyance (Game Maker's Toolkit. 2020).

2.3.3. Procedural Generation

Procedural generation uses algorithms to generate content randomly and dynamically (Shaker, Togelius and Nelson. 2016). They state that it can generate many types of content including levels, items, quests, and more. Procedural generation can help games avoid becoming stale with more combinations of content available compared to manually designed content. This can be tailored to individual players by adjusting parameters to suit their needs. The roguelike *Spelunky* (Mossmouth. 2008) uses procedural generation to create game levels, making each run unique. Sandbox games like *Minecraft* (Mojang Studios. 2011) feature heavy use of procedural generation to generate the world and most other content.

2.4. Iterative Design

2.4.1. Iteration and sprints

Iterative game design is an adaptive process where sections of the game are revisited repeatedly until completion (Keith. 2010). This can be broken down into sprints lasting a few weeks and having their own goal that will be evaluated at the end of each sprint. Sprints cover concepts, prototypes, playtesting, and evaluation (Macklin and Sharp. 2016), however could cover other areas due to iterative design's flexibility.

2.4.2. Playtesting

Playtesting is an important stage in iterative design to help inform the subsequent sprints and project (Fullerton. 2018). She states that it's important to playtest throughout the entire project's development to ensure it continues to meet your player's needs at each iteration. Fullerton continues, discussing people you should test with including self-testing, testing with people you know, testing with people you don't know, and most importantly testing with the target audience.

3. Research Project

3.1. Project Details

The game prototype for this project is a first-person looter shooter game developed in Unreal Engine 5. The player must scavenge for resources in space to survive and earn money. This project had a massive scope, so I refined the project to cover a minimum viable product and the features the playable prototype required. This includes the core loop, one mission type, and some core mechanics to show off the intended gameplay. My research focused on these core features and how they could be designed with high replay value in mind to increase the longevity of the game.

3.2. Motivation and Goals

3.2.1. Loadout System

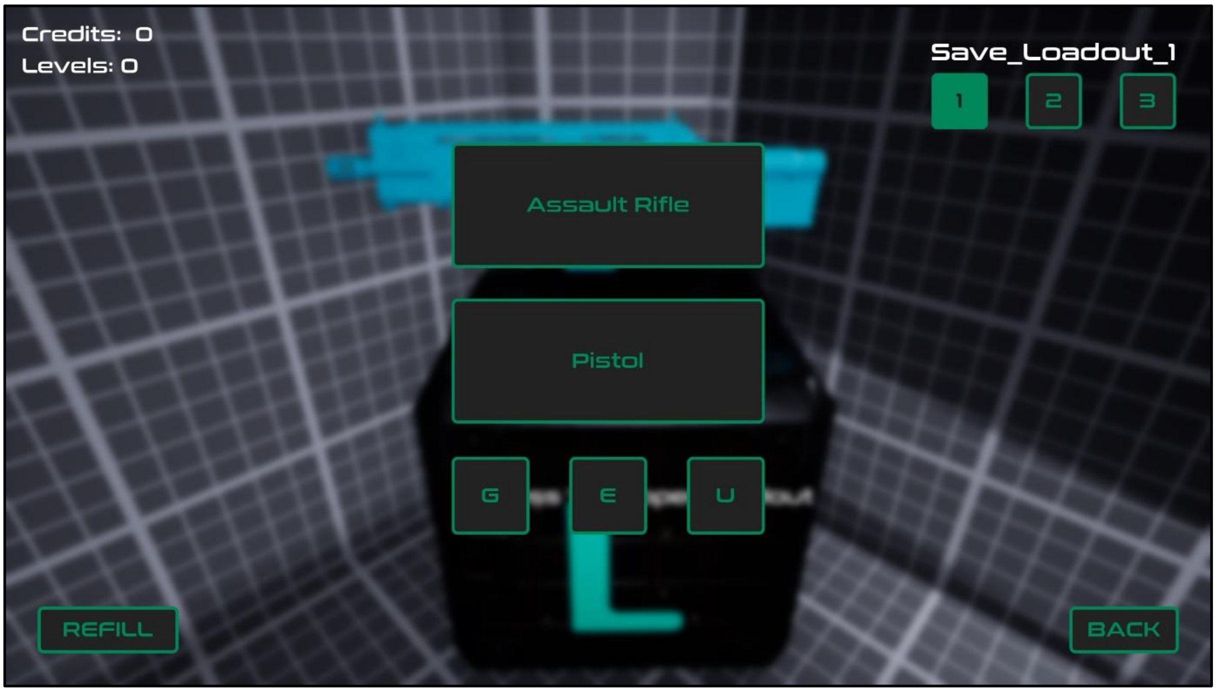

Figure 1: Loadout System

The largest impact on the prototype's replay value was motivating the player through various mechanics. Motivating features help drive the players to continue playing as they have reasons to do so. The main motivational mechanic was the loadout system which allows players to choose which weapons they bring into a mission (Figure 1). This was inspired by self-determination theory and cognitive evaluation theory to cover the player's need for autonomy. The loadout system gave players a choice of primaries, secondaries, and grenades which would alter their playstyle and possible approaches to a mission (Figure 2). Feedback found this gave players a sense of control and helped fulfil their need for relatedness, as they could impact how the gameplay felt. It also lessened the feeling of repetition despite the same core gameplay due to the variations it added.

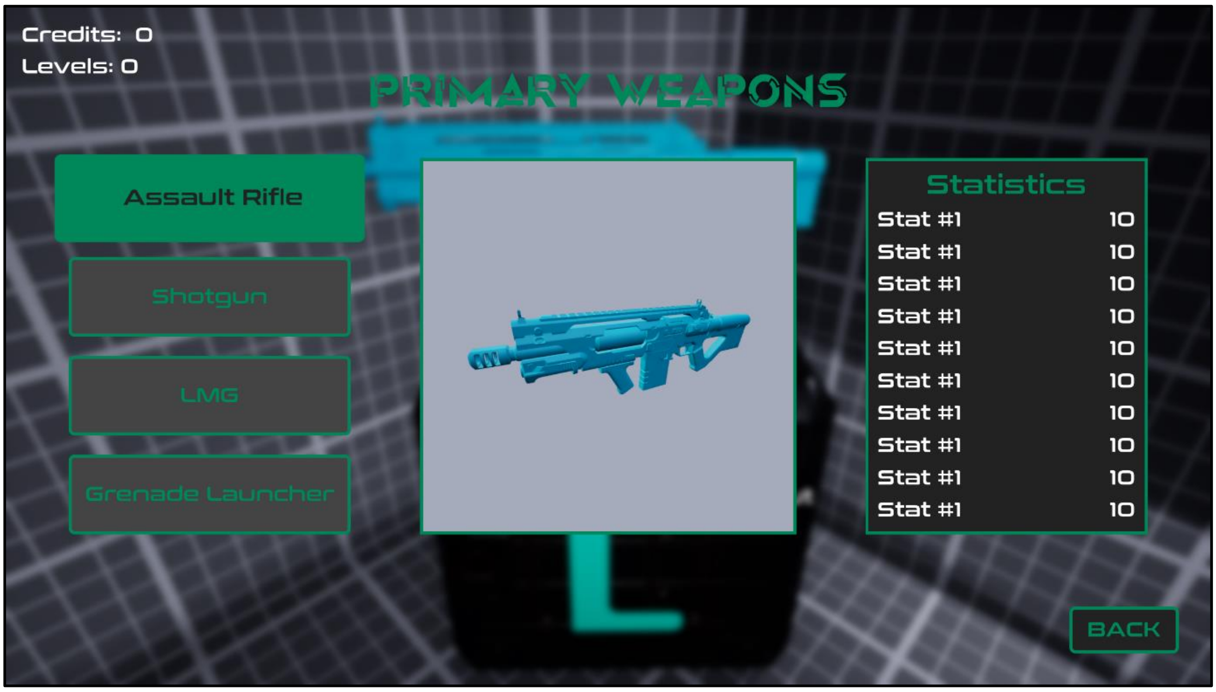

Figure 2: Loadout System (Primaries)

Originally, this system had all weapons unlocked from the start, giving them complete control immediately. I later changed this restricting access to all weapons apart from the starting weapons. I then added an unlock or shop system which allowed players to purchase the restricted weapons (Figure 3). Weapons can be purchased via credits – the in-game currency gained from killing enemies, gathering loot, or completing missions. Some weapons required a certain number of missions to be completed to unlock them. This is an intrinsic motivational mechanic as players will want to unlock these new weapons and to do so they must collect credits or complete missions to afford them. Playtests showed positive indications this type of system improved the replayability of the prototype as players could set their own goals after seeing something to work towards. These goals are not forced, and players don't have to unlock any weapons to play the game but is an opportunity for those wanting to expand their options. I planned to add an upgrade system for each weapon which would alter their function, either by boosting damage, fire rate or more. Unfortunately, this was out of scope for the project, however as this would add more choices for the player, this would have likely boosted replayability further, but may require further investigation.

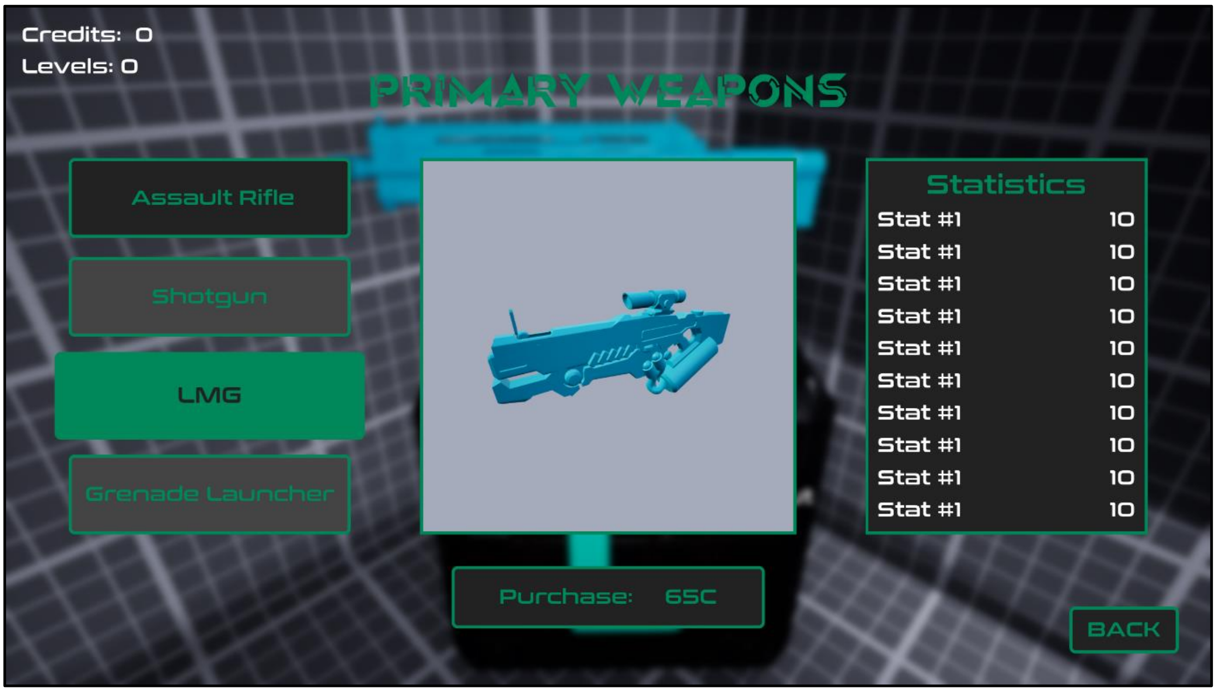

Figure 3: Purchasing Weapons

3.2.2. Missions

As the primary gameplay, players can earn money and weapons through missions. I created one type of mission called "Scavenge" where the player is sent to a space station to scavenge for some valuable loot. I broke the mission into six manageable objectives. Due to the procedural environments, I kept the objectives open to allow them to work in the environments but also give players more choices on how they approach them. This further improved the player's need for autonomy and helped keep players feeling in control. On mission completion, players are given credits as a reward (Figure 4), which while technically an extrinsic form of motivation, I found that this didn't detract from the intrinsic motivation created by the loadout system. Credits are only used to purchase weapons and can only be earned by completing missions. Though it was recommended to avoid combining intrinsic and extrinsic motivation, I found it worked well here as the overall motivation for completing missions remains intrinsic.

Figure 4: Mission Complete

3.2.3. Short-term goals

The core gameplay features short-term goals the player will immediately focus on while playing a mission. Killing enemies comes naturally as they pose a threat and players need to kill them to survive and complete missions (Figure 5). While the enemies are quite basic, they still help the player's need for competence as players can see themselves improving over time with their weapons and learning different combat mechanics. Players are rewarded with credits for killing enemies and while also an extrinsic reward, as mentioned earlier this doesn't detract from the player's intrinsic motivation to collect credits in the first place.

Figure 5: Enemies

I also added loot caches throughout the mission which players can collect for bonus credits (Figure 6). These reward players for spending time exploring the level and visiting areas they may have ignored. After playtests and feedback, I added health and ammo pickups to the caches, to give immediate value to them and help with the mission. These further motivated players to spend more time in a level as they had more health and ammo to survive longer. These loot caches provided motivated players with the tools to fully explore their environment, earning more credits, and ultimately extending their play sessions and the game's replayability.

Figure 6: Loot Caches

3.3. Game Loops

I created a small core loop for my game focusing on the actions players take when entering each room. This loop covers entering a room, defeating enemies, collecting loot, and leaving the room. This allowed players to take a methodical approach when completing missions, taking it one room at a time while constantly repeating the core loop each time they move rooms. Outside of the missions, the prototype has a larger gameplay loop covering selecting loadouts, completing missions, earning credits, and purchasing new equipment. This large loop proved beneficial for boosting replayability by allowing players to experience most core gameplay features every time they play even in shorter play sessions.

I used primarily positive loops to provide feedback to the player and reward them for their time and effort. This can be seen when rewarding players with credits when killing enemies. The more enemies a player defeats, the more credits they can earn per mission. This encourages the player to seek out enemies to earn a greater reward. The same works with the loot caches which encourage players to explore the whole level to gain further rewards. This ultimately boosts replayability by encouraging players to explore all the available content in a mission instead of taking a repetitive but more optimal route.

One feedback loop that didn't work well was when the player died (Figure 7), which is quite abrupt and offers little sympathy when they do. I did add an option to restart the level, however this is from the beginning and could get annoying, especially if players were far into the level. With further improvements, I would investigate a respawn or revive system that could give players another chance to complete the mission. This could respawn the player at set times in the mission, possibly near the last completed objective, like a checkpoint system used in *New Super Mario Bros U* (Nintendo. 2012).

Figure 7: Mission Failure / Death Screen

3.4. Randomness and Procedural Generation

3.4.1. Level Generation

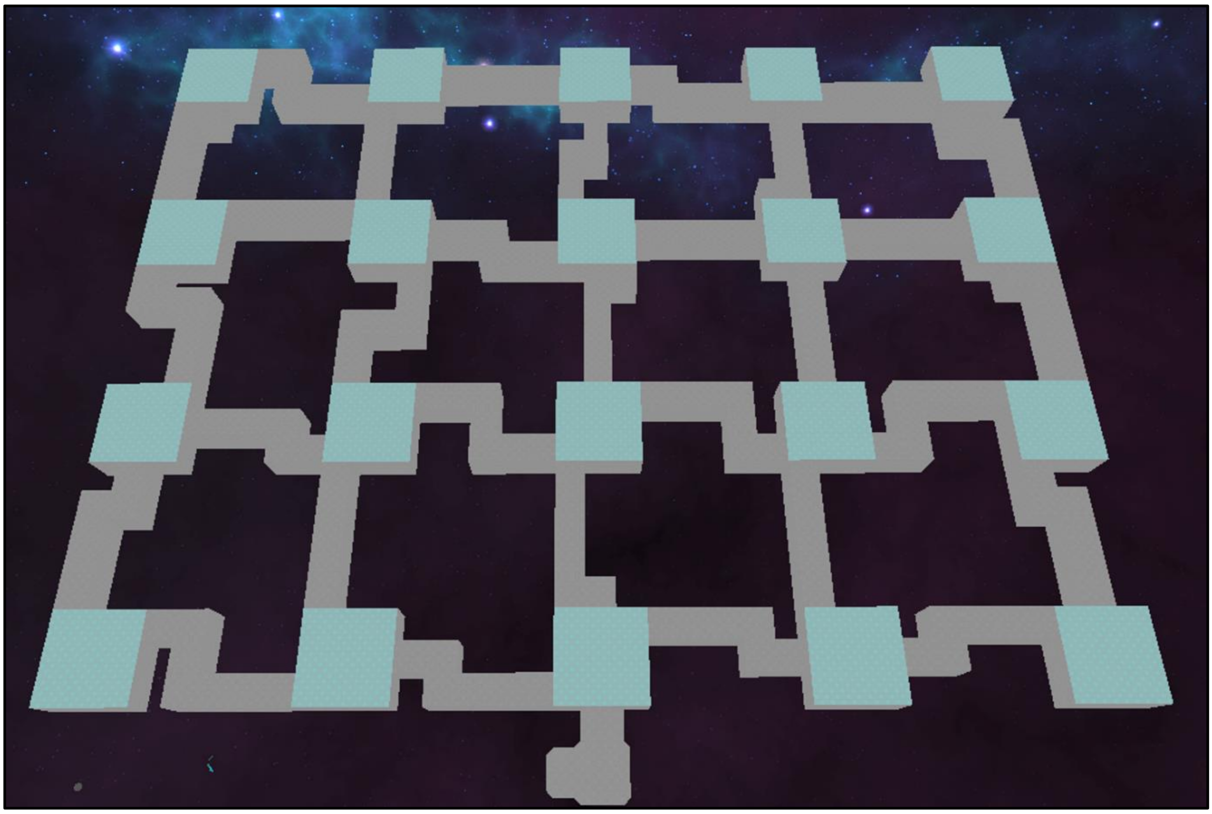

Figure 8: Procedural Grid Prototype

A large part of randomness I added was through level generation which uses procedural generation to create different variations with the same parameters. A grid system creates rows and columns of rooms, imitating a space station (Figure 8). I then connected the rooms up with hallways to allow the player to move between each room. I then increased hallway variation by changing where the entrances and exits were, along with grid adjustments to offset each room. Finally, I randomly swapped out rooms with versions required for the scavenge mission (Figure 9). This is a form of input randomness and allows the player to react to the randomness by making decisions after the randomness occurs. This effectively reduced the feeling of bad luck while increasing the variations and replay value.

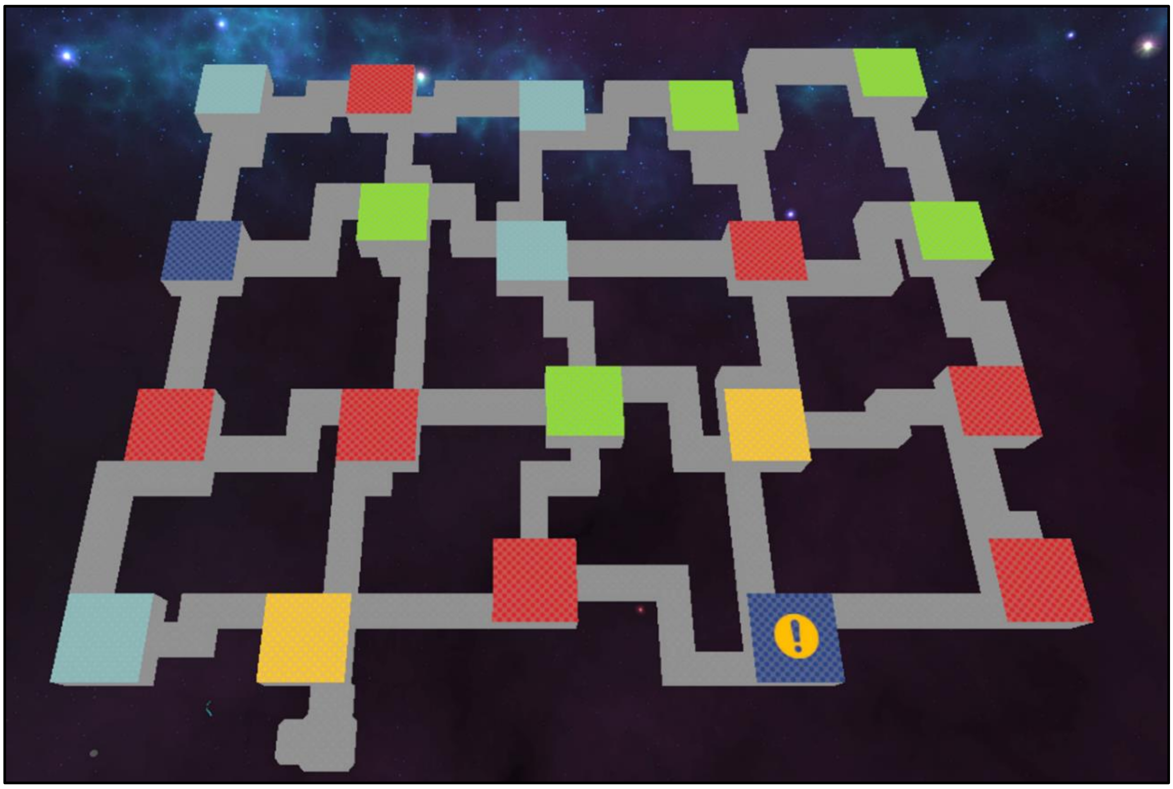

Figure 9: Final Procedural Level Generation

The mission selection screen gives players more control over level generation by letting players choose mission parameters (Figure 10). This adjusts how the level generates using length, complexity, and difficulty, without specifying exactly how the mission and level are generated. This ensures players don't have too much control over the randomness before it occurs and reduces the likelihood of replaying the same level layout again if they keep entering the same parameters.

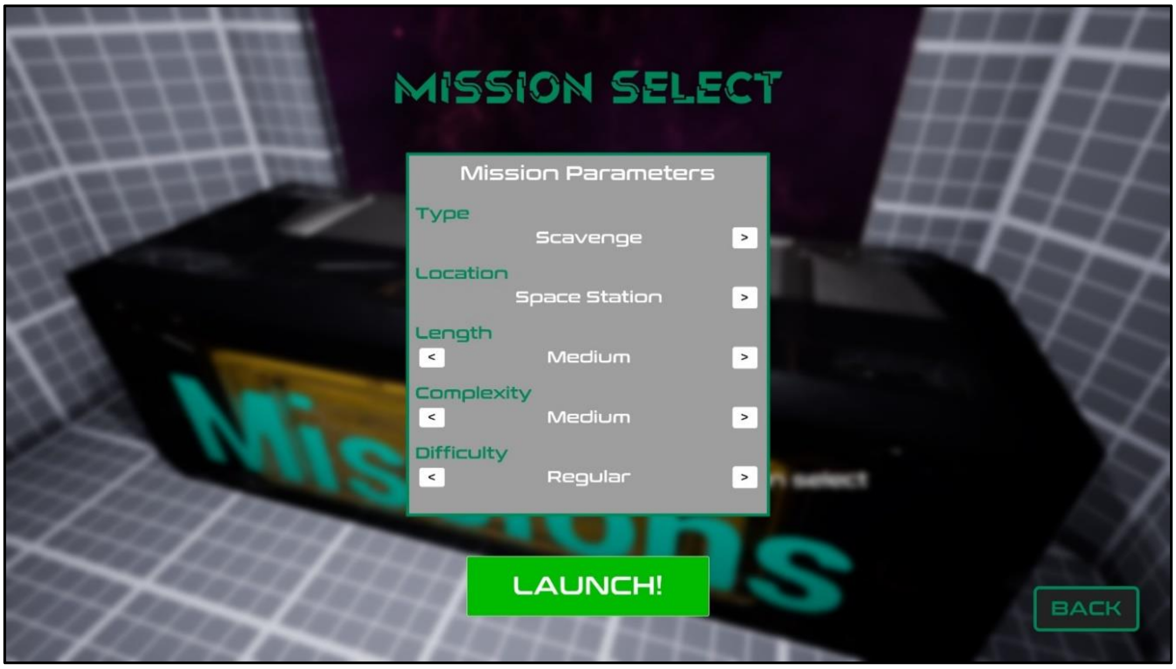

Figure 10: Mission Selection Screen

3.4.2. Missions

With only one mission type, I improved variation within the mission to change how they are played. First, I added variation through the mission parameters (Figure 11). Higher difficulties had more enemy rooms, and the longer the level the more total rooms there were. If a level was more complex, there were more options for the mission-specific rooms such as the command centre or vault. This required the player to check multiple rooms rather than knowing exactly where the room was. I also randomly placed loot rooms which would spawn loot caches mentioned earlier. Finally, there were enemy rooms that spawned a random number of enemies for the player to defeat once they got close. These features combined with the procedurally generated levels gave the missions more variation making them less repetitive.

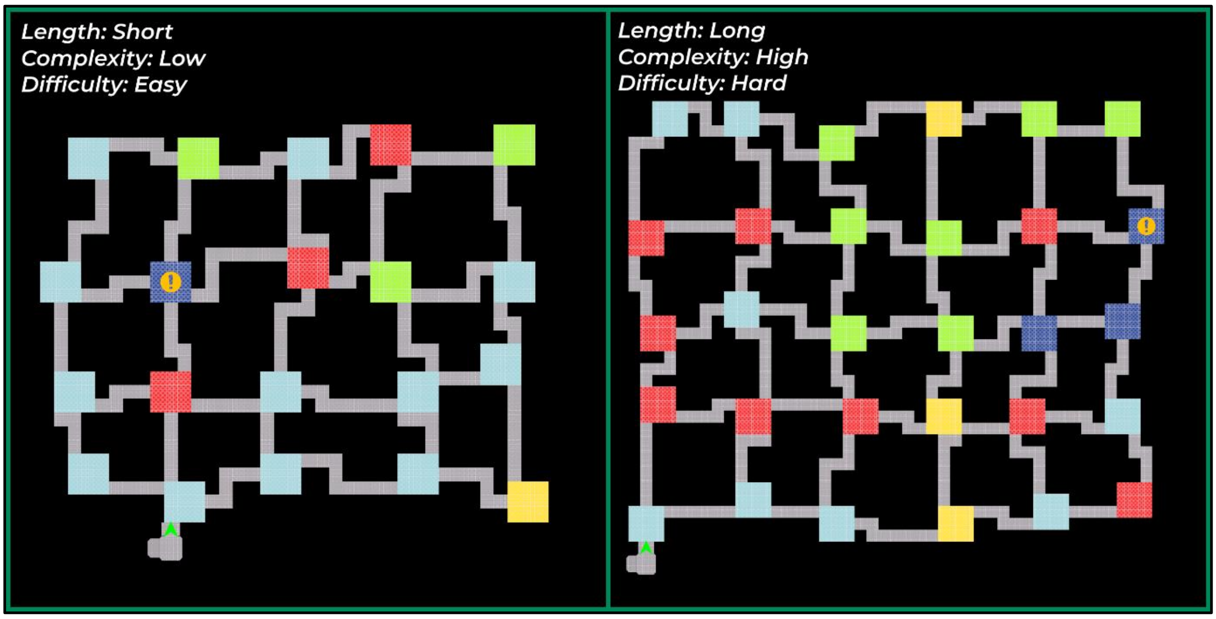

Figure 11: Mission Parameters (Minimum vs. Maximum Values)

However, playtesting showed that while these implementations did temporarily boost replayability, they added little value long-term as there were not enough variations to keep the player from getting bored. Unfortunately, due to time constraints and the large project scope, I was unable to add more variation to the Scavenge mission or add different mission types entirely. Implementing these changes would likely boost the replayability for longer and further limit the repetitive feeling.

3.5. Iterative Design

3.5.1. Sprints

Following an iterative design method, I broke down project development into six sprints (Figure 12). Each sprint had a primary goal split into smaller objectives. These helped me avoid spending too long on one area, giving me time to work on others. I kept the sprints flexible, which allowed me to adjust the project to react to illnesses and project changes. Sprints keep me on track and ensured I was organised and met the minimum viable product for the prototype, something I found useful for managing the initial project scope.

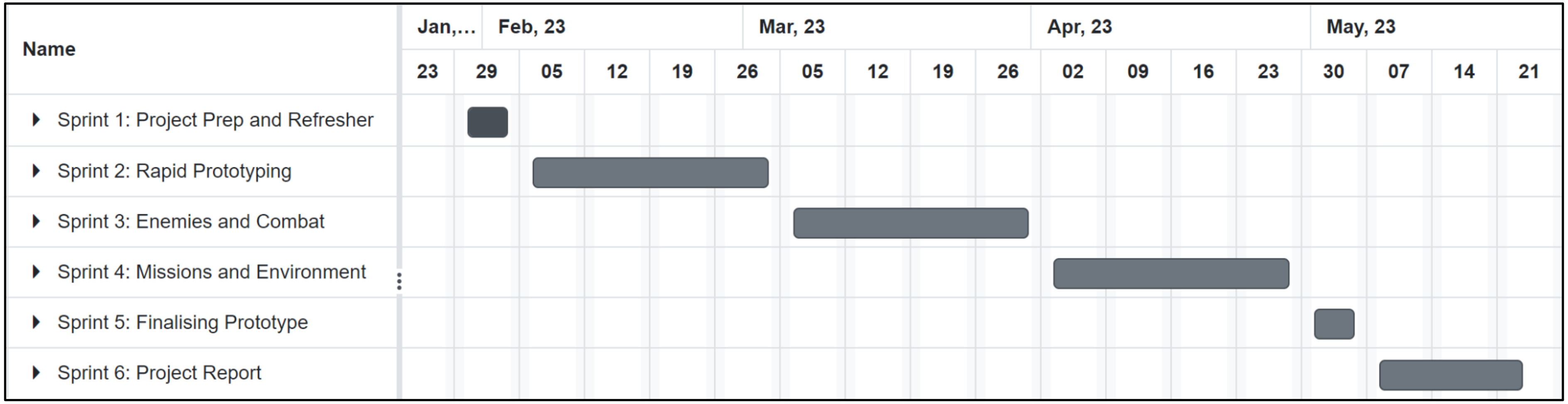

Figure 12: Project Plan

3.5.2. Playtests

I held informal playtests and gathered feedback throughout the project. This proved useful for iterating and improving mechanics and gameplay. I believe the playtests could have been improved with a more structured approach. In the early stages of the project, I didn't have game builds for the playtests as the prototype wasn't ready. This limited feedback I could gather as I relied on pre-recorded videos and in-engine content. This made the playtests more static and often restricted users from playing the prototype themselves. If I were to redo the project, I would create test plans and structure my sprints with a focus on playtesting, allowing room to create more playable prototypes to produce dynamic playtests.

4. Conclusion

Replayability is important in games, especially for longevity and avoiding repetition. One of the strongest factors for replayability was motivation and giving players a reason to keep playing. This prototype used player-led goals to intrinsically motivate players to complete missions and keep replaying for their own reasons. Self-determination and cognitive evaluation theory also proved helpful in increasing replayability and creating a more engaging game.

Strong core loops and supporting gameplay and feedback loops can also improve replay value. These build up core gameplay, which when successful, naturally improves the replayability as players will enjoy replaying the core loop. Randomness can successfully boost replayability by creating more variations and hiding repetition which allows players to play repeatedly with new experiences. Input randomness works best, letting players react to the randomness, ensuring randomness doesn't feel unfair. Procedural generation was useful for displaying randomness and providing new experiences, especially in level design. The version used in the prototype would need further iteration to make it more replayable as it is fairly limited.

To conclude, this project successfully allowed me to explore the themes of replayability in games. Due to the huge range of games and vast differences between them, it is difficult to define exactly what improves or harms replay value. A feature that improves the replayability in one game, may harm the replayability in another. However, some mechanics and gameplay I explored in this project could be applied to similar games or adapted to fit other games after further investigation.

References

- Bartle, R. A. (2004). *Designing Virtual Worlds.* Indianapolis, Indiana: New Riders.
- Burgen, K. (2018). *Three types of bad randomness, and a good one.* Available at: http://keithburgun.net/three-types-of-bad-randomness-and-one-good-one/ [Accessed: 14 April 2023]
- Burke, B. (2014). *Gamify: How Gamification Motivates People to Do Extraordinary Things.* Boston, Massachusetts: Bibliomotion.
- Cherry, K. (2022). *Extrinsic vs. Intrinsic Motivation: What's the Difference?* Available at: https://www.verywellmind.com/differences-between-extrinsic-and-intrinsic-motivation-2795384 [Accessed: 10 April 2023]
- Debus, M., Zagal, J. and Cardona-Rivera, R. (2020). *A Typology of Imperative Game Goals.* The International Journal of Computer Game Research. 20(3). Available at: https://gamestudies.org/2003/articles/debus_zagal_cardonarivera [Accessed: 5 April 2023]
- Design Doc. (2020). *What Makes A Game Replayable?* Available at: https://youtu.be/AX_59mS8mr0 [Accessed: 16 November 2022]
- Despain, W. (2012). *100 Principles of Game Design.* Indianapolis, Indiana: New Riders.
- Eng, D. (2019). *Gaming with Motivation.* Available at: https://www.universityxp.com/blog/2019/10/29/gaming-with-motivation [Accessed: 10th April 2023]
- Fullerton, T. (2018). *Game Design Workshop: A Playcentric Approach to Creating Innovative Games.* 4th Edition. Boca Raton, Florida: CRC Press
- Gałka, P. and Strzelecki, A. (2021). *How Randomness Affects Player Ability to Predict the Chance to Win at PlayerUnknown's Battlegrounds (PUBG).* The Computer Games Journal. 10(1-4). DOI: 10.1007/s40869-020-00117-1 [Accessed: 20 April 2023]
- Game Maker's Toolkit (2018a). *How Games Use Feedback Loops.* Available at: https://youtu.be/H4kbJObhcHw [Accessed: 1 March 2023]
- Game Maker's Toolkit. (2018b). *How to Keep Players Engaged (Without Being Evil).* Available at: https://youtu.be/hbzGO_Qonu0 [Accessed: 16 November 2022]
- Game Maker's Toolkit. (2020). *The Two Types of Random in Game Design.* Available at: https://youtu.be/dwI5b-wRLic [Accessed: 13 April 2023]
- Keith, C. (2010). *Agile Game Development with Scrum.* Boston, Massachusetts: Addison-Wesley.

- Lovato, N. (2017). *How To Perfect Your Game's Core Loop.* Available at: https://gameanalytics.com/blog/how-to-perfect-your-games-core-loop/ [Accessed: 21 April 2023]
- Macklin, C. and Sharp, J. (2016). *Games, Design and Play: A Detailed Approach to Iterative Game Design.* Boston, Massachusetts: Addison-Wesley
- Millard, A. (2019). *How Gameplay Loops Keep You Playing.* Available at: https://youtu.be/Sk-nbAtIUko [Accessed: 16 November 2022]
- Przybylski, A., Rigby, S. and Ryan, R. (2010). *A Motivational Model of Video Game Engagement.* Review of General Psychology. 14(2). pp 154-166. DOI: 10.1037/a0019440 [Accessed: 21 February 2023]
- Reiss, S. (2012). *Intrinsic and Extrinsic Motivation.* Teaching of Psychology. 39(2). pp 152-156. DOI: 10.1177/0098628312437704
- Rigby, S. and Ryan, R. (2011). *Glued to Games: How Video Games Draw Us In and Hold Us Spellbound.* Santa Barbara, California: ABC-Clio.
- Ryan, R., Rigby, S. and Przybylski, A. (2006). *The Motivational Pull of Video Games: A Self-Determination Theory Approach.* Motivation and Emotion. 30. pp 347-363. DOI: 1007/s11031-006-9051-8 [Accessed: 20 February 2023]
- SelfDeterminationTheory. (No date). *Self-Determination Theory - The Theory.* Available at: https://selfdeterminationtheory.org/the-theory/ [Accessed: 23 February 2023]
- Shaker, N., Togelius, J. and Nelson, M. J. (2016). *Procedural Content Generation in Games.* Cham, Switzerland: Springer International Publishing
- Spacey, J. (2016). *What is Replay Value?* Available at: https://simplicable.com/IT/replay-value [Accessed: 28 April 2023]
- Weitze, C, L. (2014). *'Developing Goals and Objectives for Gameplay and Learning' in Schrier, K. (ed.) Learning, Education and Games.* Pittsburgh, Pennsylvania: ETC Press, pp. 225-249.
- Zhang, Y., Monteiro, D., Liang, H., Ma, J. and Baghaei, N. (2021). Effect of Input-output Randomness on Gameplay Satisfaction in Collectable Card Games. 2021 IEEE Conference on Games (CoG). DOI: 10.1109/CoG52621.2021.9619020

Ludography

- Blizzard Entertainment. (2004). *World of Warcraft.* [Video game]. Blizzard Entertainment.
- Bungie. (2017). *Destiny 2.* [Video game]. Bungie.
- Coffee Stain Studios. (2019). *Satisfactory.* [Video game]. Coffee Stain Publishing.
- Colossal Order. (2015). *Cities: Skylines.* [Video game]. Paradox Interactive.
- Firaxis Games. (2016). *Civilization VI.* [Video game]. 2K.
- FromSoftware. (2022). *Elden Ring.* [Video game]. Bandai Namco Entertainment.
- Ghost Ship Games. (2018). *Deep Rock Galactic.* [Video game]. Coffee Stain Publishing.
- id Software. (2020). *Doom Eternal.* [Video game]. Bethesda Softworks.
- Infinity Ward. (2022). *Call of Duty: Modern Warfare II.* [Video game]. Activision.
- Mojang Studios. (2011). *Minecraft.* [Video game]. Mojang Studios.
- Mossmouth. (2008). *Spelunky.* [Video game]. Mossmouth.
- Nintendo. (1985). *Super Mario Bros.* [Video game]. Nintendo.
- Nintendo. (2012). *New Super Mario Bros U.* [Video game]. Nintendo.
- Nintendo. (2014). *Mario Kart 8.* [Video game]. Nintendo.
- PUBG Studios. (2017). *PUBG: Battlegrounds.* [Video game]. Krafton.
- Respawn Entertainment. (2016). *Titanfall 2.* [Video game]. Electronic Arts.
- Riot Games. (2020). *Valorant.* [Video game]. Riot Games.
- SMG Studio. (2015). *RISK: Global Domination.* [Video game]. SMG Studio.
- Telltale Games. (2012). *The Walking Dead: Season One.* [Video game]. Telltale Games.
- Valve. (2012). *Counter-Strike: Global Offensive.* [Video game]. Valve.
- Wube Software. (2020). *Factorio.* [Video game]. Wube Software.