Bachelor’s thesis

Information and Communications Technology

2021

Ville Aro-Heinilä

DESIGNING, PRODUCING, AND TESTING AN IMPACTFUL LEARNING GAME

– Case: Fantastic Factory

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR’S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information and Communications Technology | Game Technology

2021 | 67 pages

Ville Aro-Heinilä

DESIGNING, PRODUCING, AND TESTING AN IMPACTFUL LEARNING GAME

– Case: Fantastic Factory

Games are known to be impactful in many ways. They can tell inspiring stories, similar to literature or films. Although it is still relatively rare in the gaming industry, they can also be made to describe or teach about real-world events and problems.

This thesis forms an impactful game theory and tests it in action. The theory of impactful games combines real-world learning with a mood-changing auditory and visual effect scheme to guide the player to make the right decisions and make the gameplay and learning experience more immersive, entertaining, and rewarding. The theory was formed by analyzing relevant prior studies and game designing principles.

A game called *Fantastic Factory* was designed using the researched elements and produced using Unity Game Engine and other applicable software. The game was aimed at preteens and included tactical resource management gameplay and learning. Three different versions of the game using different amounts of auditory and visual effects were tested with a field test to demonstrate the formed theory and help further development.

The test results indicated that in the right circumstances auditory and visual game effects can be used to influence player behavior in predefined ways, even if the users have prior knowledge and ethical values about the topic of the learning game. The developed game was a success based on the positive feedback gathered from the testing phase, and the company continues the development further.

Keywords:

mobile game, learning game, auditory effects, visual effects, immersion, game design, game theory

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tieto- ja Viestintätekniikka | Peliteknologia

2021 | 67 sivua

Ville Aro-Heinilä

VAIKUTTAVAN OPETUSPELIN SUUNNITTELU, TOTEUTUS JA TESTAUS

– Case: Fantastic Factory

Videopelien vaikutus ihmisten arkeen ja käyttäytymiseen nousee toistuvasti puheenaiheeksi ympäri maailmaa. Aivan kuten kaunokirjallisuus ja elokuvat, pelitkin voivat kertoa inspiroivia tarinoita, mutta niitä voidaan myös mukauttaa sisältämään ja tarjoamaan kuluttajilleen interaktiivista opetusmateriaalia viihteen rinnalla.

Tämän opinnäytetyön tavoitteena oli muodostaa vaikuttavien pelien teoria ja testata se käytännössä. Teoria käsittelee opetusmateriaalin yhdistämistä pelaajan mielialaan vaikuttavien audiovisuaalisten peliefektien kanssa, jotta pelaajaa voidaan epäsuorasti ohjata tekemään oikeita valintoja, ja peli- sekä oppimiskokemuksesta saadaan viihdyttävämpi, mukaansatempaavampi sekä palkitsevampi. Teoria muodostettiin analysoimalla useita aikaisempia aihepiirin tutkimuksia sekä periaatteita.

Tutkittujen materiaalien pohjalta suunniteltiin nuorille suunnattu opettavainen *Fantastic Factory* -mobiilipeli, joka toteutettiin käyttäen Unity-pelimoottoria sekä muita pelinkehitykseen sopivia tietokoneohjelmistoja. Pelistä tehtiin kenttätutkimus, jossa testattiin kolme erillistä peliversiota, jotka sisälsivät vaihtelevan määrän peliefektejä. Pelaajien käyttäytyminen eri versioissa analysoitiin.

Tutkimuksesta kerätty data osoitti, että peliefekteillä oli looginen vaikutus pelaajien käyttäytymiseen, ja niillä pystyi ohjaamaan pelaajien valintoja ennalta määrättyyn suuntaan opetuspelissä. Pelin vastaanotto testaajien keskuudessa oli positiivinen, joka näytti sillä olevan markkina-arvoa, ja sen kehitystä jatketaan julkaisuun asti.

Asiasanat:

mobiilipeli, opetuspeli, audiovisuaaliset efektit, peliefektit, oppiminen, pelisuunnittelu, peliteoria

Contents

Figures

Tables

Glossary

Beta	A test version of developed software that may still contain bugs and crashes.
CIC	Color-In-Context theory. It states that different colors elicit different psychological associations depending on the context.
CPU	Central Processing Unit. The processor of the device.
FPS	Frames Per Second. The frequency at which images are displayed, for example, in a video game.
GBL	Game-Based Learning. The usage of game characteristics and principles within learning activities.
GPU	Graphics Processing Unit. A piece of hardware used to, for example, render 3D graphics.
IDE	Integrated Development Environment. A software application that provides comprehensive tools for software development.
LOD	Level of Detail. A technique that reduces the number of triangles rendered for distant 3D models to lower the load on hardware.
MVP	Minimum Viable Product. A version of a product with just enough features to catch the attention of early customers to provide feedback for future development.
UI	User Interface. A space allowing users to interact with the device and software.
URP	Universal Render Pipeline. A rendering pipeline made by Unity that works on a wide range of devices.

1 Introduction

Making learning more entertaining by using gamification of education has inspired many studies and papers in recent years [1–10]. The influence of auditory and visual feedback in games has been examined comprehensively [11–15], and the possibilities of incorporating auditory and visual feedback into learning games have been tested in some specific areas. Still, the broad question that has not received much attention is whether well-thought auditory and visual game effects can be used to influence player behavior to enhance learning games. This thesis aims to explore and test an impactful game theory by including three main objectives: Research, Develop and Test.

The first objective of this thesis was to research relevant prior studies and principles of games as learning tools, the effects of auditory and visual components, and the core elements of game design to form a theory of impactful learning games.

The second objective was to use the researched theory to develop a casual learning game for mobile devices that balances the gameplay and learning material to keep the users entertained as they are learning while surreptitiously guiding the players with auditory and visual clues. *Fantastic Factory* was developed as an educational game targeted mainly towards young teenagers to teach the effects of fossil fuels and renewable energy. All the stages of development, from an idea to an almost publish-ready title, were documented.

The third objective was to test the game and the effects of auditory and visual components in action using a testing period. Three different versions of the game using different amounts of auditory and visual effects were tested with a public field test which was conducted online to reach a wider audience. Gameplay choices of over 200 players were recorded using Unity Analytics, and willing participants were asked to fill out a questionnaire to give additional feedback.

Unity Game Engine and Unity Analytics were essential parts of this thesis as Punnu Games, the company commissioning this study, uses the game engine daily in development and would like to research the possibility of introducing Unity Analytics into their existing projects.

The following and third chapters of this thesis go over the background information such as prior studies and principles to form the theory of impactful learning games. The fourth, fifth, and sixth chapters showcase different parts of the production phase of the project, from predevelopment to polishing, and the seventh chapter shortly includes the implementation of Unity Analytics to the project. The last two chapters, the eighth and ninth, go over the testing methods and results, concluding the thesis.

2 Background Information

2.1 Games as Learning Tools

The gamification of education and existing learning material is a rising topic worldwide, inspiring many studies and papers in recent years to focus on analyzing and testing theories surrounding it [1–4]. In addition to the gamification of learning material, games themselves can be packed full of references and educational information about the real world, allowing learning to happen simultaneously as the learner is entertained. This learning type is referred to as GBL (Game-Based Learning), and some studies have shown it to be an effective way of enhancing learning motivation and academic performance [5, 6].

Combining playing and studying is believed to make boredom, exhaustion, and procrastination less likely to occur. During globally affected Covid-19 closures, a study carried out at the University of Exeter stated that as many as 70% of university students consider themselves procrastinators and approximately 50% of university students procrastinate consistently and problematically during their studies [16]. Children are also known for their variable attention span, and the use of games in learning – for example, the learning trivia quiz platform *Kahoot* – can improve the overall attitude of children and other age groups toward learning, boosting the scores of students [7].

Most games are made for the entertainment factor, but educational games can twist that fact by bringing in an educational aspect that benefits the users in real-life situations while sustaining the entertainment factor as high as possible. Gamification of learning material can also significantly reduce the exhaustion of longer learning sessions, raise motivation, and make users feel rewarded for their accomplishments [7].

In addition to making learning more entertaining, video gaming generally causes increases of grey matter in the brain regions responsible for working memory formation and strategic planning, fine motor skills, and spatial navigation [8]. This fact may help increase the users' learning capabilities in some instances, making games in all forms fantastic learning tools, if used correctly and in the right amounts. Games can be adapted to many different situations and topics, balancing gaming and studying to target students in various age levels. The Flow Channel featured in figure 1 balances the challenges faced with the players' skill levels, keeping the users interested through the game. Younger learners may require more activity not to feel bored, while a more mature audience may feel overwhelmed by overly gamified materials or GBL.

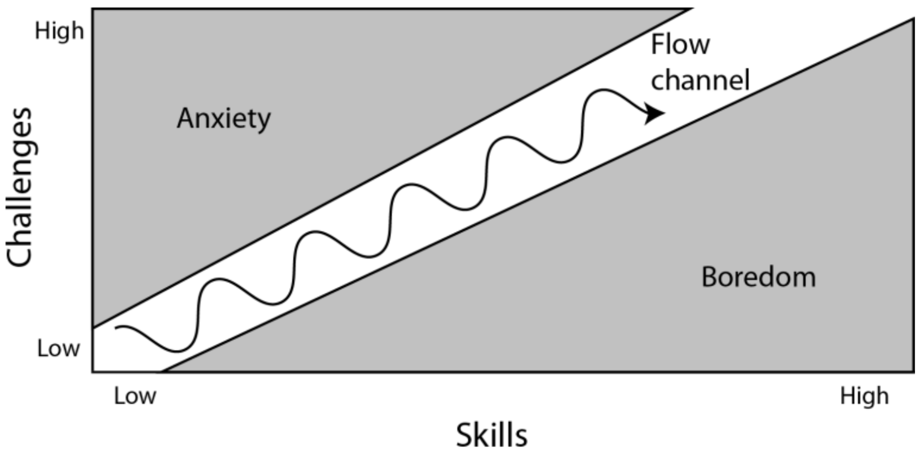

Figure 1. The Flow Channel. From *The Art of Game Design* book by Jesse Schell [17].

Another critical advantage of games in the learning scheme is their ability to evoke emotions – they allow learners to try things that are, for example, too time-consuming, dangerous, or expensive in real life. As an example component of procrastination, escapism is thought to be one of the primary motivators of gaming and can be harnessed for educational purposes in games [9, 18]. The right balance between challenging and struggling keeps the players interested, while the little positive rewards keep them hooked. Well-designed rules and goals motivate the users to continue playing. All these emotions can be harnessed to improve the outcomes of learning [9].

2.2 Impactful Ideas Behind Games

Games are known to be impactful in many ways. They can tell inspiring stories, just like literature or films. And even though it is pretty rare in the gaming industry, they can also be made to tell about real-world events and problems. Most games stay in their specific worlds (e.g., *Super Mario*), and some may remind the players about the real world, but with their unique imaginary events (e.g., *Civilization V*). But on rare occasions, games are made to teach the users something more, including information and skills that are useful in the real world. This powerful aspect of video games has been noticed and used even by the U.S. Army, which has launched a series of first-person shooter games for recruiting, named *America's Army* [10, 19].

Regular entertainment games can positively affect the players' soft skills like teamwork and reaction time. A study conducted by A. Badatala et al. found that participants playing *Modern Warfare 3* cooperatively together scored the highest in other teamwork requiring tasks, while participants playing competitively against each other scored the lowest [20]. However, some entertainment games center around these soft skills and can effectively train the players to be better at them.

Keep Talking and Nobody Explodes by Steel Crate Games is a cooperative game released in 2015 centered around defusing complex bombs. The game requires two players – one who has a bomb and the other who can't see the bomb but has the manual for defusing it. The players must work together to solve all the modules in the bomb to deactivate it before the timer runs out. The game teaches the players to keep communicating clearly under pressure. Professional bomb defusers tested this game in 2018 in a video produced by Buzzfeed Inc. The defusers commented that the cooperative gameplay required and taught good communication and management skills, almost like the real-world equivalent, but without the danger [21, 22].

Killer Flu was created in 2009 by Persuasive Games as an attempt to explain how flu mutates and spreads in a population. The game describes how ordinary flu develops and spreads and how pandemic flu is born and carried in a human population. The players take the role of the flu itself, trying to spread and mutate in a variety of conditions generated by the game. The UK Clinical Virology Network commissioned it to spread awareness of seasonal flu [23].

In 2012, a game named *Plague Inc.* was released by Ndemic Creations. It also focuses on global pandemics and has a complex and realistic set of variables to simulate the spread and severity. During Covid-19, the game got a significant surge in new users. The developers posted a statement that their game should not be used as a scientific model even though it is realistic and informative [24]. The game has been a massive success, reaching over 130 million players globally.

Games like these can really leave an impact on the players, causing discussions and raising awareness of the surrounding situations around the world.

2.3 Impactful Gameplay and Effects

Player experience, performance, and behavior are believed to be enhanced and possibly manipulated using auditory and visual feedback during gameplay events. A game design theory focused mainly on feedback and control of a game can be found in S. Swink's book *Game Feel* [25].

The feedback could be visuals, including animations, color changes, particle effects, screen shake, and other comparable visual game components, but also auditory effects and music. Sensory effects, for example, a device vibration, can also be used [25]. Positive auditory and visual effects in games have often been described as game *"juiciness"* [26, 27]. K. Gray describes games that *"feel alive"* as *"juicy",* and one of the most widely used definitions for juiciness in the game design community is envisioned by J. Juul as *"excessive positive feedback".* According to J. Schell, juicy games are giving *"continuous feedback"* for player actions and *"rewarding the player many ways at once"* [26–28].

Although most feedback does not affect game mechanics, studies have shown that excessive feedback could affect user performance. A recent study by D. Kao researched and tested juiciness levels in an action RPG and found that using none or too many visual effects negatively affected the player experience and engagement [11]. Another study by D. Kao & D.F. Harrell Toward compared the effects of different visual themes changing only the game's UI (User Interface). The study displayed that using more ambiguous visual themes improved engagement and enjoyment but hindered the performance of the users [29]. A pilot study by J. Juul found that a juicy version of a game was well received by the players and the ratings were higher, but the player performance was worse than in a regular version [12].

In contrast, a more recently developed framework for juicy design made by K. Hicks et al. determined through empirical studies that juiciness does influence user experience and motivation but did not affect the performance of the users [13]. The following subchapters go over important points in the concept of juiciness and game effect elements in games.

2.3.1 Colors

Studies have been targeted towards the use of colors, and researchers have developed a framework called the CIC (Color-in-Context) theory [30, 31]. The CIC theory makes specific propositions about the psychological features of color and serves as a starting point to predict color effects psychologically, for example, in game development. The CIC framework has six premises:

1. Different colors have their own meanings and contain psychologically relevant messages (i.e., color is not only for aesthetics).
2. Viewing a color influences psychological functioning (e.g., colors might stimulate behavioral processes, such as approach or avoidance habits).
3. Effects of color stimuli are usually processed without conscious awareness.
4. Color meaning can be learned and inherent (e.g., color associations, such as *”pink is feminine, green is nature”)*.

5. Color perception changes affect, cognition, and behavior, but also affect, cognition and behavior can change color perception. The relation between color perception and affect, cognition, and behavior is reciprocal.
6. Finally, as the term CIC signifies, the color effects are context-specific and can result in different motivational processes (e.g., green is a positive color in a tree, but not on a loaf of bread).

As the premise 3 of the CIC theory states, color information should affect the viewer even without awareness of its influence. For example, the color red has been used for students since early school education to give feedback regarding mistakes and errors, possibly leading to a specific red-failure association [32]. Red also carries the meaning of danger in threatening situations, such as blood or warning signals, and red is used extensively to convey danger-relevant information [33]. A study comparing red and blue avatars in a game avatar found that the red avatar negatively impacted the user experience, resulting, for example, in lower flow and competence. Biologically, it has been hypothesized that the color red is a distractor signal to humans [31].

In this thesis, the premises of CIC theory are used thoroughly in the development of *Fantastic Factory* to convey feedback to player actions and the state of the game world during gameplay.

2.3.2 Audio

Audio is an important component in juiciness, as it can amplify visual perception or even lead to different visual interpretations of an identical animation. A study by R. Sekuler found that participants found an animation clip to be different when presented with different audio effects [34]. S. Wolfson and G. Case found that combining the color red with loud audio gave the players the feeling of excitement [14], and S. Hébert et al. concluded that game music could also influence the psychological responses of the players [15]. Matching audio appropriately to visuals is essential to convey the intended meaning.

Most commonly, game audio has been divided into the following categories: speech, sound effects, and music, which derive from the workflow of game audio production, as each of these three types of game audio has its own process during production. As told by J. Fiberg [35], audio in games can be further classified into the following categories:

1. Avatar sounds (i.e., the sounds generated by the player character, such as footsteps or other activities)
2. Character sounds (i.e., the sounds of non-playable characters)
3. Object sounds (i.e., the presence of objects in the game world)
4. Ornamental sounds (i.e., the sounds that enrich the game's atmosphere, such as ambient music and sounds that are not necessary for conveying gameplay information but add to the complexity and fullness of the game)

The game made in this thesis uses the ornamental sounds classified by J. Fiberg with their full potential. As the game world status changes, the ornamental atmospheric sounds of the soundscape also change accordingly to further enhance the feedback of the player actions on a bigger scale in addition to the immediate feedback made by other sound effects.

2.3.3 Feedback

The placement of feedback is also to be designed with the specific context in mind. Research on feedback interventions has shown that the effects of feedback depend on many factors such as the task novelty and complexity, personality, and the frequency and type of feedback [36]. Juiciness is mainly seen as a positive factor in entertainment games, but pre-existing work provides contradictory evidence in the way learning games are supposed to work. While juicy visuals may be more engaging, they may also hinder the performance of the learners in some cases – researchers are referring to these distracting visuals as seductive details [37]. High cognitive load environments require much thought, and in these situations, seductive details can interfere with the user’s problem-solving abilities with these three problems [38]:

1. Distraction (i.e., Juiciness may cause user attention to move away from the relevant information)
2. Disruption (i.e., Constant stimulation might make it harder to create correct mental schemas)
3. Diversion (i.e., May bring up prior knowledge that is unhelpful)

The game made in this thesis takes these details into consideration and reduces the seductive details when the players are given important information or presents the information when the players are not in an action-packed situation.

2.4 Combining the Elements

By combining an interesting, deeper real-world meaning and using effective auditory and visual choices including color and audio theories to add fitting juicy effects in the right amounts that make the game entertaining to play, both the learning experience and entertainment potential of games can theoretically be maximized. A well-thought game idea that fits the specific context and can include learning material without making any part of the project feel out of place is required to make the combination work.

The combination of scientific and informational lessons with juicy effects has been proved popular by social media and its viewers. Youtube channels such as *Kurzgesagt – In a Nutshell* and *The Infographics Show* have both reached over 11 million followers on the platform by producing informational animation videos about vastly varying topics. The videos are filled with references and well though auditory elements and visual effects to make learning enjoyable for the average viewer – and it has worked out for them [39, 40].

3 Game Design Principles

To make an enjoyable game with exciting and rewarding gameplay that keeps the player wanting more, developers usually follow some basic game design principles. However, the game industry is a relatively new field of science. The theory of games is still constantly evolving – video games are a broad art form that includes visual arts, stories, and music, so creative freedom allows massive differences from game to game. Also, the genre of the game changes the needed mechanics and gameplay elements massively, so everything in the game design must fit the player's intended experience [41].

3.1 Foundation

The foundation of game design can be compressed into three elements that each have distinct goals and meanings – player, communication, and appeal. The players need to have a purpose in the game, and that is where meaningful play comes into the picture. K. Salen and E. Simmerman state that meaningful play is the process by which a player takes action within the game, and the game responds to the action. The relationship between action and outcome shapes the meaning of an action in a game [41].

3.1.1 The Player

The player's role is to move the game forward with their actions. Every game lets players take actions and assigns outcomes to those actions – this includes instant, juicy feedback and also the outcomes in a larger context of the game. As stated by K. Salen and E. Zimmerman, meaningful play happens when the actions and outcomes are both discernable and integrated into the whole context of the game [42]:

Discernable means effects that the player can perceive instantly – juicy or not, it showcases information to the player in real-time. Without discernability, the game does not feel meaningful to the player, and the player quickly starts to feel powerless and bored and usually stops playing.

Integration covers the larger context of the game. The player actions not only have immediate feedback or significance but also affect the outcome of the whole game and may change the experience at a later point in the game. An excellent example of this is the game of chess. Making simple tactical moves in the early game can make significant moves possible in the late game. E. Rubin made an analysis of chess tournaments played between 1895-1970 and found that meaningful play is so strong in chess that 96% of wins were achieved by the resignation of the opponent because they knew they would lose [43]. Integration is a powerful tool in game development when used right.

The game made in this thesis features both of these aspects when teaching the player about different energy types. Making poor choices (e.g., choosing fossil fuels) shows instant feedback with sounds and visual effects, and integration follows as the game world around the player darkens as pollution takes over it. The player is in complete control of these outcomes by their choices. Theoretically, the juicy feedback should guide the player in the right direction, but no choices are forced on the player.

3.1.2 Communication

Discernability and integration also fill the communication aspect of the game design foundation – the player needs to know how their actions are helping them to reach the goal. For example, the player hits a stone wall with a pickaxe. Discernability tells the player what happens at the moment (Pickaxe collided with a wall, sparks were shown, and a crack appeared on it), and integration lets the player know how it affects the rest of the game (The wall probably breaks if it is hit many times, there might be something hidden behind it) [42].

3.1.3 Appeal

The last part of the foundation is the appeal. The game needs to appeal to its designed target audience. The appeal varies from player to player and from game to game, so it may be hard to pinpoint precisely during the concept phase. For some players, the main appeal is breathtaking graphics with fast action, and for some, it is an immersive story [41]. It is essential to start every game project with a prototype that can help test the idea of the game quickly in action to see if the appeal is there for the target group. If the prototype does not feel right, the game idea needs further refining and finetuning.

3.2 Defining Rules

Rules in games are usually strict and rarely change during the gameplay – or if they change, they change logically and have a purpose of doing so. Good game design has **believable events and behavior,** and every event in the game needs to happen to the logic and expectations of the player. Randomness can also be logically expected in some parts, for example, how enemies are appearing in the surrounding game world, or in some cases, how the whole world is generated around the player [44]. In single-player games, the player may also have small ways to affect the rules, like setting a difficulty level in the settings. In games that involve many players, the rules may also be set from the way the players interact with the game, and additional rules can be agreed on. Rules also affect the amount of luck players experience in a game – strict rules leave no room for unexpected events. A game of perfect information does not have luck involved. For example, the possibility of solving a Rubik's Cube at random is nonexistent. In multiplayer games, strict rules cannot overpower the randomness of the players. Chess does not have luck involved in the mechanics of the game, but the dueling of two players brings surprises back to the play. There are so many possible options presented to the players that the course of the gameplay is impossible to predict beyond a few moves [42, 44, 45].

Expectations are usually based on the real world, where **physics** always works the same expected way. A game world can have its own laws of physics and events that differ drastically from the real world, but they need to stay consistent with the rule set the player learns during gameplay. Many games try to mimic the physics of the real world as a base but add their own specialties to make the game more fun, for example, double jumping or unrealistically low friction when sliding down a hill. Cal Jeffrey discussed the importance of physics in video games in an article by mentioning that most developers bend the rules for the sake of fun – if realism hinders the entertainment factor, a proper balance of physics has to be found. Striking the right balance is the responsibility of the developer [46].

The audiovisual design also needs to follow these rules and be consistent and fitting to a specific game world – the player should understand what is happening even when they close their eyes. Vice versa with the visuals if the game audio cannot be heard. Just as with physics, these aspects do not have to be realistic, but they need to stay consistent with the expectations as both affect the perception of the game world [34, 47].

Pacing is also a great and mandatory tool to use when building a game world, as it can change the feel of urgency dramatically. It is the rate at which gameplay events happen or are repeated, and spreading events out to mix high concentration and urgency can produce an exciting experience for the player. For example, puzzle games are high concentration, but if a timer is added, they can become urgent and completely different experiences [48].

If only one event happens at a time, the gameplay may feel too shallow to the player, and interest is lost. A game should have **overlapping events and behavior**, allowing more than one event to occur at any given moment, from simple details like counting points while the player is doing a task to significant details like allowing non-linear gameplay. Non-linear gameplay allows the player to choose and make alternate choices around the main focus of the game. Human beings like to feel like they are in control, and the ability to have a say on things happening is satisfying [49]. Players want to have choices, but they should never be allowed to guess what they should actually focus on, and games need clear **focus points** that guide the player to the goal of the game. This detail applies to both visual and technical aspects of gameplay.

A game should not feel unfair, and **anticipation** is the key to addressing that. When something is about to happen (e.g., an enemy is going to attack), some time is needed to inform the player (e.g., the enemy moves its weapon backward to start charging at the player). **Communicating all changes** to players is important, and the size and frequency of the change determine how it should be communicated. These changes are usually told between anticipation and the events themselves [41]. The communication may range from small visual clues (e.g., the enemy moves its weapon backward, the weapon starts glowing, enemy charges at the player) to more noticeable notifications (e.g., the player completes a quest and a popup window shows up with a sound effect).

Practical game design principles combined with the researched effects of auditory and visual feedback form an impactful game theory, which is the base of the plan made for *Fantastic Factory*. The following two chapters go over the development phases of the game.

4 Predevelopment

When resources are limited, a solo developer must adapt and design the game accordingly. New game developers are often inspired by the big game titles they like to play themselves, which can often lead to the project becoming too complex, and the developer losing interest or going over time limitations. This hindrance applies to both coding challenges and too complex aesthetics. As a time-limited game designer, one must create a balance between all the components so that every aspect wastes no time but is good enough to not feel out of place [50].

4.1 Minimum Viable Product

MVP (Minimum Viable Product) is a development technique in which a new product is introduced to the users with only the basic features implemented but with enough content to get the attention of the consumers. The idea is to get feedback from the initial consumers and testers, which helps make the desired changes to the final product before releasing it to the public [51, 52].

It is essential to take note of the limitations and requirements already available when planning the project. The game made in this thesis had many such constraints, which shaped the MVP approach – create just enough content to get the game to the testing phase to demonstrate its value on the market without wasting time. The first version of *Fantastic Factory* needed to fit modern mobile devices using the Android operating system. The company commissioning this study continues developing the game with the collected testing feedback, implementing more content and monetization to reach the game's public launch.

4.2 Target Audience

The game was targeted mainly towards preteens (Children around 13 years old). The idea was planned to be simple and easy to understand but still stimulating enough for the scientific curiosity and tactical awareness of rapidly learning children. The game is also fit for the parents of the children, allowing families to play, compare their scores and talk about the lessons and tactics learned while playing. The gameplay was designed advanced enough not to be entirely suitable for small children.

4.3 Idea

When brainstorming a game idea, the designer should scribble down ideas whenever the inspiration strikes. In a world filled with already good game titles, big and small, getting inspiration from other games and ideas that are already used usually happens a lot. Genuine innovations are few and far, and usually, new games use old ideas mixed together or add something new to an old base idea. Failures are also entirely acceptable [26]. What came to be known later as *Fantastic Factory* was pitched with this initial gameplay description:

> ”The main idea behind the game is nature, fossil fuels, and renewable energy. In the game, the player makes choices that have different effects on the progression to the main goal, but at the same time, the surroundings. The game could be a Clash of Clans style city-building experience, where some choices might help the player to move forward a lot, but at the same time pollute more than other choices. The choices could be, for example, different types of buildings or materials. The player has to find the right balance to get the best score, promoting thoughtful play. The player's choices can be seen through the visual style of the game. Polluting makes the game's visuals and audio change to be more dark and depressing while being nature friendly keeps it playful and colorful. These choices are anonymously collected to cloud database.”

The feedback was mostly positive but at the same time doubtful, as the description did not go into detail about how the game loop or other important details worked in action. The short development time was brought up, as an extensive city building experience could take too much time with a single developer, and the lack of learning material in a learning game was questioned. A constant feedback loop speeds up the development drastically, and the initial idea was quickly refined. The project prototyping phase was accepted to be started with this description:

> "Project Fantastic Factory: The player controls a factory, and the goal is to get as wealthy as possible in a year with clever choices. Game time is about 20 minutes per session. The phone is kept in a portrait position, and the game camera is from an angled top view position. Graphics are 3D, simple, and colorful cartoon-style optimized for mobile devices. The player has a building grid where buildings can be placed and upgraded, and different types of buildings have their own purposes, including energy production, product manufacturing, and storage. The player may also choose the materials to use, how to recycle waste, and how products are delivered from three stationary buildings placed automatically on the sides of the building grid. For example: choosing fossil fuels may produce more energy and products but also pollute more. At the end of the game, the player is represented by their performance data and rewarded accordingly. The game includes learning material that is unlocked according to the player's performance. The game collects anonymous user behavior from in-game events for further development and balancing."

4.4 Game Engine

The game engine was chosen to be Unity. It was already familiar, known to be easy for indie developers and mobile games, and has built-in Unity Analytics, making the implementation of in-game data collection easy and, most importantly, fast. The company commissioning this study wanted to test the possibility of using Unity Analytics data collection in their existing games, which were also made with the Unity game engine.

Unity is a game engine developed by Unity Technologies, and it is one of the most widely used engines in the game development industry. Unity has stated that more than 70% of the top mobile games are created with Unity, reaching over 2.4 billion devices quarterly [53]. It is a cross-platform engine used to create unified software and games that can be deployed across over 25 leading platforms to reach a large audience. It is known for modularity and the possibility of adding various plugins. It has its own IDE (Integrated Development Environment), which contains many elements like Assets, GameObjects, Components, Scenes, and Prefabs. Unity is a visual editor and supports the C# programming language natively. Unity can read popular 3D-file formats such as FBX, DAE, 3DS, DXF, and OBJ. Audio file formats such as AIF, WAV, MP3, and OGG, and most common image file types such as BMP, TIF, JPG, and PNG are supported. Arbitrary data can be imported using, for example, JSON files [54].

4.5 Test Devices

Three smartphones with different specifications were used to test out the game and to keep track of the functionality and performance of the project during development. This setup allowed for better compatibility of the project when the public field-testing was conducted, decreasing the possibility of errors in the collected data. Main functionality was tested using a Oneplus 7 Pro (2019), featuring a 6,67”, 1440 x 3120, 90Hz, 19.5:9, ~516 PPI (Pixels per Inch) display, Qualcomm SM8150 SoC (System on a Chip) with 8GB of RAM (Random Access Memory) and Android 11. Additional performance and compatibility were tested using two older Android devices with their corresponding specifications being: Xiaomi Black Shark (2018), featuring a 5,99”, 1080 x 2160, 60Hz, 18:9, ~403 PPI display, Qualcomm SDM845 with 6GB of RAM, Android 10. Unihertz Jelly Pro (2017), featuring a 2,45”, 240 x 432, 60Hz, 16:9, ~201 PPI display, Mediatek MT6735 with 2GB of RAM, Android 8.

The feature that affects development the most is the wildly varying resolution and aspect ratio of the displays. Unity has automatic canvas scaling settings, which are used to match the aspect ratio of the used display. Still, the UI elements need to be designed to work as intended without overlapping each other. The resolution and refresh rate of the displays affect the amount of rendering optimization needed – most of the mobile devices sold today have such a high resolution and refresh rates that rendering the 3D graphics at a native resolution and framerate would use too much power and cause thermal throttling [55].

The power of the chipset and the amount of RAM varies significantly between different phones. Still, comprehensively mobile-optimized low poly assets and in-game lighting settings allow the game to work even with low specification devices. It is still preferable to have adjustable in-game settings to allow users to change to a lower set of graphics if their devices are underperforming. For example, dynamic lights and real-time shadows can look great but are often too heavy for older mobile hardware to get consistent performance [55].

5 Development

5.1 Early Prototype

An early prototype was developed using placeholder assets to showcase the planned base UI, gameplay area, and the basic functionality of constructing and managing buildings. Two selectable buildings were included in the prototype for the player to construct and manage. From the three stationary management buildings, one was functional on the base scale so the player could sell products. The base gameplay was working, shown in figure 2.

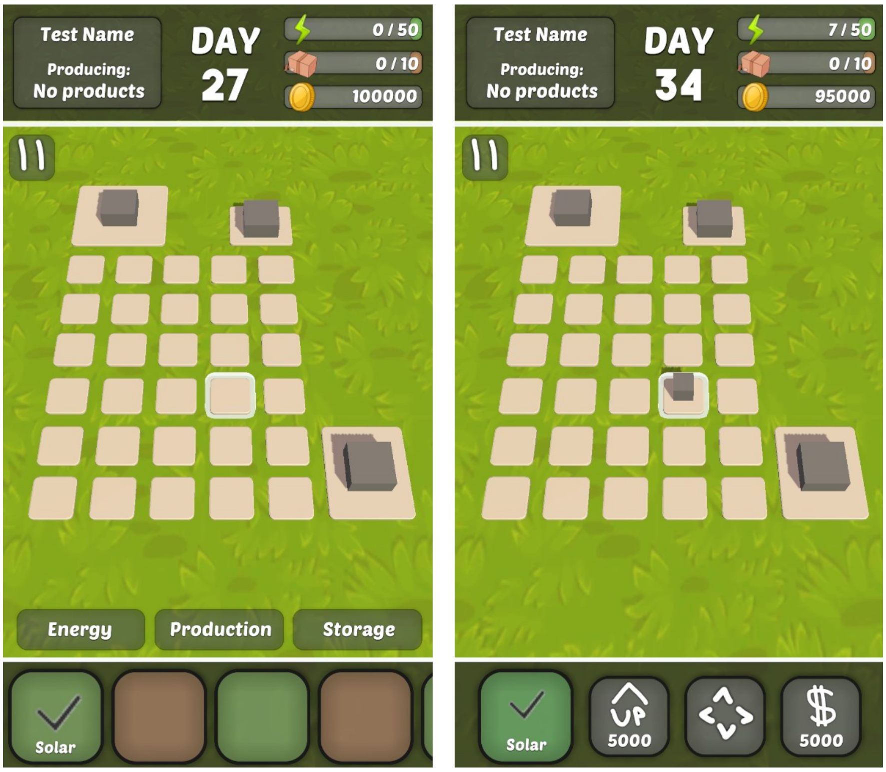

Figure 2. Gameplay captures from the prototype.

The base elements of the UI were easily understandable, and the control scheme intuitive enough to tell that the core gameplay was functional and the possibilities of deeper gameplay were presented if the elements were to be refined further. The prototype proved the concept of *Fantastic Factory* working and functioned as a base for the entire development phase.

5.2 Core Mechanics

Core mechanics refer to the basic set of functionality in a game system that produces the core interactions which the player repeats during gameplay [42]. These rules rarely change during gameplay, and in *Fantastic Factory,* the rules stay the same the whole gameplay loop. The core rules include:

- The player is always in control of the camera and the buildings in the gameplay area, and the gameplay is controller by tapping GameObjects and UI buttons.
- Player receives money by selling products or energy, which is used to enhance the factory to make manufacturing faster.
- Each building has statistics that never change. These stats include, for example, energy generation and usage, product manufacturing, and pollution amount.
- The game includes achievements to drive the player to try different play styles. By completing achievements, the player unlocks short teaching materials to read.
- Game time is limited with a timer. The player can speed up or pause the game but cannot get more time. The game ends when the time limit is met.

5.2.1 Factory Grid

All the objects in the game scene inside Unity are referred to as GameObjects, which are containers for different components such as the objects transform position [54]. The Factory Grid is the main gameplay area the player manages their factory on. It includes 33 GameObjects which are referred to as slabs.

The player interaction with the Factory Grid works with raycasting. Touches on the device's screen trigger a ray to be cast from the scene camera to the point of touch. Ray collisions with GameObject colliders in the scene are tracked. A ray is not cast to prevent accidental selections when the camera is moved.

Optimizing Raycasting and Colliders

Meshes are the main graphics primitive of Unity, and they define a polyhedral 3D object's shape using vertices, edges, and faces, which can be seen in figure 3. Collider components define the shape of a GameObject for physical collisions and are invisible to the player during gameplay events [54].

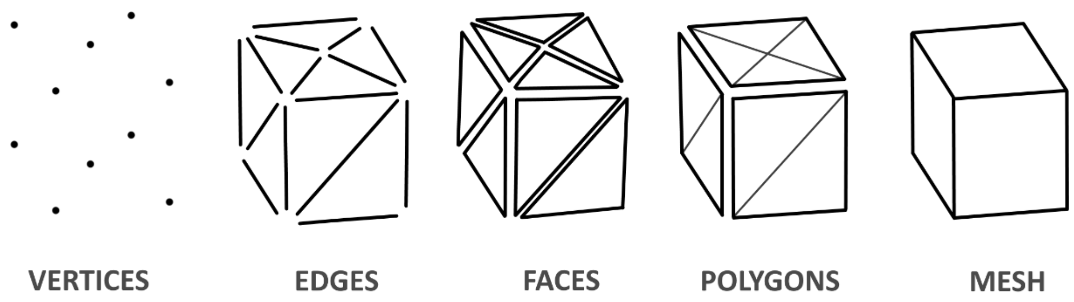

Figure 3. Elements of a polygonal mesh model.

Mesh colliders match the shape of the GameObjects mesh exactly using the stored mesh data, which creates more precise and authentic collisions [54]. As a negative, this is much more processor-intensive than primitive types of colliders and requires higher processing overhead. Mesh colliders also have some limitations – they only collide with primitive colliders, not other mesh colliders, and a GameObject cannot have a regular mesh collider and a Rigidbody component active simultaneously. A Rigidbody is a component that allows a GameObject to be affected by simulated forces, for example, gravity. The limitation of not being able to use Rigidbodies can be negated by setting a mesh collider to be convex, which limits the collider to a maximum of 255 triangulated faces.

Primitive colliders include box colliders, sphere colliders, and capsule colliders which consist of simple calculations [54]. Primitive colliders can be combined in a single GameObject to create a compound collider shown in Figure 4. Compound colliders can approximate the shape of a GameObject while keeping a low processor overhead, allowing lightweight but comprehensive collision detection for complex objects.

Figure 4. Compound collider examples. Human character (left, not used in the game) and Factory Grid (right).

Colliders in Fantastic Factory

The slabs use primitive box colliders that lessen the performance impact of raycasting and fill the gaps between the slabs on the Factory Grid to be more forgiving for the player touches (Fig. 4). As a simple mobile game, no accurate collision detection is required in this project. All movements of GameObjects in the game scene were decided to be calculated manually through code, requiring no Rigidbodies to simulate gravity or other forces.

5.2.2 User Interface

A video game UI is an interface with components that help the players to navigate, find information, and finally accomplish goals in the game. UI design requires close attention to detail and functionality to fit the requirements of a specific game type [56, 57].

The design of the UI in *Fantastic Factory* was inspired by popular “City Builder” mobile games such as *Clash of Clans* developed by Supercell in 2012, which have made users expect various features and styles of the UI in similar types of games. The elements such as resource indicators and player profiles are located on the top partition of the screen. In contrast, essential buttons of the gameplay are located at the bottom, which can be seen in figure 5. Buttons are made to be big and spaced out for reliable interaction. *Fantastic Factory* is played holding a mobile device vertically, so the UI needs to be mainly optimized for single-handed use [57]. All the elements scale with the aspect ratio of the user device’s display.

Figure 5. A gameplay capture of *Clash Of Clans* by Supercell.

Top Bar

As the mobile device used to play the game is held vertically, the top part of the screen is easy to read but also hard to reach. It is an optimal space for showing the player important feedback and statistics of the gameplay, such as popup notifications or the available resources. The top bar shown in figure 6 showcases important statistics to the player but does not include any elements the gameplay requires to be interacted with. Information is shown to the player using text elements, resource bars, and icons with high contrast to maximize visibility [56].

Figure 6. The top partition of the game UI.

Bottom Bar

The whole bottom part of the screen is possible to reach with a single thumb while playing one-handed, so it is an optimal place to include most of the interactive UI elements for easy access [56, 57]. The bottom part of the UI changes according to player selections and includes all the available actions to the player with simple buttons and elements, which can be seen in figure 7.

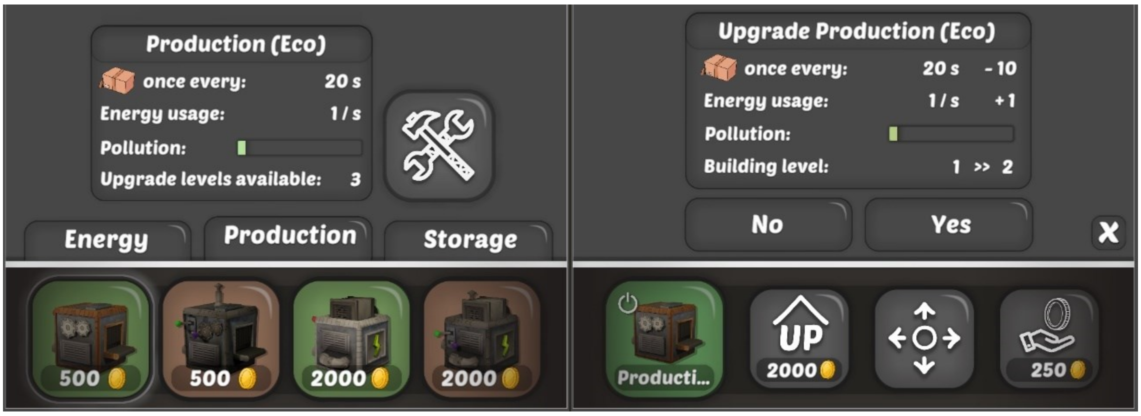

Figure 7. The shop tab (left) and the management tab (right).

Main Menu and Static Building Menus

The menu systems in a game should be consistent and use common UI elements between each other. By creating common patterns in language, layout, and design between different menus in the UI, users can learn to use the whole system quickly and feel more comfortable using it [56]. All of the menus in *Fantastic Factory* follow the same common patterns of design, seen in figure 8.

The overall style of the UI is to keep it as simple as possible and only show the necessary info to the player at once. This optimization is known as the KISS design principle (Keep It Simple, Stupid!) [58]. Simplifying UI elements for a game that specifies in resource management was challenging and turned out to be very time-consuming. Other information is darkened and deactivated whenever a menu has priority to simplify the user experience [57].

Figure 8. The main menu, settings menu, and two static building menus.

5.2.3 Buildings

The buildings in the game are stored with Unity's Prefab system. The Prefab system allows storing a GameObject complete with all its components, property values, and child GameObjects as a reusable Asset [54]. Each building Prefab stores a 3D model and visual effects such as particles and is equipped with a script property that includes all the statistics required for the in-game systems to work, for example, the Buildings ID.

Player Buildings

Player Buildings are buildings the player has complete control over. The player may choose to buy, upgrade and sell these buildings at any time during gameplay using the shop and the management tabs shown in figure 7. Each building has different statistics that the player can tactically manage to progress through the game. The building prefabs are instantiated on the play area during construction and upgrading and tracked using the script properties.

Static Buildings

Static Buildings are buildings the player has little control of and are instantly instantiated on the play area when the game starts. When these buildings are selected, a specific popup window shows up according to the static building selected, seen in figure 8.

For the purpose of the MVP, the number of player buildings was selected to be 12. Each player building had three upgrade levels making the total amount 36. The number of static buildings was selected to be eight. In total, the game produced in this thesis had 44 different buildings to be used.

5.2.4 Camera

The game scene is viewed from an angled top-view position, and during gameplay, the player is always in control of the camera. Orthographic and perspective projections were both tested, shown in figure 9, and perspective projection was chosen to be used as it made the game world feel more dynamic and natural.

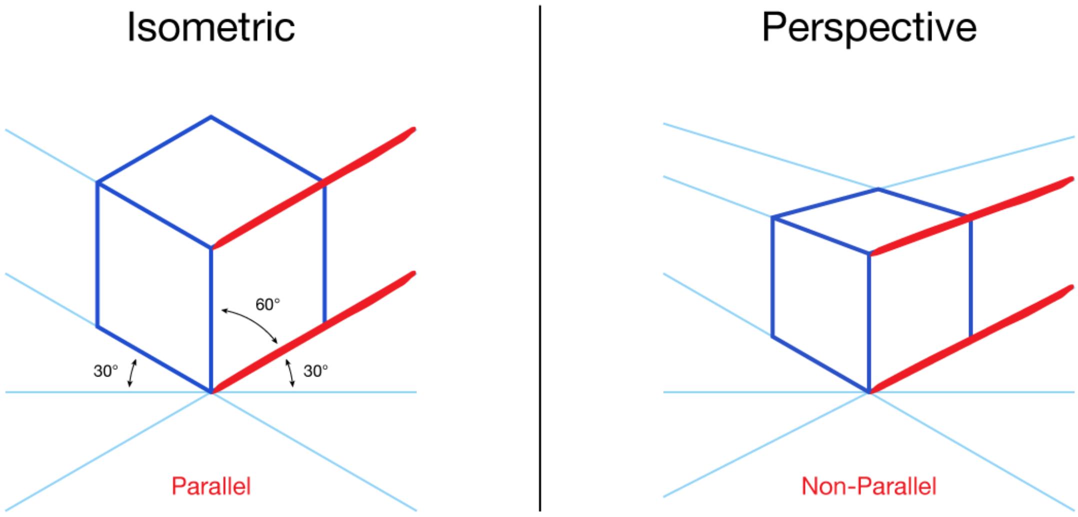

Figure 9. Isometric/orthographic (left) and perspective (right) projections compared. Figure by Majo Puterka, 2020 [59].

The camera features some basic controls seen in other touch screen applications. Holding one finger on the screen activates basic moving. The movement amount of the camera is calculated from the change of the finger touchpoint. Holding two fingers on the screen activates pinch zooming and rotating. Zoom amount is calculated from the distance change between the two touchpoints, and rotate is calculated from the location change.

Some limitations were required to keep the camera in the gameplay area. The default position of the camera is saved, and when the distance between it and the current position is over the maximum range, and there are no fingers on the screen, the camera is moved back to the maximum range. The limited space of the device's screen works as the limiter, making the camera usage feel smooth.

5.3 Advanced Mechanics

The advanced gameplay mechanics in the project include specific systems that enhance the feedback the game gives to the player. These mechanics are not necessarily needed for the game to be playable but affect the whole gameplay and learning experience and add the researched impactfulness to the game.

5.3.1 Adaptation to Player Actions

Fantastic Factory slightly adapts to the actions the player makes. This adaptation does not change the core mechanics of the game but changes the graphical style of the 3D environment.

Each building has a pollution value, which is shown to the player while constructing buildings. The more pollutive the player is, the darker the game world around the factory area becomes. Small changes like air pollution can be corrected by the player by changing the earlier choices. Allowing too much pollution causes nearby nature assets to be changed, which is a permanent change to the background graphics.

This adaptation is controlled by a sliding value that changes from the pollution values the player chooses during gameplay. The sliding value controls the game scene Volume Framework, which is featured in the URP (Universal Render Pipeline) of the Unity Game Engine [54]. A Volume component references a Volume profile containing the properties that URP uses to render a game scene. Volume profiles can be smoothly interpolated, making scene changes feel natural to the player. Different effects, such as the randomized death of surrounding trees, are activated in code once a certain level of pollution is reached. This adaptation tries to make the players feel that their actions have consequences on a larger scale.

5.3.2 Score Calculation

The score calculation at the end of the game features factors that the player's actions change during gameplay. The game rewards the players for future-proofing their factories (i.e., using renewable energy), which is told at the start of the game to be one of the goals of the factory. Using fossil fuels may be beneficial in the short term to maximize profits, but in the end scoring these buildings return only half of their value. The ending screen featured in figure 10 showcases the summary of player statistics when the game ends.

5.3.3 Achievements

Fantastic Factory combines the rewarding feeling of unlockable achievements with learning material. The learning material featured in the game is locked behind gameplay achievements, which in this project are named *Logbook Pages*. The criteria for the logbook pages to be unlocked matches the learning material stored inside, making the material feel like an expansion of the game world.

Figure 10. The Logbook (left, middle) and the ending screen (right).

The following chapter goes over how the assets featured in the project were produced and optimized for mobile devices.

6 Game Graphics and Audio

6.1 Overall Style

The overall graphical style of this project had a couple of requirements. It was to be cartoon stylized and fast to produce. As a mobile game, the game assets needed to be optimized for a large variety of mobile devices with vastly different amounts of computational power. These requirements led the 3D models to use a strict low poly asset approach. Low poly models feature polygon meshes with a relatively small number of polygons making up the 3D shape, which is an important factor for optimizing game performance. Existing assets like textures were to be reused as many times as possible to allow in-engine optimizations and reduce the amount of work needed. The 2D art of the game followed the limitations set by being also minimalized. All of the user interface base elements were to use simple shapes that could be reused in many parts of the UI.

6.2 Asset Creation

All of the assets featured in *Fantastic Factory* are original and made for this specific project. Blender was used to make the 3D models. Blender is an open-source 3D creation suite free for everyone and supports the entirety of the 3D pipeline – modeling, UV unwrapping, texturing, rigging and skinning, animation, simulation, rendering, compositing, and much more [60]. The 2D elements were drawn and edited with Krita and Paint.NET, which are both free software capable of the 2D raster graphics workflow.

6.2.1 2D

The 2D raster graphics were made to be as efficient as possible. The elements are saved as PNG files, as it is a lossless format. Unity imports textures into a number of possible GPU formats, which the developer has control over in the import settings. The GPU format defines the texture’s size in the final application, so the original file size does not matter – only the quality does [54]. Raster image formats such as JPG, GIF, and PNG are optimized for smaller file sizes, and that comes with the cost of needing to decompress the whole image to get the color value of any single pixel. All GPU texture compression formats have fixed compression ratios, meaning a single size texture (e.g., 512x512 pixels) always has the same memory size regardless of the content. Almost all GPU compression formats use 4x4 block-based compression, meaning they can only compress images that are multiples of 4x4 pixels. The compression allows GPU formats like DXT, ETC, PVRTC, and ASTC to be optimized for fast random reads, and the color value of each pixel is quick to decompress [61].

UI Elements and Icons

The entirety of the UI is constructed using a single UI element texture which is reused in different sizes and transform locations (Fig. 11). 9-slicing the UI element allows it to be reused in various sizes, keeping the UI elements in proportion [54]. The UI icons in the project are separated into three subcategories – clipart icons, cartoon icons, and object icons. Clipart icons are completely white and flat and can be adapted to many locations when needed, and cartoon icons feature detailed hand-drawn elements. Object icons are rasterized images of the existing 3D models. These elements can be seen in figures 6, 7, 8, and 10.

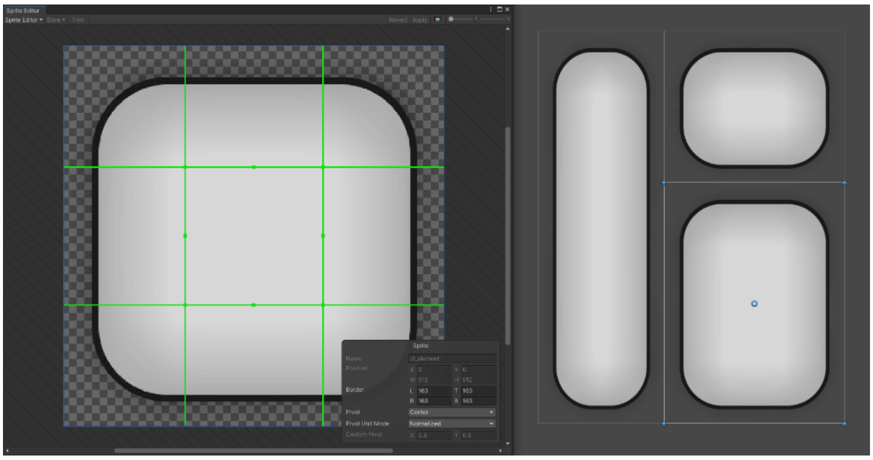

Figure 11. A 9-sliced UI element sprite.

Texture Atlases

Textures for different parts of the 3D models are stored as texture atlases, which contain many different types of textures in a single image. Figure 12 shows a texture atlas being used in Blender. These atlases are reused between many of the models in the game, which lowers the required amount of different textures.

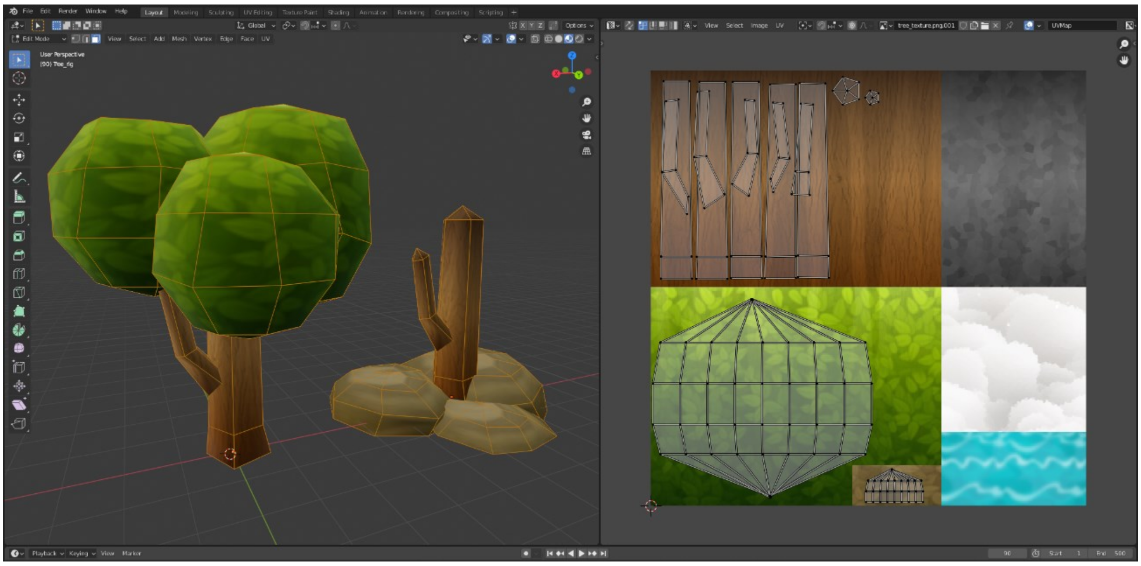

Figure 12. A low-poly tree model unwrapped to use parts of a texture atlas in Blender.

6.2.2 3D

The 3D models are modeled and UV unwrapped using Blender, and imported into the Unity project as OBJ and FBX files. OBJ is a much simpler file format that only stores the geometry and UV data and, as such, is fit for static objects. It is more compatible with creation tools than FBX. FBX is a more advanced file format that can hold much more data, including animations and materials, and is overall more capable than OBJ but at the cost of bigger file sizes. Unity can also import project files from many of the 3D creation tools available in the industry, including Autodesk Maya, Cheetah3D, Blender, and more. These proprietary files are converted to FBX files as a part of the import process. However, Unity does not recommend using proprietary files to keep big projects running smoothly, and as such, they were avoided during the project [54].

Surroundings

The background of the play area is filled with static nature assets that make the world look green and alive. Objects are created using simple shapes. Faces that the player never sees, for example, the bottom face of the tree trunk and faces inside the leaves, are removed to cut the number of polygons to the minimum. The tree model featured in figure 12 has 140 polygons, which turn into 232 tris when the object is triangulated during the import process in Unity.

Objects further away from the factory area are never seen from a close distance and have lower LOD (Level of Detail), which can be seen in figure 13. This lower LOD is achieved by decimating the geometry, which simplifies the mesh by collapsing nearby edges together [55].

Each vegetation asset has a “dead” variant activated if the pollution level reaches a certain amount during gameplay, seen in figure 12. This change in the scene quickly changes the mood of the game in combination with the other changes the game makes, like adding a color tint to the textures and changing scene rendering styles from Volumes.

Figure 13. Far away objects in the scene have a lower level of detail.

Buildings

Most of the building models feature basic elements that are reused in every asset. Every building is modified from a base cube that is rescaled and textured differently for each building. This consistency gives the buildings vastly different looks but still makes them fit together as they feature the same base building pieces. This way, they were also fast to produce. Figure 14 compares different renewable and fossil buildings (upgrade levels 1 to 3).

Figure 14. The object icons of some of the buildings.

Fossil fuels and renewable energy have their own design principles, which are followed throughout the assets. Renewable energy buildings feature much simpler and modern-looking shapes, whereas fossil energy buildings are filled with more angular shapes and extruding parts such as pipes or bolts. The textures also match, as renewable energy is colored much more pleasantly, featuring different colors of wood and small amounts of detail colors such as blue. Fossil fuels are textured using metallic textures and much darker shades and feature a small amount of detail colors such as purple. The combination of these details creates a big contrast between the fossil and nature assets and shapes the overall feel of the user buildings using the CIC theory [30].

Mascot

Even though the nature of the game world was green and reacting to player actions, it lacked something. It felt empty of life, as there was no movement other than wind and delivery vehicles. UI consisted of only basic icons, building images, and text.

There was no real feeling of connection to the game world because the player had no avatar to play as or characters to interact with. Many studies and books have been written about video game avatars and characters and their connection to human behavior [62, 63]. These connections are based on human nature, as social interaction is a part of it. Mimicry and empathy can at least partly explain why people react to game characters affectively and express human traits on them [64]. That is why the mascot of the factory, *Fantastic Frank*, was introduced to the game. The mascot is used in many parts of the game world and UI to convey information and the feel of companionship to the player.

The overall design of Frank was simplified to make the model easy to animate. The head of the model is made to look like an old computer monitor, which allows for easy facial expression changes with textures if needed. Other limiting factors were also introduced to simplify the model – the whole character armature consists of only six bones. The model is weight-painted imprecisely to allow cartoon-style bending when parts of the model are moved (Fig. 15).

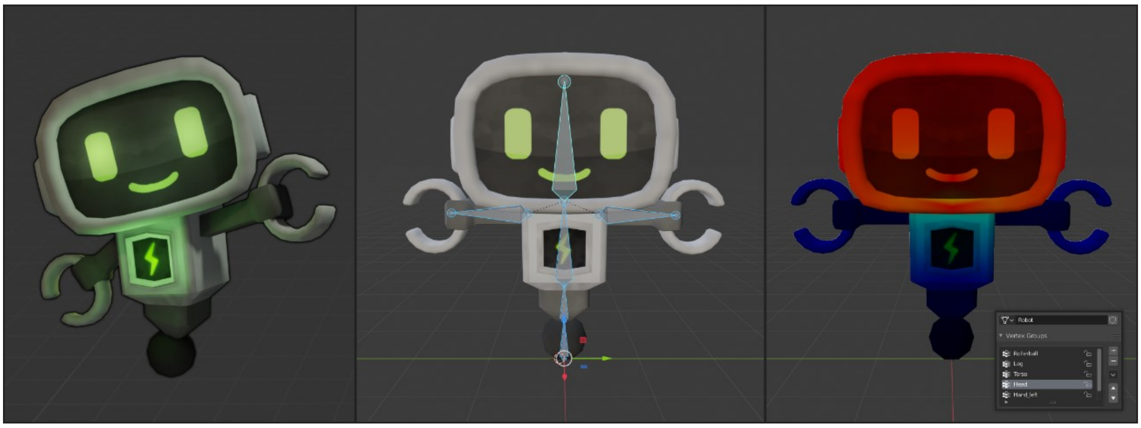

Figure 15. The mascot model. UI Icon (Left), Armature (Middle), and Weight Paint (Right).

Frank is used in almost every primary menu as a decoration. Frank welcomes the player at the start of the game and also conveys the tutorial as a conversation with the player. This interaction helps the player feel more emotionally connected to the game world right from the start [64].

Frank can also be seen working at the factory during gameplay. This behavior is randomized but includes some indirect messages the player may notice, giving the character more personality. Frank's expression changes with high enough pollution, and Frank tries to point out problems with the factory. The addition of the mascot and other interactive elements enhanced the feel of connection and companionship and made the whole game world feel more active than before.

6.3 Optimization in Unity

As the assets are imported into Unity, they can be further optimized in the game engine. Every time a GameObject is drawn on the screen, the engine has to issue a draw call to the graphics API, causing performance overhead on the CPU side [54, 55]. State changes between draw calls, for example, changing materials in the scene, cause validation and translation steps to happen, which are very resource-intensive. Draw calls are determined by many factors, such as the number of materials, lights, and shadows in the scene.

6.3.1 Meshes

The core systems required access to some of the mesh data during runtime to work. This option is called Read or Write. It keeps a second copy of the mesh data in CPU-addressable memory when the mesh is uploaded to GPU-addressable memory, allowing modification during runtime. If the data is not needed during runtime, this option is best to be disabled to save resources [54].

Unity creates animation data automatically when an FBX mesh is imported. This data increases the size of the imported object. If the imported object is not animated, the data is unnecessary and should be disabled [54].

Meshes can be combined to reduce draw calls using the manual CombineMeshes method – if GameObjects share the same material, draw calls can be reduced by merging their meshes together to load all combined assets at once [54]. This solution is fundamental when using static tile-based assets, as they can be combined into a single, seamless asset. Unity also has built-in draw call batching, which has a significant benefit compared to manually combining objects – the combined GameObjects can still be culled individually. As a sidenote, manually combining meshes requires less computational power and memory than built-in batching. The built-in batching is divided into two types:

Dynamic batching automatically batches movable GameObjects into the same draw call. It transforms vertices into world space on the CPU and is only an advantage if that work is smaller than doing a draw call. It has limitations, such as the number of vertices on the batched objects, which is limited to 300 [54, 55].

Static batching only works for objects marked as static, so they cannot move, rotate or scale during runtime. It has no limitations for the number of vertices and is more efficient than dynamic batching, requiring less processing power. As a negative, it requires additional memory for storing the combined geometry. In big scenes, such as a dense forest or a city, the increased memory usage might be a bigger negative than the benefits of using static batching [54, 55].

Fantastic Factory uses manually combined meshes for the distant background props, static batching for the nearby static elements that need to be accessed individually, such as trees, and dynamic batching for objects that can move. In addition to batching, GPU instancing can be used to render multiple copies of the same mesh at once, but it can only combine identical meshes to a single draw call [54, 55]. It is used on repetitive moving objects in the scene that exceed the vertice amount dynamic batching supports.

6.3.2 Materials and Atlases

Materials include information that describes the appearance of a surface of a mesh. Each different instance of a material in the scene issues a new draw call, so combining meshes to use the same material is highly important for optimization [54]. *Fantastic Factory* uses texture atlases (Fig. 12) that allow the use of a single material for many objects in the scene.

Another type of atlas is the Sprite Atlas, which combines all selected images into one big 4x4 GPU formatted image [54, 61]. Separate image textures each make their own draw calls, but combining them into an atlas only results in one. This solution drastically lowers the amount of draw calls the UI demands. In this project, UI's required draw call amount was lowered from 81 draw calls to 19.

6.3.3 Lighting and Shadows

Real-time lighting and shadows are very resource-intensive as they are recalculated in every frame. This recalculation is great for moving objects, but for static objects, this results in unnecessary performance load [65].

For static objects, it is best to use Baked Lights. The pre-calculation of the effects of light in the scene is known as baking. The lighting of the scene is calculated once per iteration of the scene using lightmappers and saved as a separate lightmap texture. If the geometry of the scene changes, the lightmap needs to be baked again to match the changed geometry [54]. It is usually recommended to use only Baked Lights and fake shadows for mobile-optimized games, but the graphical design of *Fantastic Factory* required some real-time shadows, and as such, Mixed Lights were selected to be used.

Mixed lighting balances the benefits and disadvantages of both real-time and Baked lighting to fit many scenarios [65]. All stationary objects have their shadows baked in, but the objects that move use real-time shadows. This optimization keeps the performance cost of shadows relatively low as most of the shadows in the scene are not recalculated every frame. Using a few real-time shadow casters in the project had no noticeable performance impact for newer mobile devices. Still, older hardware may encounter too much load resulting in lower framerates. As such, graphical settings were added to the in-game settings menu.

6.3.4 Resolution and Framerate

An easy way to optimize performance, battery usage, and heat is to decrease the resolution of the game, which makes the image look softer but makes the game require less computational power. A great balance between performance and image quality is to let the UI render at a native resolution and only scale down the resolution of the performance-intensive 3D graphics. This way, even at a lower scale, the important info of the user interface stays sharp, allowing text and icons to be easily readable, keeping a good user experience [66].

Mobile projects must also consider the framerate to achieve a good battery life and reduce thermal throttling. Games should render only as many frames as needed for the best user experience, depending on the type of the game [67]. Adaptive framerate can also be implemented to change the framerate according to what is happening in the game [54, 55].

In *Fantastic Factory*, the settings menu allows users to change between Fancy and Fast graphics, which render the game at 75% scale and 50% scale, respectively. The framerate is set to a maximum of 60 FPS (frames per second).

6.4 Audio

Audio used in the project was mixed with *Magix Vegas Pro 17*, a paid professional creation program that includes advanced video and audio editing features. Audio files were imported to the Unity project using WAV format.

6.4.1 Effects

Mixed audio effects included samples from copyright-free CC0 (Creative Commons) sound effect websites such as *Soundly Free Library* and *Freesound* and effects recorded for the project. The audio samples were imported into *Vegas Pro 17*, which allows independent timeline and track settings such as volume, timing, and many effects such as panning, pitch shift, and distortion (Fig. 16).

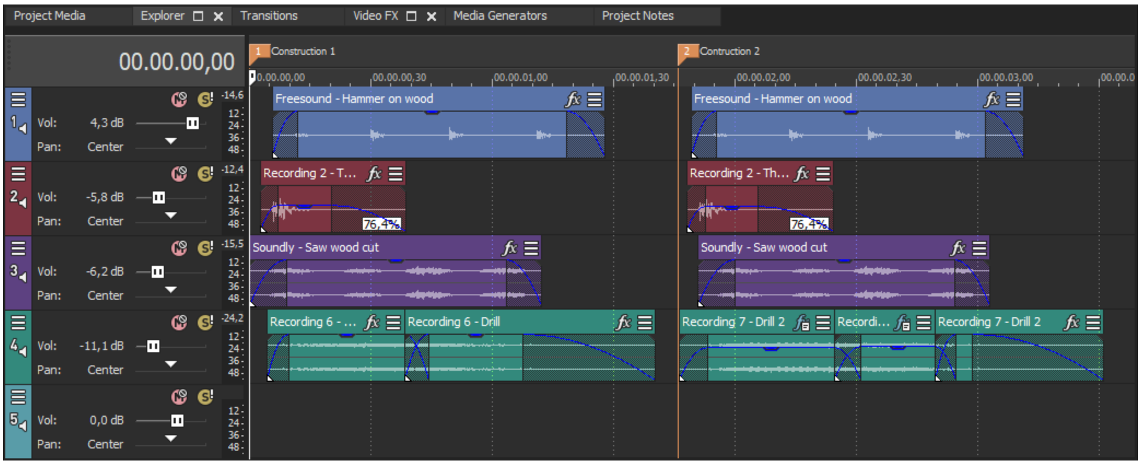

Figure 16. Two sound effects mixed from audio samples in *Vegas Pro 17*.

By the categories defined by J. Fiberg [35], audio was classified into the following categories: Object Sounds and Ornamental Sounds. It was also split into subcategories: SFX (Sound effects), BG (Background), and MX (Music).

Object Sounds include instant SFX that happens instantly during actions, such as construction sounds of buildings or UI sounds. A unique soundscape was achieved by mixing audio from household items and power tools together.

Ornamental Sounds consist of BG and MX, which include mood-changing background audio and music. They do not react to player or scene actions directly but define the feel of the game scene as they include calming music and singing birds when the nature is healthy, and factory machinery sounds when the pollution amount is high.

6.4.2 Music

The musical soundtrack was collected using royalty-free CC3 music, and the author of the music tracks is included in the credits of the game. The soundtrack included smooth jazz music to make the audio atmosphere more relaxing. The soundtrack is changed to a darker jazz theme if pollution levels get too high to enhance the dark mood of the graphics and BG audio.

7 Implementing Data Collection

Unity Analytics is a cloud-based analytics tool developed by Unity [54], and the implementation of basic data collection is made very straightforward. The project was connected to a Unity ID Organization by changing just one setting in the Unity editor. Data collection with Unity Analytics is affected by GDPR (General Data Protection Regulation), and the analytics required a possibility for the users to opt-out of data collection. A lock icon that links the player to the Unity GDPR management website was added to the settings menu of *Fantastic Factory*.

7.1 Legacy and Beta Analytics

During the making of this thesis, Unity services were going through a redesign by Unity, which created some complications as they faced some difficulties, downtime, and a lack of wide documentation. During this redesign, analytics were split into two different subcategories, *Legacy* and *Gaming Services*.

Legacy Analytics features a more robust and reliable collection of tools. It does not require setting up variables in the cloud, as every new custom event sent by the project is automatically added to the cloud database. Legacy comes with a significant disadvantage, as the data takes 8-10 hours to be processed [54].

Gaming Services Analytics is currently in the beta phase and has advantages and disadvantages. Disadvantages include longer time setting up the analytics, as every custom event must be declared in the cloud before usage. The analysis of collected data is much more limited, as Gaming Services has a restricted feature set compared to Legacy Analytics. As an advantage, the data is processed in minutes [68].

Fantastic Factory featured both of the analytics for testing purposes. Gaming Services Analytics Beta was used to make fast, almost real-time tracking of base user activity, and Legacy Analytics was included to handle the custom events and collected data.

7.2 Collecting Events

After connecting the project to a Unity ID Organization, basic data was automatically collected from the project. Both Legacy and Gaming Services Beta collect core metrics such as new installs, daily user activity, time spent in the app, and info of the device used [54, 68]. These core metrics are used to automatically update the dashboard, which can be seen in figure 17.

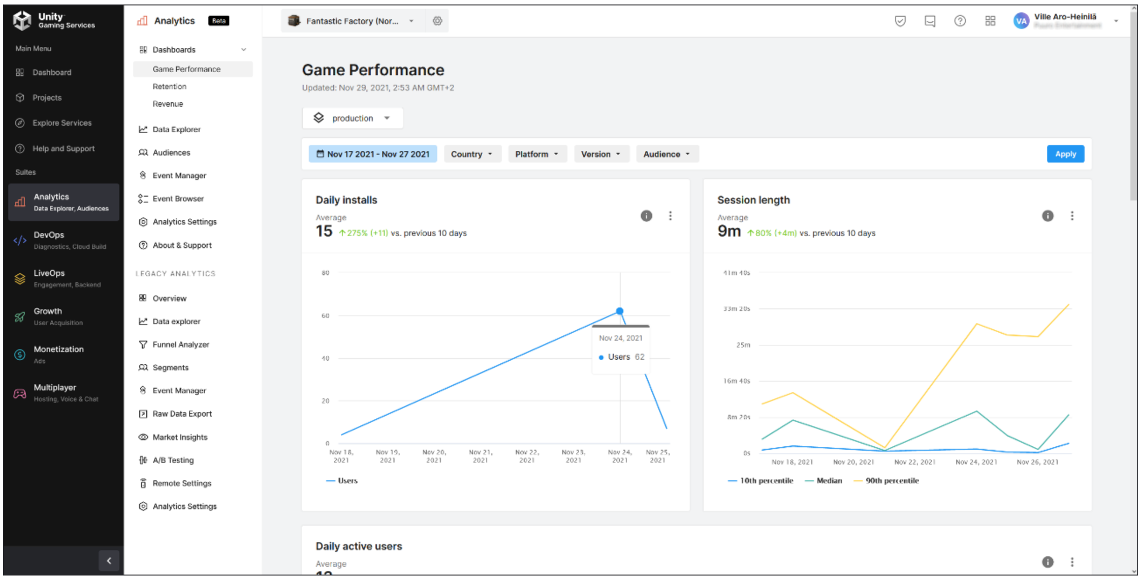

Figure 17. The game performance page in Unity Gaming Services Analytics Beta.

In addition to the automatic core metrics, standard and custom events can be sent manually. The number of collected events is limited to 100 per hour per user, which created limitations for the data collection. Collecting every action as separate events would have filled the limit of 100 events very quickly, and as such, tracked elements were decided to be combined into one event using event parameters. Each event can have up to ten parameters defined by the developer.

The free version of Unity Analytics is limited to the tools featured in the cloud, as the raw collected data cannot be downloaded using the free licenses of Unity. Raw Data Export is only available for the Pro subscription [54, 68].

8 Testing Methods and Results

8.1 Test Versions

Three different test versions of the game using different amounts of auditory and visual effects and components were modified from the original game. These three versions represent low, normal, and high amounts of auditory and visual feedback presented to the player during gameplay (Fig. 18). The core game mechanics and contents are the same between each of the game variants, as only the auditory and visual components were modified. This allows for an even testing ground for the effects of auditory and visual components on human behavior. This type of test setup is called A/B testing, and it was used with three variables as the modification of the game was relatively straightforward and fast [69, 70].

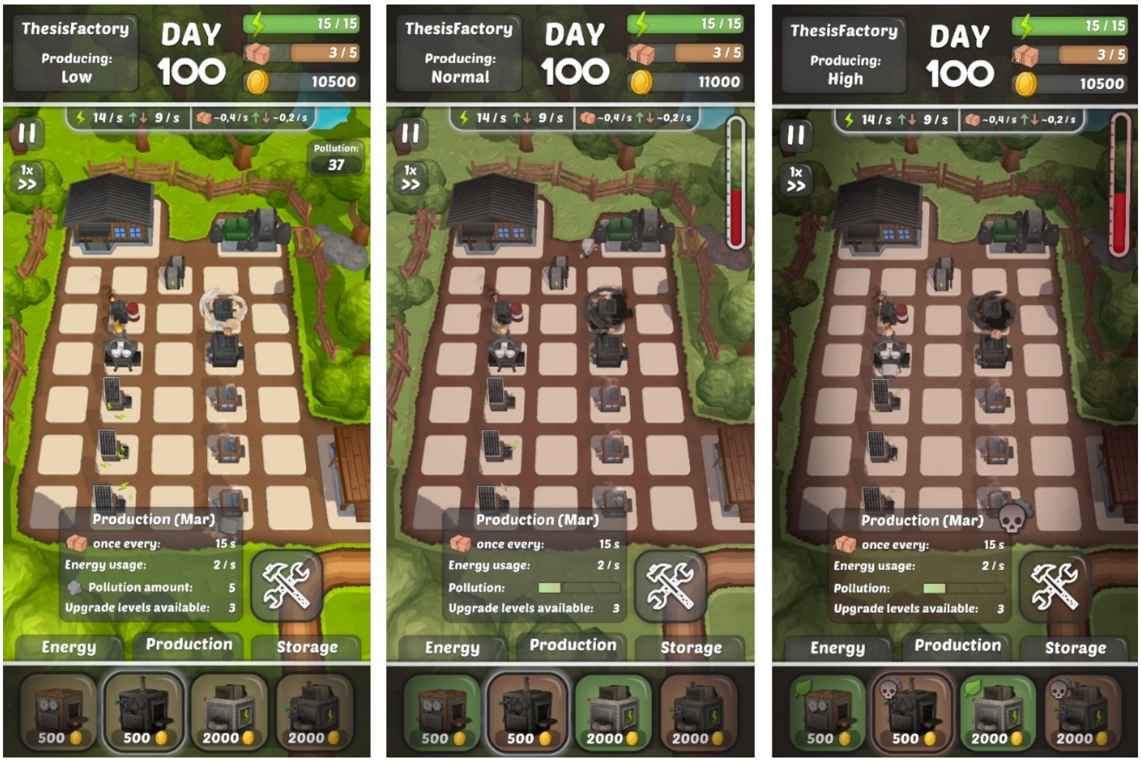

Figure 18. The test versions – low (left), normal (middle), and high (right).

The low version presents all the available buildings and choices with the same neutral UI elements, colors, and effects. Player adaptation effects are disabled to achieve an even ground for all player choices. Pollution amount is presented to the player only using numbers on the UI.

The normal version is the control version and represents how the game was planned to be as entertaining and teaching as possible. It includes different audio, colors, and effects according to the used energy type of the buildings. Player adaptation effects are enabled, and the world colors change according to the pollution amount. Pollution amount is presented on the UI using visual elements such as a thermometer.

The high version intensifies the elements featured in the normal version, giving more contrast to the different energy types. The whole UI changes color according to the selections and includes additional icons representing the pollution amounts.

An assumption of the results was made before the public testing: If the theory of impactful learning games is correct and the development of the game successful, the testers playing low version should use the different energy types more evenly as auditory and visual effects do not influence them, high version testers should prefer renewable energy, and normal version testers are guided for the optimal balance to achieve the best scoring in the game. After all, the goal of *Fantastic Factory* is to teach the players to prefer renewable energy by the current real-world situation of energy types and not totally glorify it.

8.2 Internal Testing

Internal testing was a core part of the development phase to keep the game rapidly evolving. The internal tests included giving a tester one of the game versions on one of the available test devices to collect feedback and ideas during and after gameplay. This setup allowed faster and more comprehensive discovering of bugs and issues in a closed environment. While developing a game, it is easy for some basic elements and bugs to be left unnoticed, and the testers make sure that the most critical parts for actual users are taken care of [71]. The internal testing made the game almost bug-free for public testing, which allowed the collection of more factual data as the players were not distracted by any problems with the mechanics of the game.

8.3 Public Testing

Public field testing was held online, and the game versions were advertised through popular social media platforms, including Youtube, Instagram, and Discord. Willing anonymous participants were guided through a link that redirected the testers to download one of the three different game versions. A tool called *Allocate.Monster*, developed by Anna Fergusson, was used to evenly distribute the test versions to participants [72]. The game versions were downloaded for a total of 371 times in 48 hours.

The participants were not informed that the game had many different versions and only knew they were trying out a new mobile game beta that was supposed to make learning fun. The testers were asked to complete the game at least once, but no additional information was given. The game activity was tracked with Unity Analytics during the first gameplay session, but the players were allowed to quit the data collection at any time from the settings menu of the game if they so wished.

Along with the test versions, a voluntary questionnaire was shared with randomly selected testers according to the game version downloaded. The questionnaire focused on the overall quality and feel of the game to see if it had any potential on the market and if learning games like it are generally accepted.

8.4 Unity Analytics Data

Only game completions triggered the collected data to be saved to get a complete set of data from the first playthrough of the players. Out of the 371 downloads of the game, 220 users opted in for the data collection and completed the game at least once – 75 low version testers, 73 normal version testers, and 72 high version testers were added to the Unity Analytics database.

8.4.1 Limitations

As Raw Data Export was only available for Pro license users of Unity, values had to be manually written from the Unity Analytics dashboard to external tools for further analysis [54, 68]. The dashboard shows mostly predefined calculations of the raw data, and only the final rounded mean values were accessible for this public field testing. As the raw data was not directly accessible, the statistical significance of the collected results from Unity Analytics could not be calculated. Even so, the obtained data is a valuable look into the effects of auditory and visual components.

8.4.2 Analyzing the Data

The average building popularity varied as expected between the game versions. In every version, the testers mostly preferred renewable energy. Prior knowledge and ethical values play a big role in the topic of the game, and the preference for renewable energy was expected even in the low version. Table 1 showcases collected data about player building actions, and lists how many times the testers have on average selected, bought, upgraded, and sold a building. Repeating patterns in player behavior can be seen in the data.

Table 1. Building activity results. The table showcases the average times testers made building actions in the different game versions.

	Low-version, 75 testers				Normal-version, 73 testers				High-version, 72 testers			
Building Type	UI Select	Buy	Upgrade	Sell	UI Select	Buy	Upgrade	Sell	UI Select	Buy	Upgrade	Sell
Solar	4,13	2,31	1,84	0,68	4,38	2,94	2,68	1,06	4,52	3,14	2,58	1,39
Gas	3,22	1,13	0,42	0,91	3,19	1,13	0,42	1,1	2,81	0,46	0,24	0,31
Wind	3,21	1,13	0,88	0,28	3,91	1,65	1,84	0,16	4,42	1,81	2,03	0,31
Oil	3,38	3,28	0,82	0,28	2,81	2,29	0,42	0,22	2,65	1,46	0,08	0,12
Geothermal	3,44	1,41	1,57	0,02	4,41	1,71	2,06	0,06	4,23	1,73	2,03	0,08
Coal	3,34	1,21	1,16	0,12	2,71	0,52	0,39	0,16	2,04	0,46	0,62	0,04
Production 1	5,28	2,53	2,57	0,31	6,61	3,1	3,52	0,64	6,81	3,96	5,38	0,54
Production 2	6,53	2,52	2,07	0,56	6,16	2,45	2,2	1,19	5,46	1,19	1,15	0,31
Energyseller 1	6,03	2,21	2,2	0,2	7,9	3,9	5,24	0,19	7,92	4,13	5,64	0,41
Energyseller 2	5,34	2,24	1,73	0,72	4,73	1,53	1,68	0,77	4,46	1,17	1,66	0,38
Energy Storage	4,04	4,09	3,75	0,22	3,97	3,81	5,06	0,19	4,07	3,92	4,2	0,31
Product Storage	3,34	2,41	3,02	0,2	3,29	2,81	2,49	0,32	3,23	2,42	2,05	0,16

☐ Renewable energy ☐ Fossil Fuel ☐ Passive

When comparing a lower version to a higher (e.g., the low version to the normal version), almost every renewable energy building gets progressively more activity among the testers. Likewise, when comparing fossil fuel buildings, the tester activity decreases. The upgrade and sell values do not follow this trend as clearly and have a lot of variation. It might be that some of the testers did not understand that the buildings could be managed after buying.

Figure 19 shows the overall activity of renewable, fossil, and passive buildings in the test versions. Passive building activity stays relatively consistent between the different versions, but clear trends can be seen in the activity of renewable and fossil buildings.

	Low-version		Normal-version		High-version	
Energy type	UI Select	Buy	UI Select	Buy	UI Select	Buy
Renewable	4,42	2,07	5,42	2,66	5,58	2,95
Fossil	4,36	2,03	3,92	1,72	3,48	0,95
Passive	3,69	3,25	3,63	3,31	3,65	3,17

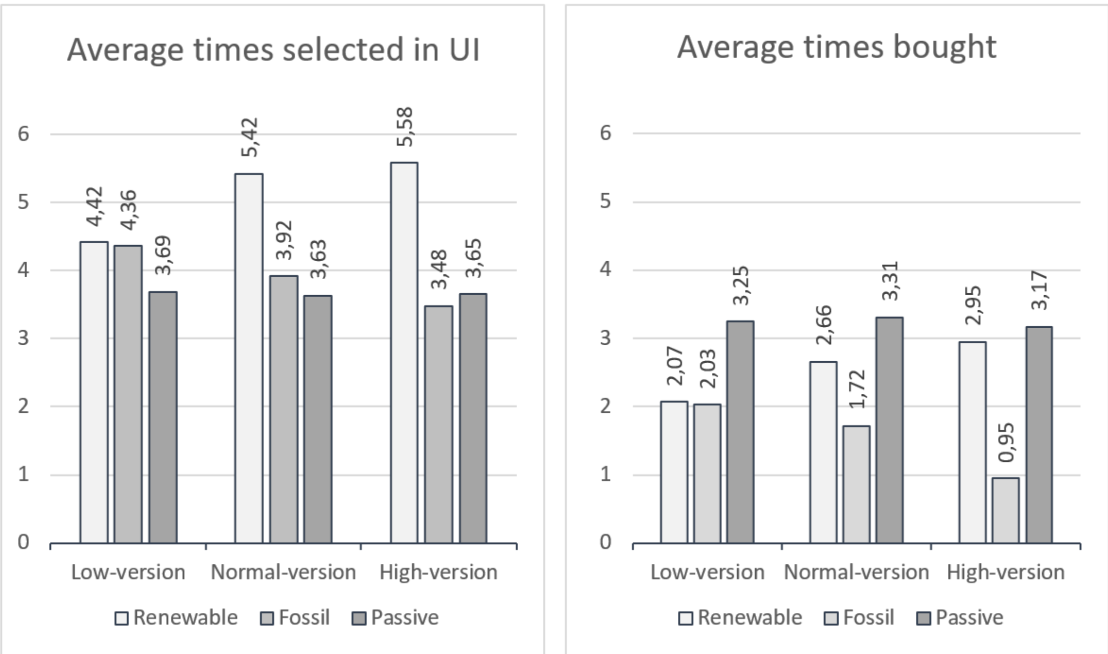

Figure 19. Building activity by energy type. The average of times the testers made building actions in the different game versions by building energy type.

Alongside the collected building activity, the average end scores of testers during their first game playthrough confirmed the assumption made before testing, as normal version testers managed to get the best scores. Low version testers got an average score of 196 630, normal version testers 254 780, and high version testers 231 810.

However, the differences in the first-time scores did not seem to affect the number of study completions. 21,33% (16/75) of low version, 21,91% (16/73) of normal version, and 23,61% (17/72) of high version testers unlocked and read all of the available learning material in the game, completing their studies. It seems that the juiciness of the game affected the motivation of the testers by a little to continue playing and unlock all of the learning material.

The average framerate of the game was also tracked during the first playthrough, and it was consistent between the different versions. Low: 47,75 FPS, normal: 47,45 FPS and high: 47,87 FPS.

From the results achieved by Unity Analytics, it is easy to conclude that the auditory and visual components did affect the player choices, as clear trends can be seen in the data. The test groups were big enough to even out anomalies in the data, and the results logically follow the researched expectations. The following subchapter goes over the feedback collected from the testers.

8.5 Analyzing Feedback

Five willing participants from the test group of each game version were asked to fill out a questionnaire. The age of the participants answering the questionnaire ranged from 11 to 35. Parents of children were informed, and approval to participate in the questionnaire was asked.

The graphics got very positive feedback from the testers. The simplicity, polish, and consistency of the UI were noted, and the cartoon stylized 3D models caught attention. The mascot was brought up many times and described as a well-thought addition to add to the feel and personality of the game. One tester wanted the mascot to be made as a plushie, which was heartwarming.

One low version tester suggested that the graphics should be updated to react to the amount of pollution by showing the changes in the environment. Likewise, one high version tester commented that the effects were a little distracting. These reactions were somewhat expected as those were the purposes of the versions. Another high version tester noted that seeing the world react made fossil fuels more intriguing to be used, which was opposite to the actual intention of the effect. But as the data in table 1 and figure 19 shows, this mindset was not common enough to significantly impact the data. Overall, the collected feedback confirmed that the graphical style of the project was finished and needed no modifications when the project is continued in the future.

On average, the feedback on the soundscape of the game was a mixed positive. The sound effects were rated good and interesting but said to have been a little repetitive. More variation should be added in the future. The music choice of jazz was generally more accepted even though one tester noted that they had heard the music before in other projects as it was royalty-free. An original soundtrack for the game would fix this issue. The normal and high versions were rated higher than the low version as they included more sound effects and music variation.

Gameplay mechanics seemed to raise a bit more variation in the feedback. The mechanics were mostly rated good but often described as *”a bit clunky”,* but *”once understood, works well”* was also a recurring statement in the feedback. The mechanics should be refined for a better first impression. A couple of testers would have wanted to double-tap the building area to build, and a handful wanted to hold and drag the buildings to move them around. These mechanics could be added on top of the current controls to make the game more intuitive to play right from the start. The possibilities of the project were also confirmed by the sheer amount of features the players wished to see in the game in future updates. These ideas included, for example, a game mode with unlimited playtime, a chance to expand the factory area, and building customization.

The learning aspect and style of the project were praised as *”not forced”* and *”actually feeling incorporated with the game world”.* The usage of achievements to evenly space out the learning material was said to be working well, and the inclusion of *”trial and error -learning in the gameplay”* was noted. The amount of learning and gameplay was rated to be very balanced. Overall, the response of the testers to using games as learning tools was very positive. Still, the effectiveness of learning games like *Fantastic Factory* needs further studies as the scope of this thesis did not measure how well the testers actually learned from the game even if they enjoyed studying with it.

The game seemed to be relatively bug-free, according to the feedback. On some rare occasions, the testers faced graphical issues such as UI images not loading. These problems seemed to primarily affect phones using older MediaTek chipsets and need further investigation. The optimizations made during development allowed the game to be played on almost any Android device with good performance, but some of the testers reported high battery usage and their phones getting warm during gameplay and wished the game included a low power mode in the future. This feature can be easily achieved by limiting the framerate.

9 Conclusion

The goal of this thesis was to successfully research, develop, and test an impactful learning game for mobile devices. The whole project was a success based on the collected data and user feedback, and the formed theory of impactful learning games was shown to be functioning as an excellent base to be used for game development.

The results gathered from the testing phase demonstrated that in the right circumstances, a combination of auditory and visual game effects could be used to influence player behavior in predefined ways. The influence can be perceived even with topics the users have prior knowledge and ethical values about if the game environment is compelling and the context is well thought out. The scope of the testing in this study only included the effects on a large crowd of anonymous people of mixed ages and gender, and the effects on individuals of different age groups and gender may play a role in the effectiveness of the auditory and visual components, which cannot be investigated with the data collected and analyzed in this project. As such, this thesis complements prior work on game feedback (both visual/auditory) and can only be used as an example of the combined overall effects of auditory and visual elements on a large audience. Further research on this topic is required to understand other factors of the effects and gain statistical significance.

The testing of the freemium Unity Analytics revealed that the service is easy to set up for any Unity project and can effortlessly collect basic analytics, but is not well fit for more ambitious projects without paying for licenses, as the possibilities of viewing data are limited with the free license.

Almost all aspects of the developed game received positive feedback and demonstrated that there is a demand for it and other games made in this style and many possibilities on the video game market. The balanced combination of learning material and entertainment was well received by the testers, and the message of the game was praised. Games like *Fantastic Factory* could, in theory, be used in combination with regular learning material to work as a motivator and fight procrastination among students.

In the end, it must be noted that this project turned out to be very excessive for a single developer to be made in a couple of months. Further studies should consider this fact and use many developers, more time, or a limited scope for the game to allow for broader testing and analysis of the effects.

The development of the project is continued by the company commissioning this study to reach the public launch of the game. As the implementation of the base systems was already completed during the making of this thesis, modification and expansion of the game content should be relatively straightforward.

References

[1] Dicheva, D. et al. ”Gamification in Education: A Systematic Mapping Study.” *Educational Technology & Society,* Volume 18, Issue 3, International Forum of Educational Technology and Society (2015), Pages 75-88.

[2] Simões, J. et al. ”Using Gamification to Improve Participation in a Social Learning Environment.” *Learning and Diversity in the Cities of the Future*, Edited by Buchem, I., Logos Verlag Berlin GmbH (2015), Pages 169-186.

[3] Miltenoff, P. et al. ”Gaming and Gamification in Academic and Library Settings: Bibliographic Overview.” *Proceedings of SOCIO-INT15 - 2nd International Conference on Education, Social Sciences and Humanities.* Ocerint Publishing (2015).

[4] Figueroa-Flores, J. ”Gamification and Game-Based Learning: Two Strategies for the 21st Century Learner.” *World Journal of Education Research*, Volume 3, Issue 2, Scholink Co, Ltd. (2016), Pages 507-522.

[5] Virvou, M. et al. ”Combining Software Games with Education: Evaluation of its Educational Effectiveness.” *Educational Technology & Society,* Volume 8, Issue 2, International Forum of Educational Technology & Society (2005), Pages 54-65.

[6] Chang, K. et al. ”Embedding game-based problem-solving phase into problem-posing system for mathematics learning.” *Computers & Education*, Volume 58, Issue 2, Elsevier Ltd. (2012), Pages 775-786.

[7] Fuster-Guilló, A. et al. ”Evaluating Impact on Motivation and Academic Performance of a Game-Based Learning Experience Using Kahoot.” *Frontiers in Psychology,* Volume 10, Frontiers Media (2019).

[8] Kühn, S. et al. ”Playing Super Mario induces structural brain plasticity: gray matter changes resulting from training with a commercial video game.” *Mol Psychiatry*, Volume 19, Nature Publishing Group (2014), Pages 265-271.

[9] Puhjo, S. *Using Games to enhance learning* [Internet], Elomatic (2019). Available from: https://www.elomatic.com/en/elomatic/expert-articles/using-games-to-enhance-learning.html. Accessed: 22-09-2021.

[10] Squire, K. & Jenkins, H. ”Harnessing the power of games in education.” *InSight*, Volume 3, IAETE (2003), Pages 5-33.

[11] Kao, D. ”The effects of juiciness in an action RPG.” *Entertainment Computing,* Volume 34, Elsevier (2020).

[12] Juul, J. & Begy, J.S. ”Good feedback for bad players? A preliminary study of ’juicy’ interface feedback.” *Proceedings of first joint FDG/DiGRA Conference*, DiGRA (2016).

[13] Hicks, K. et al. ”Good game feel: an empirically grounded framework for juicy design.” *DiGRA ’18 - Proceedings of the 2018 DiGRA International Conference: The Game Is the Message*, DiGRA (2018).

[14] Wolfson, S. & Case, G. ”The effects of sound and colour on responses to a computer game.” *Interacting with Computers*, Volume 13, Issue 2, Oxford University Press (2000), Pages 183-192.

[15] Hébert, S. et al. ”Physiological stress response to video-game playing: the contribution of built-in music.” *Life Sciences,* Volume 76, Issue 20, Elsevier (2005), Pages 2371-2380.

[16] McGlinchey, C. *Procrastination among post-16 students: how is it experienced and how can we reduce it? The views of students, teachers and educational psychologists.* Doctoral Thesis, University of Exeter (2021).

[17] Schell, J. *The Art of Game Design.* 1st Edition. CRC Press (2008).

[18] Calleja, G. ”Digital games and Escapism.” *Games and Culture,* Volume 5, Issue 4, SAGE Publications (2010), Pages 335-353.

[19] Nieborg, D. ”America’s Army, more than a game.” *Proceedings of the 35th Conference of the International Simulation and Gaming Association (ISAGA),* SAGSAGA (2004).

[20] Badatala, A. et al. ”The effects of playing cooperative video games on teamwork and team performance.” *International Journal of Humanities and Social Science Research*, Volume 2, Issue 12, (2016), Pages 24-28.

[21] Steel Crate Games. *Keep Talking and Nobody Explodes* [Internet]. Available from: https://keeptalkinggame.com. Accessed: 30-09-2021.

[22] BuzzFeed Multiplayer. *Real Bomb Squad Defuses a Bomb In Keep Talking And Nobody Explodes* [Internet video], (2018). Available from: https://youtu.be/BYunaBkn9Ng. Accessed: 30-09-2021.

[23] Persuasive Games. *Killer Flu, a game about seasonal and pandemic flus, how they mutate, and how they are spread* [Internet]. Available from: http://persuasivegames.com/game/killerflu. Accessed: 2021-09-30.

[24] NDemic Creations. *Statement on the current Coronavirus outbreak* [Internet], (2020). Available from: https://www.ndemiccreations.com/en/news/172-statement-on-the-current-coronavirus-outbreak. Accessed: 30-09-2021.

[25] Swink, S. *Game Feel: A Game Designer's Guide to Virtual Sensation*. Morgan Kaufmann Publishers (2009).

[26] Gray, K. et al. *How to prototype a game in under 7 days* [Internet], Gamasutra (2005). Available from: https://www.gamasutra.com/view/feature/2438/how_to_prototype_a_game_in_under_7_.php. Accessed: 23-09-2021.

[27] Schell, J. *The Art of Game Design: A Book of Lenses.* CRC Press (2008).

[28] Juul, J. *A Casual Revolution: Reinventing Video Games and Their Players*. The MIT Press (2010).

[29] Kao, D. & Harrell, D.F. *Toward Understanding the Impact of Visual Themes and Embellishment on Performance, Engagement, and Self-Efficacy in Educational Games.* AERA (2017).

[30] Elliot, A.J. & Maier, M.A. "Chapter two - Color-in-context theory." *Advances in Experimental Social Psychology,* Volume 45, Academic Press (2012), Pages 61-125.

[31] Kao, D. & Harrell, D.F. "Exploring the impact of avatar color on game experience in educational games." *CHI EA '16: Proceedings of the 2016 Conference Extended Abstracts on Human Factors in Computing Systems*, ACM (2016), Pages 1896-1905.

[32] Rutchick, A.M. et al. "The pen is mightier than the word: Object priming of evaluative standards." *European Journal of Social Psychology*, Volume 40, Issue 5, Wiley & Sons (2010), Pages 704-708.

[33] Pravossoudovitch, K. ”Is red the colour of danger? Testing an implicit red-danger association.” *Ergonomics*, Volume 57, Issue 4, Taylor & Francis (2013), Pages 511-524.

[34] Sekuler, R. et al. ”Sound alters visual motion perception.” *Nature,* Volume 385**,** (1997), Page 308.

[35] Friberg, J. & Gärdenfors, D. ”Audio games: New perspectives on game audio.” *ACE '04: Proceedings of the 2004 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, ACM (2004), Pages 148-154.

[36] Kao, D. & Harrell, D.F. ”Exploring the effects of encouragement in educational games.” *CHI EA '16: Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, ACM (2016), Pages 1906-1914.

[37] Kao, D. & Harrell, D.F. ”Embellishment & Effects: Seduction by Style.” *Avatar, Assembled: The Social and Technical Anatomy of Digital Bodies*, Edited by Blanks, J., Peter Lang (2017).

[38] Park, B. et al. ”Does cognitive load moderate the seductive details effect? A multimedia study.” *Computers in Human Behavior*, Volume 27, Issue 1, Elsevier (2011), Pages 5-10.

[39] Kurzgesagt – In a Nutshell [Internet]. Available from: https://kurzgesagt.org/. Accessed: 15-10-2021.

[40] The Infographics Show [Internet]. Available from: https://www.youtube.com/c/TheInfographicsShowOFFICIAL. Accessed: 15-10-2021.

[41] Salen, K. &. Zimmerman, E. *Rules of Play: Game Design Fundamentals.* The MIT Press (2003).

[42] Salen, K. &. Zimmerman, E. ”Game design and meaningful play.” *Handbook of Computer Game Studies.,* Edited by Raessens, J. & Goldstein, J., The MIT Press (2005), Pages 59-79.

[43] Rubin, E. ”How games end in master chess tournaments.” *The American Statistician*, Volume 27, Issue 3, Taylor & Francis (1972), Pages 119-122.

[44] Fort, T. "Controlling Randomness: Using Procedural Generation to Influence Player Uncertainty in Video Games." *HIM 1990-2015.* 1770. UFC STARS (2015).

[45] Holding, D.H. *The Psychology of Chess Skill.* L. Erlbaum Associates (1985).

[46] Jeffrey, C. *And Action! An Examination of Physics in Video Games* [Internet], Techspot (2019). Available from: https://www.techspot.com/article/1960-video-game-physics. Accessed: 27-11-2021.

[47] Alves, V. & Roque, L. "Guidelines for Sound Design in Computer Games." *Game Sound Technology and Player Interaction: Concepts and Developments*, IGI Global (2010), Pages 362-383.

[48] Salen, K. &. Zimmerman, E. *The Game Design Reader: A Rules of Play Anthology.* The MIT Press (2005).

[49] Toprac, P. "The Psychology of Control and Video Games." *Ctrl-Alt-Play: Essays on Control in Video Gaming,* edited by Wysocki, M., McFarland (2013), Pages 21-33.

[50] Watkins, S.C. "Being Indie: The DIY Ethos and Indie Game Development." *Young People's Transitions into Creative Work.* 1st Edition. Routledge / Taylor & Francis (2019), Chapter 2.

[51] Moogk, D.R. "Minimum Viable Product and the Importance of Experimentation in Technology Startups." *Technology Innovation Management Review*, March 2012, Talent First Network (2012), Pages 23-26.

[52] Blank, S. "Why the lean start-up changes everything." *Harvard Business Review*, May 2013, Harvard Business Publishing (2013).

[53] Unity. *Mobile Games Introduction* [Internet]. Available from: https://unity.com/solutions/mobile. Accessed: 27-10-2021.

[54] Unity Documentation [Internet]. Available from: https://docs.unity3d.com/Manual. Accessed: 29-10-2021.

[55] Krogh-Jacobsen, T. *Optimize Your mobile game performance: Expert tips on graphics and assets* [Internet], Unity (2021). Available from:

https://blog.unity.com/technology/optimize-your-mobile-game-performance-expert-tips-on-graphics-and-assets. Accessed: 15-11-2021.

[56] Innovecs Games. *Full Guide to Mobile Game Design: Theory and Best Practices* [Internet], (2020). Available from: https://www.innovecsgames.com/blog/mobile-game-design/. Accessed: 24-11-2021.

[57] Dori, A. *Game Design UX Best Practices* [Internet], UX Planet (2019). Available from: https://uxplanet.org/game-design-ux-best-practices-guide-4a3078c32099. Accessed: 24-11-2021.

[58] Interaction Design Foundation. *KISS (Keep it Simple, Stupid) – A design Principle* [Internet], (2020). Available from: https://www.interaction-design.org/literature/article/kiss-keep-it-simple-stupid-a-design-principle. Accessed: 15-11-2021.

[59] Puterka, M. *Another Dimension. New Mastery – An Isometric Illustration Tutorial* [Internet], Toptal (2020). Available from: https://www.toptal.com/designers/graphic/isometric-illustration-tutorial. Accessed: 16-11-2021.

[60] Blender. *The Freedom to Create* [Internet]. Available from: https://www.blender.org/about/. Accessed: 18-11-2021.

[61] Arntzen, H.K. *Compressed GPU texture formats – a review and compute shader decoders – part 1* [Internet], Themaister (2020). Available from: https://themaister.net/blog/2020/08/12/compressed-gpu-texture-formats-a-review-and-compute-shader-decoders-part-1/. Accessed: 19-11-2021.

[62] Worth, N. *Players and Avatars: The Connections between Player Personality, Avatar Personality, and Behavior in Video Games*. Brock University (2015).

[63] Bayliss, P. ”Beings in the Game-world: Characters, Avatars, and Players.” *IE ’07: Proceedings of the 4th Australasian Conference on Interactive Entertainment*, ACM (2007), Article 4, Pages 1-6.

[64] Lankoski, P. *Character-Driven Game Design: A Design Approach and Its Foundations In Character Engagement.* Doctoral Thesis, Aalto University, School of Art and Design (2010).

[65] Android Developers. *Lighting for mobile games with Unity* [Internet]. Available from: https://developer.android.com/games/optimize/lighting-for-mobile-games-with-unity. Accessed: 26-11-2021.

[66] Wilde, T. *PC graphics options explained* [Internet], PC Gamer (2019). Available from: https://www.pcgamer.com/pc-graphics-options-explained/. Accessed: 27-11-2021.

[67] GameBench. *Measuring Frame Rates In Android And IOS Games* [Internet], (2019). Available from: https://blog.gamebench.net/measuring-frame-rates-in-android-and-ios-games. Accessed: 27-11-2021.

[68] Unity Gaming Services Analytics [Internet]. Available from: https://docs.unity.com/analytics/UnityAnalytics.htm. Accessed 28-11-2021.

[69] Watts, J. *How A/B testing can help you optimize your mobile game* [Internet], Unity (2020). Available from: https://blog.unity.com/technology/how-a-b-testing-can-help-you-optimize-your-mobile-game. Accessed: 29-11-2021.

[70] DeltaDNA. *5 top tips for A/B Testing in free-to-play games* [Internet], (2015). Available from: https://deltadna.com/blog/top-tips-ab-testing/. Accessed: 29-11-2021.

[71] Thompson, H. *The Importance and Impact of Game testing* [Internet], The Nerd Stash (2021). Available from: https://thenerdstash.com/the-importance-and-impact-of-game-testing/. Accessed: 29-11-2021.

[72] Fergusson, A. *Designing online experiments using Google forms+ random redirect tool* [Internet], Teaching Statistics Is Awesome (2016). Available from: https://teaching.statistics-is-awesome.org/designing-online-experiments-using-google-forms-random-redirect-tool/. Accessed: 24-11-2021.