How Using Game Design Patterns in Game Design Affect Game Development

Computer Science
Department of Computing
Faculty of Technology
Master's thesis

Author:
Jaakko Uitto

April 2026
Turku

Master's thesis

Subject: Computer Science
Author(s): Jaakko Uitto
Title: How Using Game Design Patterns in Game Design Affect Game Development
Supervisor(s): Dr Jouni Smed
Number of pages: 57 pages
Date: 13.4.2026

This thesis is about game design. The thesis goes over a study that was created that asks the question: “How do game design patterns affect game development?” To answer this question, a game was designed and developed from the concept phase until the prototype phase. The game did not go to the production phase, as this was deemed unnecessary for the study. The game development process is explained as a whole from the concept phase to the post-production, focusing on the design phase of development. In designing the game, a set of game design patterns was chosen as research patterns that were used in creating the game design. A game design document was created for the design of the game. In the course of the development process, some of the chosen patterns were excluded from the final design of the game, but most of them were included. The results showed that the game design patterns did affect game design and game development, and there were both pros and cons to this. Many of the chosen research patterns directly shaped the game design, especially the game mechanics. For this reason, game design patterns should be researched further for potential uses in game development.

Key words: game design pattern, game design, game development, game design document, GDD

Table of contents

1 Introduction

Game design is a major part of game development. In the past, many have attempted to develop general game design methods. Despite this, game design methodology remains fragmented. Neves and Zagalo (2021) have done research on the past of game design as a research subject, finding that the most interest for these methods came between 1999 and 2006. According to them, research has been done following this period, but it has generally failed to attract the interest of the game industry. Neil (2016) notes that game design remains undeveloped in comparison to other aspects of game development, despite the advances in technology and methods in other parts of game development. Neves and Zagalo (2021) suggest core methods and literacy for game design that they have gathered from past research. One of these tools is *game design patterns.*

Game design patterns are game design tools that describe commonly reoccurring parts of designs of games (Björk & Holopainen, 2005). The idea for doing something related to game design for this thesis came from my own interests in video games and from a course I did on the subject. The decision of using the game design patterns for this thesis was made as it seemed like an interesting idea, and as far as I knew, there were no studies that had been made that were exactly like the one that I decided to do. Game design is explained further in Section 2.2 in this thesis, with the Section 2.2.9 going more in detail on game design patterns. The research question for this thesis is:

RQ1: How does using game design patterns affect game development?

To answer the research question, a study was made. The study is experimental research in how game design patterns affect the game development process. This was done by choosing a game idea and picking a set of patterns from a list to use in designing and creating a game based on that idea. Another part of this was seeing which game design patterns ended up in the game regardless of intention to do so. For this study, a game prototype was designed and created to research this and to see what aspects affected the result. Strategy is one of my favourite video game genres, which is the reason why I decided to create a strategy game for this study. The game was made as part of a separate project with the intention of using the results of that project for this thesis. There is more on the project itself in Chapter 3.

There were many ways that sources were found for this study. Google Scholar and Volter were used to find relevant books and articles. For the game design patterns, the words “Game Design Patterns” were used, for game design itself, the words “Game Design” were used, and for the game development process itself, the words “Game Development” were used. The website called Game Developer was also used to search for these to get the game developer side of things as opposed to researchers’ side. “Strategy Games” was used as a term when searching for information about strategy games. Google was also used to find articles in game development and for explanations for specific terms. Some sources were taken from the references in other sources.

There were some challenges with this. One was the fact that searching for “game design patterns” resulted in some results being about software patterns instead. Another was the fact that there has not really been a study quite like this one, and if there has, it did not come up during the search. When researching game design patterns, it is usually about mapping different patterns in games and labelling them (Björk & Holopainen, 2005), pedagogy-related topics (Barney, 2021a) or workshops (Sintoris, 2015), as opposed to utilizing them for creating a new game. This meant that this was somewhat of a new territory to go into.

This thesis has the following structure. Chapter 2 goes over the game development process itself, which explains the overall process that goes into game development from the concept phase all the way to post-production. It focuses on the early stages of game development as that is the most relevant for this study. Chapter 3 goes into the development process of the game that was created for this study, from the concept to the prototype stage. Later stages were not relevant for the study, so they were not included. Chapter 4 are the results, which go over the results that were found over the course of the development of the game. Chapter 5 is the analysis that answers the research question and takes other factors into consideration. After that is chapter 6, which goes over potential future work in this topic. Finally, Chapter 7 is the conclusion. There are two appendices: Appendix 1 is a Game Design Patterns Table that lists the game design patterns that were used in the study, and Appendix 2 is the Game Design Document that was created for the game that was made for this study.

2 Game Development Process

Games are usually developed in teams. There are different aspects of game development that different roles focus on, including production, design, art, programming and audio (Novak, 2012). In this chapter, we go over the process of developing a game in theory from the game concept all the way to the post-production. The *design* aspect of game development is the most important part of it for the purposes of this study, so there is more of a focus on the earlier phases of development. This chapter goes over many parts of game design, but some things not relevant for the study are not included. The most important game design tool for the purposes of this study, *game design patterns*, is also explained here.

2.1 Game Concept

The first development phase is the *concept* phase. This phase starts with the game idea and ends when the planning of the project starts. The goals of this phase are coming up with what the game is about, identifying the target market, assessing development resources and finding something that resonates with the developers (Novak, 2012). This is also called the *planning* phase (Bramble, 2023).

2.1.1 Game Ideas

Before getting started with the design process, it would be good to have some kind of goal in mind. What is the purpose for making a game? This is subjective, but there are multiple motivational factors as to why. These purposes shape the type of game that is being made. For example, there are games made for entertainment, social interaction, education, recruitment and training (Novak, 2012).

The game design process starts with an idea. Coming up with the idea in the first place can be tedious, as it can take a long time. There are many ways to help coming up with ideas. Getting an inspiration is one of these ways, and the inspiration can come from many places like already existing games and experiences that come from life. Ideas should also be recorder some way, like written down, so that they will not be forgotten. Other things that can help with this are a good night's sleep, paying attention to one's thoughts and feelings, sketching and making lists (Schell, 2019). There are other examples Schell also gives, but it becomes abstract quickly, and

the process of coming up with ideas is subjective. Other sources of inspiration include dreams, other media like books, movies and television, as well as other already existing games (Adams, 2014).

Another thing that can help with coming up with ideas is thinking of game design as a problem. We can pick a problem statement to solve to help us come up with an idea. For example, the goal could be improving on the game *Tetris,* which then becomes the problem statement of: How can we improve *Tetris*? (Schell, 2019).

However, it is not that simple to finally focus on one single idea. Usually, there are many ideas. This is where we can fall into a trap of not knowing what idea to proceed with, but it is important to pick something and go for it. After the idea is picked and refined, it can become clear that it is not the right choice, at which point it is important to be able to let go of the idea and pick something else (Schell, 2019).

Schell (2019) has eight filters that can help with testing if the design of a game is good enough for the designer's purposes. These happen in different parts of the development process. The first five of these and number seven are explained in the concept phase. The other two are explained later. The eight filters are:

1. Artistic impulse
2. Demographics
3. Experience design
4. Innovation
5. Business and marketing
6. Engineering
7. Social/Community
8. Playtesting

The first of Schell's (2019) filters comes into play early on into the concept phase. The *artistic impulse* is the most personal filter, which tests if the game designer feels the game is right for themselves. If the game feels right for the designer, it passes the test, but if it does not, it needs to be changed. The question is: Do we want to make this?

2.1.2 Target Audience

Schell's (2019) second filter is the *demographics,* which asks if the intended audience will like the game enough. Who is going to play this? It is important to understand what the game's target audience or market is. Not everyone plays the same games or has the same interests, so developing a game meant for everyone is not a good strategy for game development. Understanding the market is understanding the people who play games. This is needed to create a game that suits the market's needs. There are also many different demographics of players, which include different genders, nationalities and generations. An important question when starting to design a game is: Who do we want to reach? These people are the market (Novak, 2012). To help with this understanding, we can look at *player motivations*.

Schell's (2019) third filter is the *experience design.* It focuses on creating a good experience for the player. The experience includes things like aesthetics, theme and balancing. Why should someone play this? This is part of the player motivation. Why do people play games? This is subjective, but some examples for motivations include social interaction, seclusion, competition, mastery and escapism. The player motivation statement explains the game's victory conditions and why the players should play the game (Novak, 2012).

Choosing a game *genre* is important when figuring out what kind of game we want to make. It can also help with choosing what the target audience is. The game genres are not like genres in books and movies, which are defined by the story, plot or setting. Game genres focus on the style of the game, which is defined by how the game is played. Game genres are categories based on a combination of many things, like player perspective and game-playing strategies (Novak, 2012). There could also be many genres mixed in, which could help with the game's uniqueness. Specifically, game genres typically come from the kinds of challenges in the game, regardless of the setting (Adams, 2014).

Schell's (2019) seventh filter is *social* or *community.* It focuses on the social aspects of the game. A part of this design choice is the community around the game. If the goal is to build a strong community around the game, some social elements could be added to the game to help form the game's community.

2.1.3 Marketing

Innovation is Schell's (2019) fourth filter, which states that since we are designing a new game, we should have something new in it that the players have not seen before. This can help with selling the game to players. This is where the *unique selling proposition* (USP) should be made, which is a short paragraph that explains what makes the game unique. This could be things like potentially exceptional features, unique art style or advanced technology (Novak, 2012).

The fifth Schell's (2019) filter is *business and marketing*, which is about making the game profitable. Why should someone pay for this and how? The target audience and USP mentioned earlier are also a part of this. Another part of this is *competitive analysis,* which is a process of choosing some already successful existing games that are available on the market and explaining how the game is going to compete with them and distinguish itself from them (Novak, 2012).

Another part of the marketing aspect is the *premise* or *high concept.* It is a summary of the what the game is about, like its purpose and theme. The intention of this is to catch the attention of the customers to get them to buy the game. It is usually a short paragraph, and it is a good idea to write it in second person to address the player directly. The summary should explain to the player what makes the game unique and indicate to the player what the genre of the game is (Novak, 2012). For example: "You are a superhero. Fly around the city and fight villains." This indicates the game has a superhero theme and the genre is fighting.

Beyond these, there are other things to include in the game concept that are important. Expected *target rating* tells the customers what level of maturity the game is intended for. The *target platform* is simply the machine or machines the game is intended to run on like computer (PC), mobile or a specific console. If the game uses some adapted *licence*, like an already existing movie for the setting or sport team logos and names, that should be indicated in this phase along with the specifics of what the licenced work is (Novak, 2012).

2.1.4 Concept Document

Everything mentioned previously in this chapter should be a part of the *concept document.* The *concept document* is also known as the pitch document. It is a short document with the goal to

convey what the game and its purpose is. It is used to convince the management to assess if the game's idea is something they should invest time and money in. The purpose of the document is to pitch and sell the game idea to investors, publishers or other decision makers involved in the process (Novak, 2012). For independent developers, alternative ways for financing a game include crowdfunding campaigns and early access releases (Bramble, 2023).

Even if there are no external decision makers involved in the game development, it is still useful for the development team itself in preparation for the next phase of development, which is where the *game design* is refined.

2.2 Game Design

Game design is the process of designing a game. This includes imagining what the game is going to be, defining the way the game works and describing all the different elements that the game is made of. These elements include concepts, mechanics and artistic elements (Adams, 2014). It starts in the concept phase with coming up with ideas, but it is refined in the *pre-production* phase.

The *pre-production* phase comes after the concept phase. This is where the project proposal is created, as well as additional documentation, like an art style guide, production plan, the *game design document* (GDD) and *technical design document* (Novak, 2012).

2.2.1 Gameplay

An important aspect of game design is *gameplay.* Gameplay is what the player faces in the game, including choices, challenges and consequences. Imagining how different challenges link together can help with this, like with designing plot points of a story. For each challenge, different strategies should be made for the player to use to overcome it. The rules of the game are also a part of the gameplay. They define how the game is played and what actions the players can and cannot take. These rules are conveyed to the players in some way, often documented in an instruction manual, or with the help of hints or tutorial modes that help players learn the different aspects of gameplay (Novak, 2012). Many of the simpler games have only one gameplay mode, but more complex games have multiple. When the gameplay available, the

user interface (UI) or both change drastically, the gameplay mode has changed. Gameplay mode is defined as a subset of a game's total gameplay that is available at any one time in the game. A gameplay mode consists of a camera model, an interaction model, and the gameplay available (Adams, 2014).

There are different types of challenges a game can have. The challenges of the game are often linked to the game's genre and should be designed accordingly to fit the player expectations that come with the genre. There are *explicit challenges* that are intentional and immediate. An example of an explicit challenge could be jumping over rolling barrels or onto a moving platform. Another example could be a locked door that needs to be unlocked somehow. *Implicit challenges* are not specifically added challenges but are emergent features of the game itself. An example of an implicit challenge could be deciding what resources to use and when (Novak, 2012).

The *victory* and *loss conditions* are a part of gameplay. The *victory conditions* explain how the player can win the game, and the loss conditions explain how the player can lose. There are also games that do not have these conditions, like with some puzzle games. There are also games that do not have a victory condition, but have a loss condition, like *Tetris,* where the game continues until the player loses. In other games like *The Sims,* the victory condition is ongoing and maintained by the player characters not dying. There can also be many different victory conditions, like with strategy games like in the *Civilization* series (Novak, 2012).

Another aspect of gameplay is the information that is available to the player about the state of play. This could be *perfect information*, which means that everything about the game is provided to the players. This provides the players with logical challenges that can be resolved with assessing the information available to them. This is true with games like *chess* that show the entire state of the game to the players. *Imperfect information* gives the players only a portion of the information of the state of play. The challenges in these types of games require an ability to make guesses about the missing information. This happens in various card games where the players only see their own cards and can only make guesses or estimates as to what the opponent's hand is. With strategy games, this can be done with fog of war, which only provides information of the known portion of the game world to the player, and the enemy units can be hidden in the fog of war. Imperfect information is often used to create an element of mystery to the game and challenges players to interact with the game world to learn more about it. A

part of the game information is knowledge. There is *intrinsic knowledge* that can be learned from the game itself, and *extrinsic knowledge* that can be learned outside of it, like from a strategy guide (Novak, 2012).

The *point-of-view* (POV) determines the player's perspective that the game is viewed through. The *first-person* POV shows the game through the eyes of the character as if the player were the character. The *third-person* POV shows the player character on screen. This is also an important design choice as it changes the feel of the game. It can also affect the relationship and attachment the player develops with the player character. There are also other perspectives in games that are not necessarily tied to the character. *Omnipresent* perspective allows the player to view and affect different parts of the game world. Another is the *aerial* or *top-down* perspective, which allows for the player to use the bird's eye view, looking at the world from above. The *isometric* perspective shows the landscape at an angle of around 45 degrees. The *side-scrolling* perspective views the world as a flat side-view, like in platformers like the *Mario* games (Novak, 2012).

2.2.2 Balancing

Another important part of game design is *balancing.* If the game is balanced, it is perceived by players to be consistent and fair. It also has a consistency in the winning becoming easier as the player's skills increase. There could still be random events that potentially decrease the likelihood of winning even for a skilled player, but a better player should still succeed more than a less-skilled player (Novak, 2012).

There are many things that can help with making a game balanced. One of these is *consistent challenges* that get gradually more difficult. There should be a fair playing experiences that do not punish the player too harshly from the beginning. There should also be a lack of stagnation in that the players should never get stuck in a part of the game without any ways to proceed. There should not be any required trivial decisions that make no difference in the game, and the players should only be required to make important decisions. A fair difficulty system would also help with this. There should be a choice of a difficulty level, or the level should adjust to the player's abilities throughout the game (Novak, 2012). For example, the game could simply have easy challenges at the start, with the challenges getting harder as the game goes on.

Schell's (2019) third filter, experience design, is a part of designing the gameplay and balancing, as these are important aspects of the experience for the player.

Other things that that can help with balancing include *static balance.* Static balance exists in the rules of the game and can include things like the relative strength of units in a strategy game. This information can then be used by the player to make decisions on which units to use. Other things to consider with balancing are *obvious strategies.* The obvious strategies are meant to be things that the player can perceive as being superior in many situations of the game. An example of this is a character with an obvious strength in a skill that can be used in the game, which makes using other skills less appealing to the player. Another way to balance a game is adding *symmetry.* Symmetry means providing each player and NPC an equal starting condition and abilities. This makes the game outcome depend on the skill levels of the players (Novak, 2012).

2.2.3 Story

Games often have stories. They are a significant portion of highly visual interactive entertainment. The storytelling traditions of visual storytelling are a part of this, as are other storytelling traditions like oral, audio and text in the form of music, sound effects, narration, dialogue and on-screen text. The same storytelling aspects that appear in literature and film also appear in games. These include character archetypes, story structures and other story elements (Novak, 2012).

The *backstory* is an important part of storytelling in games. It provides information about the game world leading up to the beginning of the game. This helps with getting the player to understand the purpose of the game and the game world itself and potentially establish initial bonds with the game characters. Another important story element for games is the *synopsis.* The synopsis is the storyline of the game, which helps with immersing the player in the game world through being involved in the setting and action that take place in the game. Another part of storytelling that helps with game design is the *theme.* The theme of the game represents what the game is about. This can be some kind of philosophical question that is explored or a moral dilemma, and the theme is often related to the primary obstacle the main character or characters face in the story. Other aspects of storytelling include the *setting*, which is the world the game takes place in and is explored by the player, and the *plot,* which is the way the story unfolds

throughout the game. The *realism* of the game, or how realistic the things that happen in it are, is also an important aspect to consider during game design (Novak, 2012).

Another aspect of storytelling that is important as far as game design is concerned is *player control.* If the player character is customizable, this allows for higher level of player control, which can help immersion. This often means leaving the protagonist undefined in the story to allow the player to insert themselves into the story. Other ways to allow player control are choices that affect the story. This is often done with different paths the player can take. There is also the possibility of including world building tools that can customize the world itself (Novak, 2012). An example of world building tools could be a crafting mechanic that allows the player to build custom structures with blocks, like in *Minecraft.*

2.2.4 Characters

The game *characters* themselves are also important for the game design, both in story and mechanics. If there is only one player character, it is called the *avatar.* However, the player can also control multiple characters, like in strategy games, where the player is more concerned with managing resources than controlling a single character. In this case, the characters are usually more distant and less personal. Other game characters are the *non-player characters,* which are the characters that are not controlled by players (Novak, 2012).

Characters in games usually develop with increases in statistics. For example, these statistics can be strength, experience and skill. Many games allow for different types of characters to be selected or created, with things like class and race. These can often have unique skills attached to them or have an effect in the way the game is played. An example of this could be having a character class that is skilled with a sword and another that is skilled with a bow (Novak, 2012).

2.2.5 Level Design

Level design is the creation of environments, scenarios or missions in games. There are usually some level design tools that the level designer would use to design these levels, which can be a specific tool for creating levels, a graphics editing software or a game engine. When designing a level, the functions the levels have in the game should be considered. For example, it could

introduce a new skill or a character (Novak, 2012). However, there could also be simply one level that serves as the entire game world.

There are different types of ways levels can structure a game. One of these is the *objectives* of the level. This gives the player a direction to go in at the start of the level. The objectives can be conveyed with a cut-scene, a tutorial or a status screen where the objectives can be viewed. Another thing to consider is the *flow.* When designing a level, the designer might want to have a way the player accomplishes certain objectives on a certain area of the level before they advance. If the player can move freely, they might be able to run past enemies without engaging in a fight with the enemies, which might not be what the designer intended. In this case, physical barriers that prevent the player from leaving the area before the objectives are completed could be set up (Novak, 2012).

Another thing to consider when making levels is the *duration* of the level and how long the player is going to take to complete the level. Novak (2012) gives some ideas as to how long a level should be, ranging from fifteen minutes to two hours, with one idea being that a level should be able to be completed in one play session. However, it is entirely up to the designer to decide how long the levels should be. Another thing tied to this is time. There are different types of game time that can affect the game. There is the *authentic* time, which use a real clock system that follows the real time. There is the *limited* time, which simply limits the time available for a level to be completed in. *Variable* time allows for time to be varied, speeding up or slowing time when needed. *Player-adjusted* time allows for the player to adjust the time the level takes, such as with sports games where the player can choose the duration of the match. *Altered* time allows for time to be altered at specific points, like with slow motion when dodging bullets (Novak, 2012).

The *availability* of levels should also be considered. This is basically for deciding how many levels are available to the player at once. Novak (2012) gives some ideas for this, with examples of different genres of games allowing a different number of levels to the players. For example, role-playing game players might want multiple levels or quests available, and first-person games could have only one level available. However, this is entirely up to the designer.

The relationships between different levels are also important to consider, such as increasing the difficulty of the game for later levels. This is part of the game *progression.* The game could be

linear in difficulty, constantly increasing towards the later stages of the game, but it does not need to be. It could also be flat, where the difficulty does not vary. It could also use the *s-curve*, which starts with the flat curve with the tutorial levels, before starting to increase in difficulty in the middle of the game, before flattening out towards the end (Novak, 2012). However, this is also entirely up to the designer, and these are not the only ways to shape the game difficulty.

Another thing to consider is the scale of the game world. This includes the total size of the physical space the player can navigate and the relative sizes of objects. In a realistic game, we might want to keep the game as close to a real world as possible with the objective sizes, but games often exaggerate the size of important collectable objects so that they are easier to find. In strategy games like the *Civilization,* the character units can be shown as larger than structures to help the player see them in the world. These scales can also be distorted to allow better navigation or to show objects in the world that would realistically be hidden behind other objects, like units being hidden behind a building. With the physical space, scale distortion can be used to make the game world appear large but still take a small amount of real-world time to navigate, like only taking a few minutes to go from one end to the other. An example of this is *Grand Theft Auto,* in which the exterior world might be small enough to not take a long time to navigate, but the interiors are done in a realistic scale, which allows for the player to take a longer time to explore the interiors. Since the game world is limited, there is a need to set some boundaries. These can be done in a way that does not explicitly reveal that they exist to maintain suspension of disbelief for the player, or they can be simply shown as walls around the world (Novak, 2012).

2.2.6 Interface Design

Interface design is traditionally known as user interface (UI) design. The importance of this in games is the connection between the player and the game, since the player actions need a connection to the game for them to be possible. With interface design in games, a *player-centered design* is important. The focus should be on the player's needs, tasks and choices. When designing the interface, it is important to view it from the player's perspective to make sure it is intuitive to use and not frustrating. The interface should be helpful and functional, and it should also fit the game's style and reflect the game's story. The visual style should fit things like setting, mood, time-period and the environment of the game (Novak, 2012). There are two categories of needs for the player: actions and information.

Player *actions* are what the player does in the game. This can involve navigating the game world, picking up objects or firing a weapon. These actions need to be accounted for in the player interface. This means that each action that can be done in the game needs to be accounted for when designing the interface. Anything related to the player character should also be included, such as access to different items like weapons. If there is character customization in the game, this is often done with a specific character creation screen that allows the player to view all the information about the character (Novak, 2012).

The game information affects the interface in different ways. If there is information in the game that is not available to the game characters, the players need to be able to access this information with the interface. This could include the player status, which includes the player's remaining lives and skills the player has available. This information can change through the game, so it is important that the player can access it when needed with the interface (Novak, 2012).

The game world can also be included in the interface design. One of the most common ways to navigate the level design structure is with the use of maps. The maps can be a separate screen or shown on the screen during gameplay. The player interface can also be modified based on which part of the world is being accessed, allowing for different actions or information to be accessed at specific points of the game (Novak, 2012).

There are two types of interfaces: physical and virtual. The physical interfaces include things like keyboard and mouse and different game controllers. The virtual interfaces have different types as well. These include the *active* interface, which is things that can be interacted with easily, such as a *menu system* that allows for navigating different screens in the game. An active interface can have things like buttons, text input and scroll bars. *Passive* interface is a type of interface that the player cannot interact with. This is things that are unchangeable, like the player's status screen, which is meant to show the current state of the player character and show information. Passive interfaces are meant to show feedback to the player, conveying important information. A type of passive interface is the *heads-up display* (HUD), which overlays the interface on top of the game, usually showing important information for the player at all times while the game is running (Novak, 2012).

2.2.7 Game Engine

The *game engine* is the software that the game is built on. For the design aspect of the game, this would include information about what the engine would need to be able to do (Novak, 2012). This is more relevant in the case of the game engine being developed alongside the game, but it could be helpful when designing for an existing engine as well.

This is where Schell's (2019) sixth filter *engineering* is the most prominent. It focuses on how the game is built and if it is even possible to build it. If a game engine already exists that allows for the game ideas to be included, it would be chosen here. If an engine like that does not exist, it could be created for the game. However, it would be reasonable to ask if it is possible to create the game or not. This is also true for hardware as well, since if a device that allows the idea to be created does not exist, it would be difficult to create the game. With a bigger budget, the solution to this could be inventing new technology, like a new controller.

2.2.8 Game Proposal

The game *proposal* follows the concept document. It takes everything from the concept document and explains it in more detail. Its purpose is to explain what the game is about to someone who is already interested in the game concept, but it can also be used for explaining game details to the development team members before they begin planning development. In addition to everything in the concept document, the proposal should include some new things. These things include the *hook*, which is meant to explain the best features of the game to get players interested in playing the game, any potential online features the game is going to have, any potential special technology that the game is going to use, as well as art and audio features, among other things. The gameplay portion of the game proposal should list the elements of the game that describe the experience of playing the game. The story section of this document should include the backstory of the game, a story synopsis and character descriptions (Novak, 2012).

Another part of the game proposal are the production details. This includes things like the members of the development team, budget and schedule. Other things to consider are what the production status is, which explains the phase of development the game is in. A rough estimate of the budget should be included in this. There can also be a proposed date for when the game

is complete, which is part of the schedule. However, the schedule and any dates given at this point, as well as the budget, are not considered to be final and are only meant to be estimates that show the ambitions of the project. The development budget often includes direct costs, cost of goods sold, marketing and income estimates. An independent production is going to have a smaller budget than a big company (Novak, 2012). However, it should be noted that overestimating the budget-range is not a good idea from a practical standpoint.

A *risk analysis* is also a part of the game proposal, and it goes over anything that could go wrong in the game development process and the plans that deal with these issues. Common things to include in this are personnel recruitment, material delivery, reliance on external sources of technology, competitive technology developments, and any issues with technology or design choices that can potentially impact the schedule. Another thing to include in this is an assessment of which parts of the project are relatively safe (Novak, 2012).

Finally, if there is any concept art available for the game, it should be included in the game proposal as well. These are drawings and sketches of characters and scenes related to the game. Other things to include are visualizations of screenshots that show what the game environment and characters could look like. A separate *art style guide* is where the art is established along with a look and feel of the game, and the purpose of it is to ensure a consistent style for the game (Novak, 2012). The game proposal does not go into as much detail in the game art as the art style guide does.

2.2.9 Game Design Document (GDD)

The *game design document* is the most important document for game design. It is longer than the concept document and the game proposal and it is not meant to sell the idea like those documents are. A GDD has a table of contents. It often changes daily throughout the development process as it is updated along with the game itself (Novak, 2012). They can be thought of as blueprints for putting a game together (Riendeau, 2023).

The GDD's purpose is to be a reference guide for the development team, containing every element of the game in detail. It includes aspects from the previous documents, focusing on gameplay, storyline, characters, interface and the rules of the game. These should be done in so much detail that someone could play the game with the help of the GDD even without a

computer or the game itself on a paper prototype for example. Everything in the game proposal is a part of the GDD, along with game interface, game world, character abilities, items and the game engine. The things to include about the interface are all the different types of interfaces, as well their reasons for being there. From the game world, it includes every element that each level of the game has and why they are included. However, it is important to note that a GDD can vary in its contents based on the details of the project. There is no standard for a GDD, and there are many templates available to choose from (Novak, 2012). For a solo developer, it can be helpful as a reference later in development to remember what the earlier ideas were (Riendeau, 2023).

The *technical design document* focuses on the technical aspects of the game. It is based on the GDD and focuses on how the game engine works along with what tools will be used to create the game. Other documents following the GDD are a *project plan,* which outlines the project development path, a *resource plan,* which lists the personnel and any purchases needed for the project and how much they cost, and a *test plan*, which goes over what areas of the game need to be tested throughout the development process (Novak, 2012).

2.2.10 Game Design Patterns

Game design patterns are tools for game design. They are a collection of possible design choices to be used in games. These patterns can help with making design choices when making a game, understanding how games work and inspiring game ideas (Björk & Holopainen, 2005). Björk, Holopainen and Lundgren (2003) made a template for creating game design patterns by examining game mechanics, analysing games and game concepts, and interviewing game developers.

Design patterns come from Christopher Alexander's book *A Pattern Language: Towns, Buildings, Construction* from 1977, which is a book about design patterns in building architecture. However, his ideas have been later adopted into other fields, such as software development (Neves & Zagalo, 2021). Gamma et al. (1994) applied Alexander's concepts to software design. They explain that there are four essential elements to design patterns: pattern name, the problem that the pattern is trying to solve, the solution that the pattern offers and the consequences of using the pattern, namely the results and trade-offs that the pattern provides

when used. A key point of a design pattern is that it is unique, with each pattern being distuinquishable from one another.

A *game design pattern* similarly consists of a name, core definition, general description, uses of the pattern, consequences of using the pattern and relations between patterns. The name is supposed to be short and descriptive of the pattern. The core definition describes the idea of the pattern in a short description to help browsing through collections without having to read the whole definition. The general description is a bigger explanation of what the pattern is and how it has been used in games, with some examples of games that contain the pattern. The uses describe the common choices that accompany the pattern and how the pattern can be applied in different ways. The consequences of using the pattern describe what happens when the pattern is applied to the game. The relations between patterns describes the relationship between different patterns and how they can be applied together, as well as patterns that conflict with the pattern that do not fit together (Björk & Holopainen, 2005).

Björk and Holopainen (2005) focus on patterns that have something to do with gameplay. They categorize common game design patterns into eleven broad categories in their book. Within these categories, the patterns are categorized further into sub-categories. The eleven categories are:

- Game Elements
- Resource and Resource Management
- Information, Communication and Presentation
- Actions and Events
- Narrative Structures, Predictability and Immersion
- Social Interaction
- Goals
- Goal Structures
- Game Sessions
- Game Mastery and Balancing
- Meta Games, Replayability and Learning Curves

There are many reasons to use game design patterns. Björk and Holopainen present game design patterns as an approach into describing games that is independent of existing research fields (at the time) or a part of the field of game research. They describe game design patterns as a way

research the way game components affect gameplay. Uses for game design patterns that they list include generating ideas, developing game concepts and designing games. Other uses include identifying competition and intellectual property issues, problem solving during development, and analysing and categorizing games and genres (Björk & Holopainen, 2006). One of the uses is solving problems for game interaction design. The designer can look at the description of the patterns and find a solution to the problem. Another benefit is gaining inspiration from a collection of patterns. A collection of patterns is a list of concepts that game designers have found useful, which offers a knowledge base for designing a new game. Game design patterns can also be used as a creative design tool by starting with a desired pattern before identifying and analysing potential sub-patterns that can help with completing the game design. Patterns can also inform the designer about potential risks involved with using the pattern, which can help with problem solving later or avoiding some problems entirely in the design phase (Björk, Holopainen and Kreimeier, 2003).

Another benefit to game design patterns is the ease of communication with others. By having descriptions for design patterns, it becomes easier to communicate the concepts of the design to those who have access to the information about the pattern without having to explain every detail about the design pattern. The same can be said for naming the design patterns, which allows for those who know about the pattern to immediately understand the concept when the name of the pattern is mentioned (Björk, Holopainen and Kreimeier, 2003).

As pointed out by Neves and Zagalo (2021), game design patterns were created as an attempt to establish a common language for game design, but they failed to do so at a large scale. They state that a lack of a wide-spread game design methodology contributes to many issues with the game industry as a whole. One of these is the fact that serious games have not become more widely used. Another part of this is that while software development processes have been integrated into game development, *user experience* (UX) development in games is a well-developed industry in its own right, and other fields like game analytics and user research have followed the industry's growth, game design still remains as a more undefined field.

Neil (2016) has some ideas as to why game design tools became relevant to begin with. Part of this reason is the fact that game innovation suffered as a result of game companies relying on mechanics that they knew already worked in other games. Despite of the increase in technology and production values, game design itself remained undeveloped. Game design patterns were

just one part of the push for game design tools in the early 2000s, with other examples including Raph Koster's "grammar of gameplay" (Koster, 2005). However, despite of the attempt to build momentum for game design tools, they ultimately failed to achieve their goal of bringing self-conscious design in games to the industry. Neil suggest this is due to the industry favoring speeding and scaling up production and testing, as well as the rise of the *agile software development* model, which brought along iterative development and customer collaboration (Neil, 2016).

However, game design patterns have been brought into pedagogy for design practice more recently, as they can be used to practice designing games as exercises (Barney, 2021a). Game design patterns have also been used for creating a design framework for a specific types of games, like with Sintoris's (2015) framework for location-based mobile games, which uses a set of game design patterns identified at game design workshops. Barney (2021b) suggests exercises to create patterns to form a personal pattern language, with the hope that the game developers in the industry would eventually combine their own patterns into a unified pattern language.

There are some common issues with game design patterns. One of these is that they do not consider the design problem they address, with the issue being that while the instructions for finding patterns ask the designer to identify the problem, this is not often the case. The format of the pattern ends up as "many games do x" rather than "in order to do y, designers may wish to do x" (Barney, 2021b). Barney therefore explains that the design patterns should include a problem statement that explains why the pattern is needed. He gives an example of creating a pattern from a first-person shooter game about limiting the amount of ammo available to the player, rate of fire or ammo capacity based on seeing a game where this is the case. However, he argues that a better pattern to create from this should be more general, talking about resources in general, and specifying that the reason for limiting it is creating tension. The pattern would therefore not be "developers should limit the amount of damage a player can do by limiting ammo capacity, rate of fire, or ammo availability", but rather "to create a cycle of tension and release in game levels, a designer may limit access to the resources that drive the core gameplay loops." By making the patterns more generic, they can be applied to more situations, rather than being specific to the given situation where it is applied in the game where the pattern is observed. Other issues that Barney identifies are circular patterns that tell designers to do a specific thing by implementing elements that create the specific thing, without going into what

this means, and another is jumping to conclusions, where a designer may take a small sample size of games to create a pattern, and then proceeds to apply that to all games, without considering it fairly for other games.

2.3 Game Prototype

The next step after the pre-production is creating a *prototype.* A prototype is a piece-of software or a low-fidelity model of the game that captures the essence of the game and allows for the game features to be tested. It is used to test gameplay mechanics and making sure the game is enjoyable for the players. The visual style and of the game are not as important with a prototype (Novak, 2012). There can also be multiple different prototypes made to test different variations.

Schell's (2019) eight filter is *playtesting.* It focuses on the feedback of the play-testers of the game to see if they enjoy it or not. It is arguably the most important filter, as if no one wants to play the game, it is basically pointless to make it. For this reason, it is important to get the game to a playable state as soon as possible into the development, to ensure there is enough time to address any potential issues that arise in playtesting.

The prototype can also be used to sell the idea of the game to potential publishers and investors, so it generally should show the main features of the game in a few minutes. If all the features of the game are not developed for a prototype, the feel of them can be simulated in a prototype. An example of this could be pre-rendered assets that would be rendered in real-time in the actual game (Novak, 2012). Another example could be a world generator that generates a unique world for each play session, which could be a pre-made world in the prototype.

However, it is not usually as simple as creating a prototype and immediately moving onto the production of the game. There is an *iterative* development process, which is also used in software and web development. After the prototype phase, the game then goes to an *evaluation* phase, in which the development team goes over the prototype and decides what works and what does not. (Novak, 2012). Feedback from internal and external playtests is a common aspect of iterative development, and it is addressed in the next iteration (Mostovoy, 2024). After evaluation, the game goes back to the design phase to add any modifications to the game, before a new prototype is created for testing. This often goes on until the game is completed, even during the production phase (Novak, 2012).

2.4 Game Production

After the pre-production phase and the prototype being approved comes the longest phase of game development: *production.* This is where the game is developed. It can take up to years with bigger productions, and the time it takes varies based on the scale of the production and who is setting deadlines (Novak, 2012). For the purposes of this study, the production and subsequent phases are not focused on.

After the production has come to a certain point comes the *alpha* phase. This can vary between who is making decisions on what the alpha version of the game means, but it is often a stage where the game is playable from the start to finish. The assets may not be final at this point, but the game engine and interface should be functional. The alpha phase focuses on polishing the game features, assets and mechanics, and each aspect of it is tested to root out any bugs that will most likely still exist at this point. Once the alpha version of the game has reached a certain point, it reaches the *beta* phase of development. This phase is mainly for fixing bugs, and all assets involved should be already integrated into the game. The goal of this phase is to eliminate as many bugs as possible before the game can be released (Novak, 2012).

2.5 Game Launch

Once there is a stable beta version of the game, the *pre-launch* phase begins. This is where the game is marketed. Bigger studios may reveal the game at a big event. Things involved in this phase are trailers, demo releases and a potential early access release (Bramble, 2023). A demo version or an early access release allows for the game to be played by players at an incomplete stage to get feedback and funding from players. They are often used as tools for independent developers for game testing (Kiselev, 2024). This is also called *soft launch,* which involves managing players' needs while continuing to develop features. This is also where metrics like *key performance indicators* (KPI) like daily active users, retention and engagement, are measured and used to improve the game. An issue with a soft launch can be the game becoming more successful than expected, which can lead to issues especially with online games, like players being unable to log in, lag and other load issues (Brownell, n.d.).

The *launch* phase is where the game is officially released. If there are any bugs remaining, these should be fixed at this point. Other additions may include simple improvements and adding final artistic touches to game models and environments (Bramble, 2023). When the game is ready for release, it is considered *gold.* This is the phase of the game development, where the game is approved for release. It is also where the game is sent to manufacturers for physical releases (Novak, 2012), if there is a physical release planned, which is increasingly rare (Orland, 2022). The phrase "going gold" comes from a time when the game was sent to manufacturers on a gold-coloured disc, but nowadays the process is digital (Ubisoft, 2021).

2.6 Post-Production

The launch is followed by the *post-production* phase (Novak, 2012) or *post-launch* (Bramble, 2023), which involves anything that is done surrounding the game after release. It includes *patches* that fix any bugs that are still found in the game after release and *updates* that add new content to the game. There is also the possibility of creating *downloadable content* (DLC) and *expansions,* which are software that are sold separately from the main game and come with new features for the game. DLC are often created in the post-production phase, but sometimes the game design itself already includes any future expansions and DLCs (Novak, 2012). As an example, with the post-production of the *Paradox Interactive* game *Crusader Kings III*, the developers are considering player feedback with each update and DLC to eliminate bugs and implement new features, treating it as a live service for the players (Argüello, 2025).

3 Developing the Game

Development of the Fantasy Mission game took place over the course of the year 2025. The development process was logged into a development diary to help with reciting the process later. This chapter goes over that development process starting with the design phase.

The design process of the game is explained in detail, starting with the concept. This is followed by going over the choosing process for the game design patterns for the study and what patterns were chosen. The design of the game is described after the patterns. Following this are the details of the prototypes that were made during the development process without going into detail about the code or internal architecture of the game, as that was not deemed necessary for the goals of the thesis. For the full game design document, see Appendix 2.

3.1 Concept

The first step of designing the game was coming up with the game concept. The goal for the game being designed for this study is to study how game design patterns affect the development process, so the actual game idea itself should be reflective of that. However, the game is still meant to be for entertainment.

3.1.1 Game Ideas

At the start of the study, there were ideas created for what kind of game to develop to answer the research question. These ideas came from a personal list of ideas that had been created over the years for potential games that could be made.

Out of the list of ideas, three game ideas were chosen to explore further. The first one was an American football style of game based on a flash game called *Crunchball 3000*. The second idea was the 1000-year game, with the idea being that the player manages a colony on a spaceship for a thousand years. This is based on the game called *The Longing* that lasts for 400 days of real-time, where the player does tasks that advance the time. The third option was a game called Fantasy Mission, a turn-based strategy game in a fantasy setting, where the player manages units and resources.

Fantasy Mission was the one that was finally chosen as the game to develop. The first idea based on *Crunchball 3000* would have been a 3D version of the flash game, taking most of the details from the already existing game. This made it feel too fully developed already, which did not work well with the target of the study. The 1000-year game based on *The Longing* felt like it had potential and uniqueness. However, the concept felt too unique for the purposes of the study. It also felt like it would be difficult to apply the game design patterns to it in a compelling way, as there was no clear idea for what the genre of the game would be. Fantasy Mission on the other hand had a basic concept to it with less details for the gameplay, which made it the best candidate out of these ideas for the study.

3.1.2 Target Audience

With the idea having been decided, the concept was explored further. The first step was figuring out who the game is for. The idea for the target audience was that the game would be for people who enjoy playing strategy games, but do not necessarily want to play an overly complex game, like many strategy games tend to be. With the fantasy setting, the game is appealing for escapism to escape the stresses of everyday life, and it does not necessarily need to be realistic. The game is geographically targeted for the western audience. For the demographic, it is mostly meant for a more general audience, but not young children. This makes the target audience teenagers to adults in the western world.

The genre of the game is *turn-based strategy*. Strategy games originate from classic board games that require the player to manage resources to accomplish goals. The setting for these games is usually conflict or war. The focus is on the player's resources instead of a player character (Novak, 2012). This is also true for Fantasy Mission, which involves managing units to battle other units to gain resources. Strategy games have strategic elements to them, and as concluded by Dor (2018), strategy games as a genre consist of games that have a certain tradition that goes back to historic wargames. Strategy games typically challenge the player to achieve victory with the use of planning a series of actions taken against opponents. Turn-based strategy is one of the main sub-genres of strategy games (Adams, 2014). The *turn-based* nature of these games allows for the player to think about their decisions for longer and in more detail, which fits well with planning strategies (Novak, 2012).

The *turn-based* time interval in games comes from traditional board and card games, usually allowing for a player to do a limited set of actions like playing a card on their turn before the turn changes to the next player. Turn-based games have an unlimited time for a player to do actions, as opposed to how *time-limited* games have a limit to how long the player can take to do their actions during their turn (Novak, 2012).

3.1.3 Marketing

While the goal of making this game is not to sell it, marketing is still an important part of the design process. While designing Fantasy Mission, there were several things considered that relate to the marketing aspects.

The idea of the game is to be a combination of Sid Meier's *Civilization* games and *A Game of Thrones: The Board Game*, with a heavier emphasis on *Civilization*. Fantasy Mission is basically *Civilization* with fantasy elements. For the competitive analysis, there are other games like this already out there. Besides the *Civilization* games, there is a game called *Humanity*, which is similar. For games like this that have a fantasy element to them, there is a game called *Endless Legend,* which features similar gameplay to *Civilization* with fantasy elements. Other games like this are the *Age of Wonders* series of games, which feature similar mechanics as *Civilization,* with a fantasy setting.

The uniqueness factor considered for Fantasy Mission comes in the form of events that affect the gameplay outside of the players' choices, which changes aspects of the game between turns taken. It has hero characters that are stronger than normal units. Another idea was that the players could use cards as part of the combat system to affect the gameplay, which was unfortunately scrapped during development to simplify the development process. This card idea came from *A Game of Thrones: The Board Game*, as did the idea of the hero characters.

Other things considered as a part of this are the target platform, estimated target rating and the premise of the game. The target platform of the game is PC. The estimated target rating is somewhere between 7 to 12. There is no intention of adding any mature elements, but there is violence between units, as it is a war game, which can bump the rating up to 12. However, this is not expected to look like real life violence, which would not make it reach the 16 rating. The

premise of the game is: Take control of a faction and lead them to victory with the aid of your heroes and magic. May the best heroes win!

3.2 Choosing the Design Patterns

For developing the game, a set of game design patterns was chosen to be used in the game design. The idea was to use the design patterns as building blocks for the game to help with the design process. Initially, eighty patterns were chosen out of the design patterns listed in Björk and Holopainen's book *Patterns in Game Design* (2005) for further examination. The way these were chosen was done by examining the patterns in detail and analysing whether they would work for a turn-based strategy game. Some patterns that could have been considered for a turn-based strategy game were also left out either due to similarities with other patterns, or simply due to lack of interest in using the pattern. Some of the information on the patterns listed in that book were only available on an accompanying CD, which was not available for this study. There are 166 patterns in the book that have information on them available without the CD.

The original plan was to take one pattern from each of the eleven main categories. This was later changed to choosing around ten patterns out of the eighty initially chosen to use in designing the chosen game. For this study, the patterns were further classified into three study categories: *Mandatory Patterns*, *Desirable Patterns* and *Research Patterns*. For a table on the 166 patterns and their classification for the study, see Appendix 1.

3.2.1 Mandatory Patterns

The idea of a Mandatory Pattern is that there are game design patterns that a game designer will most likely end up using without necessarily intending to do so, as games of the chosen genre will make it almost mandatory to use them for the game to be made. These are patterns that appear in almost every game of the chosen genre. For Fantasy Mission, these are patterns that are most likely needed for a turn-based strategy game. This is also the reason why these patterns were not included as potential Research Patterns, as they would not have steered the design of the game into a different direction as opposed to some more unique patterns for the genre.

As mentioned earlier in this thesis, strategy games go back to wargames, as stated by Dor (2018), and therefore involve elements of warfare in them. They also usually have some levels of resources management in them, as stated by Novak (2012). These factors were some of the biggest reasonings for the patterns that were classified as mandatory. Some of the most important patterns included for this reason are Units, Resources, Producer-Consumer and Ownership. *Units* are game elements that are under the player's control and let the player perform actions that influence the game world (Björk & Holopainen, 2005). Units are essentially soldiers in Fantasy Mission, allowing for players to attack others. *Resources* are game elements that allow players to enable actions in game. *Producer-Consumer* is a pattern that governs the flow of gameplay as it determines the use of game elements such as resources. For example, when used with resources and units, the producer produces units and consumes resources. *Ownership* is a pattern that determines which players have access to which resources and other game components and how this happens. For example, it can dictate the ownership of Resources or Tools (Björk & Holopainen, 2005).

There are also patterns that are broad enough that one could argue they will be included in almost any game that is made regardless of genre. For example, the *game world* pattern is something that is most likely in every game. It is the environment where the gameplay or parts of the gameplay take place (Björk & Holopainen, 2005).

3.2.2 Desirable Patterns

The Desirable Patterns are patterns that were deemed patterns that the players want in a turn-based strategy game but are not necessarily needed for the game to be created. After classifying the Mandatory Patterns, the Desirable Patterns were classified by examining the remaining patterns' descriptions and using them to determine if it was something that I would personally want in a game like this. These patterns were something that fit being added to the game design when designing the game. They were later considered as potential Research Patterns to use when designing the game for this study. These include patterns like Avatars, Alarms, Score and Strategic Location.

The idea behind some of these patterns is that they are just nice to have, while others are more about the functionality of the game itself. The common idea between these patterns is that they fit into a turn-based strategy game but are not necessarily something that must be in it for the

game to function. *Avatars* is one of the patterns in this category that is nice to have, as it is a pattern that defines the player character in a lot of games. It is a game element that allows the player to affect the world and usually represent the players (Björk & Holopainen, 2005). It is not something that is needed for a strategy game, but some games in franchises like *Civilization* and *Age of Wonders* do have a player avatar, which is nice to have. On the other hand, a pattern like *Alarms* is more about functionality, as it is a pattern that shows a message to the player when something is about to happen (Björk & Holopainen, 2005). This seems like something that is good to have if there are events that occur in the game, but it is not necessarily something that must be in the game.

3.2.3 Research Patterns

Research Patterns are the patterns that were chosen to be implemented for the purposes of the study. The thought process behind which Research Patterns were chosen for the study was straightforward. After going over all the 80 patterns that were chosen from the book, and labelling which ones were Mandatory Patterns and which were Desirable Patterns, the Desirable Patterns were reconsidered for the Research Patterns. This was done by examining each of the patterns and imagining how they could be implemented to a turn-based strategy game. The Research Patterns chosen for this study are:

- Paper-Rock-Scissors
- Character Development
- Ultra-Powerful Events
- New Abilities
- Improved Abilities
- Ability Losses
- Decreased Abilities
- Pick-Ups
- Power-Ups
- Tools
- Alliances
- Betrayal
- Randomness
- Surprises

The *Paper-Rock-Scissors* pattern is based on the Paper-Rock-Scissors game, where the players choose either rock, paper or scissors to try to beat the other player. Rock beats scissors, scissors beat paper and paper beats rock. The idea with this pattern in game design is that players must somehow figure out which choice in the available options is the best in each situation (Björk & Holopainen, 2005). For example, using some type of combat system where each character has an element assigned to them that is strong against one element and weak against another. The reason why this was picked is that it is a pattern that easily brings balance to the game mechanics and the combat system if implemented correctly, and it is not overly complex way of creating a combat system.

Character Development as a pattern is broad. It consists of improving a character's skills or knowledge. Any type of change to the characters during the game can be considered character development. This could mean becoming more likely to succeed in actions or add actions that were previously unavailable to the character. Character Development has two main characteristics to it: what caused the development and what effect the development has (Björk & Holopainen, 2005). This was picked because the idea of units improving as they go through combat was appealing, and it adds some reward to the risk of losing a unit when attacking another unit.

Events are essentially things that can happen that are system-governed by the game, usually with a major effect on the current game state of the players. The *Ultra-Powerful Events* are events that cannot be affected by player actions; they happen regardless of what the players want to do. These can be a part of the game environment, or they can happen because of an action done by a player, but the player cannot influence the event once it has started. The two ways Ultra-Powerful Events are impossible for the players to affect are that the player cannot affect why and when the event starts, and they cannot affect the events after they have begun (Björk & Holopainen, 2005). This was picked as a pattern, because it can bring something unexpected to the game, as it is out of players' control, and it makes the game more interesting in theory, as it can change the tide of the game if the right event occurs when a player needs it.

New Abilities, *Improved Abilities*, *Ability Losses* and *Decreased Abilities* are classified as *Action Control* type patterns, meaning they govern how basic actions can be used in a sequence and how players can access all the different actions in the game. New Abilities consists of gaining abilities during the game. Improved Abilities is a pattern where already existing abilities

become more effective over time, increasing a chance of success with an action. Ability Losses allow for players to lose an ability to do an action during the game. Decreased Abilities is a pattern where the player's ability loses effectiveness over the course of the game (Björk & Holopainen, 2005). These were picked as patterns to help with the Character Development pattern, as there should be some way to show the units improving if they are being developed.

Pick-Ups, *Power-Ups* and *Tools* are all somewhat similar as patterns. Pick-Ups are game elements that can be collected by players. This happens usually by moving an avatar or Units to the game element to cause this action. Power-Ups are a special type of Pick-Up that often give the player an effect, like New Abilities. Tools are typically Pick-Ups. They enable the player to perform actions that are unavailable to them outside of having access to the specific Tool (Björk & Holopainen, 2005). These were picked as patterns to add more complexity to the game and to give more rewards to the players for smaller tasks like exploration.

Alliances and *Betrayal* as patterns are Game Design Patterns for Social Interaction. These types of patterns usually involve interaction between two or more players in the game. Betrayal can happen when players create an agreement to intentionally fail to do as agreed with another player. Alliances on the other hand are an agreement between players to work together (Björk & Holopainen, 2005). These were picked to formulate a diplomacy system for the game and force some level of social interaction between the players, as without this there would be no real consequences of attacking another player.

Randomness is a simple pattern. It is basically anything that makes the game unpredictable. *Surprises* differ from randomness in that they are something unexpected that happens in the game that cannot be predicted by the players. Randomness is also associated with game balancing, while surprises are narrative structures (Björk & Holopainen, 2005). Randomness is something that could be argued is important for strategic thinking, as it forces the player to think about how to adapt to the unexpected results or events that occur (Przybylski, 2021). These patterns were picked to make the game less about simply the stronger unit winning every time, which would get boring quickly, as well as to make the game potentially more interesting.

3.3 Design of the Game

Before fully getting started with developing the prototypes, the game was designed. A game design document was created to figure out the main aspects of the game. This happened around a week into game development, but the game was still in its early stages at this point, so the development was not negatively affected by the delay. The designed game was named Fantasy Mission at this point.

3.3.1 Gameplay

The gameplay happens by having each player be able to move their units, do unit actions and produce more units on their turn, after which the turn changes to the next player. The player controls a floating camera that can move around the world, with the ability to move own units around the map. The map itself consists of hexagonal map tiles that the units can move across, with each tile having its own terrain value that determines things like movement speed and resources. The point of this is to get a feel of a living board game. The main objective of the game is to keep producing units to be able to defeat the opposing factions in combat, until there is only one faction left. The game is in 3D with a top-down POV.

Fantasy Mission has only one *player mode*, which is *local*. Player mode defines how many players can play the game at the same time. The local player mode allows for multiple players to play the game in the same space, usually on the same device. These games can share the same input device, like a keyboard, or use separate ones, like multiple game controllers (Novak, 2012). Fantasy Mission is played with one mouse and keyboard. It was planned for the game to be able to be played as both single-player and local, but a decision was made to not develop an algorithm or an artificial intelligence that could control the opponent. This was mostly due to lack of time and player mode's irrelevancy to the chosen research patterns.

The main objective of the game is to eliminate the other players, and the winner of the game is the last player remaining. This happens by defeating other factions' units and taking over their cities. A player loses the game when all units and cities they control have been destroyed or taken over by other players. Other objectives in the game are upgrading units and cities by gaining experience points from battles and from creating units. The players can also explore ruins to gain power-ups.

There is also a diplomacy system, which handles the player's relationship with the other players. Each player has a relationship with one another, which is either neutral, ally or enemy. Everyone starts the game with a neutral relationship with each other player. If a player declares a war on another player or attacks its units or cities, those players both gain the relationship of enemy with each other. If a player sends an alliance request to another player and it is accepted, those players become allies with each other. If a player sends a peace offer to an enemy player and it is accepted, those players' relationship with each other becomes neutral.

There is a *resource management* system in the game. Resource management means having the ability for players to manage settings and actions associated with their resources or characters (Novak, 2012). In Fantasy Mission, there are two types of resource management. First is cities owning map tiles as territory, which provide resources to the player who owns the city. These resources can be used to upgrade units and cities. Units can only be produced when the player has enough money to produce the specific type of unit. See Table 3.1 for how much each unit costs to produce. The second type is with combat actions that the units can take, which is explained in the character section later in this thesis.

The initial design included the players having a deck of cards that give them abilities to use in combat, but this was dropped in the implementation phase. There was also an idea of completing simple quests as the game progresses, but this was not included in the game at this time. There was a betrayal system that was intended to work by having player units or cities potentially turning against the player and joining an enemy faction, but this was never implemented. Another part of the design in the beginning was to include monsters in the game that could affect the players by attacking them, but this was scrapped during the development process.

3.3.2 Balancing

The game balance in Fantasy Mission is symmetrical. Each player starts with one settler unit and that is it. This is the same for everyone. However, the random nature of where the units spawn can give an advantage to a player, if they happen to spawn in an area where there are many resources nearby. There is also a static balance used for different unit types (see Table 3.2).

Each faction included in the game has an element associated with them. These elements function with the rules of the *Paper-Rock-Scissors* pattern. There are five elements in the game that have their own strengths and weaknesses: Fire, Water, Earth, Wind and Neutral. Fire is strong against Wind, Water is strong against Fire, Earth is strong against Water and Wind is strong against Earth. These elements are weak against the elements that are strong against them. Neutral is simply neutral against the other elements with no strengths or weaknesses (see Table 3.1).

Table 3.1 Elements with their strengths and weaknesses.

Element	Strength	Weakness
Fire	Wind	Water
Water	Fire	Earth
Earth	Water	Wind
Wind	Earth	Fire
Neutral	-	-

Each unit has an element, which is determined by which faction they belong to. Each faction has percentages for how likely it is for a produced unit to spawn with each element, with one element being the most likely. These elements give a multiplier to the damage done in combat against the elements they are strong and weak against, either increasing the damage output or decreasing it. Some attacks use the element assigned to the unit, while others are always Neutral.

There are also *ultra-powerful events* called world events that the player cannot affect. These can change the balance of play with different events that occur at random intervals every now and then at the start of a turn. There are four types of world events: element boosts, unit boosts, experience boosts and no boost. Element boosts give a random event an increase in damage dealt to other units. Unit boosts are like element boosts and gives a specific unit type an increase in damage dealt to other units. Experience boosts give all units and cities a boost an increase in experience gain. No boost does not give any kind of boost and is neutral. Each world event has a random duration, which makes the players unable to predict when the event is expiring, and when the next event is occurring.

3.3.3 Story

The story of Fantasy Mission is not something that is highlighted in the game. A backstory was made to give a motivation for the fighting that happens in the game. After a catastrophe destroyed all civilization in the world, new factions have formed, each of them trying to bring peace to the world, with their own motives and reasons for doing so.

The setting of the game is a fantasy world in a less advanced civilization than our own. The technology has not advanced to a point where guns have been invented. However, people of this world are all capable of using magic. The world is full of ruins, and each faction starts with just settlers, which indicates that something catastrophic has happened before the start of the game. Each faction controls one element of Earth, Fire, Water and Wind. These factions are Earth Nation, Fire Nation, Water Nation and Wind Nation, with the names corresponding to the element they control. Most people are born with the element of their faction. However, there has been enough interaction between people of different factions to the point that each faction can also produce people with control over different elements than their own.

The plot of the game forms as the game goes on. It is not driven by a set narrative, but rather by the players' actions. The players choose where to settle new cities, which units to spawn, which ruins to explore and who to go to war with. The idea originally was to have quests that affect the story of the game, along with having events that affect the story, as well as faction-specific endings that would be displayed when a faction has won the game, but this was not implemented.

3.3.4 Characters

The characters in the game are units. They have no names and each of them has a class that indicates what type of unit they are. There are five types of units in the game: settler, warrior, archer, scout and mage. Each of these types of units can become a hero, which is a stronger class that retains the characteristics of the original unit type while gaining new abilities. Each type of unit starts with their own statistics, which are set for each of them, but they can be improved by gaining experience and levelling up the units. Each unit has an element associated with them, which is assigned at production. This defines which types of element attacks are

strong or weak against them and which element they can use in their attacks. There is no player character or NPCs. For the detailed statistics of each unit, see Table 3.2.

Settlers are the weakest units in the game. They can fight if they must, but it is not something they do well. However, they are extremely important for their ability to settle new cities, which allow for finding more resources and expanding the faction's borders. They have the least amount of action points per turn, as well as the weakest attack, defence and health of all units. They are the cheapest unit to maintain, but most expensive to produce. They also yield the least amount of experience when killed. Each player starts with one settler unit at the beginning of the game.

Table 3.2 Base statistics of the units in the game. Experience Yield means experience gained by other units when this unit is killed.

Unit Type	Actions per turn	Attack	Defence	Health	Production Cost	Maintenance Cost	Experience Yield
Settler	4	1	1	5	5	1	10
Warrior	5	4	4	10	3	2	40
Archer	5	3	3	8	3	2	40
Scout	7	2	2	7	2	1	20
Mage	4	6	4	6	4	3	50

Warriors are strong close-range fighters. Their main purpose is to fight other units. They have tied second most actions they can use per turn, the second strongest attack and defence and the most amount of health. They are the tied second cheapest unit to produce and the tied second cheapest unit to maintain. They also yield tied second most amount of experience when killed.

Archers are like warriors, with a few exceptions. They have the same number of actions they can take per turn, the same production and maintenance costs as well as the same experience yield. However, they have lower attack, defence and health. The reason for these statistics being weaker than warriors is their ability to use ranged attacks, which makes them able to attack warriors before the warriors can attack them.

Scouts are the best at moving across the map. With the highest actions per turn, they can move the farthest in a single turn. They can also attack at close-range if they must, but their attack statistic is not much better than a settler's, being the second weakest in the game. They also have a weak defence statistic. They have a decent health, and the cheapest production cost and

tied cheapest maintenance cost. They also yield the second least amount of experience when killed.

Mages are the strongest attackers in the game. They have access to a magic attack, which uses the element of the unit to deal element damage to units at long-range. The other units only have neutral element attacks, but mages can use one element that is assigned to them. This allows them to deal more damage to units with a weak element against their element, but it also means that units with a strong element against them receive less damage from their attacks. Mages have the tied least number of actions per turn, the second strongest defence and the second weakest health. They are also the second most expensive unit to produce and the most expensive unit to maintain. They also yield the most experience when killed.

Heroes are not units in the game. Hero is a special status a character can have. Any unit can become a hero and gain a boost in statistics as well as a new hero attack, which is stronger than the other attacks in the game. This attack uses the element assigned to the unit just like the mage's magic attack does. This hero promotion can happen randomly at battles, when levelling up and by exploring ruins. A unit can also be randomly produced as a hero.

3.3.5 Level Design

There is only one level in Fantasy Mission. It was originally intended to include multiple quests in the game that would act as different levels, but this was cut due to time constraints. The level in Fantasy Mission is a procedurally generated map, which is generated at the beginning of the game. The scale in the game world shows the units as big as the cities, which allows for the units to be seen easier. The game is turn-based, which allows the players to take as long as they need to do actions on their turns.

The map consists of hexagonal map tiles, which have different terrain types that affect unit movement and provide different resources to the player that controls the tile they are on. There are several terrain types. These terrain types have a movement cost that determines how many action points it takes for the unit to move to it. Resources are allocated to map tiles based on terrain type (see Table 3.3). There are four types of areas in the map: arctic, tundra, desert and variety. The arctic areas only contain Glacier and Ocean terrains. The Tundra areas contain

Tundra, Plain and Mountains terrains. The desert area contains only the Desert terrain. The variety areas contain Plain, Hills, Swamp, Marsh and Plateau terrains.

There are three types of resources that can be gained from map tiles: food, production and wood. The amount that each map tile provides for the player is randomized during map generation for each tile. There were other types of resources planned, but resources currently only work by giving the player money and the ability to produce troops. For this reason, the other resources were not added to the current state of the game.

Table 3.3 List of map tile terrains, their movement costs and the resources they provide.

Terrain Type	Movement Cost	Resources
Plain	1	Food
Hills	2	Food, Production, Wood
Mountains	4	Production
Swamp	3	Production
Marsh	2	Food, Wood
Glacier	3	-
Ocean	Impassable	Food
Plateau	2	Production
Desert	2	Production
Tundra	1	Food, Wood

There are ruins in the map that can be discovered. When a unit moves to the ruin, they gain a *power-up*, which provides them with a beneficial effect. There are four types of power-ups: statistic boosts, new action, level up, and hero promotion. Statistic boosts give the unit an increase in a statistic at random, which can be number of actions per turn, attack, defence or health. New action gives the unit a new type of action at random, which can be a ranged attack, magic attack or the ability to settle a city. Level up raises the unit's level by one. Hero promotion gives the unit a hero status and the access to the hero attack action.

There are several objectives in this level. These objectives are all up to the player to pursue as they wish. The players can create cities, produce units and combat other factions. The game win condition of being the last remaining faction to win the game is the main objective of this level. The duration of the game varies. It can take a short time if there are only two players and they play aggressively, or it can take a long time with multiple players who prefer to build up

their factions instead. The game gets progressively more difficult, since the longer it goes on, the more units and cities each player is going to have. The units also gain experience with each battle they survive, which makes them harder to defeat.

3.3.6 Interface Design

There are multiple user interfaces (UIs) in Fantasy Mission. Interface design was not the focus of creating the game or the study, so the aesthetics of the UIs were not focused on. Strategy games often have specific UIs suited for resource management. They also do not show all information at once so that the player does not have to look at all of it at once. Instead, the information is divided into separate windows or screens (Novak, 2012).

Figure 1. The name overlay UI above the units (left), cities (centre) and ruins (right). The units and cities also have a level, indicated by the number, and two bars that show the current health (top in red) and experience (below in blue).

In Fantasy Mission, the first interface the player is greeted with is the faction selection screen (see Appendix 2). This is where the players are added and the factions are selected. The faction selection happens by clicking on the drop-down box. The red buttons next to them can be used to remove players. The button on the top with a plus symbol allows for adding more players to the game. There is also a possibility to change the names of the players. The player is brought into the game when the start button is pressed.

The game HUD (see Appendix 2) shows the most important information the player needs to see. There is a resource list on the left side and a unit list on the right side of the screen. There

are multiple buttons in the HUD. There is a player list button on the top left, a next turn button on the top right, a world event button on the top centre, and a quit button on the bottom right.

Figure 2. The map tile information UI. Shows the terrain type, movement it takes for the selected unit to reach it and the resources it provides.

There are also other UI overlays in the game. The units, cities and ruins all have a UI indicator above them that tells the player information about them. These overlays show the name, type of unit, level and the faction colour they belong to (see Figure 1). There is also an overlay that shows the information about the map tile that is hovered (see Figure 2). This shows the terrain type, resources the tile provides and movement cost for the selected unit if moving. The world event button shows an overlay for what type of world event is currently active (see Figure 3). When the next turn button is pressed, the turn is changed to the next player. This displays the start of turn screen (see Appendix 2).

Figure 3. World event overlay.

The player list button opens the diplomacy screen (see Appendix 2). This is where diplomacy actions can be taken. It shows the list of players, their factions and relationships of them with

the current turn player. There are buttons alliance and war actions, which affect the relationships with the factions. The alliance offer button allows for an alliance offer to be sent. The war declaration button declares war on the other faction immediately. The war end button makes the player able to send a peace offer to the enemy player. There is also an end alliance action, which can be taken to send an alliance dissolution message to the other player.

When a unit is selected, the unit action overlay UI is displayed (see Figure 4). This is where the information about the unit is displayed, showing the unit type, faction, element and actions available for the turn. It also has a close button on the bottom left, which allows for the UI to be closed. The actions are showed on the right in a list of buttons, which can be clicked. Similarly, when a city is selected, the production screen is displayed (see Appendix 2). This shows what units can be produced, along with the production resources required to produce the unit.

Figure 4. Unit action overlay. This shows the unit information and the actions the unit can take.

There are also pop-up events that show a new event that has been activated (see Figure 5). These can be received from ruins, which give the unit that moves to the tile the ruin is on unit a power-up. The acquisition is displayed as an event. The other types of pop-ups happen when a new world event is activated, as well as all diplomacy messages that occur, like a war declaration.

Finally, when the game is over and a winner has been decided, the victory screen is displayed (see Appendix 2). This screen displays the name of the winning player, as well as buttons for restarting and quitting the game.

Figure 5. An event pop-up in Fantasy Mission. The image shows a new World Event that has been activated.

3.3.7 Game Engine

The game engine chosen for the development of this game was Unreal Engine 5. The reason for this was because the game was intended to be in 3D, and there was prior knowledge on how to use Unreal Engine prior to this work. The other option was Unity, which also has 3D capabilities. The reason that Unreal Engine ended up as the one chosen for this project over Unity was because of personal preference. There was no need to create a new engine for a project like this.

The version of Unreal Engine 5 that the game was created with is 5.5.4. The game project was created as a C++ project, started with the top-down template provided with the engine. Unreal Engine's own visual coding language called Blueprint was also used alongside C++ in the development of the game.

3.4 Prototypes

There were three prototypes created in the development process of Fantasy Mission, each with new features added to them. The third prototype is the final build of the game where the development of the game ended. If this game were to be fully developed, it would move onto the production phase next, but this is unnecessary for the purposes of the study.

3.4.1 First Prototype

In the beginning of development, the first step was to create a C++ project with Unreal Engine. This was simple enough, but there were some issues with compiling C++ code with Microsoft Visual Studio 2022, but it did not take too long to fix this. After this, the first real addition to the game was a floating camera pawn and the controls to allow for the player to move around the map. A camera zoom feature was also added to the camera, allowing for the player to adjust how close to the map the camera is. This was done using the Unreal Engine enhanced input system. Once this was done, the next step was creating the units and cities.

The units and cities are similar enough that the interaction system where the player can select units and cities handles both. The initial idea with the units was to have the player possess the unit pawn when selected, but this was quickly changed. It was also decided that it should be possible to select multiple units at once. The interaction system was also heavily modified from the one provided by the top-down template by default.

The map tile system was created after this. It generates a new map as a grid from hexagonal map tiles. This design choice was made because it is the way the maps work in Sid Meier's *Civilization* games. The hexagonal design also seemed to work best with the navigation system that was created. The map tiles keep track of their adjacent tiles, which allow for the navigation system to calculate the best path to a wanted destination. The navigation system itself takes a *Manhattan distance* between the unit's current tile and the destination tile and calculates the fastest route looking at the adjacent tiles of the map tiles, based on the movement cost of the tile's terrain. The units and cities of the players are also spawned in with this system at the start of the game, initially giving the players one city and one unit each. After this, the first prototype build of the game was created (see Figure 6).

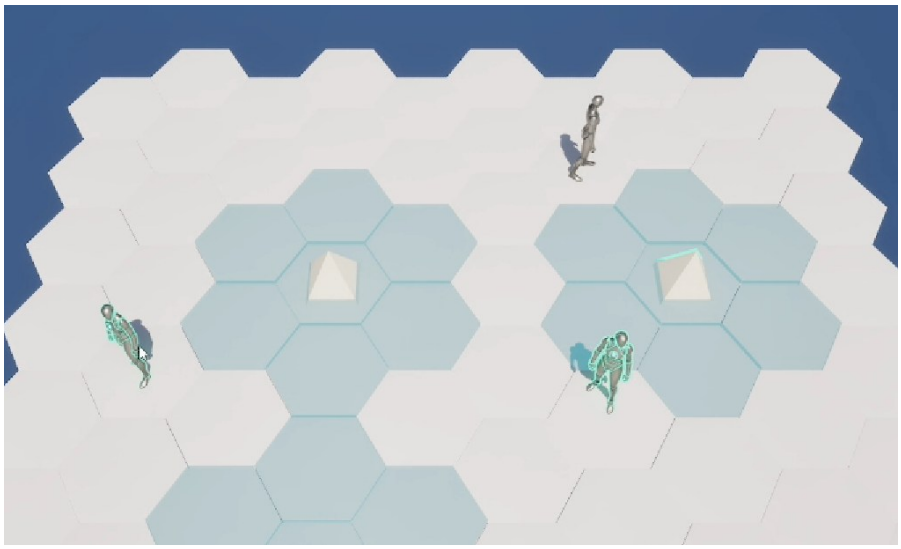


Figure 6. Screenshot of the first prototype of Fantasy Mission.

The first prototype version of the game is more of a proof of concept than an actual game. It features the generated map with plain tiles, cities with their territory tiles displayed and units that can be selected and moved around the map. The units and cities are highlighted when selected.

3.4.2 Second Prototype

For the second prototype build, the work focused on both data and visuals (see Figure 7). The first addition was adding factions to the game. This allowed for multiple players to be added to the game. The turn-based system was created following this, allowing for each player to take turns. A design choice that was made here is that the game will only feature human players to not have to worry about creating a somewhat complex enemy artificial intelligence to control enemy units. This made finishing the game for this study faster and simpler.

For the visuals, the biggest change was the addition of colours. The city territory tiles and units were changed to correspond to the faction colours, and the map tiles were changed to correspond to the territory colours. This allows for the players to immediately see which faction a unit belongs to. The territory colours are not as clear, but they add some life to the game regardless. The navigation system was highlighted to allow for the player to see the path the unit is going to take when moving.

The unit types were added to the game at this point. It was decided that the players would start with only one Settler unit. The units were given actions, mainly to finalize the movement between tiles. One of these actions is settle, which allows the Settlers to found cities. The user interface was also added to the game to give better indicators to the unit types and for using actions.

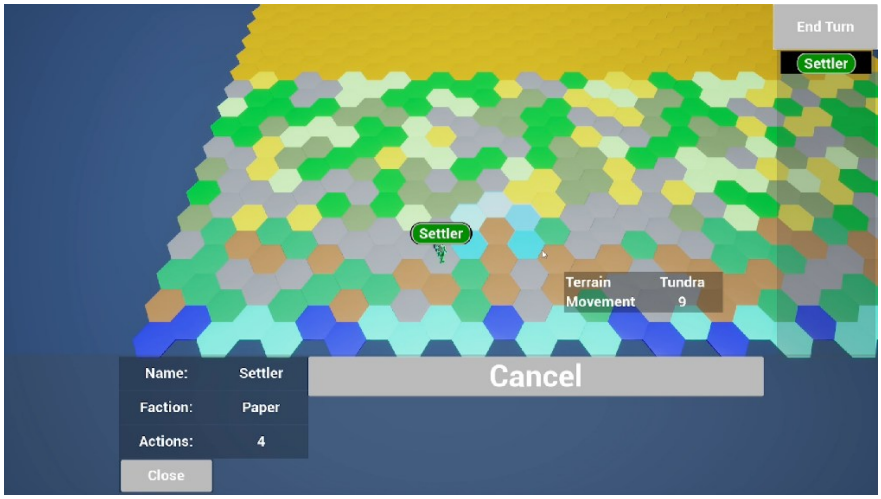


Figure 7. Screenshot of the second prototype of Fantasy Mission, showing movement. There is a blue overlay on the tiles that are on the movement path.

3.4.3 Final Build

For the third and final prototype, the resources were added to the map tiles. The initial design idea was to collect resources every turn per map tile, but this turned out to not be a good idea, since the resources would stack to unreasonably high numbers. This was later changed to having the resource amount be tied to the resource number on the map tile, which replenishes each turn. The user interface was also tweaked to include a resource list for the turn player.

The next addition was the level system for units and cities. The way this works is that the units gain experience points from combat and cities gain experience points from producing units. The production system allows for units to be produced based on what resources the player has. There is also an experience bar that shows how much experience is needed to reach the next level.

The combat system was also added. It allows for units to attack other units and cities with actions to gain experience (see Figure 8). The units have combat attributes that determine how

much their attacks do and how well they can defend against attacks. They also have health bars that show how much damage they can take. Instead of using cards as initially planned, the attacks are done with actions, same as movement. These attack actions are tied to the action points the units have available to use each turn. The elements were also added to the game. Each unit has an element that affects the damage dealt in combat.

Another big addition for the final build was the event system. The ruins that give power-ups to units were the first part of this, as the power-ups give the players pop-up events. World events were also added. Only one world event can be active at once, and the active world event is changed after a random number of turns has passed since its activation. The world event currently active can be viewed with the world event button at the top of the screen.

There were some final additions to the game as well. The diplomacy screen was added, finally allowing the players to do diplomacy actions like creating alliances and declaring war. The last thing added to the game was the win condition and the ability to be eliminated from the game. Once a player loses every unit and city, the player is eliminated from the game. Once there is only one faction remaining, that player wins the game.

Figure 8. Screenshot of the attack mechanic in the final build. There is a red overlay on the targeting tiles that show which tiles the unit can attack.

4 Results

With the game development finished for the purposes of this study, it was time to start looking at what was accomplished. This chapter goes over the results of the research, focusing on which patterns were used in the development of the game, which patterns were left out and why, and how the patterns shaped the game itself. For the full list of all game design patterns used in the study and if they were included in the game design or not, see Appendix 1.

Table 4.1: Numbers of included and excluded patterns by study category.

Study Category	Included Patterns	Excluded Patterns	Total
Research Patterns	10	4	14
Desirable Patterns	18	19	37
Mandatory Patterns	28	1	29
Total	56	24	80

4.1 Included Design Patterns

Out of the eighty design patterns chosen for the research initially, fifty-six ended up being included in the game on some level. Ten of these are Research Patterns, eighteen are Desirable Patterns and twenty-eight are Mandatory Patterns (see Table 4.1).

Ten out of fourteen of the chosen Research Patterns were included in the game (see Table 4.2). These patterns ended up shaping the game in many ways that would not have been considered for the game design otherwise.

There are many patterns included in the game that are focused on the units and their abilities in combat. *Power-Ups* are present in the game as something that can be received from ruins. Without this pattern being included, the ruins would not have been implemented into the game. Among other things, the ruins can give *New Abilities* to the units. This was also something that would not have been included in the game without this pattern being chosen. New Abilities can also be received from the units being promoted into heroes, which is a form of *Character Development. Improved Abilities* is present in the game in the form of units gaining better statistical attributes when they level up. The *Paper-Rock-Scissors* pattern comes to play in combat with the element system. This was a huge part of the game design and picking this pattern affected the direction of the game development. It inspired the idea of the fantasy setting

for the game, as well as the idea of adding magic to the game in the form of elements that the units use to power their attacks. These patterns were all a big part of the game design, and they also add to the *Surprises* pattern.

Table 4.2: Research Patterns chosen for the study.

Game Design Pattern	Pattern Category	Pattern Sub-category	Used?
Tools	Game Elements	Objects	No
Pick-Ups	Game Elements	Objects	No
Power-Ups	Game Elements	Objects	Yes
New Abilities	Actions and Events	Action Control	Yes
Improved Abilities	Actions and Events	Action Control	Yes
Ability Losses	Actions and Events	Action Control	No
Decreased Abilities	Actions and Events	Action Control	No
Ultra-Powerful Events	Actions and Events	Events	Yes
Character Development	Narrative Structures, Predictability and Immersion	Narrative Structures	Yes
Surprises	Narrative Structures, Predictability and Immersion	Narrative Structures	Yes
Betrayal	Social Interaction	Competition	Yes
Alliances	Social Interaction	Group Activities	Yes
Randomness	Game Mastery and Balancing	Planning	Yes
Paper-Rock-Scissors	Game Mastery and Balancing	Balancing	Yes

There are also patterns that affect the gameplay without player input. *Surprises* occur in the game. The *Ultra-Powerful Events* provide surprises to the players in the form of random events that occur, which the players cannot affect. There are also other factors that add to this, with the biggest one being *Randomness.* The Randomness pattern is present in many aspects of the game, including damage output during combat, the random chance for a unit to be promoted

into a hero, and which Power-Up a unit receives when investigating a ruin. It is also used in generating the game map procedurally and spawning the units at the start of the game to random locations. Without Randomness being chosen, the map would not have been made this way.

Another big aspect of the game design was the diplomacy system. *Betrayal* and *Alliances* are patterns concerned with social interaction between players. With the diplomacy system, players can form alliances with each other, but there is nothing in place to prevent players from breaking those alliances. For example, this allows for players to betray each other by declaring war on an ally.

The other patterns that ended up being included were not pursued as something that needed to be included like the Research Patterns were. Despite of this, many of them ended up as part of the game design. Almost all the patterns that were categorized as Mandatory Patterns were included in the game. This was to be expected, as the patterns classified as mandatory were patterns that appear in most turn-based strategy games.

Surprisingly, many of the Desirable Patterns also ended up being included in the game in some ways. Some of them were less prominent than others, and some could be argued were not fully implemented. One of these patterns was *Negotiation*, which normally would have some kind of negotiation system included with it (Björk & Holopainen, 2005), but in Fantasy Mission, the negotiation part is basically just a “yes or no” question on whether a player wants to form an alliance or end a war with another player who has offered it.

4.2 Excluded Design Patterns

Twenty-four out of eighty of the patterns chosen for the study were excluded from the final build of the game. Four of them were Research Patterns, nineteen were Desirable Patterns and one of them was classified as a Mandatory Pattern (see Table 4.1). The Desirable Patterns were never intended to be included in the game, so it is not surprising so many of them were left out of the game.

Out of the fourteen chosen Research Patterns, four of them ended up being excluded from the game during development. The ones that were left out were Ability Losses, Decreased Abilities, Pick-Ups and Tools. The reasons for each of these exclusions are similar. The first reason was

a time constraint. The game needed to be finished before this study could be completed, so some planned features had to be cut.

Pick-Ups and Tools did not add anything new over the Power-Ups pattern that had already been implemented. Pick-Ups, Tools and Power-Ups are all in the subcategory of Objects in the Game Elements category. Since these patterns did not affect the game design significantly over what had already been implemented, the decision was made to cut them.

Ability Losses and Decreased Abilities all had similar features as New Abilities and Improved Abilities, which were included in the game. All of them are in the Action Control subcategory in the category of Actions and Events. It would not have been too difficult to implement these, but they did not add anything crucial to the design of the game at the state that it was at the time.

There was one pattern that was initially classified as mandatory that ended up not being in the game. This pattern was *Agents*, which is a pattern describing computer-controlled game entities that act like players (Björk & Holopainen, 2005). In other turn-based games like *Civilization,* there are factions that are controlled by the game system, which do similar actions that the players do. However, for Fantasy Mission, there ended up being no computer-controlled factions due to time constraints.

5 Analysis

In this chapter, the results of the study are analysed. The research question is answered, weighting the pros and cons of how the game design patterns affect game design. Other factors that affect game design are also considered, as well as the study pattern categorization.

5.1 How Game Design Patterns Affect Game Development

There are many ways that the chosen design patterns affected the final game design. The results of this study show that choosing specific game design patterns to use when making a game do affect the design of the game. There are both pros and cons with this.

5.1.1 Pros

The Research Patterns were applied at the very beginning of the game design process, and they shaped the game mechanics of the game. This made the design process smoother in a lot of ways, as there was a framework to work with from the start, with no need to come up with game mechanics from scratch. This shows that using game design patterns as blueprints for the game design works well in the design phase of game development.

Another good reasons to use game design patterns is solving problems with the game design at the design phase, before going into the game testing phase. The design patterns come with descriptions on how they affect other patterns and potential problems that can come with them, which can help identify and solve these problems before they manifest.

5.1.2 Cons

The game design patterns also restricted the design process in some ways, as some of the patterns felt like they were added to the game for the sake of using the design pattern instead of a real need to have the mechanic in the game in the first place. That is also the reason why some of the Research Patterns were excluded during the development process, as there was no real need to include them in the game. As such, when making a real game meant to be published, if the game design patterns are used for the design of the game, it would be a good idea to consider whether the pattern is needed or not at different points during the development process.

Another potential issue with using game design patterns as blueprints for the game design is that there is a certain lack of innovation involved in the design process. While Fantasy Mission was intended to be a unique game with its own distinctive features, it ended up lacking in this department. A part of this is the restrictions that the study placed on the game design, as the use of the game design patterns was put above everything else in a priority list. The use of cards that affect the combat system in the game was a feature that was intended to bring the most unique flavour to the game, but this ended up being cut in favour of implementing the game design patterns that were chosen as Research Patterns.

5.2 Other Factors

The game design patterns were not the only thing that affected the game design. The game genre was another big thing that shaped the design of the game. The decision to make a turn-based strategy game already framed the way the game was going to look and play at the game concept phase. This means that even with the patterns that were chosen to be applied, there were a lot of other influences that contributed to the final design.

Another contributing factor was market research. By looking into turn-based strategy games that already existed prior to this study, games like *Civilization* and *Age of Wonders* were looked at. These games also shaped the final design just for the fact that they are mechanically and aesthetically like the game that was created for this study. This was both a conscious and unconscious decision, as some mechanics were based on these games, while other similarities came in the process of creating the game without necessarily meaning to do so. One of these design choices was adding ruins to the game, which is something that is also in *Civilization* games. The decision to add the ruins came from the use of the *Power-Ups* design pattern, as there needed to be a way for the units to find the Power-Ups in a way that fit the game.

Time constraints were also a big contributor to some design choices, as features needed to be cut from the final design simply for the lack of time to implement everything. In a real game development process, these features could be added in after release as either a free update or as a part of a DLC later down the line in the post-production phase. One of these choices was the decision to cut some of the design patterns that did not seem as important as others.

The developer's personal preferences also contributed to the design, as some features from games like *Civilization* were left out due to this reason. One of these was the technology tree and the different historical eras that the game has, where technology advances as time passes, which allows for new inventions to be made to advance the player's civilization.

5.3 Categorization

The initial categorization of the game design patterns into Mandatory, Desirable and Research Patterns was not without its problems. This was not the main point of the study, but as it was used so heavily in the framing the study, it should be mentioned that it had some issues.

The differences between Mandatory and Desirable Patterns were not very straightforward. Many of the Desirable Patterns ended up in the game without the intention to do so. This also left some patterns in the Desirable Patterns to be patterns that were not necessarily actually desirable and instead were placed in that category due to the reason that they needed to be placed in one of two categories. An example of this is having *Single-Player Games* and *Multiplayer Games* as patterns that were looked at for this study, and both were placed in the Desirable Patterns. These patterns do not seem like something that fits that category explicitly, instead being more of something that is fundamental to the game design itself and fits all games that are either single-player or multiplayer respectively.

There are some ways the categorization could have been done better. One better way to do the categorization could have been to have one category for patterns that are used in most games in general, and another category that is more genre-specific to turn-based strategy games. The Research Patterns could then have been patterns that are mostly outside of this scope to demonstrate better how much the patterns affect the final design of the game itself, with all the turn-based strategy game patterns being excluded from that list. Another option would have been to simply have one category that is specific to turn-based strategy games, and the other being the Research Patterns. There was no real need for Desirable Patterns in the study, but it was a way to narrow the bigger list down to patterns that could be used for the research. The categorization could have also been done to a larger set of patterns to get a better scope for it all.

6 Discussion and Future Work

In this chapter, there is discussion and potential future research that could be done with game design patterns based on the results and what was left out of the study. Some of these are outside of the scope of the study but are related to game design patterns.

As explained in the analysis in this thesis, the genre of the game affects the game design to a degree. It could be argued that it affects the game design even more than the chosen game design patterns, as the chosen Research Patterns were chosen with the genre in mind also. With that in mind, it could be interesting to study how much the genre of the game affects the set of game design patterns used in that game. It could be interesting to see if the patterns differ greatly, or if there are similarities between games of different genres. Another variation of this could be trying to deliberately use the same game design patterns to create multiple games of different genres to see how similar or different they turn out to be.

One way to take this topic further would be to do a study comparing two games of the same genre, with game design patterns being consciously chosen for one of them, while the other game would be created without the use of game design patterns. The games could then be compared to see how similar the two games turn out to be, and an analysis could be conducted for the game that did not use game design patterns consciously, to see which game design patterns are included in the game design anyway.

Software design patterns usually include sample code and a graphical representation of the structure of the architecture of the pattern, which illustrates interaction, sequences of requests and collaborations between objects (Gamma, Helm, Johnson, & Vlissides, 1994). Game design patterns are more abstract in that sense, as they do not go into the technical side of things. Something that could be done with game design patterns is something similar, perhaps creating sample code for how they could be implemented with popular game engines like Unity and Unreal Engine, or graphical illustrations on how these patterns would function with each other in such a system. Some game design patterns are probably unsuitable for this as they are too abstract, but it could be interesting to see how some game design patterns could be implemented this way.

Another thing that could be researched for the future is the use of artificial intelligence (AI) with game design patterns. Nowadays, AI is being used in many fields, and the use of AI will likely only increase in the future. As such, it would be interesting to see how it could be used with game design as well. Examples of the use of AI with game design patterns could be in either identifying game design patterns or choosing patterns for designing a game of any given genre. It could also be used to study how well an AI can create a game design or the full game all the way through from a set of game design patterns given to it.

Another potential future work related to this comes from what Neves and Zagalo (2021) touched on, with the fact that there is no real unified methodology for game design and that lot of the game design tools are used in pedagogy, while not necessarily in the game industry itself. Therefore, another related study to game design patterns that could be conducted could be research into the game design tools in use in the game industry itself, which could help determine how game designers practice game design nowadays, especially in comparison to the tools used in pedagogy. This could give insight into the state of game design in the game industry, and what steps could be taken to potentially create a standardized game design methodology.

7 Conclusion

The study made for this thesis asks the question: “How does using game design patterns affect game development?” This is answered by designing and developing a game with the game design patterns as a key part of the game design.

The game development process starts with the *concept* phase, which includes the game idea, target audience and marketing. There are usually many ideas to choose from, so the ideas can be looked at through Schell’s (2019) eight filters to help see if the idea is something that works. These filters are applied at different points of the game development process. The game concept phase ends with the creation of the *concept document* that lists everything about the concept to explain what the game and its purposes are.

The *game design* starts with the concept phase, but the *pre-production* phase is where it is refined. This where documents like the project proposal, art style guide, production plan, the GDD and *technical design document* are created. Some of the biggest part of game design are the *gameplay*, *game balancing*, *story* and *characters*. *Level design* and *interface design* are their separate categories, but also a part of the overall game design. The *game engine* is also an important part of the design. The GDD is the most important document for game design, and it can be thought of as a blueprint for the game. The GDD’s purpose is to be a reference guide throughout the development process, and it contains every element of the game. It can change daily to reflect any changes as the project develops.

Game design patterns are a collection of possible design choices to be used in games. They were created to establish a common language for game design, but they failed to do so at a large scale, but they are still used in pedagogy in game design exercises. A game design pattern consists of a name, core definition, general description, uses of the pattern, consequences of using the pattern and relations between patterns. Björk and Holpainen’s *Patterns in Game Design* (2005) book’s list of patterns was used as the main source for game design patterns for this study.

Prototypes are created following the pre-production phase, and they are used to test gameplay mechanics. The prototypes are a part of an *iterative* development process, in which after the prototype is created, the game goes to an *evaluation* phase where feedback from playtests is

considered. After evaluation, the game goes back to the design phase, where a new prototype is created. This can go on until the game is finished. Other parts of game development include the *production*, *pre-launch*, *launch* and *post-production* phases. These are not relevant for this study, but they are important parts of the game development process.

The game that was developed for the study is called Fantasy Mission. It is a turn-based strategy game. It was developed as a separate project over the course of the year 2025. A set of *game design patterns* was chosen out of Björk and Holopainen's (2005) list of patterns. These patterns were used as a basis for the game design that was created and documented in a *game design document* (GDD). The development process followed the game development process from the concept phase until the prototype phase, and it did not go to the production phase or beyond. Three prototypes of the game were created to showcase different game mechanics, with the third one being the final build that has all of it. A development diary was kept throughout the development to document the process.

The initial set of patterns chosen for further examination for the study was categorized into three categories: Mandatory Patterns, Desirable Patterns and Research Patterns. The Mandatory Patterns are patterns that were perceived as mandatory for a game of the chosen genre. Desirable Patterns are patterns that are not necessary for the game to be created, but something that players of the genre could want. The Research Patterns are the set of chosen patterns that were intended to be added to the game design of the created game to see how they affect the development.

The results of the study show that the game design patterns do affect the game design and the game development process. Throughout the development, some intended features were cut from the final game, including some intended game design patterns. However, most of the chosen patterns were included. Some reasons for excluding patterns were time constraints and similarity of some of the patterns that were chosen. Many features in the game that was designed came directly because of choosing the game design patterns, and they defined the overall game mechanics of the game. The categorization into Mandatory, Desirable and Research Patterns has some flaws, and it could be simplified. In conclusion, the game design patterns do affect the overall game design and the development process, and they should be researched further for more applications in game design.

References

Adams, E. (2014). *Fundamentals of Game Design (3rd Edition).* New Riders.

Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., & Angel, S. (1977). *A Pattern Language: Towns, Buildings, Construction.* Oxford University Press.

Argüello, D. (2025). *Crusader Kings 3 team says addressing feedback backlogs is like 'playing darts'.* Retrieved from Game Developer (9.3.2026): https://www.gamedeveloper.com/design/crusader-kings-3-team-says-addressing-feedback-backlogs-is-like-playing-darts-

Barney, C. (2021a). Application of Pattern Language for Game Design in Pedagogy. *Information*, 12 (10): 393.

Barney, C. (2021b). *Pattern Language for Game Design.* CRC Press.

Björk, S., & Holopainen, J. (2005). *Patterns in Game Design.* Charles River Media.

Björk, S., & Holopainen, J. (2006). Games and Design Patterns. In K. Salen, & E. Zimmerman (Eds.), *The Game Design Reader: A Rules of Play Anthology.* The MIT Press.

Björk, S., Holopainen, J., & Kreimeier, B. (2003). Game Design Patterns. *Lecture Notes for GDC.* Game Developers Conference.

Björk, S., Holopainen, J., & Lundgren, S. (2003). Game Design Patterns. *Proceedings of DiGRA 2003 Conference: Level Up.* DiGRA.

Bramble, R. (2023). *The Seven Stages of Game Development.* Retrieved from GameMaker (9.3.2026): https://gamemaker.io/en/blog/stages-of-game-development

Brownell, C. (n.d.). *How Games are Made: 4 Stages of Video Game Development Process.* Retrieved from Game Design Skills (9.3.2026): https://gamedesignskills.com/game-development/stages-of-game-development-process/#h.w4lib4uhceo1

Dor, S. (2018). Strategy in Games or Strategy Games: Dictionary and Encyclopaedic Definitions for Game Studies. *Game Studies*, 18 (1): 43-55.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley.

Kiselev, M. (2024). *6 steps to a successful playtesting process for an indie developer.* Retrieved from Game Developer (9.3.2026): https://www.gamedeveloper.com/programming/6-steps-to-a-successful-playtesting-process-for-an-indie-developer

Koster, R. (2005). A grammar of gameplay: game atoms: can games be diagrammed. *Presentation at the GDC.* Game Developers Conference.

Mostovoy, A. (2024). *Balancing creativity and deadlines in game development.* Retrieved from Game Developer (9.3.2026): https://www.gamedeveloper.com/production/balancing-creativity-and-deadlines-in-game-development

Neil, K. (2016). *How we design games now and why.* Retrieved from Game Developer (9.3.2026): https://www.gamedeveloper.com/design/how-we-design-games-now-and-why

Neves, P. P., & Zagalo, N. (2021). Game Design as an Autonomous Research Subject. *Information*, 12 (9): 367.

Novak, J. (2012). *Game Development Essentials: An Introduction (3rd Edition).* Delmar/Cengage Learning.

Orland, K. (2022). *Physical console games are quickly becoming a relatively niche market.* Retrieved from Ars Technica (9.3.2026): https://arstechnica.com/gaming/2022/02/fewer-and-fewer-console-games-are-seeing-a-physical-release/

Paschali, M.-E., Volioti, C., Ampatzoglou, A., Gkagkas, A., Stamelos, I., & Chatzigeorgiou, A. (2021). Implementing game requirements using design patterns. *Journal of Software: Evolution and Process*, 33 (12): e2399.

Przybylski, R. (2021). *How Randomness Contributes to Strategic Thinking.* Retrieved from Game Developer (9.3.2026): https://www.gamedeveloper.com/design/how-randomness-contributes-to-strategic-thinking

Riendeau, D. (2023). *How To: Write a Game Design Document.* Retrieved from Game Developer (9.3.2026): https://www.gamedeveloper.com/design/how-to-write-a-game-design-document

Schell, J. (2019). *The Art of Game Design: A Book of Lenses (3rd Edition).* CRC Press.

Sintoris, C. (2015). Extracting game design patterns from game design workshops. *International Journal of Intelligent Engineering Informatics*, 3 (2-3): 120–143.

Ubisoft. (2021). *Games Explained: From Game Engines to Going Gold.* Retrieved from Ubisoft News (9.3.2026): https://news.ubisoft.com/en-us/article/P8R9SX3LGlQrlXuTuwAfQ/games-explained-from-game-engines-to-going-gold

Appendices

There are two appendices in this thesis. The game design patterns table is meant to showcase the multiple game design patterns that were considered for the game design. The game design document is the design document that was created during the development of the Fantasy Mission game.

Appendix 1. Game Design Patterns Table

This table consists of the 166 patterns that are listed in the book *Patterns in Game Design* by Björk and Holopainen (2005), which were considered for the study conducted in this thesis. It is not a full list of all patterns that exists. The patterns that were not considered further are not categorized into a study category. The additional patterns that were only available on the book's accompanying CD are not listed.

Game Design Pattern	Pattern Category	Pattern Sub-category	Study Category	Used?
Game World	Game Elements	Game Worlds	Mandatory Pattern	Yes
Reconfigurable Game World	Game Elements	Game Worlds	-	-
Levels	Game Elements	Game Worlds	-	-
Inaccessible Areas	Game Elements	Game Worlds	-	-
Consistent Reality Logic	Game Elements	Game Worlds	-	-
Alternative Reality	Game Elements	Game Worlds	-	-
Moveable Tiles	Game Elements	Game Worlds	-	-
Enemies	Game Elements	Objects	Mandatory Pattern	Yes
Boss Monsters	Game Elements	Objects	Desirable Pattern	No
Deadly Traps	Game Elements	Objects	-	-
Obstacles	Game Elements	Objects	Desirable Pattern	No
Avatars	Game Elements	Objects	Desirable Pattern	No
Units	Game Elements	Objects	Mandatory Pattern	Yes

Game Design Pattern	Pattern Category	Pattern Sub-category	Study Category	Used?
Tools	Game Elements	Objects	Research Pattern	No
Controllers	Game Elements	Objects	-	-
Alarms	Game Elements	Objects	Desirable Pattern	Yes
Pick-Ups	Game Elements	Objects	Research Pattern	No
Power-Ups	Game Elements	Objects	Research Pattern	Yes
Clues	Game Elements	Objects	-	-
Extra-Game Information	Game Elements	Objects	Desirable Pattern	No
Score	Game Elements	Abstract Objects	Desirable Pattern	No
High Score Lists	Game Elements	Abstract Objects	-	-
Lives	Game Elements	Abstract Objects	Mandatory Pattern	Yes
Strategic Locations	Game Elements	Locations	Desirable Pattern	No
Outstanding Features	Game Elements	Locations	-	-
Chargers	Game Elements	Locations	Desirable Pattern	Yes
Resources	Resource and Resource Management	Types of Resources	Mandatory Pattern	Yes
Producer-Consumer	Resource and Resource Management	Resource Control	Mandatory Pattern	Yes
Ownership	Resource and Resource Management	Resource Control	Mandatory Pattern	Yes
Resource Management	Resource and Resource Management	Resource Control	Mandatory Pattern	Yes
Investments	Resource and Resource Management	Progress	Mandatory Pattern	Yes
Diminishing Returns	Resource and Resource Management	Progress	-	-
Imperfect Information	Information, Communication and Presentation	Information Quality	-	-
Perfect Information	Information, Communication and Presentation	Information Quality	-	-
Uncertainty of Information	Information, Communication and Presentation	Information Quality	-	-

Game Design Pattern	Pattern Category	Pattern Sub-category	Study Category	Used?
Symmetric Information	Information, Communication and Presentation	Information Distribution	-	-
Asymmetric Information	Information, Communication and Presentation	Information Distribution	-	-
Public Information	Information, Communication and Presentation	Information Distribution	-	-
Communication Channels	Information, Communication and Presentation	Information Access	-	-
Game State Overview	Information, Communication and Presentation	Information Presentation	Desirable Pattern	Yes
Combat	Actions and Events	Actions	Mandatory Pattern	Yes
Movement	Actions and Events	Actions	Mandatory Pattern	Yes
Maneuvering	Actions and Events	Actions	-	-
Aim & Shoot	Actions and Events	Actions	-	-
Construction	Actions and Events	Actions	Desirable Pattern	No
Privileged Abilities	Actions and Events	Action Control	-	-
Asymmetric Abilities	Actions and Events	Action Control	-	-
Limited Set of Actions	Actions and Events	Action Control	Desirable Pattern	Yes
Downtime	Actions and Events	Action Control	-	-
Experimenting	Actions and Events	Action Control	-	-
Transfer of Control	Actions and Events	Action Control	Desirable Pattern	Yes
Interruptible Actions	Actions and Events	Action Control	-	-
Focus Loci	Actions and Events	Action Control	Mandatory Pattern	Yes
New Abilities	Actions and Events	Action Control	Research Pattern	Yes
Improved Abilities	Actions and Events	Action Control	Research Pattern	Yes
Ability Losses	Actions and Events	Action Control	Research Pattern	No
Decreased Abilities	Actions and Events	Action Control	Research Pattern	No
Extended Actions	Actions and Events	Action Control	-	-
Irreversible Actions	Actions and Events	Action Control	-	-

Game Design Pattern	Pattern Category	Pattern Sub-category	Study Category	Used?
Save-Load Cycles	Actions and Events	Action Control	Desirable Pattern	No
Rewards	Actions and Events	Rewards and Penalties	Mandatory Pattern	Yes
Penalties	Actions and Events	Rewards and Penalties	-	-
Illusionary Rewards	Actions and Events	Rewards and Penalties	-	-
Ultra-Powerful Events	Actions and Events	Events	Research Pattern	Yes
Delayed Effects	Narrative Structures, Predictability and Immersion	Evaluation	-	-
Hovering Closures	Narrative Structures, Predictability and Immersion	Evaluation	-	-
Illusion of Influence	Narrative Structures, Predictability and Immersion	Evaluation	-	-
Perceived Chance to Succeed	Narrative Structures, Predictability and Immersion	Evaluation	-	-
Immersion	Narrative Structures, Predictability and Immersion	Immersion	-	-
Anticipation	Narrative Structures, Predictability and Immersion	Immersion	-	-
Freedom of Choice	Narrative Structures, Predictability and Immersion	Creative Control	Mandatory Pattern	Yes
Creative Control	Narrative Structures, Predictability and Immersion	Creative Control	Mandatory Pattern	Yes
Storytelling	Narrative Structures, Predictability and Immersion	Creative Control	-	-
Narrative Structures	Narrative Structures, Predictability and Immersion	Narrative Structures	Desirable Pattern	No
Tension	Narrative Structures, Predictability and Immersion	Narrative Structures	-	-
Characters	Narrative Structures, Predictability and Immersion	Narrative Structures	Mandatory Pattern	Yes

Game Design Pattern	Pattern Category	Pattern Sub-category	Study Category	Used?
Character Development	Narrative Structures, Predictability and Immersion	Narrative Structures	Research Pattern	Yes
Planned Character Development	Narrative Structures, Predictability and Immersion	Narrative Structures	Desirable Pattern	No
Identification	Narrative Structures, Predictability and Immersion	Narrative Structures	-	-
Higher-Level Closures as Gameplay Progresses	Narrative Structures, Predictability and Immersion	Narrative Structures	-	-
Surprises	Narrative Structures, Predictability and Immersion	Narrative Structures	Research Patterns	Yes
Cut Scenes	Narrative Structures, Predictability and Immersion	Narrative Structures	-	-
Easter Eggs	Narrative Structures, Predictability and Immersion	Narrative Structures	-	-
Competition	Social Interaction	Competition	Mandatory Pattern	Yes
Conflict	Social Interaction	Competition	Mandatory Pattern	Yes
Player Killing	Social Interaction	Competition	-	-
Betrayal	Social Interaction	Competition	Research Pattern	Yes
Cooperation	Social Interaction	Collaboration	Mandatory Pattern	Yes
Team Play	Social Interaction	Group Activities	-	-
Alliances	Social Interaction	Group Activities	Research Pattern	Yes
Roleplaying	Social Interaction	Group Activities	-	-
Constructive Play	Social Interaction	Group Activities	-	-
Player Decided Results	Social Interaction	Group Activities	-	-
Social Interaction	Social Interaction	Stimulated Social Interaction	Mandatory Pattern	Yes
Trading	Social Interaction	Stimulated Social Interaction	Desirable Pattern	No
Bidding	Social Interaction	Stimulated Social Interaction	-	-
Bluffing	Social Interaction	Stimulated Social Interaction	-	-

Game Design Pattern	Pattern Category	Pattern Sub-category	Study Category	Used?
Negotiation	Social Interaction	Stimulated Social Interaction	Desirable Pattern	Yes
Social Dilemmas	Social Interaction	Stimulated Social Interaction	-	-
Gain Ownership	Goals	Ownership and Overcoming Opposition	Mandatory Pattern	Yes
Overcome	Goals	Ownership and Overcoming Opposition	Mandatory Pattern	Yes
Stealth	Goals	Ownership and Overcoming Opposition	-	-
Eliminate	Goals	Ownership and Overcoming Opposition	Mandatory Pattern	Yes
Rescue	Goals	Ownership and Overcoming Opposition	Desirable Pattern	No
Capture	Goals	Ownership and Overcoming Opposition	Mandatory Pattern	Yes
Evade	Goals	Ownership and Overcoming Opposition	-	-
Conceal	Goals	Ownership and Overcoming Opposition	-	-
Race	Goals	Ownership and Overcoming Opposition	Desirable Pattern	No
Collection	Goals	Goals of Arrangement	-	-
Guard	Goals	Goals of Persistence	-	-
Survive	Goals	Goals of Persistence	Mandatory Pattern	Yes
Traverse	Goals	Goals of Persistence	Desirable Pattern	Yes
Gain Information	Goals	Goals of Information and Knowledge	-	-
Gain Competence	Goals	Goals of Information and Knowledge	-	-
Exploration	Goals	Goals of Information and Knowledge	Desirable Pattern	No
Predefined Goals	Goal Structures	Goal Characteristics	Desirable Pattern	Yes

Game Design Pattern	Pattern Category	Pattern Sub-category	Study Category	Used?
Dynamic Goal Characteristics	Goal Structures	Goal Characteristics	-	-
Optional Goals	Goal Structures	Goal Characteristics	Desirable Pattern	Yes
Interferable Goals	Goal Structures	Goal Characteristics	-	-
Player Defined Goals	Goal Structures	Goal Characteristics	Desirable Pattern	Yes
Preventing Goals	Goal Structures	Relations Between Goals	Desirable Pattern	Yes
Hierarchy of Goals	Goal Structures	Relations Between Goals	-	-
Tournaments	Goal Structures	Relations Between Goals	-	-
Incompatible Goals	Goal Structures	Relations Between Goals	Desirable Pattern	No
Selectable Sets of Goals	Goal Structures	Relations Between Goals	-	-
Supporting Goals	Goal Structures	Relations Between Goals	-	-
Symmetric Goals	Goal Structures	Relations Between Goals and Players	Desirable Pattern	Yes
Asymmetric Goals	Goal Structures	Relations Between Goals and Players	-	-
Committed Goals	Goal Structures	Relations Between Goals and Players	-	-
Real-Time Games	Game Sessions	Game and Play Sessions	-	-
Asynchronous Games	Game Sessions	Game and Play Sessions	-	-
Synchronous Games	Game Sessions	Game and Play Sessions	-	-
Single-Player Games	Game Sessions	Game and Play Sessions	Desirable Pattern	No
Multiplayer Games	Game Sessions	Game and Play Sessions	Desirable Pattern	Yes
Turn-Based Games	Game Sessions	Game and Play Sessions	Mandatory Pattern	Yes
Closure Points	Game Sessions	Game and Play Sessions	-	-
Player Elimination	Game Sessions	Player Activity	Desirable Pattern	Yes
Analysis Paralysis	Game Sessions	Player Activity	-	-
The Show Must Go On	Game Sessions	Player Activity	-	-

Game Design Pattern	Pattern Category	Pattern Sub-category	Study Category	Used?
Agents	Game Sessions	Player Activity	Mandatory Pattern	No
Game Mastery	Game Mastery and Balancing	Game Mastery	Desirable Pattern	Yes
Empowerment	Game Mastery and Balancing	Game Mastery	Mandatory Pattern	Yes
Timing	Game Mastery and Balancing	Game Mastery	-	-
Rhythm-Based Actions	Game Mastery and Balancing	Game Mastery	-	-
Dexterity-Based Actions	Game Mastery and Balancing	Game Mastery	-	-
Memorizing	Game Mastery and Balancing	Game Mastery	-	-
Puzzle Solving	Game Mastery and Balancing	Game Mastery	-	-
Luck	Game Mastery and Balancing	Game Mastery	Desirable Pattern	Yes
Tradeoffs	Game Mastery and Balancing	Planning	Desirable Pattern	No
Randomness	Game Mastery and Balancing	Planning	Research Patterns	Yes
Risk/Reward	Game Mastery and Balancing	Planning	Desirable Pattern	No
Predictable Consequences	Game Mastery and Balancing	Planning	-	-
Limited Planning Ability	Game Mastery and Balancing	Planning	-	-
Strategic Knowledge	Game Mastery and Balancing	Planning	Desirable Pattern	Yes
Stimulated Planning	Game Mastery and Balancing	Planning	Mandatory Pattern	Yes
Balancing Effects	Game Mastery and Balancing	Balancing	-	-
Symmetry	Game Mastery and Balancing	Balancing	-	-
Team Balance	Game Mastery and Balancing	Balancing	-	-
Right Level of Difficulty	Game Mastery and Balancing	Balancing	-	-
Right Level of Complexity	Game Mastery and Balancing	Balancing	-	-
Handicaps	Game Mastery and Balancing	Balancing	-	-

Game Design Pattern	Pattern Category	Pattern Sub-category	Study Category	Used?
Paper-Rock-Scissors	Game Mastery and Balancing	Balancing	Research Pattern	Yes
Meta Games	Meta Games, Replayability and Learning Curves	Meta Games	-	-
Replayability	Meta Games, Replayability and Learning Curves	Replayability and Learning Curves	Desirable Pattern	Yes
Varied Gameplay	Meta Games, Replayability and Learning Curves	Replayability and Learning Curves	Desirable Pattern	No
Smooth Learning Curves	Meta Games, Replayability and Learning Curves	Replayability and Learning Curves	-	-

Appendix 2. Fantasy Mission Game Design Document

Fantasy Mission

Game Design Document (GDD)

Jaakko Uitto

1. Introduction

The following is a Game Design Document (GDD) of the game.

1.1 Game Name

The game is called Fantasy Mission.

1.2 Team

The team is just one person, Jaakko Uitto.

1.3 Date of last update

The last update was on 28.3.2026.

2. Game Overview

2.1 Game Concept

The game idea is that it's a mix of Sid Meier's Civilization games and A Game of Thrones: The Board Game. The players will go around a "board" where there are cities that can be captured and they can create units, as well as heroes that have their own tools, equipment and some special abilities that can be developed further as they gain experience. These abilities manifest in a form of using cards in a battle, which can turn the tide of the battle. There are ultra-powerful events that affect the game world without the players having any way of stopping them. These events act as surprises that may change the tide of the game. The winner is the one who defeats all opposing players.

2.2 Target Audience

The target audience is people who enjoy fantasy and strategy games, but don't want to get into the overly complex grand strategy genre. More specifically, a more general audience of teenagers to adults in the western world. The estimated target rating is either 7 or 12.
The target platform is PC.

2.3 Genre

Turn-based strategy.

2.4 Game Flow Summary

The player moves through the game using a floating camera that can move around the game world. The player can select units and cities to view more information about them, as well as select actions that the player wants the units and cities to do. These actions can be moving the units, creating units, etc.

2.5 Look and feel

The game is in 3D with a top-down view. It is essentially a living board game with medieval fantasy elements. The world looks like a board game laid out on a table. The units and heroes look like living board game pieces and the use of cards for ultra-powerful events and abilities add to that feel.

2.6 Hook

Fantasy Mission has events that affect the gameplay outside of the players' choices. There are hero characters that are stronger than normal units. Rock-paper-scissors style elements in combat.

2.7 Competitive Analysis

Similar games on the market include Sid Meier's Civilization series, Humanity, Endless Legend and the Age of Wonders series.

2.8 Game Engine

Unreal Engine 5. Specifically, version 5.5.4.

3. Gameplay

3.1 Objectives

The biggest objective of the game is to defeat the other players, which happens by taking all their locations and defeating all their units. Other objectives are expanding your territory, upgrading your cities and levelling up your units.

3.2 Game Progression

The game progression happens in two ways. First is building up your cities and upgrading your units and heroes. This happens by your units gaining experience and using resources for the upgrades. This makes your units and heroes stronger, grows your army and makes your cities produce more resources. Second is expanding your territory. By taking over locations from other players and founding new locations, you can expand your territory, which gives you more resources and ability to create more units.

3.3 Play Flow

The game starts with the player having only one Settler. The player will then proceed to explore the area and use the resources gained from the land to expand their territory. When the player comes across an opponent, they can proceed to engage in diplomacy or go to battle with them for their territory. When the player's units go to battle, they gain experience that allows them to gain levels that can allow them to become stronger. As the game goes on, the opponents also grow stronger as they gain experience, which makes the game progressively more difficult. This is all completely up to the player to do as they wish. The estimate for how long the game can take is up to 2 hours, but it can end quickly as well depending on player strategies.

3.4 Challenge Structure

The challenge structures are not linear. Basically, the player must keep producing more and more units and heroes to be able to combat the opposition as the game goes on, as well as found new cities to gain more land to get more resources.

4. Mechanics

4.1 Rules

Each player takes turns to complete their objectives. A player cannot complete actions during other players' turns.

Creating units takes resources. The number of resources needed to create depends on the type of unit. Units can also be upgraded into heroes if they meet certain conditions. If a faction has no money, they cannot produce more units.

Units can move only a certain number of spaces. This number depends on the attribute values of that unit.

During battle, the opposing sides can complete only a certain number of actions based on the unit's action points and the actions the units know. The number is specific to the unit type. The battle is decided by the damage done to the units and heroes, and it occurs in turns just like any other action in the game.

Units and heroes that have magic abilities adhere to a rock-scissors-paper pattern. There are elements that work well or badly against each other based on a chart that is currently unspecified. These elements are Fire, Water, Earth and Wind. Fire beats Wind, Water beats Fire, Earth beats Water and Wind beats Earth. There is also a Neutral element for non-magic based abilities.

Capturing cities happens by completing a siege. During a siege, units and heroes battle against the city's defences until either side loses or the sieging party calls off the siege. When a city's health drops to zero, it's ownership is transferred to the attacking faction.

Ultra-powerful events occur at random points in the game. These events can change the game rules. They give power-ups to different unit types.

The winner of the game is the last faction with units or cities. A player with no units or cities is eliminated from the game.

4.2 Interaction Model

Every object on the board is constrained to moving on tiles. Some tiles are inaccessible. Each tile has its own movement modifier that specifies how many unit actions it takes for a unit to move to it. The player moves using a floating camera that can move over the game board.

Units on a board can interact with other units, as well as cities and ruins. When a unit interacts with other units, a battle ensues. When a unit interacts with a city of another faction, it sieges it. When a unit interacts with a ruin, it receives a power-up that gives it an ability that they previously did not have.

Factions interact with each other through diplomacy. The factions can be in a state of war, alliance, hostile or friendly. If a faction is hostile, it does not allow the opposition to enter its borders. If a faction is friendly, it allows the opposition to enter its borders.

Factions can collect resources from map tiles within its borders. These resources can be used to create new units and heroes.

4.3 Objects

The movable objects in the game for the player are the units. These can be moved by selecting them and then selecting a tile to move them to. Other objects in the world are cities and ruins, and they cannot be moved.

4.4 Actions

Actions that the player can take during a turn are moving units, changing production that happens in a city, communicating with other players through diplomacy, and using unit actions, like attacks, movement and founding new cities.

Diplomatic actions that can be taken are trade, alliance and declaration of war.

4.5 Combat

Combat occurs through battles. The battle is turn-based combat that occurs during the player turns same as any other actions in the game. In battle, the player order units to attack specific units or flee from the battle. Other players and units can also join in an already existing battle, as the battle occurs

on the gameboard where every action happens. The combat ends once its either called off, one side flees or all units on the opposing side are defeated.

4.6 Replaying and Saving

The replayability of the game is high due to the random nature of it. The units spawn to random tiles, and the game world itself is randomized. Players can choose from different factions that exist, which all have different appeal to them.

There are no save mechanics in the game.

4.7 Events

There are several types of ultra-powerful events that can occur randomly for a random duration lasting from 1 turn to 10 turns.

Element Boost gives a 2x damage bonus to attacks of the given element, which can be any of the four main Elements: Fire, Water, Earth and Wind.

Unit Boost gives 2x damage bonus to the units of the specific type, which can be any of the units, except for Settlers.

EXP Boost gives a 2x experience gain to all units and cities.

No Boost gives no additional effects of any kind.

4.8 Power-Ups

There are several types of power-ups the units can receive from the ruins. Attribute Boost gives the unit an extra point to a random stat, with the possibilities being actions per turn, attack, defence and health. New Action gives the unit a new action, which can be either Ranged Attack or Magic Attack. Level Up simply increases the unit’s level by 1. Become Hero makes the unit into a Hero character.

5. Story

5.1 Premise

Take control of a faction and lead them to victory with the aid of your heroes and magic. May the best heroes win!

5.2 Setting

Fantasy medieval setting with magic.

5.3 Backstory

The backstory of the game is that there are multiple factions in this fantasy world that are looking to bring peace to the world. These factions have their own motives and reasons for doing so. There has been a catastrophe that wiped out all civilization, and now the people are rebuilding their world.

5.4 Plot Elements

There are four factions in the game: Earth Nation, Fire Nation, Water Nation and Wind Nation. These all control one element in the game, these being Earth, Fire, Water and Wind. Each nation can produce people with different elements to the major element of their faction, but this is rarer than producing people with the major element.

At the start, the player doesn't necessarily know anything about the world, but as they explore the world, they gain more knowledge about it by discovering ruins and other factions.

Ultra-powerful events cause different modifiers to occur during the game at random points, which can influence the narrative.

5.5 Story Progression

The player starts as one faction and proceeds to make that faction the strongest faction in the world, competing against other factions. Diplomacy with other factions influences the story, which is self-indulgent by the players with their actions.

6. Game World

6.1 General Look and Feel

The world looks like a game board, with the medieval fantasy setting with, mages, warriors, etc. The units are bigger than the cities to help with players seeing them on the map.

6.2 Areas

6.2.1 World Area

The main area is the board itself, where everything occurs. The different map tiles are randomly added to the map, with different areas within the board that have different terrain. These tiles have their own resources and specific movement. It is randomly generated at the start of the game. The area is a 30x30 grid of hexagonal tiles. Cities can be founded on any tile that is not Ocean terrain, and ruins can spawn randomly anywhere that is not an Ocean terrain.

6.2.2 Terrains

Therea are ten different terrain types: plain, hills, mountains, swamp, marsh, glacier, ocean, plateau, desert and tundra. They each have their unique movement modifiers that affects unit movement. Plain and tundra have the modifier of 1. Hills, Marsh, Plateau and Desert have the modifier of 2. Swamp and Glacier have the modifier of 3. Mountains have the modifier of 4. Oceans are inaccessible. Resources are allocated to the tiles randomly based on the terrain type. Food can appear in any tile that is not Mountains, Swamp, Plateau or Desert. Production can appear in any tile that is not Plain, Marsh, Ocean or Tundra. Wood can only appear in Hills, Marsh and Tundra.

6.2.3 Arctic Areas

The arctic areas are located on the northern and southern edges of the of the map. The terrains that can spawn in this area are Glacier and Ocean. The arctic areas are always just one tile wide and there are two of them.

6.2.4 Tundra Areas

The tundra areas are in-between arctic and variety areas. The terrains that can spawn in this area are Tundra, Plain and Mountains. There are two tundra areas, one in the north and one in the south, totalling $1/4^{th}$ of the whole map.

6.2.5 Variety Area

The variety areas span most of the map. There are two of them in-between the tundra areas and the desert area. The terrains that can spawn in this area are Plain, Hills, Swamp, Marsh and Plateau.

6.2.6 Desert Area

The desert is an area with only the Desert terrain, located in the middle of the map. It spans $1/4^{th}$ of the map.

7. Characters

7.1 Settler

7.1.1 Description

Settlers are weak in a fight, but they can create cities.

7.1.2 Statistics

Settlers have 4 actions per turn. Their attack and defence stats are 1. Health is 5. Production cost is 5 and maintenance cost is 1. The experience yield when killed is 10.

7.1.3 Abilities

Settlers can use the Move action like all units. They can use the Attack action, which allows them to attack nearby units. They can also use the Settle action to settle a new city.

7.2 Warrior

7.2.1 Description

Warriors are basic units that fight for the faction they were born in. They can only attack at a short range.

7.2.2 Statistics

Warriors have 5 actions per turn. They have an attack and defence of 4. Their health is 10. Production cost is 3, and maintenance cost is 2. The experience yield when killed is 40.

7.2.3 Abilities

Warriors can use the Move action like all units. They can use the Attack action to attack nearby units.

7.3 Archer

7.3.1 Description

Archers are basic units that fight for the faction they were born in. They can attack long-range as well as short-range.

7.3.2 Statistics

Archers have 5 actions per turn. Their attack and defence stats are 3. Health is 8. Production cost is 3, and maintenance cost is 2. The experience yield when killed is 40.

7.3.3 Abilities

Archers can use the Move action like all units. They can use Attack action to attack nearby units. They can also use Ranged Attack to attack units at long range.

7.4 Scout

7.4.1 Description

Scouts are not necessarily fighters, but they can fight if necessary. They are fast and good at moving across the map.

7.4.2 Statistics

Scouts have 7 actions per turn. Their attack and defence stats are 2. Health is 7. Production cost is 2, and maintenance cost is 1. The experience yield when killed is 20.

7.4.3 Abilities

Scouts can use the Move action like all units. They can also use the Attack action to attack nearby units.

7.5 Mage

7.5.1 Description

Mages are mysterious people who can use the arcane arts known as Magic. Each of them can use one Element to power their attacks. Mages are the strongest units in the game. They can attack long-range with magic attacks.

7.5.2 Statistics

Mages can use 4 actions per turn. Their attack is 6. Defence is 4. Health is 6. Production cost is 4, and maintenance cost is 3. The experience yield when killed is 50.

7.5.3 Abilities

Mages can use the Move action like all units. They can also use Attack action to attack nearby units. They have a unique ability called Magic Attack, which uses the Element assigned to the unit to deal damage according to the opposing unit's Element, using the rock-paper-scissors style system of strengths and weaknesses. Magic Attack is a long-range attack.

7.6 Hero

7.6.1 Description

Heroes are created and born in battle. Everyone in the land aspires to become a hero one day. Every unit can become a hero under right circumstances. They can utilize a strong Hero Attack.

7.6.2 Statistics

Heroes retain their original statistics, as they keep the unit classification they started with, but with a times 2 boost to attack, defence and health.

7.6.3 Abilities

Heroes retain the abilities they had as normal units, but they gain a new ability called Hero Attack, which is a long-ranged attack, and uses the Element assigned to the unit just like Magic Attack.

8. Levels

8.1 Generated Map

8.1.1 Synopsis

Procedurally generated map that the players can move their units around and find cities.

8.1.2 Objectives

The objectives of the level are the same as the goals of the game: capturing opposing locations, building your army and being the last faction left standing.

8.1.3 Level Details

The players spawn on the map with one Settler and they can find a city and continue building their faction up. Combat also occurs here. The units can also find ruins here that give them a power-up.

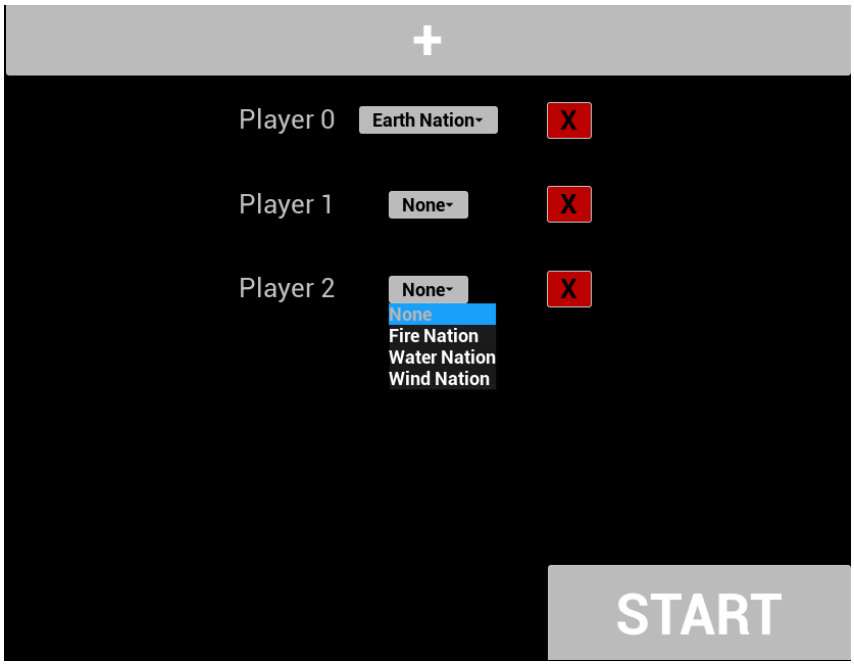
9. Interface

9.1 Visual System

The visual system in use was made with the Unreal Engine widget blueprints.

9.1.1 Faction Selection Screen

The first screen the player sees when the game is opened is the faction selection screen. This is where a new game starts. This screen allows for the player to modify the game that is being started by choosing the number of players in the game, as well as their factions and names.



9.1.2 Game Screen

The game screen is the main screen that the player will use. This screen contains the game world and the game itself. It has a HUD with a resource list, a player list button, a unit list, a button for next turn, a world event button on top of the screen that can display the current ultra-powerful event that is occurring, and a quit button. Each unit, city and ruin has a UI tag above them that displays their health, experience and name.

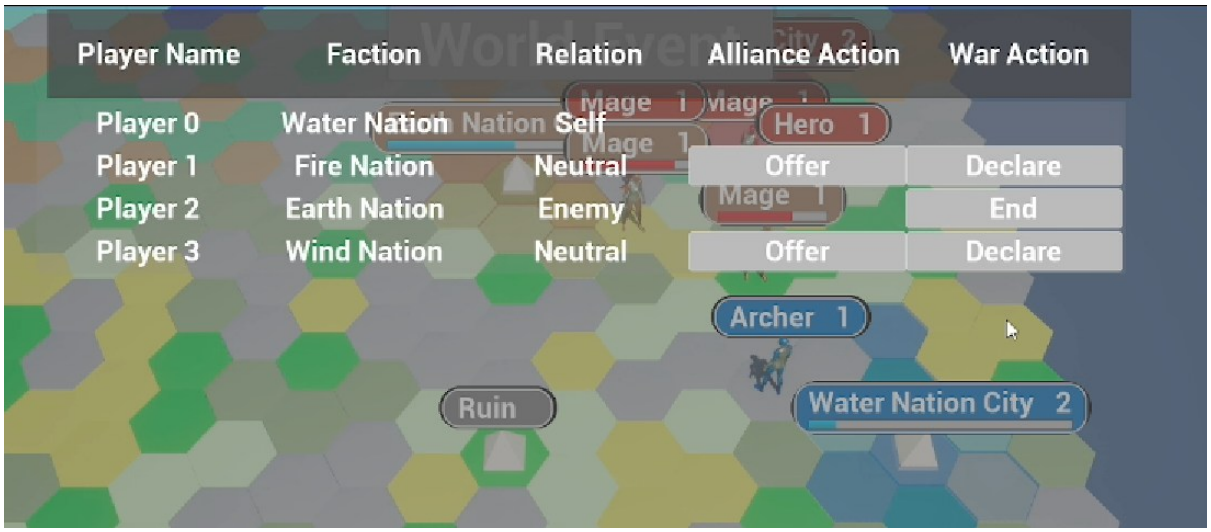
The HUD also contains an action overlay that displays potential actions the selected unit or city can do, as well as an overlay of information on the map tile that is currently hovered. When attacking, the map tiles in range of the attack are highlighted red. When moving, the tiles in the movement path are highlighted blue. Selected and hovered units and locations are highlighted blue.

9.1.3 Turn Start Screen

When the next turn button is pressed, it displays a turn start screen, which simply states the turn-player's display name, with a button to start the next turn.

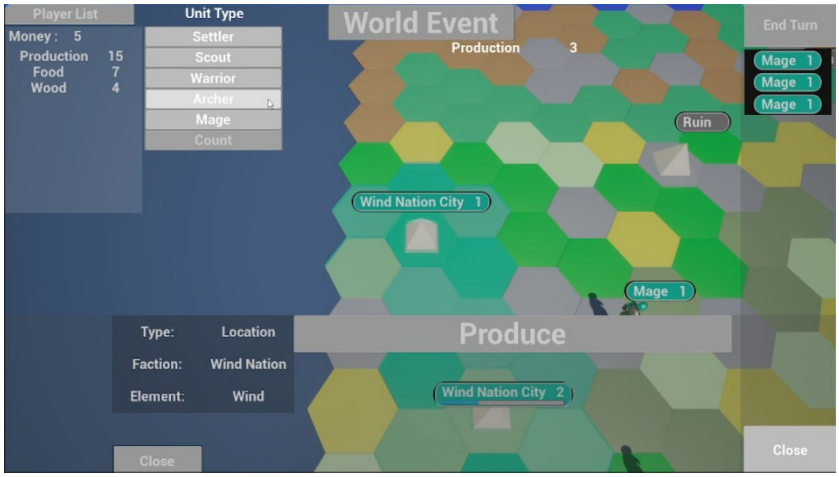
9.1.4 Diplomacy Menu

When the player list button is pressed, it opens the diplomacy menu. This is where the player can interact with other players by declaring a war, offering peace, offering an alliance or dissolving an alliance. It displays each player in the game, with their faction and their relationship with the turn-player.



9.1.5 Production Menu

When an owned city is selected, the production menu can be opened to open the production menu. This shows a list of units that can be produced and the resources needed to produce them. The production action can be cancelled.



9.1.6 Victory Screen

When the game is won, it shows the victory screen, with the name of the player, with buttons to either restart the game or quit the game.

9.1.7 Camera model

The camera in use is a top-down floating camera that can be moved by the turn-player.

9.2 Control System

The camera can be controlled with the right mouse button, WASD or the arrow keys. Units and locations can be selected with left mouse button. If holding CTRL, multiple units and locations can be selected. Buttons can be pressed with the left mouse button or by navigating to them with the arrow keys and pressing ENTER.