OULUN AMMATTIKORKEAKOULU

Johanna Pauna

THEORY AND PRACTICE OF VIDEO GAME DEVELOPMENT

THEORY AND PRACTICE OF VIDEO GAME DEVELOPMENT

Johanna Pauna
Opinnäytetyö
Kevät 2022
Tietotekniikka
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikka, Ohjelmistokehityksen suuntautumisvaihtoehto

Tekijä(t): Johanna Pauna
Opinnäytetyön nimi: Theory and practise of video game development
Työn ohjaaja(t): Anne Keskitalo, Jaana Raudaskoski
Työn valmistumislukukausi ja -vuosi: Kevät 2022
Sivumäärä: 49

Opinnäytetyön tavoite oli tutkia ja esittää videopelin suunnittelun ja kehityksen teoriaa ja käytäntöä. Toteutustapana oli teoriaosuuteen tutkimustyö alan kirjallisuudesta ja käytännön työnä toteutettiin teorian pohjalta pelikonsepti ja siihen liittyvä prototyyppi. Käytetyt aineistot olivat enimmäkseen alan ammattilaisten kirjoittamia kirjoja, kuten Fundamentals of Game Design E. Adamsilta, Advanced Game Design M. Sellersiltä ja Practical Game Design A. Kramarzewskiltä ja E. De Nuccilta. Mukana oli myös paljon löydettyä web-aineistoa, kuten BDC puhe J. Vandenberghelta, ”Applying the 5 Domains of Play: Acting Like Players”.

Videopelin kehityksen teoria on hyvin laaja aihe. Tähän opinnäytetyöhön on kerätty tärkeimmät perusasiat, joista kehitys voi lähteä liikkeelle. Huomiota kiinnitettiin ensin esittelemään teoriaa, miten videopeleistä tehdään viihdyttäviä ja mielenkiintoisia. Työ esittelee pintapuolisesti ymmärrystä pelaajan psyykestä ja mikä saa pelaajan pyhittämään tunteja pelille. Työssä keskitytään myös paljon pelikonseptin luomiseen ja varsinaisen kehitystyön rakenteeseen, kehitystiimin rooleihin ja dynaamikoihin ja kehityksen työkaluihin.

Tämän teorian pohjalta aihe jatkuu pelikonseptin luomisella. Pelikonseptissa otettiin huomioon kaikki teoriaosion tärkeät aiheet. Konseptin lähtökohtana oli, että sen toteuttaa itsenäinen kehittäjä yksin ilman varsinaista rahoitusta. Tämä asetti raamit konseptin kehitykselle ja haastoi pohtimaan, mikä on mahdollista ja mikä ei ja mikä on kannattavaa.

Opinnäytetyön kolmannessa osiossa keskityttiin prototyypin luomiseen. Prototyyppi tehtiin pelikonseptiin kehitetystä mielenkiintoisesta toiminnosta, jossa pelaaja voi luoda ja muokkailla 3D-osista kukkia. Prototyypille asetettiin tavoitteeksi vastata kysymyksiin; voidaanko toiminto toteuttaa ja mikä olisi paras tapa sen toteuttamiseksi? Prototyypin kehityksessä saatiin vastaukset kysymyksiin ja niiden pohjalta toiminnon kehittäminen voisi jatkua.

Videopelikehitys on haastava ja mielenkiintoinen ala, josta löytyy töitä monenlaisille ammattilaisille. Harrastuksenakin se on äärimmäisen hyödyllinen ja antoisa ajanviete, koska sen kautta voi oppia paljon tärkeitä taitoja, niin omiin iloihin kuin ammattiakin ajatellen. Yksinäiselle kehittäjälle kokonaisen videopelin tekeminen voi olla myös avain itsenäiseen yrittämiseen, mikä antaa myös vapaat kädet itsensä ilmaisuun tuotteillaan. Haastavaa ja raskasta se kuitenkin on itsenäiselle tekijälle, eikä peliä ole välttämättä helppoa saada esille valtavan pelikirjon seasta.

Asiasanat: Videopeli, Videopelin kehitys, Pelisuunnittelu

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology, Option of Software Development

---

Author(s): Johanna Pauna
Title of thesis: Theory and Practice of Video Game Design
Supervisor(s): Anne Keskitalo, Jaana Raudaskoski
Term and year when the thesis was submitted: Spring, 2022
Number of pages: 49

The aim of this thesis was to investigate and portray the theory and practice of videogame design. Methods for implementation were to research literature and articles of the profession for the theory part, implement a game concept based on this theory, and then create a prototype of a feature of this concept. Sources used for the thesis were mainly books by the professionals of the field, such as Fundamentals of Game Design by E. Adams, Advanced Game Design by M. Sellers, and Practical Game Design by A. Kramarzewski and E. De Nucci. Sources from the web were also used, such as BDC talk by J. Vandenberghe, “Applying the 5 Domains of Play: Acting Like Players”.

Theory for videogame design is an expansive subject. This thesis portrays the most important basic concepts from which design process can be started. Attention was focused on theory of how to make games entertaining and interesting. Theory on player psyche and what makes a player dedicate hours on a game was investigated. A lot of focus was also given to development of game concept, structure of game development, development team’s roles and team dynamics, and for tools of game development.

Game concept was created next based on the theory part. The basis for the concept was that the game is developed by an independent developer without specific funding. This created frames for the development of the concept and challenged to think about what is possible, what is not, and what is profitable.

On the third segment of the thesis focus was placed on the prototype. Prototype was made of the novel feature of the game concept, where a player can create and customize a flower from 3D-parts. The questions that the prototype was to answer were: can the feature be done and what is the best method? In the prototyping of the feature answers were found and based on them, development of the feature could continue.

Development of video games is a challenging and interesting field that employs many kinds of professionals. Even as a hobby it is an extremely useful and rewarding pastime, as you can learn many important skills through it, as much for one’s own amusement as for career. For independent developer making video games can be a key for becoming entrepreneur. Independent development of games also gives free hands for the developer to express themselves. It is however a challenging and strenuous and it is no easy task to get your game seen from the myriad of games on the market.

Keywords: Videogame, Videogame design, Game concept

INDEX

1 THEORY FOR DESIGNING VIDEO GAMES

One of the earliest known video games, Spacewar!, was created in the early 1960s. First ever video game to hit the markets was a coin operated arcade machine game called Galaxy Game, introduced in 1971 [1].

Video game industry as profitable business started with the arcade machines. First home video game console system called Magnavox Odyssey by Ralph Baer came out in 1972. Arcades bloomed and flourished until 1982. Video game industry in general had a crash at this point, as the market was getting oversaturated with cheap quality products that did not meet the growing expectations of consumers. Crash ended with the release of The Nintendo Entertainment System and following development of games [2].

Since then, video game industry has grown in varied ways. Technology has taken great leaps forward, bringing us from pixel art to beautiful 3D landscapes and graphics and from simple 2-dimensional movement to robust 3-dimensional physics. New machines and devices have been invented, from consoles to phones to virtual reality devices.

As video games grow by complexity and profits, more attention is addressed to the designing and developing games. As the crash from early 1980 has taught, trying to sell bad games do not work. More and more are expected from the games; innovation, complexity, and in general, fun. As Katie Salen and Eric Zimmerman have concisely expressed in their book: “The goal of successful game design is the creation of meaningful play.” [3].

1.1 What is a video game?

To understand how to design a successful video game, a definition of what a video game, and a game in general, is, should first be established. In the next three chapters are explored the definition for a game, definition for a video game and the elements required for a video game.

1.1.1 What is a game?

There are more categories of games than just video games, such as board games or outdoor activity games, like tag. Humanity has certainly created and played games since before recorded history.

Games are played. Ernest Adams describes playing as a participatory form of entertainment, which is nonessential, usually recreational activity. Games are usually socially significant, as they are often played together. Contrary to child's play with dolls and toys, game consists of rules and goals and players will actively participate with those rules to achieve the goals. However, according to Adams's definitions, if the game consists of just one rule that defines one goal, it is a puzzle instead [4].

Another good and concise definition of games comes from Salen and Zimmerman: "A game is a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome." [3]. However, Adams do not quite agree with such definition; according to Adams, games do not need to include competition or conflict, and as such classes creative and cooperative systems as games as well [4].

A game consists of certain key elements:

- Playing
- Pretending
- Rules
- Goals

While playing is the activity of participating in the game, pretending is the mental capacity to be able to do so. Pretending means being able to establish an imaginary reality and play by the rules of this imaginary reality. Such imaginary realities can be created, changed, and abandoned at will. Pretending is also referred to as Magic Circle by theoreticians of play [4].

Rules define how the game is played and how to reach the goal(s) of the game. Goals define the winning, or not losing, conditions of the game. The game does not need to be winnable as such to be a good game. For example, the classic game Space Invaders; players can never win the game as the game is endless. However, players will often keep playing the game, as in such case attempting to do better (and so getting better score) is what creates the fun [4].

1.1.2 Definition of video games

"A videogame is a game which we play thanks to an audiovisual apparatus and which can be based on a story." [5]

A general definition of a video game would be:

- A game, which can be played on a machine or device of some sort.
- Player interacts with the game through an input device, such as controller, or mouse and keyboard.
- The machine has sensory feedback devices, at least one for visual feedback, but can also have audio feedback or haptic feedback (for example controller rumble).

1.1.3 Elements of a video game

Video games are built from software and hardware and are defined by genres. While hardware usually is not a part of the game per se, it is necessary to be able to play the game. More details to each part are discussed next.

Software

Video game itself is the software that communicates with the hardware to create a game experience. Designing and developing video game's software program takes all sorts of software tools. To begin with, general office software tools are needed for creating text documents, graphs, spreadsheets and so forth. Also needed are email, team communication software, work scheduling software and more to be able to work efficiently with the development team and all the external stakeholders tied to the development.

Software for building the actual game requires tools such as:

- game-engines, whether third party or company's own
- graphics tools
- 3D-modelling programs
- animation software
- audio tools

A game engine is the software, which brings together all the assets created on other tools. The engine often already includes components for common game-related tasks, such as rendering, physics, animation, input, and playing audio [6]. It offers reusable components and ability to create new reusable components with the assets you bring into the project. Game engine can be thought of as the stage of the game, with lights and cameras ready.

Hardware

Hardware is the machines and devices that run the video game program. Essentially, it is always some sort of computer. Video games can be generally played with the usual desktop and laptop computers, and in modern times also on phones and tablets. There are also many devices build solely for playing video games, such as consoles, arcade cabinets, and the still rather new addition, virtual reality devices. These devices usually come with added input devices, for example controllers and cameras.

Genres

Genre is a classification for video games to identify what kind of gameplay the game offers. This helps different kinds of players to each find the entertainment they enjoy. Genres have increased and evolved over time, and their specifications sometimes change. Many games are hybrids of multiple genres instead of being purely one, while other games do not fit into any established genres. Games can also have subgenres, which help define more on the general genre, for example different kinds of shooters (battle royale, survival, looter) [4].

Genres do not specify the theme of the game, which can be anything from aliens to cowboys to country life, but rather they describe the kind of gameplay challenges player will encounter, like shooting, platforming or one versus one fighting [4].

Most common genres are:

- Shooter
- Action and arcade
- Platformer
- Fighting games
- Strategy games
- Role-playing games
- Sports games
- Vehicle simulations
- Construction and simulations games
- Adventure games
- Puzzle games [4]

1.2 Fun and entertainment in video games

Video games are mainly produced and sold on the premise of entertainment. As such, it is important to understand how to make them fun and entertaining, and for this reason, understand what exactly it means, when something is fun.

1.2.1 What is fun?

Fun is a difficult concept to define. What is fun for one is not for another. In general, it is thought that fun is whatever actions or experiences that reward the person with feelings of enjoyment. Human body releases certain hormones when brain decides that whatever is happening now, is fun or enjoyable.

Michael Sellers has defined and categorized different aspects of fun in his book Advanced Game Design. Sellers describes fun as a pleasant, enjoyable, emotionally positive experiences, which an individual would generally want to repeat. Balance is important to fun; if the game is too boring or too stressful, player will struggle to find any fun in it. Playing also needs to be voluntary for feelings of fun. If forced to play or if the consequences of playing are too high, most will not find it fun anymore. Feeling negative emotions in a fun activity can also be fun: watching a sad or a scary movie or feeling frustration (in moderation) in a game [7].

Engagement is a measure of how fun the experience is for the player. It is and ongoing cognitive and emotional state, a combination of three aspects:

- Vigour
    - Vigour means having high levels of energy for the game, mental resilience over the challenges and willingness to put effort in to succeed in the game. Overall persistence to keep playing.
- Dedication
    - Dedication is the feelings of enthusiasm, pride and inspiration for the game.
- Absorption
    - Absorption is the state of full concentration, being so deeply engrossed to the game that the player has difficulty detaching from it [7].

1.2.2 How to make video games fun?

To make video games fun, and thus successful products, there are some guidelines and tips to follow. As presented in previous chapter, engaging gameplay that bring out vigour, dedication and absorption to players should be the goal here. Game needs to be interactive, with choices that have real consequences and choices that make sense. Player should be able to make sense of the game

world and have some understanding of the consequences of their actions [7]. Choices should also matter; if one choice trumps all other choices, is there any choice at all? Instead, more choices that are all strong and weak in different situations should be added. It gives player a reason to choose different things and as such meaning to the choices [8].

While players need to be able to somewhat understand the game, not everything needs to be openly presented. Game of discovery and learning can be very rewarding. Game should also offer plenty of other accomplishments, as sense of accomplishments bring feelings of enjoyment and fun. It is also important to balance the rhythm of the game, with sequences of stress and sequences of calm. Too much stress can tire the player out, too much calm can bore them [7].

Cognitive overload should be avoided, if the player struggles just figuring out how to do actions to play the game, they will also struggle to have fun. Bushnell's Law states that game should be "easy to learn and difficult to master". This means designers should aim to make the game and control of the game easy to understand and natural, fluid feeling, but bring depth to the game in other means. Too much repetition should also be avoided, as it will quickly get boring. Having enough different activities to do will break the repeating themes and keep it novel. These activities should also be designed with care and quality however, as tedious, boring, pointless feeling or overwhelming interactions are not fun [7].

1.3 From idea to game concept

Designing a video game starts from working ideas, choosing one and then building upon the idea. After the first raw version of the idea is set for the game, a game concept is created. These are the parts that require to be worked on and researched before any other design work starts, to ensure efficient process for working the project. It is not that different from working any other software; the requirements for the software should be described at least to a degree before any development work begins.

1.3.1 Coming up with ideas

Making a video game starts from an idea. There really are no limits to what the idea can be, any idea can be turned into a video game. Idea can just pop into someone head without prompting, but it is more usual in the video gaming industry to be creating ideas. Often companies license someone else's idea, like creating a game concept of a book or movie [4].

1.3.2 Creating a game concept

Next a game concept will be made of the idea. A game concept is a more detailed plan of what the game is. While it should not go into every little detail yet, you should go over certain points on it:

- High concept statement: two- or three-sentence description of what the game is about.
- The player's role or roles in the game, if any. If player will have an avatar, brief description of it.
- A proposed primary gameplay mode: camera model, interaction model, general types of challenges the player or players will experience in the mode.
- The genre or genres and the features of different genres it will have.
- A description of the target audience.
- The machine or machines the game will be developed for and any special equipment or features the game will use.
- Brief explanation on how the game will make money.
- Any intellectual property the game will use.
- The competition modes game supports, single-or multiplayer competitive or cooperative.
- General summary of the game's progression from start to end.
- A short description of the game world [4].

Player

Understanding the player is important for game design. This is the person the game is designed for and to whom the game is sold to. Of course, not every game can be sold to everyone, as there

are many different types of players. The similar players are grouped in to target audiences, which is discussed later in this chapter.

Jason Vandenberghe has done great deal of study to the player and player types. In his 2013 GDC talk “Applying the 5 Domains of Play: Acting Like Players” he talks of the five domains of play [9], which is based on the big five personality dimensions OCEAN by S. Rothmann and E. P. Coetzer. The big five personality dimensions are:

- Openness to experience
- Conscientiousness
- Extraversion
- Agreeableness
- Neuroticism [10].

Vandenberge’s equivalent terms on these on the perspective of gamer personalities are:

- Openness to experience: Novelty
- Conscientiousness: Challenge
- Extraversion: Stimulation
- Agreeableness: Harmony
- Neuroticism: Threat?

Vandenberge says the neuroticism does not correlate to threat according to studies. Neuroticism reflects the experience (or not experience) of negative emotions. This means that low neurotic players will feel fewer negative emotions, while highly neurotic players can get so wrapped up about something in the game, that they will stop playing even if they are otherwise enjoying the game [9].

FIGURE 1. Extended facets of the four domains [9]

The picture (Figure 1) lists more facets of the four of five domains. Neuroticism is not in here; it should rather be regarded as a scale of how much negative emotions raising content you should include in the game. The next picture (Figure 2) shows an example of a player in this chart. Each player will have a dot in each of the four facets. If they are more of a builder than explorer and love fantasy over realism, the upper left square of the novelty will be coloured, and so on. This gives us up to 256 different categorized player types. This does not mean a player that likes solo games could not also enjoy multiplayer, but it does help focus the design of the game to the right target group, instead of attempting to please everyone and thus displeasing everyone [9].

FIGURE 2. An example of a player type in the previous chart [9]

Understanding these different aspects of player can help a designer construct a game right for the right target group. It helps to develop accurate empathy, which helps designers to overcome their own personal biases. It helps to find what attracts different gamer-personalities, so you can ensure the game is built to really sell with the chosen target groups. The understanding gives designers the ability to create games other than their own personal taste, to really see what the interest is for another kind of player than themselves [9].

Target audience

Target audience is the group of people that the game is designed for. Designing the game should take on a player-centric approach: the player should be understood, as a part of audience and as an individual with emotional connection to your game. It is important that designers do not just follow their intuition in their decisions for the game, as that may be very different from what the audience wants and even inappropriate for the target audience [4].

Women are typically thought not to be much into gaming, but this is not true. Women represent a large portion of gamer market, and they do not differ in their gaming taste nearly as much as it is portrayed. As women's interests in gaming overlap considerably with men's interests, designers should investigate ways to include women, and especially not to exclude women, in their designs [4]. For example, oversexualization of women, male-only characters, portrayal of women only as

fragile side-characters will deter female players. Having characters with varying physiques and different personalities, in both males and females, is a strong case for any game. This, however, does not mean you have to force to fill the game with female characters or change already planned out or important male character to female.

While women as gamers do not differ as much as portrayed, there are differences that a designer should be aware of. As much as both genders can enjoy the same games, they often enjoy different aspects of it and like to play with different attitude. Below is a list of general differences between men and women:

- Learn-by-dying: women like to generally know beforehand what is expected of them, so this approach will generally deter women.
- Different attitudes to risk: Women are less willing to risk, while men are generally more likely to experiment even at risk of losing something.
- Women are more interested in people and socializing: Online games are more successful than single-player games among female gamers.
- Different conflict resolution styles: Women are not opposed to violence, but they generally require justification for it to be interesting to them. Fighting, war and shooter games are more popular with men than women, while role-playing games are popular with women, as the violence is only a part of the game, and it has context.
- Women enjoy mental challenges: puzzle games are very popular among women.
- Customization of avatars: women generally identify with their avatars more than men, who generally treat their avatars as puppets. The avatars are treated as self-expression, and fantasy version of themselves [4].

It is important to remember that these are generalizations. There are also men who enjoy more the socialization, puzzles and dressing up their avatar, as there are also women who enjoy mindless violence and experimenting wildly while risking everything and individuals of both sexes that like both. It is advisable not to make “women’s” games or “men’s” games, but rather build quality games which takes these aspects into consideration [4].

Age is an important aspect to consider in the target audience. Making games for different age groups have different requirements. Both physical and mental abilities are still developing in children, teens and 20–30-year old’s motor skills are at their peak, while mental abilities develop through the age, unless eroded by disease or injury [4]. But physical and mental abilities are not

the only things to consider; content needs to be appropriate for the target audience's age. Swearing, explicit sexual content, violence, scary themes, or death, are all inappropriate for children. They can also be inappropriate for many adults as well.

Age and content ratings help designers to create content appropriate for their audience as much as it helps the audience to find a suitable game. Game will get reviewed for rating prior to being published. After the review, the game company can either publish the game or change the content of the game and have it reviewed again. PEGI ratings are primarily used in Europe and Asia, while ESRB is used in the US [11].

There are many more things to consider regarding the target audience. The larger your target audience by age, sex, country, culture or gaming type, the greater research needs to go in what type of content, themes, genres, and challenges can be added to the game. Be aware, that different countries and cultures are very different in how they play and what they deem appropriate. It may be in the interest of the company to not make one version of the game that fits all, but rather multiple versions that fit for the different regional and cultural needs.

Competition

It can be difficult to find out what your target audience wants from a game and what they do not want. Research into competition can help in this matter. With game post-mortem (research into a game after it has been published and been in the market for a while) research on the competitor's games information about dos and don'ts can be discovered without blindly creating and publishing a game first [8]. Important things to research through competition:

- State of the genre on the market: rising, falling?
- How much is the target audience willing to spend on a game?
- What business model does the target audience prefer: single purchase, subscription, premium items etc.
- How much should be invested in the creation of the game to keep it profitable?

Such research will help to build a game tailored for the wants of the audience, while still maintaining the budget to keep the game profitable. Or the research will reveal that it is not profitable to make a game on the genre for the chosen target audience, and wasting money, time and resources can be avoided altogether.

Mechanics

Mechanics, also referred to as core mechanics, are the ideas, rules and algorithms that define the game. As such, they implement and enforce the rules of the game and generate the gameplay. The five common mechanic types are:

- Physics: how objects influence each other on 2D or 3D space.
- Internal economies: generation and spending of resources, like money or building materials.
- Progression mechanics: story progression, level progression, triggers to events.
- Tactical manoeuvring: takes place in open / semi-open spaces. Advantages and disadvantages of positions and units, for example different movement of chess pieces.
- Social interaction: rules about creating and breaking alliances, the nature of teamplay [4].

Mechanics operate all the pieces of the game, but one mechanic as itself does not create the whole game experience. It is the interactions of the mechanics that bring the experience and depth, and it is important to research and document how these should work together without becoming too complex to play. A successful game offers great amount of depth with minimal amount of complexity.

Budget and profit-making

Unless the company consists of indie-developer(s), who work on the game on their own time with free tools while they have another source of income, making a game is going to require funding before it is selling in the market. Existing companies often already have enough capital to fund the creation of a new game. Loans are another usual source of funding for companies, especially ones just starting out. They can be a risky funding for a new company however, if the game does not end up profitable or does not even get finished, the company is still obliged to pay the loans back. Publishers offer a steady funding to games and make physical retail of game copies possible, but they often only accept submissions from known sources, and they also hold veto power over the game, meaning they can make decisions on the direction of the game development [4].

Many indie companies have turned to crowdfunding. Crowdfunding means company asks consumers to pay for the game in advance. This means there can be funding throughout the development,

turning the profits to funding. It however requires a good marketing campaign to get people onboard. Many can feel hesitant to be paying for promises that might not be kept. Crowdfunding is a form of pre-order: once the game is finished, consumers that participated in the crowdfunding then get the game for no extra charge [4].

Early access model is similar to crowdfunding. As with crowdfunding, company is selling the game before release, however, it is sold in an alpha or beta phase, where players already receive the product as is and get a chance to start playing early. This means these early players also work as a public testing group. The pro of this is the free public testing, from which developers receive feedback on bugs, issues, and suggestions from the target audience themselves. The con is the promise the company has now made with the early access. If the company does not finish the game and releases it in broken state, consumers will start to distrust the company. Early access, which is available for example in Steam (digital game store platform), has reviews of players and discussion boards showing for everyone.

Making profit out of the games comes in many ways in modern times. The old standard of direct payment model with one time purchase still stands. It is still a solid choice for games that are released and not developed further (apart from some after release maintenance of found bugs and issues) [4]. If company wants to deliver the game in parts, such as a heavily story-based game, they can use episodic delivery instead. This means the company sells standalone episodes of the game, which the consumers pay separately. This is not the same thing as game sequels. As one episode is a smaller part of the whole game, it is cheaper and easier to sell. The drawback of episodic delivery is that all core mechanics really must be in place from the first episode [4].

Subscriptions are another form of ongoing profit-making. This is usually used in multiplayer online games, where constant upkeep of servers and continuous development require constant funding. It is more a service rather than a single entertainment experience and it requires a working internet connection and access to the game's servers [4].

Extra profit of a game or game series can be made through sequels, expansions, downloadable extra content and premium items and subscriptions. For example, mobile games are often free to play, but contain in-game purchase options and premium subscriptions to unlock the game further. Sales of game soundtracks and artwork can also produce a small extra profit.

After all these discussions about profit, comes the consideration of physical versus digital game sales. Selling of games started with physical copies before internet gaming became a thing. Now with rise of the internet digital gaming has risen as well, and retail of physical copies has declined, at least for PC. Sales of physical copies rise in Christmas sales, as consumers like to give something physical as presents. While big companies can easily fund the production and distribution of physical copies for their games, smaller companies or independent developers mostly turn towards the digital market, as the costs of physical retail is too heavy on them. It also requires a publisher or distributer [4].

Digital distribution has made way for these smaller companies or independent developers to be able to bring their games to a wide audience. Publishers are no longer necessary for easy access to the market, no costs involved to creating physical copies, which might not all sell resulting in losses. While Adams claims in his book written back in 2014 that online customers often would not pay 50 dollars for a digital copy of a video game [4], this no longer seems to be the case. Digital sales have increased over the years, even for the pricier games, so much that console makers are starting to release digital only versions of their consoles (PlayStation 5, Xbox Series X). With the worldwide spread of covid-19 and resulting social distancing and quarantines, digital purchasing has risen further, as consumer are unable or reluctant to go buy physical copies [12].

1.4 How games are made

Making games involves many considerations, resources, and stages. In this chapter is investigated first what sort of work roles and teams are involved in the design and development of video games. Stages of the design process are looked through next, moving to the game loops, and then prototyping and playtesting principles.

1.4.1 Work roles and team dynamics

Video game industry's job titles and responsibilities are not standardized among companies. While big companies may have multiple designated developers for each part of the design, smaller teams have less specialized roles, and their designers have multiple responsibilities depending on their skills and abilities. That said, some roles have evolved to a certain degree of standard [4].

Lead designer is the person who oversee the overall design. They are the head of the team and the spokesperson for the project to both inside and outside the company. They are less on the hands-on roles of development and more in the role of authority, ensuring the work is getting done. A video game project has only one lead designer [4].

A general game designer is a non-specialized title for a jack-of-all-trades developer. They work more towards defining the gameplay and developing general level design. In small teams everyone is usually a general designer [4].

Mechanics designer defines and documents the core mechanics of the game. They build the mathematical model of the video game, whether it is physics or other rules. They conduct background research on the theme of the video game and assemble data, such as information and statistics about sports for a sports game [4].

Level designers, or world builders, use the components created by other designers to build individual levels for the project. They require some skills in 3D modelling and programming for the building of the game world. User interface designers on the other hand deal with the user interfaces. They design layouts of the screen in various gameplay modes and define functions for input devices [4].

Writers write the backstories, cut-scenes stories, dialogues, and any other textual contents of the game. They do not write technical documents, however. Art director manages the production of visual assets, whether it would be 3D models, textures, sprites animations or user interface elements. They are usually on the same level of authority as the lead designer. Audio director manages the audio side, from music to sound effects, ambient sounds to dialogue and narration [4].

The lead programmer oversees the coding team. They are responsible foe the technical design of the game and the quality of the software. Project manager or producer has the overall responsibility for the game as a commercial product. They work together with marketing, public relations and community managers to help build hype for the game before the release. Producer has a lot of say for the creative side of the game and they can veto on content being added or removed. They track the day-to-day progress of the project, to ensure everything is on schedule [4].

1.4.2 Stages of design process

Stages of video game design can be divided into three parts: concept stage, elaboration stage and tuning stage. The whole development is also divided into two parts that include these stages, pre-production, and full production. Design work starts in preproduction, in concept stage. In concept stage, designers plan and choose the concept of the game, the target audience they are trying to sell the game to, the genre of the game, budget and profit-making of the game and all other considerations that are described in the chapter 1.3.2 Creating a game concept [4].

While still in preproduction, after concept has been locked down, which means that the decisions made in the concept stage will not be changed anymore for the project, development moves to the elaboration stage. In this stage most of the development work is done. Development team will refine the decisions of the design, create prototypes and tech demos, and start working on what has been decided in the concept. At certain point of elaboration stage the game will be usually greenlighted to full production by a funding agency, such as publisher [4].

In tuning stage, no more features will be added anymore. This is sometimes called feature lock. This stage happens either in planned way, when designers have decided the game has all the features it should, or because the project schedule is running out of time to be adding any more features in. In tuning stage, most of the work with be doing small adjustments and polishing and balancing the game to as close to perfection as time and resources allow [4].

1.4.3 Game loops

Michael Sellers, a game designer and professor, defined game loops in his book Advanced Game Design: A Systems Approach. According to his definition, game is made of looping structures. Action by a player generates an effect, which leads the player to take another action with another effect, and so on, creating a looping experience. There are four principal loops:

- game's model loop
- interactive loop
- designer's loop
- player's mental loop [8].

Game's model loop represents the dynamic world of the game. This loop includes all the systems of the game that player can interact with. Stories, level progression, battling with enemies, all of it is included in the loop. This loop is designed by the concept and plan of the game [8].

Interactive loops determine the actions a player can take. To these actions, it then offers feedback in response. Interactive loops consist of core loop and extended loops. Core loop is the main theme and mechanics of the game. As an example, in a role-playing game the core loop is to fight enemies, get experience points and rewards to get stronger, to then fight stronger enemies. Extended loops flesh out the game: in the same role-playing game example, on top of just the main core loop, you would also have crafting, gathering, card games, interactable non-playable characters and many other things [8].

Designer's loop is the perfect model of what the game and the player experience should be. Designer's loop is the design process. Iterating on developing and testing the game concept creates the looping structure. Player's mental loop however may differ quite a bit from what the designer's loop intended. Player creates their own loop based on the game's model loop. They set their goals and execute actions, which gives them feedback and they may alter or set new goals. But player's mental loop may not match with the game's model loop, as player may set their own unexpected goals and practise unintended activities in the game. This is often not a bad result, as it means player is engaged to the game in their own way [8].

1.4.4 Prototyping the game

"A prototype is a model built to prove a concept." [8] Prototype does not apply to just video game design; it applies to all design work. Architects build prototypes of their buildings with cardboard and other craft materials, same as boardgame designers build prototypes of their games with the same materials. Prototyping is done from the designing process before final development work all the way through to the later stages of development. It is done to find out answers whether certain part of the game works. The goal is to find answers, not to be building the game. It is also important to stay open to the results; if assumption was that something works, and prototype shows it does not, time should not be wasted to try and bend it to work [8].

There are two types of prototypes: physical prototype, often called paper prototype, and digital prototype, which is done by software, usually in game engine. Both types of prototyping are useful,

and often both should be used, for different parts but also sometimes for the same parts of prototyping. In both cases, the prototypes should be kept bare and raw, free from artwork and other special features. Anything unnecessary that either takes too long to create for the prototype or confuses from the issue being prototyped, should not be added [8].

Physical ”Paper” prototype

Paper prototype is usually created with paper, cardboard, and other craft materials. It is an easy form of prototype for anyone to create, as it does not require technical profession, although being handy in crafts helps. Cheap and easy to create, to try and iterate, it often is the first step to prototyping. It can be changed and altered immediately, and it is easy to abstract [8].

All cannot be prototyped with a paper prototype for video games. Problems come from representing complex and real-time scenarios. Some things are impossible to prototype with paper, such as gameplay physics. For video games, the idea should not be to only paper prototype everything, but instead use paper prototype to create cheap and fast prototyping, then move to the digital prototyping [8].

Digital prototype

Digital prototyping is a necessary step for video games. It is a time-consuming process, but it saves time in the long run, as designers find out what works and what does not before they dedicate time to true development of those features. It is important to keep the prototype to the scope of the framed question, instead of starting to develop it towards full game. As prototype code tends to be messy and hacky, it should not be used in production; instead, results of prototyping should be documented, prototype code scrapped and in production the code should be written from scratch. It is easier then to connect the prototyped part to other later developed code [8].

1.4.5 Playtesting the game

Playtesting is the process of exposing the current build and design of your game to your target audience in effort to get feedback on how to improve the game. This helps in finding issues, flaws and bugs in the design and the code and offers feedback to what is good and what is not and what is lacking [2].

Playtesting is more than just letting people play your game and tell what they happen to think. There are different methods to do this in a more constructed manner, which helps developers pin down issues on different parts of the game. For who tests, what they test and how, depends on the different parts of the design [2].

Certain simpler things, such as character movement, can be tested internally within the development team and the game company. Anything with, for example, more novel mechanics, or story related, should be tested by members of the target audience. For what to test, ideally all parts of the game should be tested to some degree. Depending on budget for testing however, focus should be directed to the more complex or important aspects of the game [2].

Methods to use in playtesting depends on what needs to be tested. Giving a certain point of the game to test and then having tester fill out a questionnaire is one way of testing. Doing a small marketing campaign towards your target audience can be used to test art direction and brand.
"Big innovation requires early validation!" Playtesting should not be something you put the game through after the game is made. Different parts, especially more original, complex, and risky designs should be tested early in the design process [2].

Individual testing

The test is performed by individual testers separately and the testing sessions are closely monitored and often recorded. These tests are good for finding specific usability issues. You can interview players and get their opinions on the game without them being affected by peers.
Frequency of individual testing is more important than scale. It is better to do testing with 2-3 individuals every couple of weeks than doing tests with a group of 10 or more once a month. If more than one tester comes across an issue, you have an obvious problem to fix.
Individual testing can be time consuming, as each tester should be monitored by someone. Accuracy of results also depend on how closely the tester represents your target audience [2].

Group testing

Group testing can be more affordable by time than individual testing, but testers of the group can influence the opinions of each other and thus dilute the feedback. However, group testing is the

only way to truly tests multiplayer components of games. With different groups you can collect data on how different target audiences receive the game [2].

Unsupervised testing

Unsupervised testing means that the game is provided to the tester, and they will do the testing at their own leisure within testing timeframe. Results are returned via spoken or written feedback, questionnaires, and analytics. These sessions are easy and cheap to run, but quality and quantity of answers are lower [2].

Public testing

Public testing usually consists of alpha and beta phase testing. Testing can be closed, which means participants are limited and invitation only, or open where anyone willing can join the testing. To make public testing efficient, you should operate channels of communications, for example:

- Open a forum for the game for players to discuss the game
- Provide questionnaires to the testers
- Collect analytic data from the game being played
- Built-in feedback system
- Live stream questions and answers [2].

Alpha and beta public tests often now take in the form of early access. Early access means customers will buy the game for full or discounted price to be able to play early. While the potential amount of data from public testing can be large, especially with open public testing, some testers may just want to play the game and not bother themselves to answering any questions.

1.5 Game design documents and helpful tools

In this chapter is listed some common game design document types to help with researching and documenting a game project, from getting ideas to refining ideas to scheduling work and so on. Included are also some otherwise helpful tools. Many of these can be done with just some pen and paper, however there are useful software tools to making and maintaining these easier. Word, Visio, and Excel are good tools from Microsoft Office package for creating text documents, drawing diagrams, and creating spreadsheets.

1.5.1 Brainstorming

Brainstorming is a good tool for coming up with ideas. Basic principle is to come up with lot of ideas without concerning on the quality of the ideas. It is mainly done with a group, but it is possible to get some use out of it to throw one's ideas indiscriminately on paper. Focus should be kept to quantity, not quality at this point. Weeding out lesser potential ideas comes later. It is important to withhold criticism on the given ideas, as especially in groups, this could discourage people from participating with their ideas. Unusual ideas should be welcomed, this is how you invent new and novel things. Once a lot of ideas are thrown in, it is time to start combining and improving on these ideas.

1.5.2 Mind map

Mind maps can be used in different scenarios to help visualize how things are connected. Items and concepts are linked to a central issue in the middle of the map. It can be used to also weigh different options on an issue. For example, mind maps can be used to character planning: character's name is in the middle of the map, then lines go out to describe the character, present the history of the character, show the future plans of the character in the story of the game and so on. Picture shows a generic example of what it can look like (Figure 3).

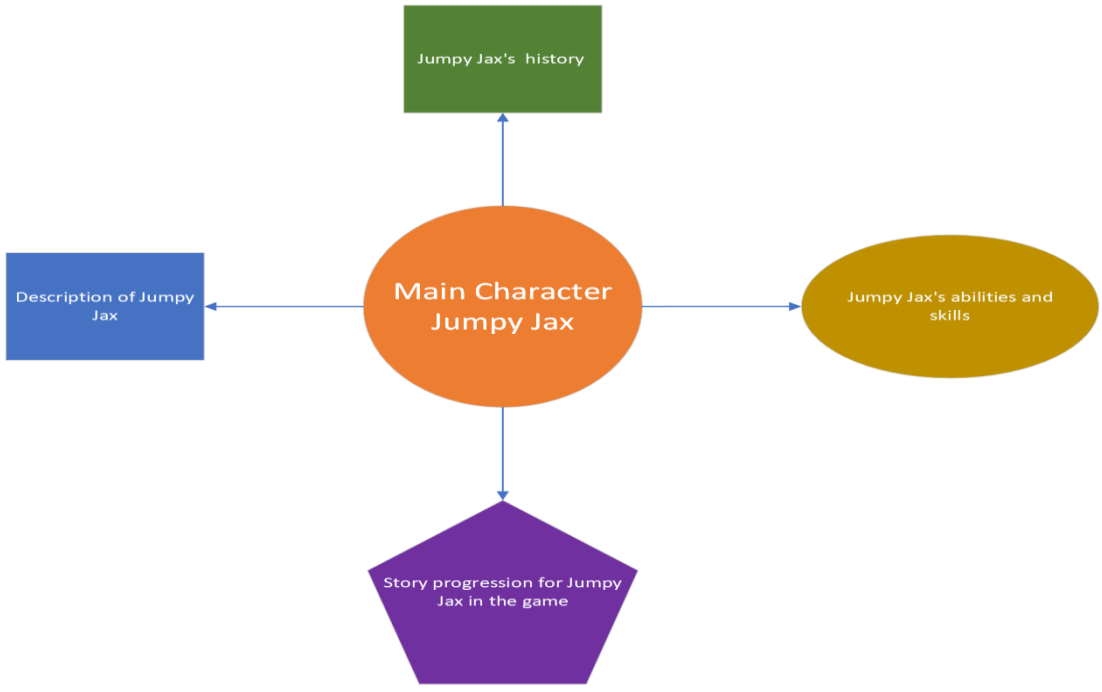

FIGURE 3. Drawn on Microsoft Visio by the author

1.5.3 Flowchart

Flowcharts present each step of a process in the order they happen. This helps visualize what things need to happen to get to a certain point. Flowchart is also useful for depicting branching paths and looping structures. This can be utilized in video game design in many ways: it can be used to show processes in the game on general level such as chart to show the general progression of the game, chart to show the order of triggers in a level, chart to show the steps of a crafting process. It can also be used to help visualize more technical sides of the game design, for example when to play each animation for a character (Figure 4).

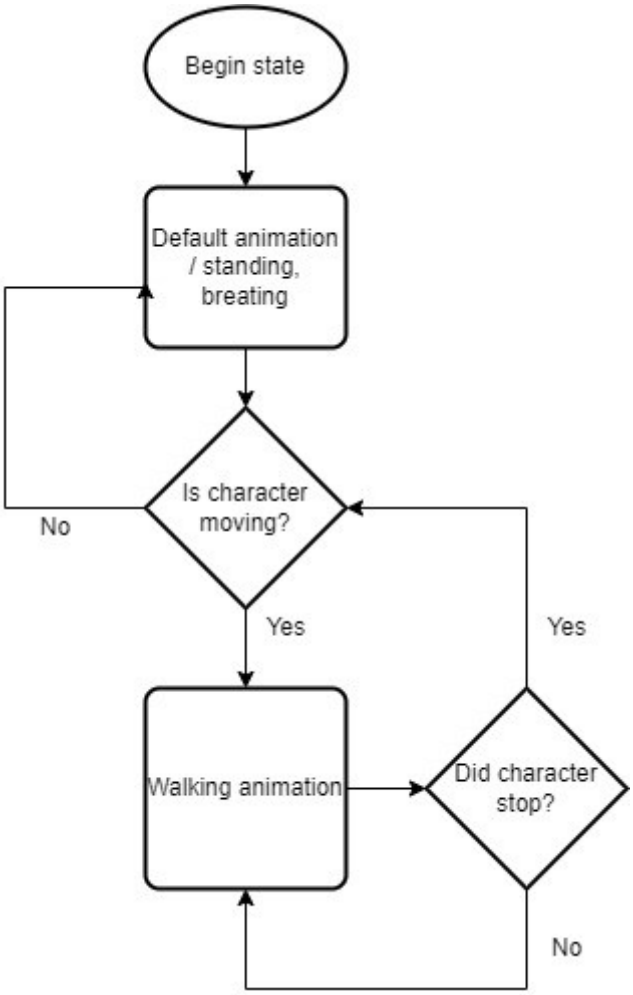

FIGURE 4. A simple flowchart for playing character animation, drawn with draw.io by the author

Flowcharts are drawn with symbols, that are each connected to other symbols that happen either before or after them. Different symbols are used to present different concepts and relationships. Most common symbols are shown in the chart (Figure 5).

Symbol	Symbol Name	Purpose
	Start/Stop	Used at the beginning and end of the algorithm to show start and end of the program.
	Process	Indicates processes like mathematical operations.
	Input/ Output	Used for denoting program inputs and outputs.
	Decision	Stands for decision statements in a program, where answer is usually Yes or No.
	Arrow	Shows relationships between different shapes.
	On-page Connector	Connects two or more parts of a flowchart, which are on the same page.
	Off-page Connector	Connects two parts of a flowchart which are spread over different pages.

FIGURE 5. Chart of common flowchart symbols [13]

1.5.4 Storyboard

Storyboards can be used to document and plan stories and cutscenes. Storyboards are formed with boxes in a sequential order of the events, each representing a shot of the story. This helps visualize how items and characters will be set up in the scene [14].

1.5.5 Spreadsheets

A spreadsheet is a file, that has rows and columns. These are used to store, organize, arrange, and sort data efficiently. Spreadsheets can also calculate numeric data. Spreadsheet programs are

very powerful tools, as they can calculate many kinds of formulas, build graphs from data and much more [15]. These tools are useful in many sectors of office work and can be very useful for listing game elements with their properties. It allows for easy editing of data for balancing games.

1.5.6 Jesse Schell's deck of lenses

Jesse Schell has created a "Deck of lenses" to accompany his book The Art of Game Design, 3rd Edition. These lenses are a small set of questions to ask about the game in development. They will help to examine the design and see flaws, issues, and further development potentials in it. These lenses have been produced into cards and the deck is available as a free smartphone app for iOS or android or as physical deck, which can be bought from Amazon online store [16].

1.5.7 Kanban boards

Kanban board is an agile project management tool, which helps visualize work and project progress. A board consists of columns and cards. The columns represent workflow, each depicting a specific activity. These can be as simple as "to do", "in progress", and "complete". Cards represent the work items, a unit of work to do [17]. With a Kanban board in development team's use, it is easy to follow the progress of the project and share work with the developers.

2 CREATING GAME DESIGN PLAN

In this chapter an example game design plan is created. Plan is created from the perspective of single indie developer. Some office programs will be used, such as Word and Visio. In chapter 3, a prototype will be created of a feature from this design.

2.1 Game ideas, brainstorming

As a single designer and developer, brainstorming is not quite as efficient as when done in group, as there will only be one person's ideas. However, it is still useful for coming up with ideas. Here are the results of brainstorming:

- Flower shop: managing flower shop and garden with feature to create new flowers through gene splitting. Flower creation happens in a game mode, where player gets to choose different parts, colours, and patterns for the flower, with each of these customizable. A casual creative management game.
- Wilderness survival and village builder: start as single player-controlled character on a plot of wilderness, gathering and crafting to build homestead. Later stage comes with hiring villagers to live and work on the land. Casual simulation game with survival elements.
- Nature spirit: spirit growing back nature in barren land. Growing plants, luring in and caring for wild animals, and terraforming the land. Simulation game.
- Doll house: a cute doll-like family simulator. 3D game with locked view from the front, where the front wall is removed to view in, like with a doll house. Casual children's simulation game.
- Creature taming with turn-based strategy: tame creatures and battle them in multi-creature battles. Combat taking inspiration from Heroes of Might and Magic games. A story rich creature catching game.
- Fantasy animal zoo: zoo with a fantasy twist. Management and simulation game.
- Nature reserve: build and manage a nature reserve in old garbage tip. Clean the barren land from garbage. Recycle, invent, and build contraptions out of garbage. Casual management with time management elements.

2.2 Choosing a game idea and creating a game concept

From the ideas of the brainstorming-session, flower shop game was chosen. Reason for choice is that the scope of the game is probably smallest from the options. As an indie developer, it is better to start small. Next a game concept is created from the idea. Definitions from chapter 1.3.2 (Creating a game concept) are used.

2.2.1 High concept statement

A high concept statement, as was defined in chapter 1.3.2, is a two- or three sentence description of what the game is about. This description should be treated as the selling point of the game. There already exists flower shop games, so why should this one be invested in?

Here is the high concept statement for this example game:
A beautiful, calming flower shop game, where players can relax with managing a flower shop and its garden. Players get to express their creativity by creating and customizing their own flowers, bouquets, avatar, the shop, and the garden.

2.2.2 The player's role

Next is description of the players role or roles in the game if there are any. If player will have an avatar, it is briefly described here. The player's role in this example is the flower shop keeper. Player controls the avatar of the shop keeper to run the shop and care for the garden. The avatar is a 3D character, and the player can customize the avatar at any point of the game. Customization includes face and body features, gender choice, and clothes and accessories.

2.2.3 A proposed primary gameplay mode

Here is described the primary gameplay mode. Camera and interaction models are described in general terms. Also, the types of challenges the player or players will experience in the mode, are listed here. In the example, there are two primary gameplay modes: the garden mode and the shop mode. Two was decided instead of just one, as these are both equally important and expansive

modes. The modes share the same camera and interaction models and some other functionalities as well.

Camera model is a top view with an angle. Camera is moved independently from the avatar with mouse and keyboard controls, while fixed to the avatar with controller controls. It can be rotated around the scene with both control schemes and panned with mouse and keyboard controls. There is a limited zoom in and out function, to keep sight of the beautiful scene of the shop and stop player losing control of their camera. In zoom in, camera dynamically tilts slightly upwards, and in zoom out back downwards, to keep a good view and giving interesting perspectives to player. Otherwise, the camera cannot be tilted.

Interaction model is omnipresent, which means that the player can act on different parts of the game at a time or avatar-based, meaning players interact with the game through the avatar [4]. Model depends on whether the player is using a mouse and keyboard or a controller. Player can select different items to bring up menus for details and orders. Avatar movement is controlled either through mouse commands to move, “wasd”-keys or arrow keys on keyboard, or with a controller’s joystick or directional pad. Avatar actions are controlled through interactions with items.

Garden mode has challenges related to creating and maintaining a garden. This involves planting and caring for flowers and plants, and building and expanding garden area. Items can be rearranged to allow players to experiment with different garden layouts. From garden mode player can also access the flower creation mode. Different seasons, weather, bugs, and plant requirements creates challenge to the gardening.

In shop mode player takes care of stocking the shop with flowers to sell and helping customers. In the backroom of the shop is storage and bouquet-making station, where player will craft bouquets for sale. Later in gameplay player can unlock a greeting card making station in the backroom, which they can use to create greeting cards with flowers, bouquets, and various backgrounds and text. These greeting cards would then be printable as pdf (Portable Document Format) or other kind of documents, which can be printed out or sent as digital cards outside the game.

Player has freedom to arrange tables, shelves, and other shop items as they like in the store space. The customers come in varieties: there will be regular customers just buying from the shelves, there will be customers who need help deciding what they want, and customers who come in to order

something specific. Shop is also visited by other non-player characters, that will have items and seeds to sell for the player.

2.2.4 The genre

In genre section the genre, or genres, of the game is explained. If just some features of different genres are present, they will be described as well. For the example game, the main genre is casual, as the game is easy and relaxed, and while there are some number of challenges to plant caring and business, none of them are game ending or difficult. The subgenres are management, as the player manages a flower selling business, and simulation, as the game simulates flowers growing, blooming, and withering.

2.2.5 A description of the target audience

Target audience is casual gamers and mainly girls and women. Game is designed to be suitable for PEGI 7 rating, and as such includes only very mild forms of violence, no scary content, no swearing, and no sexual content of any kind. Only form of violence there may be designed to the game is repelling bugs and it would be depicted as driving them away, instead of scenes of dead bugs. The visual of bugs would be cartoony and adorable, so they would not frighten children. The game could be designed also for PEGI 3 rating, however there is some textual information and concepts too complicated for such young audience.

For nationality of the target audience, the game is designed more the western populace in mind. Setting of the game is an imaginary rural tourist town in Europe, where all the seasons are present. As such the style of buildings, character's mannerism and UI design is designed towards western audiences. Game is designed for English-speaking audience first and foremost. However, later the game can also be translated to few other more commonly used western languages. Designing the game to cater for audiences in other parts of the world would be outside the scope and funding of this game.

2.2.6 The platforms the game is developed for

Here is listed the machines that the game is developed for and any special equipment or features the game uses. The example game would first be developed for computers. Controls would be developed for mouse and keyboard and for controller. After release on pc, the game would be pushed to release also on the popular consoles: PlayStation 4 and PlayStation 5 by Sony, Xbox Series X and Xbox Series S by Microsoft, and Nintendo Switch. The game will not use any special features. Controller can be used on computers and is required for consoles.

2.2.7 Brief explanation on how the game would make money

The example game would be a simple direct payment of one-time purchase. There would not be downloadable content or expansions. Unless the game turns very popular, it would have no sequels either. It would be priced as a mid-sized indie game, around 20-30 euros. Tools used for making the game are freeware or as in Unity 3D's case, freeware until certain profit-margin. There would be no employee costs as it is independent single person workforce, whom has a day job.

2.2.8 Any intellectual property the game uses

The example game would not use any outside intellectual property. Scope and funding of the game cannot afford it. There is also no need for such, as this game will have interesting novel features as selling point, instead of marketing it with other intellectual property, such as movie.

2.2.9 The competition modes game supports

In this chapter is explained the competition modes the example game supports, whether it is single- or multiplayer competitive or cooperative. As a casual game about managing a flower shop and garden, there is no need for competitive mode of any kind. While a casual game could benefit from a cooperative multiplayer mode, it does not fit into the scope of this indie game.

2.2.10 General summary of the game's progression from start to end

Player starts in the early spring season with a small shop that has a small garden behind it. At the beginning player would have limited funds and only couple basic items in the shop and the garden to start working with it. Player first must plant seeds and then care for them for few days before they can collect the first self-grown plants. In the meantime, they could buy ready flowers from a wholesaler non-playable character for sale in the shop. Flower types from wholesaler will be limited and profits from them lesser because of the cost, encouraging the player to grow their own plants.

As time goes, seasons will change. Each season comes with its own weather patterns and temperatures. Different holidays also come with seasons, creating demand for different kinds of plant products. Player will be given information and tips about seasons and upcoming holidays, so they can prepare their gardens accordingly. In winter season the temperatures go to freezing low and plants will wilt outside. Player can however build greenhouses in their garden. There will also be other products, such as light lamps, humidifiers, and heaters to adjust the rooms perfect for different plants' requirements.

Later in the game, the player will have earned enough money to not just buy new items to their shop and garden, but also upgrade their sizes. As both the garden and shop increase in size and chores, player can also start hiring non-playable characters as employees. This adds to business management, while decreases hecticness with chores. A changing lifecycle of the game will keep it fresher instead of turning it too repetitive.

When player grows new plants, they unlock genomes for the plants. These genomes add plant part items, colours, and patterns to the flower creation feature. In the flower creation station, player can choose from different petal types, stalks, and leaf types and customize each part by shape, colour, and pattern. Once the flower is complete, the player receives seeds for the flower to be grown in the garden. The genomes will also determine the requirements of the new custom plant, which are visible to the player by inspecting the seeds or plants.

The game includes a minor story theme. Player's avatar is a young adult, who has studied botany and bought a small shop and included garden with the help of a loan from their parents. The story's theme is about appreciation, as in its core, gifting flowers is about showing appreciation. Some customers will have their own little storylines of wishing to show appreciation to various close ones

for various reasons. The avatar themselves will want to show appreciation to their parents for their support and love, and this story progresses as the player pays back the loan. There is no timeline for paying back the money, this is entirely up to the player. While themes of get-well-soon stories of someone injured or ill will be in the game, any themes of serious illness or injury and death are excluded to keep the game light, happy, and appropriate for all ages.

2.2.11 A short description of the game world

The game world is set in a beautiful, sleepy, rural tourist town. The shop itself is set in the edge of the centre of the town. Player can view the front of the shop in a mode of its own, see what their shopfront looks like (this is also customizable by the player) and watch as people of all walks go by, window-shop, and come in and out. The style of the game will be dreamy, bloomy, and cartoony with 3D graphics.

3 PROTOTYPING THE GAME PLAN

The prototype is made of the flower creation feature, as this is the novel selling point of the game. The basic workings of the feature are:

- selecting parts of the flower:
    - models for petals
    - leaf-type
    - sepal (parts of the flower that encloses the developing bud) type [18]
    - stalk-type
    - pistil (the ovule producing part of the flower) and stamen (pollen producing part of the flower) type [18]
- customizing the parts:
    - width
    - length
    - thickness
    - size
    - bend horizontally
    - bend vertically
    - placement
- selecting patterns for each part; each petal can be selected separately, while leaves, stalks, sepals, pistil, and stamens have each one pattern style
- customizing patterns:
    - width
    - length
- selecting colours of each pattern

3.1 Tools

Tools used for the creation of the prototype are Unity 3D, Blender, and Gimp. All these tools are free to use. Prototype is a digital prototype, as it is not possible to prototype the physics of 3D model shaping or pattern customizing by paper prototype. Blender is used to create 3D model

templates for all the flower parts, Gimp is used for drawing sprite art, and Unity 3D is used to build it all together with code.

3.2 Concept of the feature

Concept art to roughly sketch out the scene of the feature was drawn with pen and paper (Figure 6). As the drawing shows, the feature scene is divided into four parts:

- Model window:
    - View of the flower model, player can rotate, pan, and zoom view
    - Player can drag parts in and out, move parts around, and select parts
- Customization menu:
    - When player selects a part, it's properties and customization options are show in this menu
    - Player can use the customization sliders to customize the shape of the part and modify colours of chosen pattern through colour picker (show when player clicks a colour block)
- Part library:
    - Library of all unlocked parts
    - Grouped by type: stalks, leaves, petals, sepals, and pistil and stamen types
    - Expandable / retractable drawers: click the type opens the drawer if closed, or closes if opened, arrow after title represents the state
    - Scrollable library: as all the items cannot fit in view at once, player can scroll the library left or right to view items.
- Pattern library:
    - Lists all unlocked patterns as blocks with the pattern
    - Clicking a pattern will select that pattern for use for the selected part.

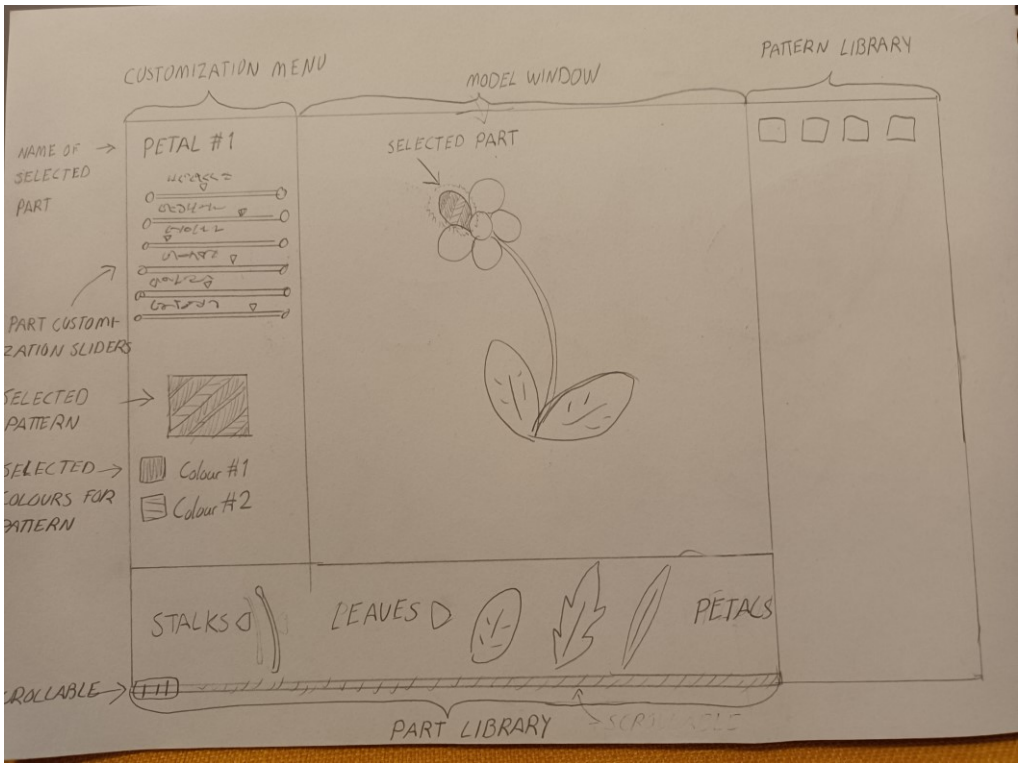

FIGURE 6. Concept drawing of the feature, drawn with pen and paper by the author

3.3 Implementation of the prototype

All the features of the concept are not created on the prototype. Prototype is used to answer to the questions:

1. Can you implement mesh model customization in Unity?
2. What is the best approach to the task?

As prototype is supposed to be quick test of concept (refer to paragraph 1.4.4 Prototyping the game.), only bare minimum of the concept is implemented. Very rough UI was created to accommodate to easier view of how things would work and to create a helpful window for testing. Image (Figure 7) shows the view of the prototype.

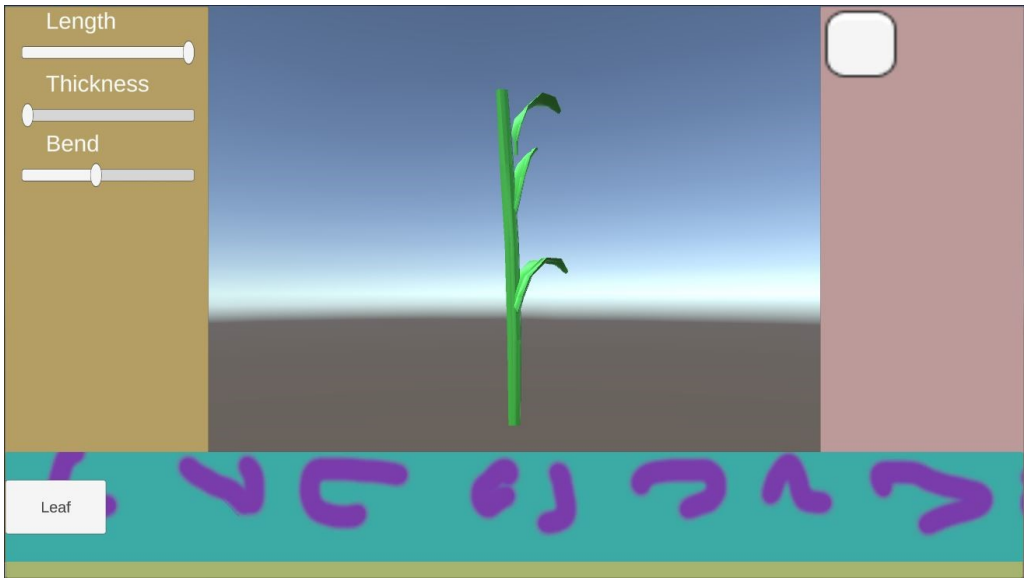

FIGURE 7. Capture of the prototype view in play mode in Unity by the author

Runtime mesh manipulation in Unity itself was first tested as possible option for mesh deformation. In theory it could work, and were able to do some deformations, however doing more specific things would get very complicated and would require in-depth research. Other options were then investigated, and a more straight-forward answer was found with shape keys / blend shapes.

In Blender, shape keys can be assigned to model. These shape keys allow smooth automatic mesh manipulation between the default shape of the mesh and the maximum deformation specified to shape key. It also allows blending different shape key values. The model can then be exported as an fbx (Filmbox) file type with its shape keys to Unity. Image (Figure 8) shows how the shape keys work in Blender. By selecting a shape key and changing the value, the model will deform accordingly.

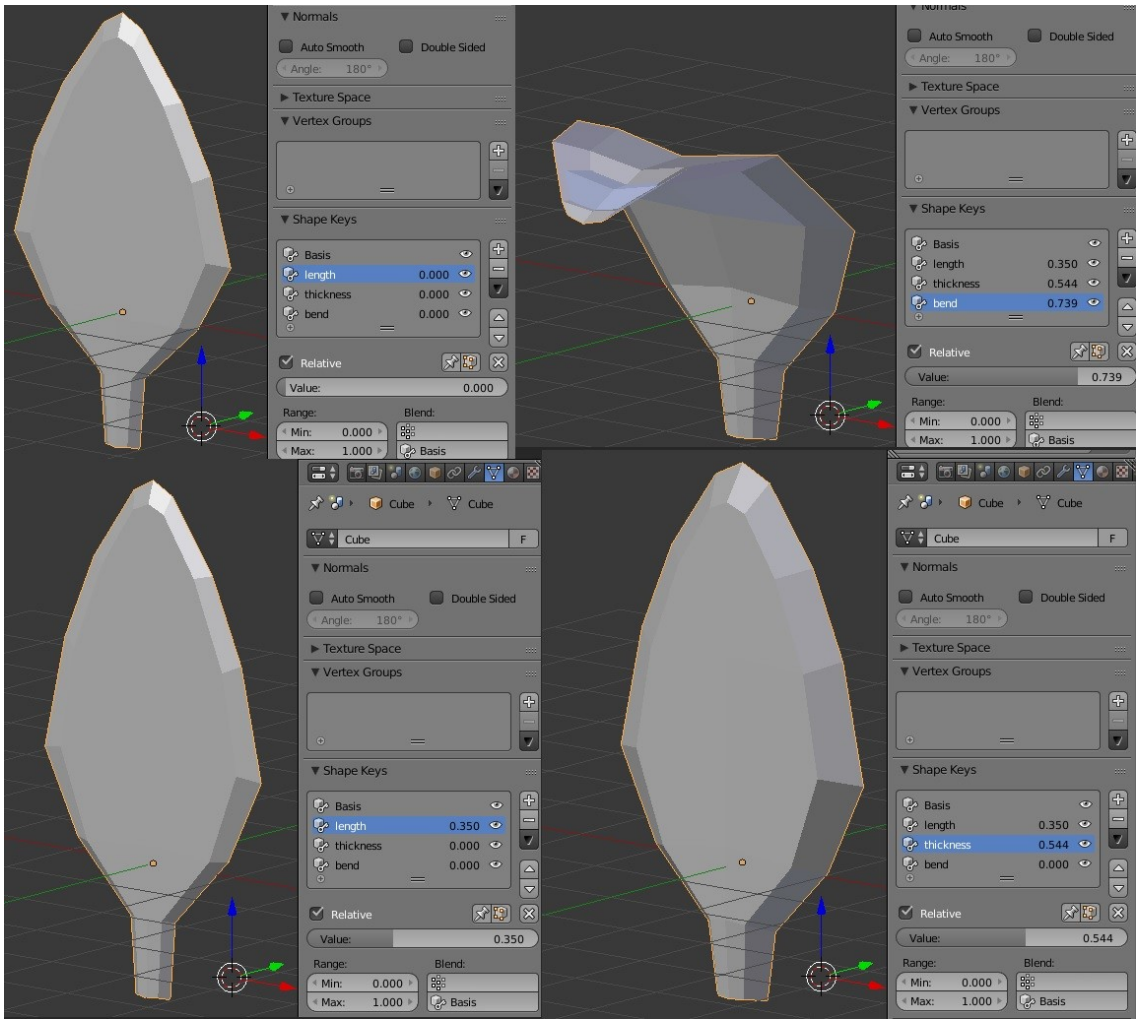

FIGURE 8. Capture of the leaf model with its shape keys in Blender by the author

Unity has the functionality to use these shape keys, or blend shapes, as they are called in Unity. The changing of these values was then attached to the sliders in the UI to enable editing of the values in game mode. Two classes handle this together, class BlendShapeSlider that is attached to a slider, and class CameraRayCast that is attached to the main camera.

```
//Add listener to slider to adjust blendshape value according to slider
//value by calling cameraRayCast.ChangeBlendshapeValue(_blendShapeName, value)
public void AssignBlendShape(string blendShapeName)
{
    _blendShapeName = blendShapeName;

    try
    {
        UnityEventTools.RemovePersistentListener(slider.onValueChanged, 0);
    }
    catch (System.Exception)
    {
    }

    slider.onValueChanged.AddListener(value =>
        cameraRayCast.ChangeBlendshapeValue(_blendShapeName, value));
}
```

FIGURE 9. Code from BlendShapeSlider class that attaches a listener to slider, snippet from Visual Studio 2019, code by author

In slider's script is a listener to follow on the slider's value change(Figure 9), and then this listener calls for function ChangeBlendShapeValue from camera's script (Figure 10). Camera's script stores the data about the blend shape names, and updates them depending on the selected mesh. The rest of the scripts and model-assets can be viewed in the reference [19]. Demo video of the prototype can be viewed in the reference [20].

```
public void ChangeBlendshapeValue(string blendShapename, float value)
{
    if (!blendShapeDatabase.ContainsKey(blendShapename))
    {
        Debug.LogError("Blendshape " + blendShapename + " does not exist!");
        return;
    }

    value = Mathf.Clamp(value, 0, 100);

    skmr.SetBlendShapeWeight(blendShapeDatabase[blendShapename], value);
}
```

FIGURE 10. Function from CameraRayCast class that slider's attached listener calls, snippet from Visual Studio 2019, code by author

4 CONCLUSIONS

Video game design is an expansive field that encompasses over many professions. In this thesis is only a very basic overview on all things to consider in the design process. It has taken countless hours and great efforts of studying and research just to compile this thesis, and yet it does not come close to the efforts you would have to make to just plan a game. However, it is evident that good designing and planning will pay itself off in time and efficiency over the course of development of a game and can result in a quality finished product.

Working the theory side of this thesis taught me a lot about the design process, which in turn helped creating the demo game concept. If it was made for publication however, more actual research would have to still be invested into it. I created the concept solely based on what I know from my own experience, which of course is biased for my views and would not make yet for appropriate publishable game concept. For this thesis, I deemed sufficient.

In the prototyping process it was easy to get side-tracked pondering and testing features that were not essential for the prototype. UI, as raw as it is in the prototype, was still thought about much more than it should have been. It was however successful prototyping, the answers were found and based on them, a working feature could be built. Perhaps in the future I will return to my prototype and game concept and create a tech demo of it as a hobby project.

5 REFERENCES

[1] P. Zackariasson, T. Wilson, *The Video Game Industry: Formation, Present State, and Future*, Taylor & Francis, 2012. [E-book] Available: ProQuest Ebook Central.

[2] M. Wolf, S. Murphy, J. Aldred, R. Baer, B. Camper, K. Collins, R. Dannenberg, L. Herman, E. Huhtamo, C. Kocurek, *Before the Crash : Early Video Game History*, Wayne State University Press, 2012. [E-book] Available: ProQuest Ebook Central.

[3] K. Salen, E. Zimmerman, *Rules of Play*. Massachusetts Institute of Technology, 2004.

[4] E. Adams, *Fundamentals of Game Design, Third Edition*. Pearson Education, Inc, 2014.

[5] N. Esposito, "A Short and Simple Definition of What a Video Game Is." *ResearchGate*, 2005. [Online]. Available: https://www.researchgate.net/publication/221217421_A_Short_and_Simple_Definition_of_What_a_Videogame_Is [Accessed: Mar. 13, 2022].

[6] J. Ward, "What is a Game Engine?" *gamecareerguide.com*, Apr. 29, 2008. [Online]. Available: https://www.gamecareerguide.com/features/529/what_is_a_game_.php [Accessed: Mar. 13, 2022]

[7] M. Sellers, *Advanced Game Design*, Pearson Education, Inc, 2018. [E-book] Available: O'Reilly.

[8] A. Kramarzewski, E. De Nucci, *Practical Game Design*, Packt Publishing, 2018. [E-book] Available: O'Reilly.

[9] J. Vandenberghe, GDC talk, Topic: "Applying the 5 Domains of Play: Acting Like Players", 2018. [Online] Available: https://www.youtube.com/watch?v=6uX6ye66NK0 [Accessed: Feb. 20, 2022]

[10] S. Rothmann and E. Coetzer, “The Five Big Personality Dimensions and Job Performance” *SA Journal of Industrial Psychology*, 2003. Available: SA Journal of Industrial Psychology, https://sajip.co.za/index.php/sajip [Accessed Feb. 20, 2022]

[11] Internet Matters, “Game age ratings explained”, *Internet Matters*, [Online]. Available: https://www.internetmatters.org/resources/video-games-age-ratings-explained/#:~:text=These%20are%20ratings%20are%20used,under%20the%20age%20of%2012. age ratings [Accessed Feb. 20, 2022]

[12] L. Plant, “Why Digital Games Could Totally Dominate Physical Formats In Just a Few Years”, [Online]. Available: https://nordic.ign.com/news/44939/why-digital-games-could-totally-dominate-physical-formats-in-just-a-few-years [Accessed Apr. 16, 2022]

[13] Tutorials Point, “Flowchart Elements”, *Tutorials Point.* [Online]. Available: https://www.tutorialspoint.com/programming_methodologies/programming_methodologies_flowchart_elements.htm [Accessed Apr. 16, 2022]

[14] A. Sherman, “What is a Storyboard?”, *StoryboardThat* [Online]. Available: https://www.storyboardthat.com/articles/e/what-is-a-storyboard [Accessed Apr. 16, 2022]

[15] Computer Hope, “Spreadsheet”, *Computer Hope* [Online]. Available: https://www.computerhope.com/jargon/s/spreadsheet.htm [Accessed Apr. 17, 2022]

[16] J. Chell, The Art of Game Design, Third Edition, A K Peters/CRC Press, 2019. [E-book] Available: O’Reilly.

[17] M. Rehkopf, “What is a Kanban board?”, *Atlassian* [Online]. Available: https://www.atlassian.com/agile/kanban/boards [Accessed Apr. 23, 2022]

[18] American Museum of Natural History, “Parts of a Flower”, *amnh.org* [Online]. Available: https://www.amnh.org/learn-teach/curriculum-collections/biodiversity-counts/plant-identification/plant-morphology/parts-of-a-flower [Accessed Apr. 23, 2022]

[19] Johanna Pauna, prototype Github repository, Available: Link to Github repository [Accessed May 8, 2022]

[20] Johanna Pauna, demo video of the prototype, Available: https://www.youtube.com/watch?v=RQzTMlkssAk [Accessed May 8, 2022]