



Indie Game Development and Marketing

Bachelor's Thesis
Computer Applications
Spring 2025
Marika Duhhanina

DP Computer Applications
Author Marika Duhhanina
Subject Indie Game Development and Marketing
Supervisors Elina Kulmala

Year 2025

The goal of this thesis was to create a simple game using Godot 4.4, market it by using social media such as Bluesky, YouTube, and TikTok, and publish it on Itch.io. The aim of the thesis was to understand how a beginner indie game developer can benefit from publishing a free simple video game without any ad revenue, in-game purchases, and additional paid content. In addition to that, the purpose was to learn the challenges of self-publishing a video game, as well as what content type creates most engagements on social media. The thesis sought to analyse the anonymous collected data from the engagements of the social media accounts of the author. As a marketing goal, it was decided to aim for 10 people to play the game.

The thesis was practice and research oriented. The theoretical part was done through independent research using materials from Google Scholar and HAMK Finna. Online materials such as forums, learning videos, and documentations were used to assist with the practical part of the project. Concepts related to game development and marketing were explained. The author had prior knowledge in game design and development, background in digital art creation, and had previously created a video game using Godot 4.3. However, social media had not been used for marketing. To reach the goal, marketing and development plans were created. Kanban board was utilized for project management. The data was analysed by utilizing charts in Excel.

The analysis indicates that the game did not receive much attention. Marketing strategy on Bluesky resulted in 24 gained followers, whereas on YouTube and TikTok 0 followers were gained. Based on the analysis, it can be suggested that the people might not be interested in short video games with simple mechanics, notably from an unpopular indie game developer. However, the developer can create small video games to gain experience in game development and receive followers by creating posts regarding the development process on social media. It can be recommended that the video game developers should not only create marketing and development plans, but detailed plans for the content that will be posted.

Keywords Indie, Video game, Development, Marketing, Social media
Pages 46 pages and appendices 1 pages

Glossary

2D	Two-dimensional
3D	Three-dimensional
AAA (Triple A)	High-budget game company
AR	Augmented reality
Blueprints	Visual scripting language used in Unreal Engine
C++	Programming language
C# (C sharp)	Programming language
DAW	Digital Audio Workstation
Digital art	Drawings inside of the computer
Godot	Game engine used to create video games
GScript	Programming language used in Godot
Indie	Independent
Open source	Software with open for the public code
PC	Personal computer
Pixel art	Digital art created from the small amount of pixels
Python	Programming language
Synthesizer	Electronic instrument that produces sound
VR	Virtual reality

Table of Contents

1	Introduction	1
2	Video games	2
2.1	Defining a game and a video game	2
2.2	Making a video game fun	3
2.3	Video game genres	4
2.4	Choosing the target audience	5
2.5	The process of game development	6
2.6	Game development tools	8
2.6.1	Game engine	8
2.6.2	Digital drawing software	9
2.6.3	Sound design software	9
2.6.4	Version control	10
2.6.5	Online video game platforms	10
2.6.6	Kanban Board	10
3	Marketing a videogame	11
3.1	Using social media for marketing	11
3.1.1	Social media platforms	12
3.1.2	Branding	13
3.1.3	Creating awareness of the game	13
3.1.4	Analysing indie game developer's social media accounts	15
3.2	Publishers	15
4	Choosing tools and creating plans	16
4.1	Social media and branding	16
4.2	Softwares	16
4.3	Websites	17
4.4	Plans	17
4.4.1	Marketing plan	17
4.4.2	Game development plan	18
5	Development and marketing	19
5.1	Game idea	19
5.2	Game development	20
5.3	Game marketing	40
6	Results	41

7 Summary	46
References	47

Figures

Figure 1. The flow chart by Csikszentmihalyi	4
Figure 2. The graybox prototype of the game.....	23
Figure 3. Concept art	24
Figure 4. Player character sprites	25
Figure 5. Updated progress bars.....	25
Figure 6. Enemy sprites	26
Figure 7. Health bar	27
Figure 8. Hit indicator shader	28
Figure 9. Empty and full pollen progress bar.....	28
Figure 10. Arrow navigating to the next level.....	29
Figure 11. Equipped power-ups	29
Figure 12. Sky variations for morning, noon, evening.....	30
Figure 13. Cloud sprites	30
Figure 14. Updated version of the gameplay.....	31
Figure 15. Story flashback	31
Figure 16. Beetle's sprite with and without yellow paint on its back.....	32
Figure 17. Title screen	32
Figure 18. Level selection screen.....	33
Figure 19. Victory screen	34
Figure 20. Game over screen.....	34
Figure 21. Interactive tutorial.....	35
Figure 22. Playable character's outline	36
Figure 23. Comic snippet	36
Figure 24. Custom "Made with Godot" splash screen.....	37
Figure 25. Additional shadow logic.....	38
Figure 26. Audio Stream Randomizer	39
Figure 27. Audio transition editor	40
Figure 28. Chart of the followers gained on Bluesky	42
Figure 29. Chart of the total sum of likes by content of the post on Bluesky	42
Figure 30. Chart of the sum of likes by the content of the post on Bluesky.....	43

Figure 31. Chart whether the humour was used in post on Bluesky	43
Figure 32. Video view count on TikTok	44
Figure 33. Video view count on YouTube.....	44
Figure 34. Game data from Itch.io.....	45

Program codes

Program code 1. Character movement	21
Program code 2. Wait a process frame	31
Program code 3. Zoom implementation	38

Appendices

Appendix 1. Data management plan	
----------------------------------	--

1 Introduction

It is known that indie game development allows creators to work independently from big studios, as well as provides them with a possibility to start own business or a company. However, being a video game developer is not easy, especially if a person is working on the project individually. That means that they will have to plan the project idea, choose an art style of the game, think about the people who will play the game, write code, create sound effects and music, and last but not least, make the game itself engaging. One of the things that must be considered before starting to develop a video game is to think about the benefits of the project, since creating a video game takes an extraordinary amount of time.

Nevertheless, to start earning profit from the game, people believe that the developer should have followers who would buy the product. One of the methods an indie game developer can do is create a marketing strategy they can use on social media. In addition, by utilizing marketing plans, game developers can spread awareness of the product. The author believes that by creating a video game and using social media will help them grow as an independent video game developer. Additionally, it was decided to learn the benefits of creating a free video game.

The research has to be conducted on the definition of a game and a video game, learn more about game development and social media marketing. The thesis is based on the problem of understanding the benefits of creating a free simple video game, learning the obstacles of self-publishing, and analysing what posts create most engagements on social media in game development community. The purpose of the thesis is to create a simple video game, learn more about social media marketing, and to conduct an analysis of the author's social media accounts. As a marketing goal, the aim is to have 10 people play the game.

The research questions for this thesis are:

- How can an indie video game developer benefit from publishing a free simple video game without any ad revenue, in-game purchases, or additional paid content?
- What are the challenges of self-publishing a video game?
- What form of content creates most engagements on social media?

2 Video games

Before starting the development process of a video game, it is important know what a game and a video game are, as well as what is needed to make a video game. In addition, understanding what makes video games engaging will allow the developer to create an enjoyable experience. After that, it is necessary to understand the tools that can be used in order to develop a video game.

2.1 Defining a game and a video game

Midgley (1974, pp. 231 – 253) states that a game requires rules to create challenge, a conflict for entertainment, an opportunity to get distracted from the life, and it is natural for animals to play and compete with each other. On the other hand, it is believed that neither the conflict, nor the competition is required to create a game, since the game can serve a purpose of education, health, or practice. In addition, this definition does not concern the aspect of being entertained by elaborating: “just because a game is not fun does not mean that it is not a game”. Moreover, in order to participate in games, one is playing a role of someone that they are not and pretending that they are placed in another environment. (Ernest, 2013, The Major Games chapter, What is a Game? paragraph)

Toys are used to create an environment to be played in. In addition, they serve as a guidance of a certain rules. However, it is not required to follow them. In addition, that the games should also have a goal. Although, in some arcade games the player has an unreachable objective, but they can aim for a higher score than the previous play session. In addition to that, an objective should be clear for the player to understand what is needed to achieve the goal. (Rogers, 2024, Welcome N00bs! chapter; Ernest, 2013, Games and Video games chapter, Conventional Games Versus Video Games paragraph)

Video games are the games that operate on a computer. Games and video games have similarity due to the fact that both are interactive medium, in which one creates an imaginary environment and plays a certain role. Video games have limited actions for the players to choose from, since the actions have to be programmed. In 1950s, computers of considerable size were used to play simple games. With time, people played games in arcades and attraction parks. With the invention of consoles, people could play video games from home. (Ernest, 2013, Games and Video games chapter, Conventional Games

Versus Video Games section; Rogers, 2024, Welcome N00bs! chapter, A (Somewhat Brief History of Video Games paragraph)

Video games can have stories, but not all games are required to have one (Rogers, 2024, Writing the Story chapter). Hadzigeorgiou and Judson (2017, p. 13) suggest that storytelling can be used to address an issue. For instance, making an environmental problem serve as a narrative device to raise awareness of the topic. According to Cassese et al. (2023, p. 1), research of Google for Games 2020 demonstrates that 70% of people find it important for games to have diverse characters and stories. Therefore, game developers of different backgrounds are encouraged to create stories for their games.

In regard to the spelling, during the research it was noticed that while some sources use term “video game”, others use “videogame” as one word. It is important to note that such dictionaries as Oxford English Dictionary and Cambridge Dictionary use the term video game as two separate words (Oxford English Dictionary, n.d.; Cambridge Dictionary, n.d.). In his blog, Scalzo (2022) has conducted research, and he concludes that an online search for the “video game” gives 318,000 results, while “videogame” gives only 7,000. Therefore, it can be speculated that using “video game” as two separate words would be advantageous for online marketing. It is worth mentioning that for the thesis, words “video game” and “game” are used interchangeably to define a computer game.

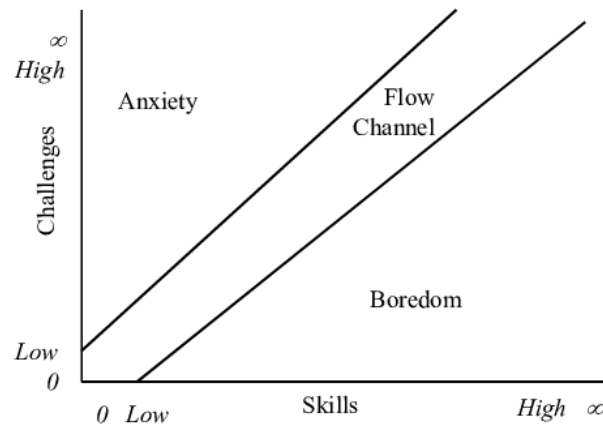
2.2 Making a video game fun

In his e-book, Koster (2013, What Games Are chapter) states that mystery creates fun, and as the player is exploring the game by learning the game mechanics, in-game world, solving puzzles, the player is having fun. If an activity becomes repetitive, it becomes boring and unengaging. However, as he claims, it does not mean that by adding multiple new activities for the player to learn all at once will make the game engaging, since the player might feel overwhelmed by the amount of the new knowledge.

Similarly, Csikszentmihalyi (1990, pp. 72-74) proposes an idea that enjoyable activities are designed for better experience by creating a certain set of challenges that require skills. By achieving the milestones of acquiring those skills, the feedback is provided. Therefore, it creates the flow state demonstrated in Figure 1. The psychologist states that every enjoyable activity allows the person to discover new things. In order to be in the flow of enjoyment, one should have the optimal level of difficulty that would create challenge.

Although, the challenge should not be too hard to create frustration. Likewise, the challenge should not be simple to make a person lose interest and therefore feel bored.

Figure 1. The flow chart by Csikszentmihalyi



Csikszentmihalyi (1990, p. 75) recommends that if a person finds themselves in the state of boredom, they should create a challenge to re-enter the flow state and enjoy the activity. Whereas if the person is frustrated or anxious, the person should gain more knowledge and skill. It is also possible to lower the difficulty, however, the person will return to the original state without gaining any skill set. It is important to note that the situation of the flow can differ from person to person, as well as it can depend on the current experience and the emotional state.

It is worth mentioning that in order to design engaging video games, not only does the game designer should be capable of designing video games, but also all different kinds of games. Therefore, if a person understands the essential elements of a game, they can create an enjoyable experience for the video game as well. (Ernest, 2013, Games and Video Games chapter, Conventional Games Versus Video Games section)

2.3 Video game genres

Unlike films, games are an interactive medium, hence the genre of the game gives the gamers more information than the theme or the setting of the game. For instance, while with the films a person can say that it is a thriller or comedy, it cannot be considered the same for the video games. Since those genres do not provide gamers with many details,

instead, video games use genres that tell what kind of game people will be playing along with the genre of the story. (2024, Preparing a Game Concept chapter, Structure of a game concept document section, Genre paragraph; Rogers, 2024, Welcome N00bs! chapter; Ernest, 2015, What Is a Genre? chapter)

Kramarzewski and De Nucci (2023, Preparing a Game Concept chapter, Structure of a game concept document section, Genre paragraph) provide with an example that the genre First-Person Shooter (FPS) can be associated with the gameplay style, whereas fantasy means the theme of the story. The genres can also be redefined to push the boundaries of it, creating a more interesting video game. It is crucial to define a genre when the developer is designing a video game, since the creation of a game heavily relies on it due to the fact that the video game genres have different characteristics.

One of the most popular video game genre is an action game. It relies on eye and/or hand coordination. This genre has the most variety of games, due to the fact that action can be designed in different ways. Other popular genre is adventure games that rely on the narrative, characters, and the story of the in-game world. The next genre is educational games with a main goal of teaching the player while being entertaining. Not only do the games have genres, but also subgenres. Additionally, it is not required for the game to have only one genre. If a developer combines two or more genres, they create a hybrid game. (Rogers, 2024, Welcome N00bs! chapter, Game Genres paragraph)

It is believed that for the game to succeed, the developer of the game should learn more about similar games. Moreover, the developer should analyse online discussions regarding the games. Therefore, they can learn what gamers like and dislike in those games. (Kramarzewski, De Nucci, 2023, Preparing a Game Concept chapter, Knowing your competition paragraph)

2.4 Choosing the target audience

One of the most important aspects of designing a video game is to consider the target audience, in other words, who the video game was made for. Whether the game will be created for casual gamers, or hardcore gamers. Casual gamers prefer easy games and not to spend much time playing. On the other hand, hardcore gamers would rather play more competitive and difficult games, as well as to spend more time playing. (Rogers, 2024, Ideas chapter, Gamers Want Good Games paragraph; see also Boyles, 2020)

By defining the target audience, the developer can eliminate certain design decisions. It can be agreed that the developer should not aim to develop a game that suits everyone. Thus, it is recommended deciding on the right target audience and focus on creating engaging experience for them. (Dreskin, 2015, Marketing Fundamentals chapter, Target Audience section; Kramarzewski & De Nucci, 2023, Preparing A Game Concept chapter, Structure of a game concept document section, Audience paragraph)

Furthermore, it is suggested that the developer should define the target audience by their interest and not by a generic definition, whether that is a casual or hardcore player. It can be explained by the fact that nowadays everyone can be a gamer. Additionally, many video game developers conduct research by publishing concepts, create trailers of the games that do not exist to see if the game is desired by the gamers. Kramarzewski & De Nucci, 2023, Game Concept chapter, What is a game concept? section, Audience paragraph)

Rogers says that it is also crucial to decide on the age of the audience. If the game is created for the kids, it is important to note that kids want what was created for the target audience that is older than they are, since they are not interested in playing games that are created for them. It is important not to underestimate the younger audience by creating a game that is extremely easy. Due to the fact that children learn and adapt faster than adults, they are better players. (Rogers, 2024, Ideas chapter, Gamers Want Good Games paragraph)

2.5 The process of game development

Both Bramble (2023) and Piskunov (2025) state that the development process can be divided into seven stages. Those stages include planning, pre-production, production, testing, pre-launch, launch, post-launch.

According to Hollstrand (2020, p. 3), Chandler (2013) states that the planning stage is for brainstorming ideas. Concept may be vague, since the planning stage is dedicated to exploring the ideas and choosing the key elements for the game, such as game mechanics, story, game setting and the world, characters. It is suggested that the ideas can be found by brainstorming, exploring topics different from one's preferences, doing everyday activities, or experiencing a bad game. (Rogers, 2024, Ideas chapter, Brainstorming paragraph)

It is different to have a clever idea, and a marketable idea. As he states, there can be an idea of a game that a game designer would enjoy playing. However, it would be difficult to market the game to the public. Nevertheless, game developers should not be discouraged by this statement, as they can create innovative and remarkable games. (Rogers, 2024, Ideas chapter, Ideas: Where to Get Them and Where to Stick Them paragraph)

During pre-production, game developers design a prototype of the game using the concept. It is important to note that the early stages play a role of exploring and testing the game concept, as well as expanding it by investigating new ideas. Depending on the goal of the prototypes, they are not necessarily playable. One of the methods that is used for prototyping is called grayboxing. In other words, a prototype that is created using only the essential and simple components, using basic geometric shapes. Grayboxing helps with fast and easy modifications, due to the fact that appearance is not the main focus. (Hollstrand, 2020, pp. 2-3)

The next stage is the production. During this stage, the prototypes are further developed into a game. In AAA game companies the cost increases during this stage due to the fact that the team of the developers expands by including artists, sound designers, and programmers. (Hollstrand, 2020, p. 4)

Throughout the testing phase it is crucial to learn whether the game is engaging. During the testing phase the feedback is received, and adjustments are made. It is noteworthy to mention that testing is not only important in the late stages of game development process, but also in the early stages. Therefore, the testing is considered to be important throughout all of the development process. As a result of testing, the product is more likely to provide with an enjoyable experience. (Hollstrand, 2020, p. 3)

The common playtesters are the developers of the game itself, friends, acquaintances, other people, and people on the internet. Additionally, there are testers who play the game and give their feedback only once, as after the person has previously played the game, they would have a different opinion on it based on the previous knowledge of the game. Since they know nothing about the game, they can give an opinion that is useful from a person who has experienced the game for the first time only once. Those testers are useful to test the first levels, tutorial, difficulty aspects, feedback on the story, and the ending. (Bond, 2022, Game Testing chapter, The Circles of Playtesters paragraph)

2.6 Game development tools

To create and publish a video game, it is required to know and understand the tools. They can represent themselves not only as softwares, but also websites, services, and methods a developer would need to use.

2.6.1 Game engine

Game engine is a software that aids a person with the development of the game by providing the required tools. Therefore, the developers do not have to create everything from scratch for every new game, by giving an opportunity to use a set of pre-made features, and thus, the developers can focus their attention on the development of the game itself. Some of the common functionalities include rendering or showcasing the game on the display, physics simulation with collisions, animation, sound, and image management, building the games on multiple platforms, such as PC, mobile phone, consoles, web. (Bradfield, 2023, Introduction to Godot chapter)

AAA game studios can afford to create their own engine by creating better flexibility in creating video games. In addition, custom-made game engines make the developers independent, compared to the pre-existing game engines. (Konvoy, 2023) Nowadays, there is a large variety of game engines available that were created for different purposes.

Unity can be used for both 2D and 3D game development. The engine uses C#. The website states that the engine allows the games to be versatile by giving access to create games on multiple platforms. Unity offers a free version, as well as a subscription model of Unity Pro. The subscription that allows developers to build games to multiple platforms, such as Nintendo Switch, PlayStation, Xbox, and apps for Apple Vision Pro. The engine has a large community of over 1.2 million monthly active users and large number of online tutorials available. (Unity, n.d.)

Godot is a free open source game engine. The engine is used for 2D and 3D games. Regarding programming languages, the engine uses GDScript and C#. It is stated in the documentation of Godot that the language is similar to Python. The engine allows the games to be exported for PC, mobile, and web. (Godot, n.d.; Gregory, 2023, Foundations chapter, Open source Engines section; Bradfield, 2023, Introduction to Godot chapter).

Unreal Engine is used to create 3D games, and it is known for a possibility to create games with high-quality graphics. The games can be built for PC, console, mobile, VR, and AR. The engine has a free licence, and for game developers who have over 1 million USD in revenue, the fee of 5% is applied. The engine uses C++ and blueprints. (Unreal Engine, n.d.)

2.6.2 Digital drawing software

Digital drawing softwares are used to create digital art. Those softwares allow users to modify, duplicate art, as well as to zoom in to see the details and zoom out to see how the picture looks from a distance. (Davenport, 2024) To create digital art, a person can use a smartphone, a mouse, or a drawing tablet (XPPen, 2024). In the game development industry, they are used to create visual assets, such as art, sprites, environments, and animations. The examples of the drawing software include:

- Krita is free and open source, and it gives an opportunity to do both art and animation (Krita, n.d.).
- Gimp is free and open source, it is used for photo editing and graphical image creation (Gimp, n.d.).
- Aseprite is a paid software that is used for pixel art and animation (Aseprite, n.d.).

2.6.3 Sound design software

Similar to visual assets, audio assets, such as music, sounds effects, dialogue are important to establish an atmosphere (Ernest, 2013, Designing And Developing Games chapter, Audio Director paragraph). Those softwares are:

- FL Studio is a paid DAW that, as the website claims, used by million of people every day. It offers 4 license options, such as Fruity edition for €99, Producer edition for €299, Signature Bundle for €319, and All Plugins Edition for €539. (FL Studio, n.d.)
- Beepbox.co is a free online tool used to create music. It has large number of synthesizer instruments that can be modified. (Beepbox.co, n.d.)
- Audacity is a free audio editing software that can also be used for recording. (Audacity, n.d.)

- Freesound is an online sound database. The users that provide with the sounds choose a license for the sound that varies from Creative Commons 0 to Creative Commons 4. (Freesound, n.d.)

2.6.4 Version control

Version control is a method of creating back-ups and is used to ensure that the code and the assets will not be removed in case of an accident. Due to the fact that the project is backed-up, it is possible to explore and experiment with the code. (Bradfield, 2023, Next Steps And Additional Resources chapter, Version control – using Git with Godot paragraph) One of the most popular options is a free tool Git with a desktop version of GitHub Desktop. In addition, GitHub allows programmers to collaborate on their projects. (GitHub, n.d.)

2.6.5 Online video game platforms

Online platforms allow game developers to publish their game, providing the game developers with an opportunity to for the gamers to play their games. Steam is one of the most popular gaming platforms available with over 100 million users. It requires approximately \$100 to publish the game to the store. (Steam, n.d.)

itch.io although does not have such a large community as steam, it allows the indie developers to publish their games without any fees, as well as to publish the web version of the game. However, it is noteworthy to mention that the payout mode and the tax information is required to be set in order to make profit. The website provides the users with two payment modes. The first mode allows the website to collect payment from users, and it will proceed with the tax information, as well as a refunding process. However, this mode requires a 7-day wait period before receiving the payment. The second mode signifies that the payment will be sent to the game developers directly. Although, the developer would need to be responsible for the taxes, as well as refunds. (itch.io, n.d.)

2.6.6 Kanban Board

Kanban Board is an agile tool that aids managing the project. Agile means that its main point is prioritization, continuous delivery, and continuous improvement (Moraes & Romão, 2022, Managing Projects and Tasks with Microsoft Planner and To-Do chapter). The board

portraits three states “waiting”, “in progress”, “completed”. It is possible to adapt the board to one’s needs as well as to change or add as many states as needed in order to understand how much work is to be done, is being done, and is done. (Dalton, 2018, p. 187)

Kanban Board can be created using Microsoft Planner service. Microsoft Planner is a tool that is included in Microsoft 365 service. Microsoft Planner provides several pre-made templates that can further be modified. (Microsoft, n.d.)

3 Marketing a videogame

In order to promote a product, marketing strategies are utilized (Seyyedamiri & Tajrobehkar, 2019, p. 76). Equally, not only it is important to create a game, but also to market it to the public for the game’s success. The frequency and the time for the developers to start to implement the marketing strategies is important. In addition, great marketing strategies allow indie developers to increase their audience and public visibility. (Dreskin, 2015, Marketing Fundamentals chapter)

3.1 Using social media for marketing

Jeswani (2023) states that by utilizing social media, businesses are increasing their visibility. By interacting with the audience, posting media content, creating a community, and receiving feedback, the company can increase brand awareness. Moreover, it is suggested that by increasing brand loyalty through communication with the audience, the customers’ satisfaction increases. Therefore, the players can share their positive experience of the brand. However, it is important to understand the responsibilities of using social media.

To successfully create a marketing tool out of a social media, it is important to develop an audience. It is suggested to start posting on social media early, since creating an audience takes time. Additionally, the developer should create posts once per week or more often, as it can often lead to higher engagement levels and audience connections. Similarly, it is recommended to plan the marketing strategy in advance. By creating a plan, it will allow developers to know their goals and help them plan the post creation schedule. (Dreskin, 2015, Marketing Vehicles That Can Work Well for Indies chapter, Audience Development—Social Media, Email and Beyond paragraph; Funk, 2015, Best Practices chapter)

Dreskin (2015, Marketing Vehicles That Can Work Well for Indies chapter, Audience Development—Social Media, Email and Beyond paragraph) mentioned that the developer should increase the frequency of the posts in the final weeks before the game gets published as the developer will have more news and content to publish. It is not necessary to only post about the game, but to also comment on other people's posts, discuss about relevant trends, and other games. Although, it is mentioned that it is important to not post about the topics that are far from the ones that the audience sees relevant, since it might lead to losing followers, hence decreasing audience's interest of the project.

Moreover, it is recommended not to rely on accidental success. Namely, if the creator of the post received much attention without planning, it is probable for them to not seize the opportunity to promote their product. In order to utilize social media for marketing, not only should a person have a plan, but also to continue posting and sharing content, even if the posts do not gather much attention. (Evans, 2010, It's Not Easy, Quick, or Cheap chapter)

It is recommended to utilize hashtags to increase posts' visibility. A person should use hashtags relevant to the topic of the post. Additionally, it is mentioned that it is possible to create own hashtags. (Funk, 2013, Platforms chapter)

Kamila and Jasrotia (2023, p. 578) state that companies have to be aware of their marketing ethics. Additionally, they claim that companies must consider long-term consequences to the marketing strategy. Equivalently, Dhiman (2023) says that individuals also must be responsible for their actions on social media. Therefore, it is important to note that users on social media, including game developers, must understand the implications of their posts, as well as be accountable of what they post on social media.

3.1.1 Social media platforms

YouTube is an online video sharing platform used by millions of people every day. Game developers can share the game trailers, implemented game features, gameplay videos, and music. Similarly to YouTube, TikTok is a social media platform that is popular with video content. Although unlike YouTube, TikTok uses short-term vertical video format. Bluesky is an open source social media platform where users can share posts with text, video and art content. It is known for being a new and growing platform with the feeds that separate the topics on their own category. (YouTube, n.d.; TikTok, n.d.; Bluesky, n.d.)

In addition to the social media accounts, in order to store social media links in one place to later share with the audience, such online services can be used as Linktree, Carrd. Both services offer free and paid license options. While Carrd allows users to create a personalized website or a page, Linktree serves as a tool with the main purpose of sharing links. (Linktree, n.d.; Carrd, n.d.)

3.1.2 Branding

A person's brand defines how they want to be perceived by others, as well as it defines a person or company's personality. It is noteworthy to mention that if a person knows the reason behind their activity, it is easier to showcase the brand. Moreover, the branding includes the title of the game and the company, logos, slogans and taglines, trademarks, advertising, websites, social media accounts, marketing strategy. (Dreskin, 2015, Branding chapter)

The advantages on creating a branding and having a name open creates more trust with game publishers and it may make a professional first impression, whereas if the person decides to use a nickname as their online name, it might lead to a different audience (Hart, 2024). One of the reasons users stay anonymous online is to prevent other people online from conducting aggressive behaviour and bullying, as well as from reaching out directly to them, if they were to use their full name (Christopherson, 2006, 3040). Hart (2024) mentions that another reason could be that the people want to be known and seen as their alias that can be that the nickname can be memorable. Creating an interesting, or abstract nickname could make a person online noticeable.

3.1.3 Creating awareness of the game

It is important to consider how the project will be announced. The method of establishing the awareness and of developing an interest differ from studio to studio. For instance, a brand-new project from a popular game developer would look significantly different compared to a game announcement from a newly established studio. Additionally, while some might publish videos, images and screenshots, extensive details of the game couple of months before the release date, others might build interest over a larger period by planning the timing of showcasing the game-related content. (Dreskin, 2015, Developing Your Marketing Campaign and Calendar chapter, Creating Awareness, Sustaining Interest and Building to a Peak paragraph)

Dreskin creates an example that the game studio could first announce the game. Then some weeks later they publish the screenshots. After that they provide with more information about the game, and after a week they publish a video. It is important to prepare the schedule and plan the marketing beforehand. This method can aid to create interest of the game. Hence the people will want to learn more about the game until its release. It is crucial to save important content and big announcements closer to the release day to create even more anticipation. The examples of the content that can be published include the following aspects. (Dreskin, 2015, Developing Your Marketing Campaign and Calendar chapter, Creating Awareness, Sustaining Interest and Building to a Peak paragraph)

Big reveals and announcements, which include in-game content, game features, release dates. It is important to consider that if one wants to announce a date, in the beginning they should include a season, for example Fall, Spring, and closer to the finishing line of the game development specify the month or the exact date. This approach allows the developers to be flexible in case of the exact date deadline misses. Some studios include the price of the game with the release date announcement and open a possibility of pre-ordering the game. (Dreskin, 2015, Developing Your Marketing Campaign and Calendar chapter, Creating Awareness, Generating Interest and Momentum as Your Game's Release Date Approaches paragraph; see also Game Marketing Genie, 2022)

Videos that showcase the game in motion. It is important to know when and how to publish the video content, as well as to demonstrate the look and the feel that represents the game. Some of the examples of using a video format include announcing the release date and/or game price, creating a big trailer of the game, creating small trailers or engaging gameplay videos that focus on different important characters, game features, creating interviews, live streams and behind-the-scenes, or publishing in-game cinematics that could demonstrate the first minutes of the game. (Dreskin, 2015, Developing Your Marketing Campaign and Calendar chapter, Creating Awareness, Generating Interest and Momentum as Your Game's Release Date Approaches paragraph)

Screenshots and game images that show the User Interface (UI), characters, parts of the cutscene, gameplay. Concept art and game-related art that demonstrates the game's appearance, art direction, emotion that the game conveys, and visual style of the game. (Dreskin, 2015, Developing Your Marketing Campaign and Calendar chapter, Creating Awareness, Generating Interest and Momentum as Your Game's Release Date Approaches paragraph)

In addition, Game Marketing Genie (2022) introduce more ways of utilizing social media to increase project awareness. In the blog, one of the strategies that can be used is to create raffles, through which people can participate by sharing the post to have a chance of winning a prize. Another example of using social media is to post behind the scenes footage showcasing how a feature was implemented. Moreover, building of anticipation is suggested, and followed by an example of creating a countdown that leads to a game release.

3.1.4 Analysing indie game developer's social media accounts

It was decided to conduct research on other indie game developers' social media accounts. The reason for that is to learn more about marketing strategies that were implemented to promote their projects. The Bluesky accounts include Cult of the Lamb video game account, Tour de Pizza, Localthunk (Cult of the Lamb, n.d.; Tour de Pizza, n.d.; Localthunk, n.d.).

It was noticed that the account of Cult of the Lamb creates posts approximately 2 times a week. One of the main aspects that they used to create content was to use humour, illustrations, content from the video game, and simple animations to promote their games. Consequently, making people engage with the posts by liking, commenting, and sharing, thus helping the creators of the content spread the posts to other people.

As opposed, Tour de Pizza and Localthunk accounts publish their content infrequently, compared to the account of Cult of the Lamb. While Tour De Pizza uses humour in their posts as well, Localthunk does not use comedy. It has been noticed that those two accounts often use updates regarding their games as content for their posts.

3.2 Publishers

In his e-book, Dreskin (2015, Marketing Fundamentals chapter, Publisher vs. Self-Publishing chapter) suggests that while nowadays game developers can market their game, they can sign a contract with the publishers to aid with the marketing process. It is stated that a publisher has contacts they can use for game distribution, prepare the marketing plan that their team will follow, create trailers and game-related content. Therefore, the developer can focus on creation of the game.

However, if a game developer decides to not contact the publishers, they preserve their independency by developing their own brand, have full control over the marketing process. It is noteworthy to mention that the developer would not have to split the profit with the publishers. It is suggested that the game developers should carefully study the contracts before deciding to work with the publishers. (Dreskin, 2015, Marketing Fundamentals chapter, Publisher vs. Self-Publishing chapter)

4 Choosing tools and creating plans

After the game developer understands the fundamentals of video games and game marketing, they can apply gained knowledge into practice. It is important to brainstorm for branding by choosing the profile picture and a nickname. After, the developer needs to choose the tools and social media. Followed by creating separate plans for game development and marketing, the developer can start working on the project.

4.1 Social media and branding

The accounts on such social media platforms as Bluesky, YouTube, and TikTok were created for marketing. As an additional tool, the account on Linktree was created to keep all social media links in one place available for the audience. It was decided to create a brand based on the nickname instead of the full name to preserve anonymity. The nickname "CoCherry" was created by combining words "coconut" and "cherry".

Regarding the profile picture, the inspiration was taken from the previous game as the first game that was created and published. The theme of the game was ghosts, thus the drawing of a ghost in a similar art style was created. Additionally, the social media websites provide users with an opportunity of adding a banner to their profile page. Therefore, it was decided to have a nickname to be written as a part of the banner, to make the profile page look appealing, and thus creating better visibility and branding.

4.2 Softwares

Since it was decided to create a 2D game, it has been decided to use Godot version 4.4. Along with that, GitHub will be used for version control and as an aid to the developer's

dairy. As a drawing software, Krita will be used to create sprites, art, and animations for both the game and marketing, since the author has experience with creating art and animations in this software. As for the sound design, Bfxr and Audacity softwares will be used. To create video content, OBS Studio will be used to record the screen, and edit the videos in DaVinci Resolve.

4.3 Websites

Bluesky Hub will be used to analyse engagement data from Bluesky to later import into Microsoft Excel. Since Bluesky has a feature of adding alternative text to the image and video content for accessibility, it will be added for all published media on this social media platform. It was decided to use Itch.io due to the absence of publishing fees. Due to the fact that Itch.io does not require any payment to publish the game, it was decided to use this publishing platform.

BeepBox.co and Freesound will be used work with sound effects and music creation. The sounds that have a license of Creative Commons 0 will be used, since it indicates that the creators of the sound allow people to use and modify the sounds for commercial purposes. To assist with the development of the game, Kanban Board was created using Microsoft Planner with the university's Microsoft account.

4.4 Plans

It was decided to create separate plan for both game development and marketing. Since the timeline of the thesis is approximately 6 weeks, the plans were created accordingly. The plans are describing first 5 weeks of the development process. The sixth week will be dedicated to the data analysis.

4.4.1 Marketing plan

Dreskin (2015, Marketing Materials chapter, Marketing Assets: Key Goals paragraph) suggests creating goals related to the monetization of the game. Therefore, he recommends making plans for such aspects as mail subscriptions, receiving a large number of preorders, and selling the game. However, since the game has been decided to be free of charge, the marketing goal was created instead.

In her e-book, Evans (2010, Goals need to be defined chapter) states that the goal of the marketing strategy can represent itself a certain number of followers or engagements, a number of created posts, receiving feedback related to content, or creating connections. The marketing goal is to have 10 people play the game after the release of the game. Therefore, the corresponding marketing strategy has to be created in order to spread awareness of the game. According to Funk's (2015, Platforms chapter) advice, hashtags will be used in posts. After conducting research on popular hashtags, it was decided to use such hashtags as "godot", "godotengine", "gamedev", "indiegame", "indiedev", "solodev" will be used for game-related content. For digital illustrations, hashtags "art" and "illustration" will be utilized.

It was decided not to announce the development of the game, since the author does not have a large number of followers. In addition, the first week was skipped. In other words, by not dedicating first week to game marketing, the attention will be focused on creation of game-related content and the development of the game itself. Bluesky was chosen to be a main posting platform due to the availability to create posts with different media content.

During the second week of the thesis, the screenshots showcasing main gameplay mechanics will be published. The posts of game developers will be interacted with. For the third week, screenshots of additional features will be created and published. Additionally, short gameplay videos will be uploaded. The fourth week will be dedicated to the game development. During the fifth week, the gameplay trailer will be created and published on YouTube and TikTok. The game will be released the following day. After that, the game-related content such as concept art of the characters and illustrations will be included in posts.

4.4.2 Game development plan

After the creation of the marketing plan, it was decided to work on the game development plan. Kanban Board in Microsoft Planner will be created to aid with the game development and organisation of the project. The pre-made template was used and later modified in other to be suitable for the project.

It was decided to dedicate the first week of the thesis to brainstorming the ideas, developing the game prototype, and establishing the appearance and the gameplay. The second week was planned to improve the in-game sprites and add the story-related features such as cutscenes and flashbacks. The main goal was to improve the visual

aspect of the game. During the third week the sounds and the music will be added to the game. Additional features will be added to the game. As mentioned previously, the fourth week will be dedicated to the error fixes and testing, in order to and create engaging gameplay experience. Throughout the development process, the game will be tested multiple times by the author of the thesis in order to find the mistakes and learn where the game can be improved.

5 Development and marketing

As a result of creating the plans, the developer can be considered to be ready to start the pre-production phase of game development. Therefore, the developer will choose the target audience, brainstorm for the game idea, and prepare the concept. For the game, it was decided to add a story to the game to interest the player.

5.1 Game idea

As the theory suggests, it is recommended to define the target audience. Since it was suggested to focus on the interest of the people who would play the game, it was decided to create a game for people who do not want to spend much time playing games. Therefore, it was decided to create a game that has approximately 15 minutes of gameplay time. Moreover, it was decided to create a 2D action game with simple mechanics.

The main concept of the game was based on the gameplay loop of controlling an insect character with a mouse cursor, thus by moving the mouse, the character will move along with the cursor. Initially, the design was focused on the idea of beetle eating the flowers that would appear in different locations on screen. However, it was later decided to make the beetle collect the pollen from the flowers. The idea was considered to be appealing to the players due to the simple game mechanics.

For the purpose of adding a challenge to the game, carnivorous plants are to be added. By choosing carnivorous plants, the player would lose a unit of health. Therefore, the player would need to choose the correct flower in order to play for as long as the beetle is alive. If the player loses all health units, the game over screen would be shown with an option of restarting the level. To add a purpose to the game, the score system would be added.

To follow the concept, the story of the game was created. The story follows a beetle that was adopted and raised by an elderly bee. The beetle had challenges living with the bees, hence the queen bee gave an order to collect the pollen from the flowers. Moreover, the medium of comics will be utilized through the cutscenes. Thus, the illustrations will be created to tell a story to the player. It was decided for the game to have a title “Beetle Bee.”

5.2 Game development

After the creation of the marketing and game development plans, Kanban Board was created in Microsoft Planner. The template was used and later edited in order to be suitable for the project. The sections “backlog”, “to-do”, “in progress”, “done” were created. The tasks were divided into subsections such as “marketing”, “game”, “gameplay”, “bug”, “graphics”, “sound/music”, “story” using the feature of labels. The tasks were assigned a priority level, namely “low”, “medium”, “important”, and “urgent”. The due date was added to the tasks for better organization.

Next step was to create a project in Godot, as well as to create a repository of the folder in GitHub. Subsequently, the work of the prototype of the playable character started. A scene called “insect” was added, and the script was attached to implement the behaviour logic. As the root of the script, Character2D was selected in order to control the scene. The code that can be seen in Program code 1 was added into the script for the character in order for it to follow the mouse cursor. Following the methodology of grayboxing, the main focus was the functionality and not the appearance. Hence, the icon of the Godot’s logo was used as a placeholder for the character’s sprite.

Program code 1. Character movement

```

@export var max_speed = 760.0
@export var min_speed = 50.0
@export var lerp_factor = 0.2 # Smooth factor, how smooth should the character move
@export var cursor_threshold = 5.0 # How close can the character move to the cursor

func _physics_process(delta: float) -> void:
    var target_position = get_viewport().get_mouse_position()
    var distance = global_position.distance_to(target_position)

    # Prevent character shaking when it is close to the cursor
    if distance < cursor_threshold:
        velocity = velocity.lerp(Vector2.ZERO, 0.2)
    else:
        # Smooth movement
        # Ensuring the character does not shake when close to cursor
        var speed = lerp(min_speed, max_speed, distance / 50.0)
        speed = clamp(speed, min_speed, max_speed)

        var direction = (target_position - global_position).normalized()
        var target_velocity = direction * speed
        velocity = velocity.lerp(target_velocity, lerp_factor)

    move_and_slide() # Making CharacterBody2D to respond to the movement
    sprite.look_at(target_position) # The sprite rotates towards the player's mouse cursor

```

After that the SpawnArea was created for flowers to appear in. As the base of the scene, the Control node was used in order to later arrange the SpawnArea in the middle of the player's screen. The next step was to create a flower and a feature of the flower appearing on the screen only where the SpawnArea is. Additionally, it was decided to make the flower disappear after a short duration of time if the player does not interact with it. Thus, creating challenge for the player.

A node with a Character2D was set as a root of the scene in order to control the interaction between the player and the flower. The DisappearTimer was added and a signal of "timeout()" was connected to the root of the flower's scene node in order to implement a mechanic of the flower disappearing after some time. To prevent the character from moving while it is interacting with the flower, a boolean was added to control the character's behavior. A simple placeholder image of the flower was created in order to create a playable prototype of the game.

The logic of detecting the flower's collision area was added in the playable character's code in order to detect the scenes in a "flowers" group, and the flower ID is recorded by utilizing the Node2D variable "normal_flower_to_eat". To interact with the flower, the player has to

be inside the Area2D of the flower and press the left mouse button, resulting in the progress bar appearing. The progress bar node was created as a separate scene and then instantiated to the scene of the playable character. The logic of progress bar appearing only when the character is interacting with the flower was implemented. Only after the progress bar reaches to the end, does the flower disappear, and the progress bar's visibility changes to being hidden. In the beginning of the project, the progress bar was filling automatically by itself, but it was later decided to add a feature of pressing the left mouse button multiple times in order to fill the progress bar, making the gameplay more engaging. In order to prevent the mistakes of detecting the correct flower, the logic of deleting the detected flower's Area2D by leaving the area was implemented using "body_exited" signal.

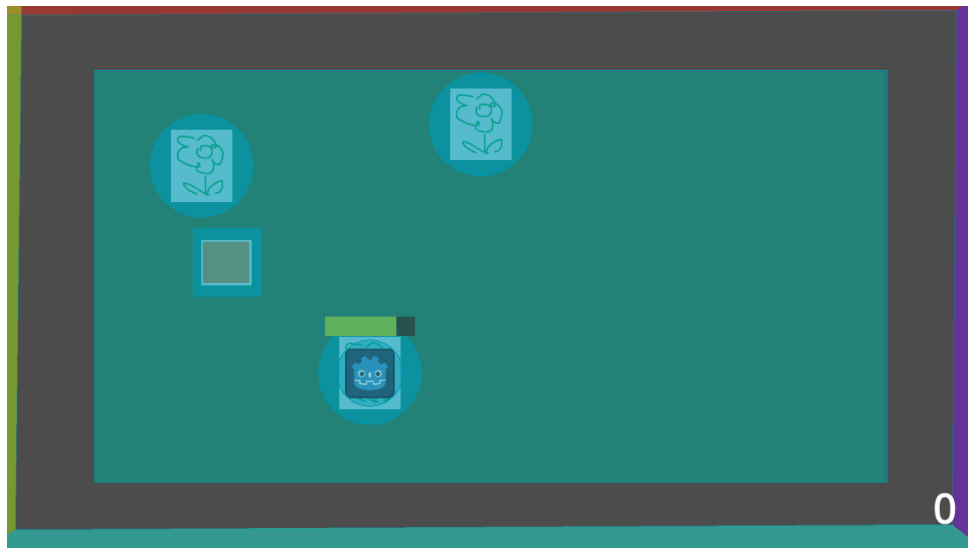
After that, the scene of the level was created and called "main" with a Node2D as a root. The timer Flower_Spawn_Timer with the properties of One Shot and Autostart turned on for the flowers to spawn. The timer's signal "timeout()" was connected to the root of the scene "main" to create a logic of the flower disappearing after the timer has timed out. The node of the player and the spawn area were added by instantiating them in the main scene. In order to spawn the flower when the Flower_Spawn_Timer times out, the reference of the flower's file was instantiated in the script of the main scene. "flower_lits" array that holds a reference of the flower's instance was created to store the list of flower scenes.

In addition, each created flower is added to the group "flowers" using code for the player to detect. The feature of the flowers spawning in different places was implemented to make the game more engaging. That resulted in a simple playable version of the game.

Then, the scene was added for the power-ups to not only aid the player, but also make the gameplay versatile and engaging. Orange square was used as a placeholder. Similar to the logic of the flowers, the PowerUp_Spawn_Timer was created and then the similar code was used to make the power-ups appear on screen.

To keep the player engaged, the score system was added by creating a new Control scene with a label. Global script was added to the project and was set as Autoload for it to work on every scene, in order to add the logic of saving the score into the "personal best" variable. The prototype of the game with the grayboxing method can be seen in Figure 2.

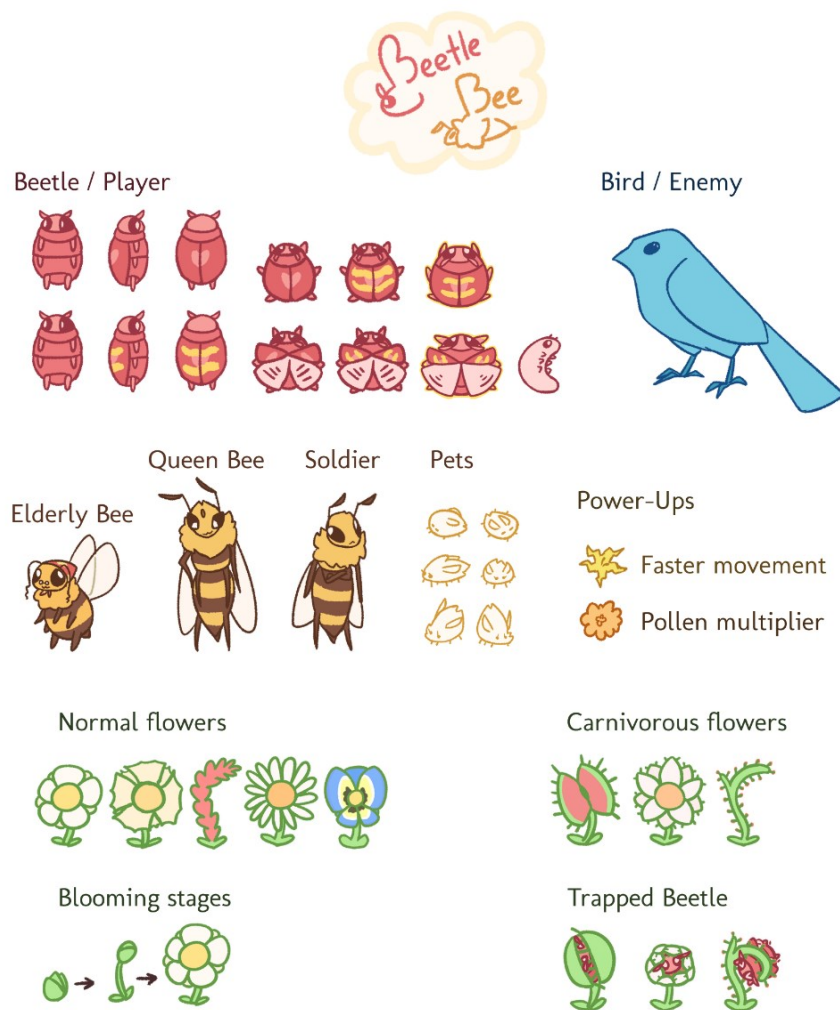
Figure 2. The graybox prototype of the game



The collision areas are set to be visible in order to see where the player can and cannot interact, as well as to see where the flowers and power-ups can appear.

When the mechanics were implemented, it was decided to work on the visual aspect of the game, and therefore the concept art of the game was created and illustrated in Figure 3. After that, separate sprites were created for the flowers in 2 stages, when the flower appears on screen, and when it has almost bloomed that was changed by using the timer node. In order for the flowers to depict the stages, `AnimatedSprite2D` was added to control the sprites with 2 animations: bloomed and blooming. While the flower is blooming, it has 2 frames for the first and the second stage, and when it has bloomed, it depicts the third stage of the flower that the player can interact with. The sprites were created in Krita by using the animation feature, the frames were created and exported as separate images.

Figure 3. Concept art



The logic of the player not being able to interact with the flowers while they are in their first 2 stages was implemented. Since the flowers are still blooming, in order to create a mechanic of the player waiting for the flowers to bloom. `AnimaitonPlayer` was added to add a bounce effect to the flowers, making the flowers play a short animation of the sprite's scale changing after the flowers enters the next stage of blooming. For the third stage, different variations of flowers were created for variety.

The flower scene was duplicated, resulting in 5 regular and 3 carnivorous flowers. Subsequently, the logic of the carnivorous plants was added, hence whenever the player interacts with the flower in the group "carnivorous", the player gets trapped in it. The `Node2D` variable "carnivorous_flower_to_eat" was created to create similar to the normal flower's logic of detecting the right flower, along with the logic of removing the flower from the variable after the beetle character leaves the `Area2D` of the carnivorous flower.

The playable character's appearance was updated by adding AnimatedSprite2D with the animations of the character flying, sitting on the flower, and an animation that plays when the character's health reaches 0.

The playable character's appearance is updated by Adding AnimatedSprite2D with 3 animations: the character flying, sitting on the flower, death animation. The animation of the character flying plays after the level has loaded, and after the player stops interacting with the flower. The animation of the character sitting plays when the user interacts with the flower. Thus, making the character seem as if it has landed on top of the flower and is collecting the pollen. Death animation plays when the character reaches 0 health units. In addition to that, it was decided to add a logic of the character moving to the centre of the flower when the player interacts with it, in order to visualize that the insect is near the pollen and not on the side of the flower. Sprites of the character are visible in Figure 4.

Figure 4. Player character sprites



The Progress bar node along with its timer were duplicated and modified to serve as a TrappedBar with its timer, making the character stop moving, as well as the player wait for 3 seconds as a form of punishment for choosing the carnivorous flower. The visuals for the progress bars were updated. It was decided to make different sprites for the progress bars to create better experience, as seen in Figure 5.

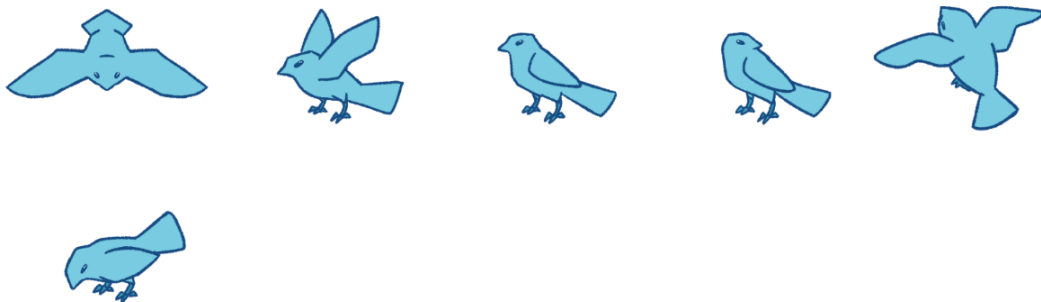
Figure 5. Updated progress bars



The bounce animation was added to create an illusion of the character trying to get out of the flower. When the timer's time is out, the character loses 1 unit of health. The carnivorous flowers were instantiated using a similar logic to the normal flowers and were saved in a separate list. The random floating-point value (randf) was added in order to create a feature of randomly choosing whether the normal or carnivorous flower would appear on the screen. To prevent the errors from happening, the limit of the number of existing flowers was created, thus preventing the flowers from appearing if the number reaches 10.

To make the game slightly more challenging, it was decided to add a bird enemy into the game. At first, the logic of following the player character was implemented for the Sprite2D of the enemy – the shadow. After that, the AnimatedSprite2D was added for the bird itself. The bird was designed to have 3 animations: normal, caught, not_caught. The normal animation plays when the bird is landing on the ground. Caught animation plays when the player is near the bird, and the not_caught plays when the player is not detected, thus the bird flies away. Two AnimationPlayers were created. The first was used for the shadow to scale up when the bird lands and scales down when the bird flies away. The second animation player is used to create a similar bounce animation to the flowers, making the sprite bounce when the sprite of AnimatedSprite2D changes by using the "frame_changed()" signal. The bird sprites are depicted in Figure 6.

Figure 6. Enemy sprites



One of the most important mechanics was the health system. To implement this feature, a scene for the heart was created with the TextureRect as a root of the scene and AnimatedSprite2D as a child. The sprite has an animation with 2 frames of the full and empty heart, as seen in the Figure 7.

Figure 7. Health bar

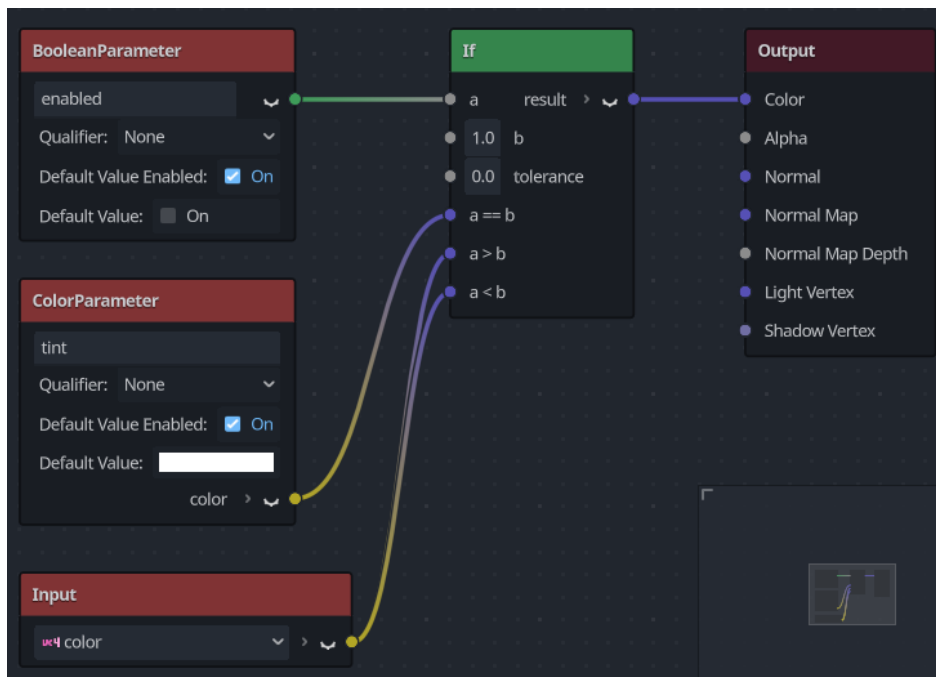


The TextureRect has a script attached with the logic of updating the sprites depending on whether the sprites are whole or empty. Then, the hearts_container scene was added to store the hearts. The script was added to update the hearts. Hearts_container scene was later instantiated to the main scene with the logic of calling the functions from the hearts_container in the level script. Later, it was decided to make the player lose 1 health unit after the character is released from the carnivorous plant, and when the character flies near the bird. Due to the simple game mechanics, healing objects were not added to the game. The VulnerableTimer was added to make the character not lose all their health at once, allowing the player to be invincible for 2 seconds after they are hit.

The Events script was created to store the global signals, and it was added to the Autoloads. The variables for the health were added into the playable character's script, with the booleans to control whether the character is alive or not. The logic of preventing the character from following the cursor when the character is not alive was added.

To indicate that the character is hit, an indicator was added. Thus, the sprite flashes white shortly after losing a health unit. The feature was implemented by creating a shader demonstrated in Figure 8 to the AnimatedSprite2D of the character, thus the sprite changes to white if the shader is enabled, and a HitFlashAnimationPlayer to regulate the speed of the flash animation, as well as the "enabled" boolean of the shader.

Figure 8. Hit indicator shader



To implement the story to the game, the features of the character shaking when the bird appears, and entering the next level were implemented. To create a shaking effect of the character, Vector2 variable was created that calculated the random coordinates. Depending on the shake_strength float, the character moved slightly to the sides.

To allow the player to proceed to the next level, the score mechanic was modified by adding the progress bar and hiding the score. Hence, the player has to completely fill the progress bar in order to complete the current level. After that, to the main level the Area2D with a child of CollisionShape2D nodes were added to the left part of the screen. By entering the area, the player can continue to the next level only if they have filled the level's pollen progress bar shown in Figure 9.

Figure 9. Empty and full pollen progress bar



The scene for the arrow was added to indicate that the player needs to move to the Area2D in order to navigate to the next level. Slow panning animation was added to capture

players' attention. The arrow appears after the player collects the required amount of pollen. The arrow is depicted in Figure 10.

Figure 10. Arrow navigating to the next level



The main level was duplicated and renamed to create 4 levels of the game. The main level was later served with the purpose of a level for the infinite mode that would get unlocked once the player completes the 4 levels of the game. For the infinite mode, instead of the progress bar of the pollen, the score is shown and then saved in the Global script to show the player their personal best score. This was possible by creating a Global integer variable "current_scene_name" and then modified depending on the current level by changing the variable in the scripts of the levels.

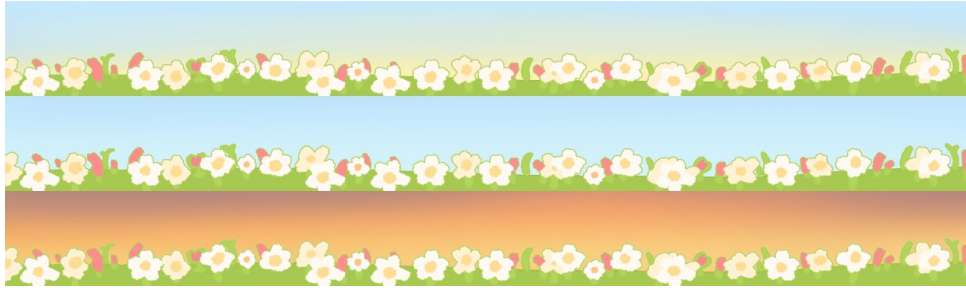
It was decided to continue working on the power-ups, which are the speed boost and the score multiplier. By adding the mechanic of the states with the global signals and the state_machine script, the current states can be regulated. Therefore, when the player enters the area of the scene that is in the group "powerup", depending on the power-up type whether it is "speed" or "pollen", the player character changes the state. The timers were added for each power-up in the character's script in order for the power-ups not to be infinite. Figure 11 demonstrates the icons that indicate how much time the power-ups have left, as well as the icons of the power-ups. The logic of the power-ups was implemented with a feature of being in multiple states at once, therefore the player can have both speed boost and score multiplier equipped.

Figure 11. Equipped power-ups



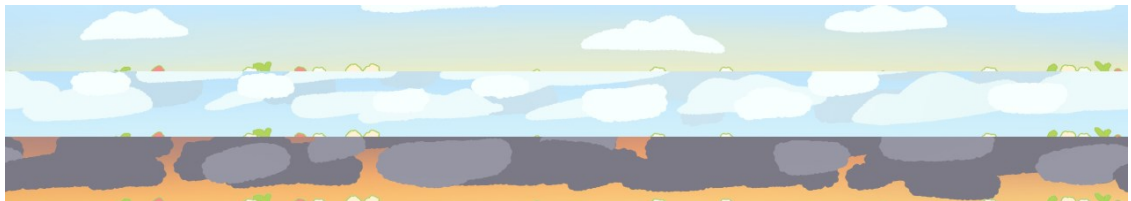
The next step was to work on the environment of the game. By creating background images of different times of the day shown in Figure 12, the player can differentiate between the levels. The backgrounds are also important to serve as a plot device for the story.

Figure 12. Sky variations for morning, noon, evening



It was decided to add the moving clouds to the levels for not only the better visuals, but also for the story of the game. Depending on the `current_scene_name`, the `AnimatedSprite2D` of the sky changes, as well as different clouds textures become visible. To the clouds, a shader was added to create an illusion of the clouds moving infinitely to the left. The clouds with the corresponding sky images are visible in Figure 13.

Figure 13. Cloud sprites



The `GPUParticles2D` node was created for the third level in order to implement the feature of the rain when the player collects the needed amount of pollen. Along with that, the logic of detecting the size of the viewport was implemented to prevent the playable character from moving outside of the game's screen and breaking the immersion of the game. The updated version of the gameplay can be seen in Figure 14.

Figure 14. Updated version of the gameplay



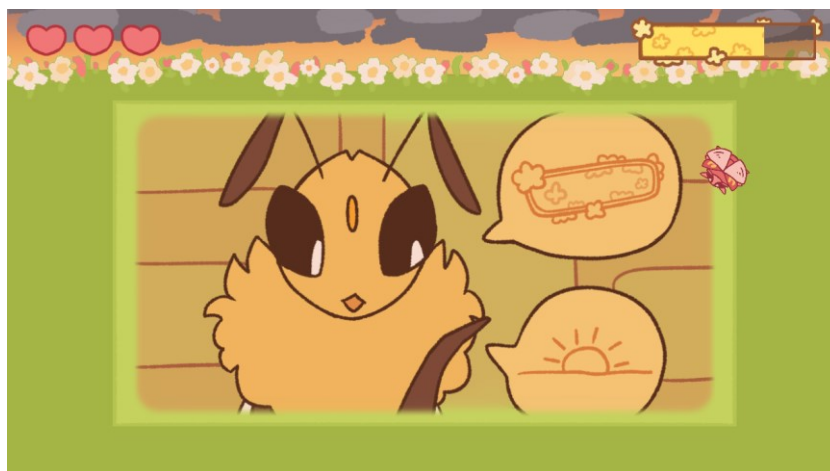
Some scripts require time to process the “current_scene_name” variable. Hence, it is required to insert a line of code from Program code 2. This allows script to wait for a process frame before implementing the code.

Program code 2. Wait a process frame

```
await get_tree().process_frame
```

To continue working on the story, the cutscenes were implemented as flashcards depicted in Figure 15. By collecting pollen from a special blue flower that appears only when the bird spawns, the game pauses and the flashcards appear to demonstrate the story. The player can navigate to the next flashcard by pressing the left mouse button.

Figure 15. Story flashback



For the purposes of the story, the new sprites of the playable character were added of the beetle without the paint on its back. Hence the feature of changing to the new sprite was implemented once the player completes the level 3. The sprites look similar to the previous versions, except for the absence of the yellow paint on the back. The AnimatedSprite2D was added with new sprites with the hidden visibility, and once the player completes level 3, it becomes visible. The previous AnimatedSprite2D of beetle with the paint becomes hidden. The sprite comparison is demonstrated in Figure 16.

Figure 16. Beetle's sprite with and without yellow paint on its back



After that, the work on the menus started. The scenes were created for the title screen, level selection, game won, game over, and the pause screen. Title screen was created with the buttons to start and exit the game. The background of the main menu depicts the gameplay as a drawing, with the title of the game, the beetle, and the bird sprites being added. The slow rotation from side to side was implemented for the sprites to create a more interesting design. Title screen is shown in Figure 17.

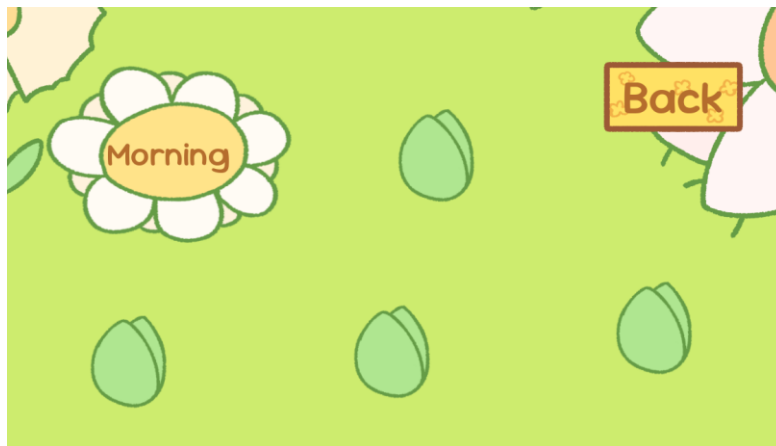
Figure 17. Title screen



The level selection menu was originally created to be a separate scene that the player would navigate to after pressing the “Play” button, but it was later decided to adapt a creative approach by implementing the level selection in the title screen. The logic of the player’s camera was implemented to move down using AnimationPlayer, depicting the continuation of the background with the buttons of the levels. The screen has a logic of disabling the buttons of the next levels unless the player has unlocked them by playing the game. For instance, when the player completes level 1, it changes the variable inside the Global script “unlocked_levels” to 2, unless the level is already unlocked, preventing the player from locking the next levels if they have already unlocked them.

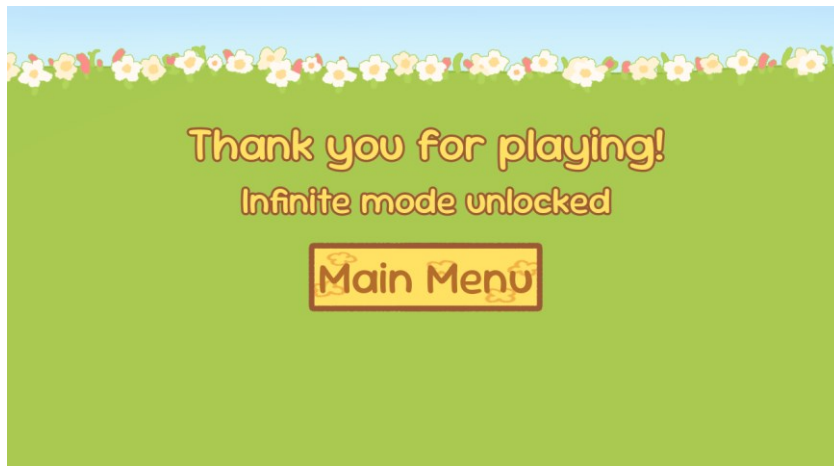
The designs on the buttons in the level selection screen were made to resemble a flower, instead of the normal buttons. It was made to not only create better visuals, but also to subconsciously prepare the player to click on the flowers during the gameplay. To follow the theme of the flower design, the disabled flowers were implemented to look similarly to the flower bud. The locked levels have a logic of not depicting the name of the level if the level is disabled, as demonstrated in Figure 18.

Figure 18. Level selection screen



When the player completes all story mode levels, they will receive a victory screen. It was implemented by creating a new scene with the intention of being navigated to once the player has completed the four levels of the game. The screen has a caption of thanking the player for playing the game and unlocking the infinite mode. Victory screen is demonstrated in Figure 19.

Figure 19. Victory screen



Game over menu was created and added to each level with original property of being hidden, and it is shown when the player character's health reaches 0. The "Retry" button reloads the current level scene. Correspondingly, "Main Menu" button navigates the player to the title screen. Game over screen is visible in Figure 20.

Figure 20. Game over screen



The scene for the pause menu serves a function to pause the game when the player presses the Escape key on the keyboard. The "Resume" and "Main Menu" buttons were added to the pause screen, and when the player presses the Escape key, if the game is not paused the game pauses, and if the player presses the key again, if the game is pause, it resumes the game. It is noteworthy that the scene of the pause menu has to have the

Process mode property to be set to Always in order for the player to be able to interact with the buttons in the pause screen.

To the first level, an interactive tutorial depicted in Figure 21 was created to teach the player the game mechanics. By creating a sprite of the mouse and using AnimationPlayer, animations of the mouse moving were created to demonstrate that the player can move the character by using a mouse. The scene for one of the flowers was duplicated and modified in order to be added as the first flower for the tutorial. Due to the fact that the carnivorous plants are depicted with spikes that indicate danger, it was decided not to teach the player about carnivorous plants to make them learn by themselves.

Figure 21. Interactive tutorial



The sprite of the left mouse button was added to show that the player can interact with the flower by pressing the button. After the player collects the pollen from the tutorial flower, the progress bar of the pollen appears, and the ordinary gameplay starts with the main objective of filling the bar. It was decided to not add the enemy bird to the first level for the player to learn not only the movement of the game, but also the different types of flowers.

It was decided to add a feature of outline appearing on the character's sprite as the player fills the pollen progress bar of the level. At first, the shader was used to create an outline. However, the solution of shader appearing and disappearing was not found, and the outline's appearance not being visually appealing. Therefore, duplications of the sprites were created, and the outline was manually added using Krita.

The feature was implemented by creating additional animations in AnimatedSprite2D. Hence, as the player collects the pollen, the animation of the sprite with an outline depicted in Figure 22 starts playing. The outline was developed to be yellow to represent pollen. Minor detail was added to the sprite by the additional yellow colour to the beetle's legs indicating that the beetle was collecting pollen.

Figure 22. Playable character's outline



The comic panels were drawn with the purpose of telling the story to the player. A small part of the comic is depicted in Figure 23. The panels were added to the new scene “cutscene”. The animation of the camera moving to serve as an interactive comic was created.

Figure 23. Comic snippet



The player can move from panel to panel using the left mouse button. In the cutscene, the indicator was added to show the player that the next panel is ready to be shown. Using the Godot's feature of the animation player being able to add the markers into the timeline and play the animation from the first marker to the second, the logic of moving from panel to panel was implemented.

Since it is popular with the game developer community that some games have intro scenes before starting the game that depict the game engine, the studio, or the addons used in order to create a game. It was decided to create a custom animation that depicts the Godot logo getting turned into a modified logo with the features of the insect with a caption “Made with Godot”. The splash screen frame is depicted in Figure 24.

Figure 24. Custom “Made with Godot” splash screen



It is believed that the logo creates a positive first impression of the game. In order for this feature to be added, the frame-by-frame animation was created in Krita. After that, it was exported as “.ogv” file format, and a scene “made_with_godot” was created with a VideoStreamPlayer node for the video. In the project settings, the boot splash image was changed to the first frame of the animation of the back screen in order for the animation to work as intended.

After the main features had been developed, it was decided to work on additional minor features for the game. The probability of carnivorous plants appearing instead of the normal flower was changed. Moreover, the levels were designed to have different chances, increasing the difficulty of the gameplay with each level.

The animation player of the flower’s sprite was added for each flower that would slightly change the skew angle of the flower to create better visual effect of the flowers moving, creating an illusion of flowers moving due to the presence of wind. The additional feature of YSorting was added to the flowers, playable character, bird sprite, and shadow sprite in order to create an illusion of volume. Additionally, it prevents a visual mistake of the sprites overlapping on top of each other.

It was decided to add small details to the bird, which are the logic of enemy’s shadow to not appear on the sky, and to prevent the bird from landing on top of the sky texture. For the shadow, the visual detail was created through the updated ordering parameters of Z Index, in order to create layers. The logic implies that the ordering of the sprites adds an effect of the shadow appearing behind the sprite of the sky, thus making it not visible. As regards the landing logic, the feature to the collision was added to make the enemy receive a signal if it overlaps with the collision of the sky. The feature is depicted in Figure 25.

Figure 25. Additional shadow logic



The label for the personal best score was added and implemented in such a way that the label is initially hidden, unless the player completes the game and unlocks the infinite mode. As a detail, when the player completes the game, in the title screen the sprite of the beetle does not have the paint on its back, due to the in-game story. The feature for the infinite mode was added, making the bird enemy spawn more and more often as the player plays the level.

The behaviour of the progress bar of the normal flowers was modified, and instead of the flowers having the same number of clicks required to collect the pollen, the amount would be randomized with every flower, unaffected by the type of the flower. The range of the variable changes depending on the level, making the clicks vary from 3 to 5 on the first level, and 2 to 11 on the infinite mode level.

It was decided to implement the feature of the camera slightly zooming in when the player interact with the flower, creating more feedback. Hence, the more the player clicks, the closer the camera zooms. The limit was taken into consideration in order to prevent visual errors. When the player finishes the progress bar of the flower, the camera smoothly zooms out to the initial position. The logic of the zoom implementation is shown in Program code 3.

Program code 3. Zoom implementation

gameplay, game over, game completed were added as clips into the stream. Figure 27 demonstrates the implementation of the transitions that regulates the music order and transition type.

Figure 27. Audio transition editor

Loop	MainMenu	GameWon	GameOver	Silence	Any Clip	From / To
	✂ Immediate ...					Loop
→ Imme		→ Immediate ...	→ Immediate ...			MainMenu
→ Imme						GameWon
→ Imme						GameOver
✂ Imme						Silence
						Any Clip

After that, the mute music and sound buttons were implemented for better user experience. The buttons were added to the Title screen. It was decided to include the buttons to the pause screen for the player to be able to remove sounds and music in any moment. The game was exported as Windows Desktop and Web versions.

5.3 Game marketing

According to the plan, the first step was to interact with other people on the Bluesky by liking their posts and following the accounts. After that, humor was used to create short-term video content of the playable prototype of the game and posted on Bluesky. After that, humor was used in an image format with the beetle character depicting in it. The second week was dedicated to creating content containing humor. Short-term video content of the playable prototype of the game was created and posted on Bluesky. The post was followed by an illustration depicting the beetle character.

During the third week, users on Bluesky tend to create posts that contain the logo of the game engine they were using for their game and included 3 images of the game itself. It was decided to create a similar post including a previously created custom-made logo of Godot as a single frame. In addition, images depicting the gameplay were added. During the course of the fourth week, a short video was published on Bluesky that demonstrates the title screen along with the level selection screen that has an animation mentioned in the implementation of the gameplay. The Itch.io page of the project was created as a draft, making it unavailable publicly in order to have the page prepared in advance for the game

release day. Video with humor was created and posted on TikTok using other user's sound.

It was decided to create a post with a custom-made splash screen animation of Godot logo on YouTube, Bluesky, TikTok. Then, a video using short-term video format with humor was created and uploaded on TikTok. Promotional illustration was drawn and published on Bluesky. Subsequently, the gameplay trailer was created. The game trailer was released on the fifth week. The following day, the game was published. After the release, the post was created on Bluesky informing the followers about the game release. After that, the post with the concept art was created.

6 Results

After the release of the game, the author was contacted with an opportunity of collaboration by a company. They suggested adding the game to their platform. Upon thoughtful consideration, the offer was politely declined. Even though the game might get a chance of having more players, the upcoming traffic would be from a third party service. Therefore, it will not be possible to analyse the data.

According to the data from Linktree, 14 people used the service by opening the author's link, 2 people opened Itch.io and YouTube links. Bluesky and TikTok links were not used. The click rate of the links in Linktree was 29%.

At the beginning of the thesis, the account on Bluesky had 4 followers, YouTube account had 3 subscribers, and TikTok account had 0 followers. By the end of the sixth week, the Bluesky account has gained 24 followers. The number of followers on YouTube and TikTok stayed the same as 3 subscribers and 0 followers. By analysing the Figure 28, it can be seen that the number of followers share a parallel trend with the number of posts. Therefore, it can be conjectured that the followers were gained as the posts were created.

Figure 28. Chart of the followers gained on Bluesky

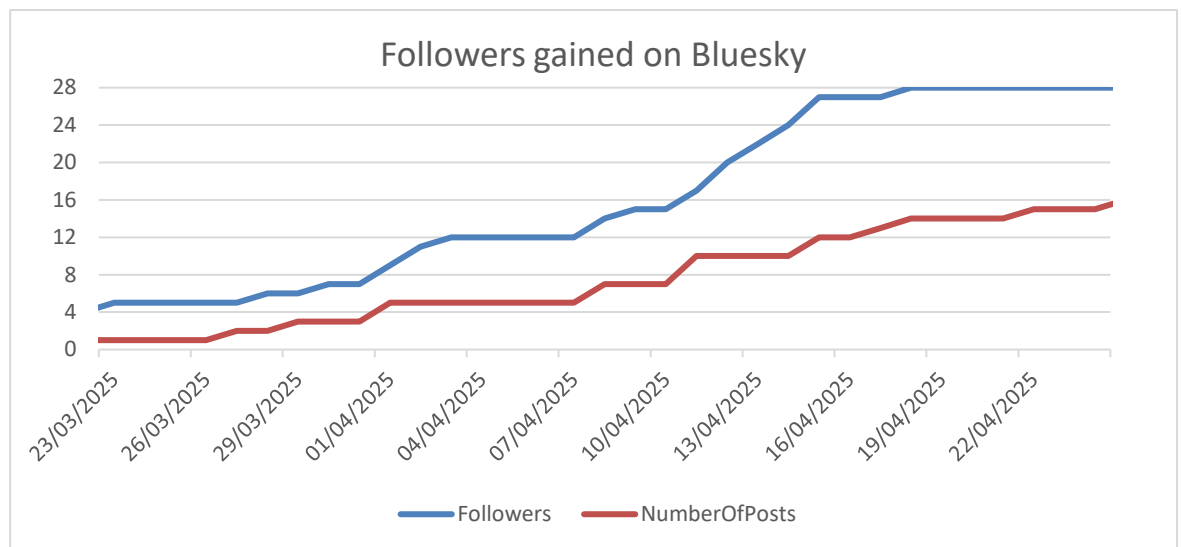


Chart from the Figure 29 demonstrates that people on Bluesky interacted with the posts that had videos attached to them the most. However, it can be speculated that this type of content received most engagement due to the video demonstrating a custom-made Godot splash screen. The post received 79 likes, 10 reposts and 3 comments. It is believed that the post was popular with the game developers that use Godot.

Figure 29. Chart of the total sum of likes by content of the post on Bluesky

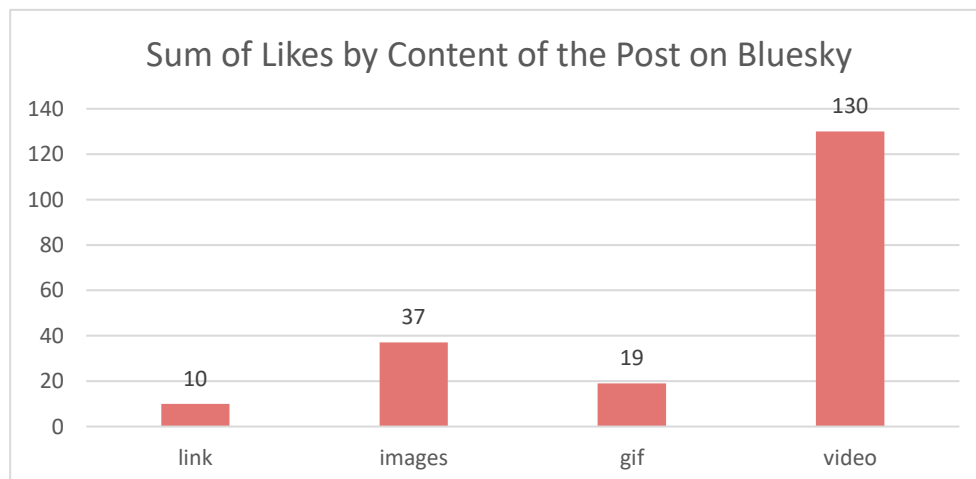
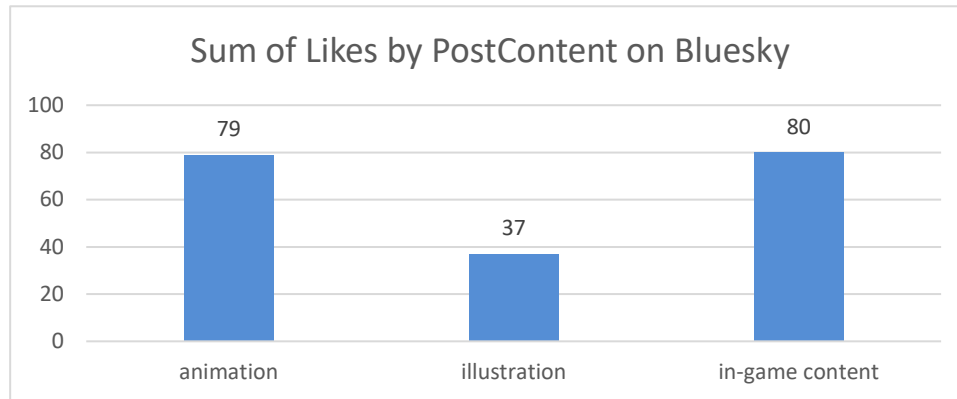


Figure 30 represents the type of attached media used in the post with the sum of likes. It can be seen that in-game content, such as screenshots of the gameplay and game updates received the most attention similarly to the post of the splash screen animation. Additionally, it was noticed that people were interested in the content related to the game

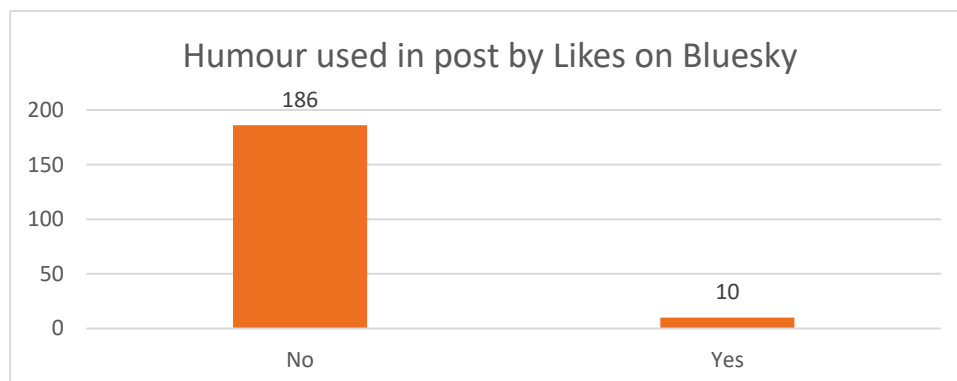
engine and the progress of game development more, compared to the art pieces created to promote the game.

Figure 30. Chart of the sum of likes by the content of the post on Bluesky



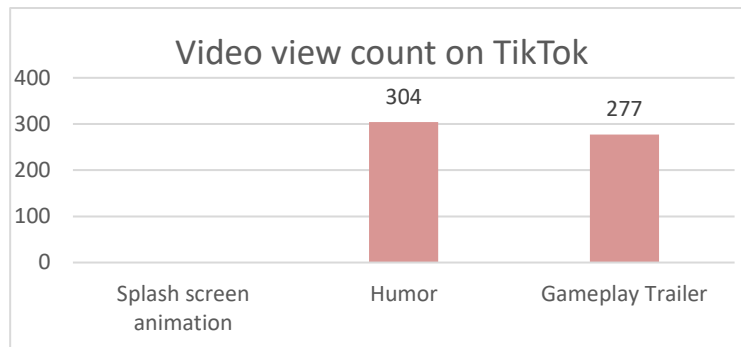
As depicted in Figure 31, users on Bluesky engaged with the post that did not include humour. However, the reason for that can be speculated due to the fact that both the animation and in-game related content did not use humour. Similarly, the posts that include humour were created in the beginning of the thesis implementation, while the account had small number of followers.

Figure 31. Chart whether the humour was used in post on Bluesky



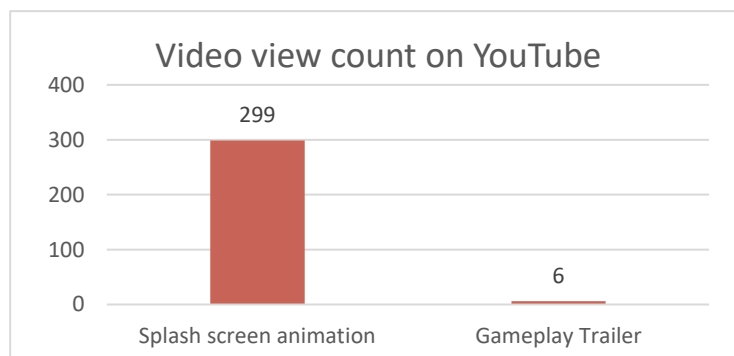
As the data from Figure 32 suggests, content on TikTok did not receive much attention. There were in total 3 videos uploaded on this platform. The first video that was posted contains the animation of the splash screen, and it received 0 views. It can also be seen that video containing humour received slightly more views than the gameplay trailer.

Figure 32. Video view count on TikTok



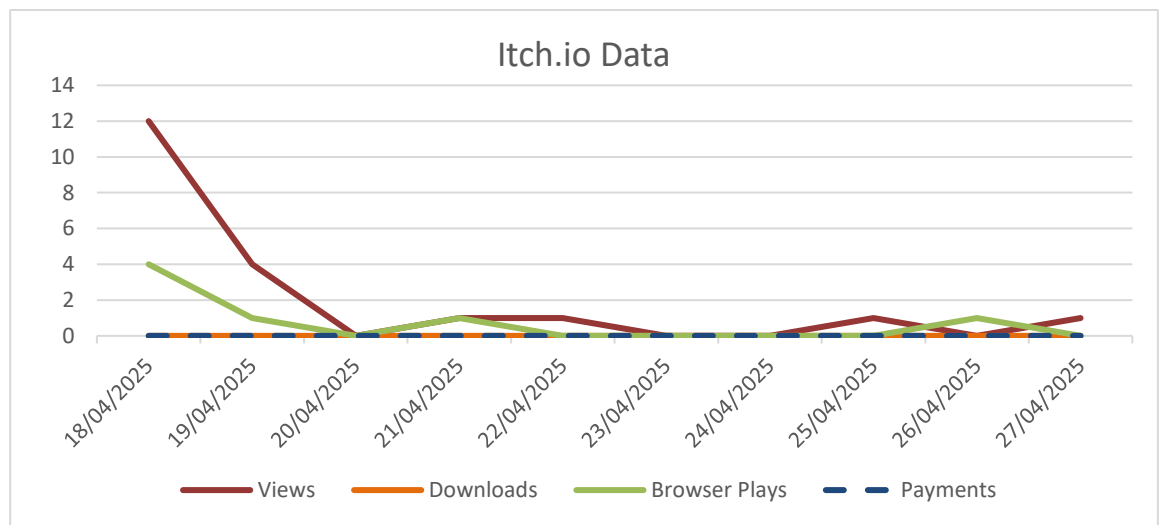
Similarly, YouTube data from Figure 33 demonstrates that the video content received low view rates. 2 videos were uploaded, which are the splash screen animation and the gameplay trailer. As opposed to the data from TikTok, on YouTube the animation received more attention than the gameplay trailer.

Figure 33. Video view count on YouTube



In accordance with Figure 34, the greatest number of people viewed the page of the game and played it during its release day. Throughout the week, the number decreased significantly. Even though the post-release stage was used to market the game, it was not as successful as pre-release stages of game development. Since the number of downloads is 0, it can be speculated that due to the fact that the game can be played in browser, the players decided not to download the game. The players did not use a feature of voluntary donations for the game.

Figure 34. Game data from Itch.io



The goal of having 10 people play the game was not met. It can be speculated that there are multiple factors for the reason of the game's low visibility rate. One of the reasons could be that the project is a small game. Compared to the previously mentioned game developers from the chapter 3.1.4, who have created bigger-scale projects, it can be speculated that people are not as interested in short games.

The other factor could be that the game has a browser version. Thus, the game allows the players to save time from downloading the game. Therefore, it can be speculated that players have spent a small amount of time playing. In addition, it can be assumed that due to the fact that the author is a beginner game developer with a small number of followers, the game did not receive much attention.

While according to the theory in chapter 3.1.3 it is recommended that the game should be marketed more often closer to the release date, the main attention was focused on the game development to work on the bug fixes. Since it was decided to spend the first and fourth weeks of the thesis implementation on the game development and not marketing, this could have affected the results.

7 Summary

To answer the first question, by publishing a free simple game indie developer can not only learn to design a variety of game mechanics, but also increase their number of followers on social media. Additionally, the developer expands portfolio that can be demonstrated online. It can be suggested that the beginner game developer can work on short games to gain followers and focus on creating longer games that they can benefit from by growing as an indie game developer.

As for the second question, game developers can encounter several challenges. One of them concerns the fact that a novice developer is not known enough on social media to have their game gain attention. Therefore, due to the small number of followers, the game will be played by a small number of people correspondingly. Another challenge is the fact that the developer needs to spend time creating marketing strategies. In addition to that, they are required to plan the content that they will post.

In regard to the third question, through analysis it can be seen that people create most engagements with the post that have main topic of in-game content and videos. The least engagement was created with the posts containing illustrations and humour. Moreover, social media users in game development community interacted with the posts regarding game engines. However, the developer should consider the fact that on various platforms, the content will be received differently.

The author of the thesis learned what makes a game fun, how to create certain game mechanics, how to use social media for marketing, as well as understood the benefits of creating a free indie game, and the challenges of self-publishing a video game. It was learnt that social media could have been utilized more through marketing strategy in order to raise awareness of the game.

Future improvements of the game could include adapting the video game for smartphones, adding more power-ups, or implementing challenge levels. In addition, a feature of switching to other languages can be added to make the game more accessible. Regarding social media, such platforms as YouTube and TikTok could be utilized more by creating a plan for video content. In regard to the thesis, it can be suggested that the received data might not be enough to accurately answer the question regarding what media content receives most engagements. Further research can be conducted with a larger number of users.

References

- Aseprite. (n.d.). *Animated sprite editor & pixel art tool*. Aseprite. <https://www.aseprite.org/>
- Audacity. (n.d.). *Free Audio editor, recorder, music making and more!*. Audacity. <https://www.audacityteam.org/>
- Beepbox.co. (n.d.). *Beepbox.co*. Beepbox.co. <https://www.beepbox.co>
- Bluesky. (n.d.). *Bluesky*. Bluesky. <https://bsky.social/about>
- Bond, J. G. (2020). *Introduction to Game design, Prototyping, and Development*. O'Reilly Online Learning. <https://learning.oreilly.com/library/view/introduction-to-game/9780136619918/>
- Boyles, J. (2020, October 4). The Difference Between Casual Gamers and Hardcore Gamers *HackerNoon*. <https://hackernoon.com/the-difference-between-casual-gamers-and-hardcore-gamers-fk253trm>
- Bramble, R. (2023, May 10). What Are The Main Stages Of Game Development? *GameMaker*. <https://gamemaker.io/en/blog/stages-of-game-development>
- Cambridge Dictionary. (n.d.). *Video game*. Cambridge Dictionary. <https://dictionary.cambridge.org/dictionary/english/video-game>
- Carrd. (n.d.). *Carrd - Simple, free, fully responsive one-page sites for pretty much anything*. Carrd Inc. <https://carrd.co/>
- Cassese, E., Conroy, M., Duckert, L., Espinoza, C., Meyer, M., Reay, E., & Shackelford, K. (2023). *Changing the Narrative: Why Representation in Video Games Matters*. <https://doi.org/10.13140/RG.2.2.22806.36162>
- Christopherson, Kimberly M. (2007). *The positive and negative implications of anonymity in Internet social interactions: "On the Internet, Nobody Knows You're a Dog."* *Computers in Human Behavior*, 23(6), 3038–3056. <https://doi.org/10.1016/j.chb.2006.09.001>
- Csikszentmihalyi, M. (1990). *Flow The Psychology Of Optimal Experience*. Internet Archive. <https://archive.org/details/flow-the-psychology-of-optimal-experience-pdfdrive>
- Cult of the Lamb. [@cultofthelamb.bsky.social]. (n.d.). Bluesky Social. Retrieved April 4, 2025, from <https://bsky.app/profile/cultofthelamb.bsky.social>
- Davenport, M. (2024, October 1). *The History of Digital Drawing Software*. Ensemble. <https://www.ensemble.art/blog/the-history-of-digital-drawing-software>
- Dhiman, B. (2023). *Ethical Issues and Challenges in Social Media: A Current Scenario*. *Global Media Journal*. <https://www.globalmediajournal.com/open-access/ethical-issues-and-challenges-in-social-media-a-current-scenario.php?aid=92953>
- Dreskin, J. (2015). *A Practical Guide to Indie Game Marketing*. O'Reilly Online Learning. <https://learning.oreilly.com/library/view/a-practical-guide/9781317625506/xhtml/Ch03.xhtml>

- Evans, L. (2010). *Social Media Marketing: Strategies for Engaging in Facebook, Twitter & Other Social Media*. O'Reilly Online Learning. <https://learning.oreilly.com/library/view/social-media-marketing/9780768694956/>
- FL Studio. (n.d.). *Overview*. FL Studio. <https://www.image-line.com/fl-studio/>
- Freesound. (n.d.). *Freesound*. Freesound. <https://freesound.org/>
- Funk. (2013). *Advanced Social Media Marketing: How to Lead, Launch, and Manage a Successful Social Media Program*. O'Reilly Online Learning. <https://learning.oreilly.com/library/view/advanced-social-media/9781430244073/>
- Game Marketing Genie. (2022, June 30). *Social Media Posts: 10 Engaging Ways to Market Your Game*. Game Marketing Genie. <https://www.gamemarketinggenie.com/blog/engaging-social-media-posts-to-market-your-game>
- Gimp. (n.d.). *The GNU Image Manipulation Program: The Free and Open Source Image Editor*. Gimp. <https://www.gimp.org/>
- GitHub. (n.d.). *Build and ship software on a single, collaborative platform*. GitHub. <https://github.com/>
- Godot. (n.d.). *GScript reference*. Godot Engine Documentation. https://docs.godotengine.org/en/stable/tutorials/scripting/gdscript/tutorials/scripting/gdscript/gdscript_basics.html
- Hadzigeorgiou, Y., & Judson, G. (2017). Toward More Effective Storytelling for Raising Environmental Awareness in Young Students. *Journal of Advances in Education Research*, 2(1). <https://doi.org/10.22606/jaer.2017.21002>
- Hart, R. (2024, July 27). *Using your real name on social media? Here's why you should think twice*. TechRadar. <https://www.techradar.com/pro/vpn/using-your-real-name-on-social-media-heres-why-you-should-think-twice>
- Hollstrand, P. (2020) Supporting Pre-Production in Game Development [Master's thesis, KTH Royal Institute of Technology]. Diva. <https://www.diva-portal.org/smash/get/diva2:1433725/FULLTEXT01.pdf>
- Itch.io. (n.d.). *About itch.io*. itch.io <https://itch.io/docs/general/about>
- Jeswani, R. (2023). *The Role and Importance of Social Media Marketing in Brand Building*. ResearchGate. <https://doi.org/10.46759/IJJSR.2023.7401>
- Kamila, M. K., & Jasrotia, S. S. (2023). Ethics and marketing responsibility: A bibliometric analysis and literature review. *Asia Pacific Management Review*, 28(4), 567–583. <https://doi.org/10.1016/j.apmr.2023.04.002>
- Konvoy. (2023, October 6). *The Complexity Of Game Engines*. Konvoy. <https://www.konvoy.vc/newsletters/complexity-of-game-engines>
- Kramarzewski, A., & De Nucci, E. (2023). *Practical Game Design—Second Edition*. O'Reilly Online Learning. <https://learning.oreilly.com/library/view/practical-game-design/9781803245157/>

- Krita. (n.d.). *Digital Painting. Creative Freedom*. Krita <https://krita.org/en/>
- Linktree. (n.d.). *Linktree: Link everything you are*. Linktree <https://linktr.ee/>
- Localthunk. [@localthunk.bsky.social]. (n.d.). Bluesky Social. Retrieved April 4, 2025, from <https://bsky.app/profile/localthunk.bsky.social>
- Microsoft. (n.d.). *Microsoft Planner | Daily Task and Work Management*. Microsoft. <https://www.microsoft.com/en-us/microsoft-365/planner/microsoft-planner>
- Midgley, M. (1974). *The Game Game*. *Philosophy*, 49(189), 231–253. <https://doi.org/10.1017/S0031819100048208>
- Moraes, G., & Douglas, R. (2022). *Microsoft 365 Fundamentals Guide*. O'Reilly Online Learning. https://learning.oreilly.com/library/view/microsoft-365-fundamentals/9781801070195/?sso_link=yes&sso_link_from=hamk-fi
- Oxford English Dictionary. (n.d.). *video game*. Oxford English Dictionary. https://www.oed.com/dictionary/video-game_n?tl=true
- Piskunov, E. (2025, January 16). Guide on Stages of Game Development: From Concept To Release. *iLogos Game Studios*. <https://ilogos.biz/stages-of-game-development-your-guide-on-game-development-process/>
- Scalzo, J. (2022, August 25). Video Game or Videogame? An Answer to the Most Important Question of Our Time. *Video Game Canon*. <https://www.videogamecanon.com/adventurelog/video-game-or-videogame-an-answer-to-the-most-important-question-of-our-time/>
- Seyyedamiri, N., & Tajrobehkar, L. (2019). Social content marketing, social media and product development process effectiveness in high-tech companies. *International Journal of Emerging Markets*, 16(1), 75–91. <https://doi.org/10.1108/IJOEM-06-2018-0323>
- Steam. (n.d.). *Steam, The Ultimate Online Game Platform*. Valve. <https://store.steampowered.com/about/>
- TikTok. (n.d.). *About*. TikTok. <https://www.tiktok.com/about>
- Tour De Pizza. [@tourdepizzaz.bsky.social]. (n.d.). Bluesky Social. Retrieved April 4, 2025, from <https://bsky.app/profile/tourdepizza.bsky.social>
- Unity. (n.d.). *Wondering what Unity is?* Unity <https://unity.com/our-company>
- Unreal Engine. (n.d.). *The most powerful real-time 3D creation tool*. Unreal Engine <https://www.unrealengine.com/en-US/home>
- XPPen. (2024). *Drawing Tablet: What Is It, How Does It Work and What to Do with It*. XPPen. <https://www.xp-pen.com/blog/drawing-tablet-guide.html>
- YouTube. (n.d.). *Ever wonder how YouTube works?*. YouTube. https://www.youtube.com/intl/en_us/howyoutubeworks/

Appendix 1: Data management plan

During the development of the project, a diary is kept using such application as university's account for Microsoft Word, in which the technical information about the project is collected, stored, and then analysed for the thesis. GitHub Desktop was used to back-up the code of the project. Sensitive or personal data was not collected. In order to understand the marketing strategies that are used by other game developers on social media, game developers' Bluesky accounts were analysed. To analyse the data from the author's social media accounts, the engagement data was exported from YouTube and TikTok. Bsky Hub was used to collect data from Bluesky. The data was saved for further Excel analysis and charts were created. Since Itch.io had a possibility to see analytics, but did not have an option to export the data at the time of the implementation of the thesis, the data was manually recorded.

Management and storage of the research data

All data will be stored on the author's password-protected computer. Data will be regularly backed up using university's Microsoft OneDrive. Additionally, backups of the data will be saved in a separate folder.

Processing of personal data and sensitive data

The personal data will not be collected in the thesis. The research will be based on the publicly available information, such as academic sources, official documentations, public social media accounts. The engagement data collected from social media accounts will be anonymous, since the collected data will only include the number of users engaging with the content. The names of the accounts will not be stored.

Ownership of research data

All collected data is owned by the author of the thesis.

Further use of research data after the completion of the thesis

The research data will not be reused. The data will be stored for one year after the completion of the thesis to ensure that the results can be verified if necessary. The data will be securely destroyed after one year.