Politechnika Wrocławska

Faculty of Computer Science and Management

Field of study: **COMPUTER SCIENCE**

Bachelor Thesis

Mobile platform game created with Unity

Anna Julia Birecka

keywords:
Unity
Mobile phone game
Puzzle-platformer

short summary:

The aim of the thesis is the development of a mobile platform game using the Unity engine. It is a puzzle-platformer game implemented for the Android operating system. The project contains an introduction to the topic, a description of design choices, implementation, and encountered problems with solutions, as well as further development possibilities.

Supervisor	……Dr. inż Marek Kopel……..		
	Title/ degree/ name and surname		
The final evaluation of the thesis			
Chairman of the Diploma Examination Committee	.....	.....	…….
	Title/ degree/ name and surname	*grade*	*signature*

For the purposes of archival thesis qualified to: *

a) Category A (perpetual files)
b) Category BE 50 (subject to expertise after 50 years)

* Delete as appropriate

stamp of the faculty

[redacted]Wrocław, 2021

Streszczenie

W obecnych czasach gry mogą być dobry źródłem wiedzy i rozrywki. Coraz więcej możemy zaobserować pozytywnego wpływu dobrze dobranych gier do wieku użytkownika.

Tematem niniejszej pracy inżynierskiej jest opracowanie gry mobilnej przy użyciu silnika Unity. Produktem tej pracy jest gra z gatunku puzzle-platformer, gdzie zadaniem gracza jest rozwiązywanie zgadek aby mógł ruszyć dalej. Zagadki zaprezentowane w tej grze są inspirowane postaciami z mitologii słowiańskiej.

Praca składa się z pięciu rozdziałów, w których szczegółowo opisano poszczególne etapy pracy and grą. Pierwszy rozdział to wprowadzenie to tematyki gier oraz porównanie istniejących produkcji. Drugi rozdział dotyczy projektowania interfejsu, rozgrywki, zasad gry oraz opis grupy docelowej. W trzecim rozdziale opisano wykorzystane techniki implementacji oraz napotkane problem. Ostatni rozdział poświęcony jest podsumowaniu rezultatów oraz opisem przywidywań dlaszych kierunków rozwoju.

Abstract

Nowadays games can be a good source of knowledge and entertainment. The positive impact of a well-chosen game for the users' age can be observed.

The subject of this engineering thesis is the development of a mobile game using the Unity engine. The product of this work is a puzzle-platformer game, where the player's task is to solve puzzles in order to move along the gameplay. The puzzles presented in this game are inspired by characters from Slavic mythology.

The work consists of five chapters in which all the stages of the work had been described in detail. The first chapter is an introduction to the subject of games and a comparison of existing productions. The second chapter deals with the design of the interface, gameplay, rules of the game, and description of the targeted audience. The third chapter describes the implementation techniques used and the problem encountered. The last chapter is devoted to summarizing the results and describing the expectations for future development directions.

Table of content

1. Introduction

This chapter serves as an introduction to the topic and terminology of mobile phone games with a description of similar games with reasoning as to why the target platform and audience were chosen.

1.1. Mobile Phone Games

The creation of a mobile phone created also a possibility for a new game format.

The first game created for mobile phone devices was Tetris on a mobile phone called Hagenuk MT-2000 in 1994 [1].

In 2007 Apple released its first iPhone, its debut opened a new era for mobile gaming. Due to its design, the design of games also changed, opening new possibilities for game developers. In 2009 the success of the game Angry Birds (a physics-based game involving launching birds to destroy structures made by pigs in order to make as much damage as possible) made Apple's altered the mobile phone game landscape. Many successful games have millions of players and have annual revenues exceeding US$100 million a year, with the top games breaking US$1 billion.

At present times mobile games do not differ much from many titles that are playable on a computer. Even though the only forms of interaction are single taps on a screen, mobile games may carry amazing experiences for a player. Due to often easy gameplay, portability, and the possibility to "end" game at any moment they are the perfect match for most casual gamers.

1.2. Terminology

1.2.1. Side scroller

A side-scroller or side-scrolling game is a genre of video games in which action is viewed from a side-view camera angle [2]. Players move a character from left to right while the screen rolls with them. This type of gameplay gained popularity especially in a connection with platform games due to the possibility of expanding the platform games, not to only single-screen format.

1.2.2. Puzzle

Puzzle games [4] is a genre of games where the player is exposed to different types of problems that test their skills. Puzzles may take a form of a logical problem, word completion, logical quiz, sequence solving, or pattern recognition.

The game that popularized this genre was Tetris. It was created by Alexey Paijitnov in 1984, who took inspiration from a traditional puzzle game called Pentomino. Ten years later Tetris was introduced as the first game on a mobile phone.

1.2.3. Platformers

Platformers or platform games are games that mainly revolve around a character controlled by the player, which runs and jumps to avoid obstacles and/or to defeat enemies [2]. This is one of the first genres of games that were created.

Platformers can be categorized into two distinct types: single-screen platformers or side-scrolling platformers.

Single screen platformers display each level on a single static screen.

Figure 1.1 A whole level is visible all the time in the game Escape Goat (2012)

These types of games have multiple levels, where each level becomes increasingly difficult as players progress. Due to technical limitations, most of the arcade platformer games are single-screen platformers.

Side-scrolling platformers are a fusion of a side scroller and a platform game. Just like single-screen platformers, scrolling platformers can have multiple increasingly difficult levels. The most popular example of this genre of games is the console game Super Mario Bros (1985) [3].

1.2.4. Autosave/ savepoint

It's a saving function in games that automatically saves the progress made in it. Autosaving is done in predetermined intervals or before a complex task is begun. Such a point in-game is called savepoint.

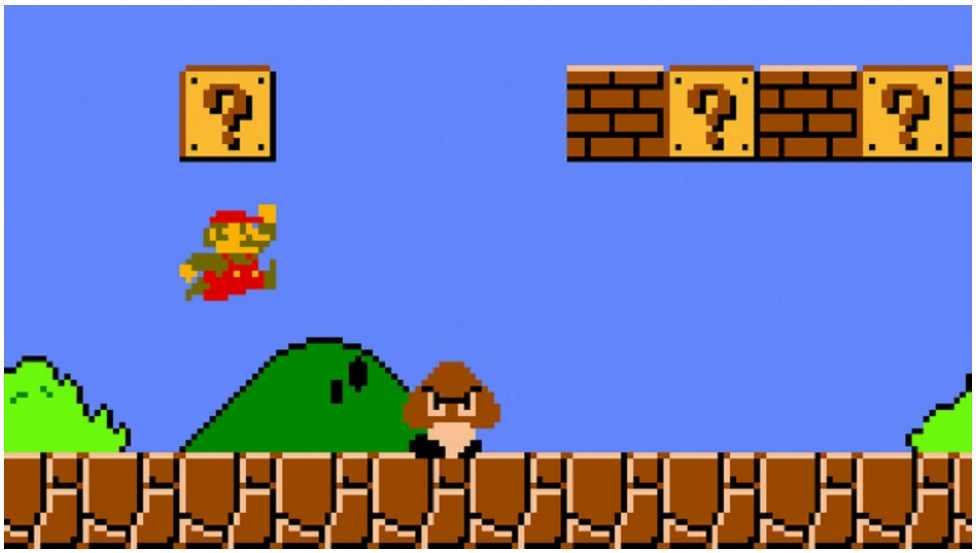

Figure 1.2 A part of a Super Mario Bros' level

First side-scroller platform game- Jump Bug-released in 1981 by Alpha Denshi for Arcade.

1.3. Thesis goal

The goal of the thesis is to create a playable mobile phone game and implement of two-dimensional puzzle side-scrolling platformer with unique character designs inspired by Slavic folklore. The game should be supported by any mobile phone device with Android 6.0 (Marshmallow) and above. The controls should be easy enough for a child to play with suitable puzzles to stimulate creative and logical thinking.

1.4. Thesis scope

The scope of a project consists of a creation of a playable alpha prototype and deployment for mobile devices operating on the Android system. The game will include original graphics and animations appropriate and consistent with a theme as well as riddles and audio layer.

Due to a time restriction, there is only a possibility to create an alpha version of the game. This version includes the most important features such as:

- providing an enjoyable simple gameplay
- the implementation of riddles
- the implementation of player movement
- health system
- collecting points
- the animations
- the audio layer

The thesis consists of five chapters. The first one is an introduction to a topic of games, terminology, analysis of existing solutions with similar themes and gameplay, and reasoning as to why such platform was chosen. The second chapter explains chosen design choices and shows the detailed Game Design Document for a game. The third chapter shows an implementation of a project with a description of used tools, chosen solutions for encountered problems, and design patterns. The last chapter serves as a summary and possible further development of a project since games require constant post-release maintenance in order to stay relevant on a market.

1.5.Similar games:

1.5.1. Limbo

Figure 1.3 Screenshot of a gameplay Source https://playdead.com/games/limbo/

Limbo [figure 1.3] is a black-and-white 2D puzzle platformer where a player controls a young boy who wakes up alone in the forest. The player is set up to escape from this unsettling, dangerous environment. During the adventure, they have to solve puzzles and most important- survive. In this game, the player dies a lot which may be frustrating. After death player respawns at the nearest checkpoint and then tries again to figure out how to solve a puzzle. There are many problems where the player simply does not know what to do or the necessary attributes for solving a puzzle are hard to notice due to the monochromatic graphic. Although its atmosphere and interesting design game are challenging and not really recommended for young or casual gamers because of difficulty and violence.

Pros:

- Atmospheric and interesting design
- Easy controls scheme
- After death player respawns at the nearest checkpoint
- Smooth animation

Cons:

- Not intuitive UI design
- The Player dies a lot in order to solve a puzzle
- Not suitable for children
- A noticeable delay in a character's movement

1.5.2. Lucid Dream Adventure

Figure 1.4 Screenshot of a gameplay Source: https://store.steampowered.com/app/773090/Lucid_Dream_Adventure/

Point and click puzzle platformer. The player plays as a young girl traveling through memory/dreamland. It is a really casual, light game with colorful and creative graphics suitable for its theme. Although the puzzles that the player encounters lack balance. At first playable character is confronted with the really easy riddle, the second on the other hand is way more difficult and lacks logical explanation. Then, again, the player is presented with an easy one. This lack of consistency during the gameplay is off-putting and straight-up discouraging. Playable character or the one's player meets on their journey are shallow and one-dimensional which makes story boring. Therefore, the player may not be invested in continuing the gameplay after the first few puzzles.

Pros:

- Colorful art style suitable for children
- Dream-like sequence
- Well thought UI design

Cons:

- Chunky animation
- The game is divided into a separate game- chapters
- User must pay and download a "new game" for every chapter
- Encountered puzzles lack balance in difficulty

1.5.3. Never Alone

Figure 1.5 Screenshot of a gameplay Source: https://store.steampowered.com/app/295790/Never_Alone_Kisima_Ingitchuna/

Captivating and beautiful platform game based on a Native Alaskan Inupiaq people's folklore and culture. Although it's an amazing story and deep understating of Inupiaq's culture there are many difficulties as the gameplay progress. The player controls a young girl, with a fox, on her journey to find the source of a powerful blizzard.

This is a standard side scroller platformer with typical for its genre mechanics: running, jumping, and pushing boxes to jump to higher ledges. What makes this game really discouraging are bugs and failing, unresponsive controls which result in deadly falls into freezing water or abyss. For a game where timing is crucial lack of well-responding controls is extremely frustrating. The frustration is compounded by the fact that the fox, controlled by AI, can move away from the platforms which fade out of view and cause characters who were once standing on them to fall. Mechanics in this game makes it almost unplayable.

Pros:

- Creative and well-thought story
- Beautiful animation

Cons:

- Problems with AI-controlled sidekick
- Unresponsive controls
- Hard to notice blizzards which may lead to a failure

1.6. Target platform

The choice of the target platform is dictated by the availability of equipment. The constant growth of smartphone popularity and possibilities that it is offering in contrast to other mobile devices. Which may be often bigger in size or more expensive and yet do not carry the same amount of advanced technology or offer as much as smartphones.

In 2019, there were 3.2 billion smartphone users across the world according to Statista. Research shows that 96% of people born between the mid-90s and early 2000s and 52% of children between eight and twelve own and actively use mobile phones [figure 1.6].

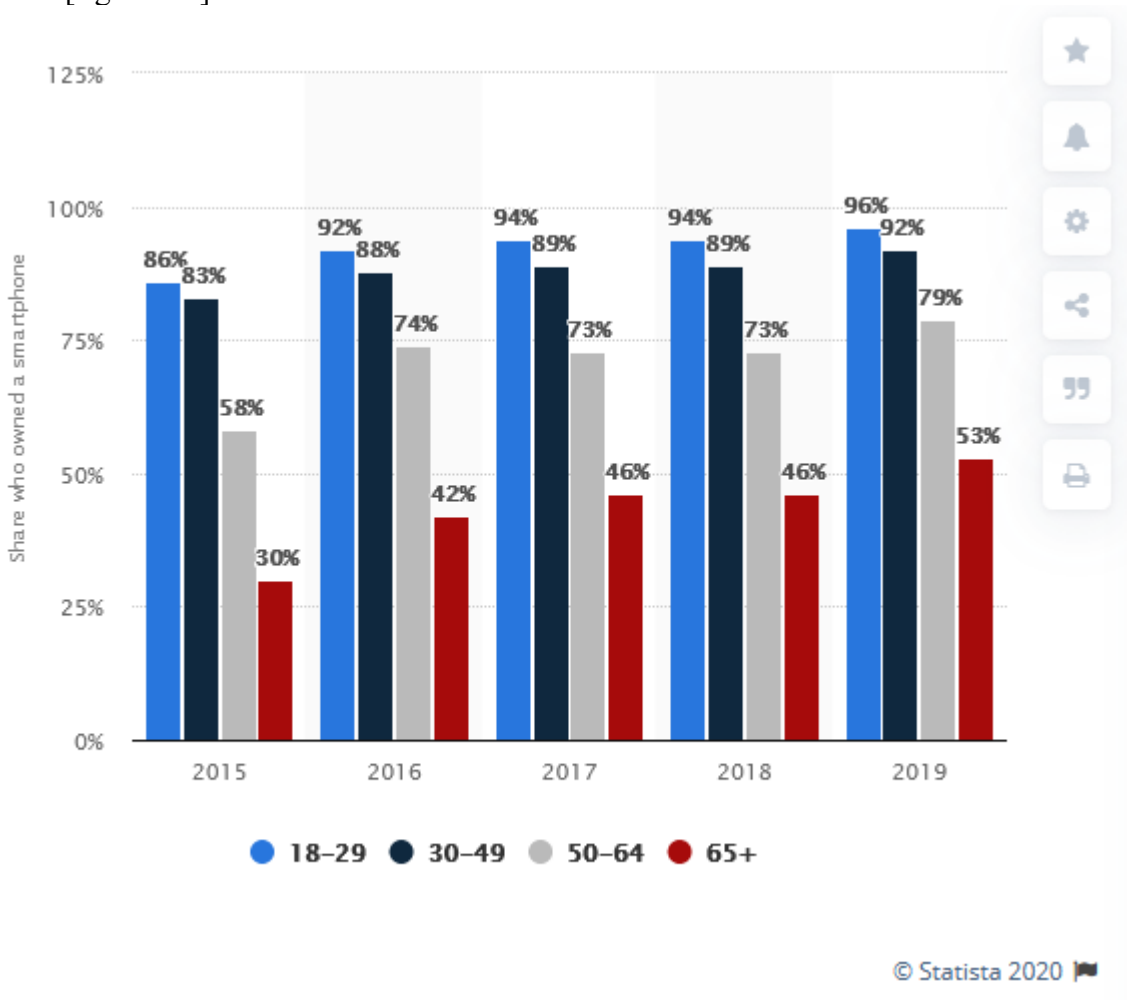

Figure 1.6 The graphic representation of smartphone ownership by age group in years 2015-2019 source: statista.com

For this project, I settled for Android Marshmallow 6.0 (and above) as a destinated platform due to the higher presence of smartphones with Android operating system on the market [figure 1.7]. Therefore it's a higher chance that the game will reach more players.

Year	2018	2019	2020
Android	85.1%	86.1%	84.8%
iOS	14.9%	13.9%	15.2%
Others	0.0%	0.0%	0.0%
TOTAL	**100.0%**	**100.0%**	**100.0%**

Figure 1.7 Percentage of Android users Worldwide, source: https://www.idc.com/promo/smartphone-market-share/os

According to market research, the popularity of mobile gaming is increasing in the upcoming years [figure 1.8]. The reason for it is the availability of a device and the short, interactive form that the game takes, as much as a possibility to end it at any given moment. Due to its form smartphone games do a great job engaging children, women, and parents. They appeal to a wider audience than traditional gaming alone.

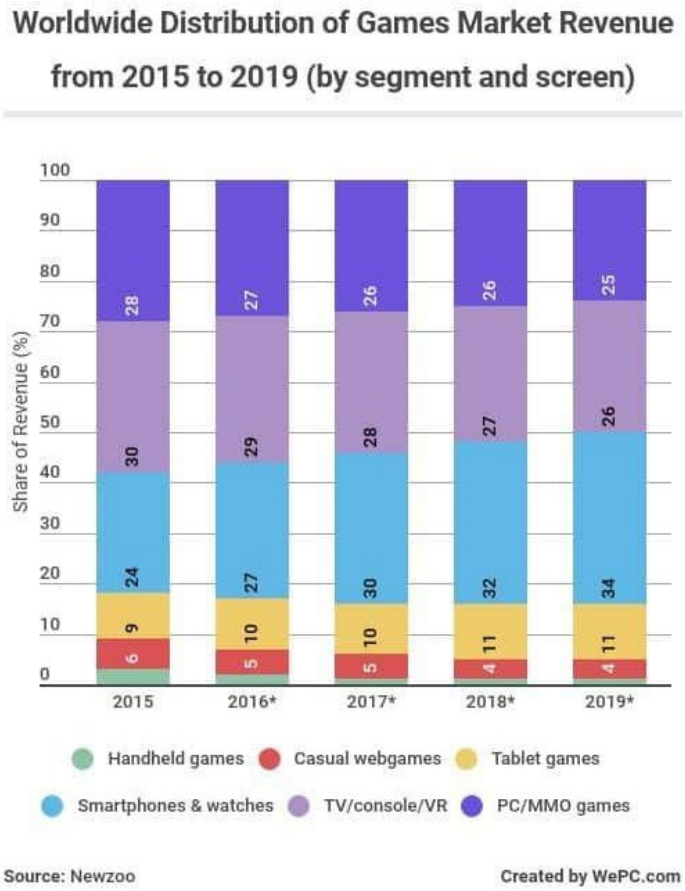

Figure 1.8 Percentage of players during the years 2015-2019, source: newzoo for wepc.com

According to mobile gaming market statistics [5], research women prefer puzzle games with more casual undertones and card games, while men are keener on strategy games with action, puzzle, and role-playing games. Also, there is a big percentage among parents who purchase mobile game apps for their children, seeking educational and family-friendly games that might be played across generations.

2. Design

The game is a 2D puzzle side-scrolling platformer. The player takes control of a child, who must travel through the forest solving quizzes and gaining points in form of glowing worms. In the later progress of a game, the player will be able to exchange points for a hint in order to solve a puzzle. The game will autosave before and after every encounter with the enemy. The player will be able to start their game at any chosen save-point that they already passed. Therefore the player will be able to see their progress throughout the game. Every enemy will present the puzzle that a player has to solve in a given time in order to move forward with the gameplay. As the game progress so will the difficulty level of the puzzles. Furthermore, the game should have simple and responsive controls.

The project will be free-to-play due to fact that free-to-play is more likely to be downloaded than paid. This will mean that the game will appear to the wider audience, which is one of the goals of the project.

2.1. Methodology

Most games do not serve as a solution to a real-world problem in order to improve some aspects of life, with the exception of therapeutic games [6].

Game development is a continuously evolving process, where the pipeline is not linear, but rather fluid and flexible. Although it can be divided into three stages: pre-production, production, and post-production.

Pre-production defines what a game is about and why it should be made with an evaluation of resources and finances available to make it. During this time the targeted audience is established. Game Design Document is added, to keep work organized. The game's prototype is made to test functionality, user experience, gameplay, mechanics, and art direction.

Production is the longest stage of a project. During this time the story is defined, assets like characters, creatures, and environment are created, the levels and code are written.

Post-production is time after the game is released, all documents, assets, and code are finalized.

Due to the fact that games differentiate in structure compared to a regular application, game development is a team effort of artists, designers, and developers. For a smaller creator, it's a challenge, because they have to exceed in a broad range of skills from the creation of art, animation to writing code and planning the story.

2.2.Main theme and reasoning

The idea for a game came from the notice of increasing interest in Slavic culture and its mythology. Thanks to games, series, and books like Witcher or, gaining popularity in western countries, witchcraft inspired by an easter culture gives a good basis for such a game.

According to Google trends, there is a big interest across the world in Slavic culture and folklore.

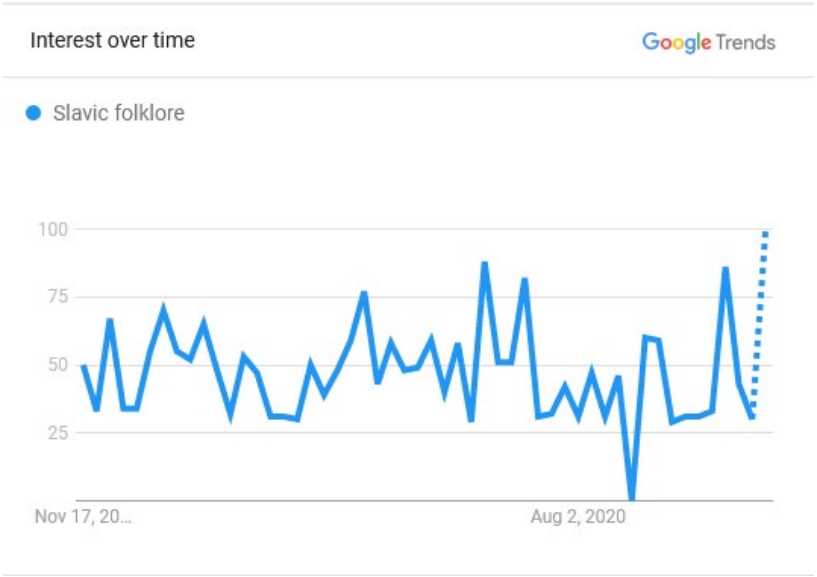

Figure 3.1 Graphic description of increasing interest in Slavic folklore during the year 2020- source: https://trends.google.pl/trends/explore?q=slavic%20folklore

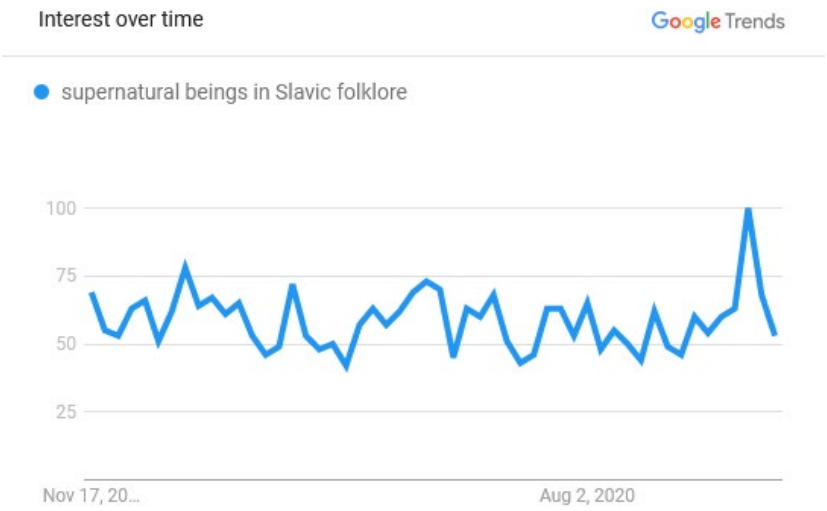

Figure 3.2 Graphic description of interest in Slavic folklore during the year 2020- source: https://trends.google.pl/trends/explore?q=supernatural%20beings%20in%20slavic%20folklore

Additionally, creating such a game may help to expand the knowledge and beauty of Slavic mythology. Even for Slavs, the knowledge about it is dim or almost non-

existing, while it's part of their culture and national heritage. This project came from the need to educate people, starting from the youngest, and sparking the interest in such a topic for further exploration [7].

2.2.1. Targeted group

The targeted group for this project is children between ages 8-12. Due to the nature of this game, Family Policies must be taken into consideration. Restrictions that need to be followed are as followed [8]:

- Content that children see is appropriate for them
- The app should not display any adds, as those may be inappropriate
- App won't collect any personal and sensitive information
- The audience won't be required to sign-in or access app content through an API or SDK that is not approved for use in child-directed services

There should not be any problem with applying those restrictions to the project.

This is also a great opportunity to educate not only children about Slavic Mythology but also their parents.

Research shows that games can help a lot with a child's development [9]. The main benefits of playing games involve enhancing their mental skills such as:

- problem solving and logic,
- hand-eye coordination,
- managing resources and logistics,
- multitasking,
- quick-thinking,
- developing reading and math skills,
- memory,
- concentration,
- pattern recognition
- perseverance

With these observations in mind, I decided to create a fun, but challenging game inspired by Slavic folklore that will educate and grab the interest of a younger player. All creatures encountered by a player have their origins in Slavic mythology. Puzzles are inspired by supernatural being's myths and logical quizzes.
Visuals are influenced by books for children in order not to scare a player. With this project, I want to spark an interest in beautiful Slavic mythology in a child's mind.

2.3. Lore

One of the most important reasons why players chose a game is its' lore. The player takes control of a child, who must travel through the forest in order to rescue their cat stolen by Baba Yaga. On their path, they met with various characters from Slavic folklore. Each creature presents a different puzzle to a player, based on their origin and story. In order to move forward with gameplay player must solve a given puzzle. During the gameplay, the player has to pick up points in form of fireflies. Slavic believes fireflies represented the gnomes' lights. If a child got lost in a forest they helped them to get out safely. [10]

Each creature was chosen based on its lore and helpfulness towards lost children in the forest.

2.4.Creatures

Creatures that the main character encounters during their journey in the forest are all inspired by beasts described in the following books "Bestiariusz Słowiański" [11], "Mitologia Słowiańska" [10] and "Wielka księga demonów polskich" [12]. These chosen creatures are one of the most interesting ones in Slavic mythology and a good introduction to further exploration of the topic.

2.4.1. Forest aunty – lady of a Kashubian forest

Nice old lady clothe in a leaf dress [figure 3.3]. When she met a child lost in the forest she gave them berries and tried to help them find their way back home. She is the first deity that the player will encounter on their journey. Her task is to guide a child and give them an easy logical puzzle.

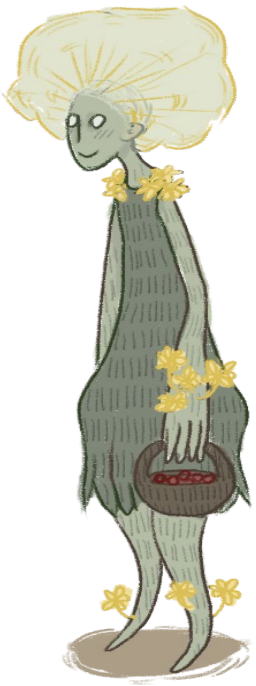

Figure 3.3 Lady of a Kashubian forest-source: own

2.4.2. Vila

A fairy, portraited as a beautiful woman with long blond hair [figure 3.4]. Vile is usually friendly to people, but they can take horrible revenge on those who insult them, disregard their orders, or uninvitedly approach their circle dance. She is the second supernatural being that the player meets. Due to her connection with water, the player will be asked to solve a riddle based on a water element.

Figure 3.4 Vila- source: own

2.4.3. Leshy

Leshy is masculine and humanoid in shape, is able to assume any likeness, and can change in size and height [figure 3.5]. He has a neutral disposition towards humans but can help someone if they ask him nicely. Due to him being a tutelary deity of the forest all his riddles will revolve around it.

Figure 3.5 Leshy- source: own

2.4.4. Baba Yaga

Is a supernatural being who appears as a deformed or ferocious-looking old woman [figure 3.6]. In Slavic culture, Baba Yaga lives in a hut usually described as standing on chicken legs and flies on a butter churn. She is often portrayed as a villain, child-eating monster (which won't be the case in the game for keeping it child-friendly). She is the last encountered creature in the game with the most difficult riddle to solve for a player. If players succeed they will be rewarded with a "happy ending", which for the main character means getting their cat back. If the player fails they will need to start over.

Figure 3.6 Baba Yaga, source: own

2.4.Concept of the environment

In order to make the game child-friendly, I decided to create a tale-like background. Strongly inspired by today's children's book and a strong connection to nature in Slavic culture I decided to create a fairy-tale-like forest [figure 3.7].
In this type of environment, the player will be more able to connect to the story presented in the game. The color palette is dim but not dark, to give a player feeling of a late-afternoon.

Figure 3.7 Concept art of a background created using Procreate and Photoshop, source: own

2.5.Main character design

While creating the main character that the user will play I tried to take into consideration that boys and girls will be playing it. Therefore the character should be accessible for both of them in order to be appealing and a connection between player-character will be made [7]. Due to these observations, few decisions have been made:

- the main character should be at an age that is relatable for a younger audience
- they should be wearing neutral-gendered clothing for not to exclude anyone
- they should wear something colorful that will be eye-catching and easily distinguishable from other characters

With these rules in mind I created the following concept art for the main character:

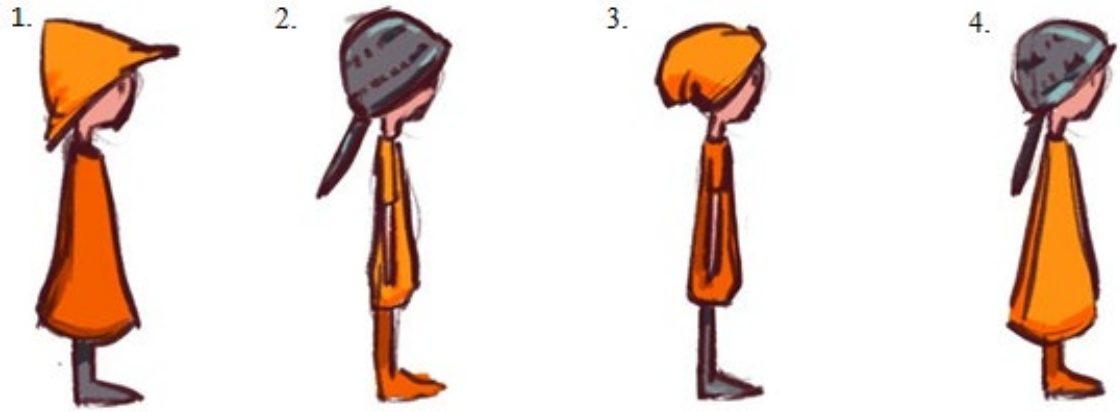

Figure 3.8 Concept art of the main character, source: own

After gathering feedback from possible users I've settled on a third [figure 3.8, option number 3] option. This main character is the most appealing and relatable for today's youth.

2.6.Game rules

The player can only move forward as it's typical for a platformer, side-scroller game. In order to move forward, the player has to answer a riddle or solve a puzzle [4].

2.6.1. Health

To make the game more exciting and challenging for a player, the health system has been introduced [figure 3.9]. Users start the game with a maximum of three hearts [figure 3.10]. Every time they answer the riddle wrongly one heart is taken away. Zero hearts results in a game over and the player has to start the new game from the beginning.

Figure 3.9 Screen from the beginning of the game after pushing the play button, source: own

Figure 3.10 The health points, source: own

2.6.2. Points- fireflies

To balance the game better, and not discourage children from playing, I decided to introduce the points system [figure 3.12]. During the gameplay, the player will have a possibility to "catch" fireflies-points. These points are stored in a jar on the right-bottom of a screen. These points can be later exchanged for a clue to solve a puzzle [figure 3.11]. When the player encounters the creature and does not know the answer for a riddle, the jar starts to jiggle indicating that the gathered points might be exchanged for a clue.

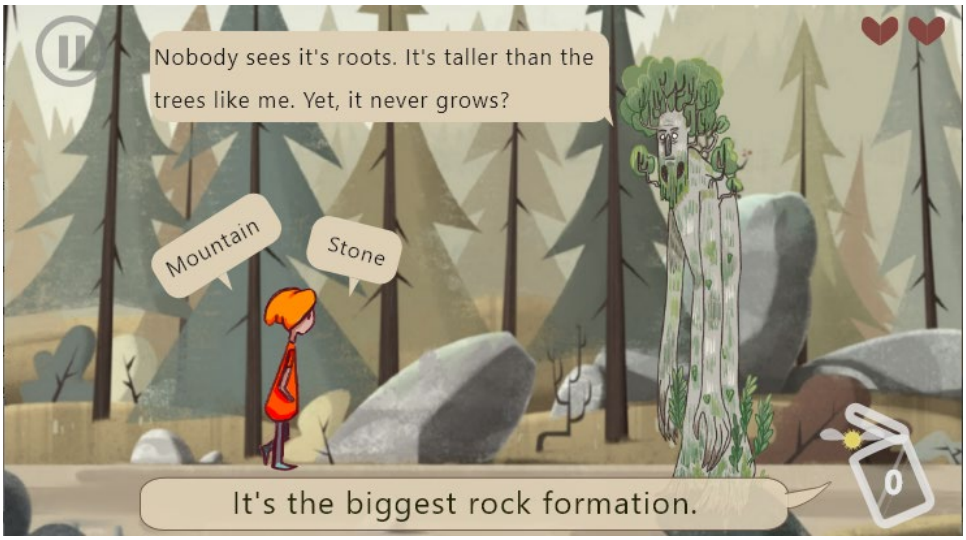

Figure 3.11 Screen from the game, using points in exchange for a clue, source: own

Figure 3.12 Number of collected points, source: own

2.7.Controls

The mobile games should have easy and intuitive controls, that all players would be able to pick-up rather quickly. In order to make sure that players will know what to do, at the beginning of the game, on the bottom of the screen, information about how to move will appear [figure 3.9 and figure 3.13].

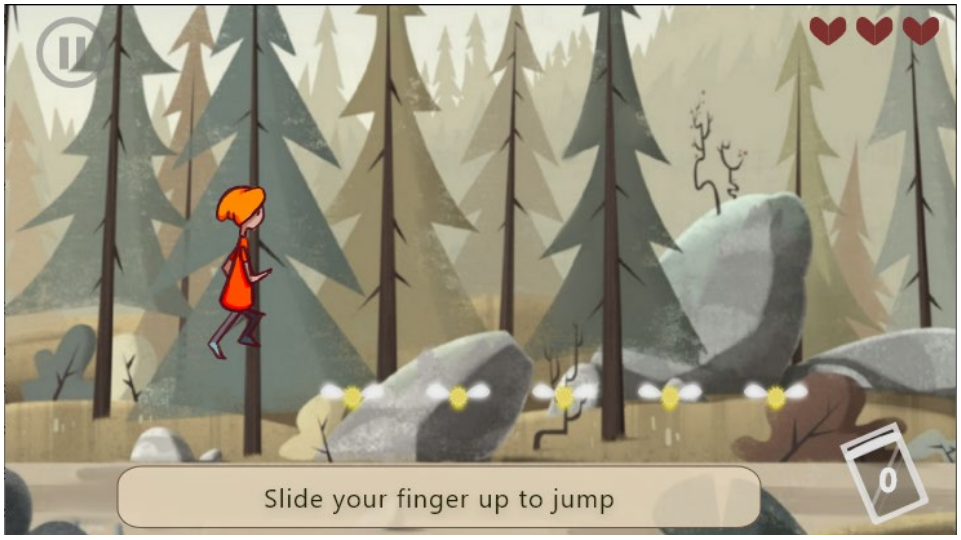

Figure 3.13 Screen from the beginning of the game, source: own

2.8. User Interface

It's important for a game to have a well-designed interface [13]. A good game UI is an interface with components that help players navigate and find information. List of game UI components—life bars, point counters, and levels. Like UI design for mobile apps, mobile game UI design requires attention to details and functionality. To design UI in this project, Adobe XD was used.

2.8.1. Main menu

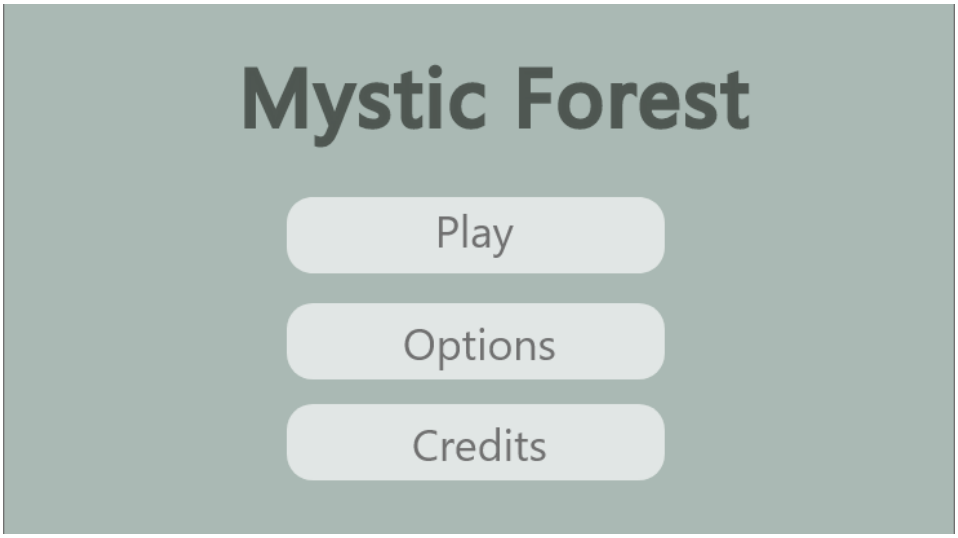

Figure 3.14 Main menu's mockup, source: own

The first screen that players see is the main menu screen [figure 3.14]. It should be clean and intuitive with all necessary functions. Order in the main menu is important, due to video game standards, therefore the play button is put as first, options button as second, and, least importantly, credits- last.[4]

2.8.2. Options

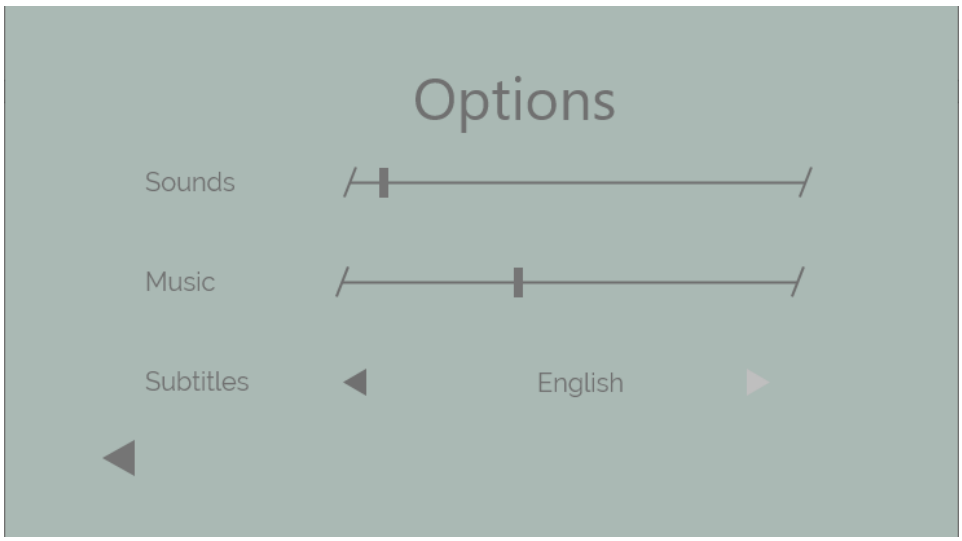

Figure 3.15 Mockup of the options menu, source: own

Due to the simple nature of a project, there was no need for more compound options menu [figure 3.15]. Here the player can easily set their preferences of volume and language.

2.8.3. Credits

Figure 3.16 Mockup of the options menu, source: own

In-game industry, even in small projects, it is important to properly credit [figure 3.16] everyone who partook in a game development process. Good game development etiquette is also credit all used work and third part assets.

2.8.4. Pause menu

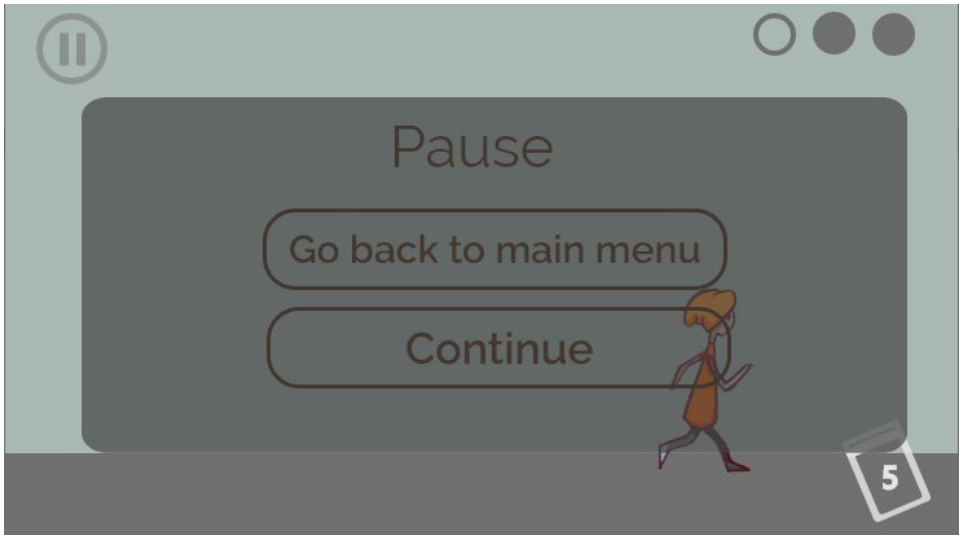

Figure 3.17 Pause menu's mockup, source: own

There is a possibility for a player to pause a game at any given moment, then the pause menu pop-up on the screen [figure 3.17]. Here the player has a possibility to return to the main menu of a game or continue their journey in the game.

3. Implementation

3.1. Development and tools

3.1.1. Unity- game engine

It's a cross-platform game engine, which allows for the implementation of two-dimensional and three-dimensional games. It gives the user the ability to write scripts in C# for Unity editor, plugins, and the game itself, although the engine was written in C++.

For purpose of this game, the selected version of the Unity engine is 2019.4.17f1. This game engine is suitable for two-dimensional games.

3.1.2. Microsoft Visual Studio 2019

It is an integrated development environment from Microsoft. It is used for the implementation of computer programs, websites, and mobile apps. This game was written in C# which is supported by Visual Studio 2019

3.1.3. Adobe Photoshop

Raster graphics editor developed by Adobe Inc. Photoshop offers users various tools for graphic design, as also for painting and creating concept arts and animation for games, that can be later imported into Unity engine. This program was used for creating all sprite sheets, background, and assets needed for my project.

3.1.4. Procreate

It is a raster graphics editor developed for the iOS system, which offers multiple tools for digital art. It also supports export/import to Adobe Photoshop which is useful in creating a well-put-together graphic for a game. This program was used for an early stage of concept art development.

3.1.5. Adobe XD

Adobe XD is a free, vector-based program for creating interactive mockups for application interfaces. It has a lot of helpful features that make creating mockups faster.

3.2. Outsourced assets

Outsourced assets, or third-party assets, are often used in game development as a possibility of decreasing the time development of the game. Due to the simple design of my game, I was able to create most of them myself.

3.2.1. Fonts

Unity engine doesn't support a lot of fonts. Therefore there was a need to find free, suitable fonts for this project. A website called "101 fonts" has to offer a wide variety of different free for commercial use fonts. Selected font had to be easy to read for children, age eight to twelve.

Roboto Font Family made by Christian Robertson:
https://www.1001fonts.com/roboto-font.html

3.2.2. Music

The creation of custom music was not possible in this project due to a lack of knowledge in this field. Therefore the free soundtrack has been used:

Main menu- Ether Vox by Kevin MacLeod:
https://incompetech.filmmusic.io/song/7014-ether-vox

In-game sounds 1- Autumn wind grabs the trees by Strategy:
https://freesound.org/people/straget/sounds/403764/

In-game music- Wind Of The Rainforest Preview by Kevin MacLeod:
https://incompetech.filmmusic.io/song/5729-wind-of-the-rainforest-preview

3.3.Graphics

The graphic is one of the most important assets in video games. It's the first thing that the player sees. Unappealing graphics might make players quit the game. Therefore it is important to make graphics in sync with the overall theme of the game.

3.3.1. Sprites

Sprites are two-dimensional bitmap mostly used in two-dimensional video games [14]. In this project sprites [figure 4.1.] were made using Adobe Photoshop and exported to a .png file format. This type of file format is better compared to others like the .jpeg file format that is lacking background transparency or .bmp file format that makes sprites lower in quality.
I decided to use sprites rather than vector graphics due to the fact that sprites were better fitted for the project theme. Sprites assets gave the project possibility to look more like a child, fairy-tale book. The filter has been set on bilinear [figure 4.2] which is more fitting for drawn assets, while point(no filter) is more fitting for pixel art assets. Bilinear filter [figure 4.3] makes lines a little bit blurry, this is a wanted effect as it makes assets look more like drawn pictures in books for children.

Figure 4.1. Sprite sheet ready for the animation of Vila, source: own

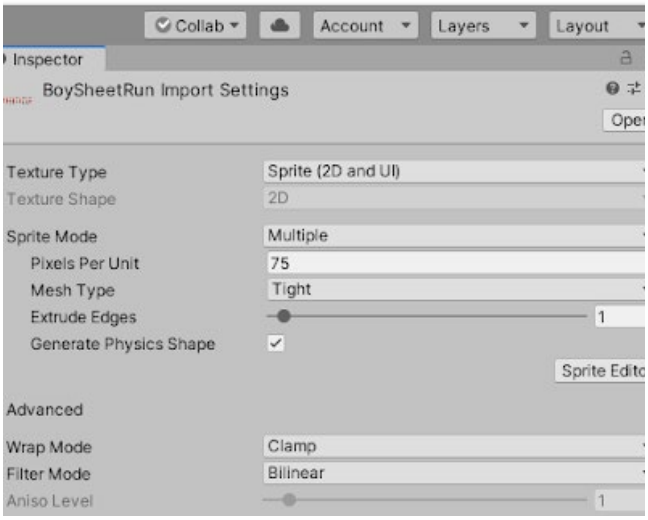

Figure 4.2 Filter mode is set on bilinear, source: own

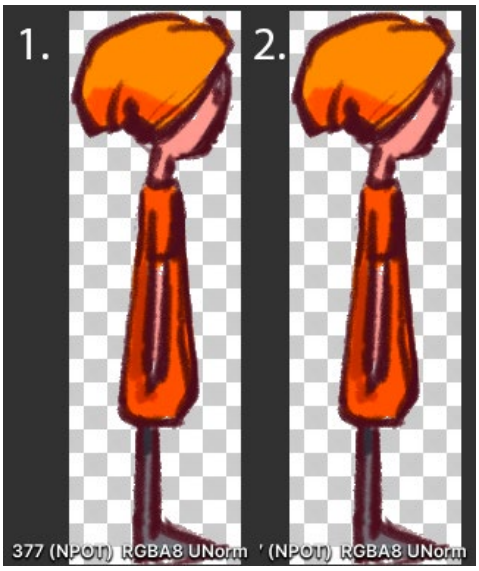

Figure 4.3 Sprite comparison: 1- with bilinear filter, 2 – with point filter, source: own

3.3.2. User Interface

The main menu's interface [figure 4.4], options interface [figure 4.5], and in-game interface [figure 4.6] were created in accordance with mockups from chapter 3.8. The background for a main menu and options is concept art of the environment from chapter 3.4 [figure 3.7]. This way when the player starts the game they will automatically have continuity with starting the game by tapping the play button.

With the in-game interface comes also the creation of better-looking assets [figure 4.6].

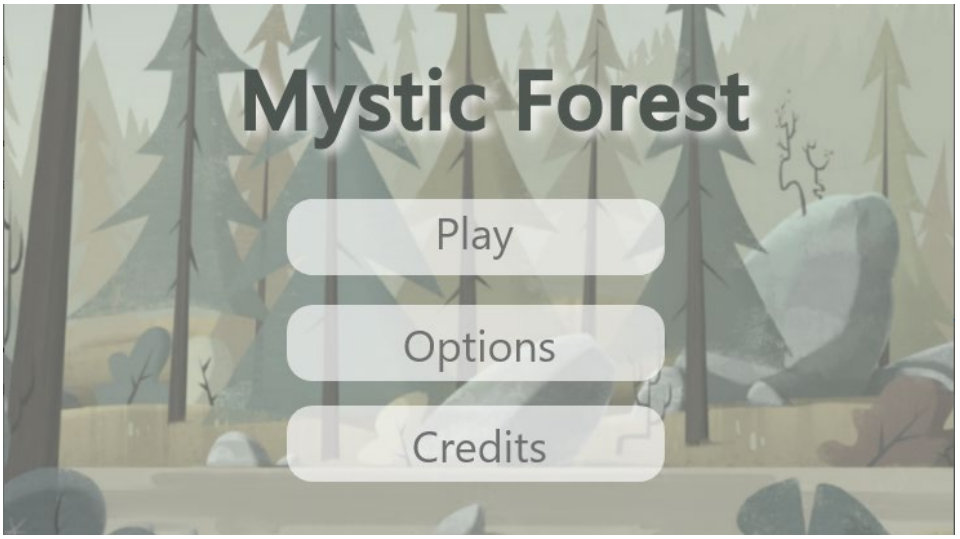

Figure 4.4. Main menu, source: own

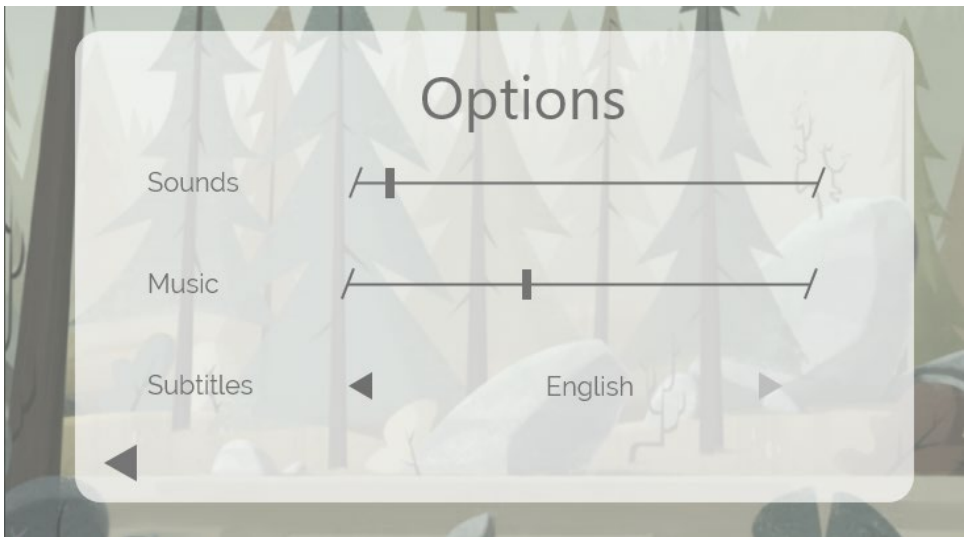

Figure 4.5 Options menu, source: own

Figure 4.6 Screen from the game, source: own

3.4.Animation

There are two main animation styles in two-dimensional video games: skeletal animation [figure 4.7] and sprite sheet animation [figure 4.8]. Skeletal animation is a technique represented in two parts: the visual, surface representation (mesh) of a character and a set of interconnected bones (skeleton) which are associated with some portions of the surface [15]. Then, the skeleton is posed and keyframed, while mesh goes alone with it. This technique gives smooth animations that do not suit the overall theme of the project.

The sprite sheet technique is more traditional. It requires that each frame of the animation have to be drawn separately. Sprite sheet animation changes which frame is rendered in quick succession to give the illusion of movement.

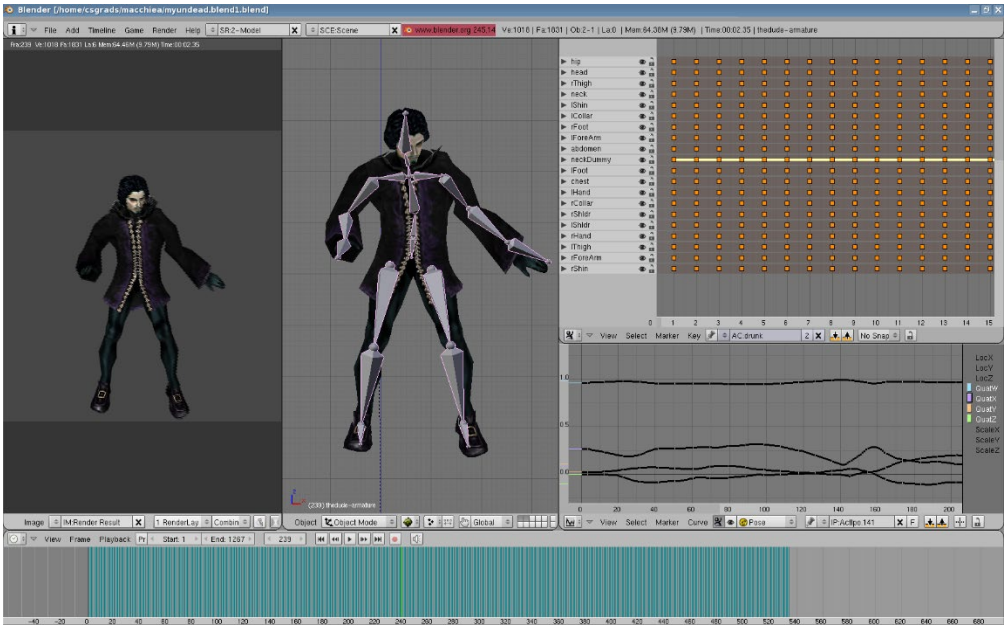

Figure 4.7 Skeletal animation, source: http://alumni.cs.ucr.edu/~sorianom/cs134_09win/anim.png

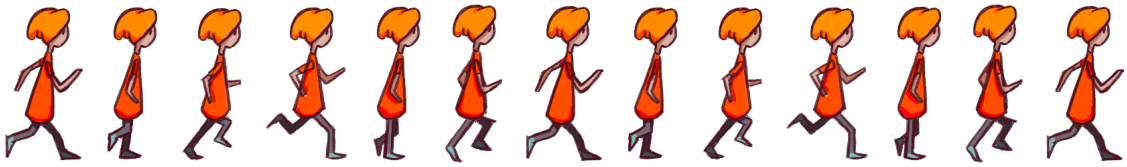

Figure 4.8 Sprite sheet used for animation, source: own

For this project, the complete, sprite sheet for each character was created [examples: figure 4.9, figure 4.8, and figure 4.1]. The animation for the main character [figure 4.8] contains: running and jumping. Encountered creatures' sprite sheets contain fewer frames because the project did not require excessive movement like in the main character case. Animation for Baba Yaga is subtle and contains most of head and rod movement as also change of height levitation.

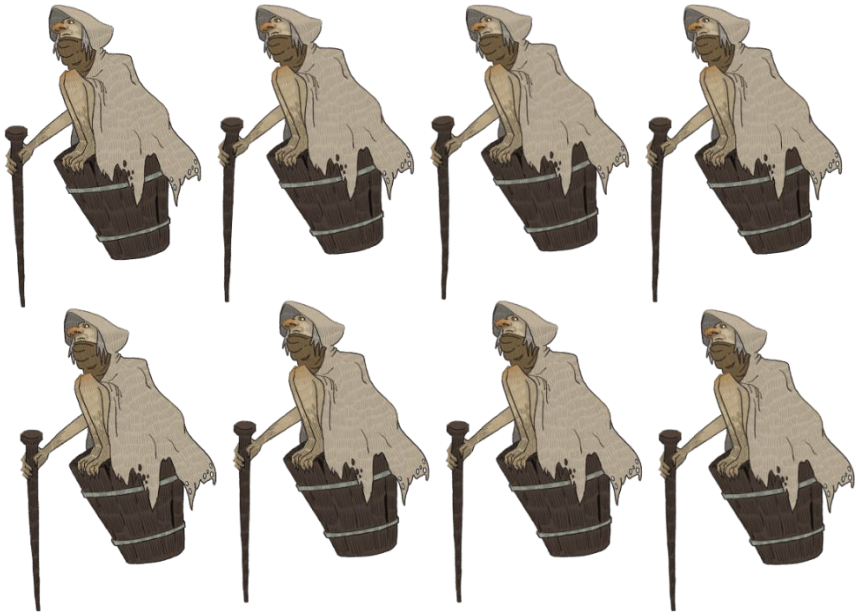

Figure 4.9 Sprite sheet used for Baba Yaga's animation, source: own

The creation of animation in the Unity engine begins with importing sprite sheets and dividing them into separate frames using Unity Sprite Editor. After that, the next step was to create an animation by selecting the necessary sprites for the Unity animation window [figure 4.10]. During this process, Unity creates an Animation Controller. This component is responsible for displaying animation, speed of animation, overlapping, delays, and many other attributes.

Figure 4.10 Unity animation window, source: own

Animation controller allows for a smoother animation transition while changing stages from, for example: running to jumping. All of the Animation controller components can be viewed and edited using Unity's animator window [figure 4.11].

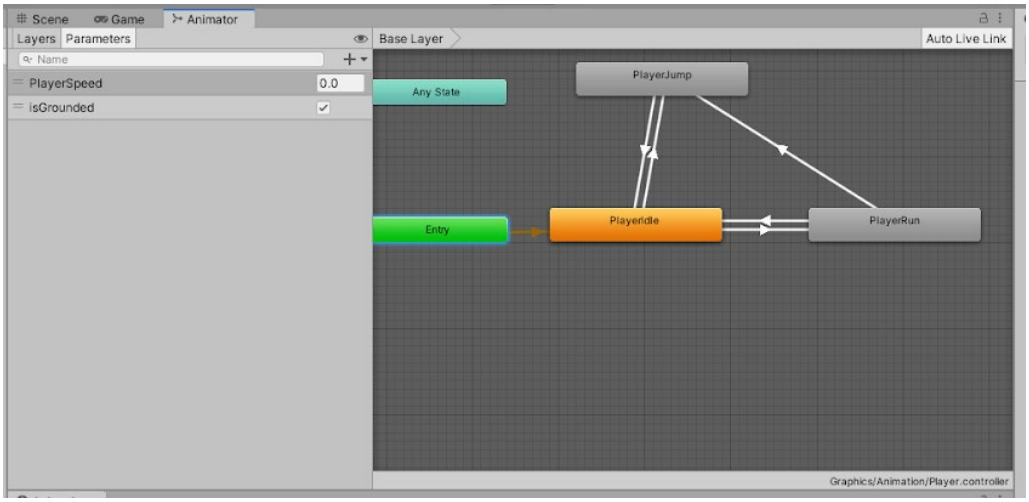

Figure 4.11 Unity animator window, source: own

Unity Animator window is divided into two parts: on the left side of the window is a list of parameters. They serve as an interface between the Animator Controller and the rest of the game. On the right side, different animations states for a given object are displayed. In this window, we can also access all detailed options for each animation and create transitions between them. Every such action is displayed as a one-directional arrow. Through the application, code parameters can be set during runtime. Figure 4.11 shows the animation controller created for animating the main character.

3.5.Audio

The audio layer is important for any multimedia project. Due to the limited time frame and lack of proper musical knowledge, as also a skill, music was chosen from external, available, free resources, listed in subchapter 4.1.7.

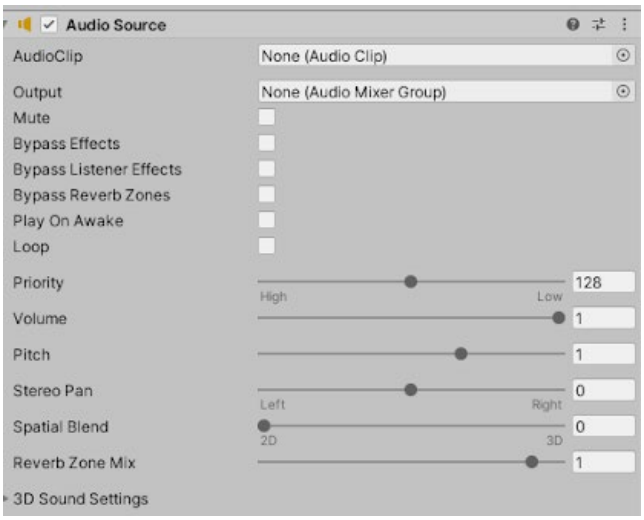

Figure 4.12 Unity's Audio Source component, source: own

Unity engine uses Audio Source component [figure 4.12] for introducing sound into the game scene. It controls starting and stopping playback of the clip and modifying other audio properties. The audiosource.PlayClipAtPoint is used to indicate for a player that they encountered a deity. This method does not need adding an Audio Source component to an object, instead, it plays a sound once from a specified position in the scene

3.6. Programming

For implementing the whole game the C# language and Unity design patterns were used [16]. One of the design patterns was Unity's Engine Event Function [17] which executes a number of event functions in a predetermined order, especially

- Update- Update is called once per every frame. It's the main function for frame updates related to display and player input.
- FixedUpdate- FixedUpdate is called more frequently than Update. While applying movement calculation inside FixedUpdate, there is no need to multiply values by Time.deltaTime due to FixedUpdate's is called on a reliable timer, independent of the frame rate.
- Start – Start is called before the first frame of the Update
- OnApplicationPause- OnApplicationPause is called at the end of the frame where the pause is detected.
- Awake- Awake is used called only once before the Start function, used for initialization of any variables or game state before the game begins.

There was also a need for the installment of the Android environment set up to ensure that game would run correctly on Android mobile devices. [18].

3.7.Encountered problems

During the implementation stage, different types of problems were encountered.

3.7.1. After jumping playable character gets stuck on the ground

Problem description:

After jumping playable character does not stay on the ground level. Instead, it sinks into the ground or starts the movement in the lower position than in the beginning.

Solution:

The solution to the problem was to set the collision detection on to continuous and add Physics Material 2D with zero friction to a Capsule Collider 2D. This made playable characters move more smoothly against the surface and stop them from being stuck in the ground.

3.7.2. The camera follows the playable character vertically during a jump

Problem description:

Since this game falls into the genre of side-siders, the moving along camera had to be implemented. Although camera moving along with the player vertically is not needed, because of the linear structure of a game, where getting into higher placed platforms is not needed.

Solution:

For this problem, the static properties of camera transformation were used. From code position, on every update per frame, a new Vectro3 was added. That allowed to set the camera on only the following playable character horizontally (on the x-axis), while the other two positions, y-axis, and z-axis, remained locked.

```
15      // Update is called once per frame
        — odwołania
16      void Update()
17      {
18
19          transform.position = new Vector3(target.position.x, transform.position.y, transform.position.z);
20
21      }
```

Listing 4.1 Source code for a horizontal camera movement, source: own

3.7.3. The player can move during riddles

Problem description:

During the encounter with the creature, the player can move freely while the riddle is presented. This serves as a problem as a player might pass the deity without answering the question and move along with gameplay.

Solution:

After the player triggers the point the movement is disabled during the riddle phase with setting Time.timeScale to 0. Due to the fact that Time.realtimeSinceStartup is not affected by Time.timeScale, there was a possibility to set up my own deltaTime for a UI riddle by subtracting the previous recorded Time.realtimeSinceStartup from the current one. After the player answers the question the Time.timeScale is set to 1.

3.7.4. The player can collect fireflies twice

Problem description:

Testers reported the problem that sometimes they can collect points for the same firefly twice.

Solution:

There was a need for creating a private function that will check if the point is collected. After OnTriggerEnter2D is completed with the collected firefly the object in-game gets destroyed.

4. Summary and further development

4.1. Results

The result of the implementation stage is a playable alpha version of a game, which fulfills the minimum set requirements. Due to the time restriction mentioned in subchapter 1.4, it was not possible to implement as complicated and well-balanced puzzles and riddles as planned. Although all the important features, such as animations, audio, user interface, scripts, and gameplay have been implemented and tested. To make the game feels more finished and well-rounded the implementation of better puzzles is needed, as also the addition of other creatures due to a short gameplay time.

4.2. Further development

The game is in an alpha version therefore finishing it is the main priority. The next step is to convert the game to other platforms, especially for personal computers, since games from the platformers genre have a high audience rate. There are also huge possibilities of expanding the game, by adding new characters and mythological stories limited only by the number of them. Adding more advanced puzzles and riddles. Which might be later divided into different difficulty levels for a larger audience. Creation of a more rounded story with a better explanation of deities and their place in Slavic culture.

4.3. Summary

The goal of creating the playable, early version of the game was reached. This version will serve as a foundation to build a fully functional game and for further development.

This game serves as a learning experience about the process of video game development and production. The whole process of creating the game was shown. From the general idea for a game, though designing process to implementation. Writing this thesis allowed me to learn more about the development of video games and the problems

that comes with them. The testing of the product was introduced at most of the stages of the game development to ensure that riddles are suitable for children and the process of playing brought them fun.

References

[1] C. Wright, *A Brief History of Mobile Games: In the beginning, there was Snake,* 2016 [Online]. Available: https://www.pocketgamer.biz/feature/10619/a-brief-history-of-mobile-games-in-the-beginning-there-was-snake/ [access 20.12.2020].

[2] J. Bycer, *Game Design Deep Dive Platformers*, CRC press, 2019.

[3] S. L.Kent, *The Ultimate History of Video Games*, California, 2001.

[4] S. Rogers, *Level Up!: The Guide to Great Video Game Design*, John Wiley & Sons, 2010.

[5] A. Kenzovic, *85+ Mobile Gaming Statistics for 2020 That Will Blow Your Mind*, 2020, [Online]. Available: https://www.blog.udonis.co/mobile-marketing/mobile-games/mobile-gaming-statistics [access 02.01.2021].

[6] L. Pscarella, M. Di Penta, F. Palomba, A. Bacchelli, *How Is Video Game Development Different from SoftwareDevelopment in Open Source?*, University of Zurich, 2018

[7] S. Swink, *Game Feel: A Game Designer's Guide to Virtual Sensation,* Taylor & Francis, 2009.

[8] Google Support Policy Center, *Designing Apps for Children and Families*, [Online]. Available: https://support.google.com/googleplay/android-developer/answer/9893335#1&2&3&4&5&6&7&87&9 [access 10.12.2020].

[9] B. Gunter, *The Effects of Video Games on Children: The Myth Unmasked*, Sheffield Academic Press, 1998.

[10] J. Bobrowski, M. Wrona*, Mitologia słowiańska*, Wydawnictwo Bosz, 2017.

[11] B. i A. Podgórscy, *Wielka księga demonów polskich*, Wydawnictwo KOS, Katowice 2005.

[12] W. Vargasa, P. Zycha: *Bestiariusz Słowiański*. Część Pierwsza i Druga, Wydawnictwo Bosz 2018.

[13] C. Hodent, *The Gamer's Brain: How Neuroscience and UX Can Impact Video Game Design,* CRC Press, 2017.

[14] J. Sobolev, *What Are Sprites And How They Work In Games?*, 2019 [Online]. Available: https://gamingshift.com/sprites-in-games/ [access 13.12.2020] .

[15] M. Soriano, *Skeletal animation,* Bourns College of Engineering. 2011 [Online]. Available: http://alumni.cs.ucr.edu/~sorianom/cs134_09win/lab5.htm [access 13.12.2020].

[16] M. Geig, *Unity Przewodnik Projektanta Gier,* Wydawnictwo Helion, 2015.

[17] Unity Technologies, *MonoBehaviour*, Unity Scripting API, 2020 [Online]. Available: https://docs.unity3d.com/ScriptReference/MonoBehaviour.html [access 05.01.2021]

[18] Unity Technologies, *Android*, Unity Scripting API, 2020 [Online]. Available: https://docs.unity3d.com/Manual/android.html [access 11.01.2021]

List of code listings

List of figures