

# ”Phantastic Wizard”

## Project Description

Lars Erik Faber, Studentnr. 193173



# Contents

<b>1</b>	<b>Requirements To Run Application</b>	<b>3</b>
<b>2</b>	<b>Unfinished Parts</b>	<b>3</b>
2.1	Game Files . . . . .	3
2.2	Application pages . . . . .	3
<b>3</b>	<b>Known Issues</b>	<b>4</b>
<b>4</b>	<b>App Description</b>	<b>4</b>
4.1	General Information . . . . .	4
4.2	Functionality . . . . .	4
4.3	Screenshots / Concepts . . . . .	5

# 1 Requirements To Run Application

## 2 Unfinished Parts

### 2.1 Game Files

#### The cactus enemy

For a while I had been working on another enemy type to mix up the gameplay. The idea for this enemy is that it is a purple cactus that will target a point in the arena room and grow outwards in four directions, separating the room into four areas. The player would then be able to destroy each of the four stems by damagin it with spells. The code is still left in the final product, but I only got around to drawing the spritesheet for it and coding basic behavior.

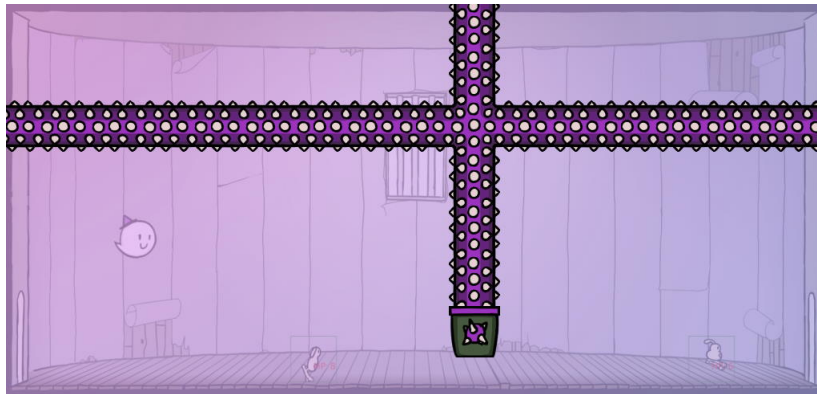


Figure 1: Concept for the purple cactus enemy

### 2.2 Application pages

#### The settings page

Originally, the plan was to let players adjust application settings and store them as settings configurations on the database. However, it seemed to overwhelming and unnecessary to connect all the settings details to the game, such as adjusting display mode, audio and keyboard button mapping. The configuration model class is still in the model and the database, along with the settings page, but they are not fully implemented. I also realized I could have utilized the settings page from Windows Template Studio, but at that point I had moved on.

#### The spell book

Originally, the intent was that the player would have to memories combinations of button presses to fire spells. For example, the ice-spell would have a combo of up-up-down-down-up using the arrow keys, but I did not get to code it. This xaml page would give a listing of all the different button combinations required to initiate each spell. However, now it just rests in the main menu and serves no purpose.

## 3 Known Issues

## 4 App Description

### 4.1 General Information

#### About the game

WizardGame is a small video game where players play as a tiny wizard ghost who can cast different spells. The objective is to survive as many waves of enemies as possible without dying, however this becomes more difficult as the waves progress. The enemy count and their damage output increases with the current wave, making it tougher and tougher to survive.

Another important place is the leaderboards. This is where players compete to get the highest wave, crowning them as number one on the board. The list is ordered by each player's personal best game in descending order. Players can also view their previous game to monitor their improvement.

For this to be possible, each player has the ability to create their own player profile that stores previous games, and will be displayed on the leaderboards. If they are unhappy with their decisions, they can update their player profile to give a more soothing name, or delete it altogether.

#### Assets and cosmetic

All the textures and sprites for the game (items, characters, backgrounds, logo, etc) are drawn by me using digital editing software.

### 4.2 Functionality

#### Gameplay

To move the ghost around use the W, A, S and D keys, and to cast spells use the up-, left-, down- and right-arrow keys. A wave progresses if all given enemies are eliminated. A normal wave consists of bunny enemies, and a special wave consists of card enemies and will occur every ten waves. If a bunny or card enemy touches the ghost, the ghost will lose some health points. Once the health points reach zero, the game is over and it proceeds to save the game to the selected profile.

The arrow keys will cast different spells that are used to damage and evade the enemies. The most useful spell is the ice-spell (up-arrow) that shoots horizontally in the headed direction and will deal a lot of damage to the enemies.

#### Application Pages

On launching the application, the user is greeted with a titlescreen displaying the name. Press any button left-click to proceed to the main menu. From the main menu, it's immediately possible to navigate to the settings screen, player screen, and spell book screen. Only once the database connection has loaded in the viewmodel, do the leaderboards and game become available. During the main menu it is possible to play around with the ghost character.

In the player page, users can select, create, update or delete any desired player profile. In the left panel, all existing players are displayed and ready to be handled by clicking on them. The selected player profile is each game will be tied to when playing. If no profile has already been selected, the rest api did not find a selected player, the application will instead select a default profile.

The leaderboards page will list all players and their best game (highest wave) in descending order. This is also the page that lists a user specific games history, and can be found by using the pivot headers. The page only serves as a monitoring tool to track one's individual progress.

The settings page and spell book page are not fully implemented.

### 4.3 Screenshots / Concepts

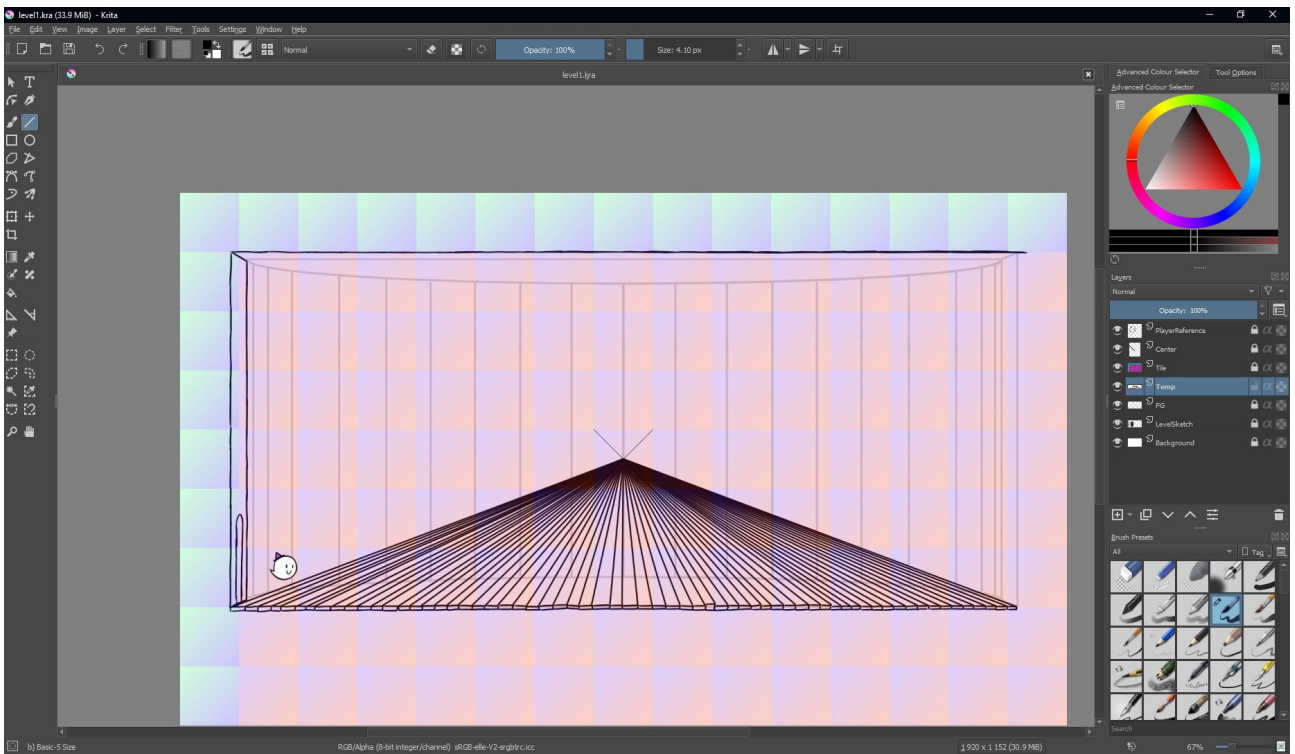


Figure 2: Early draft on the level background



Figure 3: Late development, displaying collision bounding-boxes

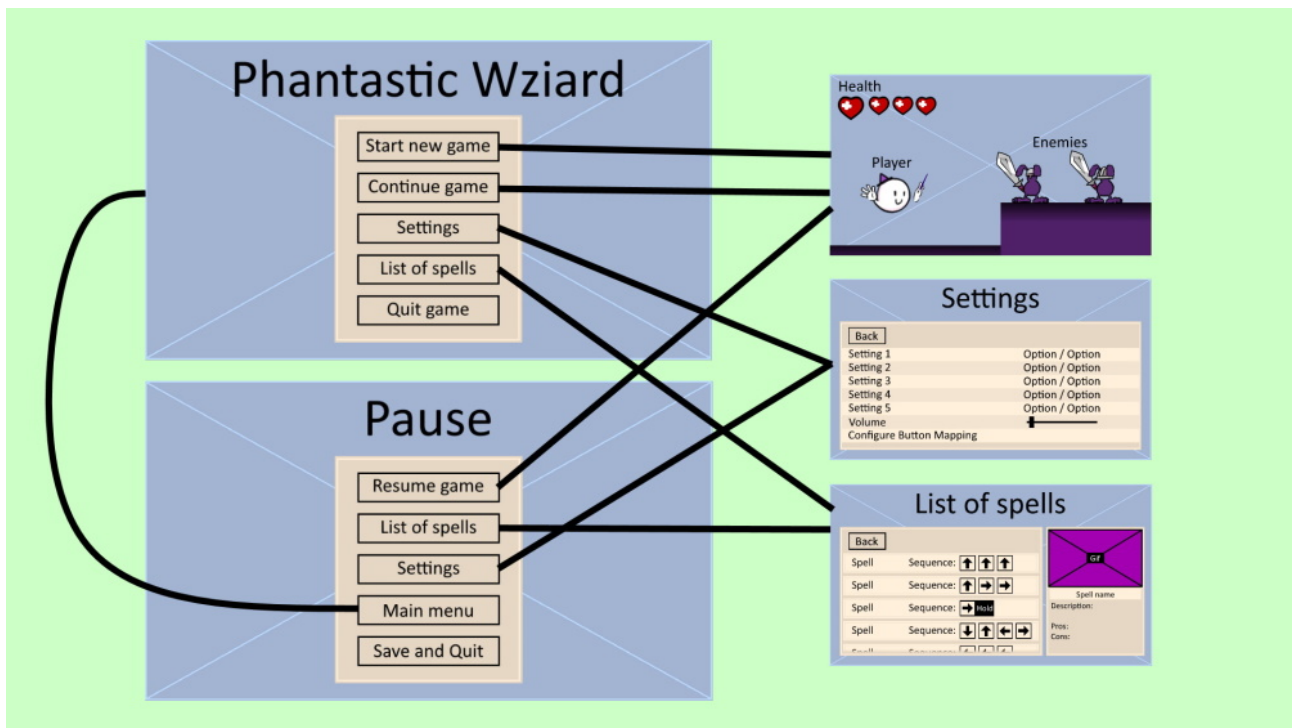


Figure 4: Menu structure sketch



Figure 5: First iteration of main menu