



Basic Information

Working title: *Wizard Duel*

Genre: *Arcade*

Number of players: 1-2

Premise

A simple arcade game intended for two players playing on one larger device, such as a tablet, or a large touchscreen. Each player operates from one side of the screen, taking the role of a *wizard* in a magical duel. A player is able to unleash *fireballs* with a swipe gesture, releasing the *fireball* in a direction based on the swipe trajectory. The *fireballs* travel in said direction, bouncing off walls, until they reach the opposite end of the screen, either hitting the opponent and dealing damage, or missing them. The amount of *fireballs* at either player's disposal is limited, but recharges continuously. If two *fireballs* hit each other, they both fizzle out.

To defend themselves, a player is also able to conjure a *shield* with an outwards pinching gesture. This *shield* is maintained as long as the player holds their fingers down and the player may not throw *fireballs* while they have a *shield* active. A *shield* can only be upkeep for a limited time, as it depletes when used, and recharges when not. Additionally, the larger the *shield*, the faster it depletes. Any *fireballs* hitting a *shield* fizzle out.

When a player's *wizard* is hit by a *fireball* enough times to deplete their health points, they lose the game.



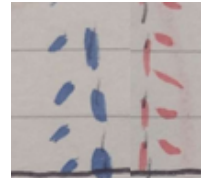
Figure 1. Sketch of the main Game Screen, with an example of a game in progress.



Game Elements

Casting Area

Represented in Figure 1 by coloured dashed lines with angled stripes coming out of them, each player a casting area, in which they can cast their spells. Any touch registered in each *wizard's* respective *casting area* is considered to be done by that side's player and so any spells thusly conjured are of their colour.



Walls

A double line represents a *wall* in Figure 1. *Walls* are the top and bottom edges of the screen when in game. When a *fireball* hits a *wall*, it is reflected in a mirrored angle.



Fireball

A *fireball* is the means of attacking one's opponent and winning the game. A swiping touch gesture is used to release a *fireball*. A *fireball* of a player's colour appears when a finger is pressed down in that player's *casting area*. This *fireball* will follow the location of the finger pressing down until it is either released, or let go and allowed to fizzle out.

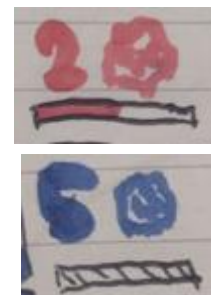


Note: The left image showing the representation of fireballs in Figure 1 has a star symbol. This symbol represents a player's finger pressed on the screen in that place. The symbol also appears in the Shield section.

If the player makes a valid swipe gesture, the *fireball* is released. The direction in which the *fireball* will travel is determined by the swipe gesture's trajectory. Only a final portion of the trajectory, taken from the last few frames before release, determines the direction. If the distance traveled by the finger in that segment is not long enough (i.e. the finger moved too slow or wasn't swiping), the *fireball* will fizzle out.

A *fireball* travels in a straight line until it hits a *wall*, a *wizard*, another spell, or the empty space behind a *wizard*. Hitting a *wall* reflects the *fireball* in a mirrored angle, allowing it to continue. If a fireball hits a *shield*, it fizzes out and is destroyed. When two fireballs collide, both of them fizzle out. Reaching the left or right edge of the screen, a fireball can either hit a *wizard*, dealing damage, or miss them, disappearing in the empty space behind them.

Each player has a limited, but continuously recharging, amount of *fireballs* they can throw. The current amount is displayed on the UI at the bottom of the screen, on the respective players' sides, as a number, accompanied by the icon of a *fireball*. Underneath is a bar, which visualises the progress of recharging a new *fireball*, represented by the filling bar. When the bar fills, the amount of *fireballs* available is incremented and the charging resets. There is a maximum amount of fireballs one





can have available. When that maximum is reached, the recharge bar becomes greyed out, or otherwise visually signifies that it is not currently charging.

Shield

A *shield* is created by a player when they perform a pinching gesture. The fingers performing the gesture must start next to each other, then pinch outwards, away from each other. The *shield* will appear between the two fingers, and exist for as long as the fingers stay on the screen.



The *shield* serves as defence, preventing *fireballs* from passing through and hitting the player's *wizard*. If a *fireball* comes into contact with a *shield*, it fizzles out. However, while a player's *shield* is present, that player cannot throw any *fireballs* of their own.

Like fireballs, the *shield* is also a limited resource. A vertical bar with a symbol of a shield underneath shows each player the amount of *shield* they have available. This bar depletes when the player is holding a *shield*, but recharges when they aren't. The bar depletes faster the longer the player's held *shield* is.



Wizards

The persona of each player in the game, it should be obvious where they are in Figure 1. The *wizards* serve as targets and hit zones for *fireballs*; the goal of a player is to hit the opposing player's *wizard* with *fireballs*. When a *fireball* hits a *wizard*, they take damage, losing one point of health, but the game continues until either *wizard* has been hit enough times to lose all their health points. If a *fireball* passes a *wizard* all the way to the edge of the screen behind him, the *wizard* is safe and the *fireball* fizzles out.



A *fireball* is just as dangerous to the enemy *wizard*, as it is to a player's own. This means that a *wizard* will lose a health point even if hit by their own *fireball*, though this is very unlikely to happen by accident.

The *wizard's*, and therefore the player's, health points are represented by the symbols of wizard hats at the top of the screen above their respective *wizards*.





Mechanics & Numbers

Note: The following values are initial and are likely to be adjusted and iterated upon for better balance once the game can be tested.

Fireballs

Max number of fireballs: 5

Time to recharge 1 fireball: 2 seconds

Travel speed of fireballs: $0.25 * \text{screen width}$ per second (approx. 3 seconds for a *fireball* to fly directly from one side to the other. Longer if directed to bounce.)

Shield

Max shield time charged: 10 seconds

Shield recharge speed: 1 second per second

Shield depletion multiplier: 1 per $0.1 * \text{screen height}$ (e.g. if a shield the size of half the vertical screen size is created, the multiplier would be 5, effectively draining a fully charged shield in two seconds)

Wizard

Health points: 3

Size: one third of the screen height

Note: an alternative idea is to only have 1 hit point, with an immediate restart and score keeping mechanism. This may be shifted towards if playtesting reveals multiple hit points to not be as engaging.

Menus, Screens, Navigation

The general flow of the application starts with opening the game and loading into a main menu. From the main menu, the player can either open Options, or press Play. The Play button does not immediately start the game, but instead presents a Player Setup menu, from where the game can be properly started, or one can return to the main menu. See Figure 2 below for a visualisation of the navigation flow.

Because the game is designed primarily for larger screens (tablets or larger), some of the screens (Player Setup in particular) can be busier than a standard phone application might be.

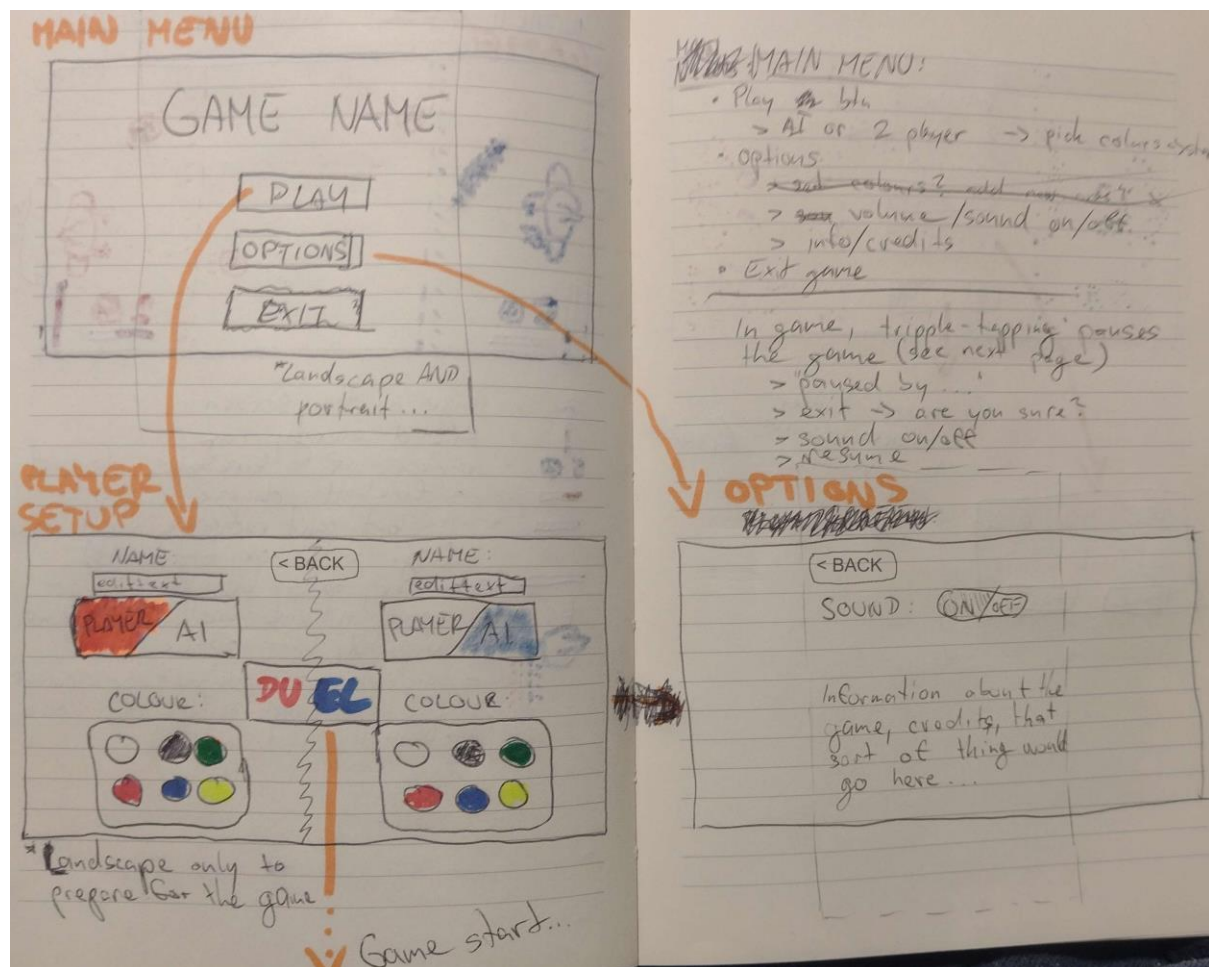


Figure 2. Sketch demonstrating the general flow of the game's screens.



Main Menu

The Main Menu screen will have a simple layout, with the game's name at the top in large letters, followed vertically by three buttons: Play, Options, and Exit. Play takes the player to the Player Setup screen, Options to Options, and clicking Exit closes the application.

This screen will respond to the phone's orientation and have a layout for both portrait and landscape mode. The simple vertical layout will remain the same, with different button sizes and spacing.

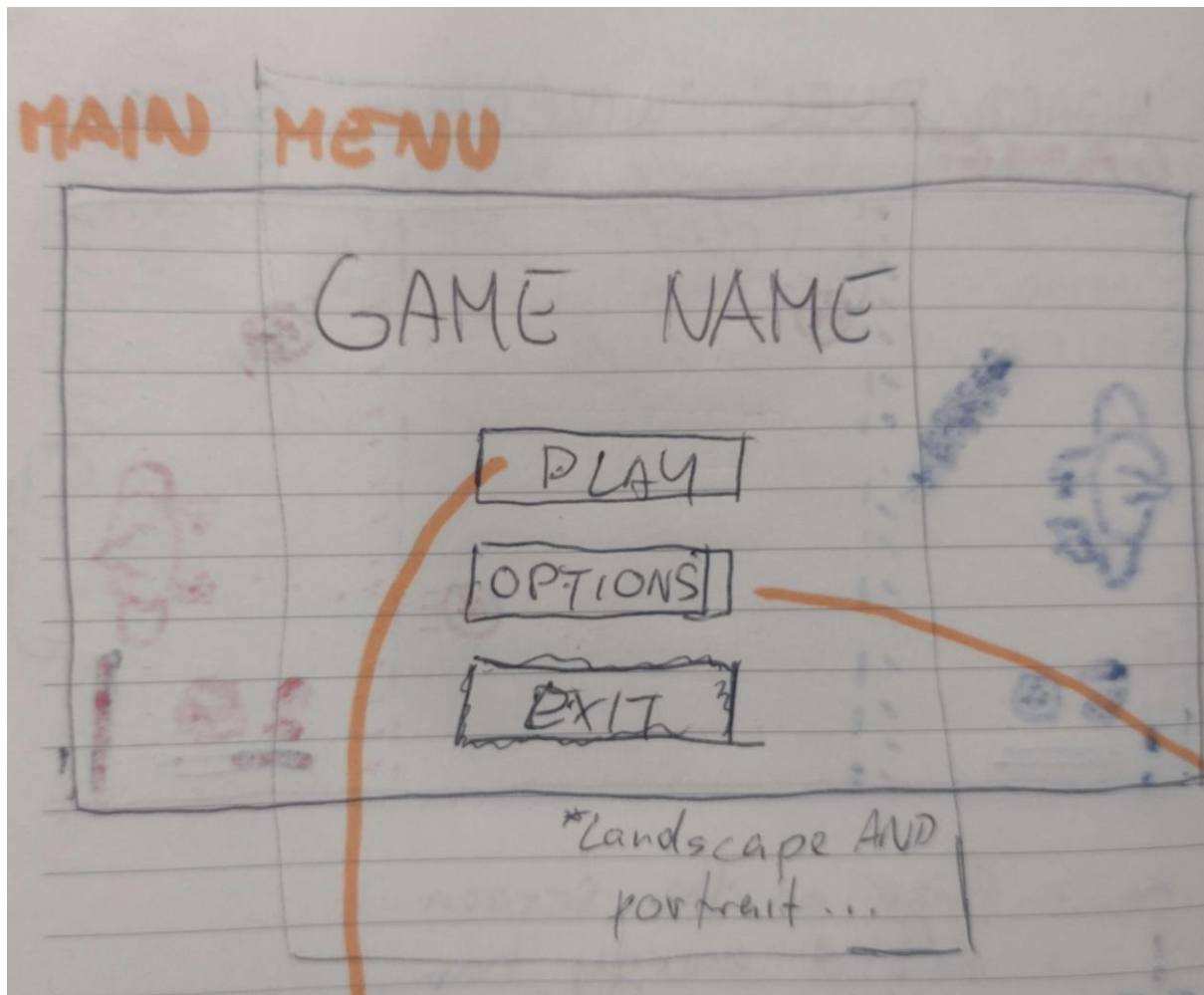


Figure 3. Sketch of the Main Menu screen.



Player Setup

This screen serves to set up the game. There are three components, mirrored on the left and right, for each player. First, at the top, is the name of the wizard. Underneath is a switch, determining whether that side's wizard will be controlled by a player, or an AI. Finally, at the bottom, is a swatch of colours to pick for that wizard. Each wizard must have a different colour.

Once the set up is complete, clicking the Duel button will start the game proper, transitioning to the Game Screen. Alternatively, one may hit the Back button to return to the Main Menu.

The Player Setup screen is forced to be in landscape mode. This is because the game is designed to be played in landscape mode and this screen is one step away from the game itself. Forcing it into landscape mode ensures the player(s) will be in the correct mode and ready once they press Duel.

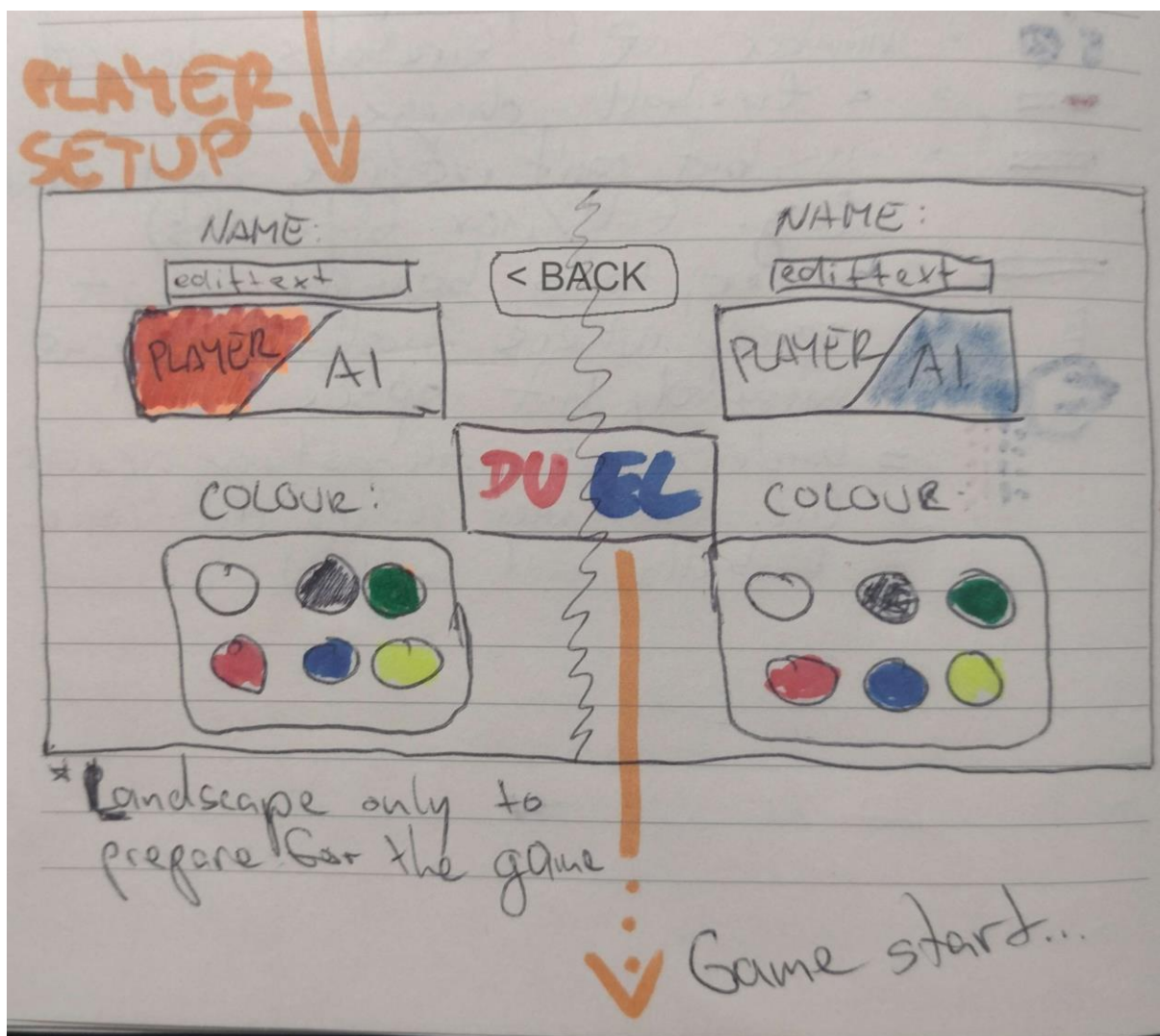


Figure 4. Sketch of the Player Setup screen.

Game Screen & Pause Menu

A visualisation of the game screen can be glanced in Figure 1 at the top of the document, alongside an explanation of the game and what is happening on this screen. The Game Screen is always in landscape.

During play, either player may pause the game by triple tapping the screen, bringing up the Pause Menu. The Pause Menu overlays over the Game Screen, with the paused game in the faded background. At the top it says “Paused by” followed by the name of the player/wizard, determining by which half of the screen the pause gesture was made in. Below is an option to toggle sound, then buttons to Exit to main menu and to Resume play. If the game is resumed, a countdown will show for a number of seconds before the game properly resumes. This is to give the player(s) time to ready themselves to continue playing. If the Exit button is pressed, a modal dialogue will open asking to confirm the action, in case the button was hit by accident. The game will also automatically pause if the application process transitions into the background.

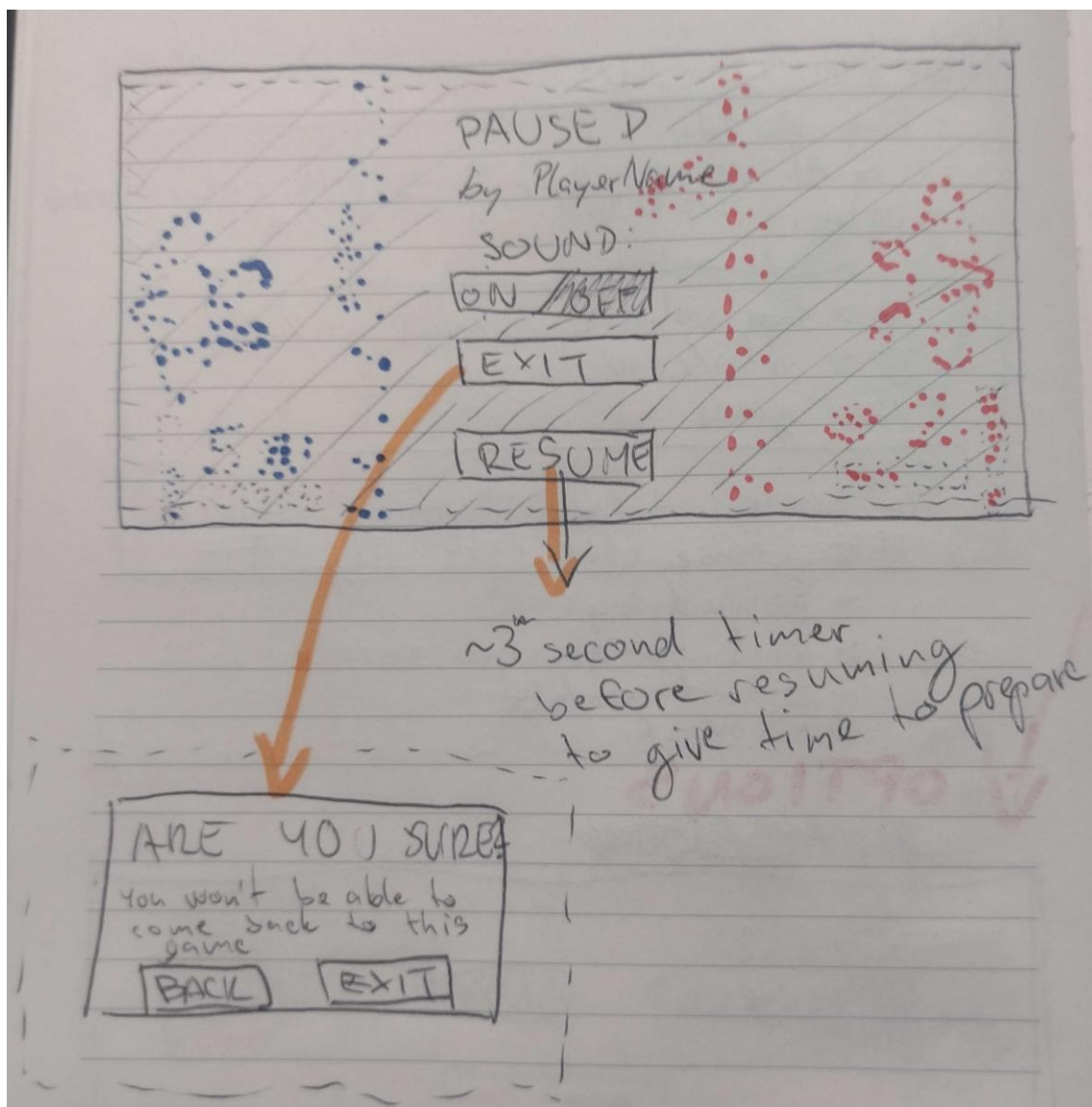


Figure 5. Sketch of the Pause Menu and a confirmation dialogue to exit the game.



Options

The Options screen is not very busy. A Back button in top left leads back to the Main Menu. Below is a toggle for enabling or disabling sound. Finally, at the bottom, there is information about the game, any potential credits, licenses, or other information relevant to the user.

This screen supports both landscape and portrait modes.

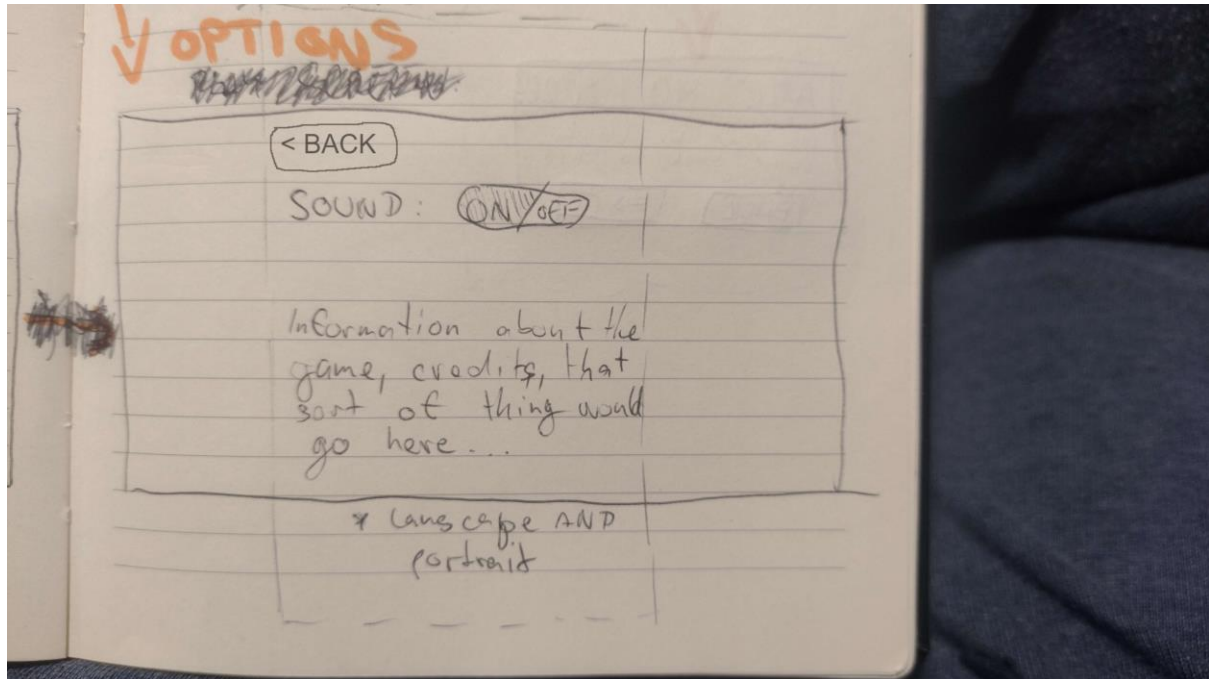


Figure 6. Sketch of the Options screen.



Prioritisation of Features

This document pertains to a number of features that can be considered individually. Due to the nature of the project as a university assignment and the limited time available to work on it, it is possible that some less important features may not be implemented in the end. Below is a list of said features, prioritised using the MoSCoW system. It is expected that both “must” and “should” features will be implemented, with the “could” ones considered as stretch goals.

- The game screen **must** be fully functioning, with a playable game, as designed in this document’s first three sections.
- The game **must** have all the specified screens with fully functioning transitions between them.
- The game’s state **must** be saved when the application loses focus and restored when focus is regained.
 - *This does not apply to the application being killed.*
- The game **should** have a functioning colour picker for the wizards.
- The game **should** have functioning soundtrack, which can be toggled on or off.
- The game **could** have SFX when throwing fireballs, holding shields, bouncing fireballs off walls, hitting the wizards, or fireballs colliding together. (*Stretch goal*)
- The game **could** have an option for wizards to be controlled by artificial intelligence, as selected on the Player Setup screen. (*Stretch goal*)