System Design

The game relies on two main scenes, the main menu scene and the game scene.

In the main menu, the player is given a landing page in order to give them time to enter the game instead of throwing them immediately into the action. It also is a menu that allows players to see extra information about the game and adjust any settings to their preferences.

In the game scene, the main action of the game takes place, where the player character is able to run and dodge obstacles. When new players access this scene for the first time, a small tutorial appears to guide the players with the controls. In this scene, you are also able to pause the game to take a break for any reason, or return to the main menu. When the game is over, a game over menu appears over the game scene, enabling the player to play again or return to the main menu.

The high score and coin collection functionality of my game requires that I use saving and loading data as a part of my application. In order to achieve this in Godot, I have used a dictionary system to serialize all of the data necessary in order to make outputting and inputting from a file easier. This was fairly easy to do as much of the important data is fairly simple, being primarily Booleans (Tracking new players, accessibility toggle) or integer values. (High score, coin count)

The other consideration made with saving and loading was when to conduct save and loads. To simplify this as much as possible, I used Godot's global variables function so that the necessary information is constantly available while the game is active, to prevent constant loading. For saving, I made sure that information was saved only when changed, for example, at the end of a run when a high score is actually set. Limiting these save and load calls hopefully serves to prevent performance issues in the long run while preserving the information we need to maintain.

UI Design

In order to manage responsiveness, I used a mixture of Godot's inbuilt stretch modes to scale along with properly constructed Parallax backgrounds that would extend to any screen type.

I also used control nodes, which are used for the majority of UI, and have built in anchor modes, which allows me to ensure that the buttons maintain their place relative to the size of the screen.

When designing the layout, I kept in mind the usual layout of similar mobile games, which will be familiar to the player and so easier to navigate. This influenced decisions

such as putting the pause button in the top right corner, putting the accessible jump and slide button in the bottom right corner, and putting the pause screen in the centre of the screen, overlayed over the game.

For game controls, I went for a swipe mechanic in order to make the controls more immersive, allowing the player to feel more connected to the movement of the player character. However, I am also very aware that these types of controls can be very difficult for some players to manage, especially as the game gets faster. Due to this, I added an ‘accessibility mode’ which enables buttons displayed on the screen that can make the player jump up and slide as well.

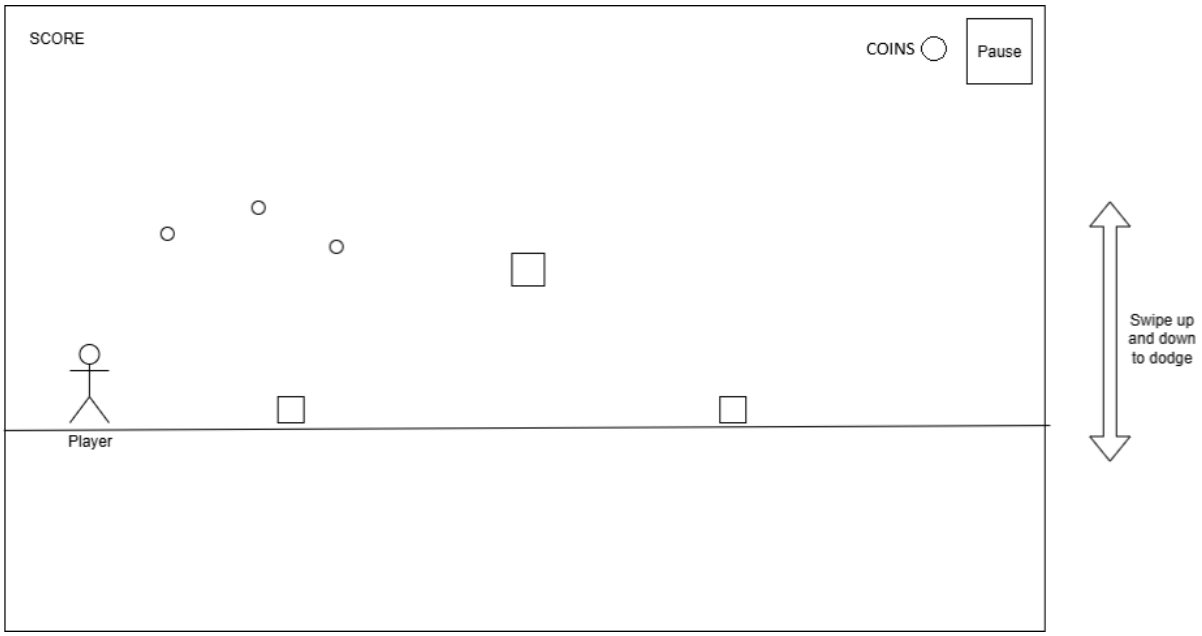
Wireframes

Main Menu

Players can play a run, or choose to upgrade with previously earned coins (if implemented)

Main screen of the game, where the runner has to dodge obstacles.

Circles are Coins, Squares are obstacles



Death Screen

Displays the users final score of the run, their high score and current coin total. Also allows the player to play again, view the leaderboard or go to the upgrade shop to spend their coins (if implemented)


GAME OVER
SCORE
HIGHSCORE
COINS
Upgrades
Leaderboard
Run Again