**Production log**

I started by making a plan for adding the criteria, this is what I needed to include.

* + Technical, tools and features
  + Programming fundamentals
  + Variables, if statements, loops, arrays, functions, and classes
  + Gameplay loop
  + Mechanics

**What is a function?**

A function is a way of packaging code that does something and can return an int, float or string value.

**What’s an if statement?**

An if statement is Use if to specify a block of code to be executed if a specified condition is true.

Text

Description automatically generated

**I used array for to determine when the platform reached its target.**

**What is a loop?**

A loop repeats a statement or a group of statements while a given condition is true.

Text

Description automatically generated

**I used a loop here to update my timer.**

**For loops**

For loops are, loops that execute a block of statements repeatedly until the specified condition returns false.

Text

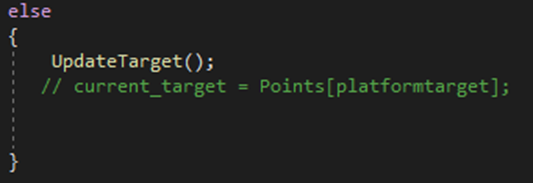
Description automatically generated

**while loop**

A while loop is a loop which goes on as long as some condition is true.

**Else statements**

Else statements check a Boolean expression and execute the code based on if the expression is true or false.



**I used an else to update the platforms target.**

**What’s an Arrays?**

Arrays make it easier to organize and operate on large amounts of data.

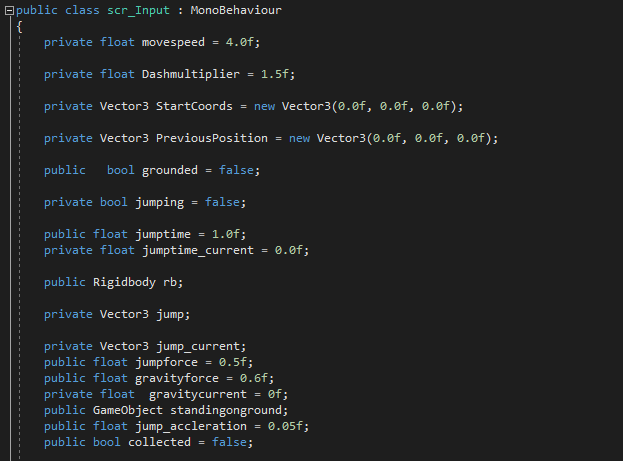
Text

Description automatically generated

I used this to keep the platform on the right axis.

**What is a class?**

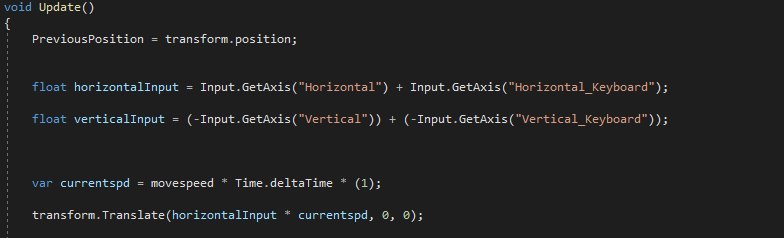
classes support polymorphism, inheritance and provide the concept of derived classes and base classes. With classes you can also use overriding functions which allow you to output a different value depending on the input factor.

****

**Classes keep things together so you can call code.**

**What’s a Variable?**

A Variables represent storage locations. Variables can be saved as float, string and int. String is words, phrases or even paragraphs. Floats are numbers that can/have a decimal point. Integers are whole numbers.



**Collisions.**

When collisions occur, the physics engine or code calls functions with specific names on any scripts attached to the objects involved. for example, my character is colliding with the platform.

A picture containing candle

Description automatically generated

I started by making my own physics and collisions which covered if statements variables classes and functions.

Text

Description automatically generated

I defined the input in the update function so that it updated every frame. Then I started to code the movement speed. I did this by setting the value of “currentspeed” in the editor. Translate means you will move in the direction that is being changed, in this instance, the x value is being changed.

Text

Description automatically generated

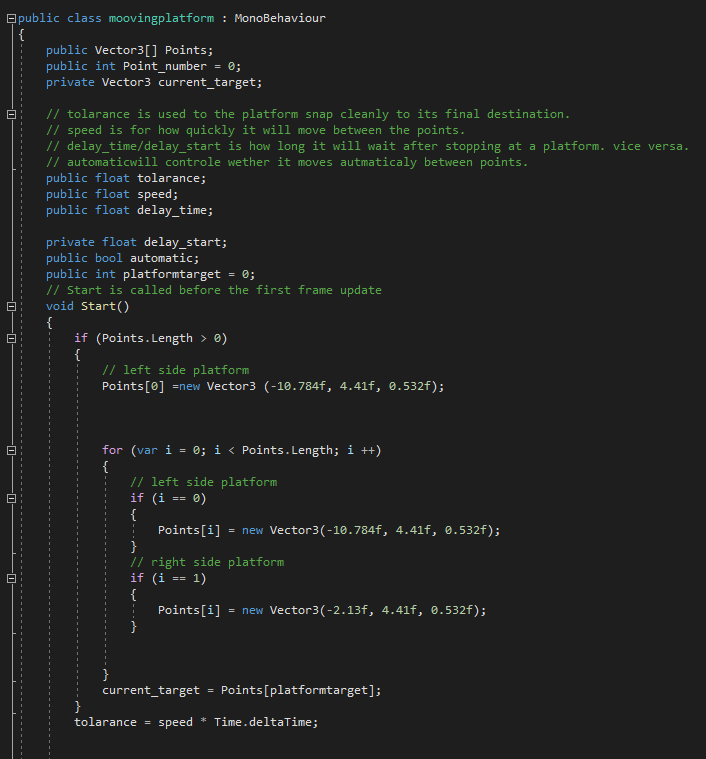
I then coded the jumping using the floor coordinates as a collider and then defined whether you were on the floor or not. I used an if statement with an else statement because I needed jumping to realise when it wasn’t jumping.

Text

Description automatically generated

I then coded the collectable so players could pick it up. The eventual goal was to make it collectable that you needed to collect to progress, but I Ran out of time to code this mechanic. Here I used an if statement to destroy the object when you collided with it. I made my own assets in 3ds max. I also added a check point using my own assets from 3ds max. While the player is in the air, grounded is set to False. This is done by using OnTriggerExit which will output code when you exit a trigger.

I decided I wanted to add a moving platform which used a for loop so that it bounced back between each coordinate.



I then made the code for the moving platform. I defined the points it needed to move between and then set the speed. I made a class for my moving platform that pulled code from the update function.

Text

Description automatically generated

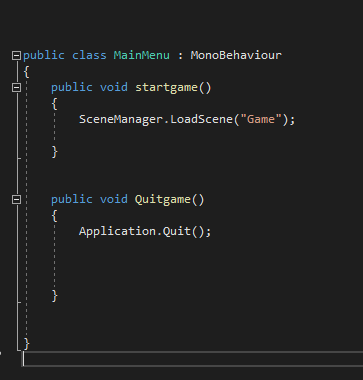
I then made the code for the moving platform. I defined the points it needed to move between and then set the speed. I made a class for my moving platform that pulled code from the update function.

Text

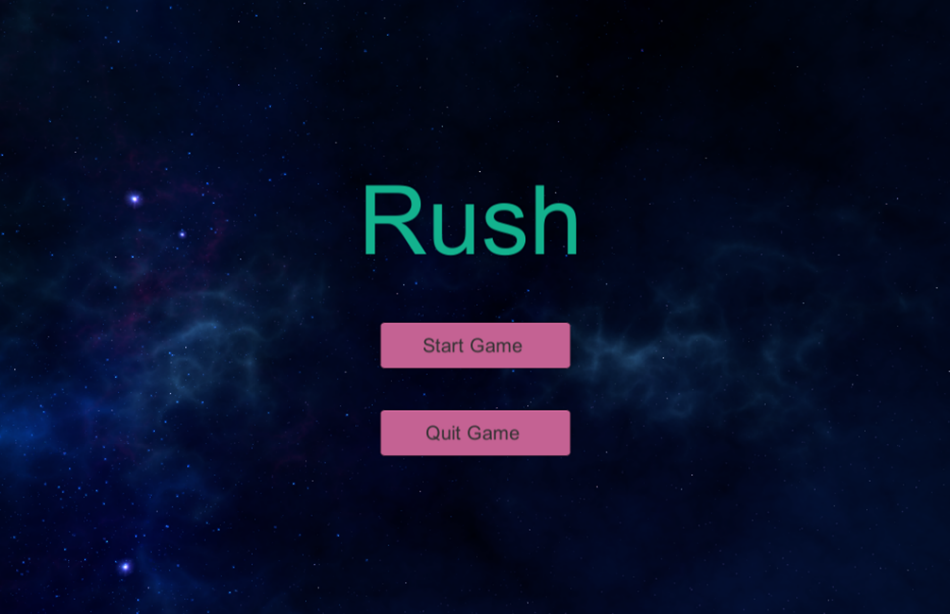
Description automatically generated

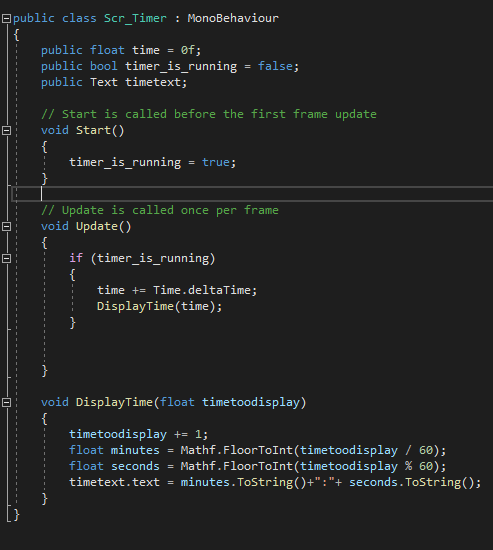
I used a YouTube video to help make the moving platform. (<https://youtu.be/9KdY4mafG_E>)

I then needed to add UI and UX so I added a menu and a timer.



I used unity’s editor and linked it to code. Application.Quit will exit the game and close the application if you are in a build. SceneManager.LoadScene will load a new scene that you have created





I used unity to make the timer and the menu screen adding it to my canvas and then programming the buttons in visual studios.

A screenshot of a video game

Description automatically generated with medium confidence