Developer Diary - Ink Run

Antaine Ó Conghaile - G00347577

# November 16th -

To start this project I created a plane so I would be able to test the basic **PlayerMovement.cs** script I added. For this purpose I added a cylinder to serve as the basic player model for early testing. The first thing I made sure the script did was move the player forward at a constant rate regardless of player Input. This is used in most Endless runners like **temple run** I came across in my research. The other aspect I implemented was that the player would only take Horizontal input that moves them a fixed distance from left to right.

This will allow the player to quickly avoid obstacles later on in development as well as making spawning simpler as there will only need to be 3 locations on the x-axis. The first hurdle was that the player will eventually fall off the plane whether that is after running too far or falling off the sides.

Because of this I will look into the optimal way to spawn an endless track and update the movement to not allow the player to move too far to the sides of the track without falling.

# November 24th -

I have updated the **PlayerMovement.cs** to only allow the player to move a fixed amount remaining inside the 3 desired lanes using if statements and managing the current lane. I then added a **CameraFollow.cs** script that makes the camera follow the player at a fixed distance which will be vital when the track is spawning and the player moves further in the game.

I also added a **Jump()** method which allows the player to avoid smaller obstacles by jumping over them but not being able to move left or right unless the player is grounded. I added a custom gravity to the player to prevent the player from rising and allow me to tweek the jump and gameplay more to suit the game which will be pivotal in smooth gameplay

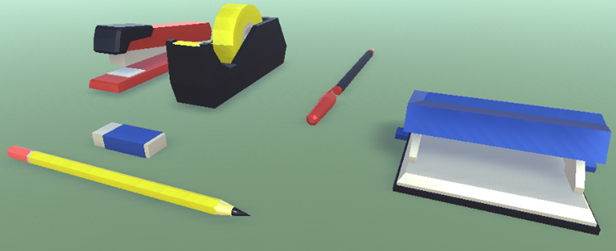
# 

# November 29th -

After researching different methods to spawn an endless track I decide to create tiles prefabs that will contain a floor object that will spawn a fixed distance away from each other at the start of the game and spawning more and deleting the ones behind the player as the player moves from one tile to another based on the player distance.

In order to make these I used **Blender** to create a notepad asset that would serve as the floor for the track in all tiles. This took longer to make than expected so I will try and use free assets from the asset store for models as this takes too much time. I will talk to the designer about using simpler assets at a later date when it is relevant.

I created a prefab of tiles and added several objects referenced in the design document from the [asset store](https://assetstore.unity.com/packages/3d/props/office-supplies-low-poly-105519). These will serve as obstacles that the player will have to jump, avoid and slide under during the gameplay. I will have to add death collisions later on to test these more.



Tiles now generate infinitely as long as the player continues moving forward while spawning random tiles from prefabs making the track different on every instance of this game using the **TileMange.cs** script.

# December 2nd -

I added the first of the UI elements with the Menu scene that allows the player to load the first level scene ‘**Level1’** and quit the game using the **MainMenu.cs**. This needs to be stylized but I have decided to leave this till after the functionality is completed. I added a simple options menu panel with no current functionality that will be added later with menu music.

I talked to the designer over email about possibly needing to focus on the functionality and to use simpler UI elements and models and that I will try to implement later on if I have enough time.

# December 3rd -

I added a basic score system that currently takes a player's distance to update a score element on the hud in game that updates as the player moves using the **Score.cs** script. I will have to change this later to be a timer however I am concerned that pausing and restarting the level on player death will cause errors with this so I will implement this after those features.

I need to add a Game Over condition and a Game over screen that will trigger in the death sequence. I must improve the hitboxes for better collisions and more tiles to give the game a more unique feel and look so the gameplay is less repetitive.

# December 5th -

I created an Ink material for the project to allow the game to have a more aesthetically pleasing look that will match the concept art in the game design document more. I decided to do this because it was a simple and fast way to improve the design without spending a decent amount of time and can be used for the player model, Ink Eave and the text Elements.

I added a pause menu to the game triggered by the ‘P’ key. It stops the time in game and activates the pause panel allowing the player to resume or quit the game. It also has an options menu with a slider for audio later on.

I talked to the designer over email about adding a quit button to the pause menu so the player can leave the game without returning to the main menu and he agreed this could be added.

I will add the wave of Ink to test the player death collision before implementing it on the objects on the tiles as they will require more testing as they can be avoided. I am considering the point of the ink wave as if players are killed by obstacles instantly because the wave would not catch up to the player.

# December 7th -

Implemented a simple death sequence triggered by the wave of ink that is constantly chasing the player as it’s shadow becomes visible the closer it gets to the player that restarts the level on collision.

I improved the hitboxes of certain obstacles to avoid the player clipping the obstacles so it doesn't collide with the player in certain situations such as jumping and crouching. I need to add more tiles to the prefab to make the level gameplay look more unique as it will be repetitive with the exact same objects. review the scope of online multiplayer.

I added a **crouch()** function that reduces the player hitbox allowing the player to crouch under the hole puncher and avoiding the collision and continuing through the map in the **PlayerMovement.cs**. I also updated the movement to appear more fluid despite being rigid in moving a fixed rate

I added a death screen that appears when the player is killed that contains their score and allows the player to quit or retry the level.

# December 8th -

I added a transition condition in the **Score.cs** script that allows the player the options to progress to the next level or resume the game after moving a fixed distance. Added a pause button to the hud element of the game as specified in the game Design Document.

I created a **DeskManage.cs** that generates a plane with a wooden floor texture to serve as the background of the game making it look like the player is running on a desk. This works like the tile generation with spawning at a certain distance and having different objects on the table to give it a used feel.

# December 9th -

Added a **SwipeMovement.cs** script that allows the player to be controlled byswiping the screen rather than pressing a key as it will be used on mobile. I then added a death collision from the obstacles. I added a ninja player model then added the Ink material making it look more like the player model in the Design Document.

I then added an animator with a basic running animation to the model as the player moves as well as a simple jump and crouch animation. I improved the UI elements on the menus to match the aesthetic suggested in the Design Document

I added a start screen that requires the player to tap the screen before the game begins allowing the player to be prepared to start as objects can spawn close to the player.

# December 10th -

I improved Tile and Desk spawning to remove clipping issues where texture would overlap. Changed the Obstacles and made it so the death collision is only triggered by colliding with objects with Obstacles killing the player to avoid any power ups triggering this.

I changed the score to record time rather than distance and did not encounter the errors I was concerned about. I added a shield power up that spawns using the **Shield.cs**. This allows the player to have one free collision without dying.

I encountered a bug where it did not work because the game detected multiple collisions before the player had a chance to move out of the way. To fix this I used the **Invoke()** that slows the player down but does not remove the shield until movement is restored giving the player a second to move out of the way.

I updated the game to fit a portrait 16:9 ratio as this is what most phones tend to use. I also added a **Highscore.cs** script that uses **PlayerPrefs** to store the highscore and displayed it on the leaderboard panel on the **MainMenu scene**.

# December 11th -

Added a level select to pick the level select and a basic lobby to the game. Added a highscore name that is stored with the high score based on the name inputed by the player. Added background music to the menu.

# December 12th -

I added a shield icon to the game hud that displays when the player has the power up and disappears when the shield is used. I talked to the designer over email about this. I added music but then changed this by implementing an audiomange with an **AudioManager.cs** script that manages all my sounds allowing to call them in other methods.

With his I was able to add sound effects throughout the game and added a main audio mixer to the game audio sliders. I then added the mirror to the project to allow the project to be run a multiplayer on a unity framework. Fixed some errors found in testing and a multiplayer menu.

# December 13th -

I added [ParrelSync](https://github.com/VeriorPies/ParrelSync) to the project which clones the project to have two editors open to test multiplayer. I created level 2&3 based on level with different aesthetics and objects. This was easy as the **TileManager.cs** and **DeskManager.cs** were reusable to create these levels. Added minor fixes.

# December 14th -

Added a multiplayer lobby using Mirror that allows the player to host or join a match. Talked to the designer about replacing the multiplayer lobby with a UI implemented by mirror to allow players to join because of time constraints.

It allows the players to connect and play the game however it is not finished and the camera does not follow the player. Possibly because there is a multiplayer player model or the network code. Added comments and uploaded Testing to github as well as editing the developer diary

# December 15th -

Uploaded final test cases, video and final Developer Diary.

Link to Github: <https://github.com/Antaine/MobileProject>

## Resources: [ParrelSync](https://github.com/VeriorPies/ParrelSync) ,

## <https://mirror-networking.com/>

[Menu](https://www.youtube.com/watch?v=zc8ac_qUXQY)

[Mirror](https://www.youtube.com/playlist?list=PLS6sInD7ThM1aUDj8lZrF4b4lpvejB2uB)