Helix Jump Prototype

Name: Olaf van Dijk

Date assignment received: 08/12/2020 evening

I started the assignment by doing some research first. I had never played Helix Jump before so let’s start with that.

While playing the game I wrote down a few things that make the game as it is:

Key features:

* Fall and bounce until you reach the goal
* Control platforms/player with touch controls
* Don't touch the coloured areas
* The more gaps you pass in a fall -> the higher you score multiplier
* After x amount of gaps (3 gaps in helix jump) you get a special power that destroys the next layer and gives you immunity for that layer
* The pitch of the gap passing sound rises for each gap that you pass without bouncing
* Adds replay value: unlock new balls/skins, beating your record
* Removes replay value: Loads of unskippable ads or ads longer then the playtime you just had

I then divided these features into tasks:

|  |  |  |
| --- | --- | --- |
| Task | MoSCoW | Done |
| Create Platforms | Must | X |
| Create Player | Must | X |
| Implement player interaction with platforms | Must | X |
| Movement Controls | Must | X |
| Spawn Different Layers | Must | X |
| Create Coloured areas | Must | X |
| Create a scoring system | Must | X |
| Create UI | Must | X |
| Implement Game Over | Must | X |
| Implement Finish | Must | X |
| Create own twist | Must | X |
| Add Sound Effects | Should | X |
| Immunity Game Mechanic | Should |  |
| Add colour palettes | Could |  |
| Implement currency | Could |  |
| Add different kind of players | Could |  |

In the period of 5 work days I haven’t been able to complete all the tasks but I have been able to complete all the tasks assigned with a must.

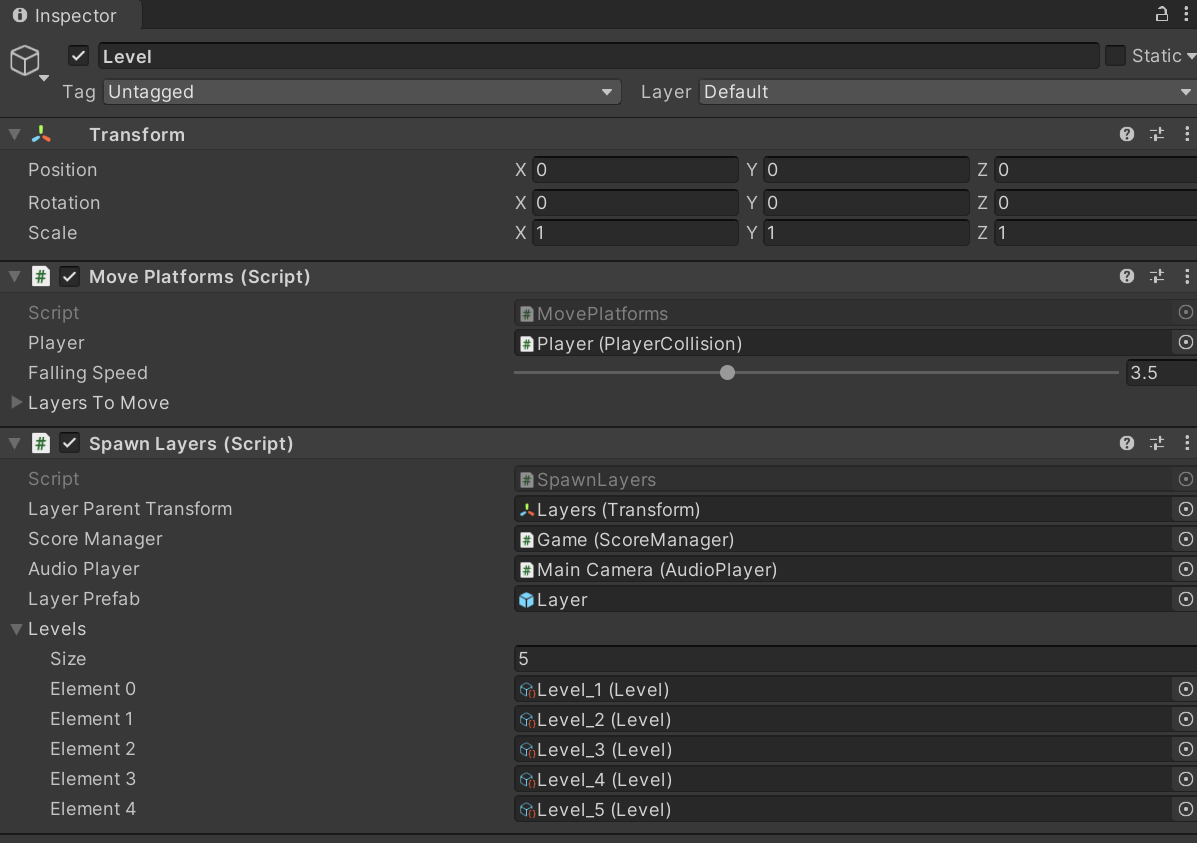
# Prototype Highlights:

Level Scriptable Object:

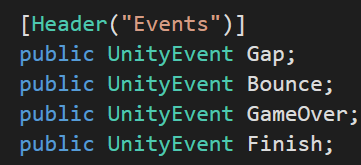
# 

In this Scriptable Object you can define how many total layers there will be, how many to always have loaded and what kind of platforms this level can have.

For each platform you can assign a maximum number. Do know that the first block will always be a good platform and that there is always 1 gap.



It is also easy to add another level. Just put it in the list.



The player collision has events. These events can be listened to through the editor as well as by script.

All methods have comments and a class has a defined structure:

1. Editor variables
2. Public variables
3. Private variables
4. Unity Methods
5. Public Methods
6. Private Methods
7. IEnumerators

# Day to Day Work:

## Day 1 Wednesday 09/12/2020

The first day I worked for 5 hours as I did not have more time.

I started the day by downloading the latest recommended version of unity: 2019.4.16f1 LTS

After opening the project I installed Probuilder and Progrids from the package manager as I knew I would need those for creating the platforms

I first created some prefabs with a ring that has a hole in it and I also added a coloured platform in a prefab.

I also created the player and made it bounce with animations as I thought that physics wouldn’t be as reliable all the time. This worked perfectly for the proto prototype I just created.

I also created the movement input system and implemented platform interaction events.

At the end of the day I had a small prototype of the game without score, winning conditions or game over conditions. I also had to manually create all the platforms.

## Day 2 Thursday 10/12/2020

I worked for 8 hours.

The second I started working on the spawning of the layers. Till this point did the layers not move and you had to create them manually.

To implement this I had to take a step backwards. I created a small segment that would fit 9 times in the ring. This segment is called a platform and you can change it to be a good/bad platform or a gap.

I placed these segments into a queue creating a pool of objects to use throughout the level.

I also created a scriptable object for a level in which you can define how many maximum segments can be in a ring.

As we are working with mobile phones we have to keep in mind to keep our game optimised.

Then I started moving the layers upwards and Enqueueing them as they went off screen. This however created a big problem with the moving of the player. The animation often went through a platform before it would recognise the collision.

I haven’t worked with physics or animations in a long time, so my skills were a bit rustic. Seeing that the animations didn’t work out the way I intended to, I decided to work on a physics solution on the next day.

## Day 3 Friday 11/12/2020

I worked for 8 hours.

I spend the whole day fiddling with the physics to recreate what I had on the first day but without animations. At the beginning of the day I made a “solution” which had a lot of rookie physics mistakes. As this did not work I consulted a friend of mine who is a student at Fontys. He then explained what my mistakes were and with that I could make an actual working solution.

I now worked with Unity Physics which means that my player would actually fall. For optimisation purposes I wanted my player to stay as close to the starting position as possible. This is why I made a HoldPlayer script which doesn’t let the player fall below a certain point. The layers containing the platforms start moving upwards when reaching this point creating a falling illusion.

## Day 3,5 Saturday 12/12/2020

I worked for 2 hours.

Because I spend the last two days working out a solution to the bouncing and falling of the player I made quite a mess in the scripts. This is why I spend two hours refactoring all my scripts. By doing this I also lowered the odds of running into problems later on for working in messy scripts.

## Day 4 Monday 14/12/2020

I worked for 8 hours.

I now had a working prototype but I didn’t have a twist of my own yet. I’m not very good with coming up with new ideas so I settled on creating a moving wall.

The wall itself wasn’t hard to make but the way I enqueued platforms caused an issue which I spend the rest of the day fixing.

At the end of the day I created my first APK Build so that I could handle any problems it would show on the last day.

## Day 5 Tuesday 15/12/2020

I worked for 3 hours (didn’t have more time)

Today I polished a few scripts, made the last build and created this document to show my work.