# Functional Requirements:

Jumping

To prepare, I had set up the animation controller for the player character, the background and foreground layers, and placed a basic platform to test jumping and ground detection. Next, I set up the on-screen controls (for ease of use and testing with both my PC and using Unity Remote 5), as well as the framework for player behaviours (i.e. Jumping, running, attacking obstacles, running into obstacles). From there I implemented jumping and ground detection, test playing after every notable addition to the code and using the debugger to track any issues.

Side-Scrolling

Created and tested a background moving script, then added a Running Behaviour script to the player. Tested to ensure it worked.

Ground To Run On + Platforms

Created column and centre prefabs, as well as a Platform script that constructs platforms with a given size. Tested and ensured it worked correctly. Fixed an issue with Platform spawn location and collider resizing. Created the Platform Manager with adjustable gap size, platform height and other options. Tested Platform Manager and adjusted Jump settings.

Interacting With Obstacles

Created Spawner to be placed on centre platform. Not working correctly due to time constraints.

Collecting items (Pick-Ups)

Created Spawner to be placed on centre platform. Not working correctly due to time constraints.

End State

End state triggers when player hits the water below. Option to reset level is given.

Obstacles Triggering End State

Obstacle that triggers end state created – not working due to time constraints

Scoring System

Obstacles and pickups either subtract or add to score.

Mobile Build

Build is targeted towards Android and tested using Unity Remote 5 (with a Galaxy S9 & S7). Builds to mobile correctly.