

## **Benjamin Pointer**

### **Week 1 (03<sup>rd</sup> August – 10<sup>th</sup> August)**

- Pieced together a prototype building system
- Created simple transparent meshes for building objects
- Begun scripting the PlayerController, which is set to handle all the building actions of the player – and some player shooting?
- Created a basic UIHandler class which is one class which has enum defined operations, so we can use one script for a varied number of UI Handles.

An issue I encountered this week was that although the player couldn't place the object in the air it would still show as 'green' or a possible placement, it will now display 'red' unless the raycast hits the ground...

### **Week 2 (10<sup>th</sup> August – 17<sup>th</sup> August)**

- Optimized the PlayerController script a heap, completely overhauled it
- Implemented a basic expense system
- Implemented Building UI – Pretty bad
- Optimizations anywhere and everywhere I could find
- Plus some bug fixes

I had a heap of weird bugs this week, some of which I didn't bother to fix. One bug is when the player is building, I am always updating the rotation of the placement object to that of the player's (basic direction) – on doing so if you flick through all the possible building options it will show you the building UI at placement position and then rotate to the players directional position – this creates a flickering view which is somewhat annoying but is easy to fix I just feel it's not that important.

Another issue I encountered whilst implementing the Building UI was the rotation of the 'hand' turret – which won't rotate to the player's position and so I had to have the UI at a fixed point which is extremely weird.

Overall, I originally started working on a UI script which would utilize one instance of the building canvas prefab opposed to utilizing it on each possible placement object (2 \* 4). It was extremely

## **Benjamin Pointer**

buggy, wouldn't rotate to the target's position yet when I tested it on a regular quad it would work perfect...