Team 2-bit Dev Log

Week 7: 30/10/23 - 03/11/23

Michael Hayes

Overview:

This week we completed our prototype and sent it to Liminal. We also completed our first small playtest and hope to organise the feedback next week. Our next step will be to wait for feedback from Liminal, and organise all feedback into sections of low / high priority and low / high time it will take to complete. We will also continue to optimise the game as that is one of the major areas that need to be improved for the next sprint. For the current Prototype sprint, there are a couple of menial tasks remaining that will be completed over the weekend.

Agile Sprint Update:

Sprint 4: Prototype (Will complete over the weekend)

Production of 2 game proposals with detailed information on the experience. To be submitted to Liminal VR for green light to begin development.

Links

Link to Minutes and Agenda: CS2 - Minutes and Agenda
Link to Team's Jira Scrum board:
https://cs2mr.atlassian.net/jira/software/projects/CS2LVR/boards/2/backlog
Miro board overview: https://miro.com/app/board/uXjVMj-Nye0=/?share_link_id=77318451600

Table of Contents:

Agile Sprint Update: **1**
Sprint 4: Prototype (Will complete over the weekend) 1
Links 1
This week's completed tasks: 3
Sprint 4 Tasks: 3
In progress tasks 3
Miro Board update 4
Resolving the raycast rotation from Laser gun issue 5
Making new targets 5
Destroy sounds 9
Internal Playtesting 10
Adding sight to laser guns 10
Adding cap to the amount of targets on the screen at one time 10
Challenges Encountered: 12
Installing apk files to the oculus quest 12
Other 13
Feedback from Lecturer 13
Team Members: **15**
Robin Pound - Co-Lead: 15
Next Week's Goals: **15**
Feedback and Comments: **16**

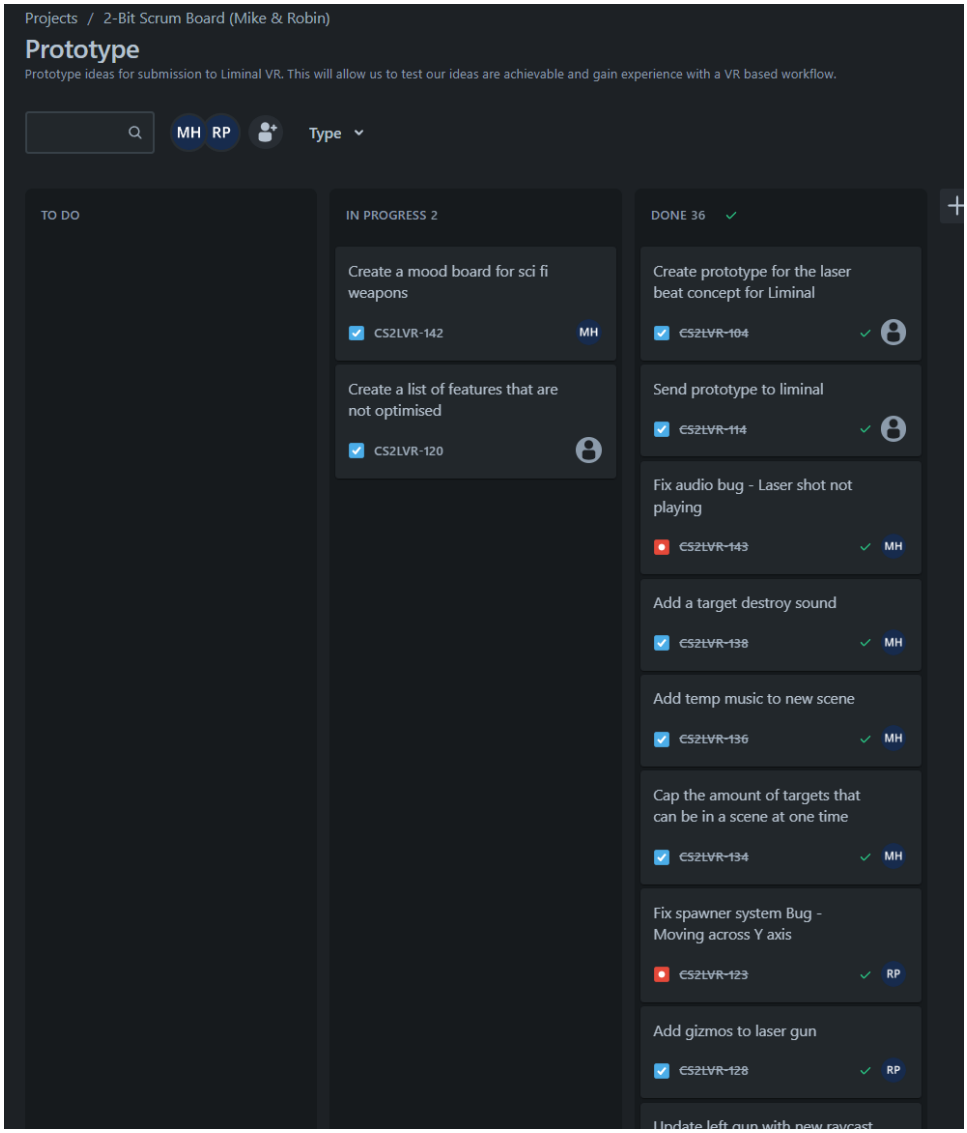
This week’s completed tasks:

Sprint 4 Tasks:

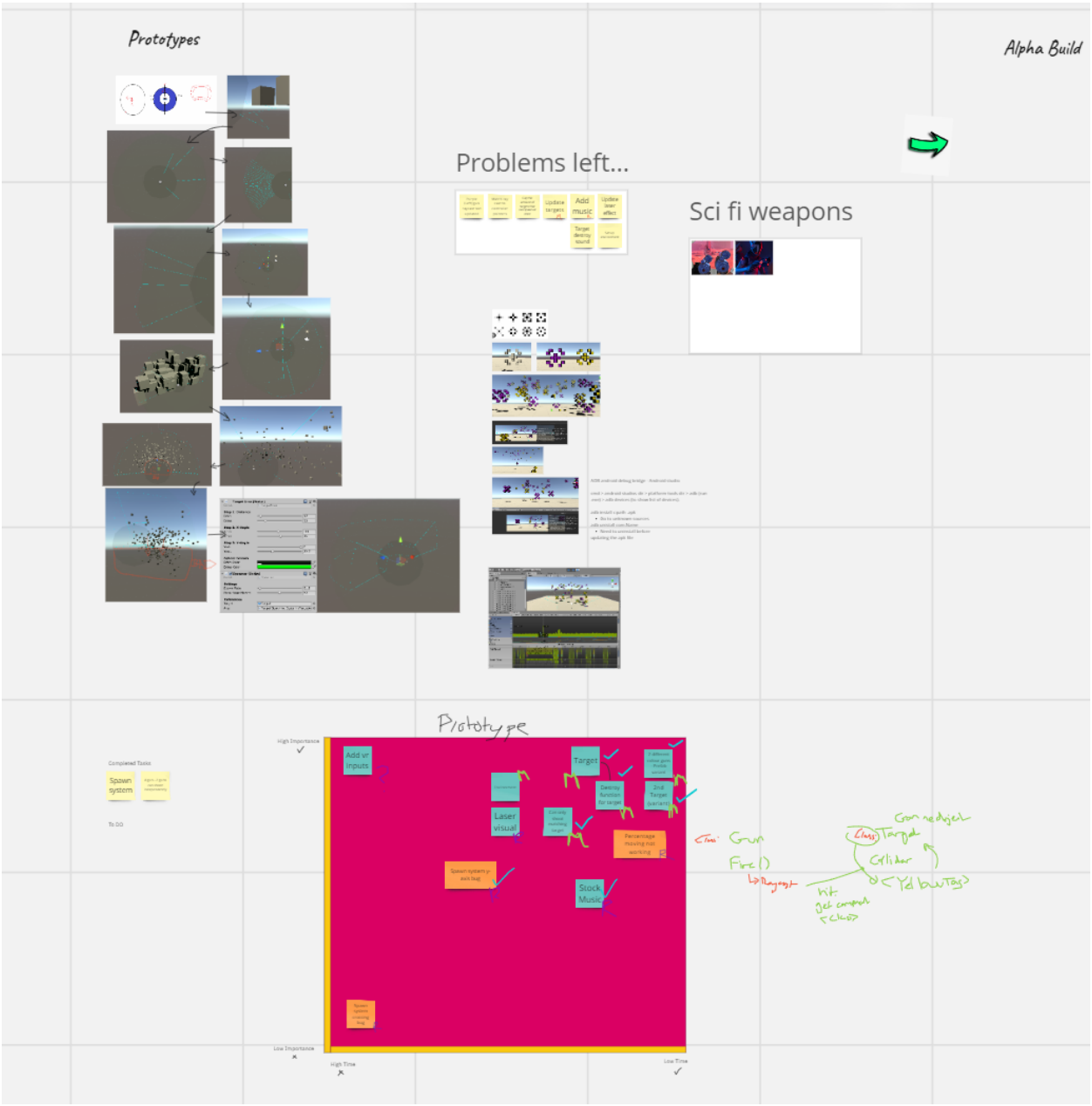
In progress tasks

The prototype sprint is now complete. The only remaining tasks in progress are small tasks of creating a mood board for the weapon design and creating a list of features that are not optimised which will be done over the weekend.

Jira Board Update (end of week 7)



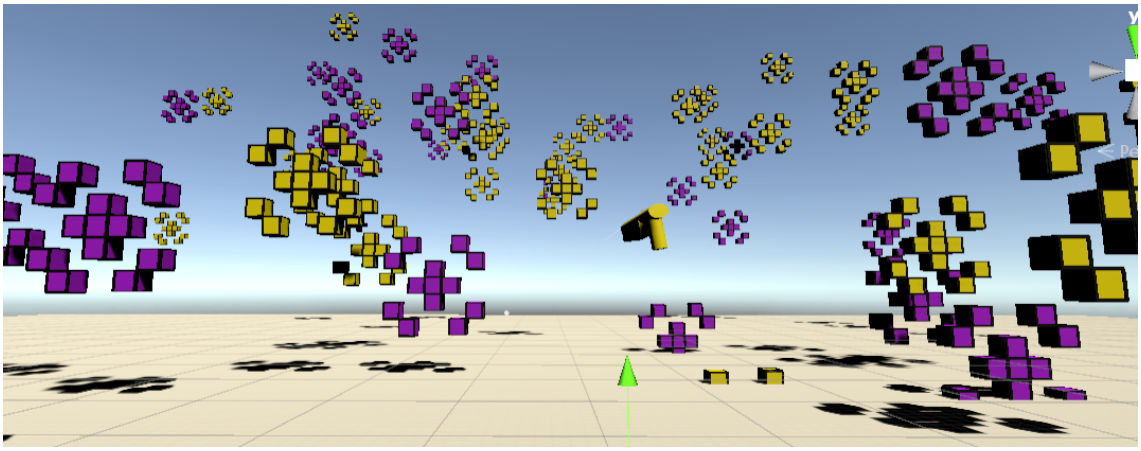
Team Miro Board update



Resolving the raycast rotation from Laser gun issue

This was a simple fix, to adjust the raycast of the gun to match the controllers direction we were initially using offsets to the rotation of the gun and adjusting them, but it was causing some undesirable behaviour when reaching a certain point, where our raycast would rotate more than the controllers ray cast. To fix the issue we just rotated the gun object to the position required and removed all rotation offsets.

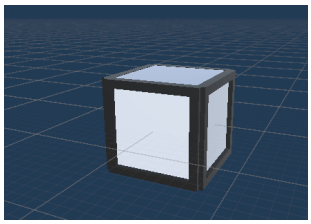
Making new targets



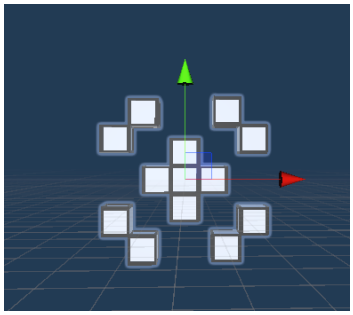
Reference image:

First, I made a prefab of a cube with borders to provide a cartoon feel to them. Making it a prefab made it easy to use them as a building block of another game object, as if I wanted to make changes I only needed to adjust the initial cube prefab, opposed to adjusting every component of the target prefab.

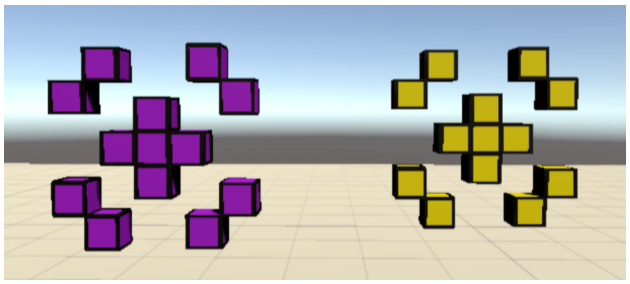
Initial cube prefab



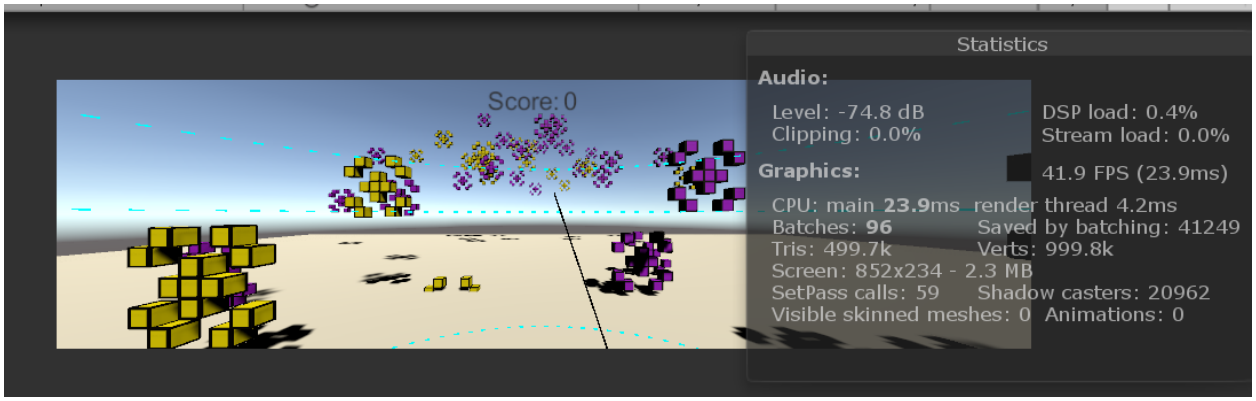
I then created a main target prefab using the initial cube prefab above, so I can edit the multiple variants at once by only changing the original target prefab.



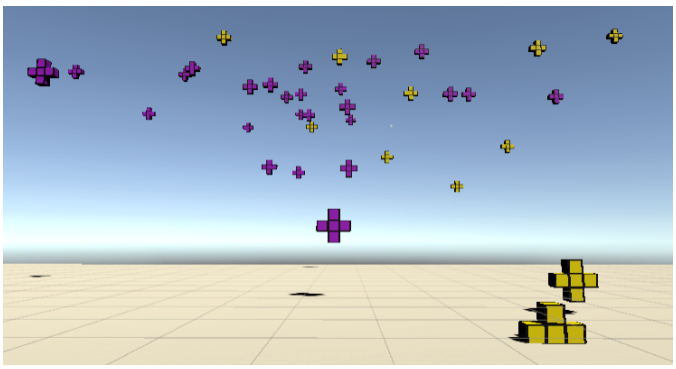
Then I created variants with their own properties (Tag script & Colours)



However, this design started to significantly impact performance as more and more spawn. The Fps dropped from 300 to 40 over a 30 second period.

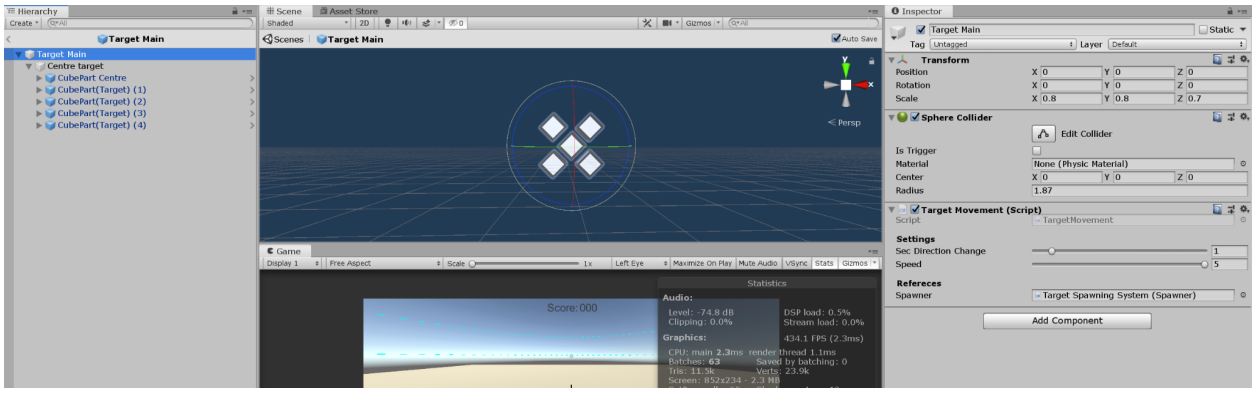


Reducing the amount of cubes saved performance, however it was still quite costly and they did not look very appealing

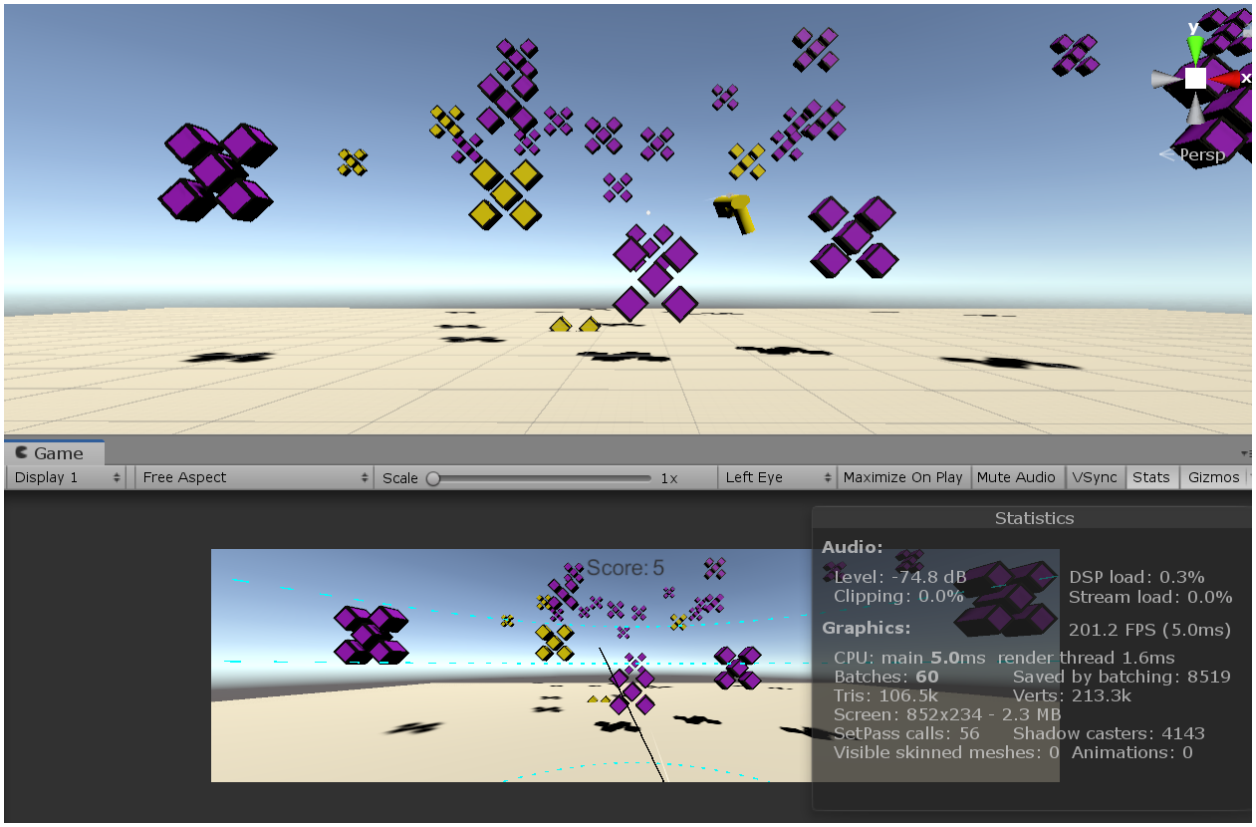


Having a main prefab made it easy to test new designs as I only had to update a single game object to affect all the variant ones in the scene.

I then changed the design of the targets to make it more visually interesting.



The targets are still quite resource intensive and therefore will need to be recreated in a different programme and imported into our project. For now they will work in the prototype.



Destroy sounds

As the main concept of the game is shooting targets, there needs to be an impactful and stimulating sound that is made when the targets are destroyed. This will hopefully provide enjoyment to the player and keep them engaged in the core game loop.

The audio is played at the source where the raycast last hit. Using a 3d sound for the audio clip makes it so the player feels the sound came from a specific direction. This will hopefully avoid any issues that arrive from non-normal sensory information.
The same was done for the laser guns, which both have a 3d audio clip component that is located on them. Therefore, when the player shoots their laser gun, they should feel as though the sound is coming from that specific weapon.

During the playtest I am using the standard Unity logarithmic curve that decreases over distance. I adjusted it slightly to match distances that make sense, however this will need to be further adjusted after more playtesting.

How the destroy clip is called at the location a target was destroyed

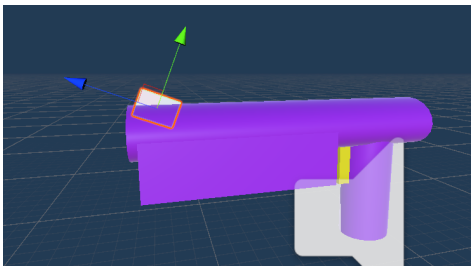
```
1 reference
private void CastRayRightGun()
{
    RaycastHit hit;
    Ray gunRay = new Ray(rightController.transform.position, rightController.transform.forward);
    Debug.DrawRay(rightController.transform.position, rightController.transform.forward, Color.red);

    if (Physics.Raycast(gunRay, out hit))
    {
        if (hit.collider != null &&
            hit.collider.gameObject.GetComponent<Target2Tag>())
        {
            lastRayHitRight = hit.point;
            //Debug.Log("Right gun hit " + hit.collider.gameObject.name);
            Destroy(hit.collider.gameObject, .1f);
            targetsHit ++;
            Instantiate(Marker, lastRayHitRight, Quaternion.identity); //spawn marker
            Instantiate(destroyEffectTarget2, lastRayHitRight, Quaternion.identity); //spawn particles
            AudioSource.PlayClipAtPoint(targetDestroy.clip, lastRayHitRight);
        }
    }
}
```

Internal Playtesting

After playtesting we found the guns were slightly too close to the player, and they needed a weapon sight to make it more obvious where the player should shoot. We also found that performance took a massive hit after around 10 targets were on the screen at one time.

Adding sight to laser guns

A cube game object was attached to the end of the gun's barrel to represent a sight

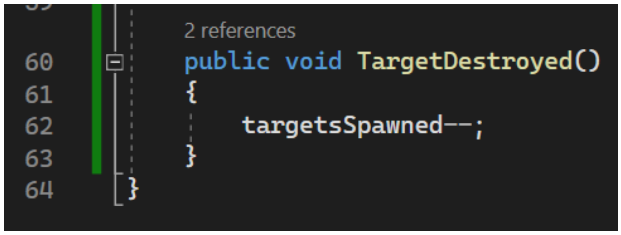


Adding cap to the amount of targets on the screen at one time

As the spawner script is originally Robins work, I made it obvious the code that I added and organised it as such.
To cap the amount of targets spawning I added an if statement (line 34) to check if the targets currently spawned is less than the set max targets. In this case the target would spawn as usual and the target's spawned integer would be increased by a value of 1. If the statement returns false then no code will be executed.

```
18
19      // Added by Mike
20      [Tooltip("Set the max targets that can appear on the screen at one time.")]
21      [Range(1,10)] [SerializeField] private int maxTargets;
22      [Header("Debug - Don't edit")]
23      [SerializeField] private int targetsSpawned;
24
        Unity Message | 0 references
25      void Start()
26      {
27          StartCoroutine(StartSpawning());
28      }
        1 reference
29      private IEnumerator StartSpawning()
30      {
31          while (true)
32          {
33              yield return new WaitForSeconds(SpawnRate);
34              if (targetsSpawned < maxTargets)
35              {
36                  spawnRandomTarget();
37                  targetsSpawned++;
38              }
39          }
40      }
```

To keep targets spawning within the game, I added a public field that could be called when a target is destroyed, to also reduce the target's spawned count, allowing targets to continue to spawn in.



Challenges Encountered:

Installing apk files to the oculus quest

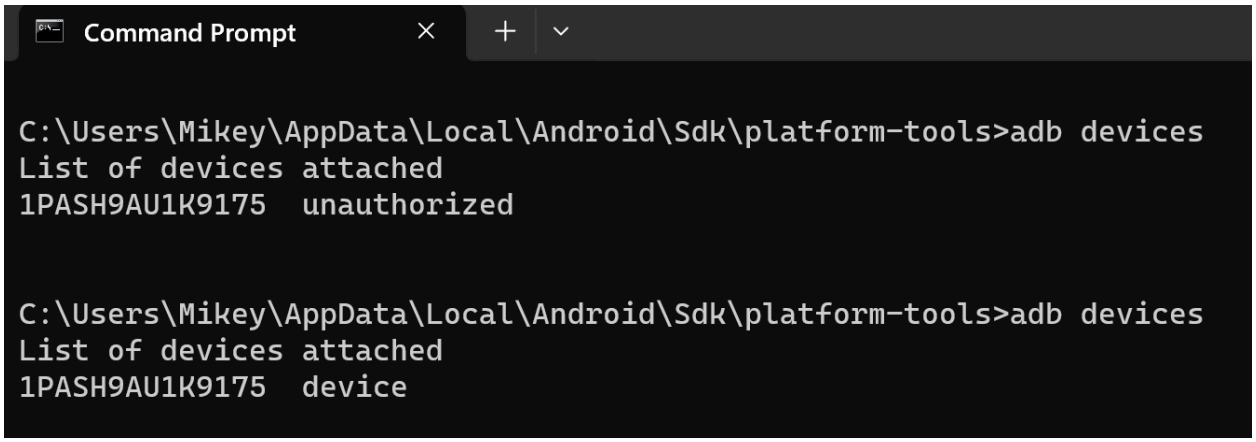
When trying to install files onto the Oculus quest, I was not able to as my pc did recognise the quest as a device.

To fix this with the help of our lecturer, I downloaded the Android studio suite, which contains tools to debug problems with connections, and side load apk files onto the quest.

To use the tool we need to first open the command prompt and find the path to the Android studio tools directory.

Through the adb executable, we use the command devices to find a list of connected devices When found a prompt on the Oculus appears allowing us to authorise the device.

Test connecting school quest with Android debugging tool

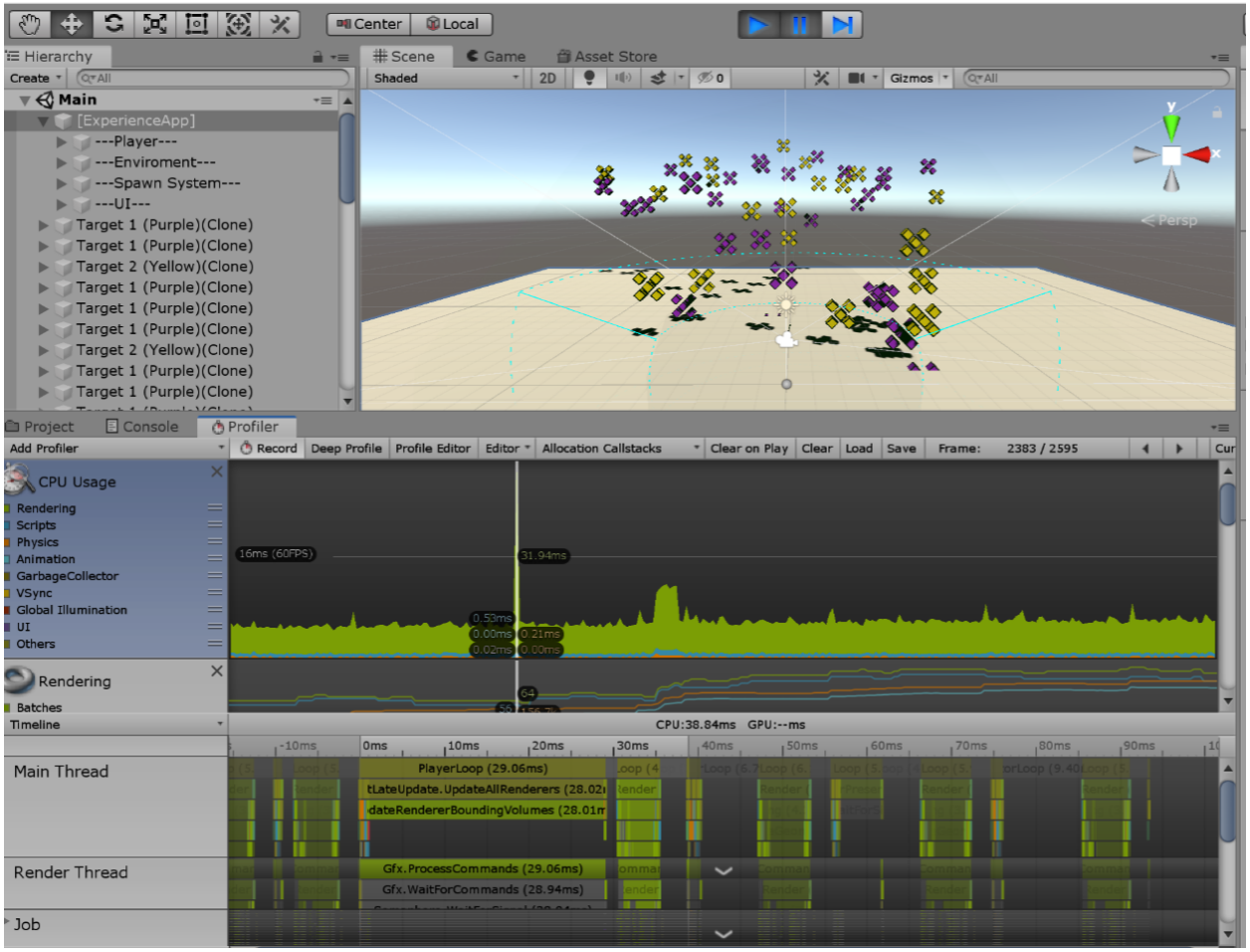


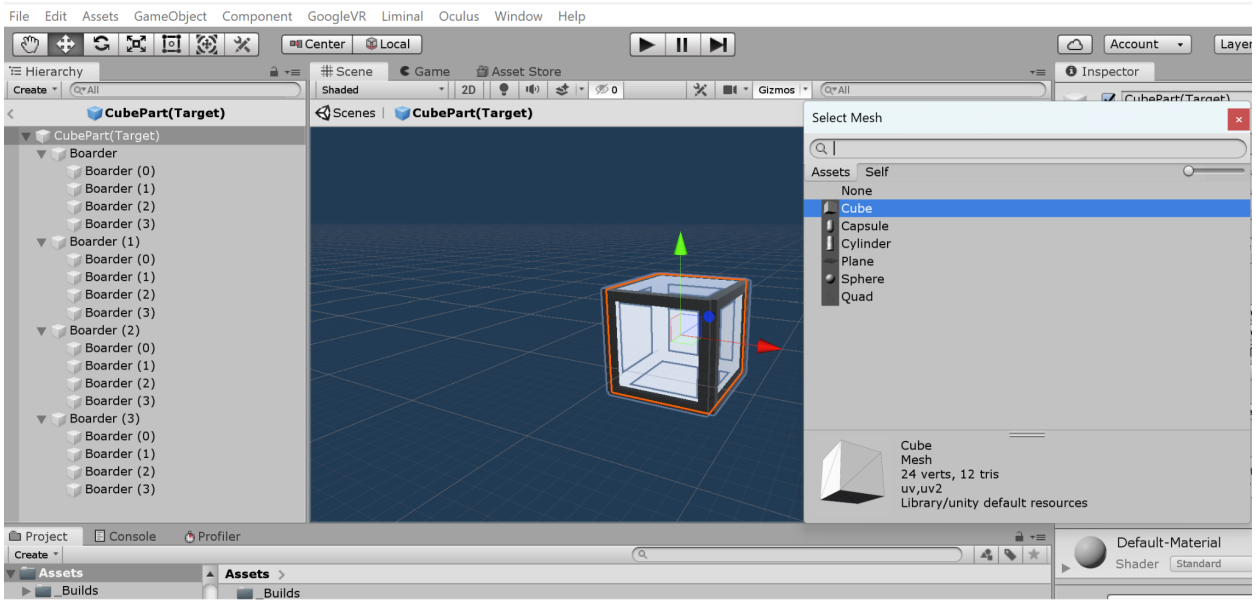
Unfortunately, after several attempts and hours of troubleshooting, I was unable to connect my device to the iphone app or the meta developer app to successfully turn on developer mode on the device. The device is found by the phone however will not connect to enter the headsets settings. Therefore, I am unable to connect my Oculus device to the Android developer mode at this time and will try again next week. This a serious blocker as it prevents me from testing the game and yet another large blocker that is slowing down the timeline for the project.

Other

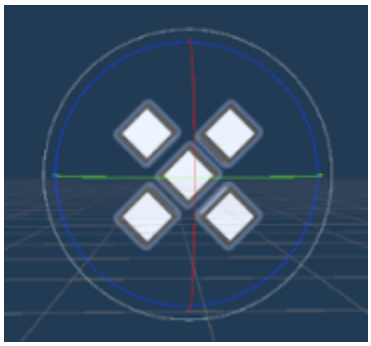
Feedback from Lecturer

As more and more targets spawn, the game's performance starts to drop. Using the profiler unity tool I found that a lot of resources were being spent on rendering.





Each target is made of cubes with 4 borders around it created with other cubes. Each of these cubes are 24 vertices and 12 triangles. That equates to 408 vertices and 204 triangles. When there are multiple of these blocks being used to create targets this multiplies further.



To solve this, we will create the object in blender as a single mesh to reduce the amount of vertices and triangles in the scene.

Team Members:

Robin Pound - Co-Lead:

Individual works:

- Playtest
- TDD (it will be finished this week, promise - lol)
- Gun ray - Quaternion rotations and gizmos
- Added marker, for debugging
- Updated the particle system

Next Week's Goals:

- Complete the outstanding tasks for sprint 4 prototype over the weekend
- Start Sprint 5: Beta Build
    - Create a feature complete beta build that has all the core components present making the game, excluding some final touches and polish

Feedback and Comments: