

Introduction

This is a project that I have done with Udacity that provides me the opportunity to create a Virtual Reality Rube Goldberg Game using the software, Unity.

Outcome

The game has been finalised after several rounds of iterations using feedback from different users.

The Rube Goldberg is to bring the ball into the goal position using different contraptions without it touching the ground. The user is given a menu to create different objects that affect the gameplay through the use of Unity's in-game physics. There are a total of 4 different levels with each level requiring different objects to be spawned. At the end of the game, the total time duration of the game will be displayed.

Story of the process

In the Rube Goldberg Project, it is important to identify the contraptions that will be used in the development of the game. Several sketches were made to identify the uniqueness of each object.

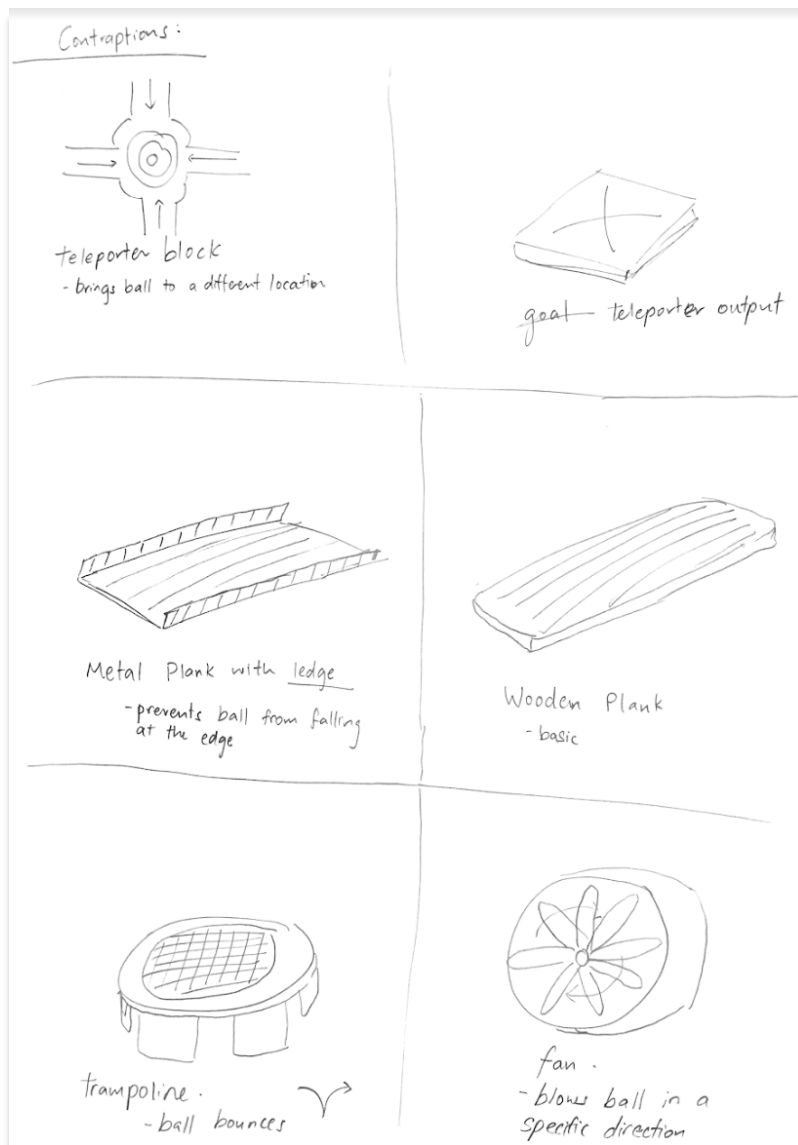


Figure 1. Sketch of Contraptions

With the objects set, a couple of stages were created where the User had to create an object and use it to get the ball across the finish line.

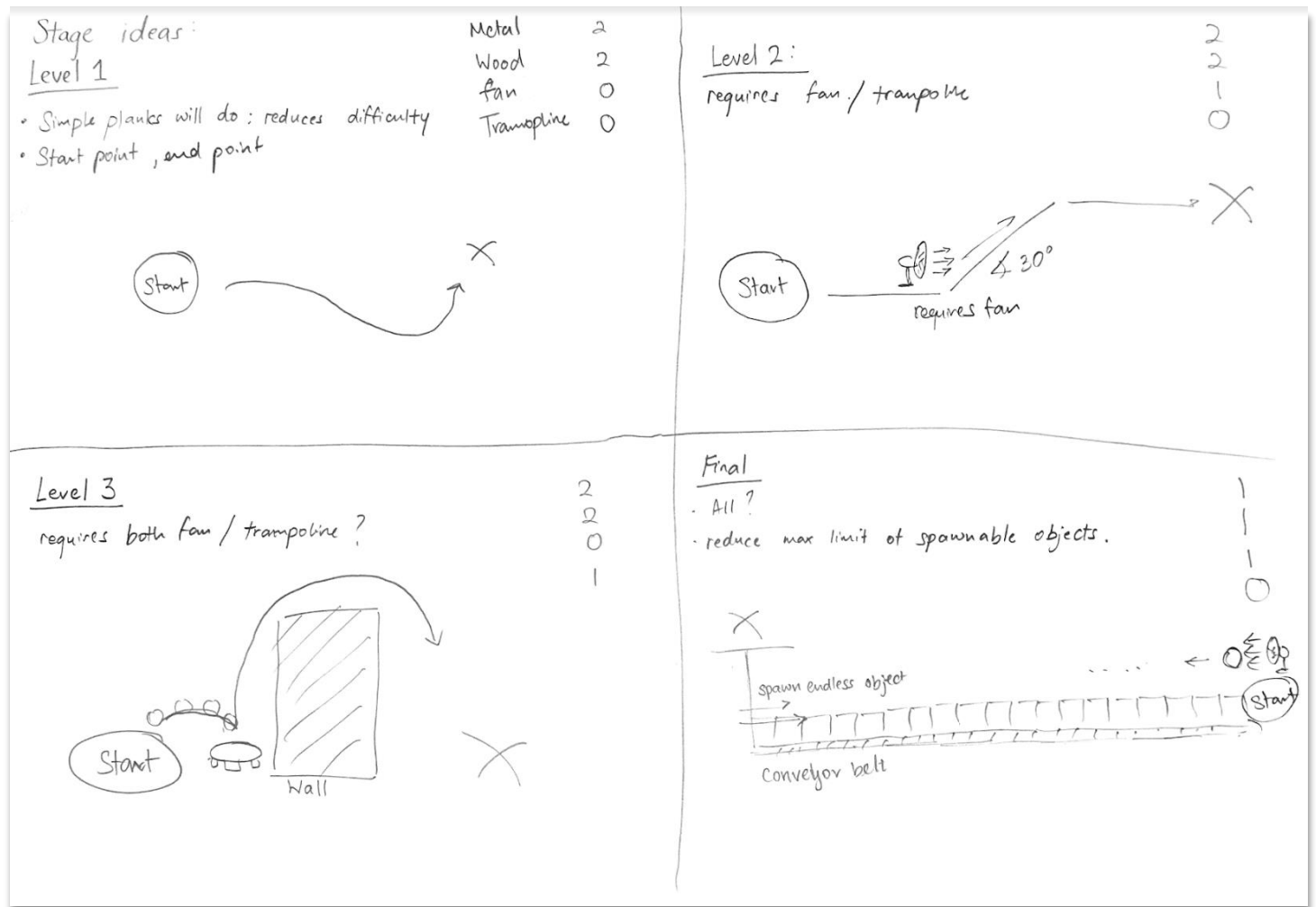


Figure 2. Level Designs

The difficulty of every stage can be varied by limiting the number of objects that can be created.

The implementation of the contraptions was prioritised over the level as it would be easier to develop the stage when all the elements of the game are ready. These included scripting for acting forces on the ball when interacting with certain objects.

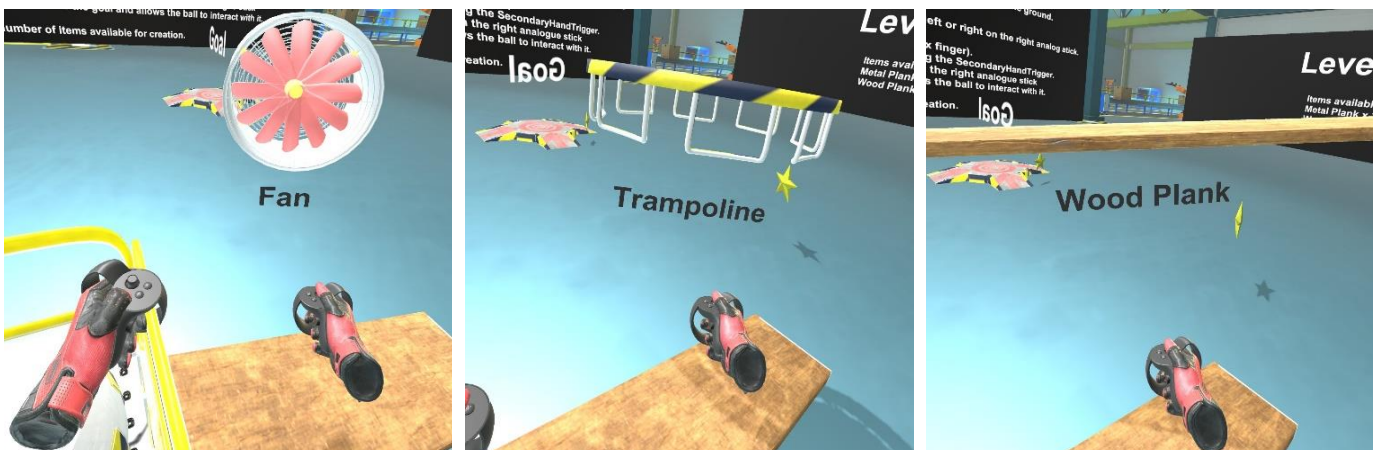
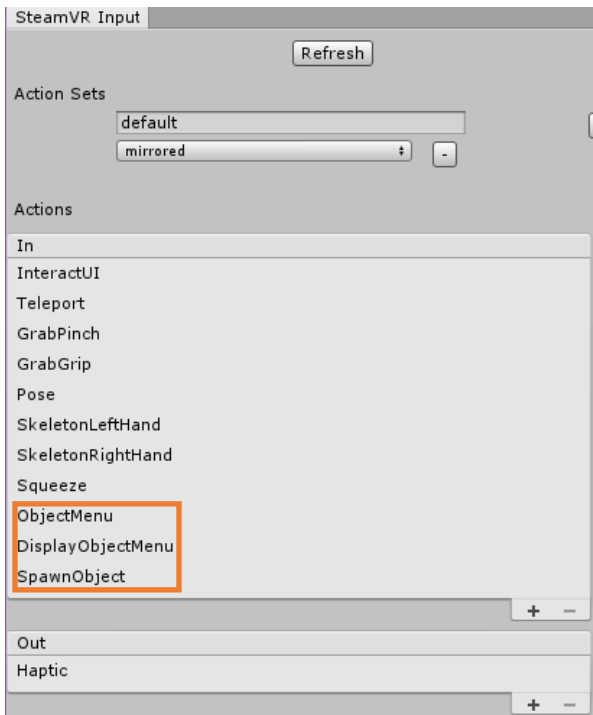


Figure 3. Screenshots of Object Menu

Build Process:

The SteamVR that was automatically launched from Unity does not work as intended. This resulted in a blank visual on my Head Mounted Display (HMD). Therefore, launching the project from Unity into my Oculus Rift required some tweaking (Close the SteamVR running in the background and manually launch the SteamVR before starting the game in Unity).

Setting up the key bindings for the Oculus Rift using SteamVR:



Additional actions included in this Project

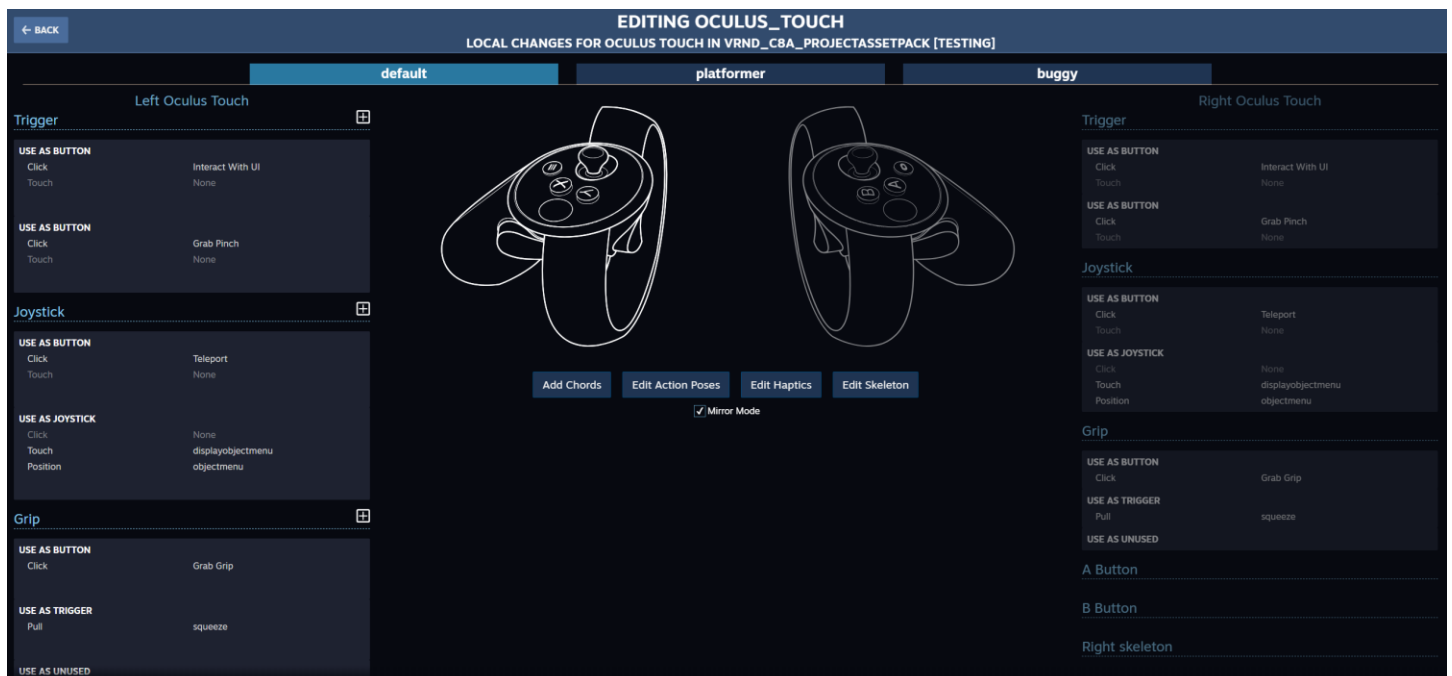


Figure 4. Key Bindings

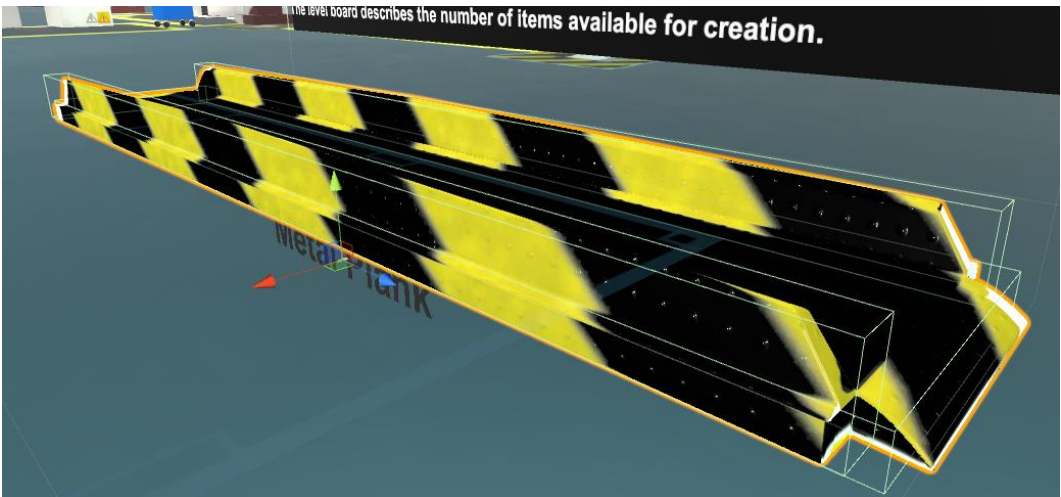


Figure 5. Multiple Colliders in Metal Plank

Accessing the HMD often allowed me to stabilise my code and provided many useful feedback such as adjusting the colliders of my objects and achieving an optimal position of my level board.



Figure 6. Level board indicating the number of items available for spawning

While testing I notice that the user was able to grab and throw the ball into a desired location. This breaks the game and makes the challenge of the ball not touching the ground pointless. A simple fix was to prevent the ball from being thrown by removing the velocity while the ball is being let go.

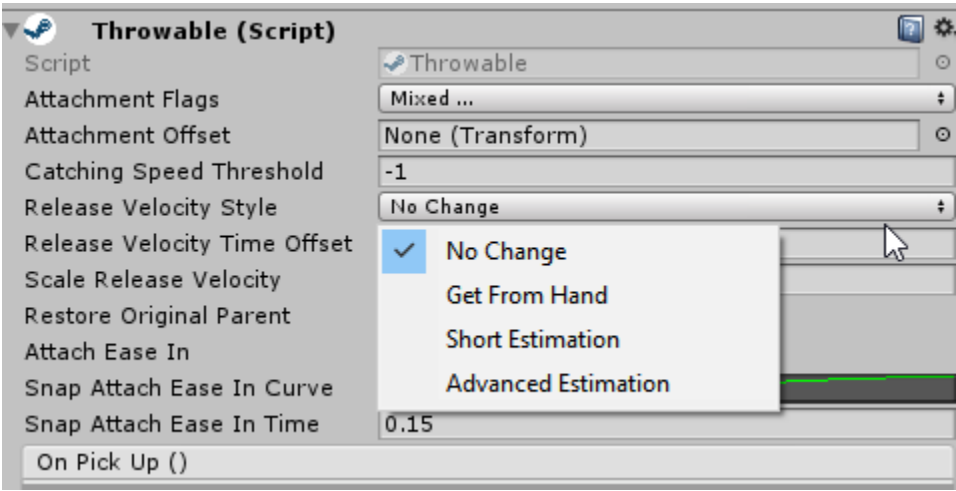


Figure 7. Selecting a different Velocity Style

User testing outcomes and iteration

Self-iterations of the game included creating a pedestal that makes it more convenient for the user to pick up the ball.

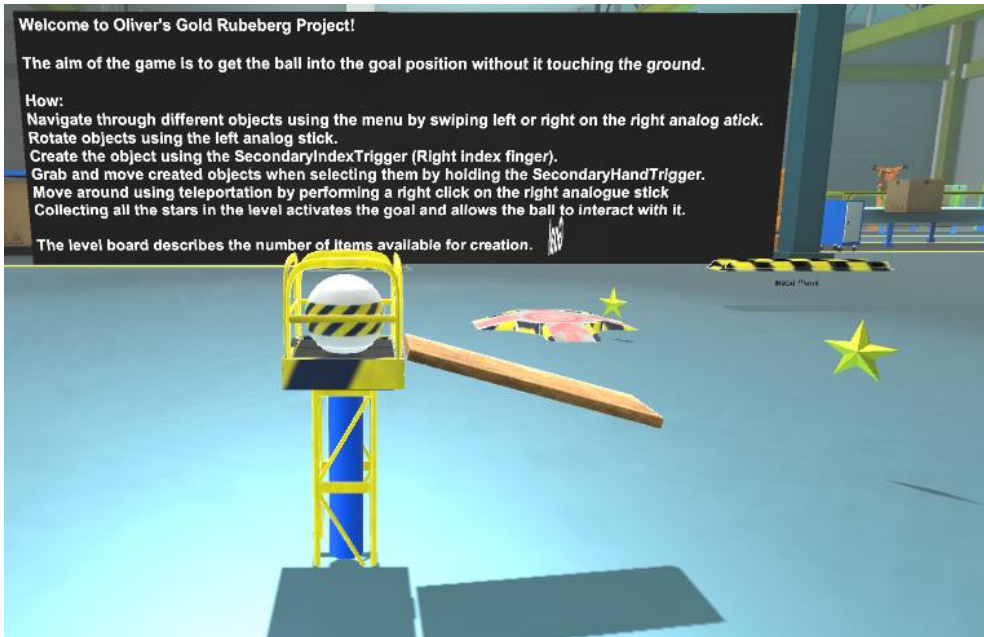


Figure 8. Screenshot of Pedestal

User Feedback #1	Iteration #1
It is difficult to see what items were available for spawning in the stage as the level board was placed in behind the camera.	Re-position Level Board from the Curiosity Zone to a more prominent area.

User Feedback #2	Iteration #2
Swiping the touch controllers left and right scrolls the object menu too many times.	<pre>if (!hasSwipedRight) { if (touchCurrent.x > 0.5) { SwipeRight (); hasSwipedRight = true; hasSwipedLeft = false; } } if (touchCurrent.x == 0) { hasSwipedLeft = false; hasSwipedRight = false; }</pre> <p>Figure 9. Introduced Booleans (<i>hasSwipedLeft</i> and <i>hasSwipedRight</i>) to combat <i>over-scrolling</i></p> <p>The checks above will only allow the User to scroll once and requires him or her to release the analogue stick back to its original position before making subsequent scroll.</p>

User Feedback #3	Iteration #3
<p>Spawned objects that were initially positioned on the ground required the user to bend down and move them. This creates a great deal of inconvenience.</p>	<pre>//Remove oldest object when object hits maximum spawn count if (currentCount[currentObject] >= maxCount[currentObject]) { Destroy (GameObject.Find (string.Concat (objectPrefabList [currentObject].name, "(Clone)"))); //Do not go below zero if (currentCount[currentObject] != 0) { currentCount [currentObject]--; } failAudio.Play (0); }</pre> <p>Adjusted code for spawning objects; By removing the oldest spawned object when spawning a new one will provide the User more convenience</p> <p>The User does not have to move the objects manually by grabbing them. But one can use the touch controllers to spawn a duplicated object in a new position while removing the old object. The conventional method, grabbing, is still available for those who prefer grabbing and moving an existing object.</p>

Breakdown of final piece

With the use of Steam Level Loader, There is a seamless transition between stages. The fade in and fade out effect provides a very professional experience.

An anti-cheat detection system was also created to determine if the user has used the contraptions to win the game. Several mechanisms were used to detect if the ball was grabbed after being released from the starting point. Thus, preventing the User from cheating mid-way.

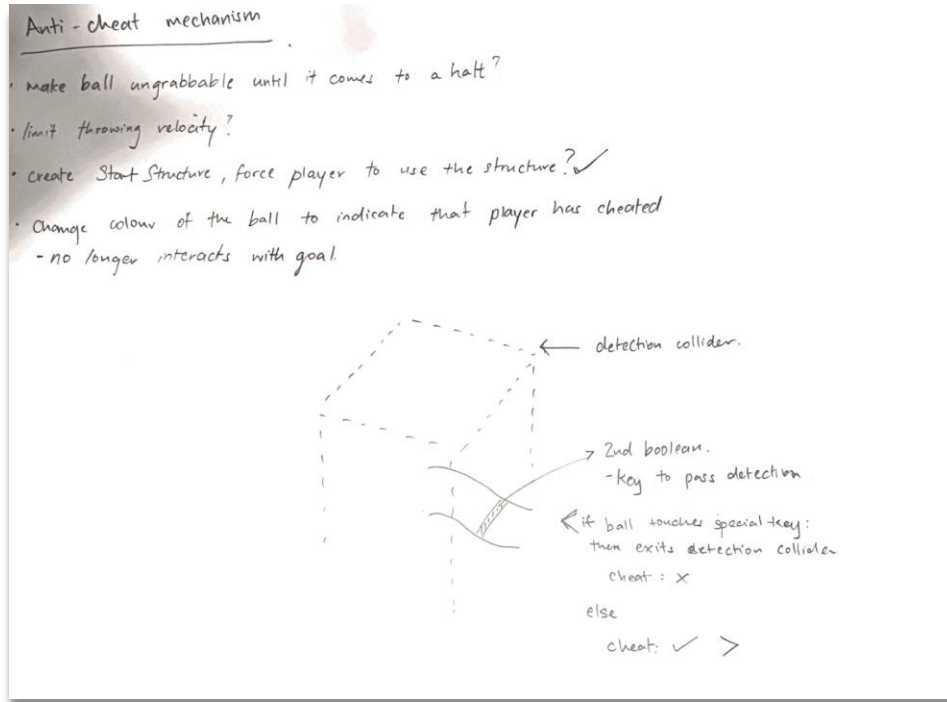


Figure 10. Sketch of Anti Cheat Mechanism

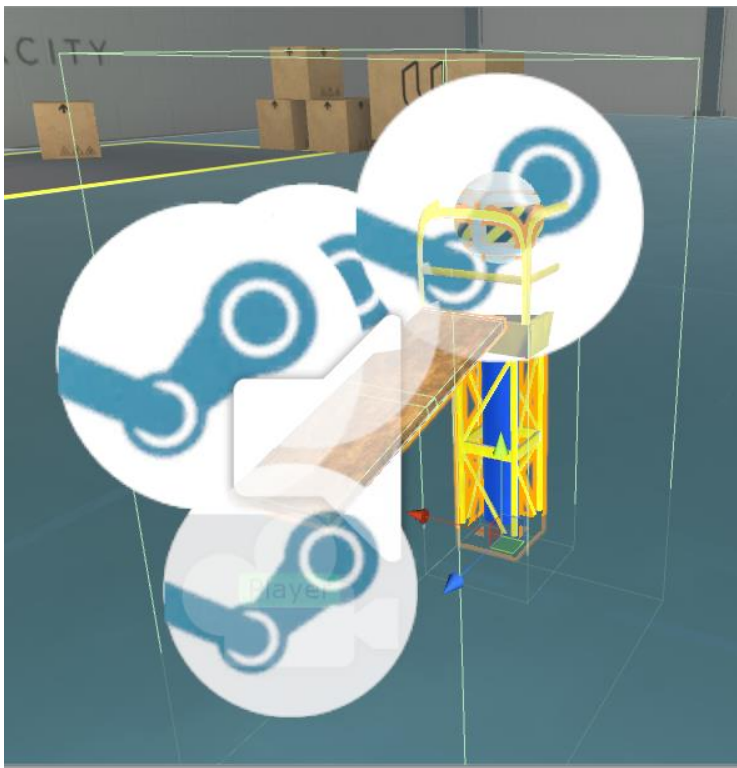


Figure 11. In-game Colliders that detect TriggerExits for Anti-Cheat Detection

Upon detection, the ball will be coloured in red to display that an abnormality has been detected. And it will no longer be able to interact with goal using Unity's Layering System.

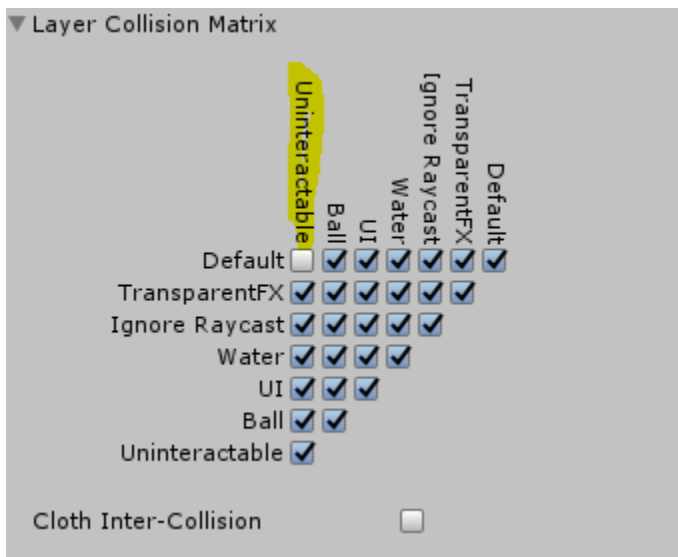


Figure 12. Setting up layer-based collision detection

With the completion of all 4 stages, the User will be presented with a winning screen that displays the time taken from start to completion of the final stage. The song 'We are the Champions' will also be played while transporting the User to a position, overlooking the entire environment.



Figure 13. Winning Scene

Conclusion

I found the Rube Goldberg project to be a good demonstration of how Physics work in a VR environment. It provides an immersive experience for those who would like explore VR. The project was developed using SteamVR, which provides access to users who own either the Oculus Rift or the HTC Vive. This project also allows the User to be creative by solving challenges in his or her own way.

From the developer's perspective, this project was a milestone for me. It provided me a huge opportunity to work on my design and technical skills. Creating an object menu to spawn objects, a cheat detection system, allowed me to advance my skills as a developer. This project also further enhances my knowledge on Unity's Physics system.

Lastly Udacity's Rube Goldberg project gives me an opportunity to present my work to my future employer.

Video of my project:

<https://youtu.be/B3b08U2x34w>

