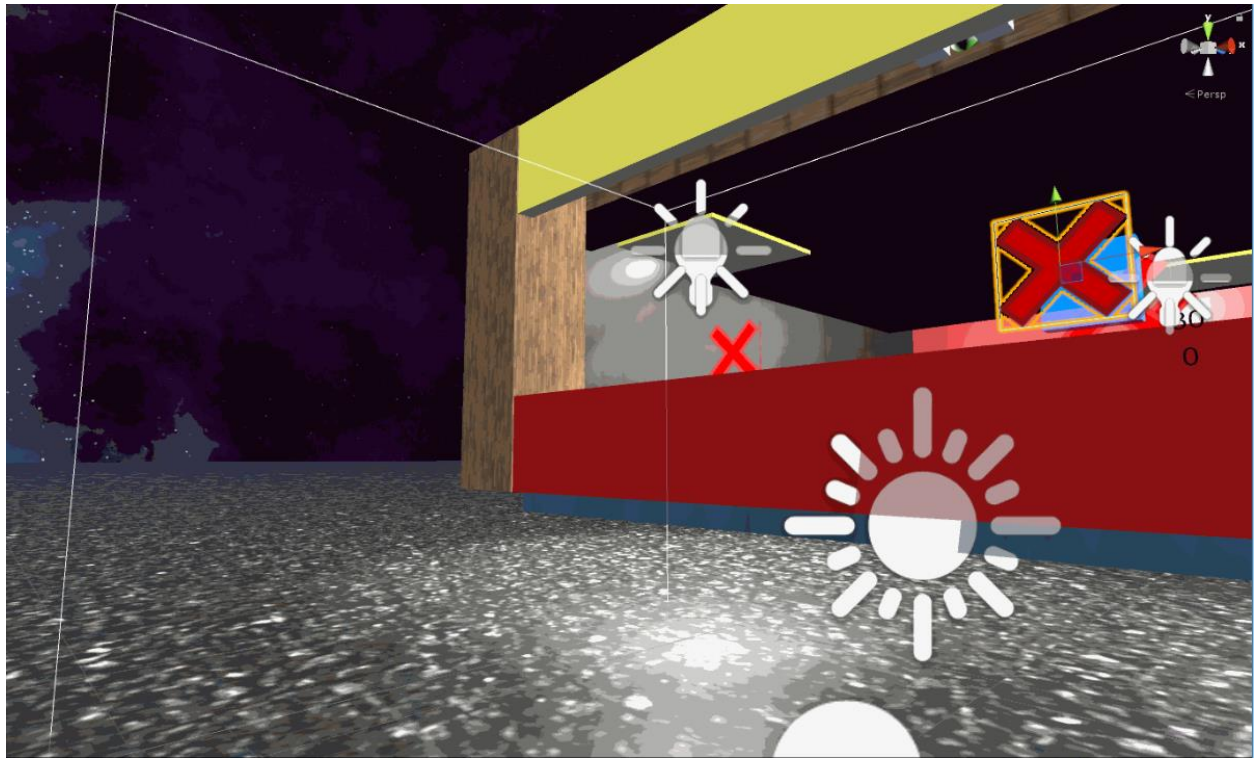


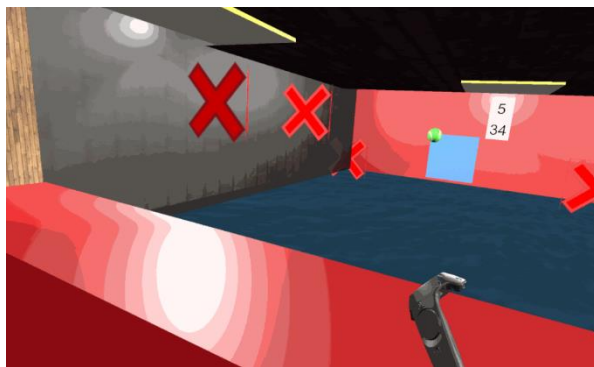
THROWGAME-VR KEVIN HOANG



***Requires: Unity
HTC Vive Headset***

Overview

The general overview of the project made for this course is a small adaptation to a game that you would typically find in carnivals, made for the HTC Vive Headset, with a twist. The idea here is that the player is trying to stay in the game as long as they could before the timer runs out. The player will receive a ball, and is trying to throw it towards the intended target in front of the player. If the player manages to throw a ball to the target, the remaining time left will be increased, allowing the player to stay in the game longer. However, the player has some obstacles in front of them, particularly floating walls that levitate and move in one direction. Three rows of walls are presented, with each row increasing in difficulty the further it goes. If the player is able to throw the ball without hitting the obstacles and hitting the intended target almost every time, the player will be able to stay in the game indefinitely.

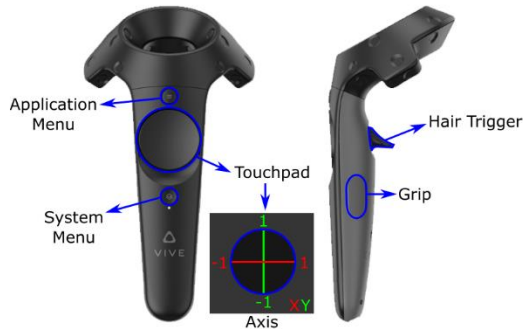


An addition to the game that was planned, but not implemented to the final project was having an NPC as a “friend” standing beside the player, watching the game, and react to how the player was doing. The player could interact with the NPC using a variety of gestures. Meaningful gestures could increase the remaining time left for the player. These gestures would have included, but not limited to, waving, pointing, shrugs, and nods. Unfortunately, the idea of gestures is difficult to implement. A lot of variables need to be accounted for, such as the orientation of the user and the requirements for a gesture to be recognized. With too much to think about, the idea of gestures was scrapped for the project.

The file attached includes the entire unity project required to run, edit, and test the game. Attached to the file also includes an executable to load the game, if opening in Unity doesn't work, and a GIF of the demo to demonstrate a run of the game. A VR headset (HTC Vive Headset specifically, other headsets have not been tested) is required to test (or even see) the game with the executable file.

Rules

You have 30 seconds until the game is over. So make sure you use your 30 seconds to say your prayers. Or if you want to stick around longer, grab the ball in front of you and throw it into the blue targetter that is moving randomly around in the area. The more often you throw it in, the longer you are able to stick around. However, nothing said that it was easy to accomplish this task. You also have moving obstacles to throw you off or block you from getting the ball into the targetter. They move fast. They move slow. They move up. They move down. Everything is fair game. Remember, you only have 30 seconds, so make it count. But how long could you really last?



Controls

Wand: The controller

Trigger Button: Used to grab the ball
(Hover inside of the ball and hit
the trigger button to grab it)

When the game ends, you must reload the
executable if you want to retry.

Scripts

Controller Grab Object: This script was made to include the functionality of grabbing and throwing objects with the HTC Vive Headset. The general idea of the script is that if the controller is inside of a throwable object, and hits trigger, then the object is attached to the controller, and will only be released if the player lets go of the trigger. Speed and rotation to the object is added depending on how the player lets go. While this script contains the core functionality of letting the user interact with the environment, it is borrowed from a site when I first started learning VR in Unity. Credits are placed at the end.

Gestures: This script was made to include the use of gestures in the game along with the ability of interacting with the environment. The idea initially was that the gesture could be used to interact with an NPC beside the player whenever they would throw. The NPC would make some sort of gesture, and the player could respond back with gestures, such as a high five, or a cut-throat gesture, either giving or losing time for the game. Some of the gestures were implemented with a good amount of bugs in them. Some of them were able to respond, but most of them were received poorly. As the entire idea of interacting with an NPC was scrapped, the script has no use and therefore not implemented to the project. The script can be tested in Unity when looking at the controller through the inspector window. This script was made from scratch.

Camera Action Response: This script was made to complement the use of gestures for interaction. Like gestures, it was intended to be used when interacting with an NPC, such as nods, to respond back to them. The script, while similar has similar problems to gestures, has less bugs than anticipated, as nodding and shaking the head seems to respond well majority of the time. As the entire idea of interacting with an NPC was scrapped, however, the script has no use and therefore not implemented to the project. The script can be tested in Unity when looking at the Camera (Eye) through the inspector window. This script was made from scratch.

Move Left / Right / Up / Down: These scripts were made to create the behaviour of the obstacles as well as the target. Obstacles inherit one or two of these scripts while the target inherits all of them to add to the complexity of the movement. These scripts were made from scratch.

Target Movement / Hit: These scripts were made to control the functionality of the target, specifically its movement and when it gets hit. The idea was to make the movement of the target behave differently to its obstacles, thus adding to the complexity of the game. While adding a different moving pattern would be unique, having the movement pattern be different every pass through (from left to right or right to left) would keep the player active and make it not predictable on where it goes, making the game harder. The hit script simply takes action when the ball manages to hit the target. In this case, it would increase the remaining timer left by five, and change the target's color to indicate that it was hit. These scripts were made from scratch.

Floor Teleport: A simple script made to make the ball teleport onto the position in front of the player if the ball were to be "out of bounds." Out of bounds in this case means if it touches the floor. This script was made from scratch.

Timer / Count Up: A simple script made to simulate a timer, both counting up and counting down. The use of counting up was to see how long the user has stayed in the game. The use of counting down was a time limit to let the player know the game is over if this timer reaches zero. This script was made from scratch.

Running / Editing the Game

As stated, the file included comes with the entire unity project of the game, an executable, and a GIF to demonstrate the game. To open the project in unity, navigate to the folder where the scene is contained.

Throw Game VR -> Assets -> Scenes -> ThrowGame

Pop ups may appear when opening it with Unity. Press the buttons that allow the program to continue loading the project.

Playing the game: The top of the screen presents three buttons that represent running the game. Press the left most button, that represents the play button. The HCI lab in room E2 – 533 has a Vive Headset that could possibly be borrowed if needed.

Changing height: It is possible that the height of where the player stands might need to be adjusted. If so, change the position of the Y for the object labeled [CameraRig].

Environment

In terms of objects laid out on the environment, majority of it was made from scratch. From the objects made, to the basic textures that was use were made with Unity's built in materials and objects. The only things that were not made from scratch were any textures that were not simple to make (i.e. any material that wasn't a basic color texture), the skybox and the entirety of the VR implementation with SteamVR.

REFERENCES

Images: This set of references was used for textures that were implemented into the project.

Baseball:

<http://www.robinwood.com/Catalog/FreeStuff/Textures/TextureDownloads/Balls/TennisBallColorMap.jpg>

Skybox:

<https://www.assetstore.unity3d.com/en/#!/content/25142>

Wood:

<https://i.pinimg.com/736x/4d/bb/f9/4dbbf9e7afb8f20859dd4b7af89e6fae--wood-material-texture-texture-floor.jpg>

Gravel:

https://s3.envato.com/files/103652550/Gravel_Road_Textures_2048x2048_7.jpg

Virtual Reality: This set of references was used for the implementation of VR in the project.

SteamVR:

<https://www.assetstore.unity3d.com/en/#!/content/32647>

VR Interaction:

<https://www.raywenderlich.com/149239/htc-vive-tutorial-unity>