Design Notes for VR_Ring

By Christopher Burst

I'm including this document to help whoever this project is passed on to understand my thought process in how I set things up to work.

- The basics ○ The experiment is as follows: An object in the scene lights up (changes its material to a self-illuminated yellow color) and plays a 3D sound. The participant should hear the sound through headphones, and look in the direction of the object with the Oculus Rift. They will then press the spacebar to confirm that they see the object, this object dims (goes back to a non-illuminated gray material) and another object lights up and plays a sound.
    - Note 1: The objects only illuminate themselves. They do not cast light. The light and shadows would give the participant more information on where the object is they are searching for than is desired.
    - Note 2: I pulled the beep sound effect from the video game Portal. Not sure how copyright works in software that is not intended for sale, but you might want to consider replacing it.
- Input/Output ○ Currently this is designed to take input from a CSV file. (input.csv) That file will have a row of numbers representing the order of which objects to light up.
    - ○ "output.csv" is exactly what it sounds like, and whenever an object lights up or the participant finds an object the program records the time this action occurred since the test started, time since the object lit up (will read zero if the action was the object lighting up), and the action that occurred (lighting up or participant finding it).
- Is the participant looking at the right object?
    - ○ In order to confirm that the participant is looking at the right object when they push the spacebar, there is an invisible cube hovering in front of the Oculus camera. So long as the object is within the cube, then the program knows that the participant is looking at the object.
        - There may be a more efficient way to detect if something is within sight of the camera. If there is, I don't know how to do it.
        - Note that the cube won't detect objects that are farther from or closer to the camera than the cube's width. If you need to place objects closer or farther away, the cube will need to be moved and scaled accordingly. Multiple cubes may be needed if the range varies greatly enough between objects.
        - The cube slowly rotates. This is because I've found Unity is better able to detect when an object is within another object if one of them is moving in some way. While the cube is technically moving when the oculus camera moves, its local

position as a child of the camera does not. So a rotation is added. How a complex array of objects is automatically generated

- To create a sphere of objects I've added an invisible polygon to the world and wrote a code to place the objects automatically at every vertex of that shape. When placed, the objects make sure that they are not redundantly occupying the same space as another object and give themselves unique names for other code to later find them with.
- This should be flexible enough to work with any 3D shape, but in this case a simple soccer ball/dodecahedron shape is used