Travel and Manipulation (10 points)

Submission due by Tuesday, February 28 at 11:59pm CT

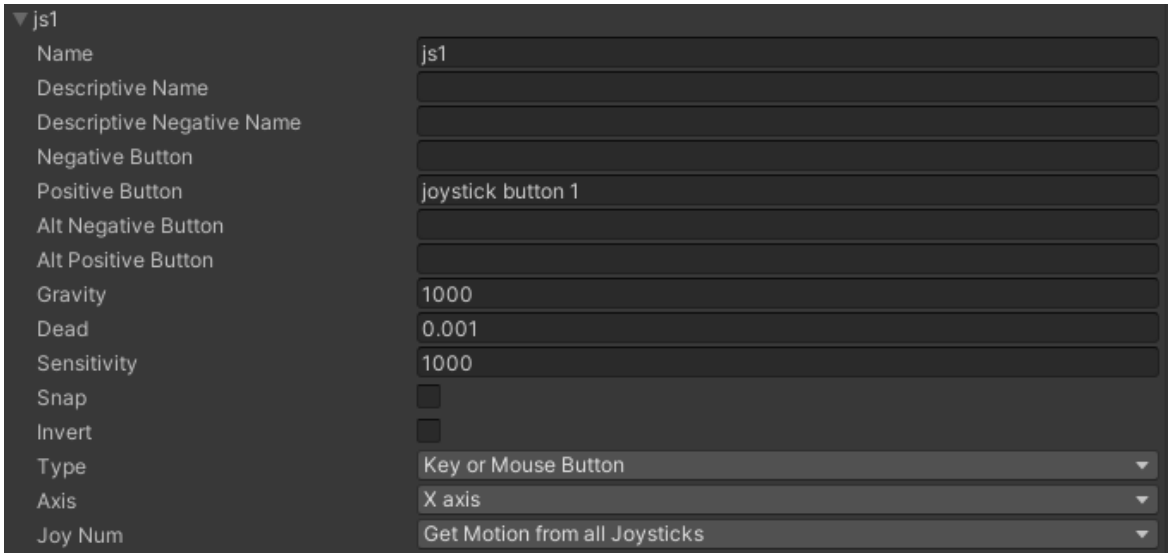
Purpose

Learn how to use Unity and the Google VR SDK for Unity to develop a 3D travel and manipulation technique for VR.

Directions

1. Open your "CS6334 + *your net id*" project and create a new scene called "assignment02".
2. Assignment #2 should be built upon Assignment #1. Make sure that you have all the proper prefabs and game objects placed correctly in a new scene. You can reuse the virtual environment that you created for Assignment #1.
3. For Assignment #2 (and Assignment #3), you will need to download and install the Unity package called CS6334-VR. Some basic functionalities like character controller, movement script, and reticle pointer are included in this package. Download the CS6334-VR package from eLearning under Downloads and install the package.
    - Import this package by going to Assets -> Import Package -> Custom Package
    - If you get the popup asking to setup TMP, select "Import TMP Essentials", and then close the dialog box.
    - Delete the "Main Camera" in your assignment02 scene.
    - Create an empty object by right clicking in the Hierarchy tab and select "Create Empty", rename it "Character".
    - Click on Character and in the Inspector tab, and then click on "Add Component". In the search dialog search for "Character Controller" and "Character Movement" and click on them to add.
    - In the Character Movement script check the "Joy Stick Mode" checkbox.
    - Click on Assets folder in the Project tab and search for "XRCardboardRig" prefab. Drag and drop it inside the Character gameobject as its child.
4. Use the Bluetooth controller and get ready for basic travel, and manipulation.
    - Connect the Bluetooth controller to your PC. You can find it by the name "Fortune Tech Wireless."
    - To use the Bluetooth controller, you need the mapping information between the physical buttons and the keywords (e.g., Fire1, Fire2, Submit, etc.) for scripting. The controller mapping can vary depending on the device and the OS.

- Setup the Input Settings to Register the controller input. To do this, go to Edit->Project Settings->Input Manager->Axes and set the number of axes to 33. For the 15 newly added axes, rename them to be js0-js14 and modify their values to be similar to the image below:



- To check the mapping information, you can find a sample scene under Assets->MiLabCardboardExtension->Scenes->MapBluetoothKey. Run the scene and for each button press on the Bluetooth controller, its mapping will be displayed.

- You can use the Input.GetButton("keyword here") function to use the different controller buttons.

5. Make sure that you can move your "Character" game object using the joystick on the controller.
6. Add 5 interactable objects (2 cubes and 3 spheres) in the virtual environment.
    - When the scene loads, all 5 interactable objects should be in the view of the user. They should not be extremely far or extremely close to the user.
    - Both the cubes should have a uniform scale of 1 meter in all three axes (x, y, z). They should be placed in mid-air at a height of 1m.
    - All three spheres should be placed on the ground and should be of uniform scale 1 meter in all three axes (x, y, z). Spheres should not be placed extremely far or extremely close to each other.
    - Refer to the image below for any confusion regarding the scale and positioning of the interactable objects.

7. Add outlines to the object for highlight effect.
    - Please download and import the object highlight asset from the following link. You can use any other assets for the same purpose.
    - https://assetstore.unity.com/packages/tools/particles-effects/quick-outline-115488
    - Refer to the Readme.txt (Assets->QuickOutline->Readme.txt) file for instructions on how to use this asset.
    - Drag the Outline.cs script to all interactable objects (two cubes and three spheres) that you created.
    - Write a script to enable and disable the Outline.cs script component to control the highlight effect. When the reticle pointer points at any of the interactable objects, the object should be highlighted with an outline color on each object. When the reticle pointer leaves the respective object, the outline should be removed.
    - The color and width of the outline can be changed through the script. By default, the outline color is set to white. If you want, you can alter the color or leave it white as well.
    - Make sure you select "Outline Visible" as the Outline Mode on all the interactable objects.
    - If outline has issues on the android device download the shader files from eLearning under Downloads and replace them under {ProjectPath}->Assets->QuickOutline-> Resources->Shaders
8. Create and write scripts to allow reticle pointer to interact with two cubes (Cube1, Cube2) and three spheres (Sphere1, Sphere2, and Sphere3) as follow:
    - **Translation:** When the pointer points at Cube1 and the button "X" is pressed, the cube should move in the direction of the y-axis. Cube1 should be continuously moved while the pointer points at the cube and the button is pressed and held. Otherwise, the movement of Cube1 should be stopped.
    - **Rotation:** When the pointer points at Cube2 and the button "X" is pressed, the cube

should be rotated in any one direction. Cube2 should be continuously rotated while the pointer points at the cube and the button is pressed and held. Otherwise, the rotation of Cube2 should be stopped.

- **Teleportation:** When the pointer points at any sphere and the button "Y" is pressed, the user should be teleported to that location and that sphere should disappear.
- **Hint:** Use the Event Trigger component together with your scripts.

9. Watch the demo video below to see how the final implementation should look.
    - https://youtu.be/5RZWSUMJIgs
10. Make sure to have colliders on Character and the objects, so that the Character doesn't pass through other objects.
11. Click on File -> Build Settings. Click on "Add Open Scenes". Now, click on "Build" to generate an APK file. Save it as assignment02.apk. If you have your android device connected, you can click "Build and Run" to run the application directly on the device.
12. After building it, save the scene as assignment02.unity.

Submission

1. Clean up your Unity project by removing any unnecessary assets from the "Assets" folder and deleting the project's automatically generated "obj", "Library" and "Temp" folders. Your submission zip file must be **500 MB** or less.

2. Create a "Source" document (.pdf) that provides a unique URL for where you obtained each virtual object within your project. If the virtual objects are created by you, please indicate that you created them by yourself.

3. Record a video showing all the required functionalities from your phone's perspective. You can either use the built-in screen recording feature or download apps like AZ Screen Recorder, OneShot, Unlimited Screen Recorder, etc. The video should not be more than 50 MB and it should not be more than 30 seconds long. Minimum resolution for the video should be 480p (640 x 480).

4. Create a zip file (.zip) that contains your entire "CS6334 + *your net id*" Unity project folder and your "Source" document. Do NOT use any compression fie types (e.g., .rar, .7z, .tar) other than .zip. Such submissions will NOT be graded, which will result in 0 points.

5. Submit the zip file on eLearning under Assignments > Assignment #2. There will be unlimited attempts to submit your work, and the last submission will be graded.

Scoring

This assignment will be scored as indicated below. The maximum possible score is 10 points.

- Cube1 can be translated using the reticle pointer as specified in the instructions. **3 points**

- Cube2 can be rotated using the reticle pointer as specified in the instructions. **3 points**
- You can teleport to any of the three spheres in your virtual environment using the reticle pointer as specified in the instructions. **3 points**
- The outline is displayed when the reticle pointer points at the interactable objects and is removed when it leaves. **1 point**

Deductions

Deductions will be applied as indicated below. The minimum possible score is 0 points.

- Your virtual environment contains inappropriately scaled or unrealistic virtual objects. **1 point per object**
- Your interaction schemes do not follow the direction (teleportation, interactions with cubes and spheres). **2 points per interaction scheme**
- You did not follow the direction correctly. **1 point per instruction step**
- Your submission is late. **2 points per day late**
- Your submission is not a .zip file. **10 points**
- Your submission is larger than 500 MB. **1 point per 50 MB over**
- Your Unity project does not properly work during initial grading. **5 points**
- The demonstration video does not show complete functioning of the assignment, or no video has been submitted. **1 point**
- Your virtual environment does not contain the Character object or the Character object does not move using the Bluetooth controller. **2 points**
- Your supplementary files are not of the specified formats or do not contain the specified information. **1 point per file**
- You did not follow the specified naming conventions. **0.5 point per file or folder**

Regrade Policy

For programming assignments, you have a window of 7 days (from when we return your assignment) to ask for a regrade. We will not consider any regrade requests outside this window. Regrade requests should be emailed to TA; regrade requests will not be considered unless they contain a clear explanation on why a regrade should be issued. The TA will respond to your regrade request within 72 hours of

receiving it.

Academic Integrity

This is an individual assignment. Each student is expected to complete his/her own work. If found guilty of academic dishonesty, you will receive 0 points on this assignment. Below is a list of things that are considered as academic dishonesty or not:

Considered Academic Dishonesty:

- If you download and copy an already developed scene from someone else or from the Internet, it will be considered academic dishonesty. Copying the scene and making changes in it is still considered academic dishonesty.
- Sharing your actual program code with other students is considered academic dishonesty. You must not share the actual program code with other students. You should not ask anyone to give you a copy of their code or, conversely, give your code to another student who asks you for it; nor should you post your solutions on the web, in public repositories, or any other publicly accessible place.
- You must not look at solution sets or program code from other years. Looking at solution sets or program codes from other years is a dangerous practice. Most assignments change in a variety of ways from year to year as we seek to make them better.
- Copying scripts directly from other students or the Internet is considered academic dishonesty. Copying the script and making slight changes (e.g., variable names) still considered academic dishonesty.

Not Considered Academic Dishonesty:

- Materials, prefabs, and 2D/3D objects downloaded from Unity Asset store or other sources are allowed to use if they are not from other students (including current and previous semesters). However, the source of the assets should be clearly described in a Source document.
- You can refer to scripting solutions on the Internet, try and understand it and then write your own scripts. However, the source of the scripting solutions should be clearly described in a Source document.

Every submission will be checked for plagiarism. If found guilty, you will receive 0 points for this assignment without any exceptions, and your case will be reported to the department and/or university for further action.

These descriptions and timelines are subject to change at the discretion of the professor.