

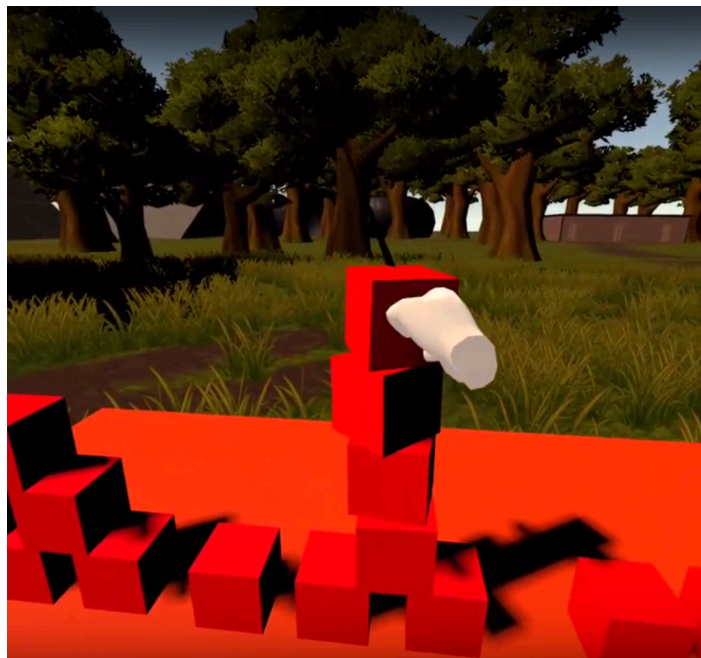
Project 1 - Hello Virtual World

Due: February 2th 2024 noon

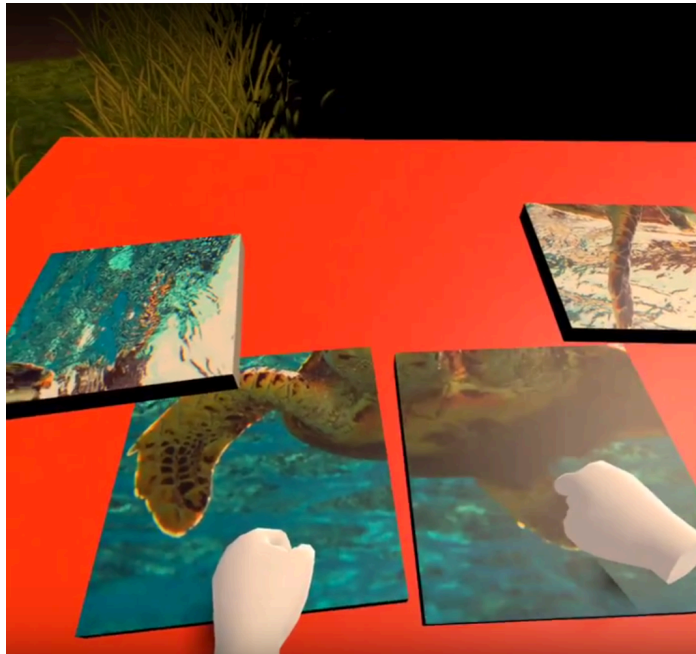
In this project, you will learn the basics of creating an Oculus Quest application using Unity. You are provided with a large virtual environment (VE) and you will add behavior so that the user can interact with objects within the VE. You also will implement two locomotion techniques to allow the user to traverse the VE. To learn more about Unity and C#, read the labs or tutorials below. **Ask on Piazza when you encounter any problem.**

Basic Functionality Requirements:

1. **(Undergraduates: 33%; Graduates: 25%) Interaction:** You must be able to grab and interact with objects using the handheld controller. For this requirement, you may use the OVR Grabbable script provided in the Oculus Integration package. These grabbable objects must also utilize the built-in Unity physics engine by having a rigidbody and a collider attached to the object. Here is a simple tutorial of how to grab an object: [How To Add Object Grabbing for Oculus Quest in Unity](#) .
 - a. One of the grabbable objects created must be a cube. In addition the cube must be textured.
 - b. Another grabbable object must be imported from an outside source. The Unity Asset store has several free models that can be imported into your project. As an example, you could have a sword to swing at the cubes. Depending on the model you import, you may need to add your own collider components and adjust them to match the shape of the object.

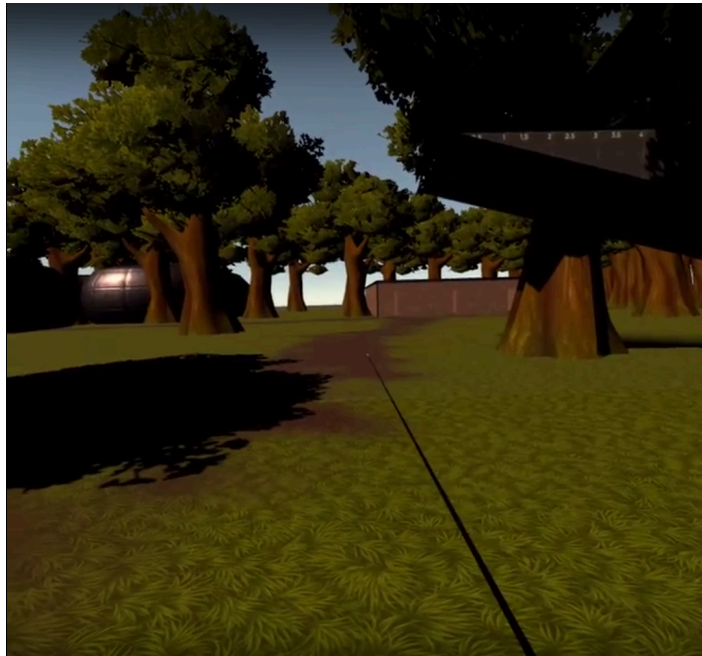


2. **(Undergraduates: 33%; Graduates: 25%) Puzzle:** The provided starter project comes with 4 puzzle pieces (located in assets > prefabs) that form a complete puzzle. You must place each of the pieces in scene and allow for the player to assemble the puzzle. You can either place these pieces jumbled up on a table, or spread out around the environment.
- You must write a script that detects when the puzzle has been completed. One way to do this is to create empty gameObjects at the center corner of each puzzle piece, and detect when the four center corners are sufficiently close together and oriented correctly.
 - When the puzzle is completed, the player should be notified with a method of your choice.
 - For example when the puzzle is completed, you could have a UI element that displays a message, or a sound play. You could also cause an object to spawn when the puzzle is completed.

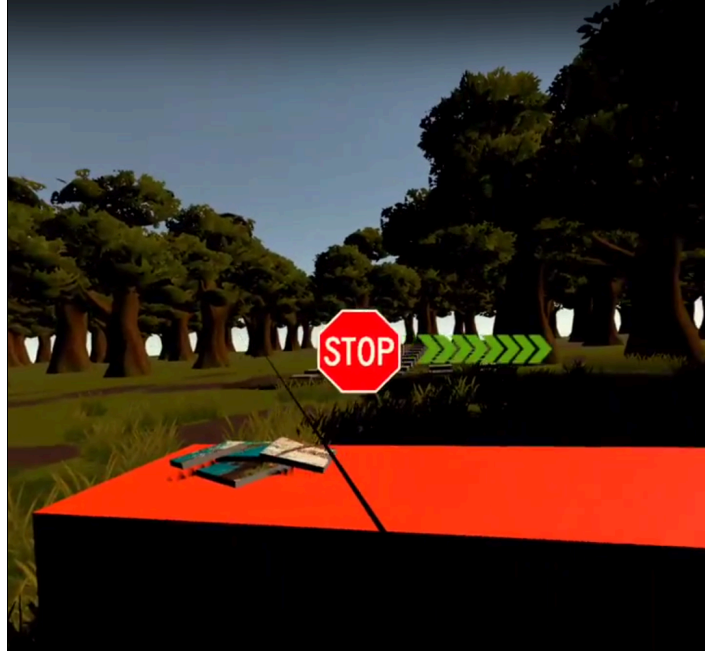


3. **(Undergraduates: 34%; Graduates: 25%) Teleportation:** You must provide a way of traversing the virtual environment through teleporting. Use a laser pointer to define the position to travel to. Pressing the primary button (either A or X depending on the controller) should teleport you to that location.
- You may incorporate the UIHelpers prefab which provides a laser pointer. However, you must create the script that teleports the player yourself. You can get the UIHelpers prefab by downloading the Oculus integration from the Unity store. You can also learn from the Meta XR interaction SDK OVR samples downloadable from the Unity store.
 - When teleporting, the user should maintain their same height from the ground.
 - One special case to consider is when the teleportation destination selected with the laser is not a point on the ground plane. In this case the user should be

teleported to the ground point obtained by translating the destination point closer to the user on the laser line and then projecting the translated point to the ground for the actual destination.



4. **(Graduate only: 25%) Reset User at the Boundary of Physical World:** You must implement a method for reorienting the user when they get close to the boundary. When a user reaches the edge of the boundary, the user should be notified to stop and turn. During this procedure, apply a gain of 2x is applied to the rotation so that the user turns 180 degrees in the tracking area, while rotating a full 360 degrees in the virtual world. The net result of this procedure is that the user is facing away from the boundary, while facing the same direction as they were before the reset.
 - a. You will need to use the OVRBoundary Guardian System API in order to get the user's distance from the boundary. A reference guide is located at [Integrate Guardian | Oculus Developers](#).



5. **(Up to 20%)** As extra credit:
 - a. Shoot cube that defines destination
 - b. Shoot camera to define destination, see picture in picture of what the camera sees, and then decide to teleport to that position or not
 - c. Multiperspective teleportation. See the following paper for more details:
 - i. [Wu, Meng-Lin, and Voicu Popescu. "Anchored multiperspective visualization for efficient vr navigation." *International Conference on Virtual Reality and Augmented Reality*. Springer, Cham, 2018.](#)
 - d. Shooting/collecting stationary/dynamic objects
 - e. A feature you would like to add, contact instructors

Turnin instructions:

All projects must be turned in through Brightspace.

You will be required to create a video that displays the functionality and features of your project. The video should demonstrate ALL required elements of the project. The video should also demonstrate any of the “extras” have been incorporated into the project.

Deliverables:

1. Video demonstrating features of the project. **(For video capturing, see the tutorial below)**
2. The entire Unity project.
3. The Android package (APK) build. This archive file (should have the .apk extension) is created when the application is built and contains all the resources necessary to run the application on the Oculus Quest.

4. A Readme file which contains any extras you incorporated into the project.

Resources

New to Unity Development

1. **Unity Official Documentation:** Unity official documentation is always a good place to consult first for questions about the systems and interfaces in Unity: [Unity - Manual](#)
2. **Unity 3D Workshop:** [James Simpson GGDA Unity 3D Workshop #1](#). This workshop is more targeted towards users without coding experience, but it is still a good tutorial for learning the basics of Unity. I would recommend this workshop if you have never worked with Unity before.
3. **Unity Overview:** [Manual: Working in Unity](#). This includes the following sections:
 - a. Asset Workflow: [Manual: Asset Workflow](#)
 - b. The Main Windows: [Unity's interface](#)
 - c. Creating Gameplay: [Manual: Creating Gameplay](#)
4. **Unity Scripting guide:** [Manual: Scripting](#)
5. **Unity Official Scripting Videos:** These also serve as a good introduction to C#.
 - a. Beginner Scripting Playlist: [C# Scripts as Behaviour Components in Unity! - Beginner Scripting Tutorial](#)
 - b. Intermediate Scripting Playlist: [C# Properties in Unity! - Intermediate Scripting Tutorial](#)
6. **Unity Collaboration:** Since you are working in teams, it may be useful to have a method to save, sync and share a Unity project between team members. Unity has a built-in collaboration service to save the project in a cloud-hosted environment. See: [Manual: Unity Collaborate](#) for more details. Note: that basic version of Unity only allows 1 GB of storage, which may be an issue if you choose to import large assets into your project.

New to C# Development

7. **Comparison of C# for C++ Developers:**
 - a. C# does garbage collection for you
 - b. C# doesn't support multiple inheritance
 - c. C# can use pointers with the unsafe keyword. Note: if you want to use pointers in Unity you will have to modify the project settings. See [How to use unsafe context in Unity - Stack Overflow](#). In most cases you can just utilize object references instead of pointers.
 - d. C# does not automatically create a default constructor for a class.
 - e. C# allows the switch statement to be used with string values.
 - f. Arrays in C# are reference type rather than value type.
 - g. In C#, structs are of value type.
 - h. C# does not support default arguments.

- i. If you have programmed in Java before, C# and Java both have very similar basic syntax.
8. [C# Programming Guide](#)

New to Oculus Development

9. **Oculus Unity Getting Started Guide:** [Set Up Development Environment | Oculus Developers](#)
10. Information on getting input on the Quest controllers with OVRInput: [Map Controllers | Oculus Developers](#)
11. Tutorial on how to setup Unity and build your first Oculus Quest application (note: you can skip step 2, it is unnecessary as long as you install Android Tools during the Unity installation process): [How to get started with Oculus Quest development in Unity - The Ghost Howls](#)