

CS 498: Virtual Reality

MP 1: Unity3D Basics

Due: Sept. 15th

In this MP you will accomplish the following: you will gain some VR experience by running various demos, then gain some Unity expertise.

Part 1

Your first assignment in this MP is to choose and try out five of the VR demos available on the lab machines in the (C:/VR-Demos) directory. You should then create a write up of these experiences. As the write up, answer the following two questions about each of the 5 demos that you chose. The questions are quite simple. Firstly, describe each of the 5 demos. Secondly, describe your experience with each demo i.e. state some pros and cons.

Your answers need only be 2 to 3 sentences for each question. Last, we would like you to find and suggest another VR demo for us to add to our library. You can begin your search here: <https://share.oculus.com/>. Please include all this information: your choices, your question answers, and your new demo suggestion, in a PDF named MP1DemoWriteUp.pdf. Submission is covered in the submission section of this document.

Part 2

The second assignment is meant to familiarize you with the Unity3D game engine. You will learn the basics of creating and lighting a scene, manipulating gameobjects, using prefabs, and writing scripts. You'll create a simple environment, detailed below.

Room: The room should be constructed entirely out of cubes and should be roughly 10x10x10 units large. Every inner surface of the room (walls, ceiling, floor) must have a solid color (not black or white).

Stimuli: Place 3 cubes of size (0.5, 0.5, 0.5) in the room.

- 1) Lying flat on the floor
- 2) Suspended in mid-air.
- 3) Suspended in mid-air. Add a rigidbody component to this cube so that it falls to the floor when the scene begins.

Player: Place an OVRPlayerController prefab (Assets/OVR/Prefabs/PlayerController) in the room, so that all 3 cubes are visible when the scene starts. The character should not be able to walk through any walls. Rotation and position tracking are enabled by default on the OVRPlayerController prefab.

Lighting: The scene should contain a directional light that casts soft shadows (you must enable shadows by modifying the Shadow Type variable on the Directional Light gameobject using the Inspector). Place a point light in the center of the room.

Scripting: Write a script ('Lightswitch') that turns the point light on or off (either by activating the gameobject or increasing the light intensity) by pressing Tab.

Challenge: Write a script ('ProximityDimmer') that modifies the intensity of the point light based on player distance from the center of the room. The pointlight becomes brighter when the player approaches the center of the room, and it becomes dimmer when the player moves away from the center.

Use parent and child gameobjects to keep your scene organized. When you are done, the only parent objects visible in the Hierarchy should be Room, Stimuli, Player, and Lighting.

Here are some things I wish I had known when I made my first scene in Unity:

- Hold v and drag a gameobject corner to align it with other gameobjects
- Ctrl + d will copy gameobjects
- Ctrl + Shift + n will create an empty gameobject. You can use this as a parent object

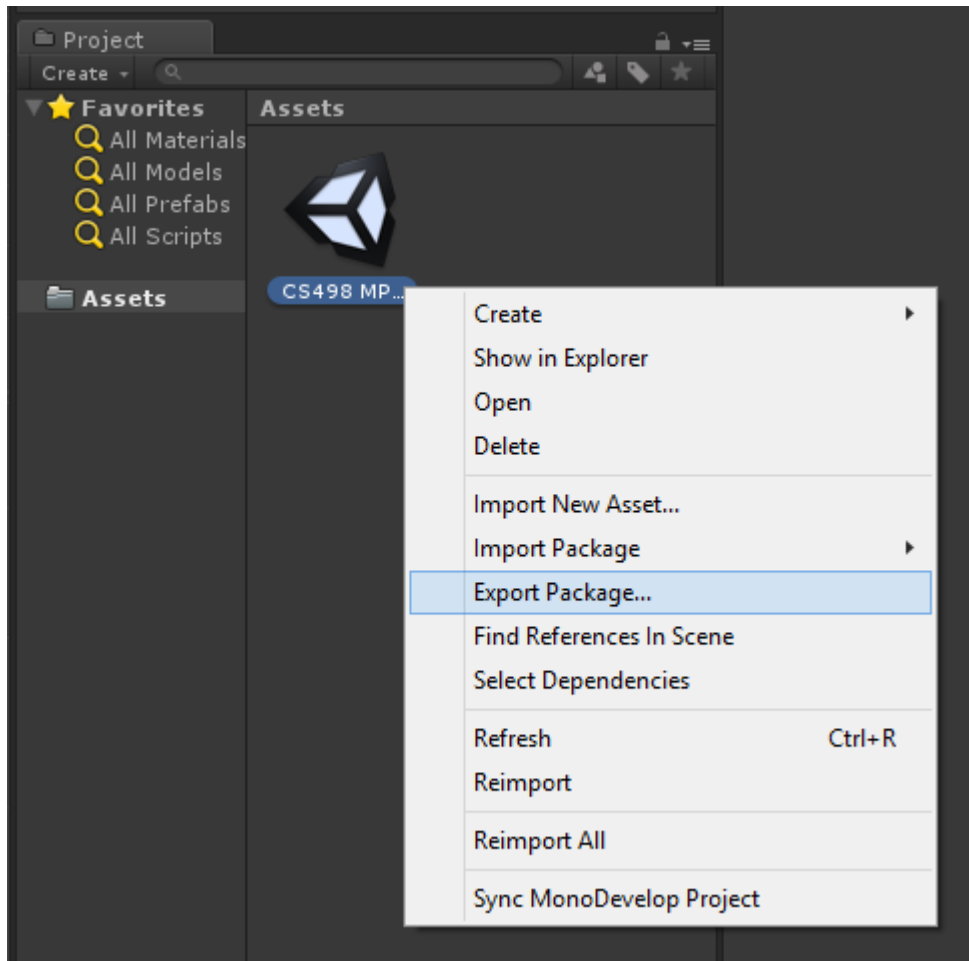
Rubric:

Room	20%	Correctly sized room made entirely out of cubes
Stimuli	20%	Correct stimulus cube size, position, and gravity.
Player	20%	All 3 cubes are visible at start of scene. WASD movement. Can't walk through walls.
Lighting	20%	Proper placement of both lighting sources. Point light is disabled at scene start.
Scripting	10%	Pressing tab enables/disables the point light.
Challenge	10%	Point light brightens and dims with player distance from the center of the room.

How to Submit the Assignment

Step 1: Create a .unitypackage file

- 1) Save your Unity scene in the Assets folder with the title “CS498MP1”
- 2) Using the editor, find the created scene in the Project menu
- 3) Right click on the scene and select Export Package...



- 4) Export the file using default settings (“Include dependencies” should be checked by default)

Step 2: Create a standalone game build

- 1) Go to Edit → Preferences → OculusVR. Make sure the "Optimize builds for Rift" box is checked

- 2) Go to File → Build Settings
- 3) Click “Add Current”. This will add the current scene to the build. You must have saved the scene to the Assets folder for this to work (you should do that anyways).
- 4) Hit “Build”. **Save the project to C:\Users\student’s netid\project name**, rather than your networked folder.
- 5) This should create a folder containing scene data, as well as executables (.exe) for running the build. For Oculus builds, Unity creates a separate DirectToRift application in the build folder. We’ll be grading the DirectToRift builds, so **make sure the DirectToRift build runs correctly before you submit it**

Step 3: Zip the files and submit them through Compass

- 1) Create a zip file containing 4 items:
 - a) The writeup PDF
 - b) The .unitypackage created in Step 1
 - c) The build folder created in Step 2 (containing a DirectToRift build)
 - d) A README.txt file containing any special instructions or notes you think are relevant for evaluating your MP.
- 2) Name the file by separating NetIDs with underscores._cs498sl_mp1.zip
EXAMPLE: If foobar1 and barbaz2 worked together, the file should be called foobar1_barbaz2_cs498sl_mp1.zip