Project 2 - One More Cup of Coffee

Documentation Due Tuesday January 03-2024

This second project is focused on immersive Virtual Reality at room scale. This project will give you some experience with building virtual worlds for people to move around in and interact with and seeing that world at human scale. It will also give you a chance to experience the world that you create through physically walking and moving your head and hands to directly interact with the space.

Back in the now mythical 1980s when I was a Computer Engineering undergrad at the University of Michigan, I spent more than a few mornings selling doughnuts and hot chocolate for Tau Beta Pi in the atrium of the (then) new EECS building. We are now a year away from our own new Computer Science building opening, so I thought it would be interesting to see how a similar kiosk might work in the new CS building before its finished.

I am supplying a starter kiosk from https://assetstore.unity.com/packages/3d/props/coffeeshop-starter-pack-160914 placed in a very simplified model of the atrium of the new CDRLC building that can run by default in the Oculus Quest headsets in the classroom and in a desktop Unity simulator, allowing you to walk or teleport around the space, and interact with some of the objects. Your job is to augment this kiosk space with additional appropriate models, sounds, lights, interactives. In project 3 we will augment the surrounding space of the atrium.

➢ Implementing the Project

You will be implementing the project in Unity 2021.3.6f1 (as with Project 1) and VRTK version 4.

One of the biggest issues over the last 30 years in VR development has been the lack of standard higher level libraries, and the difficulty in getting code to run across multiple platforms. Right now VRTK V4 is

the probably the best attempt at this, even though it has been in beta for a couple years now. VRTK also has a very nice simulator, letting you do a fair amount of development in Unity itself without a headset. However you will need to test on a real headset. Its impossible to create immersive worlds without testing them immersively as they are so tied to how our bodies inhabit and interact with space. Your projects will be graded based on how they run in the Quest headsets in class, not in the simulator so please give yourself enough time to test.

VRTK has a variety of documentation at https://www.vrtk.io/and various good videos on YouTube here (https://www.youtube.com/channel/UCWRk-LEMUNoZxUmY1wO7DBQ)with live demonstrations of getting small projects running in Unity. There is also a fairly active VRTK community on Discord. Its important when you are looking for documentation that you stick with the V4 documentation and not the earlier V3 documentation as there were many changes.

A good place to start learning VRTK is with the videos at https://www.youtube.com/c/VirtualRealityToolkit/videos, in particular [001], [002], [003], though some of the material is out of date, as dealing with packages is much easier now.

We will be spending some time in class helping people go through these tutorials and helping people get things running on the headsets and in the simulator.

The starter project is available are giving.

You can add this project to Unity Hub and take a look at it in Unity. Grab the CoffeeScene (located in Assets/CoffeeShopStarterPack/Scenes) and drag it to your hierarchy. Delete the default scene. Unity should process lighting for a bit. When you play the scene you should be able to move around with the WASD keys, use the mouse to look around. You can use the 2 key and 3 key to take control of the simulated left and right hand controls respectively. If you place one of the two simulated controllers into one of the red muffins you can use the left mouse button to pick them up. The 1 key takes you back to controlling the player.

If you plug in one of the Quest headsets in the classroom, switch the platform to Android, turn off the CameraRigs.SpatialSimulator and turn on the CameraRigs.UnityXRPluginFramework, you can build and run the same scene on the headset and move around with your body and interact with the real controllers using the trigger on either controller to grab the red muffin. Putting your thumb on the right joystick brings up a teleporter and then pressing the joystick teleports you to that location (note that once you get into position you should not use the teleporter to move around your kiosk - everything should be accessible by walking). Pressing the Oculus button on the right controller will allow you to exit the application.

I highly suggest trying out the project in both simulator and headset mode early in your development process to understand how to swap between them and how interaction works in both of them.

The numbers below are for a single person project. You can do Project 2 alone or in a 2-person team. If you work in a 2-person team then all the requirements are doubled (aside from the frame rate which remains the same).

50% of the points on the project is for creating a basic kiosk with

- 15 unique models from the web (remember to cite the creators)
- 5 unique models that you create on your own (more on this below)

- at least 5 of the models need to have appropriate physics and collliders to be grabbable, droppable, and tossable, and collide with other parts of the kiosk and the floor
- create a new appropriate lighting scheme with at least 2 new lights for the kiosk
- at least 1 relevant (and not annoying) ambient sound or piece of music

30% of the points on the project are for ...

- the user needs to be able to interact with at least 2 objects in the scene using their hand and have each of those objects produce new objects (i.e. every time you open the oven a new pie pops out, or every time you touch the ticket machine it produces a new ticket). Interaction must happen locally with the users hand intersecting the object, not at distance.
- at least 2 of the models should be animated and move, or move some of their parts
- at least 5 more of the models need to have appropriate physics and collliders to be grabbable, droppable, and tossable, and collide with other parts of the kiosk and the floor
- at least 4 unique sounds that get louder as you get closer, or that are triggered by interacting with objects in the scene, or that sound when objects hit other objects.

20% of the points on the project are for ...

- each member of your team needs to use Makehuman and Mixamo to create a 1:1 scale kiosk worker that looks (reasonably) like you, wearing something G rated, with a suitable G rated idle animation. The worker could be trying to attract new customers, or working the register / pad, or ignoring the customers and looking at their phone, or sleeping, etc. The worker should say something when you touch them with either controller.

Graduate students in the class also need to be able to ...

Create a seating area for at least 2 customers with tables and chairs and 1 more animated person seated there. Note that this area may be outside the 10' (3m) square walking area so teleporting may be needed here).

In all these cases the frame rate should remain decent (at least 30 frames per second in stereo on the Quest 1 headsets). Ideally the frame rate should be more like 60, but with an architectural model involved, 30 is a more realistic minimum.

The most important decision you need to make is what you want to sell in your kiosk. Anything you sell needs to have a physical representation and a price shown - there has to be a 'thing' to pick up and hand to the cashier or a price scanner, even if that thing is a representation of an app or a song or a movie that is digital. You could sell doughnuts and coffee. You could sell toys, books, apps, drinks, pet rocks, tickets to events. That part is pretty flexible and ideally you should sell things that are of interest to you, but please clear your concession with andy. What you sell should also be G rated and generally not offensive. It does not need to realistically be able to make money as a kiosk.

You can remove all of the current items from the kiosk model, rearrange the kiosk model, or even replace the kiosk models with your own kiosk models, as long as it looks like it could function as a kiosk in that space. The walls, floor and ceiling of the CDRLC are off limits in this project - enhancing those will be project 3. You can not remove or change any models from the CDRLC and SEL building.

Be careful when you are collecting or creating models for your space as the polygons start to add up. You want to make sure you maintain a good interactive frame rate in stereo on the Quest 1 headsets. For measuring and documenting your frame rate we are going to use the OVR Metrics Tool which is discussed here and there is a nice YouTube video here. It should be pre-loaded on the main classroom Quest and you can see it as an overlay while you are testing your application.

For models you find on the web to use, if you take the entire model and place it somewhere then it counts as one model. If you take it apart and position those pieces separately then each of those parts counts as its own model. i.e. if you have a playground model and place it as a playground then it counts as one model, but if you take the playground model apart and position the swings, and the seesaw, and the jungle gym separately, then that would count as 3 models.

The models that you create on your own need to made of multiple geometric parts and should look like what they represent but do not have to be AAA game quality. The Kiosk scene shows a simplified cartoony representation of reality where the objects are still identifiable.

Keep in mind that your objects can be at different scales - some of them might be the size of tables that help fill the space but others might be smaller objects on a table, but they all need to be visible at some point (i.e. they could be hidden in drawers).

While developing your application you can teleport around the scene but for your final demo to the class and evaluation will be done by walking and not teleporting, so make sure your scene is human scale and walkable within a 10' by 10' (3m by 3m) area. There will be no interaction at a distance during your final demo or the evaluation so make sure people can reach and interact with your objects using the actual tracked controllers in that space.

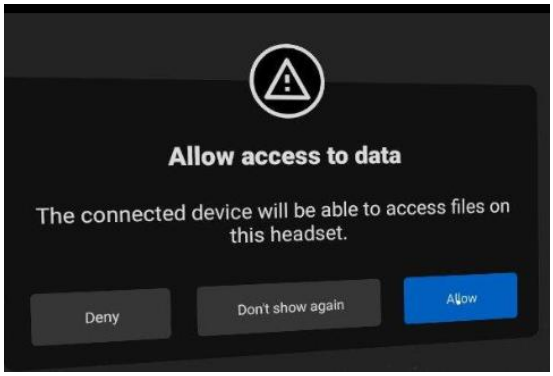
Avoiding some common problems:

- Make sure that either CameraRigs.SpatialSimulator or CameraRigs.UnityXRPluginFramework is on and the other is off depending on whether you want to run in the headset or the simulator. If both are on then things get very confused, or if the wrong one is on things wont work.
- Make sure your Build Platform is set to Android for deploying on the Quest
- If the connected quest is not showing up as a Run Device, check in the headset to make sure you have allowed the connected computer to access files on the quest.
- some models come with built in cameras. These can confuse unity, so be sure to remove or deactivate them.
- this one could have a couple possible causes - the effect is that you see a lot of warnings in the simulator because the VRTK libraries do not completely compile and you can't use the left mouse button to grab. One possible cause is: having a space in the path to get to the project, i.e. /new folder/ and the alternative possible cause is having too long a path to get to the project.
- if unity is not letting you upload your project to the headset, you may want to check and see if there is another copy of Project 2 already there. If that is from another user then sometimes

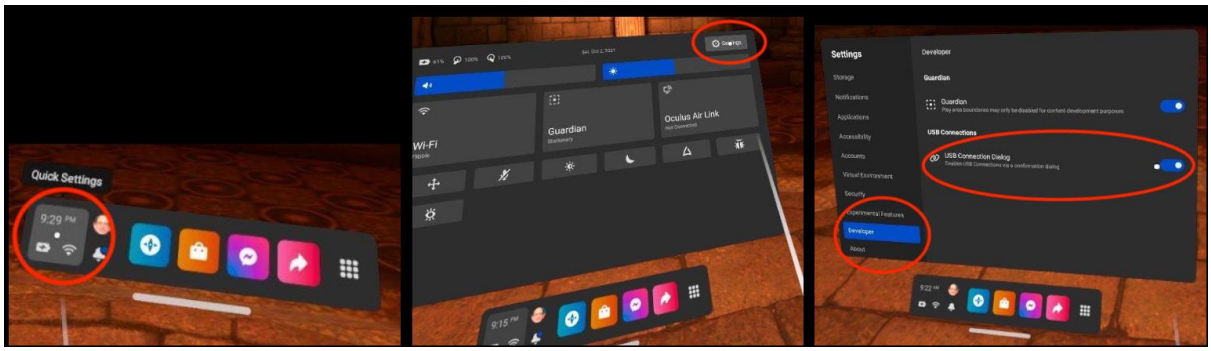
unity doesn't want to overwrite it, especially if you haven't given it a new name, so you may want to manually delete that older version on the headset.

Getting your Unity project to run on the Quest headset:

- connect the Quest headset to your laptop via a suitable USB cable (one that passes data)
- put on the Quest headset and use the controllers to tell the Quest to accept the PC connection and always allow that connection.



If you don't see that appear in the Quest then you should make sure that developer mode is on. It should be on for all the evl headsets. Click on the Quick Settings menu on the left side of the thin menu, then click on the settings menu on the upper right, then scroll down in the menu on the left to Developer and turn on the USB Connection Dialog.



- in your Unity project hierarchy make sure:

  - you have CameraRigs.UnityXRPluginFramework enabled

  - you have CameraRigs.SpatialSimulator disabled

- under Unity Build Settings make sure:

  - you are building for the Android Platform

  - the particular Quest you connected should show up under the Run Device list of compatible connected devices. If the Quest headset does not show up in the list you may need to unplug and re-plug the USB cable and again tell the quest to accept the connection to the laptop

you may want to save the project and restart unity to make sure Unity re-configures itself appropriately

- click on Build and Run

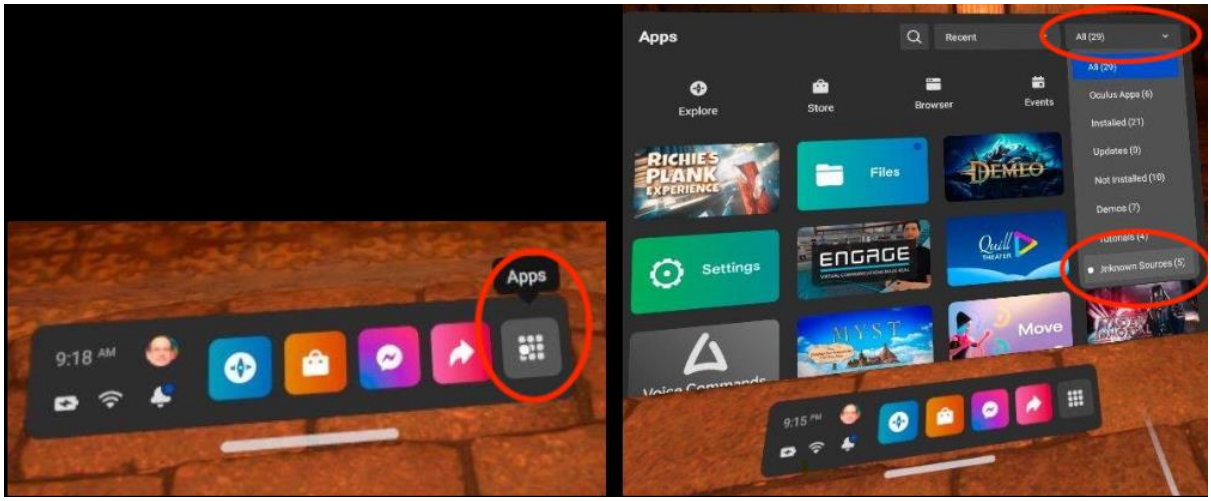
it may take a while (5-10 minutes) the first time while Unity converts all the assets to a form suitable for the Quest. Next time it will be faster.

- disconnect the USB cable from the Quest (so you don't pull your laptop off the table, or trip on the cable while in VR)

- put on the Quest headset and grab the two controllers

- make sure you have enough space around you so you wont hit anything (your laptop, the table) or anyone

- your app should start automatically. If it doesn't, or if you just did a Build but not Build and Run then click on the application menu (9 dots in a grid on the right of the thin menu) then in the upper right of the big menu click on all Applications, then Unidentified Developers at the bottom of the list and click on your application to start it



Once your application is running:

- make sure your hands / controllers are tracking

- make sure you can physically turn around and walk around and the world reacts correctly. If not you may still be running the simulator configuration and you need to go back to Unity and make sure you are only running the CameraRigs.UnityXRPluginFramework.

The Quest has Stationary (sitting in a chair) or Room Scale (walking around) tracking. While you can test in Stationary mode, the app will be demonstrated and evaluated at Room Scale without teleporting in the classroom so you should do a fair amount of testing in that mode.

If you want to remove your project from the Quest then you can go back to Apps / Unidentified developers and find your application. If you click on the kebab 3-dot menu to the right you can uninstall the application.

If you want to cast your application to one of the walls in the classroom (or to your phone) to show your work to others then go to the Sharing menu on the right side of the thin menu, then choose Cast from the upper left of the Sharing Menu, and then pick the destination device. The headsets should all be on the VRtheater wireless network giving them access to the Chromecast. You can check by clicking on the Quick Settings on the left as before.

If you want to take screenshots on the quest for your documentation you can quickly press the Oculus button on the right controller and then press any button on the left controller. You will see a dialogue telling you that a new photo was added to the collection. If you connect the Quest to your computer and you have allowed Unity to install apps on the Quest then you should be able to see the files on Windows, or use Android File Transfer (https://www.android.com/filetransfer/) on the mac. You can see all the snapshots under Oculus/Screenshots/

Turning in the Project

You should create a GitHub repository for your project. You can integrate Unity with git so that it will track all your changes, or just regularly push files to git. The final project will need to be turned in via git so we know the timestamp on the files, but it can be also helpful to have regular commits in case something goes wrong so you can get partial credit. Initially this repository should be private to yourself, and then you can make it public for turning it in.

Note that there is a very big difference between getting something working and getting it working well. The first is not that hard. The second takes much more time. You are expected to have things working well.

There are two due dates for the project.

The unity source and application is due first. This will be turned in via GitHub by making your repository public to at least everyone at UM. Be sure to email the location of your repository to Andy before the deadline.

The second deadline is for the documentation.

You should create a public web page with multiple sections (visible to anyone for at least the duration of the course) that describes your work on the project. You can host your web page at UIC

(http://people.uic.edu), GitHub, or the provider of your choice, as long as it remains publicly available to all. You can use any publicly available templates as long as you cite them, or create your own.

This page should have several sections including:

- introduction and description of how to use your application and the things you can do with it. What is your kiosk selling?
- link to your git page that allows someone to easily download the source code to your entire project to be built and run on the Quest. This page should have instructions on how to build your application and list the supported version numbers of all relevant software (Unity, VRTK, etc.).
- listing of the source for any assets (models, textures, sounds, music) that you used that you didn't create yourself. and a list of which models satisfy which project requirements (e.g. if 15 models are required, or 5 need to be grabbable, then please give us a numbered list of them including showing them in the scene)
- show a screenshot, or several, showing your app running in the headset with the frame rate showing, and describe how the frame rate changes in different areas of your scene or during different activities
- link to a 5 minute video showing off your project (see below)
- at least a one page / 500 word discussion of how viewing and interacting with your world is different in the simulator and the headset.
- all of which should have plenty of screenshots with meaningful captions. Web pages like this can be very helpful later on in helping you build up a portfolio of your work when you start looking for a job, giving you something to point potential employers at, so please put some effort into it.

My sample scene included these resources:

- UFO model by hana - https://3dwarehouse.sketchup.com/model/273a556e078133532296d88107d065f6/UFO
- UFO sound by Daniel Simion - https://soundbible.com/2213-Alien-Spaceship-UFO.html
- Cow model by Max M - https://3dwarehouse.sketchup.com/model/288b4047-1c6b-43e5-9b76-eb14ff9152fa/Cow
- Grey alien model by Tainted Angel - https://3dwarehouse.sketchup.com/model/c3c00f04-75b2-4500-a3e8-d0664c2be100/Alien-Grey avatar idle animation from Mixamo.com

You should also create a 5 minute YouTube video showing the use of your application including narration with decent audio quality. That video should be in a very obvious place on your main project web page. Capturing video from a headset would be best, but you can capture interaction from the desktop simulator. You can try to narrate while interacting but you will most likely find its useful to do some editing afterwards to tighten the video up. This video is helpful for me to know which parts of your project work, and may be useful in the future when you want to show off some of the projects you did but you can't get the code to compile anymore or you don't have the hardware handy.

Once you have your web page done, send the URL to Andy and the TA before the deadline. I will respond to this email as your 'receipt'.

We will be linking your web page to the course notes so please send Andy a nice representative jpg image/photo of your application running highlighting your chosen phobia for the web. This should be named p2.<your_last_name>.jpg and be roughly 1024 x 768 in size.

Presenting the Project

An important part of creating VR applications is getting feedback and using it to improve your design. This also allows you to see how others approached the problem, and maybe they solved some problems that you had issues with, and vice-versa, so it helps everyone get better for the next project.

We will be spending time in class for each person or group to show off their work live using one of the Oculus headsets streaming to the big classroom wall.

Links to some Projects:

https://github.com/FarahKamleh/Pokemon-Themed-Kiosk-VR

https://www.youtube.com/watch?v=rchKpBvVKxM

https://github.com/angelica-v/angelica_villegas.428.P2

https://sites.google.com/uic.edu/cs428-angelica/project-2

https://www.youtube.com/watch?v=CjS-CUZA7JA