

VR/AR CA2 Report – Personal Room

Conor Kilbride – N00193286

Contents

Design	3
Technical.....	7
Development Section	7
Testing	12
Reflection	13
References	14

Design

VR Project Design Document

03 | 11 | 2022

Conor Kilbride

1

App Info

Tentative Title:

Personal Room

Education & Training

Mental Health & Fitness

Travel & Discovery

Media & Entertainment

Productivity & Collaboration

Gaming



Art & Creativity

Other: _____

2

Pitch

To goal is for users to [learn | experience | practice | review | design | **play** | other]:

Play a game of Billiards against each other and interact with objects around the room that showcase what we've learned in class.

This will be especially [impactful | educational | memorable | effective | **fun** | other] in VR b/c:

At a high level, during the app, users will:

Interact with video and audio sources, light switches and particle systems, teleportation areas, a UI canvas and play a game of Billiards

This experience will be targeted at devices with:

[3 6]	degrees of freedom, giving users control over the	[rotation movement & rotation]	of their head & controllers.
---------	--	--	------------------------------

3

Basics

The app will take place in:

and the user will get around the scene with:

[teleport continuous other N A - user will be stationary]	movement.
--	-----------

The user will be able to grab:

- **Devices with Video, Audio and Light Sources**
- **Cue Sticks for billiards**
- **An object with a Particle System**

There [**will** | will not] be sockets:

- **On the rack for the Cue sticks**
-
-

4

Events & Interactions

There will be haptic / audio feedback when:

- **A user hovers over an interactable object**
- **A user selects an interactable object**

There will also be 3D sound from:

- **A fireplace, a record player, and a television**
-

If the user is holding:

TV Remote	and presses the trigger,	A video will play on the TV
Flashlight	and presses the trigger,	The flashlight will turn on and off
Phone	and presses the trigger,	A video will play on the phone

	Suggestions: a UI change, a sound/video plays, a particle plays, an object is spawned or destroyed.
--	---

By default, the left hand will have a:

[Direct | **Ray**]

interactor.

and the right hand will have a:

[**Direct** | Ray]

interactor.

And you [**will** | will not] be able to toggle on a [Direct | **Ray**] interactor using the [**thumbstick** | button].

The main menu will be located:

In front of the user when the scene loads

and from the main menu, the user will be able to:

- **Reset the scene and access Settings**

-

[Optional] There will be additional UI elements for:

- **Enabling Snap Turning**

-

5

Optimization & Publishing

To make the user experience more accessible / comfortable:

- **There will be a fade canvas in place when the scene loads and when the user teleports**

- **Optimized to visually run smoother**

-

Given that this app is targeting the [Oculus Quest 2], target metrics are:

Frames per second:	>= _____	FPS
Milliseconds per frame:	< _____	ms (= 1,000 / FPS)

Triangles per frame:	<200	tris
Draw calls per frame:	_____ - _____	batches

Lighting strategy:

	All baked
--	-----------

✓	Mostly baked with some mixed
---	------------------------------

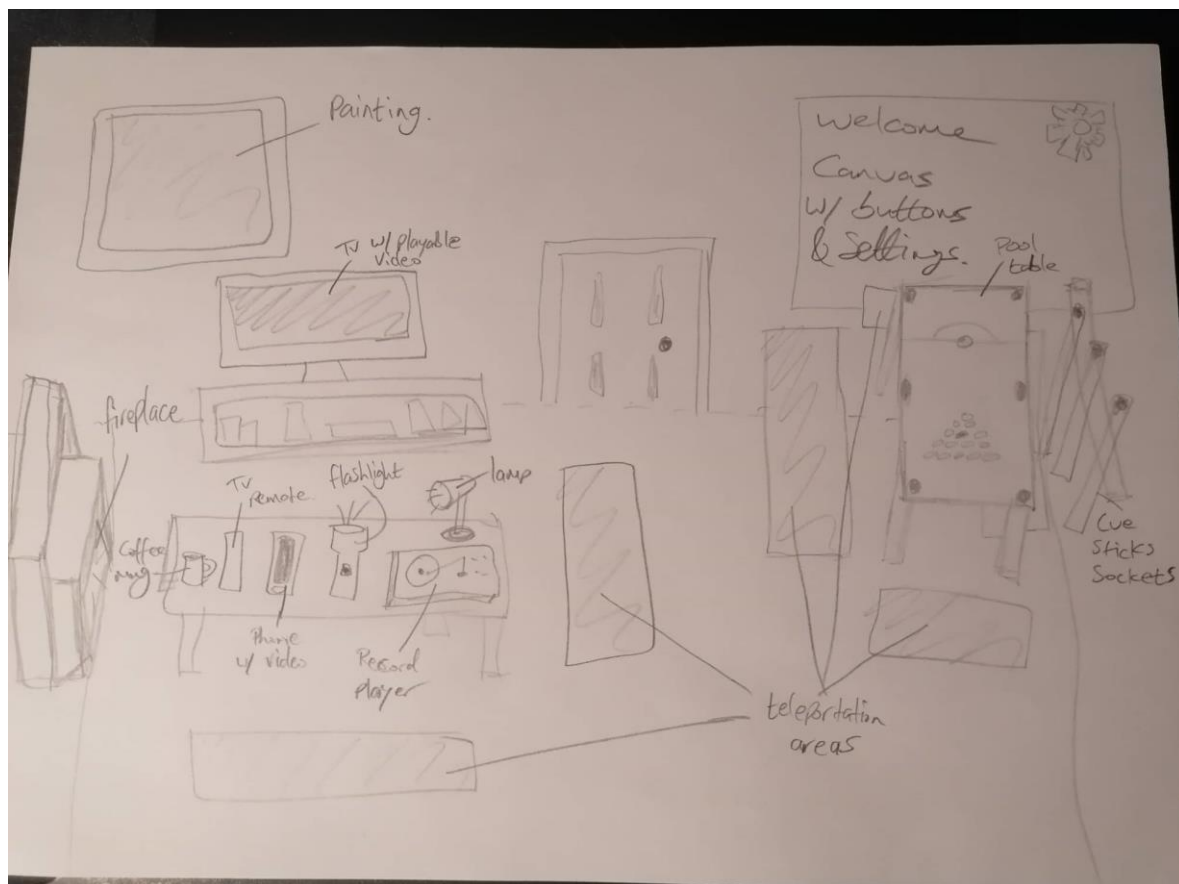
	All real-time
--	---------------

Light probes [**will** | will not] also be used for more realistic mixed lighting.

6

Sketch

(Optional)



Technical

For this project we used the Oculus Quest 2 VR Headset. A lot of things about the Oculus Quest 2 make it a suitable headset for this project. It has wireless capabilities which makes using apps more comfortable and immersive for the user. It's also in a good price range, making it more accessible than other headsets. It supports 6 degrees of freedom with head and hand tracking through integrated Oculus Insight technology which is exactly what we want for this kind of project. It has a display resolution of 1832x1920 per eye and 256GB of storage. It uses the Qualcomm Snapdragon XR2 platform. There are alternative options but the Oculus Quest 2 has the best balance in specifications and for the price range I think it was definitely the best headset to use.

The framework I used when developing this application was the XR Interaction Toolkit. This is a "high-level, component-based, interaction system for creating VR and AR experiences." It makes 3D and UI interactions available from Unity input events. It does this through Interactor and Interactable components, with an interaction manager to link them together. It also allows for cross platform controller input, hover, select and grab components, haptic and visual feedback, UI canvases. These made this framework very effective at developing this project.

Development Section

To begin the development of this application I started to build the pool table. There were no free assets on the Unity assets store I could use so I decided to make it myself within Unity. I started by making the table's surface and adding a raised perimeter. Unity doesn't allow you to cut or remove parts of objects, and would require me to create my asset in Blender. Instead, to make the 6 pockets around the table's edge, I remade the perimeter to include holes on each corner and each side along the length of the table. After this I wanted to make my cue stick and pool balls. I created a cylinder object with an attach point for the cue and a sphere I could then duplicate for my pool balls. I coloured the balls and arranged them correctly on the table. As I could not create holes in the table's surface, I added 6 black circles to sit on top of the pockets to make the table look more accurate and added colliders to each of them. On each collider I added scripts so that when one of the balls collided with it, it would trigger the ball to disappear. I later added a script so that if the black ball was potted, the scene would reload. I tried writing my own script for the cue ball to respawn on the table if it were potted but it caused problems with another script so I removed it, and made it a grabbable component so that the user could place it where they wanted if the opponent potted it. I made a Balls Physics material and a PoolTable Physics material to apply to the table and balls. I duplicated my cue stick and added a Hook object from an assets folder to act as my cue rack. On the Hook object I made a child object named Socket 1 with a sphere collider and a Socket Interactor component. For this socket I made another child object with an Attach object. I moved and rotated the attach object to where I wanted it on the rack and assigned it to the Attach Transform property of the socket interactor. I

duplicated the socket 2 times and place them correctly on the rack. I then made a CueStick Layer to assign to my cue sticks and the Interaction Layer Masks of my socket objects so that only the cue sticks could interact with my sockets.

Using the assets folder provided with one of Unity's online tutorials, I added an empty room to my scene and placed the pool table, cue sticks and cue rack where I wanted them. I also decorated the room and added furniture to it. Next, I wanted to add teleportation areas so the user could move around the room. I downloaded an assets folder with various rug designs and placed a few around my room and on each side of the pool table. I added a teleportation area component to these rugs with my XR rig as the interaction provider and the interaction layers mask set to Raycast, so only the users raycasts could interact with them. These would later be changed to teleportation areas with fade components to make the application more comfortable for the user. After this I added audio and haptic feedback to the controllers. In the interactor components, under the Audio Events and Haptic Events dropdowns, I ticked the select entered and hover entered boxes and chose my audio clips to play. Now when a user hovers over an interactable object, an audio clip will play and the controller will vibrate, and when the user selects an interactable object, a different audio clip will play and the controller will vibrate differently.

I put a TV on a table on one side of the room and started to add a video source to my application. In the "Screen" child object of my TV, I made a Video Player component and changed the Material Property to _BaseMap to project the video to its material rather than its texture. I made an Audio Source component and set the spatial blend to 1 so the audio will sound different depending on where the user is in the room. I then added a Play Video script and put the video I wanted to play into this script. I put a remote control from my assets into my scene and followed the same steps for the TV audio to add a click sound to the remote when it's activated. I also added a change material script to the indicator light at the top of the remote, so when a user activated it, the light will turn red and turn off when it's deactivated. Finally, under the activate events section of the remote, I added a PlayVideo.TogglePlayStop script to work with the TV screen, so that when a user activates the remote, the video will play or stop. Next I placed a phone to play a video on the table because it followed the same steps as the TV, making a video player, audio source and play video component and putting the video I wanted into the play video script with a TogglePlayStop script in the activate part of the phone object.

I decided to add a lamp and an interactable flashlight because they're also similar to make. From my assets I put a flashlight on the table in my scene and created an Attach object and a "Light" object with a Light component and a Toggle Light script, with "Is On" unchecked so the light will be off when the scene loads. I made the Light component a spotlight changed the colour and range of the light and placed it correctly on the flashlight object. I added an audio source component and a play quick sound script to the flashlight object with a click audio clip. In the activate interactable events, for the flashlight I put a PlayQuickSound.Play script, so the click sound will play when the flashlight is activated, and I put a ToggleLight.Flip script for the light object so the light will switch on or off when it's activated. The lamp was very similar to make, but instead of an XR Grab Interactable

component it needed an XR Simple Interactor. It also needed a Light child object, an Audio Source and a Play Quick Sound with a click sound, and in the Select events section of the interactable component it needed PlayQuickSound.Play script to play the click sound when selected, and a ToggleLight.Flip script to switch the light on and off.

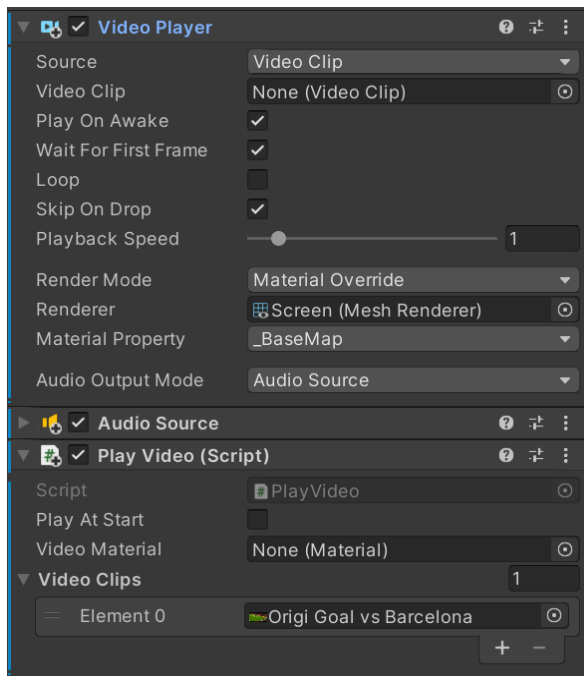


Figure 1 - Scripts on the TV screen

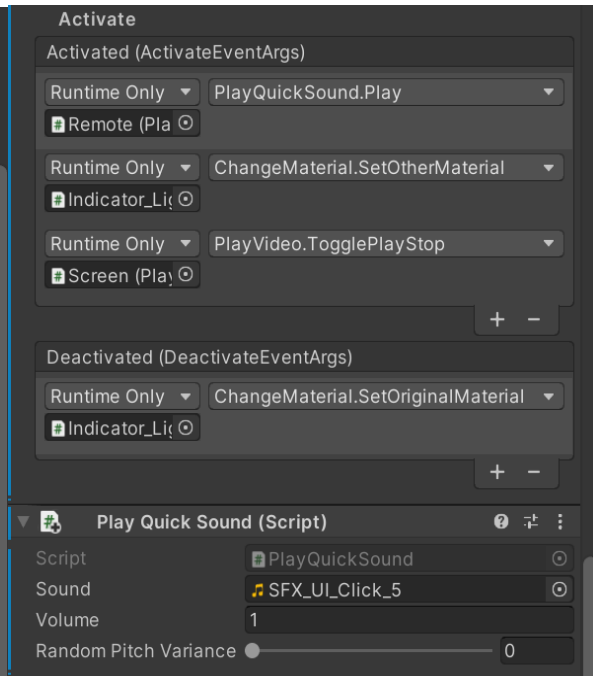


Figure 2 - Scripts on TV remote



Figure 3 TV, Phone and Flashlight working

Now that I had a Video object I wanted to add an Audio object to my scene. From my assets I added a record player to my scene with an Audio Source component, Play Continuous Sound script component and an XR Simple Interactor component. In the audio source I set the spatial blend to 1. I put a song from my assets folder into the play continuous sound script and unchecked the play on start option. I put a Rotate Object script inside the platter child object of the record player and in the Select events of the simple interactor made a PlayContinuousSound.TogglePlay and RotateObject.ToggleRotate so when the user selects the record player, the song will start to play, and the platter will rotate the record.

I downloaded a painting assets folder from the assets store and put 2 paintings on the walls of my room. I downloaded 2 images I wanted for my room and placed them in the textures folder within this asset folder. I then made 2 new materials, one for each painting and set the image texture as the Base Map for the material. I could now drag and drop this material onto my painting in the scene and change its image. At first I had 1 painting and 1 mirror in my scene, but the mirror object was causing the batches to be too high so I changed it to another painting.

Before creating a UI Canvas I wanted to have an interactable particle system in my room. I downloaded another asset with coffee mugs and bowls. I put a mug on a table and made an attach point child object and a particle system child object. I put an XR Grab Interactable component on the mug and also an On Tilt script. I could then set ToggleParticle.Play for the particle system in the On Begin box of the script and ToggleParticle.Stop for the particle system in the On End box of the script. This allows a user to pick up the mug and if they tilt it, a particle system will pour out of the top and will stop when the user stops tilting it.

Next, I wanted to make a UI canvas. This consisted of a Welcome Background canvas with some text objects and a button and a Reset Trigger canvas with a button. In the welcome button there is an On Click script that toggles the welcome canvas and the reset canvas, so when clicked the welcome background disappears and the reset canvas appears. I made another Reset Background canvas and made the Reset Trigger button toggle the trigger canvas and the background canvas. Now a user can click the Welcome button and load the Reset Trigger button, and when they click that it will open a Reset Background canvas. Within this there are 2 more buttons, Yes and Cancel. The cancel button closes the Reset Background and sets the Reset Trigger to be active again. The yes button has a LoadScene.ReloadCurrentScene script that will reload the scene. Finally I added another Settings Background with 2 text objects, each of which with a child object containing a toggle script component. One toggle component was used to enable or disable the ActionBasedSnapTurnProvider and the other toggle was used to enable or disable the Raycast for each Hand Controller. This UI Canvas was tricky to develop because there are a lot of buttons interacting with different objects within the canvas, but by gradually testing each new addition it made it easier to develop.

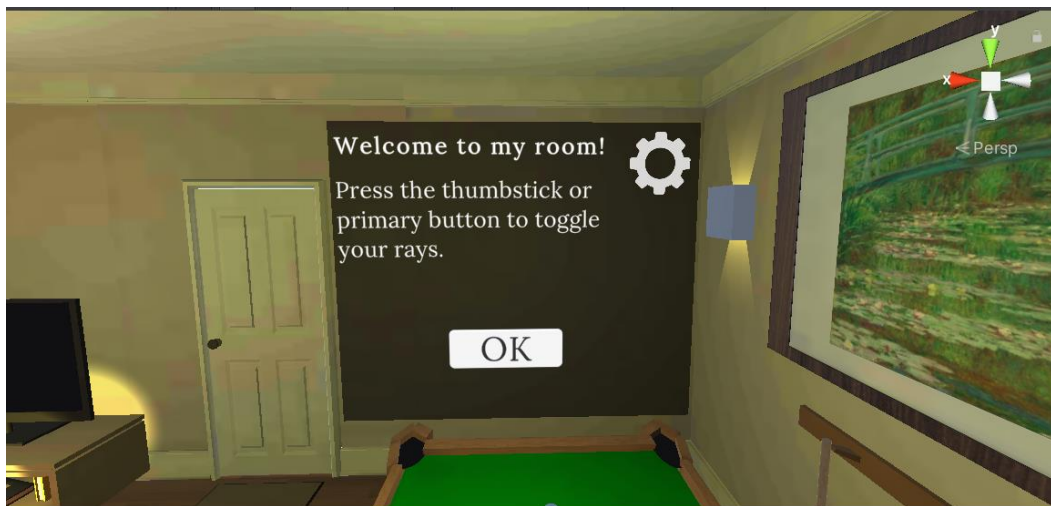


Figure 4 - Welcome Background



Figure 5 - Reset Background

The last thing I did to my scene when developing it was adding a Light Probe Group to my room object, changing my light objects to Baked Mode and then generate a lightmap for my scene.



Figure 6 - Final view of my room

Testing

Coding:

When creating the scripts for my pool table and pool balls, the collision detectors I had made for the pockets were being triggered when the scene loaded, even when none of the balls were colliding with them. This meant that when the scene loaded, all the balls would immediately disappear. I spent a long time trying to fix it to no avail, so I moved on to some of the other features in my room so I didn't fall behind. When I came back to try fix this issue, without changing anything, 5 of the 6 colliders worked and the last 1 caused the same issue of making the balls disappear when the scene loaded. I disabled the collider on that pocket and decided to work on something else again. I came back another time to fix this last pocket and when I enabled the collider again, the issue had gone away but now none of balls were disappearing when they collided with the pocket. My fix for this was to just add a bottom part of the pocket that the balls could sit in. The pockets are big enough to fit multiple pool balls and you cant see them due to the black circles I placed at the mouth of each pocket. I then tried adding a script to the colliders for the black ball, so that when it's potted, the scene would reset. Like the first issue with the colliders, it was being triggered when the scene loaded, regardless of if the black ball had touched the colliders or not. This resulted in an infinite loop of the scene reloading so I removed the script. I thought a suitable fix for this would be for the user to use the scene reset button beside the pool table if the black ball was potted.

I also tried to write a script so the cueballs position would be moved back to its starting point if it was potted, but this caused a problem with one of my other scripts when I tried adding it to my files so I deleted it. Unfortunately I didn't have enough time to resolve this problem, but I know with enough time I would have added a SpawnObject script, so that if the cue ball collided with the pocket, rather than resetting its position, a new cue ball would be made on the table. I think this would have worked although I suspect I would run into my original issue of the script being triggered on scene load and resulting in an endless amount of cue balls being spawned. As the cue ball is an interactable object, if a user pots it, they can take it out of the pocket and place it back on the table instead.

Another problem I had was with my Left and Right controllers and rays. My laptops graphics card wasn't strong enough to run the scene effectively on the Oculus, so I had to use the XR Device Simulator instead. I had tried moving my project to a PC, but after hours of moving files and downloading packages, the hand controllers or camera weren't responding to my headset, although I knew they worked when tested with the device simulator. This meant having to build and run my application on the headset, which still took some time to do with my laptop. There was an error with my controllers and rays, where the right hand would act as a direct interactor, and the ray interactor could be toggled with the thumbstick as intended, but the left hands rays wouldn't toggle. Unfortunately, I couldn't resolve this problem, but given that the right hand works as intended, and the left hand was only meant to be used for teleporting I felt as though it didn't inhibit the user too much.

User Testing:

I had gotten 2 participants to test my scene for me, my brother and a friend outside of college. The interactive objects in my scene ran smoothly, although they suggested I could have added an option to play multiple videos or have a selection of songs to listen to.

The other main piece of feedback was that the left hand wasn't being used for anything, I went back through the unity tutorial we had followed in class and couldn't see where the problem was coming from, but using the right hand to interact with objects and teleport wasn't an issue for them when testing.

I wanted to add more videos and audio to play on the TV and record player but I didn't have enough time in the end after trying to fix my other errors, but I was glad I got one of each and showed that the functionality worked.

Reflection

Overall, I enjoyed making this project. I had no previous experience with VR or Unity but had always wanted to learn and make a VR application. I had some difficult issues when creating my pool table and particularly when trying to move my project from my laptop to a pc.

I wanted the project to be on my laptop so that I could work on it at home, as I felt I wouldn't have gotten enough done had I only been able to work on it during class hours, but it proved difficult to test at the later stage of development. I have since gotten an external hard drive, and in the future I would begin with my project on that, so that I could load it on my laptop at home and test it in class time. Unfortunately that wasn't available to me at the beginning of development.

I also would have liked more time to add other interactable objects to my scene. I spent a lot of time trying to fix the pool table interactions and meant that I could not implement as much as I would have liked.

I really liked using Unity too. The ability to make or import objects and add components and scripts to them was a lot easier than having to code everything from scratch, and was very accessible and fun to learn and progress.

References

Moody, M. (2021, September 11). *Oculus quest 2 specifications*. Studio X. Retrieved December 2, 2022, from <https://studiox.lib.rochester.edu/oculus-quest-2-specifications/>

Published by David LayzelleWriter - Tech Guru. (2022, January 4). *Top VR headsets of 2022*. Unity Developers. Retrieved December 2, 2022, from <https://unitydevelopers.co.uk/top-vr-headsets-of-2022/>

XR Interaction Toolkit: XR interaction toolkit: 2.2.0. XR Interaction Toolkit | 2.2.0. (n.d.). Retrieved December 2, 2022, from <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.2/manual/index.html>

<https://learn.unity.com/tutorial/0-1-set-up-unity-and-your-vr-device-1-1?uv=2020.3&courseId=60183276edbc2a2e6c4c7dae#>