

Coding Tasks

Subhaan Elburz

May 27, 2025

Task 1: Develop a simple VR video player that play videos streamed by a server

1 Initial Thoughts

When I first read the coding task, I was stumped. Instead of sitting around, though, I emailed Raoa with some questions I had about the task and got started. She told me that:

- The lab uses a Meta Quest 3S headset
- The recommended Unity Version that I should use is Unity 2022.3 LTS

2 Environment Setup

To begin, I set up my development environment since I've never worked with VR before now. With the information Raoa provided, I started researching how to set up Unity for development on a Meta Quest 3S headset. While researching, I found that Meta provides step by step instructions on how to set up the exact environment I needed [5]. Following the Meta documentation, I installed Unity 2022.3 LTS with Android Build Support and the necessary XR plugins.

3 Learning Unity

So, I have worked with Unity a little bit since I took a game development class in high school (not VR just regular Unity), so I needed to familiarize myself on Unity for VR. So, I turned to youtube and starting watching videos. Most notably, I watched many videos from Valem Tutorials, who also answered a big question I had at the start: How would I even test anything in VR without a headset? Which he answered in his "How to Make a VR Game WITHOUT a VR Headset" Video [12]. Additionally, I also watched:

- Fastest Way to UI with Unity Meta XR SDK [15]
- How to Make a Draggable Progress Bar for Videos in Unity [8]
- UNITY: Video Player in VR [14]
- How To Make A 2D Video Player in VR 2023 [9]

4 Creating a VR Video Player (Non-Server)

After watching a few videos, I decided to start by creating a regular VR Video Player without any server streaming implementations.

First, I had to create the UI for the video player, which wasn't that difficult because in the Meta XR Interaction SDK, the example scene "ComprehensiveRigExample" had a video player UI that I could use as a base [6]. I removed some elements, and changed the colors, and had my UI ready.

I then had to create the C# script for the video player so that the buttons/toggles on the video player worked. So, I turned to YouTube again and watched some videos on how to create working buttons on video players:

- Playing Video In Unity Playlist [13]
- Unity3D 5.6 Video Player Load, Scrub, Loop, Play & More [1]

Additionally, the Unity Documentation was also a resource I used quite a bit [2]. After a lot of trial and error, I finally managed to get the video player to work with a volume slider, +10s/-10s buttons, play/pause/replay button, and also a time slider.

5 Server and MPEG-DASH Research

- Found out that setting up a server is pretty straightforward with Python on StackOverflow [11]
- So, after finding out about Python servers, I started one up and tested to see if it would work with the default Unity video player component, and it did! (regular mp4 file on server)
- In addition to searching online about MPEG-DASH, I watched a quick video on it as well [7].
- In the video, he goes over ffmpeg, so I also turned the 3 videos into a singular manifest file using the command:

```
ffmpeg \
-i 144p.mp4 \
-i 360p.mp4 \
-i 720p.mp4 \
-map 0 -b:v:0 250k \
-map 1 -b:v:1 800k \
-map 2 -b:v:2 2500k \
-use_template 1 -use_timeline 1 \
-seg_duration 2 \
-f dash output_video.mpd
```

- I also looked into MPEG-DASH in Unity to see if there were already implementations, and I did find AVProVideo. So, I decided to use it to play the manifest file on my local server with the following command:

```
python3 -m http.server 8000
```

And it did work! However, there was a watermark, so I decided not to use it.

- After all this research, I decided to try and create my own DASH video client.

6 Creating a VR Video Player (Server)

- I first created a script that would handle everything, called *DASHManager.cs*
- How it works is that it will first download the .mpd file that's hosted on the local Python HTTP server using UnityWebRequest [3]
- It then loads it as an XML document and grabs the relevant Representation elements (Video Qualities) from the file [10]
- After, it sorts each video quality into a list of their respective bandwidth and URL on the server
- The way it determines which video quality to pick is as follows:
 - Uses a stopwatch to time the download of the current MP4 video file
 - Calculates throughput in kilobits per second: $\text{kbps} = (\text{bytesDownloaded} \times 8 / 1024) / \text{seconds}$.

- Checks if I've forced a quality via UI; if so, it picks that index.
- Otherwise multiplies measured kbps by the safetyFactor and picks the highest video quality whose bitrate (in kbps) is below that threshold.
- Also logs time (s), quality, bitrate (bps), throughput (kbps), delay (ms) into Assets/log.txt
- If your internet speed slows (lower safety factor), it will pause the video, change the quality, and then resume the video from where it left off
- After completing the script, I connected it to the Video Player I made earlier, which allowed me to finalize my VR Video Player that streams from a Server NOTE: It is not exactly MPEG DASH, but I do use a manifest file and determine which quality to pick based on throughput

Task 2: Build and run dragonfly successfully

1 Initial Thoughts

The second task seemed a lot more straightforward than the first task, so I wasn't too worried. However, this was definitely not the case since I was only able to build Dragonfly, and I never got it to run successfully [4].

2 Work I Completed

- The furthest I got was building Dragonfly, but not running it, so this is how I built it:
 - I first attempted to just paste in the commands on the Dragonfly README.md file in my current system which was WSL Ubuntu 22.04 but that did not work...
 - I then realized that the commands on the README.md were for Ubuntu 16.04, and 18.04 machines, so I quickly downloaded WSL 18.04.
 - Once I had the correct Ubuntu version, it was pretty straightforward building it, as I just copy-pasted the pre-requisite command on the README and ran it as sudo.
 - Once I had the pre-requisites installed, I cloned the repo on my machine and installed both fmt and folly with the given commands as sudo
 - Lastly, I built Dragonfly with the build command that was given, however, this was the extent of my progress as I never figured out how to actually run the server/client in time before the submission
- I did attempt to run the server and client and here were some of the mistakes I did:
 - Followed the commands in the Docker Image README, which led me to no-where... I genuinely have no idea if I even needed to do that, I think I just built Dragonfly twice somehow
 - Followed the commands in the Video Preparation README, which also led me no-where, as running


```
python3 run.py video
```

 took over an hour and I did not know where to proceed after that
 - Attempted to just execute the server and client like this: ./server and ./client , however I could not figure out what parameters I needed for it to work in time

References

- [1] Renaissance Coders. Unity3D 5.6 Video Player Load, Scrub, Loop, Play More. May 20, 2017. <https://youtu.be/ffJoBfj41GU?si=qGU3veBun51htq-B>.
- [2] Unity Documentation. Scripting API. May 24, 2025. <https://docs.unity3d.com/6000.1/Documentation/ScriptReference/index.html>.
- [3] Scott Durkin. Unity3D API Request - Get display data from the internet. Jul 28, 2022. <https://youtu.be/cU4UaqrPb84?si=lfy5jxK1Z0Mm7ZE1>.
- [4] Ehab Ghabashneh, Chandan Bothra, Ramesh Govindan, Antonio Ortega, and Sanjay Rao. Drag-onfly: Higher perceptual quality for continuous 360° video playback. In *Proceedings of the ACM Special Interest Group on Data Communication, SIGCOMM '23*, 2023.
- [5] Meta Horizon. Set up Unity for XR development. Apr 8, 2025. <https://developers.meta.com/horizon/documentation/unity/unity-project-setup>.
- [6] Meta Horizon. ComprehensiveRigExample Scene. Aug 7, 2024. <https://developers.meta.com/horizon/documentation/unity/unity-isdk-comprehensive-rig-example-scene/>.
- [7] Tech With M. MPEG-DASH Streaming Explained for Beginners. Aug 19, 2021. https://www.youtube.com/watch?v=vq_1aQtY354.
- [8] Max O'Didily. How to Make a Draggable Progress Bar for Videos in Unity. Aug 29, 2023. <https://www.youtube.com/watch?v=OuNhyibFdqU>.
- [9] Fist Full of Shrimp. How To Make A 2D Video Player in VR 2023. Dec 12, 2023. <https://www.youtube.com/watch?v=u5vI4oRonoM>.
- [10] Brian (Able Opus). C - Parse an XML file . Jul 13, 2013. https://youtu.be/ulzFqXb13dI?si=Kyfm6WKtCAZtJR_3.
- [11] Stack Overflow. How do I setup a local HTTP server using Python. Jan 16, 2015. <https://stackoverflow.com/questions/27977972/how-do-i-setup-a-local-http-server-using-python>.
- [12] Valem Tutorials. How to Make a VR Game WITHOUT a VR Headset. Dec 8, 2024. <https://www.youtube.com/watch?v=vrP91DqS1Bs>.
- [13] Unity. Playing Video In Unity. Jul 14, 2017. <https://youtube.com/playlist?list=PLX2vGYjWbI0TTsE01qf1dWNqwMHeNEbUo&si=QZtB0-0icpfzP140>.
- [14] New Media Workshop. UNITY: Video Player in VR. Mar 8, 2023. <https://www.youtube.com/watch?v=dFR1gh4UspA>.
- [15] Let's Do XR. Fastest Way to UI with Unity Meta XR SDK. May 17, 2025. <https://www.youtube.com/watch?v=vIw-dz1oL00>.