

VR Hand Interaction Multiplayer Room

Leonid Shleifer
Yotam Portal

Supervisors:
Yaron Honen
Boaz Sterenfeld



Contents:

1. Introduction
2. Equipment
3. Technologies
4. Application Description:
 - User hands
 - Grab
 - Scale
 - Two hand grab
 - Request ownership
 - Other user control
5. Closing Thoughts



Introduction:

In this project we built a Virtual Reality application that enables two users to enter the same VR shared space and a mutual physical room and then together observe an object, move it around and scale it.

In the application the users can grab the objects and move them in all axes, spin and tilt the objects in every direction, enlarge and diminish the objects, all that while the other user will see in real time the object's motion and also the first user's hands representations as two spheres.

The application is built as a VR game in Unity and runs on (Oculus) Meta Quest 2 VR headset, the application is not cross platform since it requires the Meta Quest 2 hand tracking capabilities, more on that later.

A good use for this type of application can be a design team working on a certain product, entering the application and being able to see the design in 3D, and discuss it as if it was in front of them in real life.

Equipment:

- 2 Headsets of:

Oculus Quest 2 (rebranded as Meta Quest 2 in November 2021) is a virtual reality (VR) headset developed by Reality Labs (formerly Oculus).

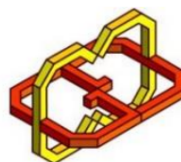
The Quest 2 is capable of running as both a standalone headset with an internal, Android-based operating system, and with Oculus-compatible VR software running on a desktop computer when connected over USB or Wi-Fi.





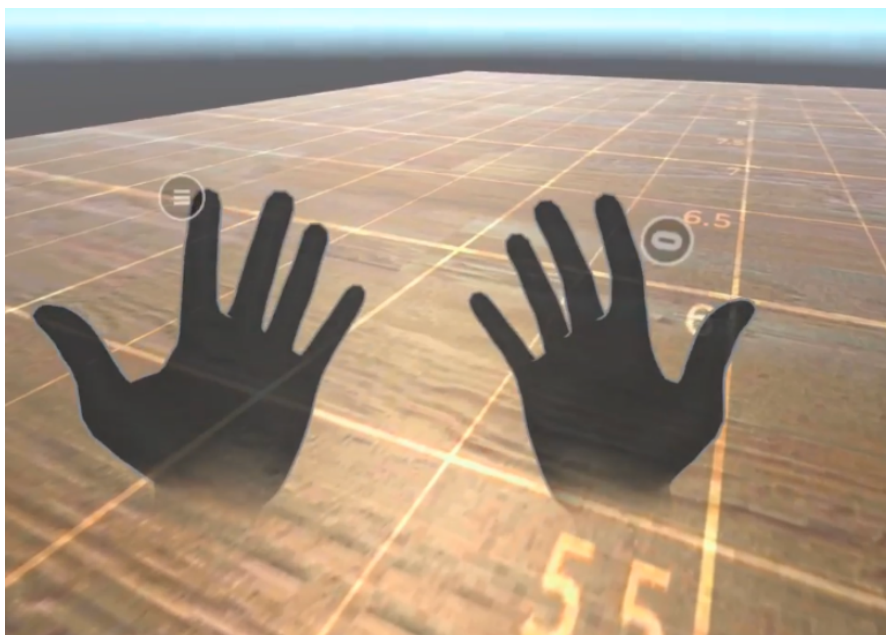
Technologies Used:

- Unity - Unity is a cross-platform game engine developed by Unity Technologies, It is particularly popular for iOS and Android mobile game development and is considered easy to use for beginner developers. The engine can be used to create three-dimensional (3D) and two-dimensional (2D) games, as well as interactive simulations and other experiences.[7][8] The engine has been adopted by industries outside video gaming, such as film, automotive, architecture, engineering, construction, and the United States Armed Forces.
- Oculus XR Plugin - The Oculus XR Plugin enables you to build applications for a variety of Oculus devices including the Rift, Rift S, Quest, and Quest 2.
The input subsystem provides controller support, haptics, and tracking for the controllers and HMD.
- Oculus Integration - The Oculus Integration SDK for Unity provides support to develop Oculus apps in Unity.
The Oculus Integration brings advanced rendering, social, platform, audio, and Avatars development support for Oculus VR devices and some Open VR supported devices.
- Photon PUN 2- Photon Unity Networking (PUN) is a Unity package for multiplayer games. Flexible matchmaking gets the players into rooms where objects can be synced over the network. RPCs, Custom Properties or "low level" Photon events are just some of the features. The fast and (optionally) reliable communication is done through dedicated Photon server(s), so clients don't need to connect one to one.



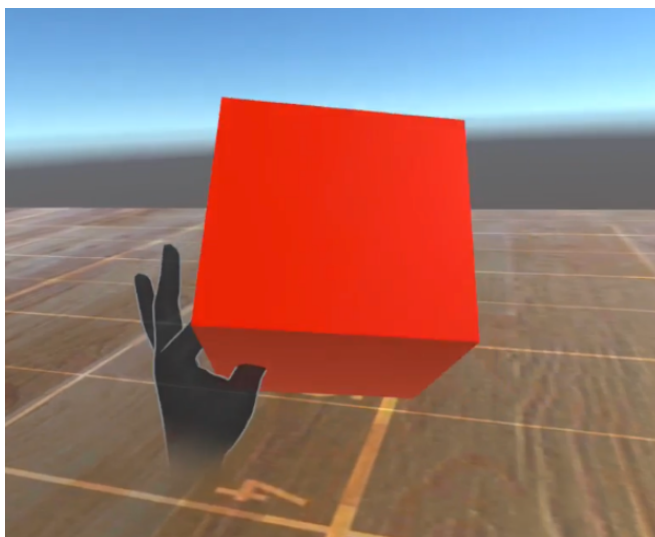
Application Description:

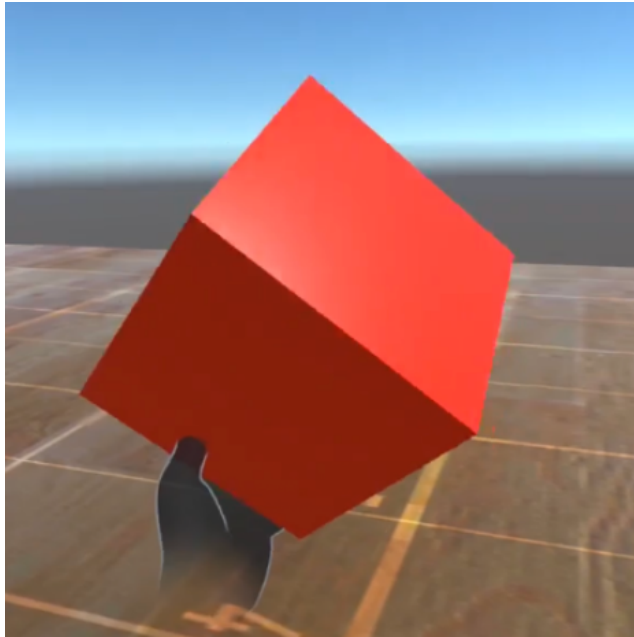
User hands are seen on the app in their exact positioning when in the headset's field of view



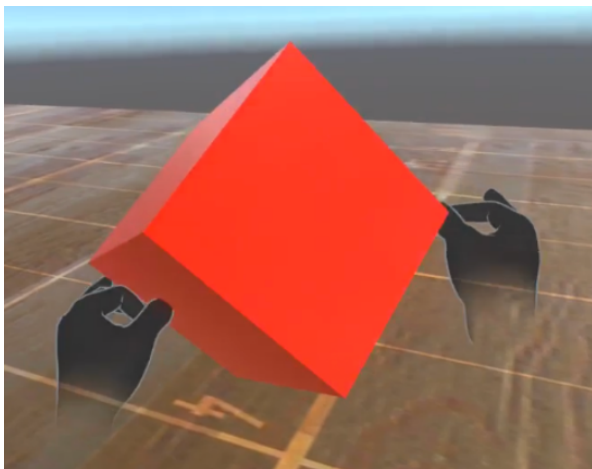
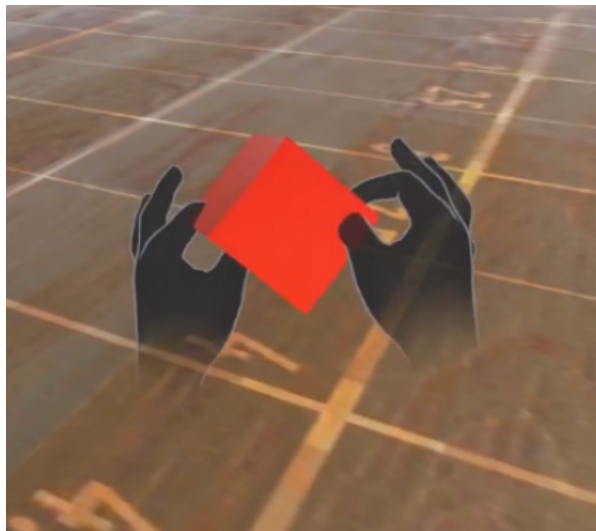
Other user hands are seen as spheres, a sphere for each hand.

The user is able to perform *grab* interaction with a pinch or closing a fist, enabling it to be moved to every direction and rotated on every axis.



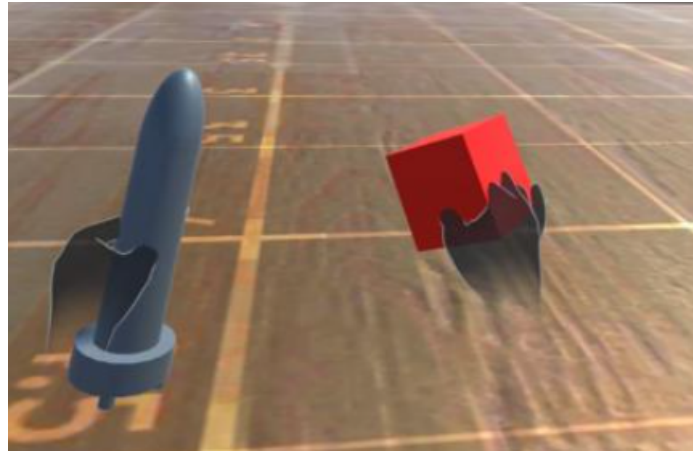


The user is able to *scale* an object using both hands, with an intuitive two point grab and pull.



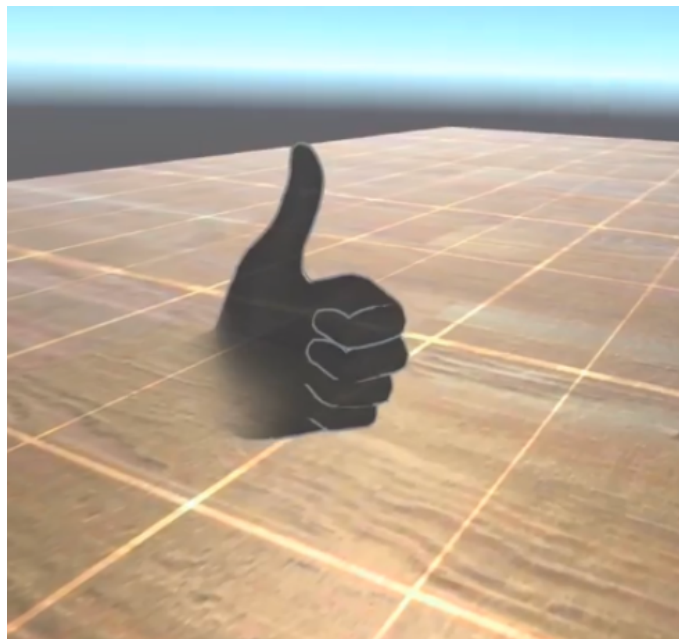


The user is able to grab more than one object simultaneously.



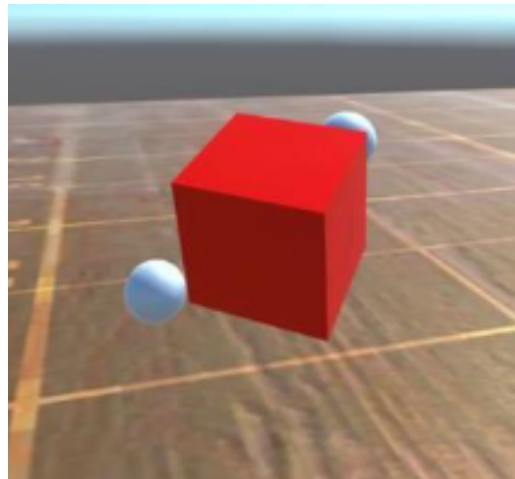
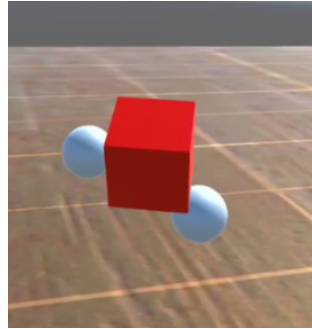
Multiplayer applications can have difficulty in determining which user is the “owner” of a certain object, for us that means who can move and scale the objects.

We decided that for this purpose we would like to use another nice capability - Gesture Detection. In order for a user A to claim ownership over the objects in the game, it should put up a “Thumbs up” gesture with it’s right or left hand, the application will detect the gesture, identify it and will transfer the ownership of the objects to him, so that he will be able to move and scale them.





The Other user is also able to perform all of the above listed capabilities. As mentioned earlier, his hands will be displayed as a pair of spheres, each for one hand. This is how the other user scaling the object looks like (The size of the spheres is constant).



Closing thoughts:

The VR/AR topic is an ever growing field with big financial potential in it, and a lot of the big tech companies are looking to take their share in it. That means that it is very quickly developing and changing, in our own experiences that meant dealing with libraries and capabilities that are still very new and not always offering full support for advanced features like hand tracking. On top of that, new libraries can mean less support to be found online.

That being said, we are glad and thankful for the opportunity to have our first steps in this fascinating field being done in this project.