# High Acuity Haptics

Peyton Rapo, Alexander Fernandez

## Problem

One of the biggest issues facing how immersive the VR experience feels is the lack of physical feedback you get from interacting with objects. When you touch something in VR you may get a vibration from your controller, but it doesn't give you an impression of the shape of the object. We want to be able to give users the ability to better feel objects in VR and thus make the experience more immersive.

## Related Work

Haptic feedback can be achieved through vibrations, electrical stimulation, or mechanical methods. Vibrations use varying levels of vibration to achieve the sensation of touch. Electrical stimulation involves using a special material which conducts electricity and can stimulate your finger when touching objects. Mechanical methods are physically pushing something against your fingertip.

## Your Approach

We chose to use a mechanical method, specifically in our design air bubbles press against the finger pad. We found our approach, which we arrived at independently, is like the approach HaptX uses except they use 12 air bubbles and we used 16. Since the bubbles are spread out along the finger pad and each bubbles is programmable, you can capture the feeling of touching objects.

## Acknowledgements

Big thanks to the following people for helping brainstorm our ideas and getting us the parts to be able to execute this project: Douglas Lanman, John Akers, and Nicholas Colonnese

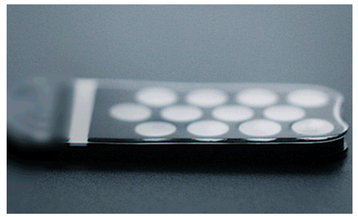
## Method

Our system pipeline is as follows:
Hardware Side:

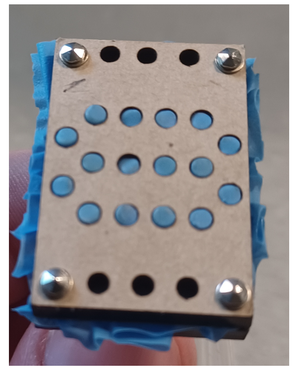
- We have 3 air pumps supplying air pressure to the system
- We then have tubing which connects the air pumps to the valves
- We then have a series of 16 valves which are controlled by transistors which are connected to our Arudino and can be turned on and off whenever
- The valves limit air to our finger pad which is two pieces of acrylic laser cut with holes and clamping a piece of latex between them

Software Side:

- We setup a unity scene with objects that we set to be interactable with Oculus's hand tracking.
- The right index finger pad then will be constantly shooting our rays and detecting if those rays collide with any objects and if they do, they will send a signal to our Node Js server, which is sending signals to our Arudino.
- When our Ardunio receives signals from the Node Js server it will open and close valves depending on what we want to render, which allows air to flow to our finger pad.



Haptx finger pad



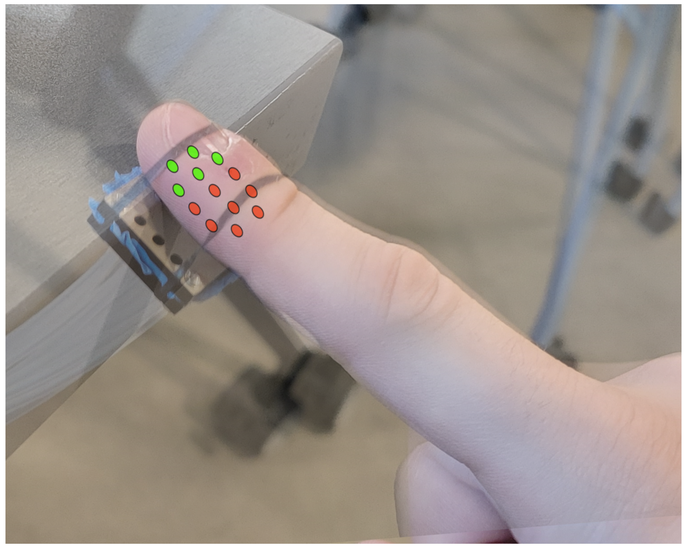
Our finger pad design

## Results

Overall, we were able to achieve our goals in being able to render specific finger pad configurations in order to achieve the sensation of touching objects. While our latency was decent, we ran into issues with regards to the hand tracking and object collision detection. The hand tracking component we couldn't really change as it was done by the Oculus itself, and the object collision detection had issues due to the meshes of the objects themselves.

We also did manage to implement pressure detection using a pressure sensor, which we used to regulate how much pressure was in each of our tubes. Unfortunately, our pressure sensor broke so we had to scrap this idea.

Another idea we scrapped was the idea of a duty cycle for each tube based on the distance to objects to try to increase the range of feelings instead of it just being on and off on the finger pad, but this ended up not feeling great.



Visual rendering of our system in action