

# CUBED AN INTERACTIVE 3D DRAWING EXPERIENCE

JESSICA YIN | INTRODUCTION TO PHYSICAL COMPUTING | FALL 2017

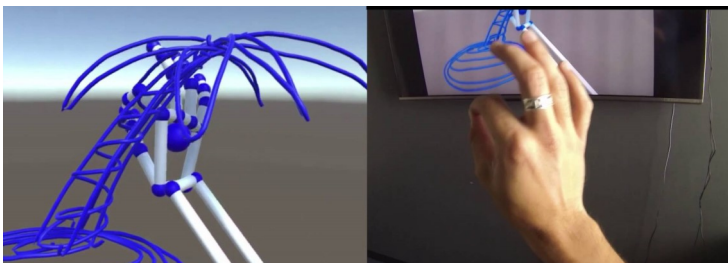
## ORIGINS

Two technologies that I've always wanted to explore have been the Unity Game Engine and Leap Motion Sensor, so I took this final project as an opportunity to do that.

I was very inspired by Ralf Breninek's project (pictures below), which was a simple game that used the rotating cube interface to represent portals to different worlds. His project used Arduino, Unity, and Leap Motion to demonstrate his interest in 3D projection mapping.



I was also impressed with Leap Motion's utility "Pinch Draw", which was made for integration with a VR headset. As someone who recently started to rely heavily on SolidWorks for visualizing larger projects, I loved the idea of being able to draw in 3D.



## PROCESS

I started off by breaking down the project into different sections:

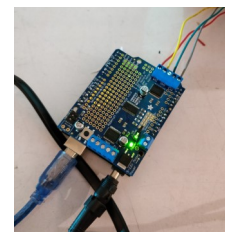
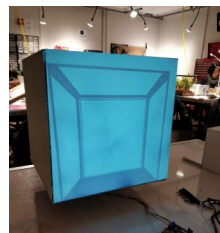
- Arduino/Adafruit Motor Shield + stepper motor
- Leap Motion + Unity
- Unity/Uduino + Arduino
- Projector + Cube

Since I know that I tend to struggle with programming and I'm not familiar with C# (for Unity), I started with working on the components that required programming. Thankfully, I was able to adapt the Pinch Draw program from Leap Motion for use without a VR headset pretty easily by configuring some options.

## PROCESS CONT.

Adafruit's motor shield streamlined the process of powering a 12V stepper motor while having it connected to an Arduino— powering components is still something I am not confident with, so I was glad to have it.

The part that I definitely struggled with the most was putting it all together at the end. I spent many hours debugging my code in both Arduino and Unity while trying to make Leap Motion control the Arduino. I had issues with figuring out how to rotate the drawing with the cube, as well as synchronizing both the virtual and physical cube rotation. I used Uduino, an asset from the Unity Asset Store, to greatly simplify the process in controlling the Arduino through Unity, but I still had to spend hours debugging due to my lack of experience with C-based languages/Unity and restricted access to the Serial port (for print debugging). I honestly did not think that I would finish in time, but miraculously everything started working (for the most part) in the last three hours before it was due.



## FURTHER WORK

While I am happy with what I have accomplished so far with this project, there is still a lot of room for improvement. One of the most pressing things I would like to fix is syncing the rotation of the virtual and physical cube. It honestly doesn't work too well, and that's a major part of the user experience that I would like to improve. I would also like to improve the form factor— I originally had designed a stand to hide all the wires going everywhere, but didn't have the time or materials to build it. Some other features that I wanted to include were the ability to change the color of the line drawing by pressing buttons, as well as a reset/new canvas button, as well as the ability to save drawings.