Development and Testing of an Input Device for Capturing Hand and Finger Movements in a Virtual Reality Environment

Hunter Carter, Leslie Jenkins, Matthew Tilton
Louisiana Tech University

Introduction

Virtual reality can be defined as an immersion of virtual space using technology that lets one see the virtual space around them and, in most cases, lets one interact with the virtual environment. This technology is still young and there are many unanswered questions. Many problems are still in the process of being solved, such as motion sickness from moving around the environment without physically moving and being able to use hands to the full extent.

The overall goal of the project described here is to test the idea of a controller wrapping a person's body, not allowing them to move naturally, and taking input via pressure to determine how the user wants to move in the virtual world. The original idea for this project came from the observation that most virtual reality controller research revolves around a player using their full body in the virtual world in a limited space. While that approach is promising, we felt it might be beneficial to look at the control issue from a different angle.

The idea presented here focuses on finding a solution for in a unique way, but largely focuses on the controller aspect. The idea is that, although a user may not be able to fully move their body naturally, they will be tricked into feeling like they are naturally moving from applying force naturally and seeing that force be applied to objects in the virtual world. A prototype for a pressure sensitive controller was created to test this idea and real-world testing was carried out on volunteers

Goals

When starting this project, there were many ideas and visions for how the controller was going to look as well as function. The first goal was to have the user rest their hand in a natural position, slightly cupped. The second goal was to have a user wearing an HMD to visibly see their hand moving from input in the controller, including wrist movement. To test the ease of use for the controller, a goal for playing Rock Paper Scissors was created. Another test goal was to be able to pick up an object in virtual reality.

Implementation

Force-sensitive resistors were used to measure how much pressure a person was applying to a finger. The output was taken from the sensors, converted into the corresponding finger input, and then applied to the hand model and appropriate finger animation. The virtual reality environment was made possible through Unreal Engine

4.20. The combination of visual feedback from virtual reality and the physical act of trying to move your hand, the controller could make the user feel as if they were moving their hand normally.

To fabricate the controller, the following items were used:

- Three acrylic sheets
- 4 sheets of foam cut to size
- Teensy 3.6 as the microcontroller
- Breadboard
- 10,000 ohm resistors
- 12 force-sensitive resistors
- 22-gauge wire soldered to the force-sensitive resistors
- Quarter inch bolts that were six inches long
- 12 nuts to fit bots

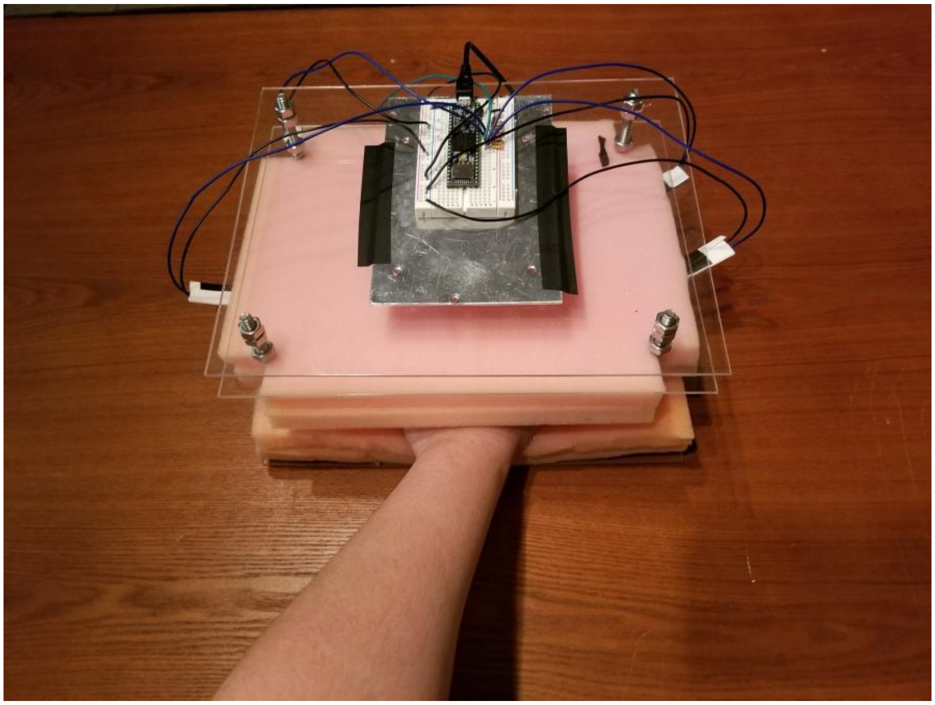

Figure 1. Final product of the controller.

Holes were drilled one inch from the corners of each of the sheets of acrylic. Next, the base acrylic sheets were fitted to the bolts and two layers of foam were applied on top. The sensors were then placed on the foam and held down by twist ties. They were attached in a position so that one's fingertips would rest in the center of each of them for a left hand. The next two sheets of foam had the corners removed so they could easily be removed when necessary. The next layer of acrylic was then added, this layer was adjustable but allowed a person’s hand to be held in place by tightening bolts. Another acrylic sheet made the topmost layer of the device and held the breadboard and the teensy. Finally, the wiring for the breadboard was completed to attach all sensors to the microcontroller.

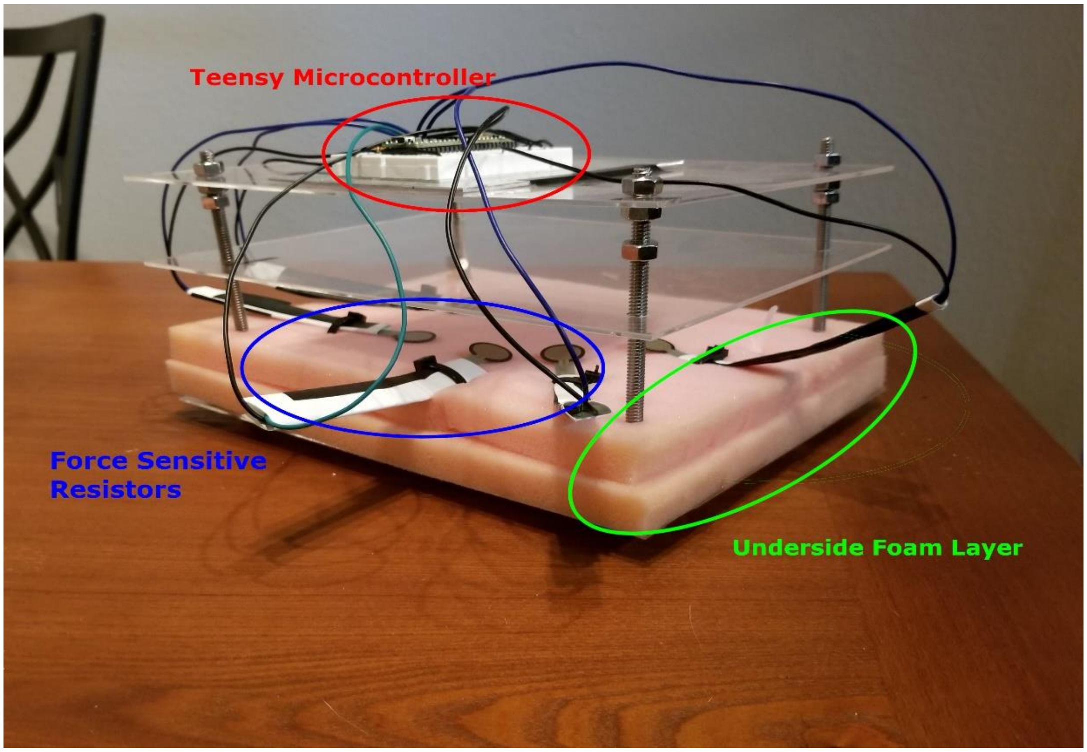

Figure 2. Structure of the controller without top foam layer. Microcontroller and breadboard on top layer of acrylic, second layer of acrylic for adjusting tightness, left out layer of foam, space for hand, FSRs, and finally the underside foam layer.

A Teensy 3.6 microcontroller was coded to send the output from the sensors to the computer. In the beginning, we had planned for the teensy to be used as a generic USB device. Sending and processing the data turned out to be easier than expected but once we started trying to integrate it with unreal engine, all progress stopped. In the end, we went with emulating a joystick and using 5 different axes (one for each finger). This greatly simplified the code on the teensy reducing it from around 200 lines with the manual USB to less than 20 when emulating the joystick.

```
void setup() {
  for (int i=0; i<9; i++) {
    pinMode(i, INPUT_PULLUP);
  }
  Joystick.useManualSend(true);
}
void loop() {
  Joystick.X(analogRead(0));
  Joystick.Y(analogRead(1));
  Joystick.Z(analogRead(2));
  Joystick.Zrotate(analogRead(3));
  Joystick.sliderRight(analogRead(4));

  Joystick.send_now();
  delayMicroseconds(500);
}
```

Figure 3. Code for the driver controlling the input actions.

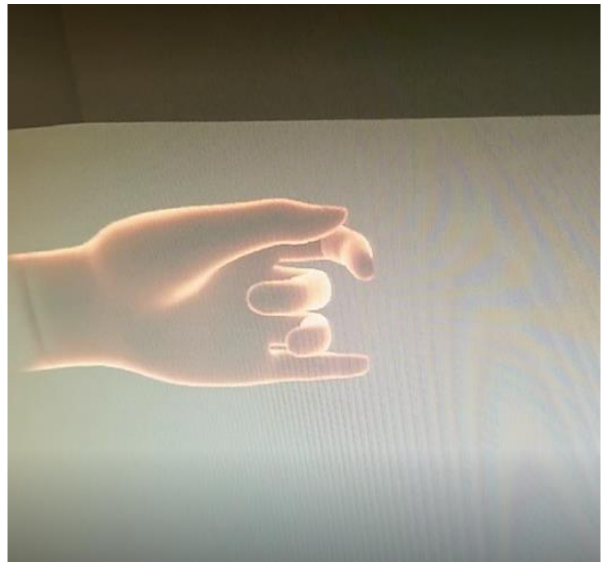

Figure 4. Handle model in Virtual Environment with closed fingers.

Although this does not allow for additional axes to be added through Unreal Engine due to the default setup for joysticks on a Teensy microcontroller, a customized joystick can be created for the Teensy if needed.

An environment was created in Unreal Engine to test the controller. One map was created with a Rock Paper Scissors Artificial Intelligence. This allowed a user to be able to see the hand gestures created when using the controller. Thanks to UVRF created by iNFINITE Production, a modified version of a natural looking hand model was able to give the animations necessary for the controller. While the hand model controlled the animations and kept track of which fingers were “open” or “closed”, code was created to check if the hands were making the appropriate signs, rock, paper, or scissors. An extra hand sign, a thumbs up, was added to let the game know the user was ready to play a round. Using Unreal Engine’s raw input plugin, we tied the controller output to Unreal Engine inputs, which allowed them to be used as normal inputs such as a mouse button being clicked or a keyboard key being pressed and released.

Figure 5. A user making a thumbs up with his left hand and controlling the entire hand's location in space with the motion controller in his right hand.

End Result

Throughout the process, many obstacles were encountered and overcome; changes to the original plan were made due to time constraints and available materials. Many improvements were also noted or given from feedback. With the acrylic sheets being hard and stiff, the user's hand position was more flat than in the natural position as intended. Initially, the sensors were planned to be on top and bottom of a finger due to the resting position of a finger not being fully opened. We also intended to have sensors take pressure from the palm of the hand to control the rotation of the hand. After using the controller with the hand in a flat position, it was found that sensors on top and bottom of each finger caused the motion when using the control to feel less natural. Placing the sensors on the underside of the fingertips worked best, and the sensors on the top of each finger were removed. It was also a goal to be able to pick up an item with the controller, but due to the timeframe, this environment was not implemented.

We tested the system on volunteers who were not involved with the creation of the controller. Their feedback included: (1) the controller could use better ventilation since it starts to get warm after continuous use; (2) the sensors have the potential to give better readings if the surface they rested on was harder. This was found from the foam allowing the sensors to bend on the edges rather than the middle when slightly misaligned fingers rested on them.

Conclusion

This project was a very valuable learning experience. Although every goal was not accomplished, a device was built that provides an immersive experience of naturally using a hand. We were able to produce an experience that consumer products have yet been able to give to the public. Given our proof of concept, we believe this approach of providing an immobilizing environment that prevents large scale movement while measuring the pressure subjects are applying in particular directions should be scaled up. If it works at larger scales (full arms, legs, complete bodies) this approach might help solve some of the issues associated with capturing human movement for input into VR without requiring large physical spaces in which people can move (as with motion tracking) or the opposite where movement is provided by controller input (which can cause extreme motion sickness in some individuals.