How to setup

List of hardware:

* Raspberry PI
* Myo Armband
* Phidget servo control board
* Phidget interface kit
* LED light
* 2 x Servo motors
* Wires

Both Phidget boards have to connect with Raspberry PI via USB, Myo has a wireless USB connector that goes in the Raspberry PI as well. The two motors will have to connect to the servo control board in channels 1 (for the motor that will get controlled by Myo) and 7 (for the motor the will get controlled via user input), this is the default set in the code it can always change. Then the motor that will get controlled with user input will have a square made out of wires or paper clips to conduct electricity (as you can see in the pictures provided in this folder), that square will have a ground wire attached to it and plugged in the ground slot of the interface kit. The other motor will have a wire that will be straight in the front of the motor (as seen in the pictures) and a wire attached to the straight line and the LED’s cathode. On the LED’s anode there will be another wire that plugs in the 0 slot of the interface kit (the default slot set in the code is 0)

To run the code, you need to go to “PyoConnect\_v2.0\PyoConnect\_v2.0\” folder in the terminal and run PyoManager.pyc. When the interface loads turn on MyoEngine script and press “Connect Myo”.

How does it work?

The person that has Myo on will have to double tap his thumb and middle finger to unlock Myo, this gesture needs to be performed a second time to start the round. Then the person has to choose a command (either “left” or “right”) and press Enter. The person that has Myo needs then to perform one gesture (either “waveIn” or “waveOut”) to send the command to the motor to move and match the movement direction of the other motor. The motors will then move according to the commands, if the guess was right then the person with Myo needs to perform the “fist” gesture to send the command for both motors to come back to the default position and then “doubleTap” to start another round. When the guess was wrong the circuit will complete and the red LED light will turn on and the person at the keyboard needs to input the “lost” command and the game ends and prints the score to the console. Who has the highest score wins.

To create the game, I first needed to install an SDK (Fernando Cosentino. (n.d.). *PyoConnect.* Available: http://www.fernandocosentino.net/pyoconnect/. Last accessed 26th Oct 2018.) that made possible for Myo armband to send signals to the Raspberry PI and provided a python API that I used to process the gestures. Then I needed to install Phidgets library for linux (PhidgetsInc. (n.d.). *Phidgets linux library.* Available: https: //www.phidgets.com/ docs/OS\_-\_Linux. Last accessed 14th Dec 2018.) and Phidgets python library (PhidgetsInc. (n.d.). *Phidgets python linux library.* Available: https: //www. Phidgets.com/docs/Language\_-\_Python\_Linux\_Terminal. Last accessed 14th Dec 2018.) that provided me the API to communicate between Raspberry PI and the Phidgets. You can find the code I wrote to manage this project here: PyoConnect\_v2.0 \ PyoConnect\_v2.0\scripts\MyoEngine.py.