A Project involving controlling a game with the Gestures performed on a MYO Armband

Gesture Based User Interfaces

Gesture Based Project

Student Name: Cathal Donohoe
Student Number: G00344919
Lecturer: Damien Costello

Link to Video: https://youtu.be/BtpdREksLD0

Link to Project: https://github.com/CathalDonohoe/MYO-Game

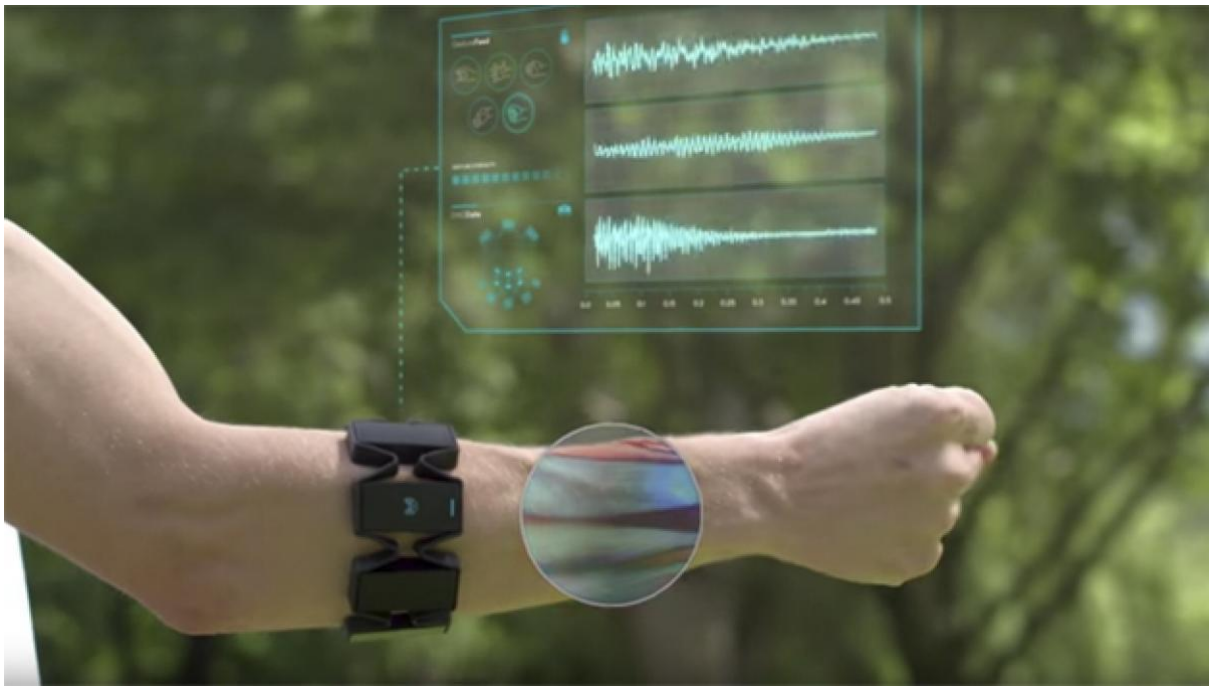
Gesture Based User Interfaces

Introduction

This project involves developing an application with a Natural User Interface. We have been given the opportunity to combine any number of technologies that we have worked with in the past four years. We can use real-world hardware to prove a concept. We are allowed to reproduce a classic game or system using a gesture-based interface. The application must be an original work, as we will not receive any credit for previous applications. The Technology I decided on using in the end was the MYO Armband. I decided on this as my lecturer had some available, I didn't own any other forms of gesture interaction, and I had already made a voice-controlled game and wanted to expand upon my skills. I always found Gesture control to be an interesting concept and I seen the MYO Armband as a good opportunity to work on it.

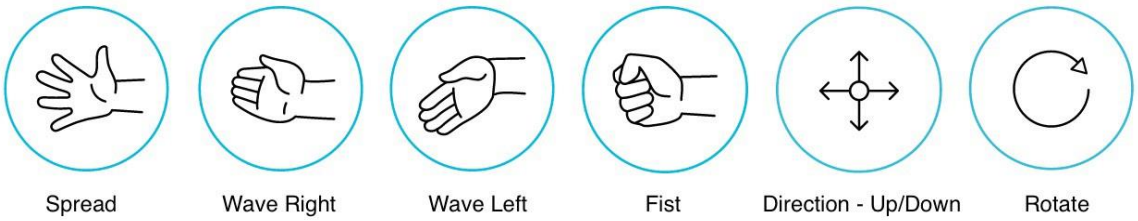
The MYO Armband

The MYO armband is a gesture controller that uses Medical grade Stainless Steel EMG sensors. These sensors detect the electrical pulses that the muscles in your arm produce when you move. The armband can be used in a variety of ways, from being used with existing applications, such as a mouse and keyboard for a PC or to be programmed into completely new games, which is what I did with the following.



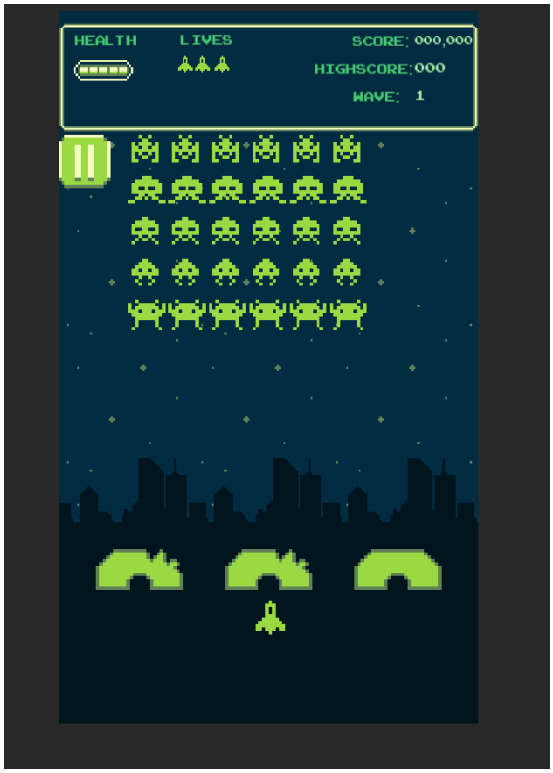
Purpose of the application - design of the application including the screens of the user interface and how it works.

The purpose of this project is to develop a game with a Natural User Interface. As a developer the first decision I had to make was what type of technology to use to complete this. I decided I would use a combination of the MYO Armband; for moving the player, shooting, and pausing and unpausing the game; and Voice Control; for the menus and for muting the game. I chose MYO due to the wide variety of gestures it has, such as the hand, arm, and arm rotation gestures.



I then began to look at a wide variety of games that I felt could be improved or complemented with by the MYO armband. I looked at many retro old school games such as Tetris, Pacman, and Streets of Rage. I eventually settled on the idea of developing a Space Invaders like game in Unity. My reasoning for this was how natural the game and the MYO armband sync up together. The wave hand gestures are easy to execute and even easier to understand from a user's point of view.

What the Game looks Like:



Gesture Identified as appropriate for this application

I will refer to the diagram above demonstrating all the available gestures that the MYO armband can detect and will explain where I used them in this application. You should be aware that this application is designed with the MYO armband to be worn on the user's left hand. Therefore, the Wave out will equal to Wave Left and the Wave in will be equal to the Wave right.

- The Wave Left Gesture (referred to wave out in Unity). This Gesture is used to move the player character to the left of the screen.

-The Wave Right Gesture (referred to as Wave In in Unity). This Gesture is used to move the player character to the right of the screen.

-The Spread Fingers Gesture. This Gesture is used to pause the game and to open the pause screen User Interface.

-The Fist Gesture. This Gesture is used to un-pause the game and to close the pause screen User Interface.

-The Double Tap Gesture. This Gesture is used to cause the player character to fire a projectile vertically up the screen at the enemies falling.

Play	This Voice Command will start the game from the main menu

Quit	This Voice Command Will Quit the game from the main menu
Mute	This Voice Command will mute all audio within the game

Each of these Voice Commands have multiple items that will result in the same rule. For example, the rule “Play” has these items:

<item>play</item>

<item>start</item>

<item>start a new game</item>

<item>play a new game</item>

<item>play game</item>

<item>start game</item>

This means that if the user says any of the above phrases then the application will associate them all with “play” and will execute the function the switch statement looks for.

```
switch (phraseWord)
{
    case "play":
        phraseWord="";
        Play();
        break;
```

Hardware used in creating the application

The MYO Armband

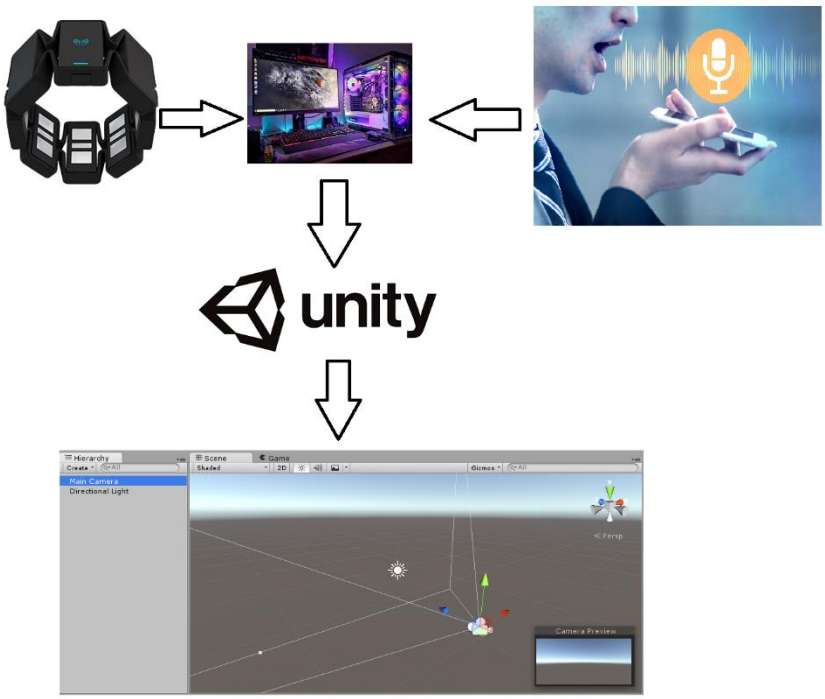
The main hardware used in this project is the MYO Armband. The MYO Armband uses an external USB that communicates with each other via Bluetooth. The Armband also needs an SDK to be installed on the computer. The armband works with Windows, MAC, and Android.

Voice Recognition

Voice recognition is a computer software or program that can decode human voice. I use the voice recognition to allow the user to control the menu aspects if the game.

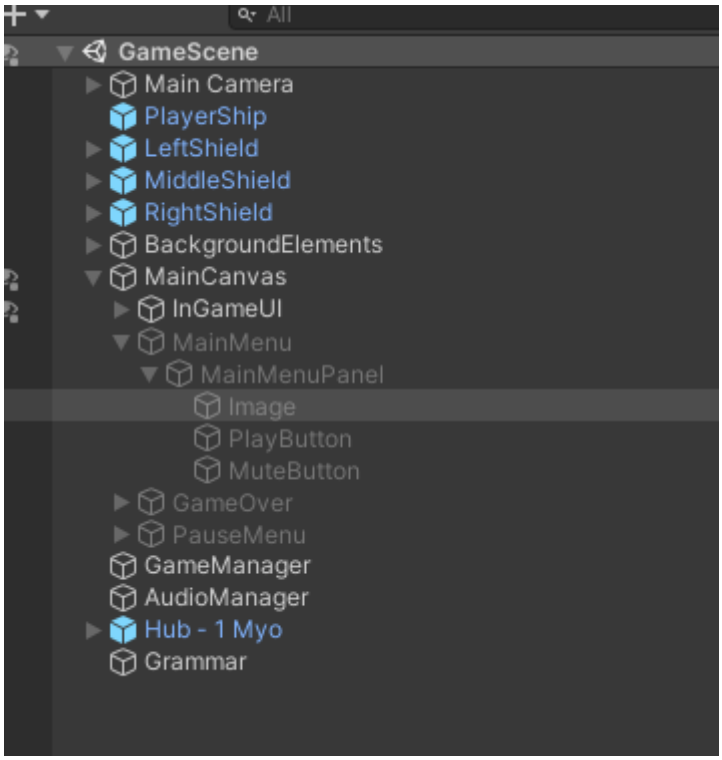
Architecture for the solution

The game is designed in Unity in a 2-Dimensional environment where the player can control the player character with the MYO armband and can execute34 menu functions via voice control. The Aim of the game is to get the highest score possible until the player has run out of lives. This is the Architecture of the game:



The Game uses the MYO Armband to detect what gestures the user executes in order to control the player. There are some main programs that detect the gestures and execute the functions connected to these gestures.

Unity Scene Objects:



This is a list of the objects within my scene.

Unity Scripts:

The Player Controller:

This script keeps track of the MYO Armband and of the gestures the player executes.

The Menu Grammar:

This script keeps track of the phrases read in from the user's default microphone. If play is stated, it will hide the menu UI and Display the game's UI. This also works with Mute and Quit.

There are many other scripts in the application, but these do not interact with gestures at all and only run the game, so I won't be explaining them here.

Conclusions & Recommendations

I am happy with my finished project. I set an objective to create an application in Unity that will implement the MYO Gestures, and that is what I accomplished. I also added some voice recognition for added functionality that I feel really compliments the MYO Gestures making for a fun and interactive experience for the User and the Gamer. My goal was to make a game for everyone to enjoy that tried it, and I believe I achieved that.

I was very excited to implement the MYO Armband into an application of my own, but unfortunately, I cannot say that the hardware is exactly what I had hoped for it to be. The equipment must be worn in a very specific way to optimally work for the User, and without the proper guidance from someone with this knowledge, the game will more than likely be unplayable for this User. The MYO Armband requires to be work as tight as it can be on the arm, and to be facing a certain way up.

Once the armband has been played on and has powered on, it must be synced up. This requires the user to place their hand in a certain position and hold, not only can this result in the users hands becoming tired, as mine often did when working on this project, it must be warmed up once it has synced, and once it has warmed it, it must be re-synced. This can result in the user wearing the band for a solid 10 minutes before they even start the application. This is somewhat understandable as the hardware hasn't received an update since 2013, but non-the-less, I felt that I should specify my own quarrels when using the hardware.

Overall, I am still very happy with how the project has turned out and have lost more hours than I care to admit playing the game instead of actually working. The chance to work on something as interesting as this has been fun and I'm pleased with the outcome.

How to Run

To Run the application, you will need the following:

- Windows device (make sure English (US) is your speech language in your settings)
- MYO Armband
- A working microphone
- Unity

Then open the project in unity and run it.