# Gesture Based UI Project

Emmanuel Osabuehien (G00373559)

May 5, 2022

# Contents

# 1 Introduction

For this project, we had to develop an application with a Natural User Interface where we are given creative freedom to choose whatever programming language and technology we wish to use to create our application. This project must be a gesture controlled application in which we were given the option of using either the Myo Armband or the Leap Motion Sensor as our form of technology.

The concept our my game is to recreate my very own version of old arcade game 'Space Invaders' using the programming language 'Java'.

I decided to go with the Myo Armband as it's use of pre-determined gestures coincided greatly with the concept of my application. "Rather than using special cameras that sense motion and depth, startup Thalmic Labs' Myo armband detects a wearer's movements and translates them into computer controls. After slipping the Myo armband on your forearm, its sensors begin reading electrical activity in your muscles".

This application also incorporates the use of voice commands which I also learned in the duration of this course, the user can control when the game starts, when the game can be paused and resumed, and when the game is reset just by the sound of their voice. The actions of our player can be controlled manually using the keyboard or by using the Myo Armband.

# 2 Purpose of the application

## 2.1 What Is The Game

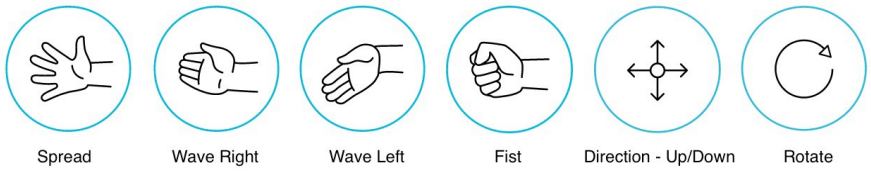


Figure 1: Example of Different Gesture Controls

The purpose of this game is to learn, understand, explain and implement the use of voice commands and gesture control with various technology, for the voice commands we will be using microphones and for the gesture control we will be using Myo Armband. For my application, I decided to recreate the classic arcade game 'Space Invaders' with the programming language Java, to create our we used an integrated development environment (IDE) known as 'Eclipse'. The Game uses gesture control to control the movement (moving left and right)

and actions (shooting bullets) of the player and uses voice commands to control actions of the game where you can start, pause, resume and reset the game.

## 2.2 Design Of The Game

The design of the game should look as follows:

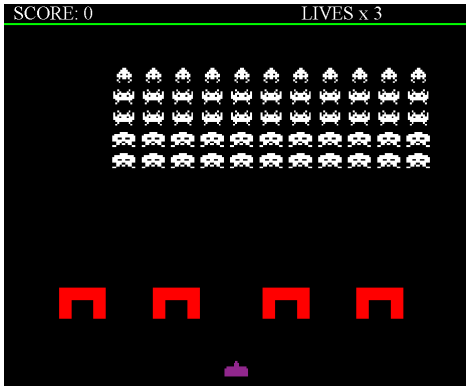


Figure 2: Image of Java GUI

## 2.3 How To Run Application

For our application, we have a Myo Script using the programming language Lua where have a function that connects the gesture controls to our application where if it locates the title of the applicatin 'GUI Project' it will return true, the code is as follows:

```
function onForegroundWindowChange(app, title)
    if title == "GUI Project" then
        reference = getRoll()
        return true
    end
end
```

## 2.4 Voice Commands

When implementing voice commands, I used various jar files that I researched and found online to complete the use of voice controls, the main jar files known as 'Voce' that can be used by both Java and C++.

```
public <String> = [start | reset | stop | continue ];
```

We set up an array of strings which contain the four voice commands that our game will use which include:

- Start: To start the game
- Stop: To pause the game
- Continue: To resume the game when paused
- Reset: To reset the game from the beginning

# 3 Gestures identified as appropriate for this application

## 3.1 Various Gestures Identified By Myo

Unfortunately my the partner that I originally was working on this project with, decided to sadly leave this course meaning I had full control and freedom to decide how I would incorporate gestures into my application.

I was given two different options for hardware when using gesture control, using either the Myo Armband or the Leap Motion Sensor, I researched both pieces of hardware and after completing my research I decided to use the Myo Armband due to it's pre-built gestures which I decided would work great with my concept of the application, the Myo Armband has seven pre-built gestures which include:

- Wave Left
- Wave Right
- Rotate
- Spread Fingers
- Double Tap
- Make Fist
- Direction - Up/Down

When implementing the gestures into my application, we initially used the spread fingers gesture to control the shooting of the player and the wave left and wave right to control the movement of the player but as we progressed in the development of the project we decided to change this due to different reasons.

After using the wave left and wave right gesture to control the movement, this became strenuous to continually have to wave your arms in both directions so we decided to use the rotate gesture instead as it didn't exhaust the user to rotate your hand clockwise and anti-clockwise to move the player from right to left and from left to right.

After using the spread fingers gesture to control the shooting of the player, this gesture was not quick to react which meant I had to switch to something else, so I decided to go with the make fist gesture as it reacted on time and was much more precise.

# 4 Hardware used in creating the application

## 4.1 Myo Armband

When using gesture control, we decided to use the Myo Armband. "The Myo armband is a gesture controller that triggers a variety of actions on the computer based on the contractions of your muscles and the movements of your arm".

Figure 3: Image of Myo Armband

To use the Myo Armband, you will need to download and install the Myo SDK and Connector, this is available for Windows, Mac, Linux and Android.

## 4.2 Microphone

We also used voice commands for the application, using a pre-built computer microphone. Voice recognition software lets you dictate documents, search the web, e-mail, and more on your computer quickly and accurately just by using your voice.

# 5 Architecture for the solution

## 5.1 Java

I used Java as my programming language to recreate the famous arcade game 'Space Invaders'. I separated the Java classes into different packages, we have

packages fileld will classes to control the Enemies, to control the Blocks, to control the Scoreboard, to control the Bullets, to control the Sounds and to control our Player. You will see down below the structure of the Java files:


Figure 4: Java Project Structure

## 5.2 Lua

I used Lua as my programming language to connect Myo Armband to the application, we also used to this to map the gestures to the keyboard. Here are some exmaples of Lua being used in our application:

```
8  function moveRight()
9      myo.keyboard("right_arrow","down")
10     rHold = true
11 end
12
13 function fire()
```

```
14      myo.keyboard("space","down")
15      myo.vibrate("short")
16  end
```

Above we can see an example of mapping the gesture to the keyboard to move right and to shoot, so when the user rotates the armband right/press the right arrow key then the player should move right or when the user makes a fist/-presses spacebar then the player should shoot.

## 5.3 External Jar

We used external jar files for various functions for our app, for example we used the jar file 'Voce' to add voice recognition to our app. Here we have the various libraries that were used to create this application.

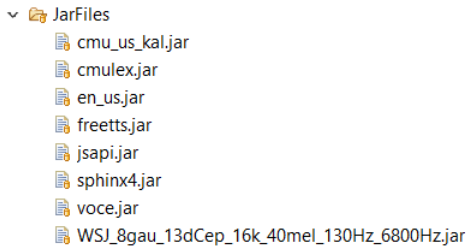


Figure 5: External Jar Library

## 5.4 Voice Commands

Using the External Jar Library knownn as 'Voce', I have been able to add voice commands to my app, I use these commands to start, pause, resume and reset the current game once it has been run.

```
17      public void speech() {
18
19          System.out.println("Say Start!");
20
21          boolean start = false;
22          while (!start) {
23              // Normally, applications would do application-
                    specific things
24              // here. For this sample, we'll just sleep for a
                    little bit.
25              try {
26                  Thread.sleep(300);
27              } catch (InterruptedException e) {
28              }
```

```
29              while (voce.SpeechInterface.getRecognizerQueueSize
                    () > 0) {
30                  String s = voce.SpeechInterface.
                        popRecognizedString();
31
32                  // check if s is equal to start
33                  if (s.contentEquals("start")) {
34                      // if it is then end the loop
35                      // start = true;
36                      // start the game
37                      startGame = true;
38                  } else if (s.contentEquals("stop")) {
39                      paused = true;
40                  } else if (s.contentEquals("continue")) {
41                      paused = false;
42                  } else if (s.contentEquals("reset")) {
43                      startAgain = true;
44                  } else {
45                      s = "empty";
46                  }
47                  System.out.println("You said: " + s);
48              }
49
50          }
51
52          voce.SpeechInterface.destroy();
53
54      }
```

# 6 Conclusions & Recommendations

I am thoroughly impressed with the outcome of my application, we had some bumps in the road with the development but it all came together in the end and we completed our goal in developing a Natural User Interface that incorporates the use of gesture control. The purpose of my app was to recreate a personal version famous arcade game for others to have fun with and I personally believe I have completed that task.

Now I know what you're wondering, were there any issues or challenges, just like in the development of completing any software product there is always gonna be some issue, error, fault or challenge to overcome, here are some issues that I face along the way:

- Research was difficult as there was a very limited amount of content I can find that helped me achieve what I was aiming for
- There were issues with the Myo Armband itself, it was at times difficult

for the gestures to be recognized and the Armband at times was difficult to connect and disconnect.

- Learning gesture control for the first time became a challenge trying to get all the technology working together.
- My partner chose to leave this course during the development of the project which was a real challenge in conjunction with the struggle to figure out what the idea for the application was going to be.
- At one point during the development, all my java classes strangely had errors that were difficult to understand what the issue was, I then had to start from scratch and had to redo all my work that I had up to that point.

In conclusion, the overall experience with this project has been a rather pleasant, entertaining and quite intuitive, I have gained so much knowledge in regards to gesture control and voice recognition throughout this whole course, whether it's how they work, how it was invented or how it has progressed throughout history as opposed to before this project where I had little to no knowledge of gesture control technology. In today's world, while Myo might be considered not the best or outdated software, I believe it is a great piece of software in the use of projects such as my own to gain an understanding on this type of technology, a lot of time and effort was put into this project and in the end the outcome was favorable and to my satisfaction.

# Appendices

- Click This Link To View GitHub Repository Used To Develop This Project
- Click This Link To View A Demo Of The Application

## References


[1] Pekka Aho et al. "Automated Java GUI modeling for model-based testing purposes". In: *2011 Eighth International Conference on Information Technology: New Generations*. IEEE. 2011, pp. 268–273.

[2] Axel Andersson et al. "Space invaders". In: (2015).

[3] Marco E Benalcázar et al. "Hand gesture recognition using machine learning and the Myo armband". In: *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE. 2017, pp. 1040–1044.

[4] Marco E Benalcázar et al. "Real-time hand gesture recognition using the Myo armband and muscle activity detection". In: *2017 IEEE second Ecuador technical chapters meeting (ETCM)*. IEEE. 2017, pp. 1–6.

[5] Nona Fernández Silanes. *Space invaders*. Vol. 771. Fondo de Cultura Economica, 2020.

[6] Jože Guna et al. "An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking". In: *Sensors* 14.2 (2014), pp. 3702–3720.

[7] Roberto Ierusalimschy. *Programming in lua*. Roberto Ierusalimschy, 2006.

[8] Roberto Ierusalimschy, Luiz Henrique de Figueiredo, and Waldemar Celes. "The evolution of Lua". In: *Proceedings of the third ACM SIGPLAN conference on History of programming languages*. 2007, pp. 2–1.

[9] Josh Juneau et al. "GUI Applications". In: *The Definitive Guide To Jython*. Springer, 2010, pp. 347–357.

[10] Amit Mehta. "Voice recognition". In: *PACS*. Springer. 2002, pp. 281–302.

[11] Leap Motion. "Leap motion". In: *San Francisco, CA, USA* (2015).

[12] A Naressi et al. "Java-based graphical user interface for MRUI, a software package for quantitation of in vivo/medical magnetic resonance spectroscopy signals". In: *Computers in biology and medicine* 31.4 (2001), pp. 269–286.

[13] Tyler K Perrachione, Stephanie N Del Tufo, and John DE Gabrieli. "Human voice recognition depends on language ability". In: *Science* 333.6042 (2011), pp. 595–595.

[14] Seema Rawat, Somya Vats, and Praveen Kumar. "Evaluating and exploring the MYO ARMBAND". In: *2016 International Conference System Modeling & Advancement in Research Trends (SMART)*. IEEE. 2016, pp. 115–120.

[15] Mais Yasen and Shaidah Jusoh. "A systematic review on hand gesture recognition techniques, challenges and applications". In: *PeerJ Computer Science* 5 (2019), e218.