

2021

# Space Invaders Game

SAGHEER AHMAD - STUDENT

Gesture Based UI Project – Gesture Based  
User Interface

## Project Link:

<https://github.com/sagheerahmadGmit/GestureBasedProject>

## Video/Screencast Link:

<https://www.youtube.com/watch?v=JKnMI29xOpl>

## Introduction

---

The aim of this project was to develop an application with a Natural User Interface. There were a number of options available to us such as the Kinect, Myo armband, voice and the oculus quest. The project I will be developing will be incorporating the Myo armband as the main technology for gestures. It will also contain voice controls for navigating through the game. The game that I decided to create is Space Invaders. I chose this game because it can use the Myo more effectively than other games. The user can control the game through voice and hand and arm movement. I decided to write this game in Java using Eclipse IDE.

## Purpose of the application

---

The purpose of this application is to demonstrate how various gestures, such as voice and a Myo armband, can be used to control a game. The game contains voice controls to start the game, pause the game, reset the game and exit the game, while the Myo is used to control the player ship left and right and shoot the enemies. The other purpose of this application was an experimentation process to see how gestures work and where they are headed in the future.

### Myo Armband:

In order to run the Myo armband with the game we need to download the Myo connector application and make sure the Myo is connected to the computer. In the project I have a file for setting up the Myo armband to the game. In order to connect the Myo to the game we need to run this script by double tapping the script and Installing the Myo. After installing the Myo script we will be able to connect to the

game and play using the Myo armband. The user can rotate their arms to move the player ship and clench their fists to shoot the enemies.

Voice Gestures:

The game contains voice controls to start the game and allow the user to restart and pause. In order to run the voice commands in Java I had to use external libraries such as Voce and Sphinx4. I have a grammar script that contains the main keywords for the game. The user will be given instructions on how to run the game and the voice controls through the CLI Main menu.

```
-----  
Welcome to Space Game!!  
In order to start the game please say 'Start'!!  
In order to pause the game please say 'pause'!!  
In order to continue the game please say 'continue'!!  
In order to restart the game please say 'restart'!!  
In order to quit the game please say 'exit'!!  
-----
```

This is just a simple main menu that gives instructions to the player on how to play the game.

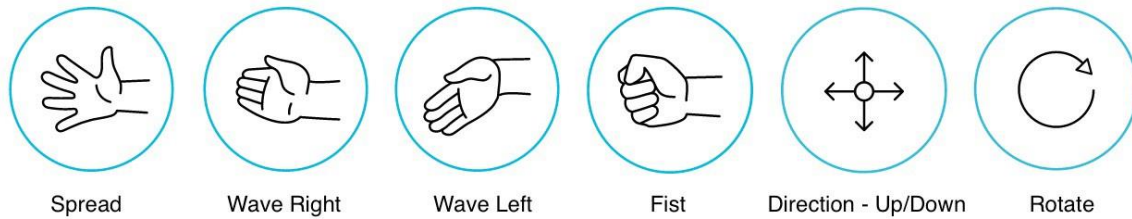
```
1 #JSGF V1.0;  
2  
3 /**  
4  * JSGF| Digits Grammar file  
5  */  
6  
7 grammar start;  
8  
9 public <String> = [start | play | restart | stop | pause | resume | continue | exit];  
10
```

Depending on what the user says from the above grammar script, the game will run different commands.

## **Gestures identified as appropriate for this application**

---

When researching into the Myo armband, I discovered that the Myo has a number of gestures built into it. See the image below for examples.



I am using the rotate gesture to move the player left and right on the screen. While testing the Myo I realised that making gestures can be tiring and can put strain on your arm. Because of this I wanted to use the easiest gesture from the Myo to move the player. That is why I am using the rotate gesture. The player just rotates their arm to the left or the right and the ship will go that way. In order to shoot the aliens, I am using the fist gesture. I chose this gesture because it felt more comfortable for a shooting action than spreading your fingers and hand. It also imitates a person pulling a trigger in a way.

The other gesture that seemed appropriate for this game was voice. Instead of doing a complicated GUI that the user can navigate using the Myo armband causing the player to be tired and have a sore arm, I went for a simpler approach and decided to use voice instead. The user is presented with a simple command line interface that will tell them how to start the game and how to play it.

## Hardware used in creating the application

---

The hardware that I decided to use for my project are the Myo Armband and my Microphone. There were a lot of good options to go with such as the kinect and the oculus quest or even a camera for image recognition. The reason I went for the Myo and voice gestures is that I feel like it is more of a natural way of playing a game where the player uses their arm movement to control an object and voice to tell the game what to do. This makes the user feel more involved in the game and makes them feel like they are a part of the game.



In order to run the Myo band we need a couple of things such as the Myo SDK and the Myo Connector application. These can all be obtained from the Gesture Based User Interface module on Learn online.

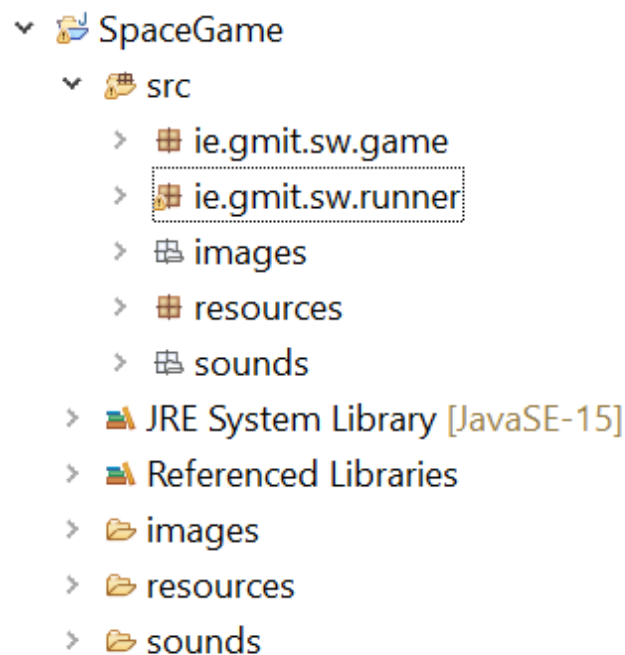
- <https://learnonline.gmit.ie/course/view.php?id=2799>

All we will require for the voice gestures is a microphone or an integrated microphone already built into the computer for allowing the user to control the menu aspect of Space Invaders. For my testing I will be using an external microphone.

## Architecture for the solution

---

This is the structure of my game in eclipse. I am using multiple packages to sort the different aspects of the game such as sounds, images and the game itself. The game is made through JFrame windows using the Javax swing library.





```

if (s.contentEquals("start") || s.contentEquals("play")) {

    start();
    System.out.println("Game Started");
    System.out.println("Rotate your arm left and right to move!");
    System.out.println("Make a fist to shoot!");

} else if (s.contentEquals("stop") || s.contentEquals("pause")) {

    pause();
    System.out.println("Game is currently paused - say Continue to unpause");

} else if (s.contentEquals("continue") || s.contentEquals("resume")) {

    pauseWindow.setVisible(false);
    gameWindow.setVisible(true);
    System.out.println("Game is unpaused - say Stop to pause");

} else if (s.contentEquals("restart")) {

    gameWindow.dispose();
    gameWindow.repaint();
    start();
    System.out.println("Starting a new Game");

} else if (s.contentEquals("exit")) {

    System.out.println("Exiting Game...");
    System.exit(0);
} else {
    s = "Keyword Not Detected";
}
System.out.println("You said: " + s);

```

Scene: This class is responsible for connecting all the sprites together and painting the scene that will be passed to the runner class and turned into a game window.

Pause Scene: This class paints the scene that will appear when the user says pause and continue.

Alien, AlienGroup, AlienShot: These classes are responsible for drawing the aliens using the images and allowing the aliens to move as a group. There are 50 aliens in the game. When the player kills the alien, that alien gets removed from an array and is destroyed. It also makes sure that the aliens stay inside the boundaries. The AlienShot class creates the bullet for the enemies and fires them randomly.

Entity: This class contains the getters and setters for the player ship.

Obstacle, Saucer: The saucer class paints the red saucer that flies above the game and is worth the most points (100 points). It flies from the right to the left. While the obstacle class draws the four shields just above the player so the player can hide and protect itself.

Player, PlayerShot: These classes are responsible for creating and drawing the player and the player bullet that will be used to kill the player.

## Conclusion

---

I am happy overall with the outcome and I feel like I achieved what I wanted. There were a couple of challenges with using the Myo such as:

- The Myo armband is very buggy and did not work a few times.
- The arm band also takes a couple of minutes to warm up which can be a little annoying for the user.
- There are not a lot of resources online related to the Myo armband at this time and the websites that did cover the Myo content are no longer working either. I originally wanted to make a game in python but there are no resources that work effectively so I had to change to java.
- If we compare the Myo to a keyboard, we can clearly see that the Myo is slower in getting a response. By the time the Myo sends a response to the game it might be too late.

Overall, using the Myo and voice commands was a great learning experience. The Myo was not too difficult to setup and implement into a java game. I found a couple resources that helped me with this part and made it easier to setup. I believe that this module has given me great insight to where gesture-based technology is headed and what it could look like in the future. I feel there are a lot of things that were wrong about the Myo armband such as the fact that it is very tiring and puts a lot of strain on the user's arm. It is also difficult to use Myo with other platforms as each platform requires different packages and SDK files.

There are also an excel sheet for tests for this application in the documentsAndTests folder of the project.

## How to run

---

Please make sure you have the following software and hardware:

- Myo Armband
- Myo Connect software
- A microphone
- Myo SDKs
- The required libraries for the project such as the ones below

Once all those are setup then all that is required is to git clone the repository, open the project in an IDE such as eclipse and run the runner class.



## Resources

---

[1] Myo Scripting Orientation and Gestures Part 2

[[Myo Scripting Orientation and Gestures Part Two: Wait for it...](#)]

[2] Myo Scripting Orientation and Gestures Part 3

[[Myo Scripting Orientation and Gestures Part Three: Don't YAWn, this is important!](#)]

[3] Space Invaders in Java

[[https://www.youtube.com/watch?v=0szmaHH1hno&ab\\_channel=gasparcoding](https://www.youtube.com/watch?v=0szmaHH1hno&ab_channel=gasparcoding)]

[[https://www.youtube.com/watch?v=afVI-TRFVKE&ab\\_channel=StefanMaurer](https://www.youtube.com/watch?v=afVI-TRFVKE&ab_channel=StefanMaurer)]

[[https://www.youtube.com/watch?v=oNPqnfsC6J8&ab\\_channel=TheWalkthroughKiller](https://www.youtube.com/watch?v=oNPqnfsC6J8&ab_channel=TheWalkthroughKiller)]