LAURA OGBA & TIANLE SHU- STUDENT

Project for Gesture Based

UI Development

Contents

[**Purpose of the application** 2](#_Toc5923636)

[**User interface** 2](#_Toc5923637)

[**Gestures identified as appropriate for this application** 2](#_Toc5923638)

[**Hardware used in creating the application** 3](#_Toc5923639)

[**What Is Myo-armband?** 3](#_Toc5923640)

[**How is MYO connected to the paddle?** 4](#_Toc5923641)

[**Turning paddle to the left:** 4](#_Toc5923642)

[**Turning paddle to the right** 4](#_Toc5923643)

[**Releasing the ball** 4](#_Toc5923644)

[**Conclusions & Recommendations** 5](#_Toc5923645)

**Purpose of the application**

The application my group partner and I have decided to design is a classic game called ‘Brick Breaker’. Brick Breaker (platformer) is a Breakout clone in which the player must smash a wall of bricks by deflecting a bouncing ball with a paddle. The paddle may move horizontally and is controlled with the BlackBerry's trackwheel, the computer's mouse or the touch of a finger (in the case of touchscreen). When all the bricks have been destroyed, the player advances to a new level. There are 34 levels. There are many versions of brick breaker, some in which you can shoot flaming fireballs or play with more than one ball if the player gets a power up.

This game was developed through unity, which is the ultimate game development platform. This platform is used to develop 2 and 3 dimensional games. Our game was develop using 2D. once the game was developed, the next step was to connect myo-armband to unity. This was done using the myo-SDK which can be found online at the MYO website (please find link in our git repository). This package is then imported into unity and then you can follow the steps of synching the armband to your machine, through a Bluetooth that comes with the band.

# **User interface**

Our game has only one interface, this is just the start screen. Once the game is played in unity this is the screen that is shown to the user, the game starts with a countdown of three seconds.

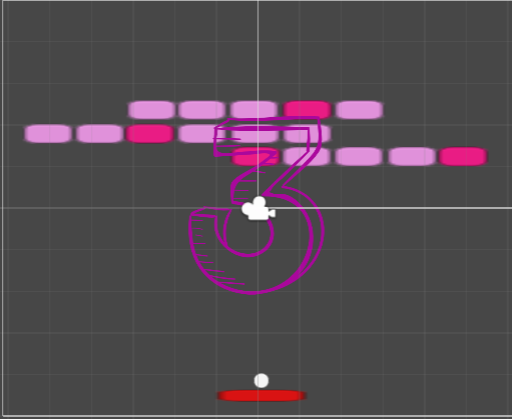


Fig: 1.

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

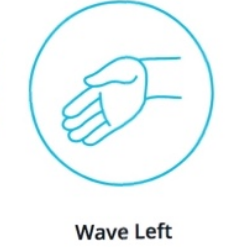
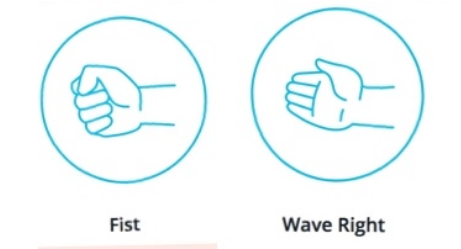
As the purpose of the game is to have User Interface gestures, we incorporated this into the game. Originally, the game is controlled using the basic mouse and keyboard to control the paddle board in the game. The ball that breaks the bricks is also controlled by the space bar in the game. We changed this into a gestures game to allow a device called myo-armband to control the game.

The first steps to controlling the game, is to ensure that the armband is worn on the arm and is then synchronised with the computer. Then you can begin to play the game. We designed the game to sync with the myo-armband, for moving the paddle the user must wave in to move left and wave out to move right. This allows the paddle to move across each side of the game. To release the ball the user must use the fist position.

. In order to control the game, the user must use these gestures:

* Fist: To release the ball.
* Wave in: To slide the paddle left.
* Wave out: To slide the paddle right.

We felt these were the only appropriate gestures for the game, as it was difficult to incorporate any other type.



# **Hardware used in creating the application**

The hardware that we used for developing the gestures of the game is a Myo-armband. This allowed the user to control the paddle and the ball in the game.

## **What Is Myo-armband?**

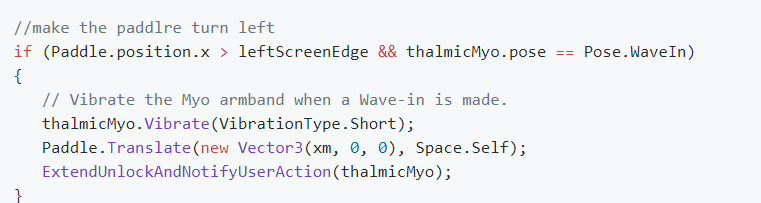
The Myo-armband is a gesture recognition device worn on the forearm and manufactured by Thalmic Labs. The Myo-armband enables the user to control technology wirelessly using various hand motions. It uses a set of electromyographic (EMG) sensors that sense electrical activity in the forearm muscles, combined with a gyroscope, accelerometer and magnetometer to recognize gestures. The Myo-armband can be used to control video games, presentations, music and visual entertainment. It differs from the Leap Motion device as it is worn rather than a 3D array of cameras that sense motion in the environment.



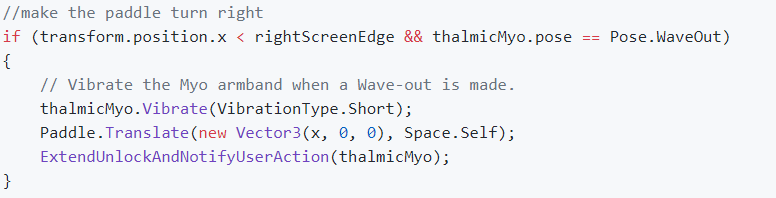
## **How is MYO connected to the paddle?**

In order to connect the MYO to allow it to control the paddle board in the game. We must set the position, of the paddle on the screen so that MYO would recognise it each time the gestures are made. Below are screen shots of the different position set for each gesture when playing the game.

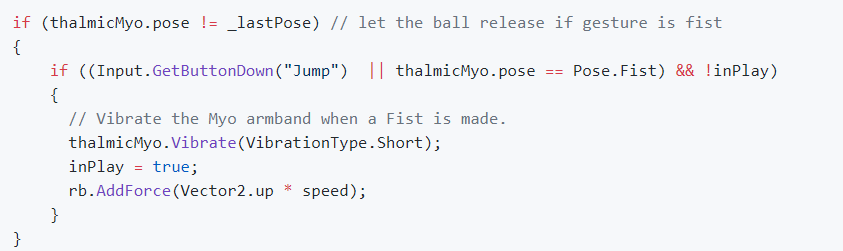
### **Turning paddle to the left:**



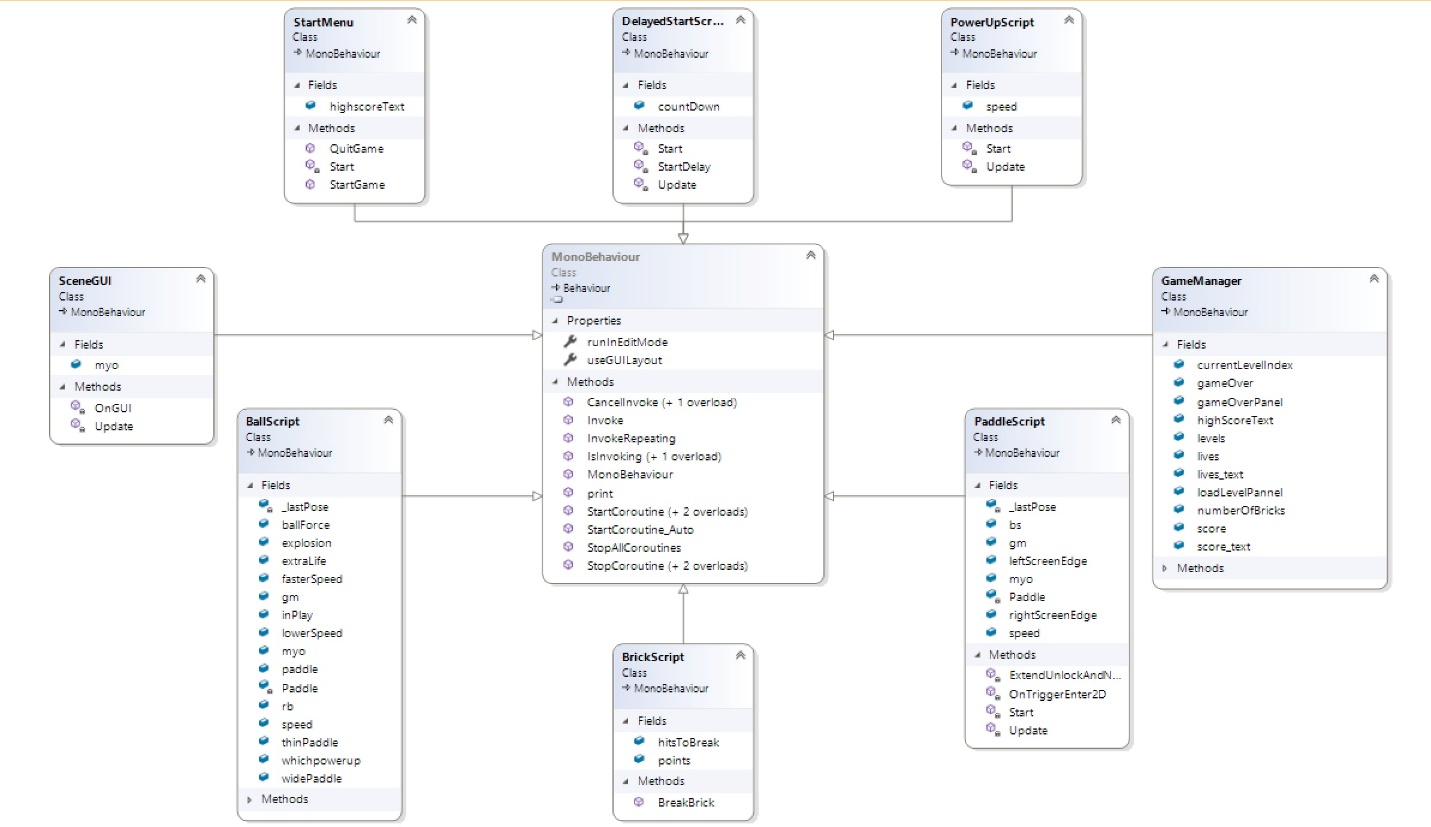
### **Turning paddle to the right**



### **Releasing the ball**



# Architecture for the solution



The main script connects with myo, it is BallScript and PaddleScript for connect myo, like that’s code explained above. And have 3 script for the game like StartMenu for game start scene for the quite button click function and start button click function. And DelayStartScript allow user have 3 seconds for prepare when the game started.

And the SceneGUI sript that is for tell user if the myo connect with computer.

**Conclusions & Recommendations**

In conclusion, we learnt a lot from this project. Creating a user interface gestures project and working with technology that is new was very mush exciting. Although at first, we ran into major problem.

Originally, we wanted to develop a gestures project for a fruit slicing game ‘Ninja fruit’. We ran into a lot of problems, in trying to connect the myo-armband and the knife that would be used to cut the fruits which were thrown up into the air. This was difficult as it was hard to get the right position for this, the knife didn’t have a certain direction it was going. It went in all directions in which the fruit was displayed. Hence, we changed the project to something much simpler to control.

The brick breaker game was just that, it allowed us to be in control. Gave us a bit more freedom to use the myo-armband, when developing the previous game, the user experience would have been very poor. This is because the gestures would have been hard for the user to get, and they would get tired easily. Whereas, with the brick breaker game, the user just must swipe left or right to move the paddle and fist to release the ball. It’s a more user-friendly game.

Our recommendation, would be that if anyone was to develop a game using the myo-armband they should take the following into consideration:

1. Choose a game that can be easily controlled and user friendly.
2. When developing the game, make sure simple gestures are used to prevent users from getting tired.
3. Try other hardware instead of MYO, this is because MYO is not reliable.
   * It gets disconnected easily.
   * It doesn’t always detect the right gestures.
   * Its uncomfortable for long usage periods.
   * Unpredictable behaviour when battery is low.