**Gesture Based Programming Project**

Team Members: Garry Cummins, Tim Cassidy, Hugh Brady

Student IDs: G00335806, G0333333, G00338260

Contents

[**Project Title** 3](#_Toc6285796)

[**Github Repository URL** 3](#_Toc6285797)

[**Project Rundown** 3](#_Toc6285798)

[**Gesture Consideration and Use** 4](#_Toc6285799)

[**Menu Mode Gestures** 4](#_Toc6285800)

[**Game Mode Gestures** 4](#_Toc6285801)

[**Select Mode Gestures** 4](#_Toc6285802)

[**Problems Faced** 5](#_Toc6285803)

[**Unity Version** 5](#_Toc6285804)

[**Merge Conflicts** 5](#_Toc6285805)

[**Accelerometer Issues** 5](#_Toc6285806)

[**Project Workload** 6](#_Toc6285807)

[**February 15th  (Github setup, project outline)** 6](#_Toc6285808)

[**February 23rd – 25th (Singleton functionality , Menu UI, Locks) - Tim** 6](#_Toc6285809)

[**March 7th(Create Branch, Mode specific testing, fixing merge conflicts) Menu Branch** 6](#_Toc6285810)

[**March 7th(Create Branch, Setup , Merge branches) Create Branch – Hugh** 6](#_Toc6285811)

[**March 7th(Create Branch, Gesture behaviour script, error fixes) Select Branch – Tim** 6](#_Toc6285812)

[**March 15th(Myo functionality, testing objects added) Select Branch – Tim** 6](#_Toc6285813)

[**March 29th(Project version fix, methods setup for testing) Create Branch – Hugh** 7](#_Toc6285814)

[**March 29th(Save load scripts, project version fix, menu functionality) – Menu Branch** 7](#_Toc6285815)

[**March 29th(Fix versioning issue, object focus script) – Select Branch - Tim** 7](#_Toc6285816)

[**March 30th(Array index, Myo functionality) – Create Branch – Hugh** 7](#_Toc6285817)

[**March 30th(Object movement with gyroscope) – Select Branch – Tim** 7](#_Toc6285818)

[**March 30th(Save and load features, object passing to select scene) – Menu Branch** 7](#_Toc6285819)

[**April 1st(Object error fix) – Menu Branch** 8](#_Toc6285820)

[**April 3rd(Delete and exit options added, updated select mode script) – Select Branch - Tim** 8](#_Toc6285821)

[**April 3rd(Main scene and merge conflict fixes) – Menu Branch** 8](#_Toc6285822)

[**April 4th(Menu and myo functionality, buttons) – Menu Branch** 8](#_Toc6285823)

[**April 4th(Myo functions, toggle highlight, change poses, fixed merge conflicts, UI messages resolved) – Create Branch** 8](#_Toc6285824)

[**April 11th(Load UI added) – Master Branch** 8](#_Toc6285825)

[**April 12th(Saving data to file) – Master Branch** 9](#_Toc6285826)

[**April 14th(Merged scripts) – Master Branch** 9](#_Toc6285827)

[**April 15th(Save and Load) – Master Branch** 9](#_Toc6285828)

## **Project Title**

Myo Sandbox

## **Github Repository URL**

<https://github.com/yrrag5/Gesture-Based-Programming>

## **Video Demo URL**

<https://www.youtube.com/watch?v=eK1AQSL-siE>

## **Project Rundown**

For this project we used the Myo armband, a device that uses a Gyroscope and Accelerometers to track arm movement and position, to capture user gestures made, the Myo comes with 5 pre-programmed gestures. We created a 3D sandbox environment in Unity in which the user will be able to instantiate objects and position them using the pre-programmed gestures with the Myo armband. The user will have three objects to choose from: cuboid, cube and cylinder. These objects can be stacked and will have physics acting upon them. The user can create or destroy objects in the sandbox environment.

## **Gesture Consideration and Use**

Since the Myo armband only provides 5 predefined gestures we could use, we decided on splitting the project into three different modes (Menu, Game, Select) In order to make better use of these gestures. We made sure that the gestures would make sense in the context of our project.

### **Menu Mode Gestures**

* Wave Left, Right – Used for navigating in the menu. The user will first be highlighted on the continue button. Wave right will move the user through the button array to highlight the load button and continue moving through the buttons. Wave left will go the opposite direction and initially, will highlight the cylinder button.
* Fingers Spread – Will notify the user that the application will be exited if this gesture is repeated.
* Double Tap - Used to click button to perform its function.

### **Game Mode Gestures**

* Wave Left - Parses left through the array, highlighting the selecting object.
* Wave Right - Parses right through the array, highlighting the selecting object.
* Fingers Spread - Exits to Menu Mode.
* Double Tap - Enters Select Mode, passing the highlighted GameObject.

### **Select Mode Gestures**

* Wave Left, Right - Used to change the axis of movement (X, Y, Z) of the object. It was mainly used because parsing through the axes was like the parsing of the game objects and buttons from the Game and Menu modes.
* Fingers Spread - Used to exit Select mode and enter the Game mode. User will need to repeat the gesture in order to confirm.
* Double Tap – Used to delete objects from the scene state array of game objects. Repeat gesture to confirm. When the object is deleted, the Game mode will then be entered and displayed
* Close Fist - Used to move the object on the selected axis. While held the user can change the pivot angle of the elbow to move. Increase angle to increase axis value. Decrease angle to decrease axis value. This gesture was used because it is like holding an object.

## **Problems Faced**

### **Unity Version**

When we began the project, we all had different unity versions. This caused us issues when merging branches on GitHub, so we all decided to update to Unity version 2018.2.12f to avoid future conflicts.

### **Merge Conflicts**

When the branches were combined, merge conflicts occurred in the main scene, this was due to minor changes being made to the main scene that were not pulled down before editing the scene. This was solved by ensuring a pull request was preformed from GitHub before any scene changes were made.

### **Accelerometer Issues**

When retrieving the accelerometer readings from the Myo we faced multiple issues, the first of which was erratic readings which produced undesired object movement. Another issue we faced with the Myo’s accelerometers was a dropout of accelerometer readings, the only way to temporarily fix this issue was to turn off the Myo, turn it back on and re-sync it. These issues were found on two different Myo armbands, so the issue could lie with the devices themselves. The solution to these issues was to avoid using the accelerometers and to use the gyroscope instead, this achieved a constant and stable result which provided greater control over object movement.

## **Project Workload**

### **February 15th (Github setup, project outline) - Garry**

We began initial setup of the project by creating the github repository for it as well as making project outline on what we intend to do with the project**.**

### **February 23rd – 25th (Singleton functionality , Menu UI, Locks) - Tim**

The basic structure for the project was added here. This included the singleton ModeRunner script, which handled the switching of locks to control program flow, MenuMode script, which provided an entry point into the menu functionality, CreateMode script, which provided an entry point into the Game mode functionality, and the SelectMode script, which provided an entry point into the select mode functionality.

### **March 7th(Create Branch, Mode specific testing, fixing merge conflicts) Menu Branch - Garry**

Created a menu branch focusing specifically on the menu UI of the project as well as its functionality. Setup of menu testing scripts to test methods and functions of the UI. Some merge conflicts occurred while we were testing but were resolved.

### **March 7th(Create Branch, Setup , Merge branches) Create Branch – Hugh**

Created a new branch to develop my part of the project. While this branch and the scripts inside it were named create, the actual responsibility of creating objects was lifted from this mode and given to the menu mode later on in development. I also added a script, “CreateTest” to test that Create mode was being entered.

### **March 7th(Create Branch, Gesture behaviour script, error fixes) Select Branch – Tim**

Initially, a gesture behaviour script was developed here to handle the Myo’s gesture state. This however, was removed due to the necessary Myo Hub script already being included in the Myo SDK.

### **March 15th(Myo functionality, testing objects added) Select Branch – Tim**

Here the linking up of the Myo and the already implemented program structure was achieved. A selection of testing objects were made and a script was used to add these to the scene state, this allowed the testing of object functionality for both the Select and Create modes.

### **March 29th(Project version fix, methods setup for testing) Create Branch – Hugh**

Wrote a simple function for creating game objects in the environment. This function was later scrapped, and the mode’s responsibility became navigating through the game objects in the scene. There were also merge issues due to different versions of Unity that had to be ironed out.

### **March 29th(Save load scripts, project version fix, menu functionality) – Menu Branch - Garry**

Added save and load scripts for saving and loading the game scene. Although not implemented correctly and further work will be needed. Changed unity version to match team members to avoid other conflicts. Implemented some functionality for menu buttons to exit application and enter the game mode.

### **March 29th(Fix versioning issue, object focus script) – Select Branch - Tim**

This is where the versioning issue, which is outlined above, was solved. A script was created that was attached to the main camera and would focus on a selected object.

### **March 30th(Array index, Myo functionality) – Create Branch – Hugh**

Added code to the Start function to check the amount of shapes in the objects array and sets the selected variable to the last object in said array. Created methods needed for the mode’s functionality; ParseLeft, ParseRight, ReturnToMenu and ToggleHighlight. Finally, I added code for the Myo to read the incoming gesture and call the appropriate functions.

### **March 30th(Object movement with gyroscope) – Select Branch – Tim**

Object movement using the gyroscope added, which uses the gyroscope pitch to translate the movement of objects in the scene.

### **March 30th(Save and load features, object passing to select scene) – Menu Branch - Garry**

Updated save and load features to save scene to file. Added objects to pass to the select mode.

### **April 1st(Object error fix) – Menu Branch - Garry**

Fixed some errors with the objects. Further work will need to be implemented for correct functionality.

### **April 3rd(Delete and exit options added, updated select mode script) – Select Branch - Tim**

Here object deletion was implemented via the double-tap gesture, program flow was controlled via Boolean locks for the Select and Menu modes when the double-tap gesture is performed. Exiting via the Finger-spread gesture, in all modes, was also achieved by using Boolean locks, when the finger-spread gesture is recognised the user enters an exiting state which can be confirmed by making the gesture again or exited by making any other gesture.

### **April 3rd(Main scene and merge conflict fixes) – Menu Branch -Garry**

Experienced merge conflicts when pulling from the master as unity wasn’t closed properly which caused conflicts with the main scene. Issues were fortunately resolved.

### **April 4th(Menu and myo functionality, buttons) – Menu Branch - Garry**

Implemented myo functionality to allow the predefined gestures to interact with the menu UI and its buttons. Wave left and right will be used for navigating the menu buttons, double tap to click a button to perform its function, finger spread to close application, and fist to bring up a menu for gesture controls and uses.

### **April 4th(Myo functions, toggle highlight, change poses, fixed merge conflicts, UI messages resolved) – Create Branch**

Fixed the logic for checking Myo gestures. Wrote code in the ToggleHighlighted function that works. Also added a function to move the camera to the selected object.

### **April 11th(Load UI added) – Master Branch**

Added a new script, DisplaySaves, that is attached to a new dropdown menu in the scene. This script opens a directory and looks for .csv files, which will act as the format for our save files. These files are added to the dropdown menu. A button was also added and on click, will pass the file name in the dropdown to the LoadState function when the button is clicked.

### **April 12th(Saving data to file) – Master Branch**

Created a singleton script, SaveScene, that loops through the objects array and writes the objects’ names and coordinates to a csv file.

### **April 14th(Merged scripts) – Master Branch**

Merged the SaveScene script with the SceneState script, eradicating the need for a second singleton class.

### **April 15th(Save and Load) – Master Branch**

Re-designed the Save and Load features to use the already implemented singleton functionality from the SceneState script. Minor error fixes and housekeeping preformed.