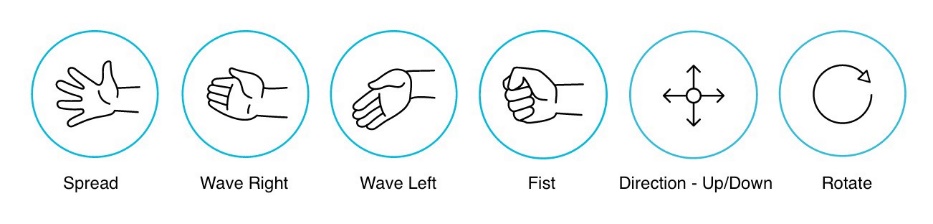
Sean McGuire

**Gesture based UI project**

GitHub: <https://github.com/smcguire56/GestureBasedUIProject>

***Myo Drag Racing***

For my project I will be using the Myo Armband to develop a real-life driving experience in a unity game. The game will be developed in unity and coded in C# and will be focused mostly on the first person steering of a car going around a simple 3D track. At the end of the project I will aim to get the Myo Armband to connect to the game as a user gesture-based control for various gestures such as steering the car, changing gear, and pausing the game.



From the image above these are some of the primitive gestures the armband can provide and through further research I will aim to apply some if not all these gestures for my game.



The Myo armband works by using many sensors to detect small changes in the muscles it is in contact with. Using these sensors, the software can take in the raw input from 8 of the sensors and graph the changes to time. For my project in gesture-based UI I can use these inputs to make the user believe they are driving a car in this game and hopefully make it feel as realistic as possible.

The software and languages I will be using for this project are Unity and C#.

For tracking my progress throughout the project, I will be using GitHub.

**Inspiration for the game**

The inspiration for this game that I am developing is called Nitro Nation 6 which is freely available on the app store. The reason why I choose this game is because there are not too many user inputs required to play the game, changing gear and boost. Therefore, I believe it would have been easy enough to cross over using the Myo armband and the controls for the user would feel easy and natural. Below I have included a screenshot of Nitro Nation and the sort of look and feel I am going for in my game.



Update as of 2 weeks:

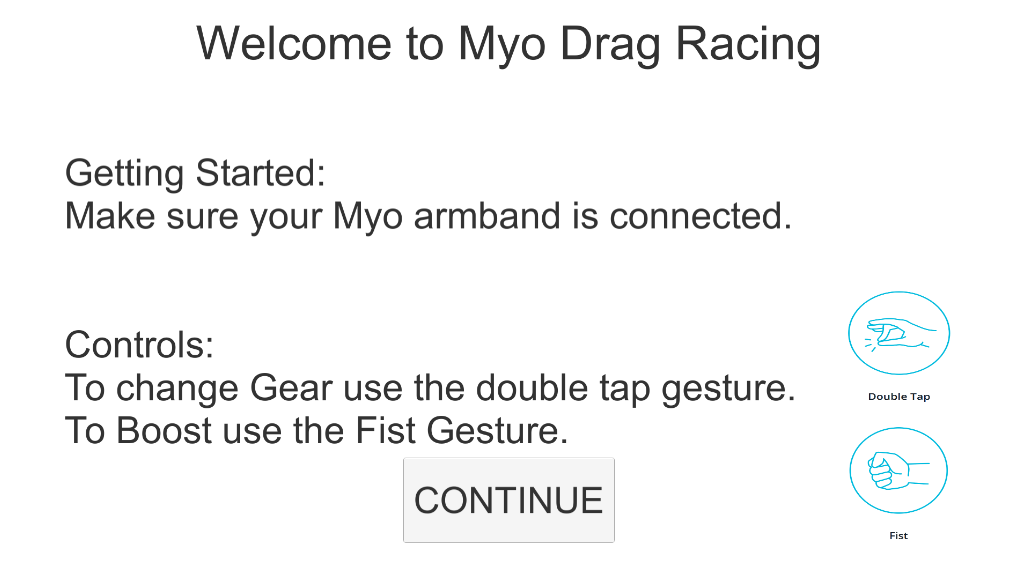
* The game is now converted to just drag racing mode now as I am only testing one of the features for double tap.
* The UI allows the user to double tap their fingers in game to change the gear and make the car go faster.
* The terrain in the game is in 3D.
* The AI other driver is set currently to a max speed, will change later.
* Now the car can either win the game or lose.

Update as of 4 weeks:

* I’ve got 2 different gestures working now, the double tap still working for the change gear mechanic and now there is a boost function when user clenches fist.
* This allows the user to gain a more advantage.
* Currently working on menus.

**Gestures**

Before the game starts, there is a splash screen describing the various gestures the user can use.



Since this game is a drag racing game, the player only goes in a straight line, knowing when to change the gear and apply the Boost is key in order to get a good time.

The reason behind choosing the double tap gesture is it is the most recognizable gesture that the Myo SDK can pick up. After some testing, personally the 2 easiest gestures to recreate at a rapid time movement are the Double tap and the Fist. Changing the gear at the right time is key to winning the game and I required an action the user could recreate many times and could be detected very easily.

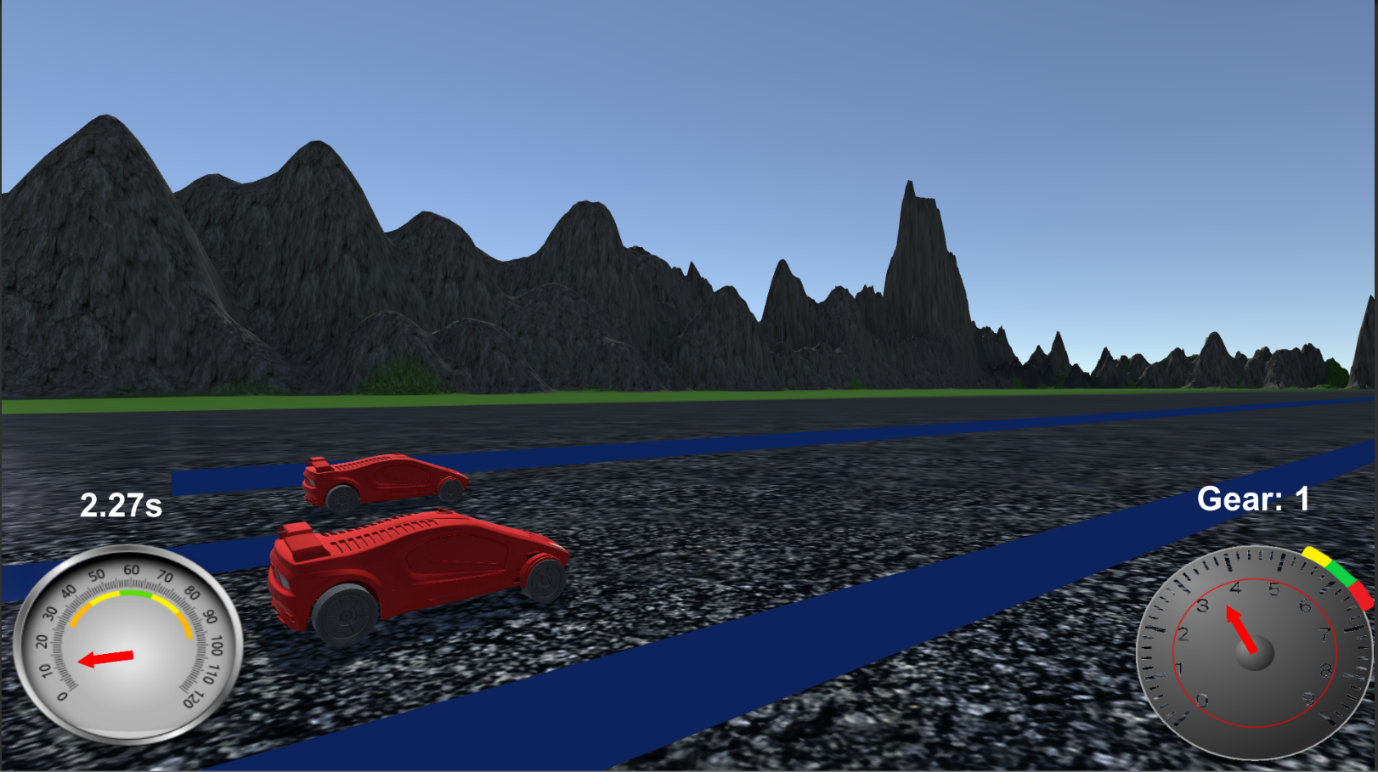
The boost function allows the car to travel at a 2x speed difference to the previous speed allowing the car to move at a more rapid speed. This is done by initially lowering the default traction control in the 2 cars (the players and ai’s car) and setting the traction back to a higher value after the user applies the boost gesture.

**Running the game**

Initially when the player loads the game, they are presented with a main menu screen where the user can select from 3 different options: start the game, options and exit. the start game loads up the next scene which is the how to play screen. this screen gives a detailed description on the specific mechanics of the game and the controls required to play the game. the options menu will display the main volume controller and the exit button will close the game. there is also a Pause menu option in the game which will display similar menus: resume game, main menu and exit.

Above I have included some screenshots of the main menu screen and the options menu. These menus are loaded up initially when the player enters the game.

Below is a screenshot of what the game currently looks like. After a countdown is displayed counting down from 3, 2, 1, GO the player and the AI move forward. The speedometer in the bottom left of the screen indicates the current speed of the car. The tachometer on the right indicates when the player Is required to change the gear. Ideally changing gear when it hits green. There is also a timer displayed on the bottom left and a gear indicator on the right to let the user know when they changed the gear.



For the pause screen below, it darkens the overall screen being displayed indicating the game is paused, this pause function will pause any movement and sound in game. The options given are resume the game, go to the main menu or exit the game. 

If the user wins or loses the game a screen will pop up displaying either.



This user can then exit the game once they are done playing by pressing the escape button and selecting exit.

**Conclusion**

Using the Myo SDK with Unity 3D has been a great learning experience. Having had experience with Unity 2D crossing over to using the similar 3D environment was surprisingly achievable. Myo’s SDK was similarly easy enough to use, the little YouTube videos and blogs available on the installation was enough however more resources would have made it easier. Overall this project has given me great experience with using a gesture based user interface.