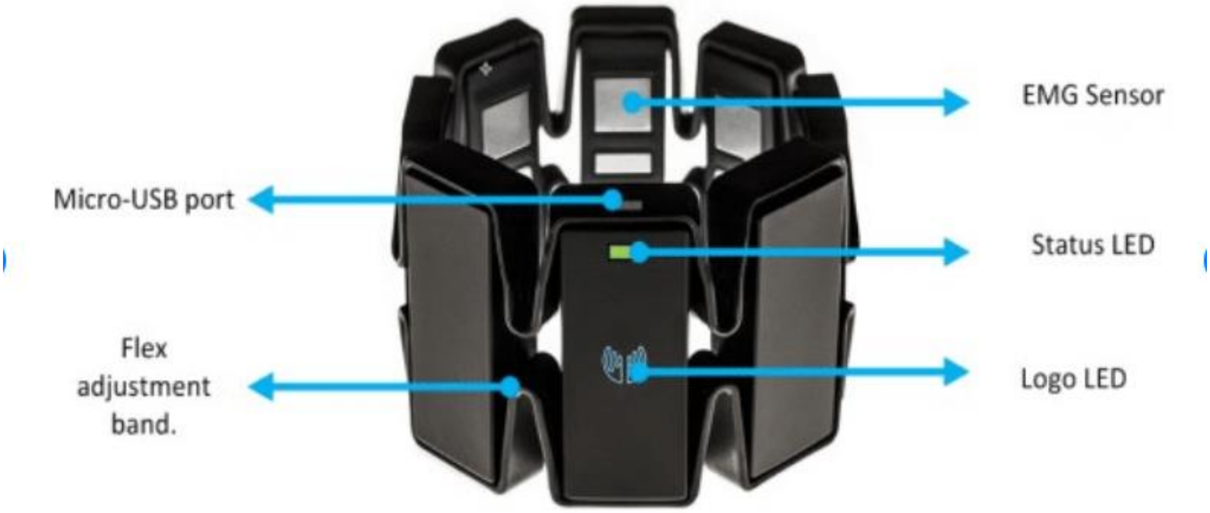
Gesture Based User Interface

By

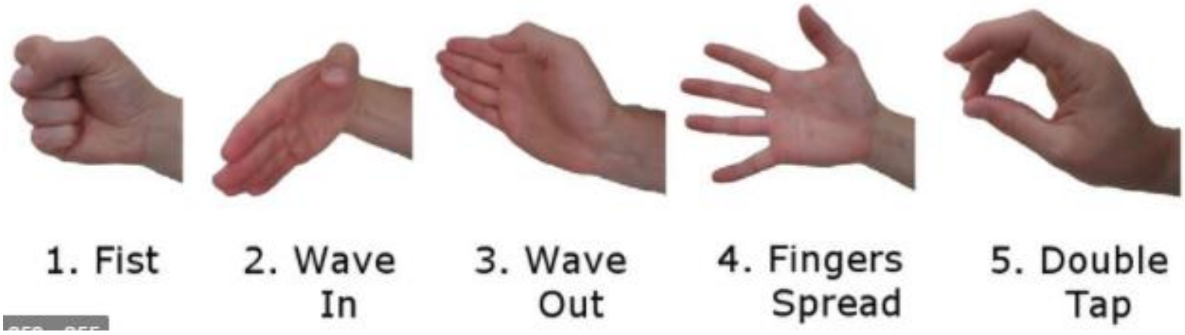
Kevin Flanagan G00228079

Given the task of building an application that can be controlled with gestures, I decided to use a MYO Armband. The MYO armband [fig 1] is a gesture-controlled device that uses sEMG signals in the forearm to detect the electrical signals of the muscles. These muscle movements are then turned into gestures that are used to control the application. The software can be downloaded here www.myo.com/start and will show you how to get started. The MYO armband from *Thalmic Labs* lets you control different applications with the flick of the wrist.



[fig 1 MYO Armband]

Data acquisition in one of the first stages of setting up and detecting sEMG (Electromyography). This involves different gestures performed by the user [fig 2]. This will help configure the armband to the user and give them a sample of how to use it.



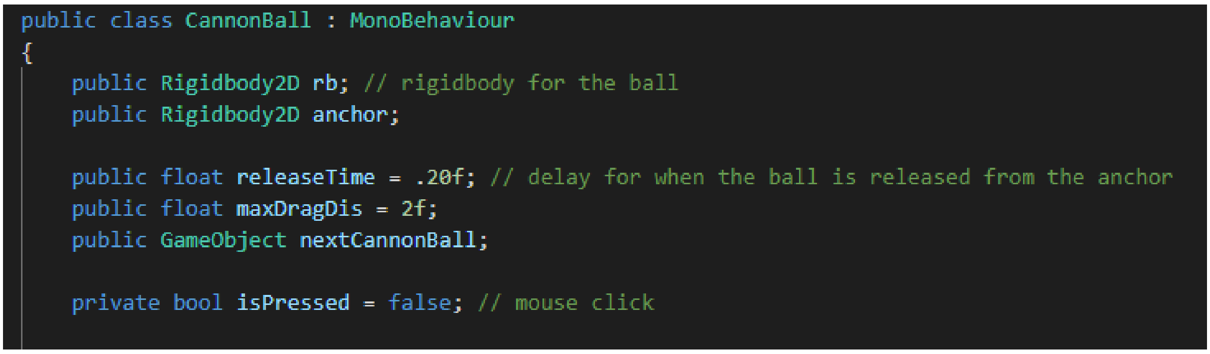
[fig 2 Gestures]

After setting up my MYO armband, having decided to create a game and use the armband to control the game, I began creating my game in unity. This game will be a simple game of destroying the enemies with a cannonball to move onto the next level. I have created three levels which get harder as the player advances through the levels.

Apart from the MYO armband other or alternative devices for gesture-based systems are the *MioConnect* or the *Microsoft Kinect.* I do feel that the MioConnect would be more in common with the MYO armband when compared to the Kinect.

Cannonball Game

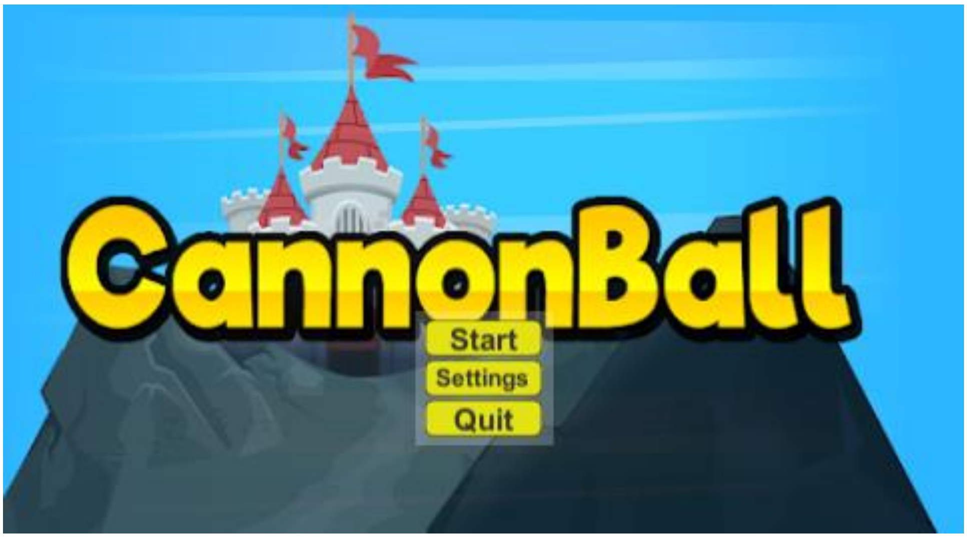
I created a simple angry bird's clone and tweak game; the player must destroy all the enemies to progress onto the next level. I used different physics methods to get a ball on a string type movement that once pulled back and released will carry on knocking over the structure and destroy the enemies.



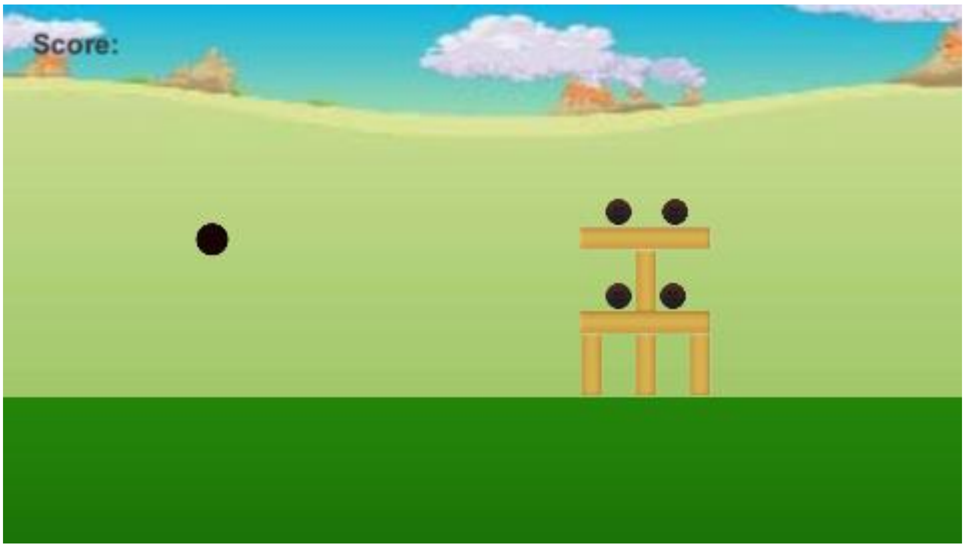
```
public class CannonBall : MonoBehaviour
{
    public Rigidbody2D rb; // rigidbody for the ball
    public Rigidbody2D anchor;

    public float releaseTime = .20f; // delay for when the ball is released from the anchor
    public float maxDragDis = 2f;
    public GameObject nextCannonBall;

    private bool isPressed = false; // mouse click
```

[fig 3 Anchor]

I used an anchor to create this effect [fig 3]. I also setup a way to keep the score when an enemy is destroyed. I created a main menu [fig 4] below. The first level [fig 5] is simple and easy for the player to complete, I also added in three cannonballs giving the player three shots to complete the level. Each level is connected and once the enemies have been destroyed the next level/scene loads.



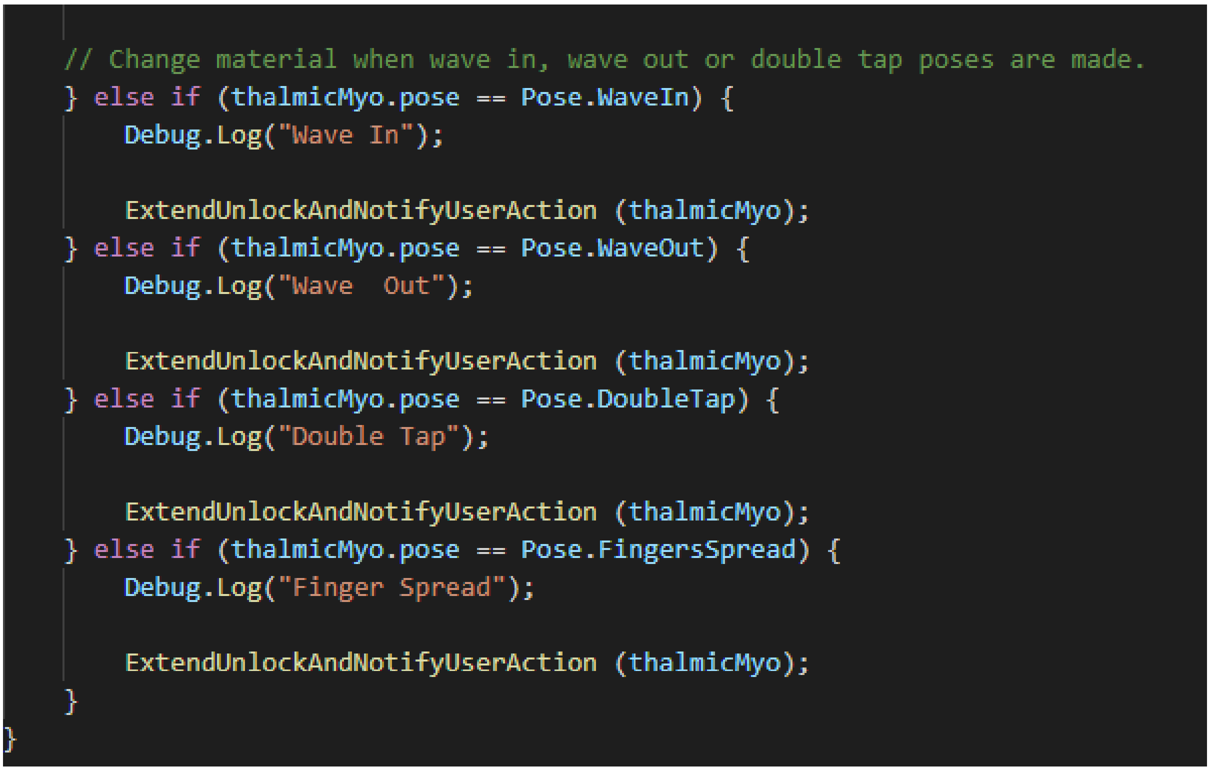
[fig 4 Main menu]



[fig 5 First level]

The first level was setup with six enemies that need to be destroyed to move onto the next level which has eight enemies and finally the last level with twelve enemies.

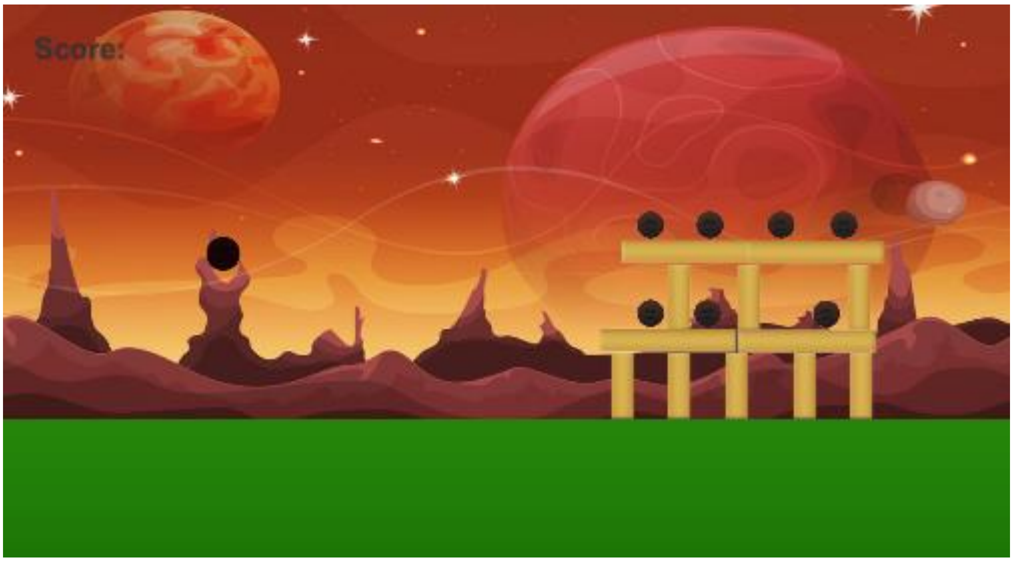
Some of the background code I had to set up and use for my own purposes [fig 6], this allowed me to set up the gestures and allowed me to control the application.



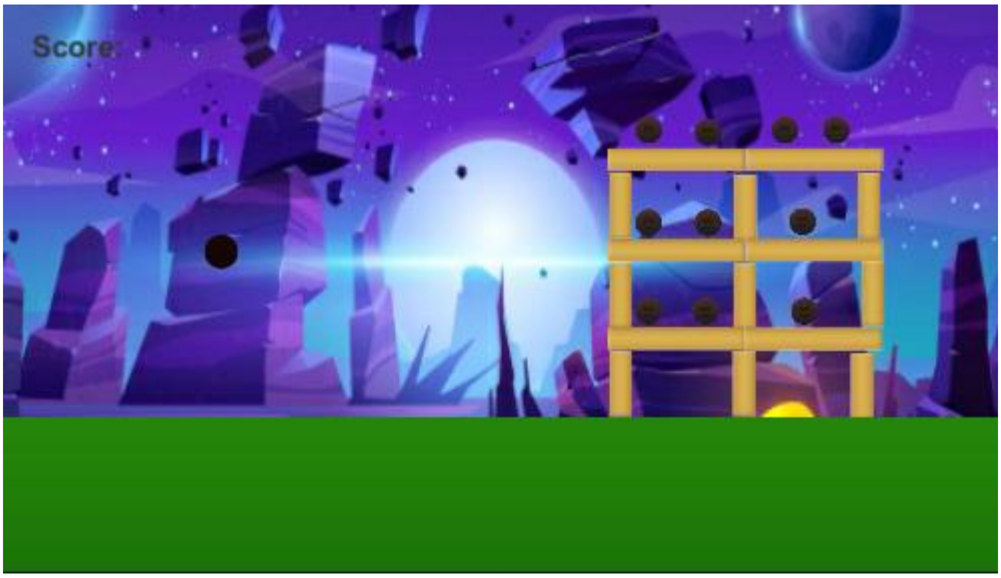
```
    // Change material when wave in, wave out or double tap poses are made.
    } else if (thalmicMyo.pose == Pose.WaveIn) {
        Debug.Log("Wave In");

        ExtendUnlockAndNotifyUserAction (thalmicMyo);
    } else if (thalmicMyo.pose == Pose.WaveOut) {
        Debug.Log("Wave  Out");

        ExtendUnlockAndNotifyUserAction (thalmicMyo);
    } else if (thalmicMyo.pose == Pose.DoubleTap) {
        Debug.Log("Double Tap");

        ExtendUnlockAndNotifyUserAction (thalmicMyo);
    } else if (thalmicMyo.pose == Pose.FingersSpread) {
        Debug.Log("Finger Spread");

        ExtendUnlockAndNotifyUserAction (thalmicMyo);
    }
}
```

[fig 6 MYO Armband code]

When I created my players avatar(cannonball), I attached my MYO movement script to the cannonball for the MYO armband to pick up the users' gestures. This will allow the player to control the angle and power needed to fire the cannonball. The levels get harder with more enemies and more structures to destroy [fig 7]. Below is the design of levels 2 & 3 where it gets harder for the player to win the round. Unfortunately using the Myo armband I could not click and drag the cannon ball, so I decided to add the free roam. This allows the player to just have some fun with the cannon ball which now becomes a wrecking ball that can be controlled using the MYO armband and I will show this in my screen cast.



[fig 7 Level 2]



[fig 7 Level 3]

I really enjoyed using the MYO armband as it was interesting to see what can be done and what could be designed in the future if more money and research is put into these devices. I do find that the MYO armband can become uncomfortable after long periods of wearing it. For testing and practicing for this assignment I was wearing it for long periods of time and found that I could be repeating the same gesture over and over for the MYO to pick up the gesture which lead to some horrible hand cramps.

There is two different ways to play this game (1) is with the mouse and key board where you click and drag the cannonball and it goes flying into the enemies and the Esc button with pause the game, (2) the second way to play this game is to use the MYO armband which makes the cannonball into more of a wrecking ball that the player can control with the movement of their arm/wrist. While wearing the MYO armband the player can pause the game by using the double tap of their fingers and making a fist to resume the game.

Overall, I really enjoyed working with this technology and would definitely be interested in working with it again in the future if the opportunity arose. From my research in this area I've discovered that *Thalmic Labs* have been rebranded *North.* North have moved away from armbands and are working on Alexa-powered holographic glasses.

Screencast link: https://youtu.be/1merq_jvnAY

References:

https://time.com/4173507/myo-armband-review/ Reviews on MYO armband

https://www.youtube.com/watch?v=w6cKglRcpmo&t=982s YouTube where I got some ideas from

https://learnonline.gmit.ie/course/view.php?id=2799 Samples from Moodle and starter code