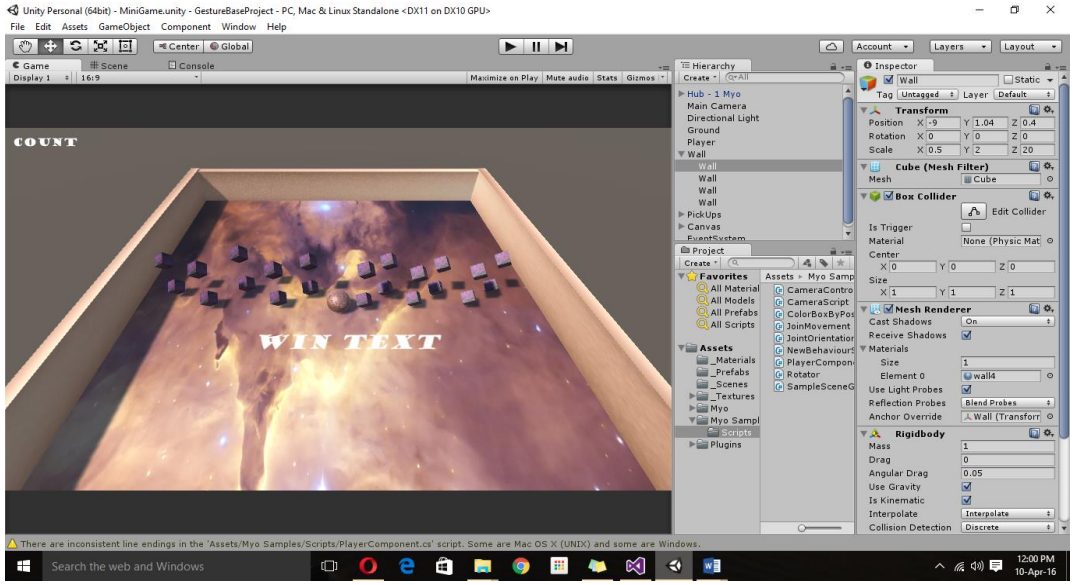
MYO PROJECT

By Alina Danci

Purpose of the application

- Simple 3d Unity Ball game to show the MYO working.
- This rolling ball will pick up spinning game objects using the MYO.



Gestures identified as appropriate for this application

1. Waving left, right and double tap are the three gestures I have working for this application.
2. When you wave or double tap the ball changes colors.
3. Also the MYO is linked to the ball allowing to ball to move as the hand moves.

Hardware used in creating the application

The MYO:

- I chose to use the MYO because it was a new technology
- I was interested in getting to use it
- Even though the MYO had a few disadvantage, for example:
- There was a limited amount of MYO armbands.
- As I never used it before it took more research.
- Time consuming.
- Since its new there wasn't a lot of help on the internet in how to solve problems.

Trail and Error

- Since I didn’t have the MYO from the beginning I created the ball game before.
- I created the game without the SDK inside at first.
- After I got a hold of the MYO I just dragged the SDK in the project and tried to connect it to the game.
- Because I wasn’t using the SDK stick and square object it was harder to connect it.

Code I tried

Here I was trying to read the x, y, z values of the ball that was moving originally. Track them then map them. Unfortunately this didn't work for me.

```
void ballLocation()
    {
        transform.localRotation = new Quaternion(_myoQuaternion.Y,
_myoQuaternion.Z, -_myoQuaternion.X, -_myoQuaternion.W);
        accelerometer = new Vector3(_myoAccelerometer.Y,
_myoAccelerometer.Z, -_myoAccelerometer.X);
        Vector3 eulerAngles = transform.localRotation.eulerAngles;

        Debug.Log("Y:" + eulerAngles.y+ "X:" + eulerAngles.x +"Z:" +
eulerAngles.z); }
```

Code sample

Because my code wasn't getting the MYO to more the ball I decided to add some gestures to see if it connects with the game. I added the wave in, out and double tap to change the ball color.

```
else if (thalmicMyo.pose == Pose.WaveIn)
        {
            GetComponent<Renderer>().material= waveInMaterial;

            ExtendUnlockAndNotifyUserAction(thalmicMyo);
        }
        else if (thalmicMyo.pose == Pose.WaveOut)
        {
            GetComponent<Renderer>().material = waveOutMaterial;

            ExtendUnlockAndNotifyUserAction(thalmicMyo);
        }
        else if (thalmicMyo.pose == Pose.DoubleTap)
        {
            GetComponent<Renderer>().material = doubleTapMaterial;

            ExtendUnlockAndNotifyUserAction(thalmicMyo);
        }
```

Code sample

After a few days of researching all of the code from the MYO sample code I found another way to connect the MYO to the ball and get it to move.

```
 if (updateReference)
        {
            referenceVector = new Vector2(myo.transform.forward.x * 50,
myo.transform.forward.y * 35);
        }
        transform.position = new Vector2((myo.transform.forward.x * 50) -
referenceVector.x, myo.transform.forward.y * 35 - referenceVector.y);
}
```

Finish!