Gesture Based UI Development

Myo gesture control armband

BSc in Software Development – Year 4

G00353420

Jina Kim

GitHub repository:

https://github.com/JinaKim77/MyoUnityGameProject_2021-.git

Contents

Developed using Unity version: 2019.4.11f1

- Fist pose used to shoot
- Spread Fingers pose used to shoot continuously
- Wave out pose used to cycle up through weapons list
- Wave in pose used to cycle down through weapons list
- Double-Tapping fingers pose used to throw a grenade

Game overview

This application has been developed as part of the Gesture Based UI Module, working alone to build a game controlled using a MYO armband. Using Unity and C# to create a 3D zombie shooter game that allows you to traverse a spontaneously created environment and shoot obstacles and zombie enemies with a variety of weapons.

Purpose of the application

The purpose of the application is to allow the user to control a gun weapon with the Myo Armband while shooting zombie enemies and to demonstrate the use of gesture based UI using the Myo Armband.

The purpose of this project was to explore Gesture-Based UI. The technology I chose to use was the Myo Armband, designed by Thalmic Labs. I designed a game where zombies are spawned on screen and you have to shoot them before they attack you or get past you and attack the village. The Myo armband is used to control your aim and to fire shoots as well.

Main menu scene

The user will make a fist to click the button and wave in or out to search buttons.

It's zombie shooting game the with the creepy background music and image will suit for this game.

Game level 1 scene

Game level 2 scene

The instructions for the Myo gestures are displayed on the main game screen.

- The user will get scored when collecting each pickup. (10 scores each)
- The player has 3 lives. That will be displayed on the screen with health bar.
- There are spawn triggers, so when the player enter these triggers zombies will be spawned from different places.
- During the creation of this project I found that using the fist pose for firing the weapons was best and the most natural position.

Dead scene

The user will make a fist to click the button and wave in or out to search buttons.

Game complete scene

The user will make a fist to click the button and wave in or out to search buttons.

Gestures identified as appropriate for this application

The Myo armband responds to a variety of gestures you make with your hand. Gestures include: Fingers spread, wave right, wave left, fist, rotate, double tap.

To take aim at an enemy zombie you simply move your arm in the direction of the zombie.

To shoot at the zombie you have three gesture options:

Make a fist.

Spread fingers

> At first I only had the make a fist option then I decided to incorporate the spread fingers to shoot continuously.
>
> I thought it's just better design when multiple zombie enemies are close to the player.

To change weapons you have two gestures options :

Wave-in

Wave-out

There are 6 different guns for the player to choose from.

It's easy to remember for the player to change different waspons.

To throw a grenade to destroy objects you have one gesture

Double tapping fingers

Hardware used in creating the application

The Myo armband designed by Thalmic Labs was the main hardware used for this project. The armband allows for hands-free control of a wide range of technology. It enables the user to control technology wirelessly using various wrist and forearm motions. It uses a set of electromyographic sensors that sense electrical activity in the forearm, combined with a gyroscope, accelerometer and magnetometer to recognize gestures.

I found the Myo armband problematic at times.

Although when changing poses(gestures) within the game it seemed responsive, but I had a lot of problems when syncing the myo with my laptop and the responses sometimes didn't seem to work.

- Unity
- Visual Studio Code
- Thalmic Myo SDK 0.9.0
- Myo Unity plugin

Game play and Controls

Character movement is a mixture of keyboard input and Myo armband.

- Up arrow or W key: To move the game character in forward direction
- Down arrow or S key: To move the game character in backward direction
- Left arrow or S key: To move the game character in left direction
- Right arrow or D key: To move the game character in right direction
- Myo Armband with player arm rotation: To control the aim direction. This will turn the game character to face this enemies
- Esc key : To pause the game

Conclusion

Although I really like the idea of using the Myo armband, I wasn't so convinced by its performance. The gestures weren't always picked up straight away. I even had to reinstall it quite often before using it. I also had trouble with the double tap gesture. That gesture is still in my game but it's not always accurate.

I found that the wave-in and out gestures are quite sensitive. I used these gestures for changing weapons and it moves really quick.

The make a fist and spread fingers gestures are my favourite gestures.

Although I had to over emphasize when make a fist.

I also didn't like the way I had to constantly re-sync the armband or reinstall the myo quite often. The product itself was a nice design and pretty comfortable.

However, developing a game with an armband, it makes my wrist quite tiring and sometimes gives a pain.

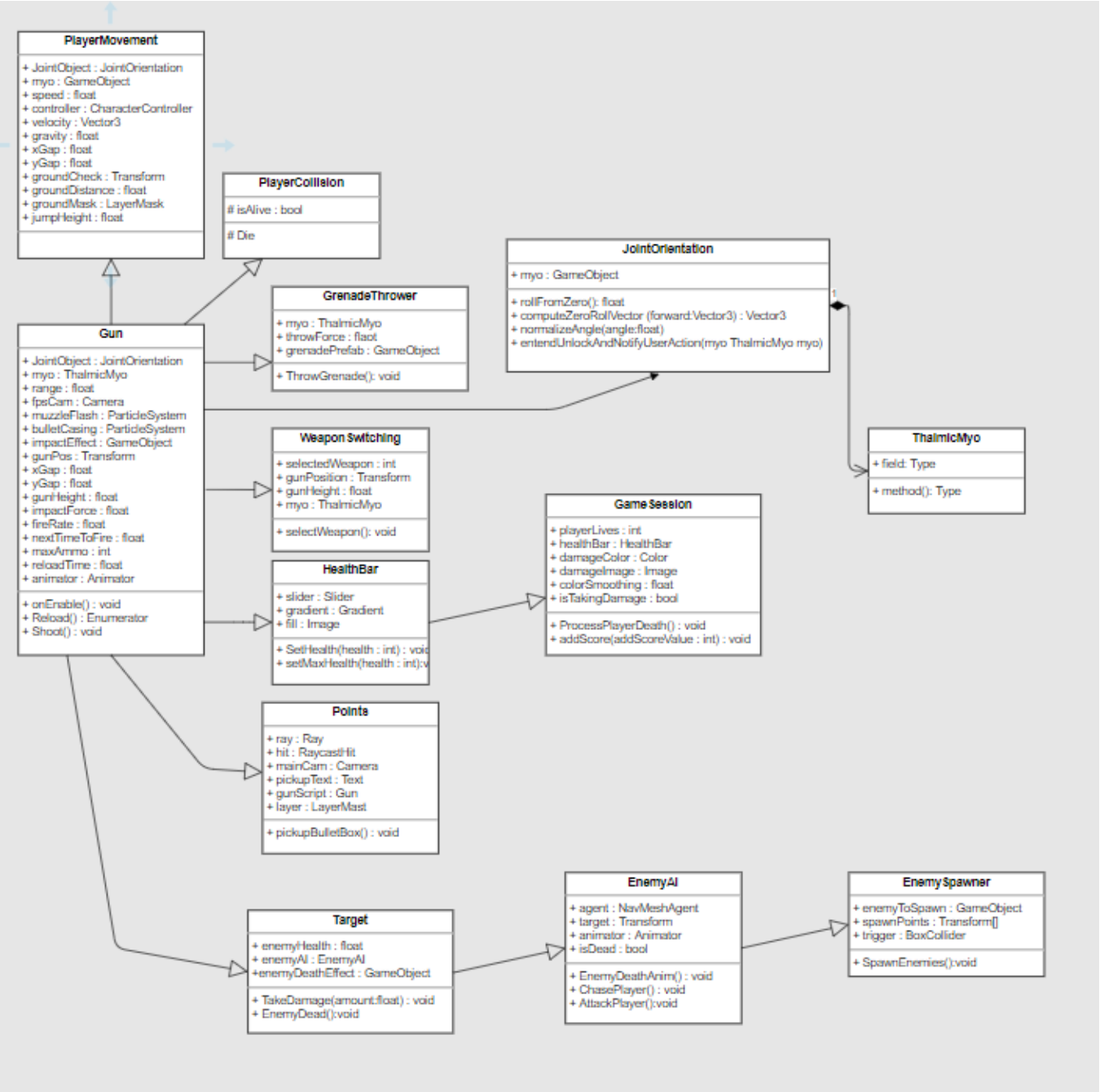
Gesture recognition on Myo works like magic, when it does; but, more often than not, Myo fails to recognise the gestures correctly, and sometimes it doesn't even recognise them at all.

Recommendations

The problem with Myo is that it's limited to just those few gestures(there are only 5 gestures available to use.), a few of which can actually be uncomfortable to perform and it doesn't recognize those gestures correctly 100 percent of the time.

Although sometimes it's difficult to sync with, it's very interesting tool for the unity game. It took me quite a lot of time to figure out how to implement the Myo to my Unity 3D game, I really enjoyed the progress of sorting out the problems I had.

Architecture for the solution



Main scripts for project

- ThalmicMyo & ThalmicHub - These two scripts comes as standard with the Myo SDK and are what connects to the actual armband itself.
- EnemySpawner – This controls the spawning of enemy game object(zombies)
- GameSession– This manages player's health, score, game over and restart game methods.
- PlayerMovement – Controls how the player's movement and aim direction with Myo armband.
- PlayerCollision – This manages player's death condition when it's touching enemy game object.
- WeaponSwitching – This manages switching weapons for the player.
- Gun – This manages for the player gun to shoot the enemy game object with the Myo armband gestures.
- JointOrientation – Controls how the player aims.
- Target – This manages enemy's health and death conditions
- EnemyAI – This controls the enemy movement. Enemy will follow the player game object.
- AudioManager – This manage multiple sounds list.
- ZombieAnimator – Animator to create zombie walk effect.
- Crosshair – Rotate the crosshair
- Pause – To pause or resume the game with ESC key.
- Points – This manages score system.
- SceneGUI – Displays myo instructions
- GrenadeThrower – Throws grenade with the Myo armband gesture.
- MouseLook – Follows player game object
- LevelControl – Manages loading levels.
- CanvasController – For the menu scenes.
- LoadOnClick-For the buttons in menu scene.

References

Thalmic Labs; https://developerblog.myo.com/author/thalmic-labs/

developerblog.myo; https://developerblog.myo.com/

A gesture Control Revolution with Myo Armband;

https://www.thecoolist.com/thalmics-myo-armband-brings-gesture-control-whole-new-level/

15 things you can do with Myo;

https://medium.com/thalmic/15-things-you-can-do-with-myo-f3f05bcc6f9c

Myo gesture control armband review;

https://newatlas.com/myo-gesture-control-armband-review/39103/

5 awesome ways the Myo gesture control wearable is being embraced;

https://www.wareable.com/wearable-tech/myo-armband-uses-gaming