




MARCH 21, 2020

# TRADITIONAL SNAKE GAME WITH MYO ARMBAND INTEGRATION

GESTURE BASED URI DEVELOPMENT

JAMES NELLY & USMAN SATTAR  
BACHELOR OF SCIENCE (HONOURS) IN SOFTWARE DEVELOPMENT  
Galway-Mayo Institute of Technology



## Table of Contents

Introduction .....	2
Research Ideas.....	2
Game 1 .....	2
Game 2 .....	2
Game 3 .....	3
Purpose of the Application .....	3
User Interface Screenshots .....	3
Gestures Identified as Appropriate for This Application .....	4
Hardware Used in Creating the Application .....	5
Myo Armband .....	5
Architecture for the Solution .....	6
Design .....	9
Game Assets.....	9
Control Mechanisms.....	9
Front-End .....	10
In-Game Menus.....	11
The Game .....	12
Test Plan .....	13
Conclusions & Recommendations.....	14

## Introduction

We have taken the traditional snake game and recreated it using modern technology to meet the standards with today's trends. Initially we created the classic snake game which will be further explained below and have integrated Myo armband with the game. Instead of using arrows on our keyboards to control the mouse, we are now controlling the movement of the snake in the game with the gestures of our hands. Lastly, we will be talking about how we can improve on what we are developing and the difficulties we came across while constructing the application in the conclusion and recommendation section.

## Research Ideas

### Game 1

The first game we researched was called **Fruit Ninja**. Classic fruit ninja involves slicing fruits with the mousepad or touch on touch screen devices.

- We wanted to use Xbox Kinect to transform classic fruit ninja to a whole new level.
- The camera of the Kinect would've detected arm movement.
- Basically, users arm movement would've been the knives to slice the fruit.

s



### Game 2

The second game we researched upon was **Penalty Shootout**. Like in FIFA we play a round of penalty shootout based on the control on the controller. We wanted to create a simple game of one player having a penalty shoot round using an Xbox Kinect.

The Kinect would have detected our movement with its camera and infrared. Instead of using controls we wanted the camera to detect our leg movement.



### Game 3

The third game that we researched was the traditional snake game which we have developed. We took the traditional snake game and transferred it into a modern-day snake game. We used the Myo armband which connects with the device with Bluetooth. Instead of using arrows on the keyboard, we are now using the gestures of our hands to control the snake.

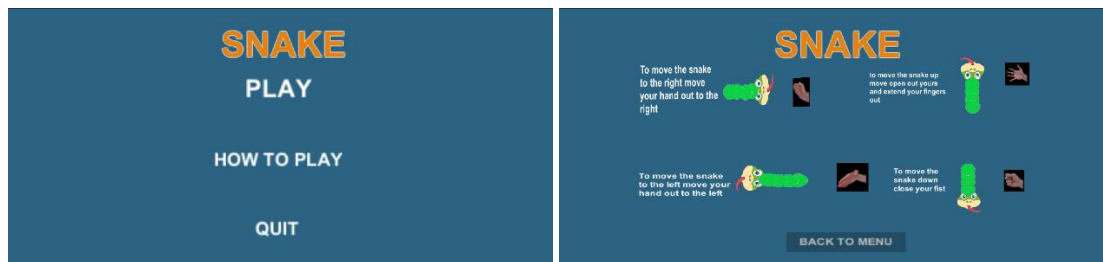
The way this person in the picture is controlling the kart with his wrist, the exact same way we are controlling the snake with this type of gesture. Bear in mind this is one direction, there are different gestures to implement different controls.



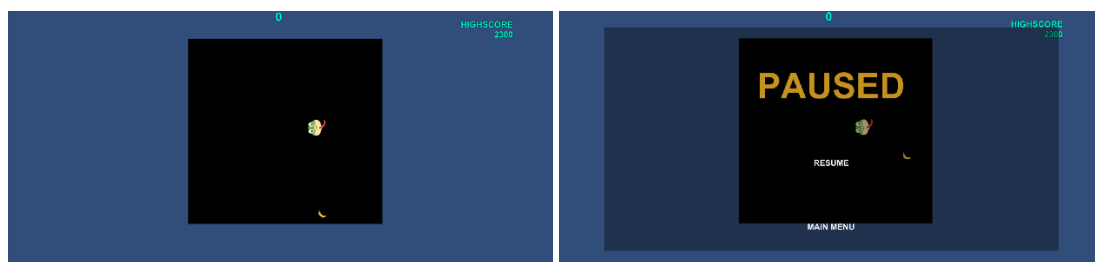
## Purpose of the Application

### User Interface Screenshots

The application is called Snake and it is created on unity. We implemented the software and hardware of Myo. Traditionally snake is played with arrows on the keyboard. We decided to implement the Myo armband and its software, so the game is created to be controlled from the gestures of the user.



Starting the game on unity and making sure that the Myo armband is connected via Bluetooth. If accordingly, to plan, the Myo armband is programmed correctly, it should sense the impulses from the user's muscles to carry out gestures like "wave right", "wave left", "double tap", "fist", "open hand".



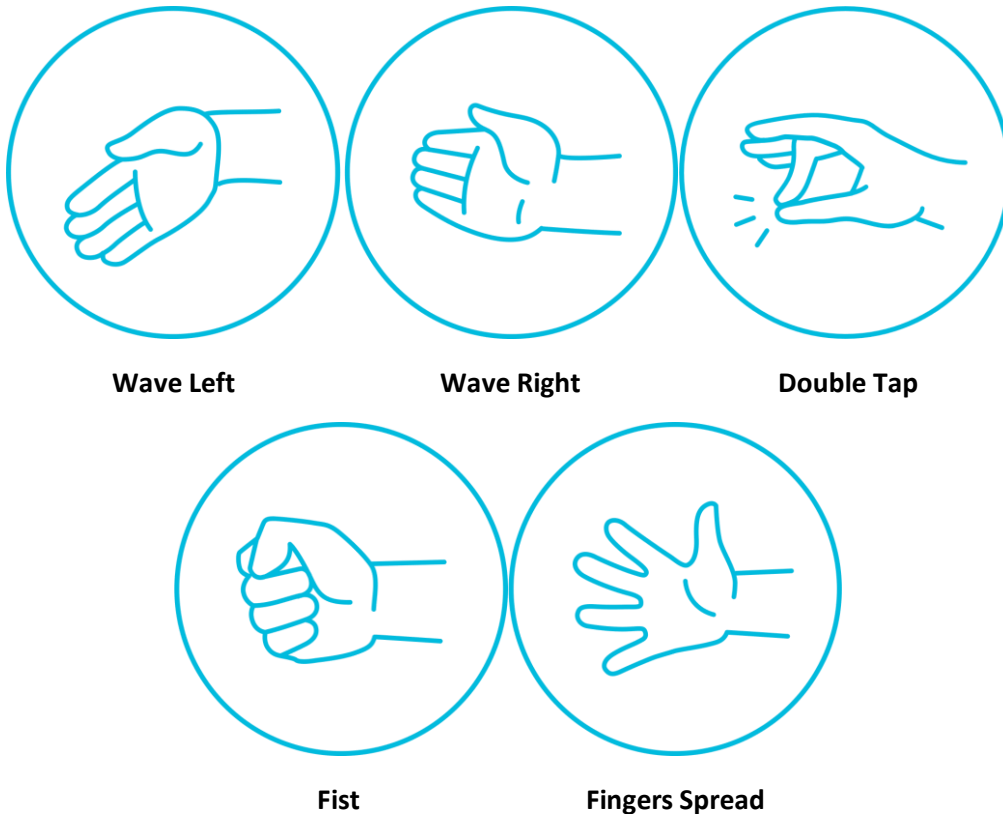
Gestures Identified as Appropriate for This Application

## Hardware Used in Creating the Application

### Myo Armband

There are various reasons behind choosing the Myo armband. The Myo armband measures electrical activity from your muscles to detect five gestures made by your hands. The Myo armband transmits this information over a Bluetooth Smart connection to communicate with compatible devices.

The Myo armband recognizes 5 pre-set gestures. They are:

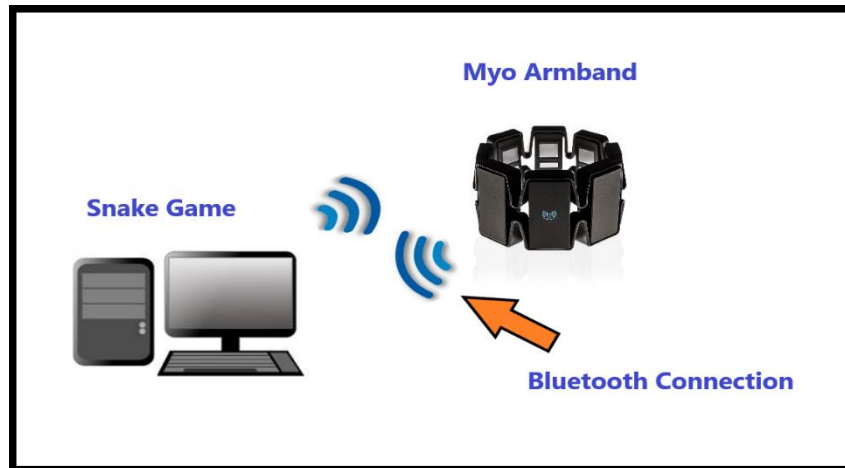


Developers can combine these pre-set gestures with arm motions (data from the Inertial Measurement Unit (IMU) to create new gestures. This gives the potential for hundreds of different gestures. They also have access to raw EMG data from the Myo that can allow them to create their own custom gestures. Myo Connect software can be downloaded from their official website where Myo SDK and Myo Firmware is also available.

We felt like using Myo armband was the best way to bring back a classic childhood game back into our lives in a unique way. In addition, we really love the way the Myo armband works and it give us an opportunity to learn more into it.



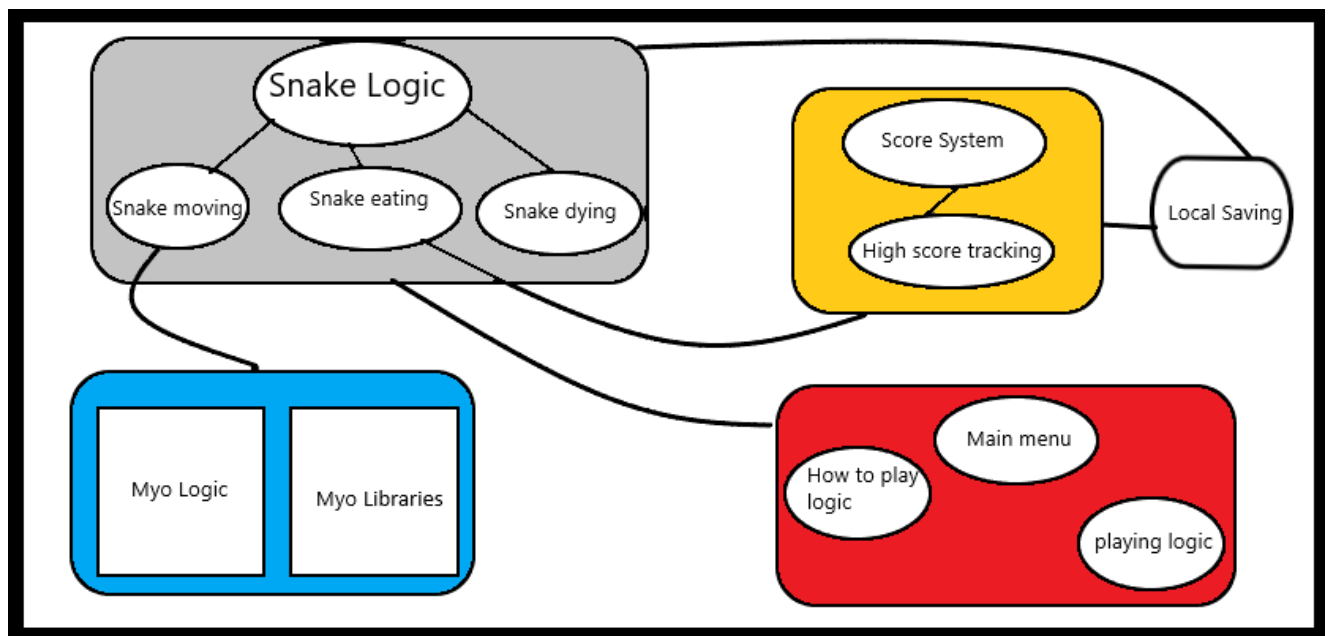
## Architecture for the Solution



## The Background of the Architecture

- We used the Unity Game engine to develop our game. We First developed the game in unity, and later we then connected the game to the unity armband to carry out the various gesture movements so that the snake was able to move around the grid and able to eat the fruit and in turn grow.
- There are many components in the project with then Snake Logic and also with the connection of myo. We incrementally added code to the snake Logic on a weekly basis, also there is many of the lines commented to make sure that there was no confusion as to what was being done.
- The Snake class controls much of the snake logic in the game. Whether it being his movement on the grid that was implemented with the keys first and then was gradually developed for the myo gestures. The snake class controls the eating of that food by the snake and its growth. The death of the snake is implemented in the snake class as when his head hits any part of his body the snake immediately dies.
- The Menu class controls the main classes of the application, you have the how to play button which brings you to a screen which gives instructions about the game. Also, the Play button which is the most important boutons which starts the game
- The Scoring System works like this try to beat the high score every game, Once you die and have beaten the high score it is saved locally in the app, It will appear at the end of the game on the screen showing your high score.

## Architecture





## Class Diagram



## Libraries that were used in unity

- using LockingPolicy = Thalmic.Myo.LockingPolicy;  
using Pose = Thalmic.Myo.Pose;  
using UnlockType = Thalmic.Myo.UnlockType;  
using VibrationType = Thalmic.Myo.VibrationType;  
using UnityEngine.UI
- The Libraries were easily found when searching through google and the implementation of the libraries took some time to wrap my head around but there was plenty documentation on them

## Design

### Game Assets



**Fruit**



**Snake-Body**



**Snake**

### Control Mechanisms

- The mechanisms that are in this game include the user to move the player within the game using actions of their fists.
- The player (snake) can move left, right up and down using arrows on the keyboards or fist gestures as explained above in Myo armband section.
- The snake should be able to move in any direction, when snake eats a fruit, one snake-body should be added on to the snake at a time.

- The snake dies when it gets too big and wraps around itself.
- Scores increases as snake eats more and more fruits, but it is reset to 0 when snake dies or game restarted.
- The most important mechanism in this game is that the snake goes around eating fruit and while the snake is eating fruit, it grows.
- When the snake dies you will be brought back to the main menu

**There are no levels in this game. This is a single level game with a high score.**

#### Front-End

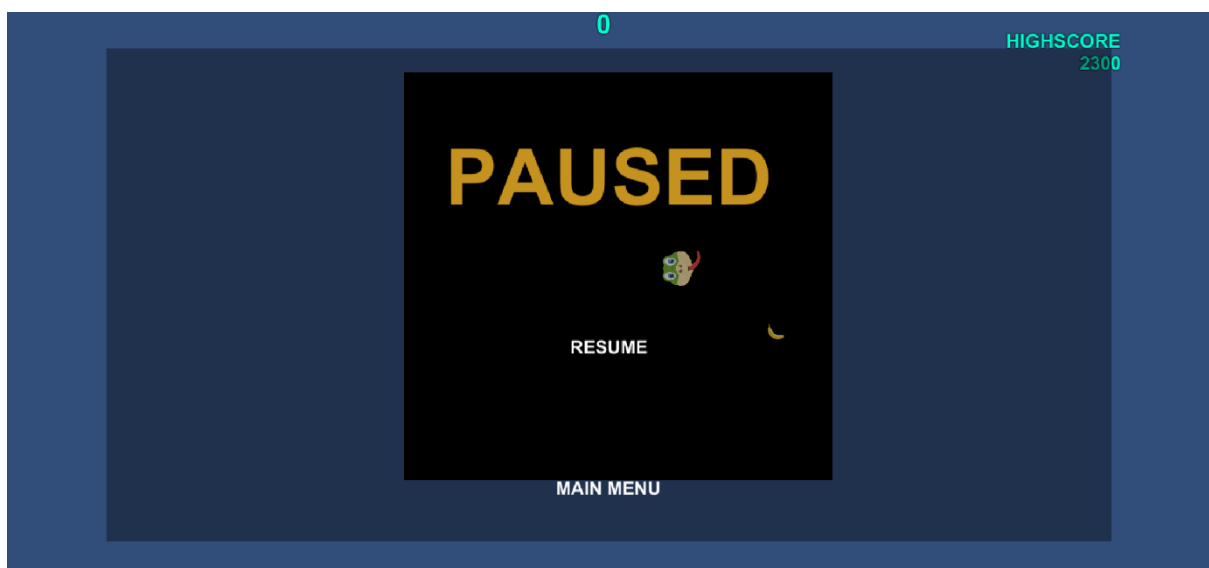
- The first screen you see will be the title screen, it will have 3 inputs.
- Player can choose between **play**, **how to play**, or **quit** the game.
- The first input you can choose from is the play button to start the game. This will begin the game.
- The second input will be “**How to Play**”, it will bring user to an instructions page on how to play the game with the Myo armband.





### In-Game Menus

- There will be a pause menu within the game it should be in the top right-hand corner of the game screen. Once clicked the entire game will pause. In the pause menu you will have one option whether to return to the home menu. Once that button has been clicked you will be returned to the home menu.



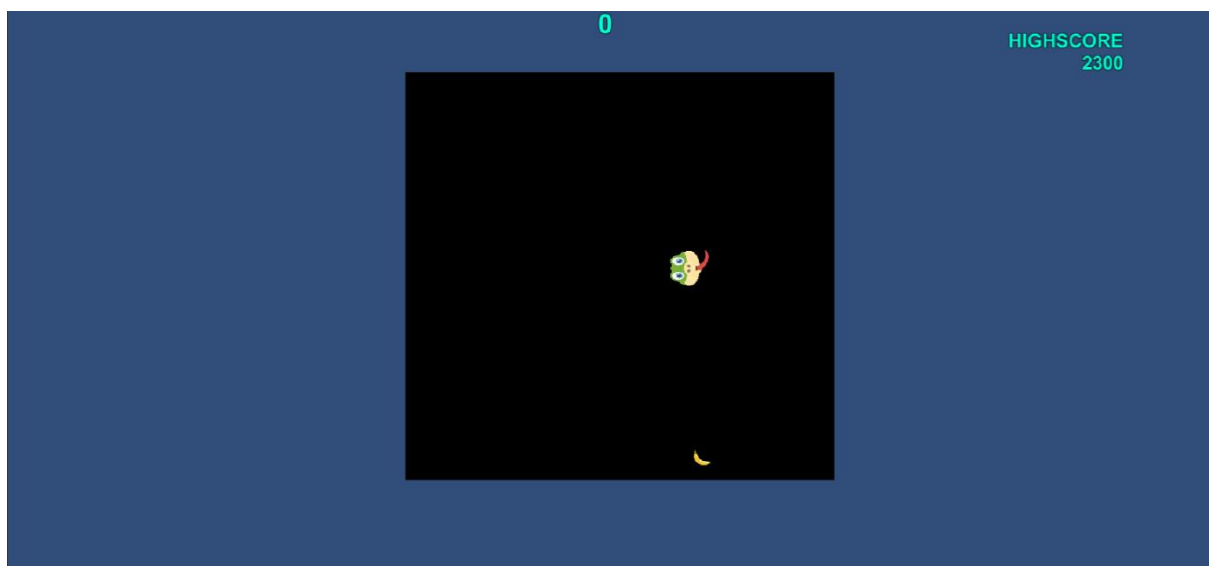
## The Game

Once you click on play in the main menu the game will load. The game has a black, grid-based background. Snake will start with just the head and as it will eat the fruit (which would be randomly generated on the grid), the snake will grow and get longer. You can control the movement of the snake with the movement of the wrists, once the Myo armband is worn. As the snake gets bigger by eating the randomly generated fruit, the high score will keep increasing accordingly.

Snake: **+1 Health**

Death: **Touching (Wrapping around itself)**

High score: **Implemented**



## Difficulty of Game:

This game is not based on levels, it is based on high score. There are no levels of difficulties in this game. It gets difficult as the snake grows bigger and bigger, where it will be hard to control the snake then. The snake within the game has only one life.

## Test Plan

Test Case No	Test Case Name	Parameters	Checks	Expected Result	Actual Result
1	Game Launch	TBD	If the game will launch or not.	The game should launch	PASS
2	Pause Menu	TBD	If the game will pause once pause button clicked	The game should pause	PASS
3	Snake Movement	TBD	If the snake moves without glitches.	The snake should move smoothly	PASS
4	Snake Damage	TBD	If snake wraps around itself, it should die.	The snake should die.	PASS
5	Myo Armband Connectivity	TBD	The Myo armband should have no difficulties connecting to unity.	The Myo armband should connect when needed.	PASS
6	Gestures	TBD	Gestures should output on screen as programmed	Each gesture has a different output.	PASS

## Conclusions & Recommendations

In conclusion, by conducting this project, we have learned a lot about the technology of Myo armband. We now know how the Myo armband measures electrical activity from your muscles to detect five gestures made by your hands by the usage of EMG sensors. Myo uses a 9-axis IMU, it also senses the motion, orientation and rotation of your forearm.

<https://support.getmyo.com/hc/en-us>

<https://github.com/jamesnelly/Gesture-Based-Snake-Game>

<https://developerblog.myo.com/setting-myo-package-unity/>

<https://github.com/thalmiclabs/myo-unity/tree/master/project/Assets/Myo%20Samples/Scripts>

<https://noobtuts.com/unity/2d-snake-game>

[https://www.google.com/search?q=making+snake+in+unity&rlz=1C1CHBF\\_enIE886IE886&oq=making+snake+in+unity&aqs=chrome..69i57j0l5.9263j0j7&sourceid=chrome&ie=UTF-8#kpvalbx=\\_NiEXqfaDdOV1fAPhZK88A821](https://www.google.com/search?q=making+snake+in+unity&rlz=1C1CHBF_enIE886IE886&oq=making+snake+in+unity&aqs=chrome..69i57j0l5.9263j0j7&sourceid=chrome&ie=UTF-8#kpvalbx=_NiEXqfaDdOV1fAPhZK88A821)