Gesture Based UI Development Main Project

Aaron Moran – (G00356519)

GitHub: <https://github.com/Moran98/pokemon-battle-2D>



Table of Contents

[Introduction 3](#_Toc69577887)

[Purpose of the application 4](#_Toc69577888)

[Main Menu 4](#_Toc69577889)

[Instructions 5](#_Toc69577890)

[Battle Selection 6](#_Toc69577891)

[Battle 7](#_Toc69577892)

[Gestures identified as appropriate for this application 8](#_Toc69577893)

[Hardware used in creating the application 9](#_Toc69577894)

[Voice Commands 9](#_Toc69577895)

[Architecture for the solution 11](#_Toc69577896)

[Conclusions & Recommendations 12](#_Toc69577897)

# Introduction

We have been tasked with creating a Gesture Based game using either Hardware or Voice Controls to perform the functionality of the game. I have decided to create a clone and tweak of the battle sequence in the Pokémon game series. I have used what I have learned over the past year to incorporate the Grammar Recognizer functionality and applied a State Machine so that it determines the sequences throughout the battle.

The game has a character battle selection that is completely controlled by voice commands which then will load the enemy’s Pokémon and you have the option to either Attack / Heal / Flee.

# Purpose of the application

The purpose of this application is to demonstrate the use of voice commands in a state defined battle sequence.

## Main Menu

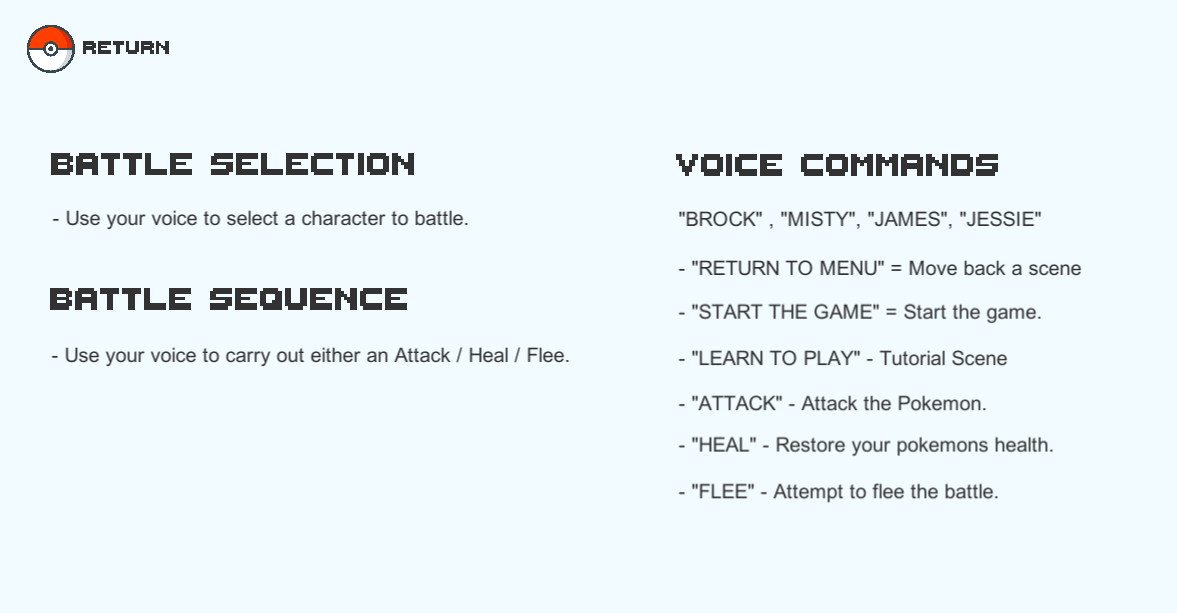
In the image below this is the screen that the user will be greeted with. The player can choose either to start the game or learn the games mechanics and commands. Either option is controlled by voice commands.



(Fig 1.1. Main Menu)

## Instructions

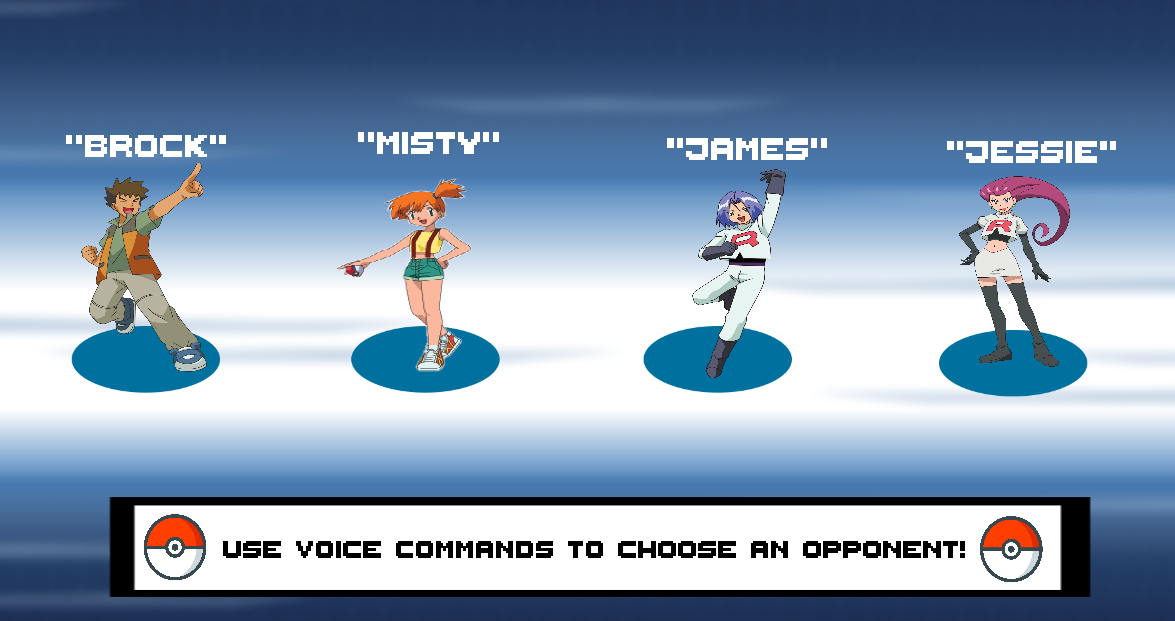
The Instructions scene consists of the voice commands and instructions for the Battle Selection scene and the Battle Sequence scene. The commands are basic as it means the game is easy to play which makes the user experience smooth and enjoyable.



(Fig 1.2. Instructions Scene)

## Battle Selection

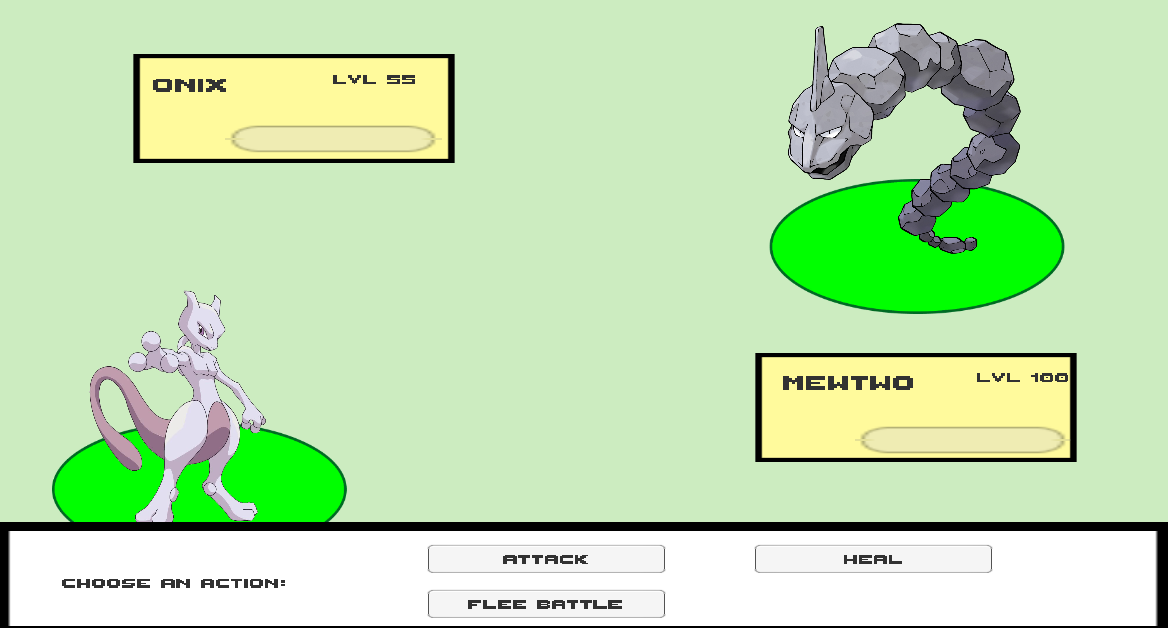
In the image below you can see that there are four opponents to choose from to battle. Each opponent has a different Pokémon who has different health and damage attributes. You must use voice commands to choose who to battle. Once an opponent has been chosen the battle will commence.



## Battle

Once the battle begins you will be prompted to choose an option between – ‘ATTACK’, ‘HEAL’ or ‘FLEE THE BATTLE’. Once you have chosen an action whether it be ‘ATTACK’ or ‘HEAL’ the action will be performed and the state will change to the opponents turn.

If you choose to ‘FLEE THE BATTLE’ this will select a random number between 1-10 and if the number is greater than five and less than or equals to ten you will be unable to flee. Once your opponents Pokémon reaches zero health the state will change to ‘WIN’. If your Pokémon’s health reaches zero first the state will change to ‘LOSS’.



# Gestures identified as appropriate for this application

I researched the use of VR for the implementation of this game style and the possibilities are quite impressive. The ability to use the VR headset to interact with the battle and choosing attacks with hand gestures would make the experience a lot more intuitive.

Movements such as looking at your wrist to check your ‘Pokedex’ would be a suitable feature

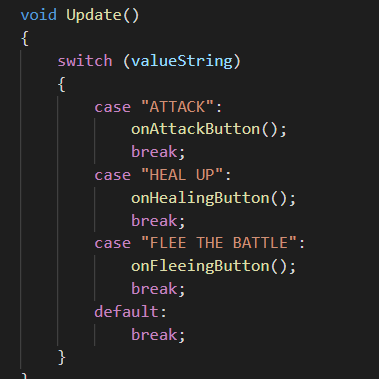
No doubt we will see an official Pokémon release on VR in the near future.

# Hardware used in creating the application

For this application the only feature that I have implemented and utilized are the Voice Recognition controls. I have a set file of grammar which is used throughout the game and it is loaded once the game has started.

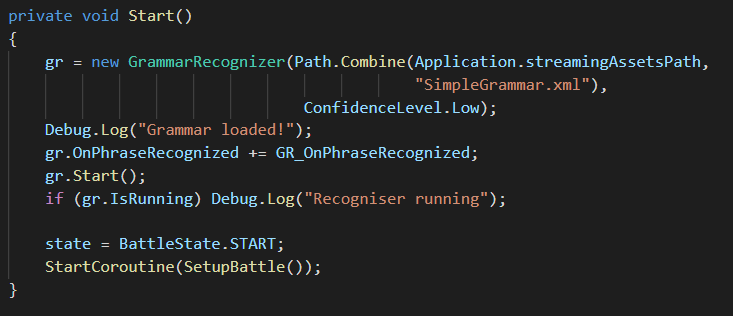
## Voice Commands

* By using voice commands, I have a switch statement setup to perform the actions depending on the recognized phrase. Each of these methods will carry out a function and then change the games state. Below is an example of the battle sequence phrases.



(Fig 2.1. Switch statement using the Grammar Recognizer to perform actions.)

* Once the game has started the Grammar is loaded into the application. Along with the Grammar loading in, the state of the game is set, and the battle gets configured.



(Fig 2.2. Loading in the Grammar file and setting up the battle.)

# Architecture for the solution

# Conclusions & Recommendations

Overall, I am happy with the outcome of this project as I utilised material that I was taught over the past year between Voice Recognition and State Machines. From this project I have learned that making things complicated will only slow down the development process, so flattening my idea out to its most basic components with the Implementation of Voice Controls to navigate and perform actions in the least amount of commands was the best way to achieve a smooth and enjoyable user experience.