

Gesture Based UI Project
Richard Cooke & Cian Doyle
April 2020

<https://github.com/CookeRichard94/Gesture-Based-UI/>

Contents

Introduction	3
Project Overview	3
Reason for choosing this project	3
Implementation.....	4
Original Conception	4
Speech Recognition	4
Research	5
Further Development	5
Conclusion.....	6
Conclusion (Richard).....	6
Conclusion (Cian).....	6
References	7

Introduction

This project was developed by students, Richard Cooke and Cian Doyle, as part of the Gesture Based UI Development module, taught as part of the 4th year of Computing in Software Development at GMIT.

Project Overview

The project in question is a top down maze escape game, that is a throwback to classic dungeon crawlers such as the earlier iterations of the Legend of Zelda series, however this iteration of the game has been given a sci-fi makeover as opposed to the usual medieval look associated with these games. The game was developed using the Unity platform and utilizes speech recognition as the implementation of gesture-based controls, as per the updated document regarding the project and was originally intended to be controlled using the MYO armband technology. The goal for the player of the game will be to acquire the orb which will be present on the level, before finding their way to the end point of 3 separate mazes, hitting switches to open areas along the way.

Reason for choosing this project

As we had no previous experience in developing gesture-based projects, the first step we made upon the formation of the group, was to discuss where we had most frequently encountered gesture-based mechanics. From this, we determined that aside from our mobile phones, we most frequently encountered gesture-based mechanics when interacting with video games, as such we concluded that we would find our best suited project in this area. Having enjoyed our previous experience developing Unity based projects in the previous semester we asserted that we would use Unity for this project. After this we determined that using gesture-based controls to guide an object through a maze would be ample in proving a robust understanding of gesture-based UI implementation.

Implementation

Original Conception

When we originally conceived the idea to develop a maze-based game, we intended on fulfilling the gesture-based side of the project using eye-tracking technologies. However, after a discussion with the module supervisor it was determined that this may become exceedingly difficult to implement as the movement involved with scanning the maze to find the best possible route would potentially lead to the gesture-controlled character moving in potentially hazardous directions.

After this early dismissal we settled on using MYO armband technology to control the character, although this was a feasible implementation, due to outside circumstances beyond the control of all involved in the development of the project this too was dropped from development.

Speech Recognition

As per suggestion in the project document, development of the game was moved forward using speech recognition software.

Using this technology, several commands are implemented. To control the player object, there are 5 commands. "Right" moves the play on a trajectory to the right, "left" moves the player on a trajectory to the left, "up" moves the player on an upwards trajectory, "down" moves the player on an upwards trajectory and "stop" brings a halt to the players movement.

Further commands are implemented in the menu, "play" or "start" begins the playthrough of the game and "quit" or "exit" exits from the application, these are typical commands associated with gaming main menus and the menu will feature similarly labelled buttons as the original implementation involved development without gesture-based functionality.

With the use of speech recognition, the only hardware that we employed was the built in microphones present on both of our laptops.

Research

As part of the research performed for this project, we checked on what possible speech recognition software we could potentially use, through this research we narrowed the decision down to the Microsoft Speech Engine and the UWP speech recognition system. It was decided that we would move forward with the UWP speech recognition system because it allows for very easy integration to the Unity system.

Through research on this topic we also concluded that users are more favourable to single word commands such as the formerly mentioned directional commands as opposed to using full sentences.

Further Development

As we continued development of this project, we became interested in the idea of how we would further polish and develop our game. A consistent topic that we broached was the implementation of random mazes either models of mazes pulled from an array at random or randomly generated mazes.

Another consideration we had was to introduce a “Pac-Man” like element where characters akin to the “ghosts” present in Pac-Man would patrol along the corridors of the maze and would need to be avoided, as contact with these elements of the game would result in the loss of health and possibly a game over screen.

An additional development which was attempted was the inclusion of a set timer whereupon if the designated time was to run out before the player had reached the end point then the result would be a game over. However, the inclusion of text-mesh pro libraries caused the entire game to crash and the inclusion of this content had to be abandoned as no clear solution was in sight.

Conclusion

Conclusion (Richard)

The overarching conclusion that I reached upon the conclusion of this project was that not all gesture-based controls are fully applicable to every scenario. Although I believe that in this scenario the speech recognition works to an acceptable level, in some areas where we wanted the player to utilize swift “in-and-out” movements we found the slight delay in speech recognition software to limit the potential of this idea. The conclusion that I drew from this was that our original intended gesture control, the MYO armband would be much more applicable in this scenario.

However, I personally thoroughly enjoyed the development of this project as I felt this bared the closest resemblance to gaming development and I can see why gesture-based controls are seen as the way forward for the gaming industry because of the inclusivity it provides.

Conclusion (Cian)

The conclusion that I reached as I finished this project was that getting swift movement from the character using speech recognition is in fact hard to do as after several attempts there was still about a second delay in the character’s movement. This would also lead me to agree with my group partner that the MYO armband would be more applicable to this kind of project.

Regardless of this I enjoy working with the Unity engine and this project was no different as I found learning how to implement speech controls was interesting and the overall development was very enjoyable.

References

- <https://docs.microsoft.com/en-us/windows/uwp/design/input/speech-recognition>
- <https://docs.microsoft.com/en-us/windows/mixed-reality/voice-input-in-unity>
- https://en.wikipedia.org/wiki/Dungeon_crawl
- http://exclaim.ca/film/article/maze_runner_origins_a_history_of_mazes_in_retro_games
- <https://lightbuzz.com/speech-recognition-unity/>
- <https://www.youtube.com/watch?v=7NrovM-pA3Q>