Gesture Based UI Project by Ultan Kearns & Paul Kenny

Ultan Kearns

March 2020

Link to Github

Contents

1 Introduction

This section will outline the premise & design of the project. Please note that these sections may be taken from our github wiki[1]

1.1 Premise

This is a fourth year project made by Paul and I for the module Gesture Based UI Development. In This project we aim to provide the user with the following:

- A UI that looks good and gesture based controls that feel intuitive
- A fun and challenging experience that will help train the users memory
- A blast from the past with a nostalgic feel

1.2 What Is The Project

This project will utilize speech recognition and will be based on the 1978 game Simon[2] which is a memory game where the user has to remember a number of colours and repeat them after being shown them by the game. The game has a simple premise but will become more difficult as the user progresses through harder and harder colour patterns. The patterns will be made up of a series of colours and sounds, and we will have speech recognition for each colour.

1.3 Why We Chose This Project

We chose this project because Paul & I love retro games, we chose this game in particular because it provides a challenging and rewarding experience to the user and can also help improve brain function as it has been shown in multiple studies that some games such as Simon can improve brain function Link to article. This game is suitable for all ages and is very addictive!

1.4 Our Experience With Gesture Based UI

Neither Paul(I assume) or I have had previous experience with Gesture Based UI so this will present a challenge, but luckily thanks to the fantastic lecturers at GMIT, we have learned a lot about different programming concepts so the lack of experience should not prevent much of a challenge.

2 Implementation

2.1 Why voice input?

Simon is a memory based game where the user must memorize patterns of colors and repeat the pattern, so I was thinking to myself why not get rid of the tedium of pressing the buttons and instead use vocal input? Since Simon only has a limited number of colours(red, blue, green & yellow) it was very simple to implement vocal commands that were intuitive, memorable and easy to understand as most people can tell colours apart(my apologies if you happen to be colour blind). I also thought that some people may love to play games like Simon but due to having disabilities they may not be able to interact with a keyboard and mouse which is another reason why I figured vocal inputs would be in many ways superior to physical inputs. Rationale for Commands

For the menus the following voice commands are used:

- quit, exit or close - These commands exit the application were chosen because it is intuitive and expected that these commands would close the app

- Play, start, and game - These commands start the game and are again intuitive as the button displayed to the user literally say ”play game” so pretty much anyone can figure these out
- Developers or credits - These commands show the developers of the application being myself and Paul
- level, change or difficulty - These commands redirect the user to the difficulty screen and allow them to choose which difficulty they want the game to be

All these commands search the string given by the users vocal input for the keyword, for example say I said the following sentence: ”pasta, spaghetti, play” the application throws away the words ”pasta” and ”spaghetti” and starts the game anyway.

2.2 Starting research

To start with I heavily researched many speech recognition libraries in Python and came across this one: Speech recognition library python this library allowed me to use the Google Cloud Speech API which was able to discern what the user had said. This library was a bit of a hassle initially as my Linux computer said that there was no default microphone on the laptop so I had to check all the hardware and everything and ensured it was all working the issue was with me not using the sudo command before running the app. After many headaches setting the library up and installing dependencies such as PyAudio once I got into the coding implementation of voice commands I found the speech recognition library very easy to use and it was very intuitive. Coming up with voice commands

Throughout development of this application voice commands were something I was very worried about, I had to consider many things such as: Were the voice commands I was using intuitive? Were they simple and memorable? Were they detected easily by the voice recognition software? I found this very hard at first but slowly I began to come to the realization that most people will just say what they see on the screen or some combination for example they may say: ”I want to play a game” or ”Play this game” or something including game, that is why I decided to just search the commands inputted by the user for certain keywords pertaining to the app. The voice commands work very well and I have tested the app rigorously and can say the app tends to get my vocal inputs right about 95% of the time, it does seem to detect some words easier than others which is to be expected and it may have issues in loud environments and with certain accents.

2.3 Difficulty

This application has three levels of difficulty, easy, medium and hard. The way difficulty is implemented in this application is via a life system, on easy you have three lives, medium you have two & hard you have only one life, to adjust the difficulty you have to go to the difficulty screen and say the level of difficulty you want for this application. If you get a colour in the pattern wrong the pattern will start over and you will have to repeat the sequence, if your lives run out you will be returned to the main screen.

2.4 Music

The music was recorded by me(Ultan Kearns) using my guitar and I chose random notes for each colour.

2.5 Where to use this app

- In quiet environments - DON'T USE IT IN LIBRARIES THEY GET VERY TRIGGERED
- In moderately noisy environments
- Anywhere there isn't too much noise interference

2.6 Where not to use this app

- In extremely loud places - canteens, college etc. (Also I scared some people by screaming at my laptop in public)

2.7 Notes

- This app requires internet to use the speech recognition library
- This app was made using the kivy graphics library for python

2.8 Graphics Library(Kivy)

Kivy is a graphics library for python[3] that came in very handy for designing the GUI, neither Paul or I have used Kivy before so we thought it would be a good idea to learn the library and use it for this project to diversify our knowledge and programming skills. Kivy is open source and cross-platform which makes it an ideal library to learn to create mobile applications and computer applications.
Initially Kivy seemed very hard to learn but thanks to a fantastic documentation [4] we soon grasped the library and were able to navigate through menus by using voice commands.
I would highly recommend Kivy to any developer to make cross-platform applications in Python as it offers tonnes of

functionality and ensures that our application can be used on Windows, Linux, Android, IOS or Mac.

I feel as though learning this library has taught me an awful lot about how a developer can learn something new be it a new technology, program, or library in a few weeks and how to go about it in a logical manner.

2.9 Game Logic

Implementing the game logic was pretty difficult, I started this project initially by getting the voice commands working and printing them out to the terminal. I started implementing the game logic by first getting navigation working through menus with simple commands then I diversified the commands and ignored garbage commands by using branching conditionals or other garbage in the commands by focusing in on keywords such as: ”game”, ”quit”, ”red” or ”blue” so for example the sentences ”Play game”, ”play this game”, ”start this game” or simply ”game” will all launch an instance of a new game.

Once I had finished getting the navigation working I immediately set about coming up with the board commands which were simple as the game board only consists of four colours(red,green,blue & yellow).

After getting all of the above out of the way I worked on how the game would actually be played, I decided upon initializing an array with four numbers(1,2,3,4) all of which represent a colour(red,blue,green,yellow respectively), then once the user guesses this starting sequence I then append a random number between 1 and four to the array and ask the user to repeat the red,blue,green,yellow sequence with an extra random colour, it is in this way our application mimics it's 1978 counterpart.

Upon beginning the game the sequence is shown through the squares turning white and then the user must repeat the pattern, if the user fails to repeat the pattern exactly the game ends and they will be returned to the main menu.

2.10 Why Python?

Python is a high-level programming language which is extremely useful for making an application quickly and it has a very strong and growing community and a tonne of libraries, it is also widely used in industry(I personally have been asked have I used Python in job interviews), due to the reasons previously listed we chose Python for this application. We have also not used Python much in this course so we thought that it would be a good idea to become more familiar with the language.
We both found Python to be a fantastic programming language and it made it very easy to get the basic skeleton of the application up and running in a couple of weeks. I also highly recommend Python's SpeechRecognition library [5]

2.11 Why Simon?

Simon is an ideal game to implement voice recognition in as it has a limited number of commands that the user can choose(it's basically limited to four colours). It is also an extremely fun game to play and challenge friends with especially if you are intoxicated.
Simon is also a widely popular game which many people love to play, I also believe that some people may want to play the game but due to reasons such as disabilities or lack of hardware eg: the original 1978 game, they are unable to play. I aim to provide this fantastic classic to a new generation of players who may be ignorant of the original and may want a new updated version to play.

3 Finishing Code

4 Conclusion

4.1 What I learned

- I learned a lot about voice recognition and how to use it effectively and implement it correctly
- I learned a lot about Python and it's libraries and the strengths and weaknesses of the language and what type of applications it should be used for
- I rediscovered a gem from the past which is incredibly fun to play and to test your memory and to train your mind to become better at memory(slight digression here but if you're interested in how memory works and how to freak your friends out check out Joshua Foers book Moonwalking With Einstein[6])

4.2 In Closing

In closing the application was very fun to develop and to play also the voice commands worked perfectly with the game although they may fail in noisy atmospheres and may fail due to different accents as we are both Irish the only accent we had available was Irish(Sorry to everyone who wants to play this who are not Irish or have strong accents, also the commands are all in English)
Special thanks to our lecturer Damien Costello for giving us this project as it opened up a whole new area of computer science to study and hopefully master
In conclusion the app was extremely fun to develop and has very few errrors(although I can almost guarantee someone will find a way to break it they always do!), and I believe it is as fun to play as it was to develop.

References

[1] U. K. P. Kenny, “Our github wiki.” https://github.com/Ultan-Kearns/GestureBasedUIProject/wiki.

[2] Multiple, “Simon.” https://en.wikipedia.org/wiki/Simon_(game).

[3] Multiple, “Kivy library.” https://kivy.org/#home.

[4] Multiple, “Kivy documentation.” https://kivy.org/doc/stable/.

[5] Multiple/Unknown, “Speechrecognition library.” google.com.

[6] J. Foer, “Moonwalking with einstein.” https://www.goodreads.com/book/show/6346975-moonwalking-with-einstein.