Interface Programming

24678805@edgehill.ac.uk

Coursework two

CALLUM MCLAUGHLIN

2022

Contents

[Introduction 2](#_Toc103174635)

[Literature Review 3](#_Toc103174636)

[Methodology - Design 4](#_Toc103174637)

[Implementation 4](#_Toc103174638)

[Individual Critical Evaluation 6](#_Toc103174639)

[Bibliography 7](#_Toc103174640)

[Appendices 8](#_Toc103174641)

[Appendix One – Menu Screen Design 8](#_Toc103174642)

[Appendix Two – Programme Screen Design 8](#_Toc103174643)

[Appendix Three – Implemented Main Menu Interface 9](#_Toc103174644)

[Appendix Four – Implemented Hello World Programme Interface 9](#_Toc103174645)

[Appendix Five – Implemented Timer Programme Interface 10](#_Toc103174646)

[Appendix Six – Implemented IF Statement Programme Interface 10](#_Toc103174647)

# Introduction

The requirement for this task was to implement a visual programming language in which the end users can create simple programmes using gestures such as hand movements, head nodding, facial expressions etc. For this task it was decided that the main form of gestures would be hand gestures and perhaps voice commands as well to support save and exit functions.

In order to complete this task a webcam would have to be used to detect the gestures and the main task was to then write a simple program without using the keyboard. There is a good amount of research now in the adoption of gestures as a means of providing users with a way to give input to a software environment, especially as gesture-based input can be perceived as a more natural means of interacting with a computer, e.g., being able to demonstrate the effect that the user wants to receive from the computer or software.

Gestures play an important role in the interaction between people as they are motions of movement which contain a lot of information such as emotion and intent. You can tell a lot by a simple gesture to put it bluntly. There is a saying that a picture speaks a thousand words, the same could be said for human gestures (Hulteen, 1990).

The programming language that will be created will be aimed at novice programmers and will be very useful to illustrate some concepts of programming and will follow a multimodal Interface design as this seems to be the best fit for a solution as mentioned by Lenman, multimodal interface solutions tend to work better for projects which require hand gestures and specific poses which are then used as commands in a command language (Soren Lenman, 2002). The language will have the ability to construct meaningful programmes and the gesture-based language will support sequence, selection, iteration and will have modules or sub programme.

The plan is that, at the end of this task, a fully functionally gesture-based programming language using a perceptive UI that will add extra usability and ease of use for the user, these types of interfaces provide input to the computer while leveraging human communication and motor skills using gestures (Turk, 2000). will have been created which can detect a user’s hand and use it as a cursor on the screen to select button/code snippets for programmes. Three programme scenarios will be created for users to interact with, a “hello world” programme, a counter programme and a timer programme.

# Literature Review

There is a vast range of research on the topic of gesture based Human Computer Interaction across multiple different disciplines which ranges from military use to basic hand gesture recognition developed by final year university students. After conducting extensive research, it was found that many programmes that use gesture-based recognition systems or control systems only require a webcam/camera and the users’ hands, there doesn’t need to be any other special equipment like special gloves or brightly coloured and contrasting objects.

An example being a research paper on “camera-based recognition of military notations from hand gestures”, outlining “Mgestyk technology” which includes hand gesture language processing with a 3D camera. This technology makes it possible to capture small hand movements and translate them into useful commands for controlling almost any windows application (Kilinc, 2022).

In the aforementioned project, Microsoft’s Kinect was also used to recognise the hand gestures using RGB and Depth data from the Kinect sensor which involves looking at specific hand motions in addition to full body motions to assist in the accurate recognition of more refined and complex gestures. By using this approach, the researchers were able to recognise ‘grasp’ and ‘drop’ gestures with over 90% accuracy (Kilinc, 2022).

There has also been extensive research into the field of Computer vision based on gesture interfaces for HCI such as the paper “Computer Vision Based Hand Gesture Interfaces for HCI” which gives an overview of hand gesture interfaces and outlines the early stages of a project about gestural command sets. The paper outlines an initial prototype for exploring the use of pie- and marking menus in gesture-based interaction.

The purpose of this study is to find out if such menus could support the development of autonomous gestural command sets with practice, however it also mentions the problems that were faced when the research was conducted such as user fatigue and precision of gestures, the “future work” section of the paper discusses how introducing flow menus could help with the reduction of fatigue (Soren Lenman, 2002).

The paper also discusses the two main scenarios for gestural interfaces where one aims at developing Perceptive User Interfaces (PUI), which strives for automatic recognition of natural human gestures integrated with other human expressions such as body movements or facial expressions and speech.

The second approach being Multimodal User Interfaces where hand poses and specific gestures are used as commands in a command language (Soren Lenman, 2002). The solution design for this task falls under the Multimodal user interface type as the programme will be designed to be run on hand gestures only and maybe some voice commands to save and exit the programme.

# Methodology - Design

For this solution, as previously mentioned, a multimodal user interface design will be adopted in order to ensure that the final solution fits the brief and to ensure an optimal solution upon completion.

Firstly, the user menu design shows the options the user has of three programmes which they can try and code, a timer, a hello world programme and a stopwatch programme will be the options for the user to choose from on the main menu. A design for this menu can be seen in Appendices (Appendix One).

The user will have to use one of their hands which will act as the mouse for the programme, to select an item they will have to hover over the button or item for a few seconds and then it will click the item, if possible, the click event will be changed to when the user makes a fist.

Once the user has selected a programme tutorial to try, they will be presented with a screen with the basic main method of the programme and options of code snippets on the left side of the screen and they will have to select the correct code snippet that is missing from the script in order to fix it.

In the design shown in Appendix 2 the user would have to select the correct code snippets to create a programme that outputs the message “Hello World”. In this case there is only once code snippet needed so as you can see it is at the top of the code snippet box, to make the programme work the user would have to use their hand and hover the cursor over that code snippet, make a fist to grab the snippet and then drag it onto the blank space on the main code area and then click run using the same gesture sequence to run the code, upon successful completion of the tutorial the user will be shown a “programme run successfully” message and if it is the wrong code snippet(s) used it will display “code run failed, try again”.

The other two programme screens will be identical in design to the one shown above obviously the code will be different as will the code snippets provided on the menu on the left side of the screen. If possible, there will also be a hints section of some kind on the screen that will give the user some direction if they fail the task multiple times.

# Implementation

The final programme that has been developed is not exactly what it was planned to be. There were lots of problems trying to get the hand gesture recognition to work as a mouse pointer so in the end this application is the result of those problems. The programme has the entire functionality required that was specified in the brief apart from the gestures being able to control the application.

For most of the programme the GUI was created using the tkinter libraries (Python, 2022). Extensive research and done on the topic and the Interface Programming blackboard site was searched multiple times and there were no examples or any material in relation to hand gestures being able to control the computer mouse or using them in the code.

As a result of this the final application is simply everything that was required except it must be controlled by the mouse. Upon running the programme, the user is presented with the main menu screen which gives them three different options for programmes they can try and fix, a Hello World programme, a timer programme and an IF statement Programme. There is also an exit button which closes the application. A screenshot of the final interface can be seen in Appendix Three.

Once the user selects a programme, they are presented with a new pop-up window which shows them the code with the missing line, the code snippet buttons on the left of the screen and the return to menu button. A screenshot each of these screens can be seen in the Appendices (Appendix Four – Appendix Six)

Each of the screens shown above have an identical layout so ease of use for the user and the buttons on the left-hand side are how the user tries to fix the code. Once they one of the snipper buttons it will run the code and tell them whether that code is correct. An example of a correct code selection and an incorrect code selection for the Hello World programme are shown below:

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

When the user has successfully completed one of the exercises, they can click on the menu button which will take them back to the programme selection screen to select one of the other two programmes.Most of the functionality in the programme comes from using the Tkinter libraries. After extensive research, some very useful documentation was found on different Tkinter objects and how to use them in Python programmes (Tutorial Kart, 2021).

There was also some difficulty faced around opening multiple windows using buttons and being able to close specific windows when the user wanted to, for example, when the user runs the programme they are presented with the main menu and if they click one of the programmes, that programme opens in a new window over the top of the main menu window and to close that window again and return to the menu, the close button on that programme instance would have to only close that window.

After some research a few very useful documents were found that helped to create the final solution. This solution was to open each programme tutorial window as a new instance (Sharma, 2021) and close that specific window with a “Menu” button which make use of the .deiconify() and .withdraw() methods to show and hide the window and would return the user to the main menu window (Schick, 2014).

When the user clicks on one of the programmes, they are presented with a screen with code snippets on the left and the code in the middle of the screen with a single line missing as mentioned before. The code in the middle of the screen was created using a basic Label from the Tkinter package but the original plan was to insert an image of the code with the missing line taken out.

The problem was that whenever an attempt to add the image to the window was made, the code would run fine and in theory it should have worked but the image was not showing up. A useful tutorial was found online which outlined how to add icons, images and exit buttons using Tkinter but it was decided after a few hours of trying to fix the image version of the code that the label approach would be much easier and efficient (Codemy.com, 2019).

# Individual Critical Evaluation

Throughout this project there have been several hurdles on the development side when coding the programme. Starting off, the code was difficult as trying to get the webcam to pick up the gestures and turn that into controlling the cursor proved a challenge. The rest of the coding was quite simple but due to a lack of understanding on how to create the solution the gesture control could not be implemented.

As stated before, extensive research was done to try and work out how to implement the hand detection to control the mouse and no resources would be found that related to the task, not on blackboard or online. Although the gesture implementation was not completed the main body of code works perfectly fine and the user can still complete each programme using the mouse controls.

The user Interface is clean and consistent throughout the project and is easy to navigate. The button events work quickly, and the application is very responsive. Originally the font of the text on screen was edited using a font variable which was then assigned to each piece of text to alter their size and weight but for an unknown reason after a few more edits to the code the font Variables stopped working and only the default text was appearing.

As for what could have caused this, it is unknown but if you look at the code you can see that there are titleFont, buttonFont and codeFont variables all of which were assigned to different buttons a text objects, but the text does not change. Logically, everything has been completed that was asked except for the gesture control implementation which I could not feasibly do as I had no knowledge of and could not find any resources on this on BlackBoard.

Overall, most of the main objectives of the project have been achieved and I feel as though the user interface while simple, works well and is easy to navigate. Sometimes simpler is better. As I was working alone and not in a group I do no have any notes or minutes from meetings. Finally, a short video demonstration of the programme running through all three code challenges can be found here: https://www.youtube.com/watch?v=z4bbUw0ntZI

# Bibliography

Codemy.com, 2019. *Using Icons, Images, and Exit Buttons - Python Tkinter GUI Tutorial #8.* [Online]   
Available at: https://www.youtube.com/watch?v=NoTM8JciWaQ  
[Accessed 10 05 2022].

Hulteen, K. a., 1990. *Hand Gesture Recognition Approach for ASL Language Using Hand Extraction Algorithm.* [Online]   
Available at: https://www.scirp.org/reference/ReferencesPapers.aspx?ReferenceID=1555059  
[Accessed 02 05 2022].

Kilinc, E., 2022. *Camera Based Recogition of Military Notations from Hand Gestures.* [Online]   
Available at: file:///C:/Users/callu/Desktop/CW-Resources-Gestures/19)%20EREN%20KILINC%20projesunum-1.pdf  
[Accessed 02 05 2022].

Python, 2022. *tkinter — Python interface to Tcl/Tk.* [Online]   
Available at: https://docs.python.org/3/library/tkinter.html  
[Accessed 09 05 2022].

Schick, A., 2014. *How To Hide Tkinter python Gui.* [Online]   
Available at: https://stackoverflow.com/questions/22764927/how-to-hide-tkinter-python-gui  
[Accessed 09 05 2022].

Sharma, P., 2021. *How to open a new window by the user pressing a button in a tkinter GUI?.* [Online]   
Available at: https://www.tutorialspoint.com/how-to-open-a-new-window-by-the-user-pressing-a-button-in-a-tkinter-gui  
[Accessed 10 05 2022].

Soren Lenman, L. B. B. T., 2002. *Computer Vision Based Hand Gesture Interfaces for Human-Computer Interaction.* [Online]   
Available at: file:///C:/Users/callu/Desktop/CW-Resources-Gestures/CID-172.pdf  
[Accessed 03 05 2022].

Turk, M., 2000. *Perceptual User Interfaces (Introduction).* [Online]   
Available at: https://cacm.acm.org/magazines/2000/3/7720-perceptual-user-interfaces-introduction/fulltext  
[Accessed 01 05 2022].

Tutorial Kart, 2021. [Online]   
Available at: https://www.tutorialkart.com/python/  
[Accessed 10 05 2022].

# Appendices

## Appendix One – Menu Screen Design

Diagram

Description automatically generated

## Appendix Two – Programme Screen Design

Graphical user interface, application

Description automatically generated

## Appendix Three – Implemented Main Menu Interface

Diagram

Description automatically generated

## Appendix Four – Implemented Hello World Programme Interface

Graphical user interface, text

Description automatically generated

## Appendix Five – Implemented Timer Programme Interface

Graphical user interface, text

Description automatically generated

## Appendix Six – Implemented IF Statement Programme Interface

Graphical user interface, text

Description automatically generated