

Computer Science
Non-exam Assessment

gesture-controlled puzzle game to aid with rehabilitation for individuals with motor impairments

Ekta Nair

Table of Contents

Analysis	3
Problem Identification	3
Prospective users	3
Interview	4
Interview questions	4
Motor impairment rehabilitation	4
Explanation for questions	4
Casual use	5
Explanation for questions	5
Answers	7
Analysis	8
Research	8
Background	8
Similar programs currently in use	9
Proposed solution	10
Limitations	11
Requirements	11
Hardware	11
Software	12
Client requirements	12
Design and functionality	12
Hardware and software	13
Objectives	13
Documented Design	15
User Interface	15
Login Window	15
Registration Window	15
Forgot Password Window	16
Sudoku Puzzle Window	16
Usability features	17
Algorithms	18
Login System	18
Gesture Recognition	20
Web Scraping	22
Check Puzzle	23
System Flowchart	24
Page Navigation	24
Top-down design	25
Data	26
Inputs and Outputs	26
Permanent Data Storage	27

Internal Data Storage	30
Class Diagram	31
Technical Solution	32
System Overview	32
Login System	32
Sudoku Puzzle	37
Virtual Mouse	43
Sectioned Code	46
Testing	47
Test Plan	47
Test Data	47
Input & Output Testing	48
Navigation Testing	52
Alpha Test	53
Input & Output Testing	53
Navigation Testing	60
Retests	61
Testing Outcome Screenshots	66
Beta Test	78
User Feedback	79
Evaluation	82
Evaluation of Objectives	82
General Evaluation	85
Potential for Future Developments	85
Bibliography	87

Analysis

Problem Identification

The growing popularity of computer games has given way to an ever-expanding variety of games. As a result, the increasing focus on the accessibility of computer games should come as no surprise as developers look to expand their target audience. However, there is a vast range of disabilities that exist in the world that may in some way hinder human-computer interaction. The main categories of disability encountered are physical impairments; limitations in mobility, vision, hearing, and cognitive abilities.

The creation or modification of a simple game allowing for the use of gesture controls would provide those with motor disabilities with easier access to computer games. After all, why shouldn't a person with motor impairments or disabilities have the same access to services and leisure as others in society? Furthermore, games like this can be used in rehabilitation to improve manual dexterity and cognitive abilities. They could also be used for casual leisure and enjoyment or adapted for educational purposes.

Individuals with motor impairments often find themselves frustrated with repetitive exercises and unmotivated to continue their rehabilitation. I would like to create a system that would entertain them while improving their motor skills and cognitive abilities.

The features of a computer system that would be required for this solution would include computer hardware capable of image processing with computer vision. A webcam would also be required with a suitable operating system. In essence, the solution would take a picture of the user using the camera, and the software would identify the location or gesture of the hand and perform adequate commands in order to complete the game.

Prospective users

The clients for this program are those with motor impairments going through rehabilitation. The program would specifically target elderly patients as they are generally less comfortable with the traditional methods of using a computer. They also tend to struggle more with rehabilitation as their bodies do not heal as fast; hence require more motivation to progress along with their rehabilitation. A simple game would allow them to do as such in a fun and enticing way. The critical thinking required to complete the game would also encourage critical thought, helping to prevent the loss of cognitive abilities due to old age and compromising circumstances.

Nevertheless, the umbrella of clients for this program is very large. As the motion and gesture controls can technically be used by anyone whether disabled or otherwise. It can be used by teachers and students for educational purposes, as the required movement in the controls would encourage children, especially younger ones, to maintain their interest in the game for longer. It may also be used for the elderly to strengthen or maintain their motor abilities as they age or by the casual user for leisure purposes.

Interview

Interview questions

I will outline key questions to be asked to clients to gain a better understanding of their opinions of the software and their ideal method of using it. I may ask follow-up questions catered to the individual client and their responses to obtain as much useful information from the client as possible.

Motor impairment rehabilitation

My questions for Ambika (72 years old) and Aravind (78 years old) who have gone through rehabilitation due to their motor impairments are as follows:

1. What has been your experience with rehabilitation?
2. What is the biggest challenge you have faced in your recovery and how have you overcome it?
3. Do you play online games? If so, what are your favourites?
4. Have you ever used online games to aid the rehabilitation process?
5. If yes, what sort of game was it? If not, do you think this is a feasible idea?
6. *What was your experience with it? Did you find it accessible/beneficial?
7. What do you think about using hand gestures to control a game? Do you think it will be beneficial for the rehabilitation process? Do you think you would enjoy it?
8. What sort of gestures would you be comfortable making?
9. Do you have any other comments or concerns about such a program?

**Question 6 will only be asked if the client has experience with using online games to aid their rehabilitation*

Explanation for questions

Question 1 establishes the client's background with rehabilitation allowing me to assess whether they are a reliable source and have a lot of experience on the matter.

Question 2 identifies the client's mentality towards rehabilitation as well as their willingness to solve a problem. This could be important in judging how user-friendly a system is, as some clients may put more effort into attempting to use and understand the system than others. It may also identify further problems the client has that could be solved by including specific features in the program.

Question 3 identifies the client's knowledge of games as well as the type of game that may come across as appealing to the client. Similarly, questions 4-6 establish the client's fondness for and experience with computer-based games. This will be crucial in tailoring the program to satisfy the client.

Questions 7 and 8 identify the potential of gesture controls and the client's willingness to use them. It also identifies popular gestures that can be used.

Question 9 allows the client to raise any concerns. This will be useful to highlight any limitations to the program that have not previously been thought of. It will also be a good gauge of the client's overall satisfaction with such a program.

Casual use

My questions for Rekha (68 years old) who represents using the program for casual gaming are as follows:

1. Do you play online games? If so, what are your favourites?
2. What do you think about using hand gestures to control a game?
3. Do you think it will be beneficial to improve dexterity? Do you think you would enjoy it?
4. What sort of gestures would you be comfortable making?
5. Do you have anything else to add?

Explanation for questions

Questions 1 and 2 identify the client's knowledge of and comfortability with games as well as the type of game that may come across as appealing to the client.

Questions 3 and 4 determine the potential of gesture controls and the client's willingness to use them. It also identifies popular gestures that can be used.

Question 5 allows the client to raise any concerns or add any opinions that may have been missed.

Answers

Ambika

1. What has been your experience with rehabilitation?

“My experience with rehabilitation is based on my recovery from a bilateral knee replacement operation. It was indeed painful yet you have to move, that is, start walking to avoid any embolism (clot) and create mobility.”

2. What is the biggest challenge you have faced in your recovery and how have you overcome it?

“The biggest challenge is sitting & getting up when both knees have a new joint implanted, your muscles and nerves inside are cut and left to grow to support the knee. The pain is excruciating yet you have to do the various physio exercises.”

3. Do you play online games? If so, what are your favourites?

“Online games are not my forte but I have tried doing word puzzles and precision shooting of things as it moves. Activities that control my hand movement, sight, and brain are important for coordination at my age “

4. Have you ever used online games to aid the rehabilitation process?

“Yes, I have.”

5. If yes, what sort of game was it? If not, do you think this is a feasible idea?

“There was one activity my therapist did to build my thigh muscles called Virtual Biofeed. I had to do a leg press with 15 kg and control a ball so that it goes up and down correctly following the curves on the computer screen... This is definitely a feasible idea”

6. What was your experience with it? Did you find it accessible/beneficial?

“I did find it beneficial in terms of my rehabilitation.”

7. What do you think about using hand gestures to control a game? Do you think it will be beneficial for the rehabilitation process? Do you think you would enjoy it?

“I guess it would be most beneficial for neurorehabilitation and based on my previous experience I suppose I would enjoy it.”

8. What sort of gestures would you be comfortable making?

“Hand gestures; finger gestures.”

9. Do you have any other comments or concerns about such a program?

“In this day and age, I would encourage the making of programs that help to rehabilitate any person to overcome chronic aches & pain related to their age or related to their work.”

Aravind

1. What has been your experience with rehabilitation?

“After my stroke, I employed a physiotherapist to help me with exercise.”

2. What is the biggest challenge you have faced in your recovery and how have you overcome it?

“I have no patience to do on my own, so a therapist helps - or more like forces - me to do the routine twice a week.”

3. Do you think it will be beneficial to improve dexterity? Do you think you would enjoy it?

“Occasionally I play rummy on the computer.”

4. Have you ever used online games to aid the rehabilitation process?

“No”

5. If yes, what sort of game was it? If not, do you think this is a feasible idea?

"I do think it is feasible, however, I'm not sure I will have the patience to learn how to use it."

6. What was your experience with it? Did you find it accessible/beneficial?

Omitted due to client's lack of experience

7. What do you think about using hand gestures to control a game? Do you think it will be beneficial for the rehabilitation process? Do you think you would enjoy it?

"I think it would be. I'm not comfortable using the computer for such things, however, I suppose I would enjoy it once I get the hang of it."

8. What sort of gestures would you be comfortable making?

"Hand and finger movements to exercise the unused muscles."

9. Do you have any other comments or concerns about such a program?

"I am mostly into traditional workouts like hiking twice a week and personal training in the gym twice a week but perhaps I need to start with new exercises to keep me motivated."

Rekha

1. Do you play online games? If so, what are your favourites?

"Yes, I play Words With Friends, which is like Scrabble."

2. What do you think about using hand gestures to control a game?

"I think it is a feasible idea but I don't have much experience with it."

3. Do you think it will be beneficial to improve dexterity? Do you think you would enjoy it?

"I think it would help to improve dexterity but for me, it pays to be focused in the game. I think creating hand signals would distract me, hence I wouldn't enjoy it as much. But that opinion will differ from person to person."

4. What sort of gestures would you be comfortable making?

"In Words With Friends, I use a thumbs-up emoji or clap emoji to appreciate my opponent's moves. So, perhaps something similar to those."

5. Do you have anything else to add?

The client did not have any further statements to add

Analysis

From the interviews it seems clients are fond of the idea to create a game that encourages the movement of muscles for rehabilitation. However, as Ambika mentioned in her interview, prospective users need to only be those who are impaired due to medical illnesses; it could also be those who have aches and pains due to the nature of their job (eg. working hunched over a computer all day).

Gestures used would need to be extremely simple to make it as easy as possible for users to understand and execute. As well as for them to not lose patience when trying to play the game. The option to play without the use of gesture controls will also be included. Ideally, a puzzle game would be created as that seems to be the game of choice within my target demographic.

Research

Background

Referencing a report by the Board of Community Health Councils in Wales, it was concluded that “many patients feel time passes very slowly when they are in treatment” which “often leads to feelings of loneliness and depression”¹. This can slow recovery rates.

The use of computer games to aid in rehabilitation is not a recent development. Many hospitals across the world have signed up to the “robotics rehab” program that is having positive results in patients who have not progressed with traditional therapy in years, according to an ABC article². Similarly, an article by Brunel University London claims new technology is helping people regain strength and movement in their arms and hands by making dull daily rehab exercises more fun³. Devices like Microsoft Kinect and WiiFit have been used to aid in the rehabilitation of stroke and physically impaired patients. However, these require the user to hold a remote to play games which may be a struggle for clients who do not have adequate finger dexterity or are unable to perform fairly intense full-body exercises. They also often require additional assistance from a caretaker and are costly.

While the systems mentioned above may be better suited to full-body rehabilitation, my solution will be focused on an affordable and accessible simple game to pass the time while simultaneously executing strengthening gestures and also challenging the mind. This is to avoid patients from feeling lonely or depressed while not tiring them out with constant strenuous activities.

To create my solution, I would have to identify a game suitable for my clientele as well as suitable gestures to control the game.

According to a survey conducted by AARP⁴ of 3737 people aged 50 or older, puzzle and logic games were the most popular type of game played by people of this demographic

(49%). This was followed by card and tile games (47%). The most common of such games include Scrabble, Sudoku, and other word puzzles such as crosswords or word searches. These games require logic and memory skills. They also encourage strategic thinking and concentration which prevents the regression of the mind⁵. This aligns with what types of games the clients I interviewed play.

Similar programs currently in use

- Leap Motion⁶

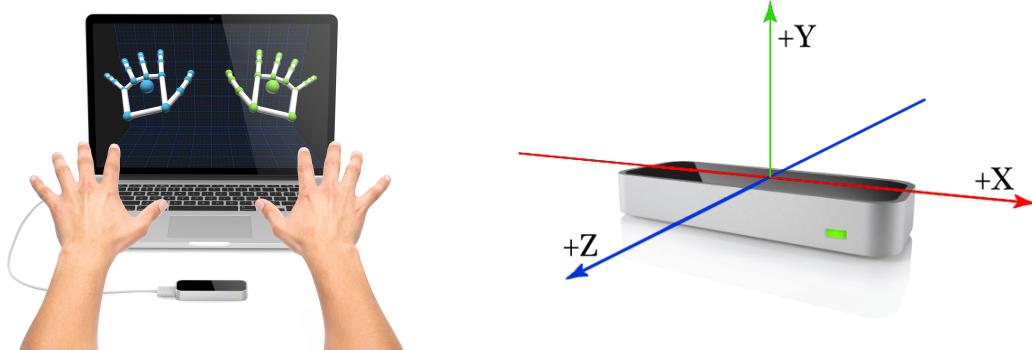


Fig. 1.1: Skeletal model of hand on Leap Motion software; Leap Motion controller

Overview:

Leap motion is a computer hardware sensor device that supports hand and finger motions or gestures as input equivalent to a mouse, but requires no hand contact or touching. Leap Motion uses a right-handed three-dimensional Cartesian coordinate system that recognizes and tracks hands and fingers and their related gestures. It is a device that operates in close proximity with high precision using optical sensors and infrared light. As the Leap Motion controller tracks hands and fingers in its field of view, it provides data detailing the location of points on the hand at any time. The Leap Motion software uses an internal model of a human hand to provide predictive tracking even when parts of a hand are not visible.

Aspects I can use:

I would also use hand gestures as input analogous to a mouse, however, I plan on using a cheaper and more readily available device than an infrared sensor to make my system more accessible and appealing to a wider audience. Leap motion also identifies key parts of the hand (eg. fingertips, knuckles, palm) in order to track the gestures accurately. This is something my system will need to do as well.

- Sudoku4ALL⁷



Fig. 1.2: Sudoku4ALL user interface

Overview:

Sudoku4ALL is a Sudoku game for people with motor impairments. It allows a person to control the game with their voice or with a single switch on the keyboard. It contains six difficulty levels (all fully accessible to play using voice commands) with special commands and options dedicated to making the game playable for blind people. It uses algebraic notation similar to that used in chess to allow users to identify and select individual squares in the sudoku puzzle and fill them in using their voices. It also contains buttons that can be used in substitute to the voice command and an AI to check and solve puzzles for the user if necessary. All puzzles are saved and can be continued at a later time if the user decides to leave the game incomplete.

Aspects I can use:

I intend to create a puzzle game with varying levels of difficulty as this will keep the game enticing as the user is motivated to progress through the levels as well as appeal to potential users of varying skill levels. I would also like to program artificial intelligence (AI) or create an algorithm that would check the puzzle for the user. The algebraic notation could be useful to more easily identify sections of the puzzle, however, using gesture controls as arrow keys or to control the mouse pointer seems like a more simple method. This is feasible in my program as it is not suited to blind people as Sudoku4ALL is.

Proposed solution

My initial concept, taking into account my research, would be a simple game of Sudoku that would allow users to play by using various hand gestures to select boxes and numbers.

Users would initially be shown to a “Login” window where they would have to enter a valid username and password to be able to log in and start playing. If this is their first time playing they can click the “Register” button which will take them to a window where they can set their username and password. If a user has already registered but has forgotten their password

they can click on the “Forgot Password” button which will have them answer a question whose answer was pre-determined during registration. If answered correctly the user will be able to reset their password and can proceed to log in normally.

Once users log in 2 windows will open. One is a front-facing camera display that will detect a hand when in the frame. The index finger will be used to drag the cursor and a ‘click’ will be registered when the middle finger is raised.

These gestures will be used to control the cursor to play a Sudoku game. The index finger will be used to move the mouse pointer, and raising the middle finger alongside it will be used to ‘click’ the mouse.⁸ There will be 3 levels of difficulty of the puzzle that users can select by clicking the corresponding buttons. The Sudoku puzzles will be retrieved from an online sudoku website using web scraping. This will allow the program to constantly generate new puzzles, keeping the game fresh and interesting, without the need for overly complex code within the program itself.

Users may enter values into the vacant grid spaces by selecting the grid space, then selecting a value on the onscreen keypad. Users may also choose not to use gesture controls and solve the puzzle in a similar fashion while using their device’s mouse to control the cursor and keys to input the values. Once a user has completed the puzzle they can click the “Check” button to see if their answers are correct. Grid spaces with incorrect answers will be shaded red. They may then click the button corresponding to a harder level to generate a more difficult Sudoku puzzle or they may click the button corresponding to the same level to generate a new Sudoku puzzle of equal difficulty.

Limitations

The main limitation is that a simple webcam cannot detect depth, reducing the range of gestures available. For this reason, simple pre-determined gestures are used to avoid unnecessary hassle for users when trying to identify appropriate gestures. As such, users would not be able to alter the gestures to suit their individualised rehabilitation plans. However, a gesture calibration feature can be added in future iterations to combat this.

Another limitation would be the environment the user is in. The software would work well against a plain background but if the user was in a crowded environment or one with insufficient lighting the software might struggle to correctly identify the user's hand gestures.

If the website the puzzles are retrieved from changes in layout or in the way they store data the program would lose functionality. The code would then have to be altered to suit the new website format. Additionally, users are not able to review puzzles they have already solved. This feature is something that could be added in future iterations as a way for users to track their progress.

Requirements

Hardware

A computer capable of supporting standard input and output (IO) peripherals is required. The computer will be required to support a webcam, which it will use to allow gesture controls to work appropriately, as well as a monitor. The webcam should be of sufficient resolution (above 480p) to ensure the image recognition software can clearly identify gestures. Most desktop and laptops computers meet these requirements and have a fast enough processor to run the software.

Software

Python and corresponding IDE and interpreter. The code will be written in Python, therefore it requires the appropriate software and interpreter in order to run. Tkinter will be used in conjunction with the above to produce the GUI.

MySQL to create a database of user information that can be used to validate if a user is registered and hence able to login to the system and play the game.

Pygame to create the window with Sudoku puzzles and iterative algorithms to check them.

OpenCV for Python. This is the software that will be used to create gesture controls using image recognition/computer vision.

BeautifulSoup to be able to carry out web scraping to generate the Sudoku puzzles. The website “nine.websudoku.com” has been identified as the website to scrape from as it is able to generate its own new Sudoku puzzles and has different difficulty levels which align with the needs of my program.

A standard operating system (eg. Windows, macOS, or Linux) that can support Python and OpenCV so that the system can successfully be run on the computer.

Client requirements

Design and functionality

Requirement	Explanation
Simple GUI, lightweight design	Easy for users to use and understand. Crucial as the target demographic are generally less competent in using computers.
Clear webcam feed	So the user can adjust their camera accordingly to ensure their hands are detected for the gesture controls to work.
Use of webcam to move the mouse	This is one of the main functions of the program. It uses the feed from the webcam

	so that you can control the computer's cursor by moving your hands.
"Login" window with the "Forgot Password" option	Users are able to log in easily even if they have forgotten their password. Crucial for the target demographic as forgetfulness can be a normal part of ageing.
A game that is simple to play and understand	To ensure users don't get frustrated when using the program.
Ability to use the program without gesture controls	If users cannot control the mouse properly using gesture controls they can still use the program.
Controls linked to simple gestures, namely gestures to move and click the mouse	A set of gestures need to be mapped to different controls to allow users to play the game. Gestures should be simple so that users can pick them up easily and as many people as possible can play the game.

Hardware and software

Requirement	Explanation
A computer with standard peripherals (monitor and webcam) that can run basic image processing	The user must have a basic computer in order to access, run and navigate the program.
Webcam of at least 480p	The webcam is used to control the controls of the game hence it needs to be clear so that the computer can easily identify gestures and run the program smoothly.
Windows, Mac, or Linux operating system	These are operating systems that are supported by Python.
Python with OpenCV, Mediapipe, Numpy, Tkinter, and AutoPy	The program will run on Python with OpenCV, which uses Numpy and Mediapipe. Tkinter and AutoPy will be used for the user interface.
BeautifulSoup with requests	The program will use requests to extract data from HTML and BeautifulSoup to sort through the data in order to generate Sudoku puzzles.
MySQL with MySQL Python connector	MySQL will be used to create a database of users that will serve as a dictionary to validate who is able to log in to the program. The connector is used to ensure it can be referenced by the Python program.

Objectives

1. Username and password must be validated before allowing the user to access the gesture-controlled game
 - 1.1. Compare user input to data stored in database
 - 1.2. If invalid, show error message
 - 1.3. If valid, open the gesture-controlled game window on top of login window
2. When new users are registering their accounts, user data must be validated before it is stored in database
 - 2.1. If invalid, show error message
 - 2.2. If valid, store data in the correct columns in the database and show success message
3. Feature to allow users to reset passwords to their account
 - 3.1. Validate username, security question, and answer by comparing it to data stored in the database
 - 3.2. If valid, delete the original password from the database and replace it with the user's input for a new password
 - 3.3. If invalid, show an error message
4. Virtual Mouse must be able to detect hand from camera feed
 - 4.1. Camera feed must be shown to user
 - 4.2. Drawings are overlaid on the hand in the feed to convey to the user that the hand has been identified
5. Hand gestures must be identified and appropriate action must be taken
 - 5.1. Move index finger to move mouse
 - 5.2. Raise middle and index fingers to click
6. Virtual Mouse must be able to interact with the Sudoku puzzle (ie. be able to select buttons and cells on the Sudoku grid)
7. The onscreen keypad must be able to be used to enter values into the Sudoku grid to solve puzzle
8. The gesture-controlled game must be able to generate new Sudoku puzzles of varying difficulties
9. If the user does not wish to use the onscreen keypad, valid values can be entered into the Sudoku grid by pressing the appropriate keys on the keyboard
10. A feature that will check if the Sudoku puzzle completed correctly by the user is correct
11. The different windows of the system must be interconnected and open and close smoothly when needed
 - 11.1. Users should be able to navigate between windows easily and independently

Documented Design

User Interface

Login Window

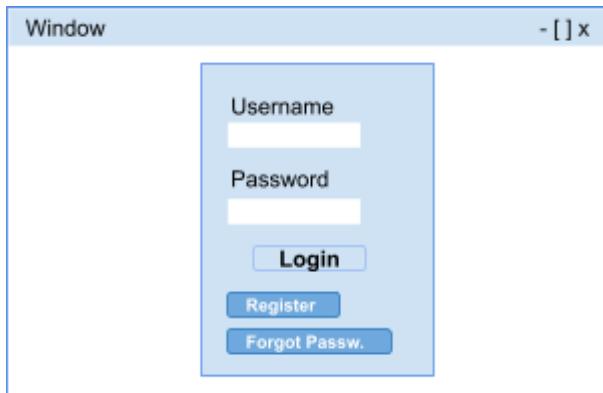


Fig. 2.1: Outline of the login window's graphical user interface

This is the first window that opens when the program is run. Users would be required to enter their username and password into the appropriate fields before pressing the “Login” button. If the login is successful (ie. the username and password entered are valid; match with ones stored in the database) the “Sudoku puzzle” window would open. Otherwise, an error message would pop up to tell the user their username or password is invalid.

The user may press the “Register” button which will lead them to the registration window where they can enter the necessary details to create a username and password for themselves. The user may also press the “Forgot Password” button which will lead them to the forgot password window where they can enter the necessary details to create a new password.

Registration Window

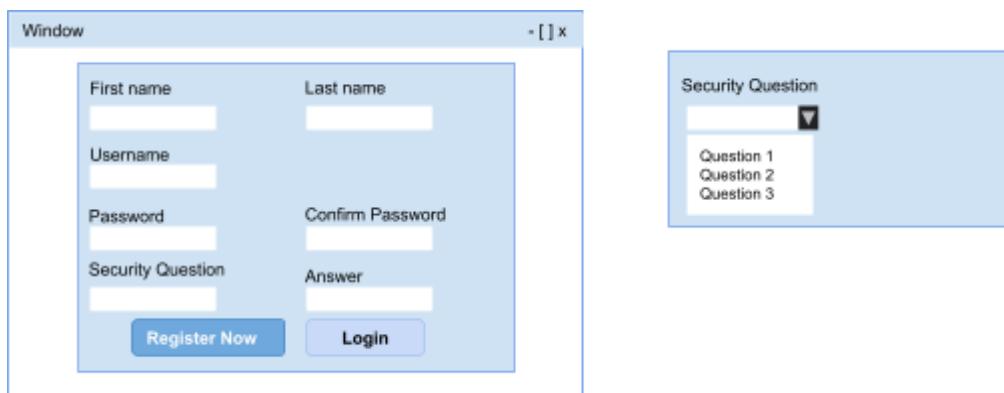


Fig. 2.2: Outline of the registration window's graphical user interface, with a closer look at the dropdown feature of the security question's combo box input

This window will open if the user presses the “Register” button in the login window. Users will have to enter the required details (as shown in the diagram above) before pressing the “Register Now” button to have their details saved. They would then need to press the “Login” button to be directed back to the login window where they can continue to log in as stated above.

Rather than typing in a question in the given space as the other fields, the “Security Question” field contains a dropdown menu of questions to select. Users would press the down arrow at the right-hand side of the field to produce the dropdown menu, then select one of the predetermined questions by clicking on it. They would then answer the selected question in the space given below the heading “Answer”.

Forgot Password Window

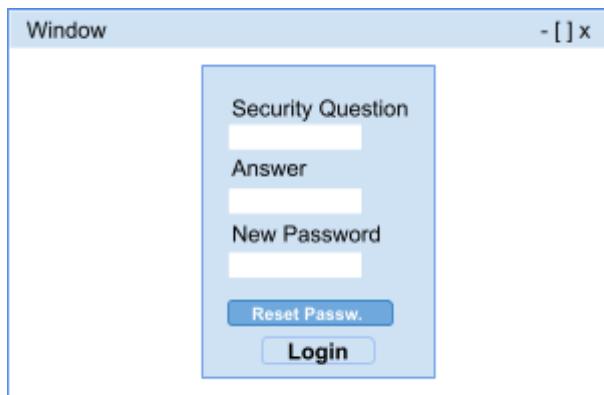


Fig. 2.3: Outline of the forgot password window’s graphical user interface

This window will open if the user presses the “Forgot Password” button in the login window. Users will have to select the security question and provide the answer they previously gave when registering. They would then have to enter a new password into the appropriate field before pressing the “Reset Password” button to have their password updated. If the security question and answer align with what the user previously determined (this will be checked by referencing the database with their username) their password will be updated to the one they entered into the “New Password” field. They would then need to press the “Log in” button to be directed back to the login window where they can log in normally with their new password. Otherwise, an error message will appear to notify the user that the information they entered is incorrect.

Sudoku Puzzle Window

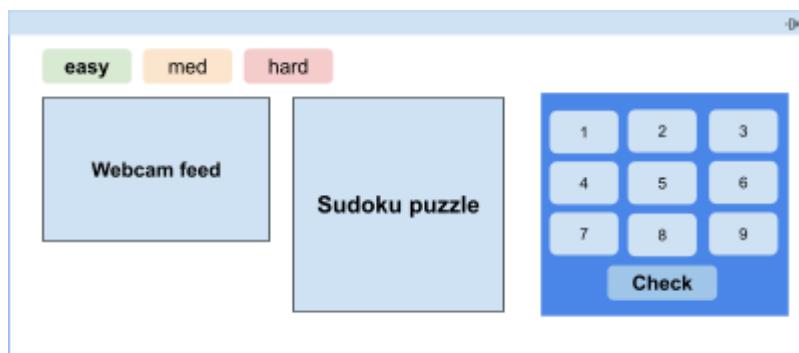


Fig. 2.4: Outline of the sudoku puzzle window's graphical user interface

Once the user successfully logs in the Sudoku puzzle window will appear. Users can press the “easy”, “medium” or “hard” buttons to generate a new sudoku puzzle with the corresponding difficulty. The puzzle will appear in the box labeled “sudoku puzzle” above. Grid spaces with numbers already in them will be shaded grey. If a user clicks on an empty grid space it will be shaded light blue indicating that it has been selected. The user can then press one of the numbers on the onscreen keypad to enter that number into the selected grid space. The user can then press the “Check” button to check if their complete puzzle is correct. If a number is in the wrong place that grid space will be shaded red.

The box labeled “Webcam feed” will contain the image the webcam ‘sees’, with some image analysis overlaid so the user can see what the program is doing. The overlaid information would be green circles over key points of the hand (eg. the fingertips) and a purple circle over the pointer finger, indicating that the computer is tracking that finger; its movement corresponds to the movement of the cursor. If the middle finger is raised the purple circle will appear between the pointed and middle finger to indicate the mouse has been clicked.

Usability features

The usability features that I have considered to make sure that the program is easy to use for as many types of users are that I have made the buttons fairly large with different colours to their backgrounds and clear bold text to make it easy for users to see and click. Considering that some of the time the mouse will be controlled by hand gestures, aiming the cursor may be more difficult. The buttons will change colour when clicked to notify the user that the action has occurred.

Any text needs to be large and easily readable for people with vision difficulties. I have used a bold sans serif font in black to ensure this is the case⁹. Some people may use screen readers with a text-to-speech facility, so for this, I need to make sure the text on the program is the actual text and not images so that screen readers can pick it up more easily.

The multiple windows may be confusing for the user, so I have automated the windows to appear and disappear when no longer needed to avoid confusion for the user and to declutter the user’s screen providing a simple, lightweight design. For example, once a new user has successfully registered and pressed the “Login” button the registration window will

automatically close and the “Login” window will open in its place. Similarly, when a user successfully logs into the system, the login window will disappear and the Sudoku puzzle window will appear.

I chose green and purple as the colours for the image processing overlay as the colours contrast each other¹⁰ so it will be obvious to the user when the hand is simply being detected and when an action is occurring due to the hand gestures. The frames per second of the user’s webcam will also be automatically calculated and displayed in the top left-hand corner of the webcam feed so that the user is aware if the resolution of their camera is too low for the image processing to work properly.

Algorithms

Login System

The main login page will only allow users to log in if they have a valid username and password, that is one that has already been registered and is saved in the 'mydata' MySQL database.

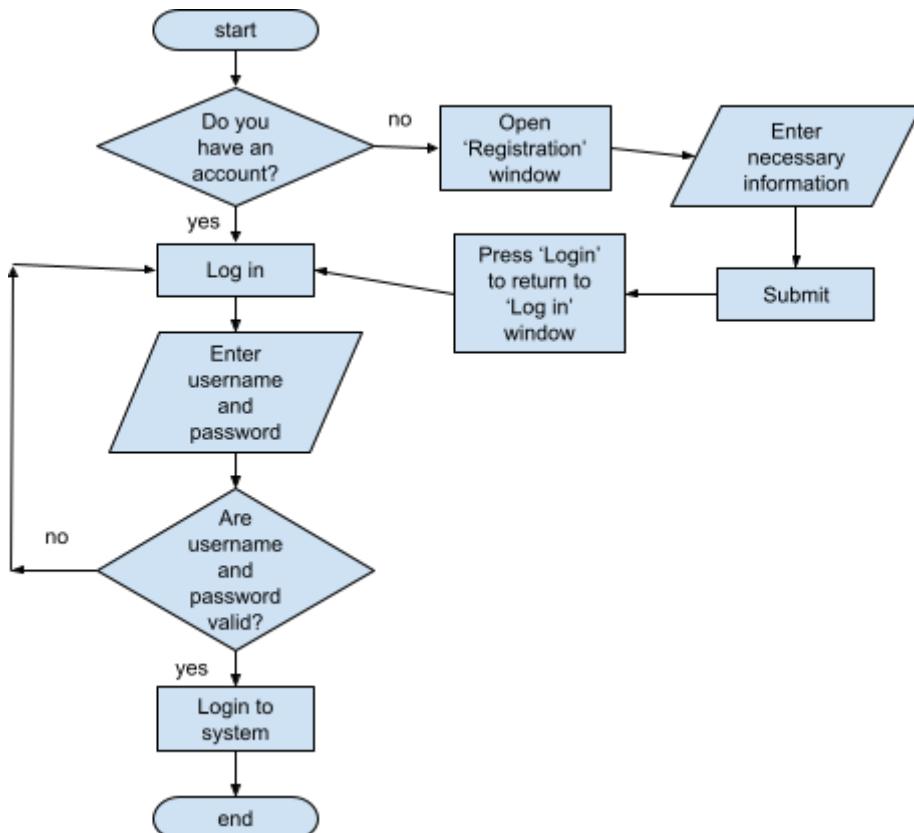


Fig. 3.1: Flowchart of logging-in process

The username and password entered by the user would be retrieved using the query:

```
select * from register where Username=%s and Password=%s
```

If nothing is retrieved then the user's input is invalid.

Note: 'register' is the name of the table that the data is stored in and %s will be replaced by the data input by the user.

If users are new they may register a username and password for the first time. To ensure the system is secure and that the data stored is normalised, no 2 users are allowed to have the same username. This can be checked by retrieving the username entered by a new user registering using the query:

```
select * from register where Username=%s
```

If a username is retrieved the new user will not be able to register their account using that username and will be notified that they need to change it.

If users are successful in their registration, their data will be added to the database using the query:

```
insert into register values(%s,%s,%s,%s,%s,%s)
```

Existing users who forget their passwords will also be able to reset them with the 'Forgot Password' window. The process to reset passwords is as below.

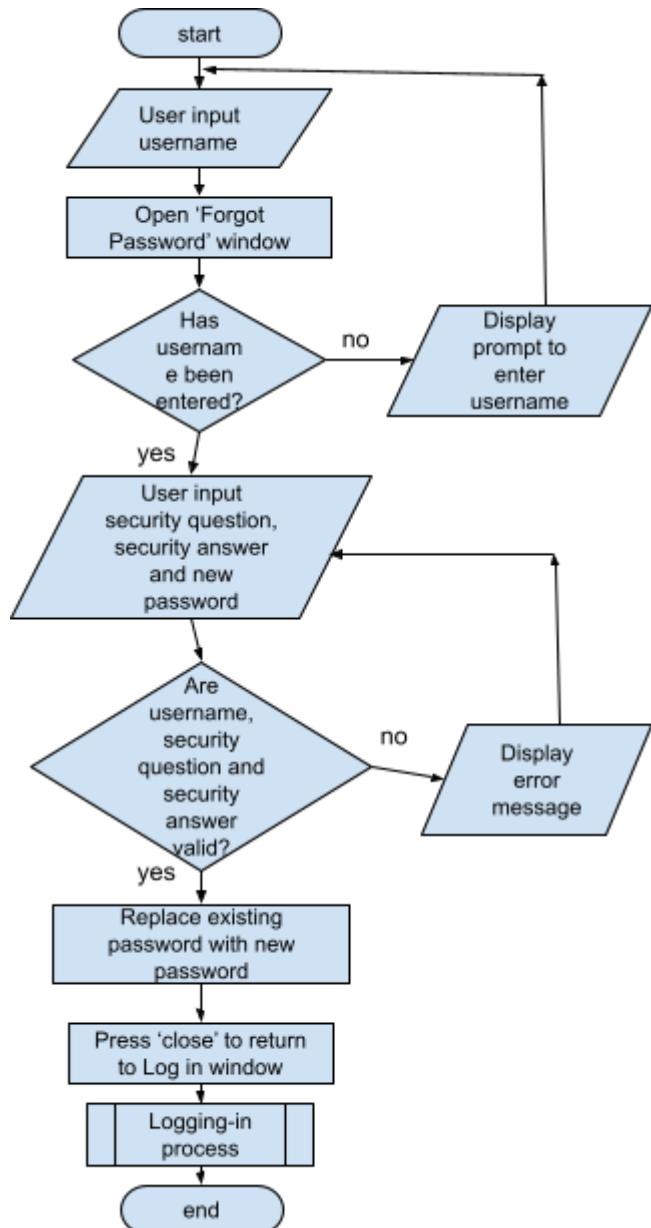


Fig. 3.2: Flowchart of resetting password process

The above can be achieved using the queries below to validate the user then subsequently replace their existing password with the new one. In order for the user to be validated, they must enter a valid username and select and answer their predetermined security question correctly. This can be done by retrieving the user's username, security question, and security answer from the database and comparing it to the user's input.

Query to retrieve data:

```
select * from register where Username=%s and SecurityQ=%s and SecurityA=%s
```

Query to reset password:

```
update register set Password=%s where Username=%s
```

Gesture Recognition

The main challenge is the image processing technique using OpenCV. The user interface will rely on this technique as it is what is controlling the mouse and selecting the various buttons within the user interface. The basic algorithm of the user interface is:

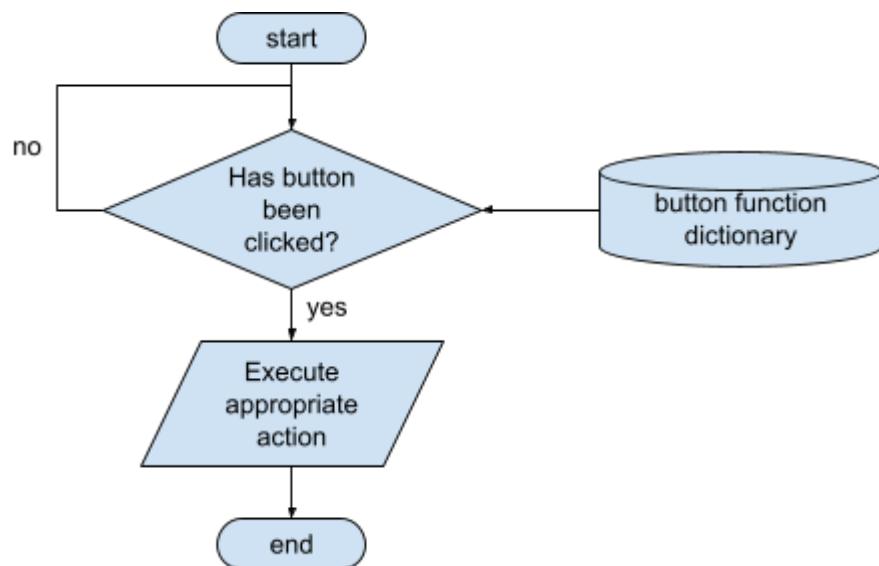


Fig. 3.3: Flowchart of button clicking process

However, in order to identify if a button has been clicked in the first place, the hand gestures that signal a click or move of the cursor must be identified. A transformational algorithm is used to identify the coordinates of the fingertips from the webcam screen to the computer window for full screen controlling of the mouse. OpenCV and Mediapipe are used in the algorithm to capture and analyse the webcam feed. The camera resolution will be 850 by 800 pixels and is in the RGB colour space for the gestures to be identified.

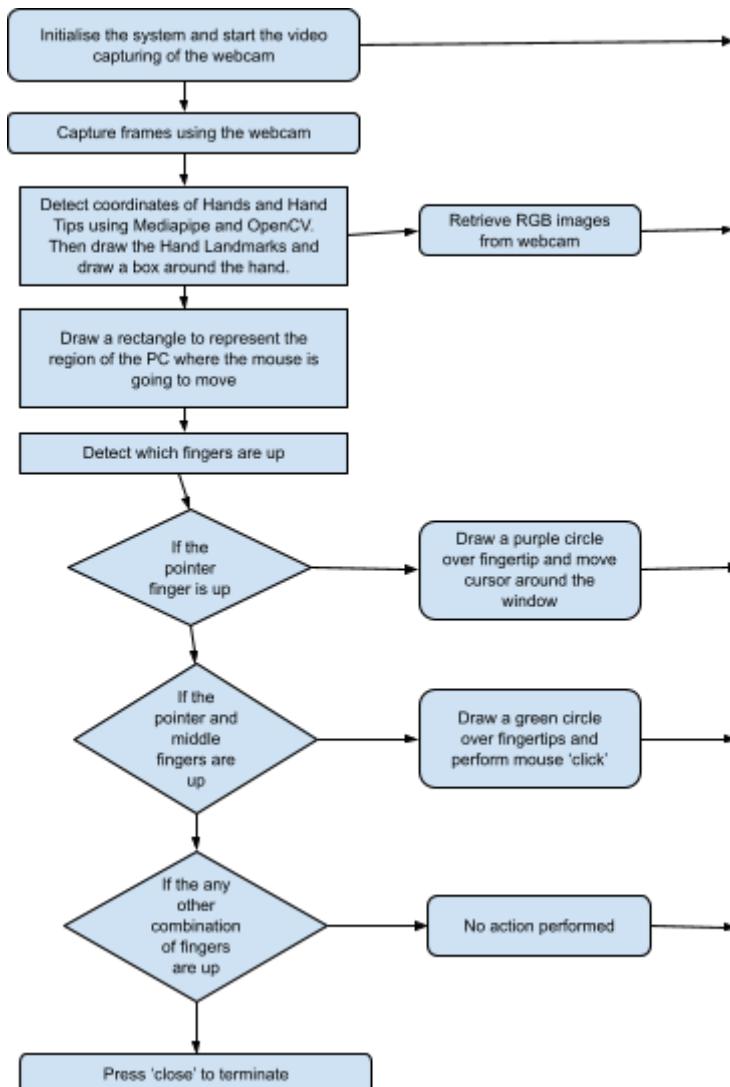
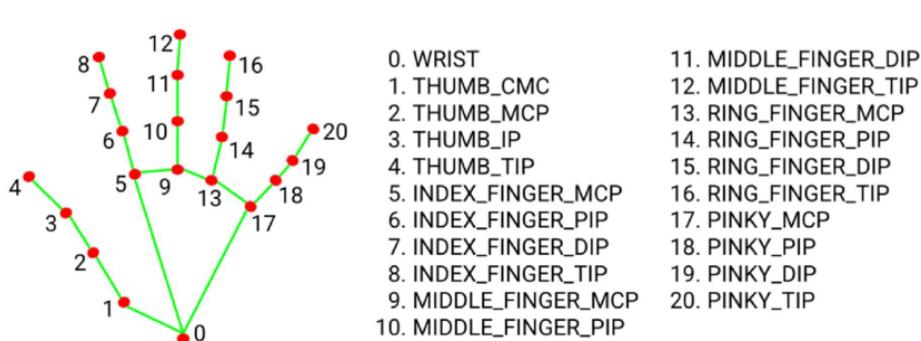


Fig. 3.4: Flowchart of gesture recognition process

Hand landmarks can be identified using Mediapipe, whereby each point on the hand corresponds to a number. Hence specific points on the hand will be referenced in the code using their corresponding numbers, shown below.



11

Fig. 3.5: Diagram of hand landmarks and their corresponding ID numbers

Web Scraping

Using the automated extraction technique of HTML parsing and the Python library BeautifulSoup, Sudoku puzzles are scraped from the website “nine.websudoku.com”. The difficulty of the puzzle retrieved from the website is specified by the user by clicking the corresponding button (ie. Easy, Medium, or Hard). The ‘click’ of the difficulty level button is the trigger for the automated extraction to occur.

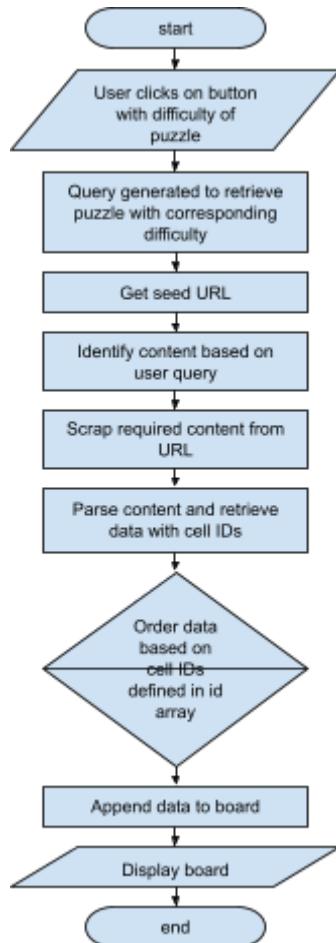


Fig. 3.6: Flowchart of web scraping process

When BeautifulSoup extracts the source code from the website, it stores it in a parse tree.

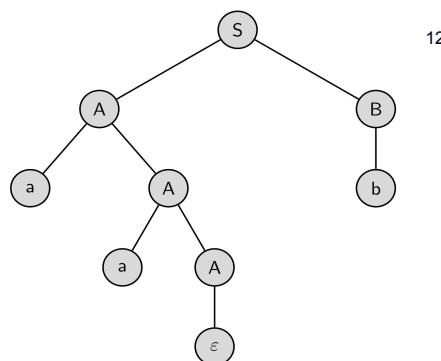


Fig. 3.7: Parse tree structure

This hierarchical structure is to make it easy to search for the necessary data within the source code. I chose the website “nine.websudoku.com” because the data for the Sudoku puzzles on the website are neatly stored with `<div>` tags, and each cell has a unique ID which will make it easy to identify and extract during parsing.

```

<p style="margin:0;"> </p>
<div id="puzzle_container" style="position:relative;"><table id="puzzle_grid" cellspacing="0" cellpadding="0" class="t">
<tbody>
<tr><td>9</td><td></td><td></td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>3</td><td>7</td><td>9</td><td>2</td></tr>
<tr><td>8</td><td>2</td><td>7</td><td>5</td></tr>
<tr><td></td><td>3</td><td>2</td><td>6</td><td>5</td></tr>
<tr><td>6</td><td></td><td>1</td><td></td><td>2</td></tr>
</tbody>
</table>
</div>
<input name="prefix" id="prefix" type="hidden" value="3i0cq">
<input name="start" type="hidden" value="1645751114">
<input name="inchallenge" type="hidden" value="">>
<input name="level" type="hidden" value="1">
<input name="id" id="pid" type="hidden" value="6618365488">
<input name="cheat" id="cheat" type="hidden" value="956143278"

```

Fig. 3.8: HTML source code to be extracted from “nine.websudoku.com” website

Check Puzzle

An iterative algorithm is used to check if the puzzle the user solved is correct. The algorithm checks for repeating numbers by comparing the numbers to a predefined array called ‘possibles’ which contains the numbers from 1 to 9.

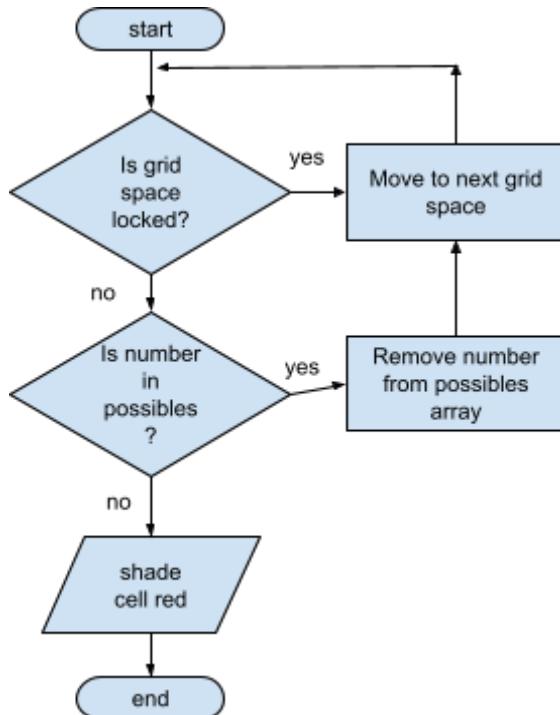


Fig. 3.9: Flowchart of puzzle checking algorithm

When the ‘Check’ button is pressed this algorithm will run 3 times: once for each row, once for each column, and once for each 3 by 3 grid within the 9 by 9 puzzle. The algorithm will iterate through the grid spaces - from left to right and up to down - until it has completed every row, column, and 3 by 3 grid. Only if no errors are found in all 3 checks will the puzzle be considered correctly solved. If there are any errors, that grid space will be shaded red to convey the error to the user.

System Flowchart

This system will follow a modular design. That is each window will be coded separately using OOP, then linked together using buttons to open and close the windows. This is ideal as each window has a unique function and works independently from the rest. A modular design will also aid with the maintenance of the system and allow certain modules to be used in other projects, saving time and resources.

Using OOP it is easier to trace which class(window) each object belongs to in the case of an error when linking the pages together, as each window requires different data inputs.

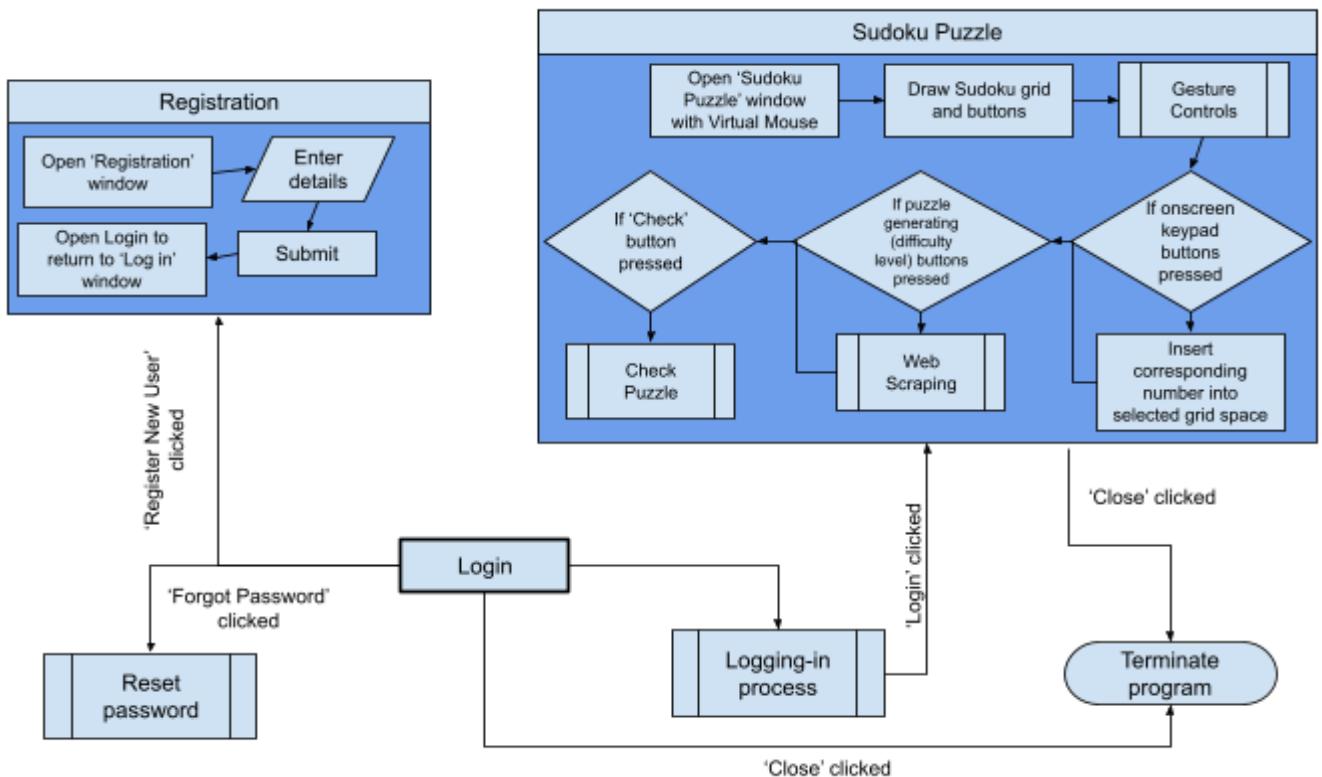


Fig. 3.9: Flowchart showcasing how the different parts of the system interact

Page Navigation

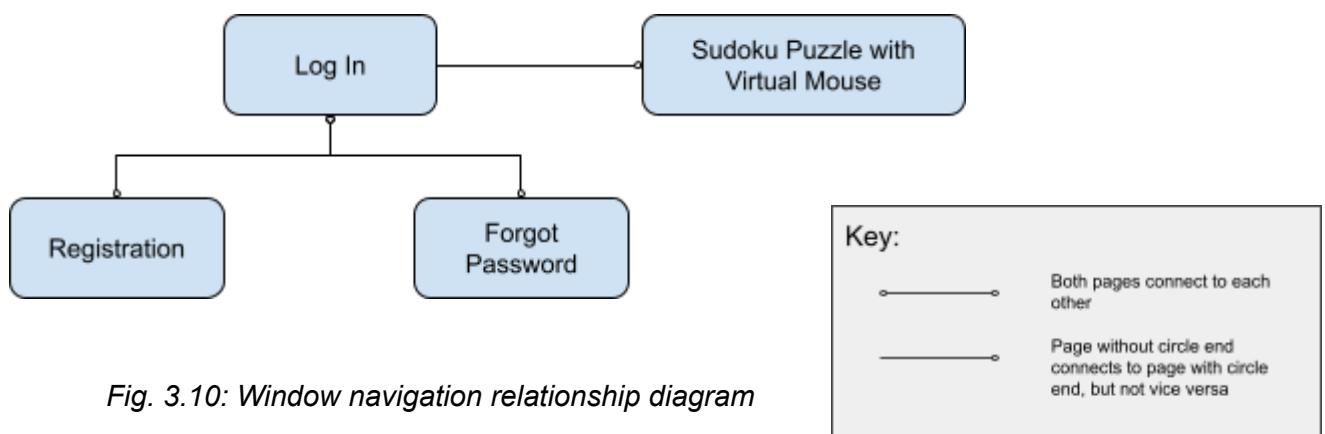


Fig. 3.10: Window navigation relationship diagram

Top-down design

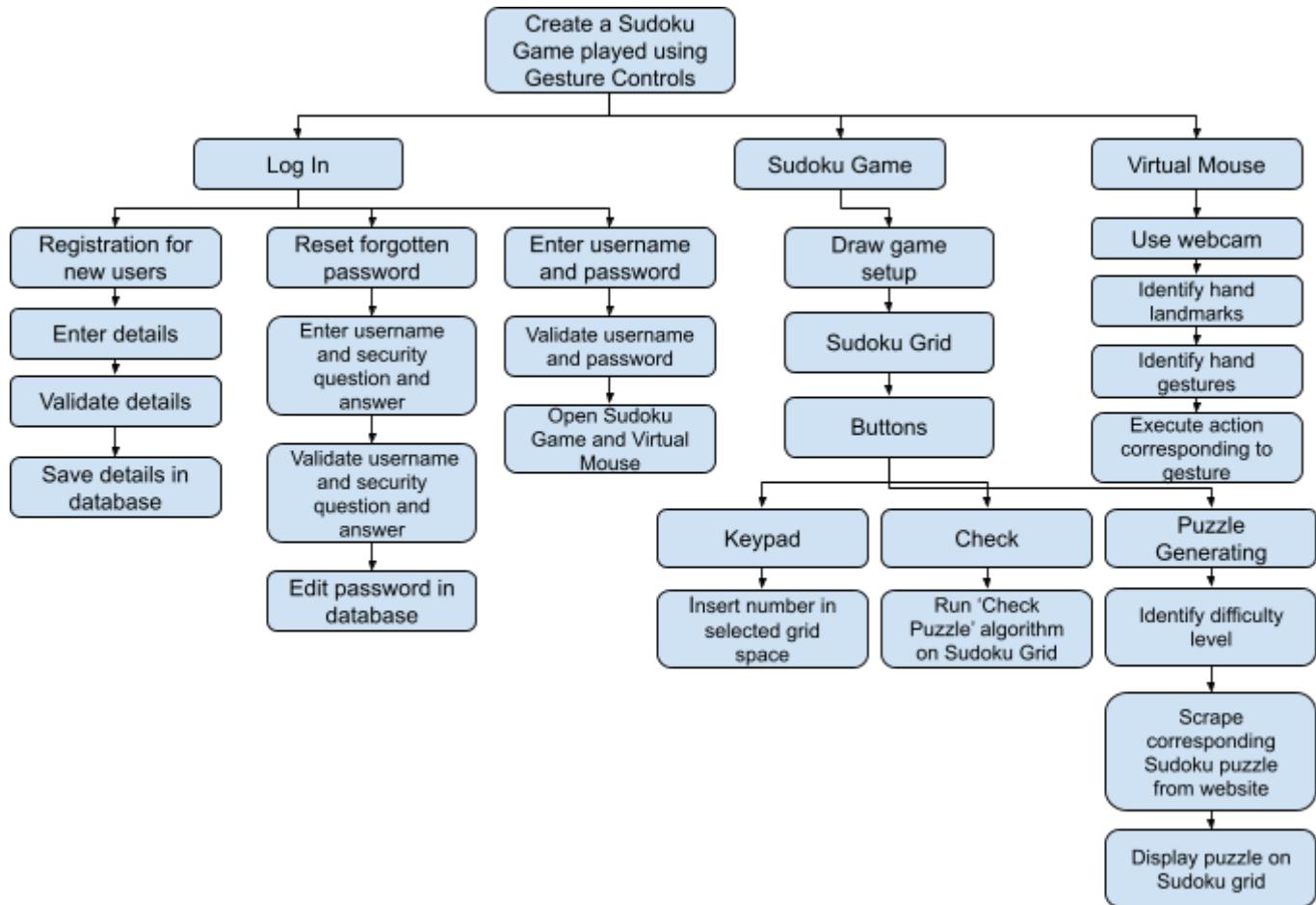


Fig. 3.11: Top-down design diagram showcasing various features/aspects of the system

Data

Inputs and Outputs

Input	Process	Storage	Output
User details: <ul style="list-style-type: none">● First Name● Last Name● Username● Password● Security question● Security answer	Validate input	'Register' table in MySQL database	Message stating registration was successful
Window buttons: <ul style="list-style-type: none">● Login● Register New User● Forgot Password	Tkinter in Python will open a new window corresponding to the button pressed		The windows will open ontop of the existing window
Keypad buttons	It will trigger the program to validate if an unlocked cell is selected	Array of values ('board')	The number corresponding to the button pressed will be displayed in selected cell
Puzzle generating (difficulty level) buttons	Sudoku puzzle from website will be scraped and parsed	Array of values ('data') and array of values ordered in the Sudoku grid layout ('board')	Display puzzle on Sudoku grid
'Check' button	The check puzzle algorithm will check for errors		Shade incorrect cells red
Image from webcam	The hand landmark recognition part of the gesture recognition algorithm		Image overlay on the image from the camera
Hand gestures	The gesture recognition algorithm will identify the gesture		An action that corresponds to the gesture recognised will be performed
Physical mouse/trackpad	The program will recognise mouse		An action that corresponds to the

	input		movement/input of mouse will be performed
Physical keypad	The program will recognise keyboard input		Keyboard input will be displayed on the screen if it is entered in a valid field (eg. integer between 1 and 9 is entered into a selected grid space on the Sudoku grid)

Permanent Data Storage

The main data stored by the program will be the user's registration details. This contains personal information about the user such as their first and last names as well as the answer to their security question (eg. their place of birth, their place of study, etc.), therefore it is important that the system has the functionality to delete the data so that it complies with the GDPR.

For this reason, I have opted to use an SQL database as it optimises data storage and ensures data integrity. MySQL databases are also encrypted to ensure user data is secure. Furthermore, if a future iteration of the program is required to store greater complexity or a greater quantity of information the method of storage need not change.

The data collected will be the user's first name, last name, username, password, the retyped password, the selected security question, and the answer to the aforementioned question. However, there will be validation in place to ensure the retyped password is identical to the password, hence it is redundant to store it as it is repeated data.

In the 2nd and 3rd normalised form (NF), the database splits into multiple tables. Therefore, a foreign key (FK) is used to link the tables. The primary key (PK) of each table is in bold below; it is a unique identifier for each record stored in the database.

Database in 1NF:

First Name	Last Name	Username (PK)	Password	Security Question	Security Answer

Database in 2NF:

User ID (PK)	First Name	Last Name	Username

User ID (FK/PK)	Password

User ID (PK/FK)	Security Question	Security Answer

Database in 3NF:

User ID (PK)	First Name	Last Name	Username	Security Question ID

User ID (FK/PK)	Password

Security Question ID (FK/PK)	Security Question

Security Question ID (FK/PK)	Security Answer

The DDL statements for the tables the program's data will be stored in is:

```
CREATE TABLE Register (
    UserID CHAR (5) NOT NULL,
    FirstName VARCHAR(40) NOT NULL,
    LastName VARCHAR(40) NOT NULL,
    Username VARCHAR(45) NOT NULL,
    SecurityQID INT() NOT NULL,
    PRIMARY KEY (UserID));
```

```
CREATE TABLE User (
    UserID CHAR (5) NOT NULL,
    Password VARCHAR(45) NOT NULL,
    FOREIGN KEY (UserID) REFERENCES Register (UserID),
    PRIMARY KEY (UserID));
```

```
CREATE TABLE Security (
    SecurityQID INT() NOT NULL,
    SecurityQ VARCHAR(45) NOT NULL,
    FOREIGN KEY (SecurityQID) REFERENCES Register (SecurityQID),
    PRIMARY KEY (SecurityQID));
```

```
CREATE TABLE Answer (
    SecurityQID INT() NOT NULL,
    SecurityA VARCHAR(45) NOT NULL,
    FOREIGN KEY (SecurityQID) REFERENCES Security (SecurityQID),
    PRIMARY KEY (SecurityQID));
```

Field	Data Type	Length	Validation Check	Validation Description	Valid Data	Erroneous Data
First Name	String	1 - 40 characters	Presence, length	User must enter a name and it must be 40 characters or less	Ekta	(Blank)
Last Name	String	1 - 40 characters	Presence, length	User must enter a name and it must be 40 characters or less	Nair	(Blank)

Username	String	1 - 45 characters	Lookup, presence, length	User must enter a username that is not used by any existing users and it must be 45 characters or less	ektaN1	(Blank)
Password	String	1 - 45 characters	Presence, length	User must enter a password and it must be 45 characters or less	pAssW2%D	(Blank)
Confirm Password	String	1 - 45 characters	Lookup	Must be identical to the data entered in the 'Password' field	pAssW2%D	(Blank or does is not identical to password)
Security Question	String	1 - 45 characters	List	Must select one of the predefined questions	In what city were you born?	(Blank or question which was not given)
Security Answer	String	1 - 45 characters	Presence, length	User must enter an answer and it must be 45 characters or less	Kuala Lumpur	(Blank)

Internal Data Storage

Field	Data Type	Length	Validation Check	Validation Description	Valid Data	Erroneous Data
Selected Cell	Integer	1 character	Range	Number entered must be an integer between 1 and 9	6	N
Sudoku Puzzle IDs	String	3 characters	Lookup	ID must match one in	f02	X3 (Anything)

				'ids' array		that doesn't start with 'f', followed by 2 digits between 00 and 88)
Sudoku Puzzle values	Integer	1 character	Range	Number entered must be an integer between 1 and 9	9	0

Class Diagram

The class diagram below illustrates an overview of the class structure of the OOP system.

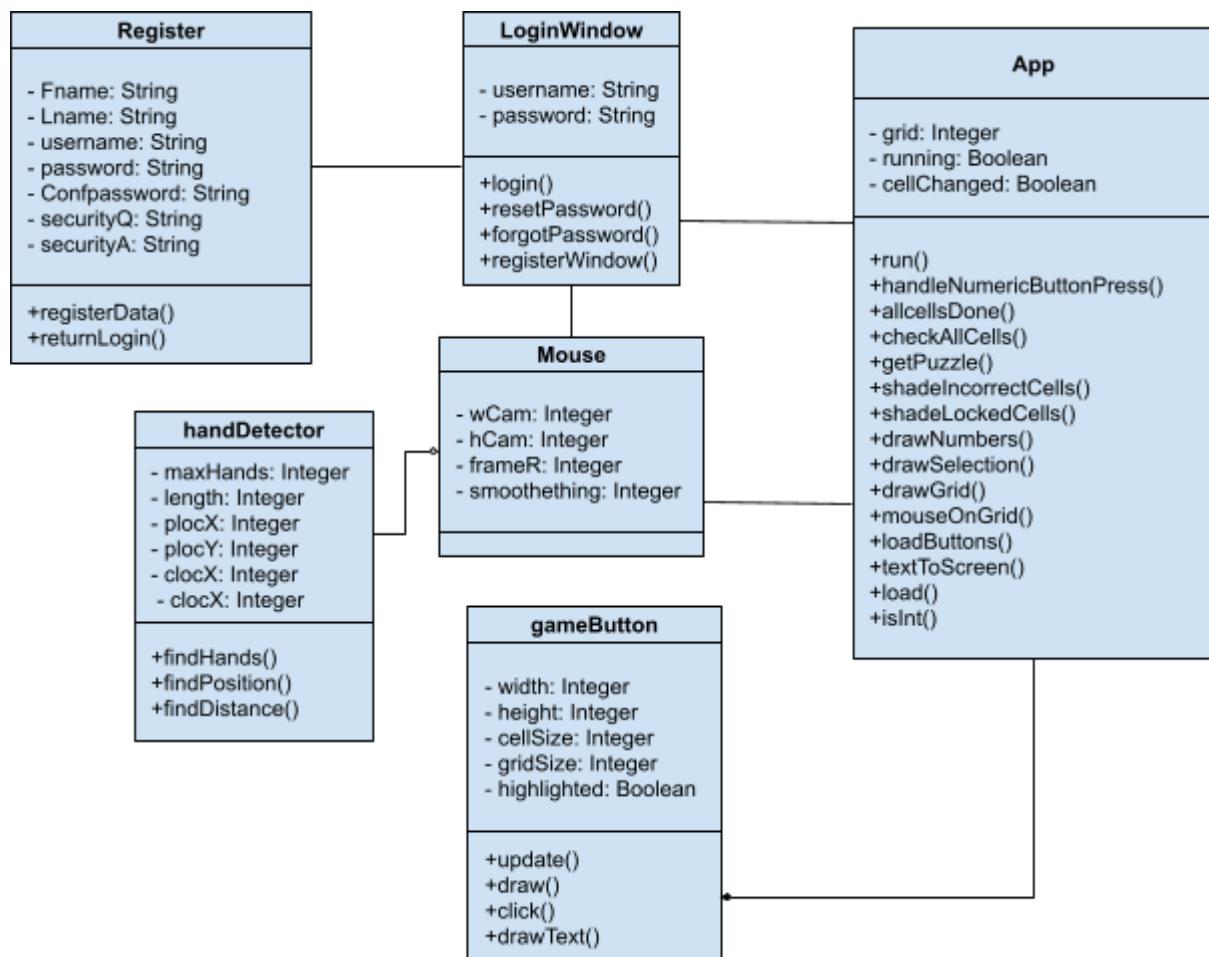
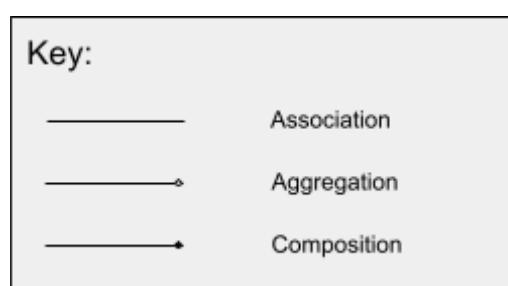


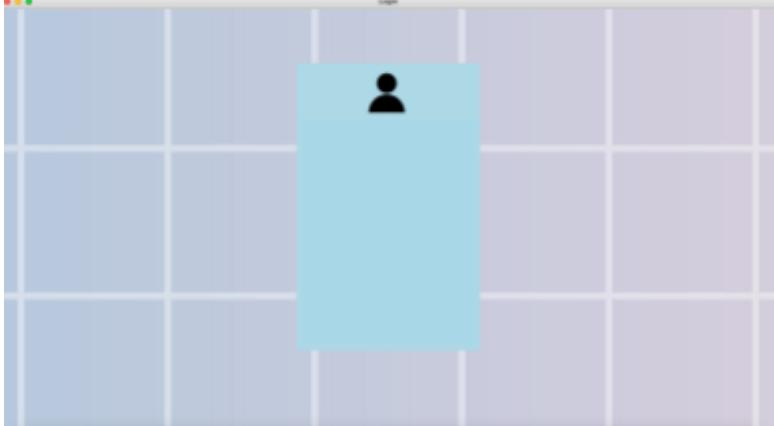
Fig. 3.12: Class diagram highlighting the variables and subroutines associated to each class

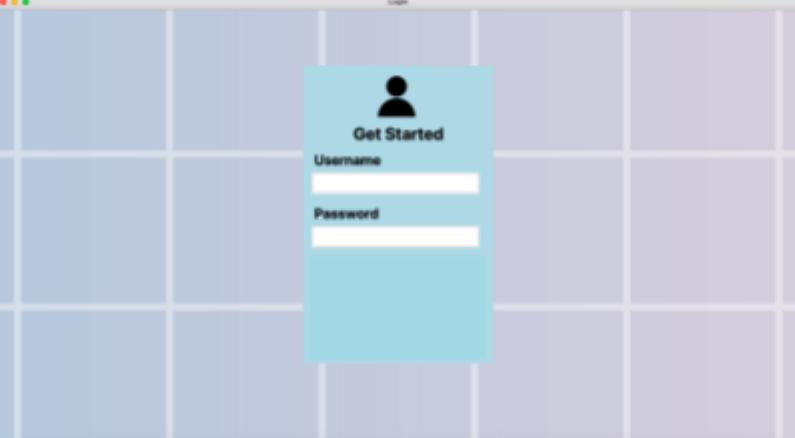


Technical Solution

System Overview

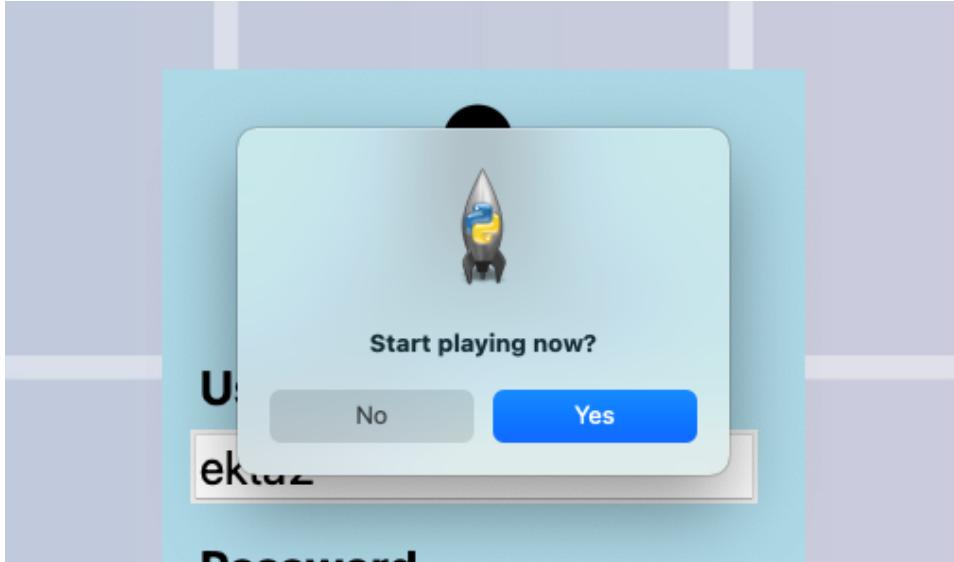
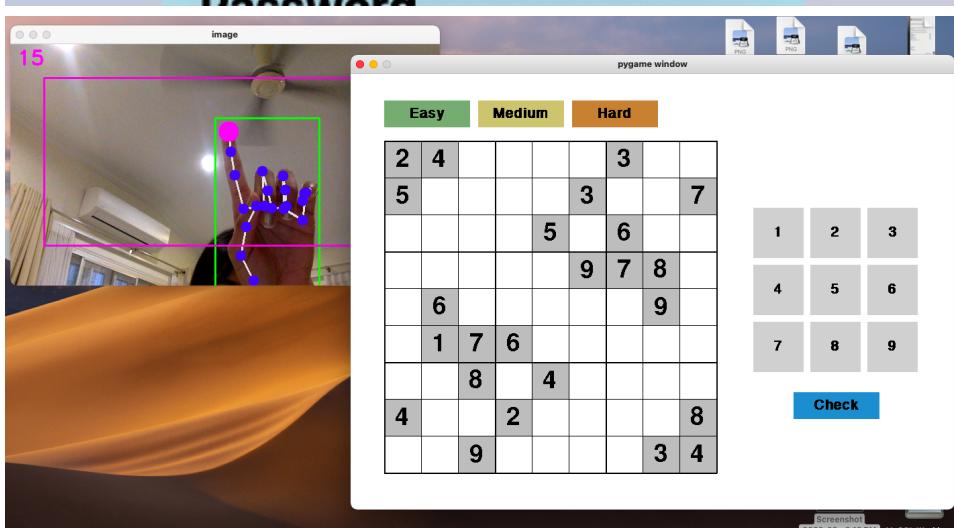
Login System

1.	Initialisation of Login window
code	<pre> class LoginWindow: def __init__(self, root): self.root = root self.root.title("Login") self.root.geometry("1550x800+0+0") self.bg = PIL.ImageTk.PhotoImage(file=r"/Users/rakesharavind/Downloads/ pastel-memphis-blog-banner-template/nea_bg2.jpg") lbl_bg = Label(self.root, image=self.bg) lbl_bg.place(x=0, y=0, relwidth=1, relheight=1) frame = Frame(self.root, bg="light blue") frame.place(x=550, y=100, width=340, height=530) img1 = PIL.Image.open(r"/Users/rakesharavind/Downloads/person_icon3. png") img1 = img1.resize((100,100), PIL.Image.ANTIALIAS) self.photoimage1 = PIL.ImageTk.PhotoImage(img1) lbl_img1 = Label(image=self.photoimage1, bg="light blue", borderwidth=0) lbl_img1.place(x=667, y=105, width=100, height=100) </pre>
output	
explanation	<p>Here the Login window is created; its attributes are defined its components are created alongside it. Specifically referring to the images that make up part of the design of the window, they were found online from non-copyrighted sources and downloaded. Their path URLs (<i>highlighted above</i>) are used by the Pillow (PIL) library to retrieve them and open them on the window. The two images are saved in the same folder saved in the same folder (Downloads) on my computer so that they are easily accessible. The images are stored in different formats (JPG and PNG), so the PIL library is used to easily format and manipulate them to suit the window. I used the resize and Image.ANTIALIAS methods to resize the images without changing the aspect ratio so that</p>

	they do not appear distorted if the user changes the size of their window.
2.	User input boxes and database connection
code	<pre> username = Label(frame, text="Username", font=("veranda", 25, "bold"), fg="black", bg="light blue") username.place(x=15, y=150) self.textuser = ttk.Entry(frame, font=("veranda", 25)) self.textuser.place(x=15, y=190, width=300) . . def login(self): if self.textuser.get() == "" or self.textpass.get() == "": messagebox.showerror("Error", "All fields are required") else: conn = mysql.connector.connect(host="localhost", user="root", password="Test@123", database="mydata") myCursor = conn.cursor() myCursor.execute("select * from register where Username=%s and Password=%s", (self.textuser.get(), self.textpass.get())) row = myCursor.fetchone() if row == None: messagebox.showerror("Error", "Invalid username or password") </pre>
output	
explanation	<p>I added text as Labels to tell the user what to enter into each available input box. I used a bold sans serif font to make it easier for the user to read the text. The input box was created using the <code>ttk.Entry</code> widget. The inputs into these boxes are saved as objects of the <code>LoginWindow</code> class and global variables (<code>self.textuser</code> and <code>self.textpass</code>) so that they can be called at any point in the program to be validated.</p> <p>The input box itself is created using Tkinter, hence it is robust. However, it does allow for an infinite number of characters to be entered into the box, while the maximum number of characters for a username and password allowed is</p>

	<p>45. Exception handling at the Login window is unnecessary as it is already carried out at the Registration window and users will not be able to register an invalid username or password. Therefore, if users try to enter a username or password above the character limit, the program will simply inform them that their username or password is invalid as there is no way it could have been successfully registered.</p>
3.	Registration of accounts for new users
code	<pre> def registerData(self): if self.varFname.get() == "" or self.varLname.get() == "" or self.varUsername.get() == "" or self.varPassword.get() == "" or self.varConfpassword.get() == "" or self.varSecurityQ.get() == "Select" or self.varSecurityA.get() == "": messagebox.showerror("Error", "All fields are required") elif self.varPassword.get() != self.varConfpassword.get(): messagebox.showerror("Invalid Password", "Your password and confirmation password do not match") else: conn = mysql.connector.connect(host="localhost", user="root", password="Test@123", database="mydata") myCursor = conn.cursor() query = ("select * from register where Username=%s") myValue = (self.varUsername.get(),) myCursor.execute(query, myValue) row = myCursor.fetchone() if row != None: messagebox.showerror("Error", "This user already exists. Please try a different username.") else: myCursor.execute("insert into register values(%s,%s,%s,%s,%s,%s)", (self.varFname.get(), self.varLname.get(), self.varUsername.get(), self.varPassword.get(), self.varSecurityQ.get(), self.varSecurityA.get())) conn.commit() conn.close() messagebox.showinfo("Success!", "You have been registered. Login to play.") </pre>

<i>output</i>	<p>The screenshot shows the MySQL Workbench interface. At the top, there's a registration form with fields for First Name, Last Name, Username, Password, Confirm Password, Security Question, and Answer. Below the form are 'Register Now' and 'Login' buttons. At the bottom, there's a table definition for a table named 'register' with columns Fname, Lname, Username, Password, SecurityQ, and SecurityA, each defined as VARCHAR(45).</p>
<i>explanation</i>	<p>If the user clicks the 'Register New User' button on the Login window it will bring them to the Registration window where they can register their new account. This window, similar to the Login window, is linked to the MySQL database by the name of 'mydata', specifically to the table called 'register'. This is done using MySQL Python Connector. When the user clicks the 'Register Now' button, the data the user has input into the input boxes are validated. The queries highlighted above are used with the MyCursor method to retrieve data from 'mydata' in order to perform validation checks and to insert the data into the 'register' table if it is valid, respectively. The variables self.varFname, self.varLname, etc. are also retrieved from the input boxes using the get() function for this to occur.</p> <p>If any erroneous data is entered (eg. fields are left blank; data entered exceeds character limit), error messages, created using Messagebox from Tkinter, pop up to notify the user that the data they have entered is invalid. Likewise, if the user enters valid data, a message will pop up notifying them that they have been successfully registered.</p>
4.	Dynamic opening of new windows
<i>code</i>	<pre> else: openMain = messagebox.askyesno("Play", "Start playing now?") if openMain > 0: self.newWindow = Toplevel(self.root) </pre>

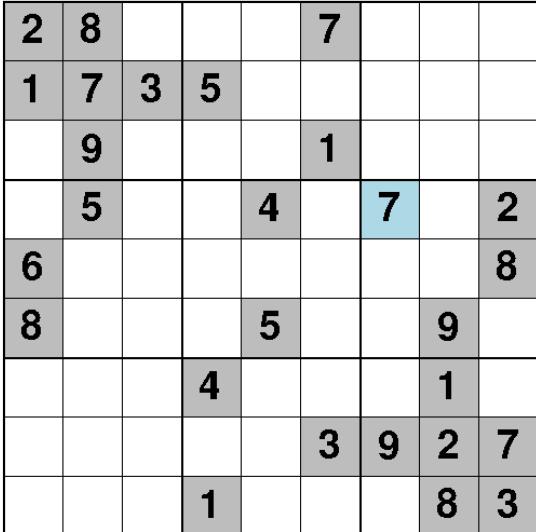
	<pre> self.app = App() self.app.run() thread = threading.Thread(target=self.app.run()) thread.start() self.newWindow = Toplevel(self.root) self.app = Mouse(self.newWindow) </pre>
<i>output</i>	 
<i>explanation</i>	<p>When a valid username and password is entered into the appropriate fields in the login page and the 'Login' button is pressed, a message will pop up asking the user if they wish to start playing. If the user clicks 'No' the message disappears and no further action is taken. Otherwise, if the user clicks Yes, the Sudoku puzzle and Virtual mouse windows will open on top of the Login window, ready for the user to start playing.</p> <p>The Tkinter method <code>Toplevel()</code> is used to open the windows on top of one another and allows them to remain open simultaneously. Based on the user's input the <code>App()</code> and <code>Mouse()</code> classes are instantiated and this OOP model will dynamically open new classes and objects so that the newly opened windows are functional to the user.</p>

	<p>As the Virtual mouse and Sudoku puzzle windows run on different clocks (ie. the Virtual mouse is constantly refreshed while the Sudoku puzzle only refreshes when an action such as a click of a button occurs) they are unable to run simultaneously. The threading library is imported to fix this issue. The threading.Thread() function allows python to execute other code while waiting for a response or input from another task or function. Hence the two windows will be able to run simultaneously and interact with each other, allowing the user to play the game.</p>
--	--

Sudoku Puzzle

1.	<p style="text-align: center;">Initialise Sudoku game</p>
code	<pre>class App: def __init__(self): pygame.init() self.window = pygame.display.set_mode((WIDTH, HEIGHT)) self.running = True self.state = "playing" def run(self): while self.running: if self.state == "playing": self.playing_events() self.playing_update() self.playing_draw() pygame.quit() sys.exit() def playing_events(self): for event in pygame.event.get(): if event.type == pygame.QUIT: self.running = False</pre>

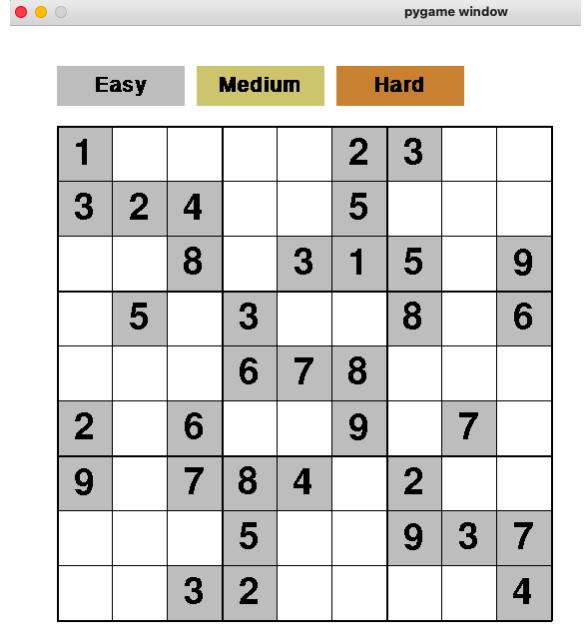
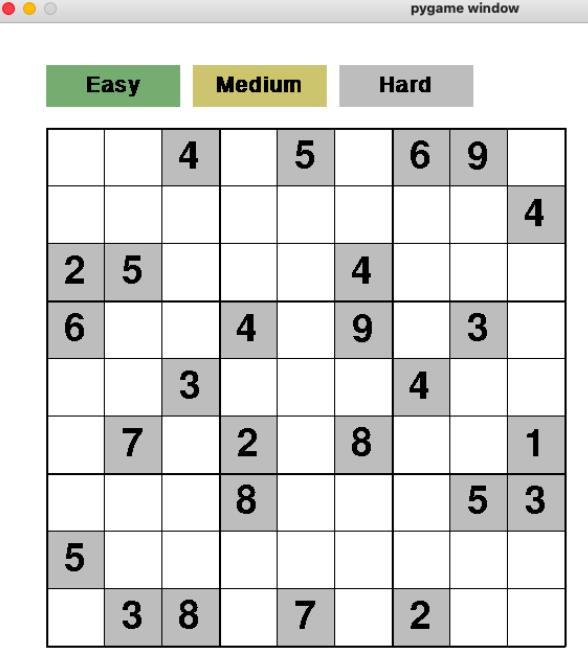
<i>output</i>	
<i>explanation</i>	<p>In the code above the pygame Sudoku puzzle window is created under the class App. The functions self.playing_events(), self.playing_updates(), and self.playing_draw() are immediately called to create the window seen above and to make it responsive to mouse clicks. These functions are in a while loop that repeats infinitely until the user presses the 'close' or 'X' button on the window. This user input changes the state of the at which point self.running = False. This causes the while loop to terminate destroying the class and its objects.</p>
2.	<h3 style="text-align: center;">Cell inputs</h3> <p>code</p> <pre> class App: def __init__(self): pygame.init() self.window = pygame.display.set_mode((WIDTH, HEIGHT)) self.selected = None self.mousePosition = None self.finished = False self.cellChanged = False def playing_events(self): . . . if event.type == pygame.MOUSEBUTTONDOWN: selected = self.mousePosition() if selected: self.selected = selected else: self.selected = None for button in self.playingButtons: if button.highlighted: </pre>

	<pre> button.click() if event.type == pygame.KEYDOWN: if self.selected != None and self.selected not in self.lockedCells: if self.toInt(event.unicode): self.grid[self.selected[1]][self.selected[0]] = int(event.unicode) self.cellChanged = True </pre>
<i>output</i>	
<i>explanation</i>	<p>When the mouse is clicked on a non-locked cell on the grid the cell is highlighted with a light blue colour. Each non-locked cell is treated as a button and stored in the playingButtons array, allowing the function that will colour the cell light blue to be called each time the cell is clicked on. As this is the same function that will be called each time a non-locked cell is clicked the process is made more efficient by using an if loop nested in a for loop.</p> <p>Once the cell is selected the user is able to enter a number by pressing a key on their keyboard or on the onscreen keypad. When a user presses a key on their keyboard, the function <code>toInt()</code> is called to perform a validation check to make sure the key pressed is one of the valid inputs (ie. the numbers 1 to 9). If it is the call is identified and its contents are changed to the number that was input, otherwise, no action occurs. A similar process occurs when the user presses the onscreen keypad. However, it is unnecessary to call the <code>toInt()</code> function to validate the inputs as all the buttons on the onscreen keypad input single predefined digits which are all valid.</p>
3.	Puzzle checking feature
<i>code</i>	<pre> def checkAllCells(self): self.checkRows() self.checkCols() self.checkSmallGrid() def checkSmallGrid(self): for x in range(3): for y in range(3): possibles = [1, 2, 3, 4, 5, 6, 7, 8, 9] </pre>

	<pre> for i in range(3): for j in range(3): xidx = x*3+ i yidx = y*3+j if self.grid[yidx][xidx] in possibles: possibles.remove(self.grid[yidx][xidx]) else: if [xidx, yidx] not in self.lockedCells and [xidx, yidx] not in self.incorrectCells: self.incorrectCells.append([xidx, yidx]) . . . </pre>
<i>output</i>	Web scraping to generate new puzzles

code

```
def getPuzzle(self, difficulty):
    html_doc =
        requests.get("https://nine.websudoku.com/?level={}".format(diffi
culty)).content
    soup = BeautifulSoup(html_doc)
    ids = ['f00', 'f01', 'f02', 'f03', 'f04', 'f05', 'f06',
'f07', 'f08', 'f09', 'f11',
    'f12', 'f13', 'f14', 'f15', 'f16', 'f17', 'f18', 'f20',
'f21', 'f22', 'f23',
    'f24', 'f25', 'f26', 'f27', 'f28', 'f30', 'f31', 'f32',
'f33', 'f34', 'f35',
    'f36', 'f37', 'f38', 'f40', 'f41', 'f42', 'f43', 'f44',
'f45', 'f46', 'f47',
    'f48', 'f50', 'f51', 'f52', 'f53', 'f54', 'f55', 'f56',
'f57', 'f58', 'f60',
    'f61', 'f62', 'f63', 'f64', 'f65', 'f66', 'f67', 'f68',
'f70', 'f71', 'f72',
    'f73', 'f74', 'f75', 'f76', 'f77', 'f78', 'f80', 'f81',
'f82', 'f83', 'f84',
    'f85', 'f86', 'f87', 'f88']
    data = []
    for cid in ids:
        data.append(soup.find('input', id=cid))
    board = [[0 for x in range(9)] for x in range(9)]
    for index, cell in enumerate(data):
        try:
            board[index//9][index%9] = int(cell['value'])
        except:
            pass
    self.grid = board
    self.load()
```

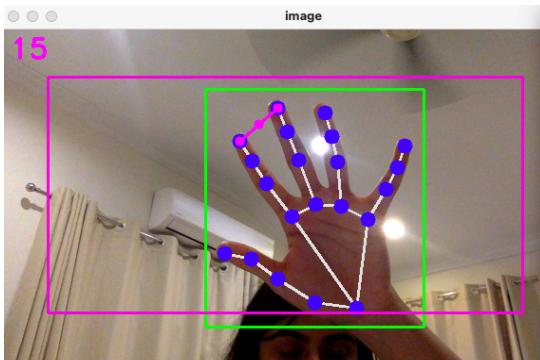
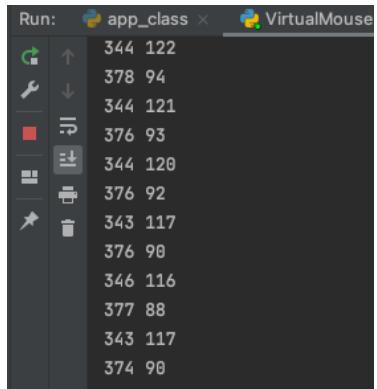
<p><i>output</i></p>	 <p>A screenshot of a pygame window titled "pygame window". At the top, there are three buttons: "Easy" (gray), "Medium" (yellow, highlighted), and "Hard" (orange). Below the buttons is a 9x9 Sudoku grid. Some cells contain numbers: Row 1, Col 1: 1; Row 1, Col 3: 2, 3; Row 2, Col 1: 3, 2, 4; Row 2, Col 3: 5; Row 3, Col 2: 8; Row 3, Col 4: 3, 1; Row 3, Col 5: 5; Row 3, Col 7: 9; Row 4, Col 2: 5; Row 4, Col 3: 3; Row 4, Col 5: 8, 7, 8; Row 4, Col 6: 8; Row 5, Col 1: 2; Row 5, Col 2: 6; Row 5, Col 3: 9; Row 5, Col 4: 7; Row 5, Col 5: 2; Row 5, Col 6: 7; Row 6, Col 1: 9; Row 6, Col 2: 7; Row 6, Col 3: 8, 4; Row 6, Col 4: 2; Row 6, Col 5: 9; Row 6, Col 6: 3; Row 6, Col 7: 7; Row 7, Col 2: 5; Row 7, Col 3: 9, 3; Row 7, Col 4: 7; Row 7, Col 5: 4; Row 8, Col 2: 3; Row 8, Col 3: 2.</p>  <p>A screenshot of a pygame window titled "pygame window". At the top, there are three buttons: "Easy" (green, highlighted), "Medium" (yellow), and "Hard" (gray). Below the buttons is a 9x9 Sudoku grid. Many cells now contain numbers: Row 1, Col 1: 4; Row 1, Col 3: 5; Row 1, Col 5: 6, 9; Row 1, Col 7: 4; Row 2, Col 1: 2, 5; Row 2, Col 3: 4; Row 2, Col 5: 9; Row 2, Col 7: 3; Row 3, Col 1: 6; Row 3, Col 2: 3; Row 3, Col 4: 9; Row 3, Col 6: 5, 3; Row 4, Col 1: 7; Row 4, Col 2: 2; Row 4, Col 3: 8; Row 4, Col 5: 4; Row 4, Col 7: 1; Row 5, Col 1: 8; Row 5, Col 3: 5, 3; Row 5, Col 5: 2; Row 5, Col 7: 5; Row 6, Col 1: 5; Row 6, Col 2: 3, 8; Row 6, Col 4: 7; Row 6, Col 6: 2.</p>
<p><i>explanation</i></p>	<p>Here the requests library is used to send an HTTP request to the website and returns a response object with the apparent encoding of the website, a dictionary or response headers, and a JSON object of the result. The request is formatted so that the program scrapes different parts of the source code based on the user-specified puzzle difficulty.</p> <p>The BeautifulSoup library can then be used to pull data out of the HTML website. It creates a parse tree from the program's source code that can be used to extract data in a hierarchical and more readable manner. The data is parsed and the values of the cells that make up the Sudoku grid are identified</p>

	based on their cell IDs (which are stored in the array ids) and extracted. They are then appended to the array called data. The index of the array is now used to identify each value instead of their cell ID. The index value is used to order the values in a Sudoku board layout. This board is then displayed on the Sudoku grid for the user.
--	---

Virtual Mouse

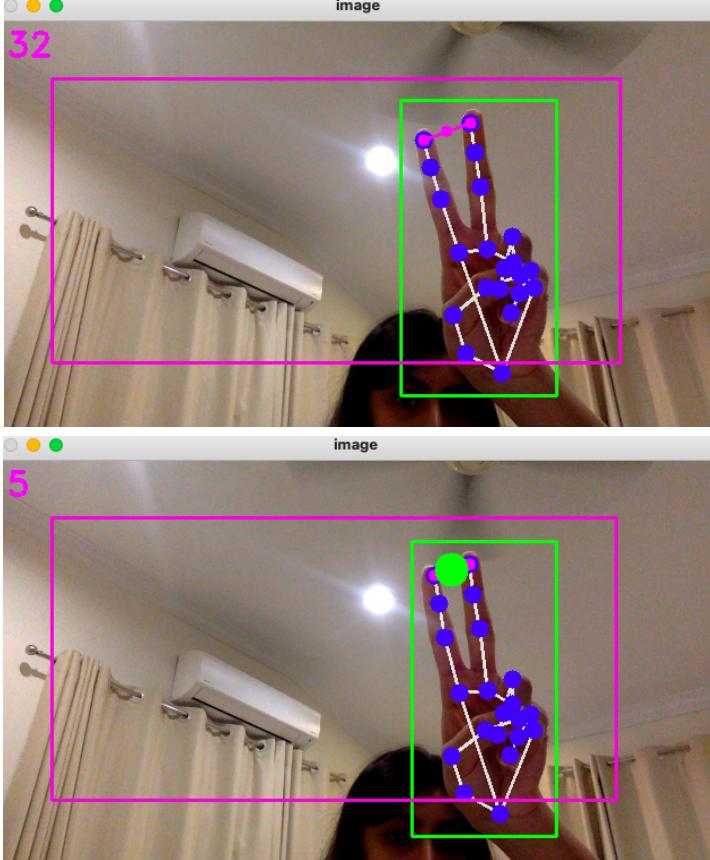
I could not incorporate the Virtual Mouse into the Sudoku Puzzle window as I had initially planned due to the clocks of the 2 programs working differently (as mentioned above). Hence, I gave the virtual Mouse its own window that would open alongside the Sudoku Puzzle window.

The artificial intelligence (AI) virtual mouse program was written as 2 separate modules. The ‘Hand Tracking’ module contains all the functions required to detect the hand and all its landmarks using the webcam, while the ‘Virtual Mouse’ module customises the prior module to suit this specific program. For example, this program only requires the AI image processing algorithms to detect the index and middle fingers on 1 hand. This is done so that the hand gestures the program identifies can easily be customized if necessary. The ‘Hand Tracking’ module may also be used on other similar programs, saving time and resources as the code need not be re-written. In this way, the ‘Virtual Mouse’ module acts as the child class and inherits attributes from the ‘Hand Tracking’ module.

1.	Detecting hand and important hand landmarks
<i>code</i>	<pre>while True: success, img = cap.read() img = detector.findHands(img) lmList, bbox = detector.findPosition(img) if len(lmList) != 0: x1, y1 = lmList[8][1:] x2, y2 = lmList[12][1:] print(x1, y1) print(x2, y2)</pre>
<i>output</i>	 <div style="display: flex; justify-content: space-between;"> <div style="flex: 1;">  <pre>Run: app_class x VirtualMouse 344 122 378 94 344 121 376 93 344 120 376 92 343 117 376 90 346 116 377 88 343 117 374 90 343 117</pre> </div> </div>

<i>explanation</i>	<p>While the program is running, the AI image processing algorithms are used on the webcam feed to identify the hand if it is visible and to find the coordinates of the important hand landmarks.</p> <p>In this program, since the index finger being up alone and alongside the middle finger are the main gestures used to control the program, they are the main fingers being tracked. Index 8 and 12 are used to identify the tips of the index and middle fingers respectively and their coordinates are stored in the local variables x1, y1, x2, and y2, though frequently updated as the window refreshes. The use of [1:] beside the index, shows that the program is specifically tracking the second and top knuckle of these 2 fingers. This is to identify if the finger is up or not, as the gesture controls depend on that.</p>
2.	Image conversion into RGB colour space
<i>code</i>	<pre>def findHands(self, img, draw=True): img = cv2.resize(img, None, fx=0.5, fy=0.5, interpolation=cv2.INTER_AREA) imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) self.results = self.hands.process(imgRGB)</pre>
<i>output</i>	
<i>explanation</i>	<p>The BGR image obtained by the webcam is converted into an RGB image format using the method cv2.COLOR_BGR2RGB. This is because the RGB format is a simple additive colour model that makes identifying key areas of the image (in this case the hand) much easier. This image format is not shown to the user and is simply used by the program to easily identify the hand landmarks in the camera feed.</p> <p>The image obtained by the webcam is resized, using the function resize(), by a scale factor of fx on the x axis and fy on the y axis to fit the window. The method INTER_AREA interpolates the data of the image in order to allow it to be resized.</p>
3.	Overlay drawings on hand
<i>code</i>	<pre>def findPosition(self, img, handNo=0, draw=True): xList = [] yList = [] bbox = [] self.lmList = [] if self.results.multi_hand_landmarks: myHand = self.results.multi_hand_landmarks[handNo] for id, lm in enumerate(myHand.landmark): h, w, c = img.shape cx, cy = int(lm.x * w), int(lm.y * h) xList.append(cx) yList.append(cy) self.lmList.append([id, cx, cy]) if draw: cv2.circle(img, (cx, cy), 8, (255, 0, 0), cv2.FILLED) xmin, xmax = min(xList), max(xList) ymin, ymax = min(yList), max(yList)</pre>

	<pre> bbox = xmin, ymin, xmax, ymax if draw: cv2.rectangle(img, (bbox[0]-20, bbox[1]-20), (bbox[2]+20, bbox[3]+20), (0, 255, 0), 2) return self.lmList, bbox </pre>
<i>output</i>	
<i>explanation</i>	<p>The local variables <code>xList</code> and <code>yList</code> are arrays that store the coordinates of the tip of the index finger. They then use these coordinates to draw a purple circle on top of the tip of the index finger to convey to the user that that is the main point on the hand being tracked as that is the point whose movement corresponds with the movement of the mouse.</p> <p>The minimum and maximum of these stored coordinates are found and used as reference points for the corners of the green boundary box that surrounds the hand. The reference points are in the middle of the fingers, so padding of 20 pixels is added so that the boundary box surrounds the whole hand and doesn't cut off the tips of the fingers.</p>
4.	Identify gestures
<i>code</i>	<pre> def findDistance(self, p1, p2, img, draw=True, r=5, t=2): x1, y1 = self.lmList[p1][1], self.lmList[p1][2] x2, y2 = self.lmList[p2][1], self.lmList[p2][2] cx, cy = (x1+x2)//2, (y1+y2)//2 . . if fingers[1] == 1 and fingers[2] == 1: length, img, lineInfo = detector.findDistance(8, 12, img) if length <40: cv2.circle(img, (lineInfo[4], lineInfo[5]), 15, (0, 255, 0), cv2.FILLED) autopy.mouse.click() </pre>

<i>output</i>	
<i>explanation</i>	<p>Here the variables cx and cy are the coordinates of the median of the coordinates of the tips of the index and middle fingers that were previously obtained. A line is drawn between them to tell the user that the distance between the fingers is being tracked.</p> <p>If the distance between the index and middle fingers is less than 40 pixels, the program identifies this as the gesture that clicks the mouse. A large green circle is drawn over the point with coordinates cx and cy to tell the user that the mouse has been clicked. This action is achieved using the <code>mouse.click()</code> function from the AutoPy library.</p>

Sectioned Code

For detailed code, see Appendix file

Testing

Test Plan

The testing that will occur will be twofold: alpha and beta testing.

During alpha testing, to ensure that the system will be able to correctly handle all user inputs (both valid and invalid) as well as correctly navigate between windows and execute algorithms correctly, I have planned several testing methods seen below.

After which the system will go through beta testing where it will be trialled by a small group of prospective users and evaluated based on their opinions of usability and how well the system meets its objectives.

Test Data

The use of different types of test data is imperative to ensure that the system keeps performing smoothly even in case the end-user enters invalid information while using it or chooses to do that deliberately in order to toy with the system.

The three types of test data I will use are typical, erroneous, and boundary data.

Type	Expansion
Typical (T)	<p>Also called 'normal data', these are the types of inputs that are valid and one would expect to be entered.</p> <p>Typical data helps to verify the system functions by checking if the expected output is received when a valid input is provided.</p>
Erroneous (E)	<p>This is the type of data that is invalid and should be rejected with a relevant error message to notify that the data is improper to function.</p> <p>This type of data is used to test if the system is able to handle exceptions and is working correctly or not.</p>
Boundary (B)	<p>This type of data helps in removing the defects that are connected while processing the boundary values. The data included in this data type is a combination of extreme values (eg. the lowest and highest values in a range or the first and last items in an array).</p>

Input & Output Testing

Test No.	Description	Test Data (T/E/B)	Expected Outcome	Actual Outcome	Comments and Corrective Actions
1.	Check that user can register if all fields are completed correctly	(T) First Name: John Last Name: Smith Username: Sm1thJ Password: t£Stp4SS Confirm Password: t£Stp4SS Security Question: What was the make of your first car? Security Answer: Honda Click 'Register Now' button	1. A message confirming the user has been registered 2. User details are added to database		
2.	Check that all fields are required for the user to register	(E) First Name: John Last Name: (Blank) Username: Sm1thJ Password: t£Stp4SS Confirm Password: t£Stp4SS Security Question: What was the make of your first car? Security Answer: Honda Click 'Register Now' button	1. Error message appears asking the user to ensure all fields are filled 2. No user details are added to database		
3.	Check that Password and Confirm Password match	(E) First Name: John Last Name: Smith Username: Sm1thJ Password: t£Stp4SS Confirm Password: p4SS Security Question: What was the make of your first car? Security Answer:	1. Error message appears telling the user that the Password and Confirm Password fields must match 2. No user		

		Honda Click 'Register Now' button	details are added to database		
4. (a)	Check that user is unable to register if input data is longer than character limit	(B) First Name: John Last Name: Smith Username: SmithJonathan2004 UsernameTestAcom puterscience Password: t£Stp4SS Confirm Password: t£Stp4SS Security Question: What was the make of your first car? Security Answer: Honda Click 'Register Now'	1. A message confirming the user has been registered 2. User details are added to the database		
		(E) First Name: John Last Name: Smith Username: Sm1thJ Password: t£Stp4SS Confirm Password: t£Stp4SS Security Question: What was the make of your first car? Security Answer: I'm pretty sure it was a Honda although I could be mistaken Click 'Register Now'	1. Error message appears telling the user that the data entered exceeds the character limit 2. No user details are added to database		
5.	User is allowed to reset the password if all necessary fields are filled correctly	(T) Username: ekta2 Click 'Forgot Password?' button Security Question: In which city were you born? Security Answer: KL New Password: TestPass2	1. Message confirming user's password has been reset 2. Password has been replaced with new password in database		

		Click 'Reset Password' button			
6.	Check that username must be entered in the Login window for the password to be reset	(E) Username: (Blank) Click 'Forgot Password?' button	Error message appears telling the user to first enter username		
7.	Check that username entered in the Login window must be valid	(E) Username: InvalidUser Click 'Forgot Password?' button	Error message appears telling the user to enter a valid username		
8.	All fields are required for the user to reset their password	(E) Username: ekta2 Click 'Forgot Password?' button Security Question: Select Security Answer: KL New Password: TestPass2 Click 'Reset Password' button	1. Error message appears asking the user to ensure all fields are filled 2. No user details are updated in database		
9.	Check that if the user enters a valid username and password the Sudoku and Virtual Mouse windows will open on top of the Login window	(T) Username: ektaanair Password: Test@123 Click 'Login' button Click 'Yes' to 'Start playing now?' message	1. Sudoku Puzzle window should open first, followed by the Virtual Mouse window to the left of it. 2. Both windows should open on top of the Login window		
10.	Check that	(B)	Only one hand		

	Virtual Mouse detects one hand only	Place 2 hands in the camera's view	should be detected and drawings indicating the hand landmarks should be overlaid on top of the image of the hand in the camera feed		
11.	Moving the index finger while it is raised moves the cursor	(T) Raise the index finger and move it in a beeline from one corner of the purple box to another	Mouse should follow the movement of the index finger from one corner of the screen to the other		
12.	Mouse should click when index and middle fingers are raised with a short distance between them	(T) Raise index and middle finger with a short distance between them when the mouse is on a window behind the Virtual Mouse window	Mouse should perform a 'click', causing the clicked window to jump in front of the Virtual Mouse window Green circle should appear between the index and middle fingers		
13.	No other gestures cause any action to occur	(E) Raise only pinky finger and move it in a circle within the purple box on the Virtual Mouse window	No action		
14.	Check that the frames per second of the camera are displayed	(T) Open the Virtual Mouse window and look at the top left corner of the window	<ol style="list-style-type: none"> 1. Purple text should display the numerical value of the frames per second of the camera in use 2. Frames per second should be continuously updating 		
15.	Check that	(T)	Cell should turn a		

	Virtual Mouse is able to interact with Sudoku Puzzle	Use the Virtual Mouse to click on an empty cell on the Sudoku grid	light blue colour, indicating that it has been selected		
16.	New Sudoku puzzles are able to be scraped and displayed	(T) Click on each of the buttons 'Easy', 'Medium' and 'Hard'	A new puzzle of appropriate difficulty should be generated each time a button is clicked		
17.	Check that locked cell values cannot be changed	(E) Click on a locked cell, then type '5'	No action		
18.	Check that the onscreen keypad works	(T) Select an unlocked cell, then click the button '1'	The digit 1 should be entered into the selected cell		
(a)	Invalid inputs cannot be entered into selected cells	(B) Select an unlocked cell, then click the button '9'	The digit 9 should be entered into the selected cell		
		(E) Select an unlocked cell, then type '0R'	Selected cell remains unchanged		
20.	Check that checkAllCells() function works	(T) Click the 'Check' button	Non-locked cells with the incorrect input should be shaded red and correct cells should remain white		
		(T) Complete the Sudoku puzzle correctly	Message should appear congratulating the user on successfully completing the puzzle		

Navigation Testing

The system was coded as individual windows, hence it is necessary that the user is able to navigate between the windows in order for the system to function properly. In order to test

this, I have created a table that can be filled in with ticks and crosses to test if the navigation between each window works as expected.

It is worth noting that not all windows are meant to be connected to each other. This is shown by the greyed-out cells in the table and the elimination of the Sudoku Puzzle and Virtual Mouse windows from the ‘Navigation from →’ row. The Sudoku Puzzle and Virtual Mouse windows need not connect to any other as they would only appear once the user has successfully logged in and therefore has no use for the other windows.

Navigating from →	Login	Registration	Forgot Password
Navigating to ↓			
Login			
Registration			
Forgot Password			
Sudoku Puzzle			
Virtual Mouse			

Alpha Test

Input & Output Testing

The table below is an overview of the basic input and output test conducted on the system to ensure the expected outcome of every action is true to that outcome in reality. After the tests, all necessary corrective action will be taken before retests are done. This is to ensure the expected outcome is a reality.

Test No.	Description	Test Data (T/E/B)	Expected Outcome	Actual Outcome	Comments and Corrective Actions
1.	Check that user can register if all fields are completed correctly	(T) First Name: John Last Name: Smith Username: Sm1thJ Password: t£Stp4SS Confirm Password: t£Stp4SS Security Question:	1. A message confirming the user has been registered 2. User details are added to database	As expected	None required

		<p>What was the make of your first car? Security Answer: Honda</p> <p>Click 'Register Now' button</p>			
2.	Check that all fields are required for the user to register	<p>(E)</p> <p>First Name: John Last Name: (Blank) Username: Sm1thJ Password: t£Stp4SS Confirm Password: t£Stp4SS Security Question: What was the make of your first car? Security Answer: Honda</p> <p>Click 'Register Now' button</p>	<ul style="list-style-type: none"> 1. Error message appears asking the user to ensure all fields are filled 2. No user details are added to the database 	As expected	None required
3.	Check that Password and Confirm Password match	<p>(E)</p> <p>First Name: John Last Name: Smith Username: Sm1thJ Password: t£Stp4SS Confirm Password: p4SS Security Question: What was the make of your</p>	<ul style="list-style-type: none"> 1. Error message appears telling the user that the Password and Confirm Password fields must match 2. No user details are added to database 	As expected	None required

		<p>first car? Security Answer: Honda</p> <p>Click 'Register Now' button</p>			
4. (a)	User is unable to register if input data is longer than character limit	<p>(B)</p> <p>First Name: John Last Name: Smith Username: SmithJonathan2004 Password: t£Stp4SS Confirm Password: t£Stp4SS Security Question: What was the make of your first car? Security Answer: Honda</p> <p>Click 'Register Now'</p>	<ul style="list-style-type: none"> 1. A message confirming the user has been registered 2. User details are added to the database 	As expected	None required
(b)		<p>(E)</p> <p>First Name: Jane Last Name: Doe Username: DoeJane Password: t£Stp4SS Confirm Password: t£Stp4SS Security Question: What was the make of your first car?</p>	<ul style="list-style-type: none"> 1. Error message appears telling the user that the data entered exceeds the character limit 2. No user details are added to the database 	<ul style="list-style-type: none"> 1. No error message appears 2. As expected 	Create a message using Tkinter in the registerData() function to notify the user of error

		<p>Security Answer: I'm pretty sure it was a Honda although I could be mistaken</p> <p>Click 'Register Now'</p>			
5.	User is allowed to reset their password if all necessary fields are filled correctly	<p>(T)</p> <p>Username: ekta2</p> <p>Click the 'Forgot Password?' button</p> <p>Security Question: In which city were you born?</p> <p>Security Answer: KL</p> <p>New Password: TestPass2</p> <p>Click 'Reset Password' button</p>	<ol style="list-style-type: none"> 1. A message confirming the user's password has been reset 2. Password has been replaced with a new password in the database 	As expected	None required
6.	Check that username must be entered in the Login window for the password to be reset	<p>(E)</p> <p>Username: (Blank)</p> <p>Click 'Forgot Password?' button</p>	Error message appears telling the user to first enter username	As expected	None required
7.	Check that username entered in the Login window must be valid	<p>(E)</p> <p>Username: InvalidUser</p> <p>Click 'Forgot Password?' button</p>	Error message appears telling the user to enter a valid username	As expected	None required
8.	All fields are	(E)	1. Error	As expected	None required

	required for the user to reset their password	Username: ekta2 Click 'Forgot Password?' button Security Question: Select Security Answer: KL New Password: TestPass2 Click the 'Reset Password' button	message appears asking the user to ensure all fields are filled 2. No user details are updated in database		
9.	Check that if the user enters a valid username and password the Sudoku and Virtual Mouse windows will open on top of the Login window	(T) Username: ektaanair Password: Test@123 Click 'Login' button Click 'Yes' to 'Start playing now?' message	1. Sudoku Puzzle window should open first, followed by the Virtual Mouse window to the left of it. 2. Both windows should open on top of the Login window	As expected	None required
10.	Check that Virtual Mouse detects one hand only	(B) Place 2 hands in the camera's view	Only one hand should be detected and drawings indicating the hand landmarks should be overlaid on top of the image of the hand in the camera feed	As expected	None required
11.	Moving the index finger while it is raised moves the cursor	(T) Raise the index finger and move it in a beeline from	Mouse should follow the movement of the index finger from one corner of the screen to the other	As expected	None required

		one corner of the purple box to another			
12.	The mouse should click when index and middle fingers are raised with a short distance between them	(T) Raise index and middle finger with a short distance between them when the cursor is on a window behind the Virtual Mouse window	1. The mouse should perform a 'click', causing the clicked window to jump in front of the Virtual Mouse window 2. A green circle should appear between the index and middle fingers	As expected	None required
13.	No other gestures cause any action to occur	(E) Raise pinky finger and move it in a circle within the purple box on the Virtual Mouse window	No action	Performs as expected. However, a minor issue may be that the program tracks the index and middle finger despite any other fingers that may be raised alongside it. Therefore, even though the appropriate gesture to move the mouse is to raise only the index finger, if the index and pinky fingers are both raised the mouse will still move.	This issue seems unnecessary to fix given my prospective users. Individuals with motor impairments may not have the dexterity to close all of their fingers and only raise one. This feature will allow users to continue using gesture controls in spite of this. In future iterations, the gesture controls could be made more accurate by tracking other hand landmarks. However, for the purposes of this system, no corrective action is required.
14.	Check that the	(T)	1. Purple text	As expected	None required

	frames per second of the camera is displayed	Open the Virtual Mouse window and look at the top left corner of window	should display the numerical value of the frames per second of the camera in use 2. Frames per second should be continuously updating		
15.	Check that Virtual Mouse is able to interact with Sudoku Puzzle	(T) Use the Virtual Mouse to click on an empty cell on the Sudoku grid	Cell should turn a light blue colour, indicating that it has been selected	As expected	None required
16.	New Sudoku puzzles are able to be scraped and displayed	(T) Click on each of the buttons 'Easy', 'Medium' and 'Hard'	A new puzzle of the appropriate difficulty should be generated each time a button is clicked	The puzzle is displayed on the grid as expected, however, a new instance of the board is created every time each of the buttons is pressed. This results in multiple versions of the board being loaded unnecessarily.	Reset the playingButton array to an empty array at the beginning of the load() function
17.	Check that locked cell values cannot be changed	(E) Click on a locked cell, then type '5'	No action	As expected	None required
18.	Check that the onscreen keypad works	(T) Select an unlocked cell, then click the button '1'	The digit 1 should be entered into the selected cell	As expected	None required
19. (a)	Invalid inputs cannot be entered into selected cells	(B) Select an unlocked cell,	The digit 9 should be entered into the selected cell	As expected	None required

		then type '9'			
(b)		(E) Select an unlocked cell, then type '0R'	Selected cell remains unchanged	As expected	None required
20. (a)	Check that checkAllCells() function works	(T) Click the 'Check' button	Non-locked cells with the incorrect input should be shaded red and correct cells should remain white	As the function compares the values in the cells to an array of possible values, the first cell checked (ie. top-left cell) will always be considered correct even if it is not as no other values have been eliminated from the possibles array yet.	Create another loop in each function that will iterate through the entire row/column/3 by 3 grid once more when it finds a number that is not in the possibles array to ensure that number is not already in that row/column/3 by 3 grid. If it finds an identical number it will append that number to the incorrectCells array instead.
(b)		(T) Complete the Sudoku puzzle	<ol style="list-style-type: none"> If completed correctly, a message should appear congratulating the user on successfully completing the puzzle If completed incorrectly, the output should be the same as 20(a) 	As expected	None required

Note: Test 20(b) was completed after the 20(a) retest, hence the outcome of test 20(b) was not compromised by the initial unexpected outcome of test 20(a).

Navigation Testing

Key:

- ✓ Navigation successful
- ✗ Navigation unsuccessful

Navigating from →	Login	Registration	Forgot Password
Navigating to ↓			
Login		✓	✓
Registration	✓		
Forgot Password	✓		
Sudoku Puzzle	✓		
Virtual Mouse	✓		

The diagram below shows how all windows in the system link to each other. This can be compared with the Page Navigation and System Flowchart diagrams on page 26.

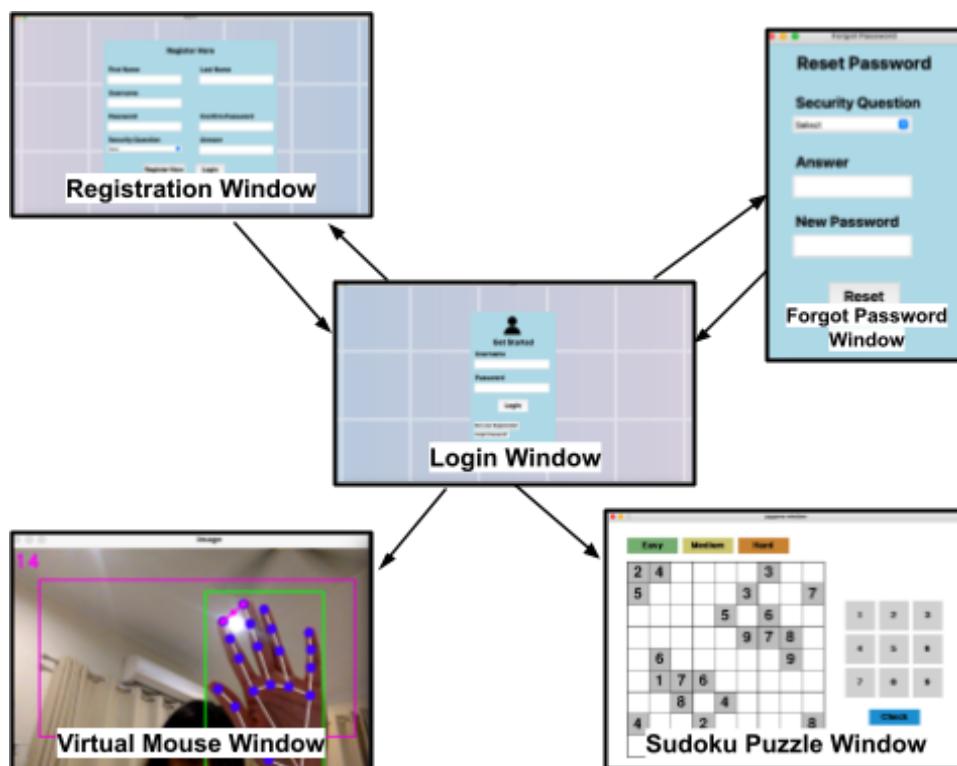


Fig. 4.1: All windows in the system with arrows indicating the direction of possible navigation between windows

Retests

All code highlighted in light blue is code that had been added to fix the issue that caused unexpected outcomes in the original tests.

Test No.	Original Code	Updated Code
4. (b)	<pre> def registerData(self): if self.varFname.get() == "" or self.varLname.get()=="" or self.varUsername.get()=="" or self.varPassword.get()=="" or self.varConfpassword.get()=="" or self.varSecurityQ.get()=="Select" or self.varSecurityA.get()=="": messagebox.showerror("Error", "All fields are required") elif self.varPassword.get() != self.varConfpassword.get(): messagebox.showerror("Invalid Password", "Your password and confirmation password do not match") else: conn = mysql.connector.connect(host="localhost", user="root", password="19Feb2004!", database="mydata") myCursor = conn.cursor() query = ("select * from register where Username=%s") myValue = (self.varUsername.get(),) myCursor.execute(query, myValue) row = myCursor.fetchone() if row != None: messagebox.showerror("Error", "This user already exists. Please try a different username.") else: myCursor.execute("insert into register values(%s,%s,%s,%s,%s,%s)", (self.varFname.get(), self.varLname.get(), self.varUsername.get(), self.varPassword.get(), self.varSecurityQ.get(), self.varSecurityA.get())) conn.commit() conn.close() messagebox.showinfo("Success!", "You have been registered. Login to play.") </pre>	<pre> def registerData(self): if self.varFname.get() == "" or self.varLname.get()=="" or self.varUsername.get()=="" or self.varPassword.get()=="" or self.varConfpassword.get()=="" or self.varSecurityQ.get()=="Select" or self.varSecurityA.get()=="": messagebox.showerror("Error", "All fields are required") elif self.varPassword.get() != self.varConfpassword.get(): messagebox.showerror("Invalid Password", "Your password and confirmation password do not match") else: conn = mysql.connector.connect(host="localhost", user="root", password="19Feb2004!", database="mydata") myCursor = conn.cursor() query = ("select * from register where Username=%s") myValue = (self.varUsername.get(),) myCursor.execute(query, myValue) row = myCursor.fetchone() if row != None: messagebox.showerror("Error", "This user already exists. Please try a different username.") else: myCursor.execute("insert into register values(%s,%s,%s,%s,%s,%s)", (self.varFname.get(), self.varLname.get(), self.varUsername.get(), self.varPassword.get(), self.varSecurityQ.get(), self.varSecurityA.get())) myCursor = conn.cursor() </pre>

		<pre> myCursor.execute("select * from register where Fname=%s and Lname=%s and Username=%s and " "Password=%s and SecurityQ=%s and SecurityA=%s"), (self.varFname.get(), self.varLname.get(), self.varUsername.get(), self.varPassword.get(), self.varSecurityQ.get(), self.varSecurityA.get()) row = myCursor.fetchone() if row != None: messagebox.showerror("Error", "This information you entered exceeds the character limit of this program") else: conn.commit() conn.close() messagebox.showinfo("Success!", "You have been registered. Login to play.") </pre>
16.	<pre> def load(self): self.loadButtons() self.lockedCells = [] self.incorrectCells = [] self.finished = False </pre>	<pre> def load(self): self.playingButtons = [] self.loadButtons() self.lockedCells = [] self.incorrectCells = [] self.finished = False </pre>
20. (a)	<pre> def checkAllCells(self): self.checkRows() self.checkCols() self.checkSmallGrid() def checkSmallGrid(self): for x in range(3): for y in range(3): possibles = [1,2,3,4,5,6,7,8,9] for i in range(3): for j in range(3): xidx = x*3+i yidx = y*3+j if self.grid[yidx][xidx] in possibles: possibles.remove(self.grid[yidx][xidx]) else: if [xidx, yidx] not in self.lockedCells and [xidx, yidx] not in self.incorrectCells: self.incorrectCells.append([xidx, yidx]) def checkRows(self): for yidx, row in enumerate(self.grid): </pre>	<pre> def checkAllCells(self): self.checkRows() self.checkCols() self.checkSmallGrid() def checkSmallGrid(self): for x in range(3): for y in range(3): possibles = [1,2,3,4,5,6,7,8,9] for i in range(3): for j in range(3): xidx = x*3+i yidx = y*3+j if self.grid[yidx][xidx] in possibles: possibles.remove(self.grid[yidx][xidx]) else: if [xidx, yidx] not in self.lockedCells and [xidx, yidx] not in self.incorrectCells: self.incorrectCells.append([xidx, yidx]) </pre>

```

possibles = [1,2,3,4,5,6,7,8,9]
for xidx in range(9):
    if self.grid[yidx][xidx] in possibles:
        possibles.remove(self.grid[yidx][xidx])
    else:
        if [xidx, yidx] not in self.lockedCells and [xidx, yidx] not in self.incorrectCells:
            self.incorrectCells.append([xidx, yidx])

def checkCols(self):
    for xidx in range(9):
        possibles = [1,2,3,4,5,6,7,8,9]
        for yidx, row in enumerate(self.grid):
            if self.grid[yidx][xidx] in possibles:
                possibles.remove(self.grid[yidx][xidx])
            else:
                if [xidx, yidx] not in self.lockedCells and [xidx, yidx] not in self.incorrectCells:
                    self.incorrectCells.append([xidx, yidx])

```

```

if [xidx, yidx] in self.lockedCells:
    for k in range(3):
        for l in range(3):
            xidx2 = x*3+k
            yidx2 = y*3+l

            if self.grid[yidx2][xidx2] == self.grid[yidx][xidx] and [xidx2, yidx2] not in self.lockedCells:
                self.incorrectCells.append([xidx2, yidx2])

def checkRows(self):
    for yidx, row in enumerate(self.grid):
        possibles = [1,2,3,4,5,6,7,8,9]
        for xidx in range(9):
            if self.grid[yidx][xidx] in possibles:
                possibles.remove(self.grid[yidx][xidx])
            else:
                if [xidx, yidx] not in self.lockedCells and [xidx, yidx] not in self.incorrectCells:
                    self.incorrectCells.append([xidx, yidx])
                if [xidx, yidx] in self.lockedCells:
                    for k in range(9):
                        if self.grid[yidx][k] == self.grid[yidx][xidx] and [k, yidx] not in self.lockedCells:
                            self.incorrectCells.append([k, yidx])

def checkCols(self):
    for xidx in range(9):
        possibles = [1,2,3,4,5,6,7,8,9]
        for yidx, row in enumerate(self.grid):
            if self.grid[yidx][xidx] in possibles:
                possibles.remove(self.grid[yidx][xidx])
            else:
                if [xidx, yidx] not in self.lockedCells:

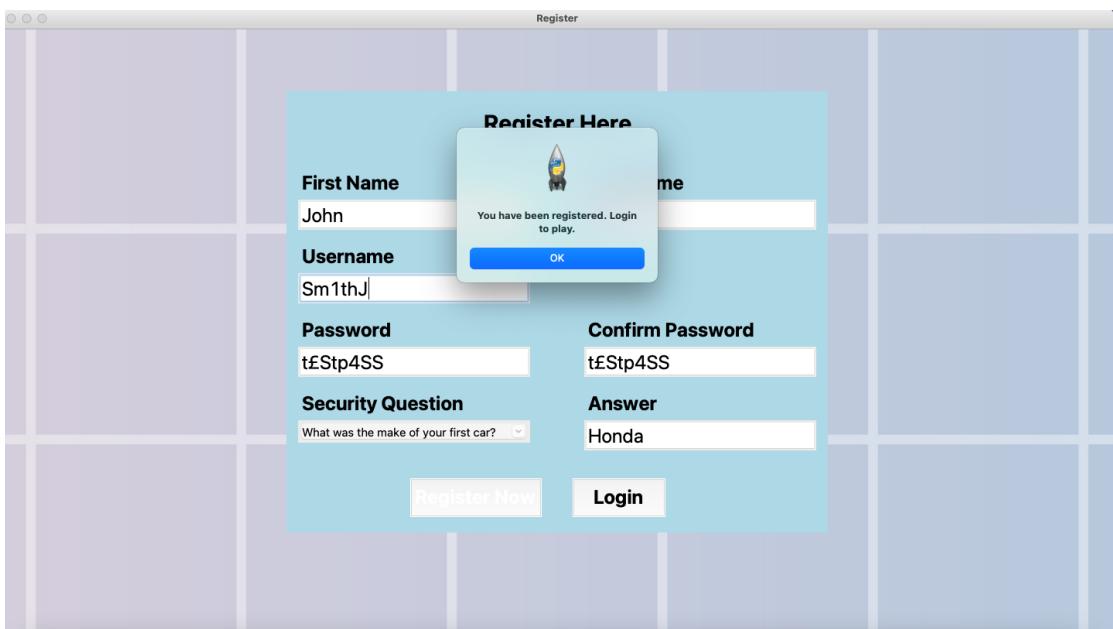
```

		<pre> in self.lockedCells and [xidx, yidx] not in self.incorrectCells: self.incorrectCells.append([xidx, yidx]) if [xidx, yidx] in self.lockedCells: for k, row in enumerate(self.grid): if self.grid[k][xidx] == self.grid[yidx][xidx] and [xidx, k] not in self.lockedCells: self.incorrectCells.append([xidx, k]) </pre>
--	--	--

Test No.	Description	Test Data (T/E/B)	Expected Outcome	Actual Outcome	Comments and Corrective Actions
4. (b)	User is unable to register if input data is longer than character limit	(E) First Name: John Last Name: Smith Username: Sm1thJ Password: t£Stp4SS Confirm Password: p4SS Security Question: What was the make of your first car? Security Answer: I'm pretty sure it was a Honda although I could be mistaken Click 'Register Now'	1. Error message appears telling the user that the data entered exceed the character limit 2. No user details are added to database	As expected	None required
16.	New Sudoku puzzles are able to be scraped and displayed	(T) Click on each of the buttons 'Easy', 'Medium' and 'Hard'	A new puzzle of the appropriate difficulty should be generated each time a button is clicked	As expected	None required

20. (a)	Check that checkAllCells() function works	(T) Click the 'Check' button	Non-locked cells with the incorrect input should be shaded red and correct cells should remain white	As expected	None required
------------	---	---------------------------------	--	-------------	---------------

Testing Outcome Screenshots

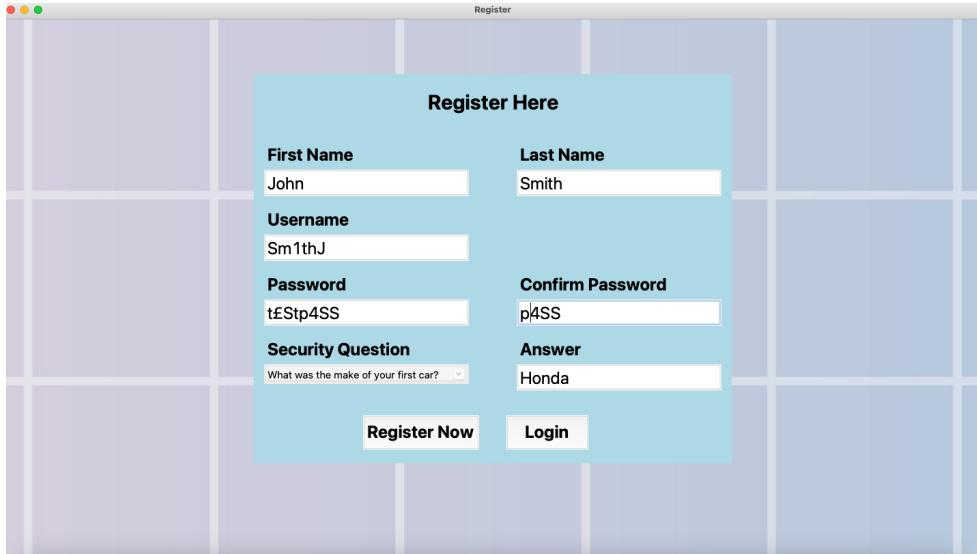
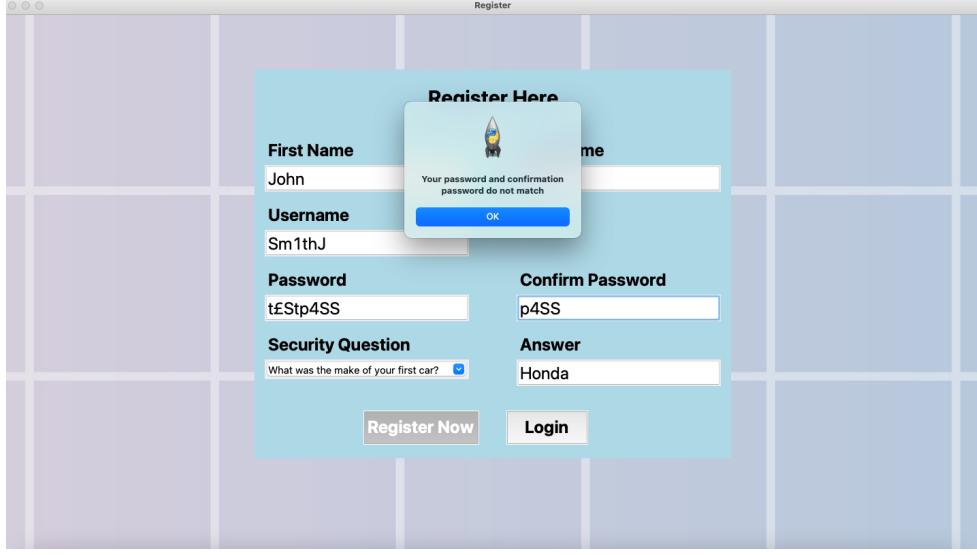
Test No.	Test Outcome Screenshot(s)
1.	 

	Result Grid					
	Fname	Lname	Username	Password	SecurityQ	SecurityA
▶	cute	skni	cuteskni	anushari...	What high school did you...	HIS
Ekta	Nair	ekta2	7Feb		In what city were you born?	KL
Ekta	Nair	ektaaanair	Test@123		In what city were you born?	KL
Hello	World	HelloWorld	tX14g#		What was your favourite f...	pizza
John	Smith	Sm1thJ	t£Stp4SS		What was the make of yo...	Honda
Ekta	Nair	user1	pass1		In what city were you born?	KL
	NULL	NULL	NULL	NULL	NULL	NULL

2.

The screenshot shows a 'Register' window titled 'Register Here'. The form contains fields for First Name (John), Last Name (empty), Username (Sm1thJ), Password (t£Stp4SS), Confirm Password (t£Stp4SS), Security Question (What was the make of your first car?), and Answer (Honda). Below the form are 'Register Now' and 'Login' buttons.

The screenshot shows a 'Register' window titled 'Reaister Here' (likely a typo for 'Register'). A modal dialog box is displayed with a warning icon and the text 'All fields are required'. The 'OK' button is visible at the bottom of the dialog. The underlying form fields are identical to the previous screenshot.

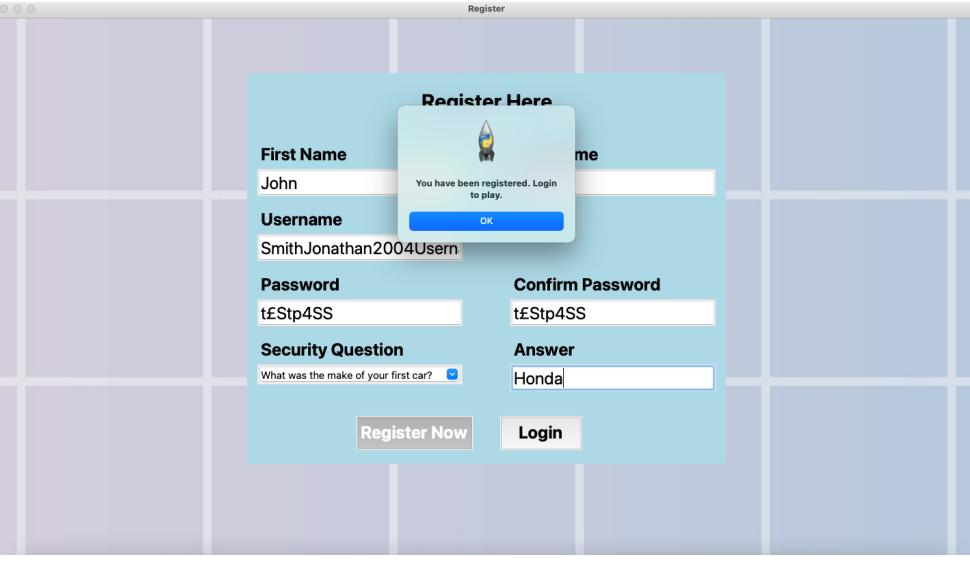
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="7" style="text-align: left; padding-bottom: 5px;"> Result Grid Filter Rows: <input style="width: 150px;" type="text"/> Search Edit: </th></tr> <tr> <th style="width: 40px;"></th><th>Fname</th><th>Lname</th><th>Username</th><th>Password</th><th>SecurityQ</th><th>SecurityA</th></tr> </thead> <tbody> <tr><td>▶</td><td>cute</td><td>skni</td><td>cuteskni</td><td>anushari...</td><td>What high school did you...</td><td>HIS</td></tr> <tr><td>◀</td><td>Ekta</td><td>Nair</td><td>ekta2</td><td>7Feb</td><td>In what city were you born?</td><td>KL</td></tr> <tr><td>◀</td><td>Ekta</td><td>Nair</td><td>ektaaanair</td><td>Test@123</td><td>In what city were you born?</td><td>KL</td></tr> <tr><td>◀</td><td>Hello</td><td>World</td><td>HelloWorld</td><td>tX14g#</td><td>What was your favourite f...</td><td>pizza</td></tr> <tr><td>◀</td><td>John</td><td>Smith</td><td>Sm1thJ</td><td>t£Stp4SS</td><td>What was the make of yo...</td><td>Honda</td></tr> <tr><td>◀</td><td>Ekta</td><td>Nair</td><td>user1</td><td>pass1</td><td>In what city were you born?</td><td>KL</td></tr> <tr><td></td><td>HULL</td><td>HULL</td><td>HULL</td><td>HULL</td><td>HULL</td><td>HULL</td></tr> </tbody> </table>	Result Grid Filter Rows: <input style="width: 150px;" type="text"/> Search Edit:								Fname	Lname	Username	Password	SecurityQ	SecurityA	▶	cute	skni	cuteskni	anushari...	What high school did you...	HIS	◀	Ekta	Nair	ekta2	7Feb	In what city were you born?	KL	◀	Ekta	Nair	ektaaanair	Test@123	In what city were you born?	KL	◀	Hello	World	HelloWorld	tX14g#	What was your favourite f...	pizza	◀	John	Smith	Sm1thJ	t£Stp4SS	What was the make of yo...	Honda	◀	Ekta	Nair	user1	pass1	In what city were you born?	KL		HULL	HULL	HULL	HULL	HULL	HULL
Result Grid Filter Rows: <input style="width: 150px;" type="text"/> Search Edit:																																																																
	Fname	Lname	Username	Password	SecurityQ	SecurityA																																																										
▶	cute	skni	cuteskni	anushari...	What high school did you...	HIS																																																										
◀	Ekta	Nair	ekta2	7Feb	In what city were you born?	KL																																																										
◀	Ekta	Nair	ektaaanair	Test@123	In what city were you born?	KL																																																										
◀	Hello	World	HelloWorld	tX14g#	What was your favourite f...	pizza																																																										
◀	John	Smith	Sm1thJ	t£Stp4SS	What was the make of yo...	Honda																																																										
◀	Ekta	Nair	user1	pass1	In what city were you born?	KL																																																										
	HULL	HULL	HULL	HULL	HULL	HULL																																																										
3.	 <p>The screenshot shows a registration form titled "Register Here". The fields filled are: First Name (John), Last Name (Smith), Username (Sm1thJ), Password (t£Stp4SS), Confirm Password (p4SS), Security Question (What was the make of your first car?), and Answer (Honda). Below the form are two buttons: "Register Now" and "Login".</p>  <p>The screenshot shows the same registration form. This time, the "Confirm Password" field contains "p4SS" instead of "t£Stp4SS". A modal dialog box appears with the message "Your password and confirmation password do not match" and an "OK" button. The other fields remain the same as in the previous screenshot.</p>																																																															
4. (a)																																																																

Register

Register Here

First Name	Last Name
John	Smith
Username	
SmithJonathan2004Usern	
Password	Confirm Password
tEStp4SS	tEStp4SS
Security Question	Answer
What was the make of your first car? <input type="radio"/>	
Honda	

Register Now **Login**



The screenshot shows a registration form titled "Register Here". The fields filled are First Name (John), Last Name (Smith), Username (SmithJonathan2004Usern), Password (tEStp4SS), Confirm Password (tEStp4SS), Security Question (What was the make of your first car? - radio button selected), and Answer (Honda). Below the form, there are two buttons: "Register Now" and "Login". A modal dialog box is displayed in the center, showing a small rocket icon and the text "You have been registered. Login to play." with an "OK" button.

Register

Register Here

First Name	Last Name
John	Smith
Username	
SmithJonathan2004Usern	
Password	Confirm Password
tEStp4SS	tEStp4SS
Security Question	Answer
What was the make of your first car? <input type="radio"/>	
Honda	

Register Now **Login**

This screenshot is identical to the one above, showing the same registration form and the "You have been registered. Login to play." modal dialog.

4. (b)

Register

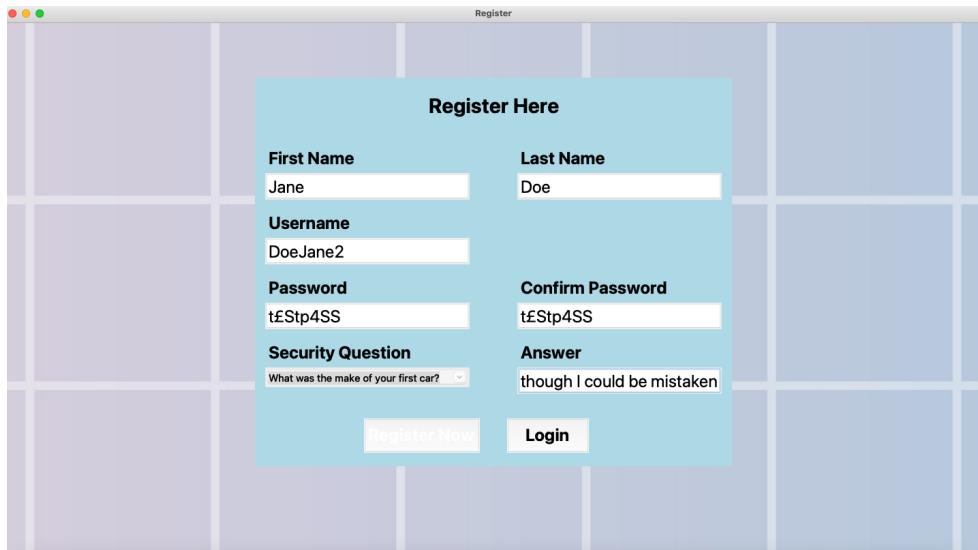
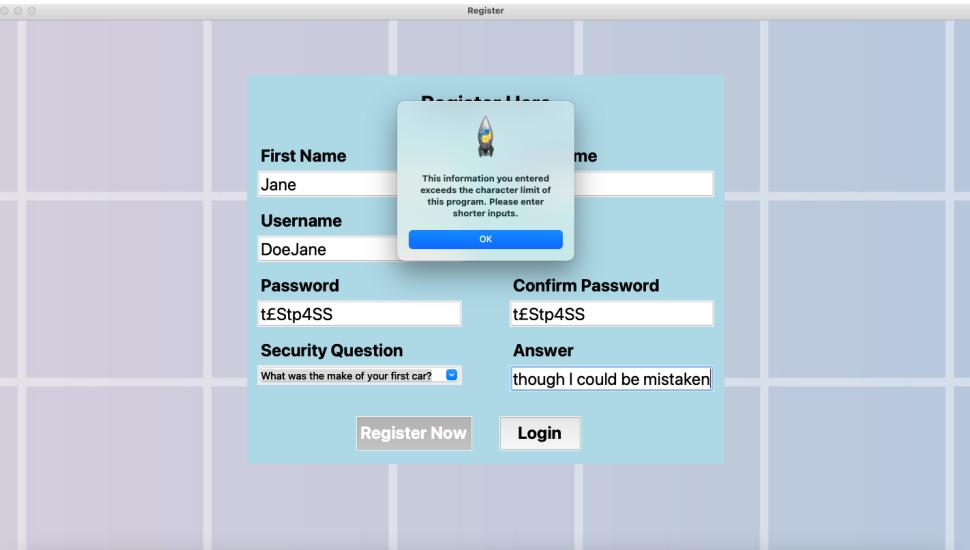
Register Here

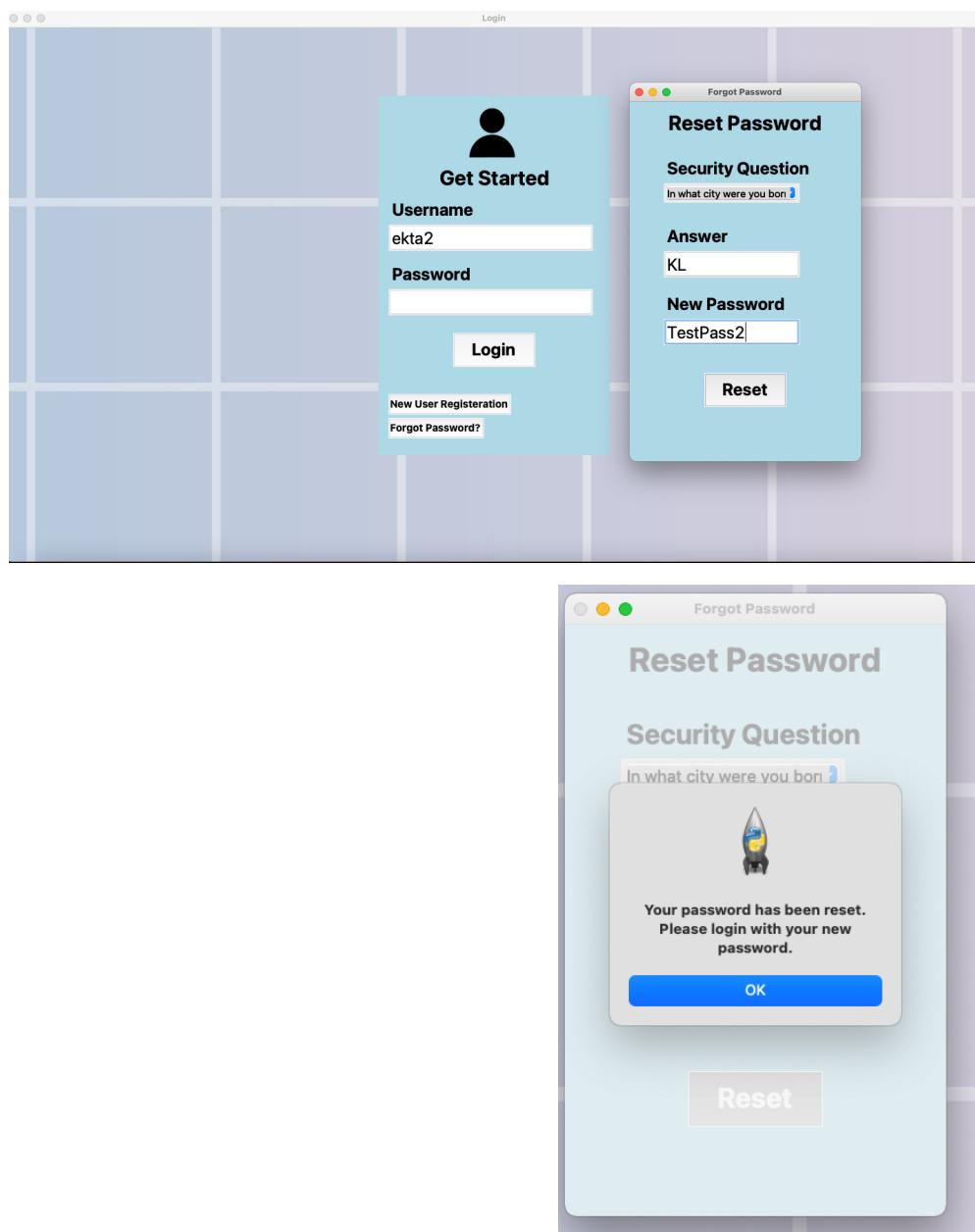
First Name	Last Name
Jane	Doe
Username	
DoeJane	
Password	Confirm Password
tEStp4SS	tEStp4SS
Security Question	Answer
What was the make of your first car? <input type="radio"/>	
though I could be mistaken	

Register Now **Login**



The screenshot shows a registration form titled "Register Here". The fields filled are First Name (Jane), Last Name (Doe), Username (DoeJane), Password (tEStp4SS), Confirm Password (tEStp4SS), Security Question (What was the make of your first car? - radio button selected), and Answer (though I could be mistaken). Below the form, there are two buttons: "Register Now" and "Login".

	 <table border="1" data-bbox="484 752 1391 1134"> <thead> <tr> <th>Fname</th><th>Lname</th><th>Username</th><th>Password</th><th>SecurityQ</th><th>SecurityA</th></tr> </thead> <tbody> <tr><td>cute</td><td>skni</td><td>cuteskni</td><td>anushari...</td><td>What high school did you...</td><td>HIS</td></tr> <tr><td>Ekta</td><td>Nair</td><td>ekta2</td><td>7Feb</td><td>In what city were you born?</td><td>KL</td></tr> <tr><td>Ekta</td><td>Nair</td><td>ektaaanair</td><td>Test@123</td><td>In what city were you born?</td><td>KL</td></tr> <tr><td>Hello</td><td>World</td><td>HelloWorld</td><td>tX14g#</td><td>What was your favourite f...</td><td>pizza</td></tr> <tr><td>John</td><td>Smith</td><td>Sm1thJ</td><td>tEStp4SS</td><td>What was the make of yo...</td><td>Honda</td></tr> <tr><td>John</td><td>Smith</td><td>SmithJo...</td><td>tEStp4SS</td><td>What was the make of yo...</td><td>Honda</td></tr> <tr><td>Ekta</td><td>Nair</td><td>user1</td><td>pass1</td><td>In what city were you born?</td><td>KL</td></tr> <tr><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td><td>NULL</td></tr> </tbody> </table>	Fname	Lname	Username	Password	SecurityQ	SecurityA	cute	skni	cuteskni	anushari...	What high school did you...	HIS	Ekta	Nair	ekta2	7Feb	In what city were you born?	KL	Ekta	Nair	ektaaanair	Test@123	In what city were you born?	KL	Hello	World	HelloWorld	tX14g#	What was your favourite f...	pizza	John	Smith	Sm1thJ	tEStp4SS	What was the make of yo...	Honda	John	Smith	SmithJo...	tEStp4SS	What was the make of yo...	Honda	Ekta	Nair	user1	pass1	In what city were you born?	KL	NULL	NULL	NULL	NULL	NULL	NULL
Fname	Lname	Username	Password	SecurityQ	SecurityA																																																		
cute	skni	cuteskni	anushari...	What high school did you...	HIS																																																		
Ekta	Nair	ekta2	7Feb	In what city were you born?	KL																																																		
Ekta	Nair	ektaaanair	Test@123	In what city were you born?	KL																																																		
Hello	World	HelloWorld	tX14g#	What was your favourite f...	pizza																																																		
John	Smith	Sm1thJ	tEStp4SS	What was the make of yo...	Honda																																																		
John	Smith	SmithJo...	tEStp4SS	What was the make of yo...	Honda																																																		
Ekta	Nair	user1	pass1	In what city were you born?	KL																																																		
NULL	NULL	NULL	NULL	NULL	NULL																																																		
4. (b) RETEST																																																							
5.																																																							



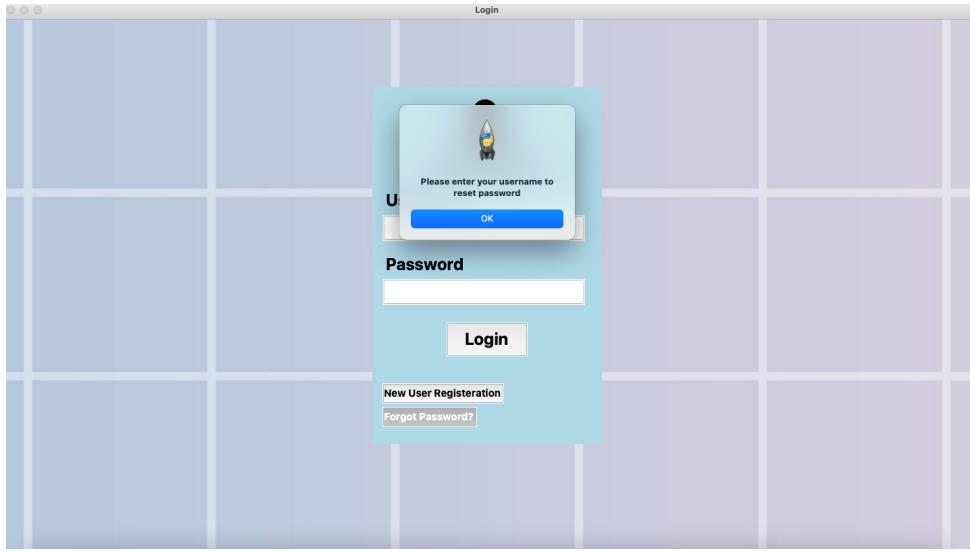
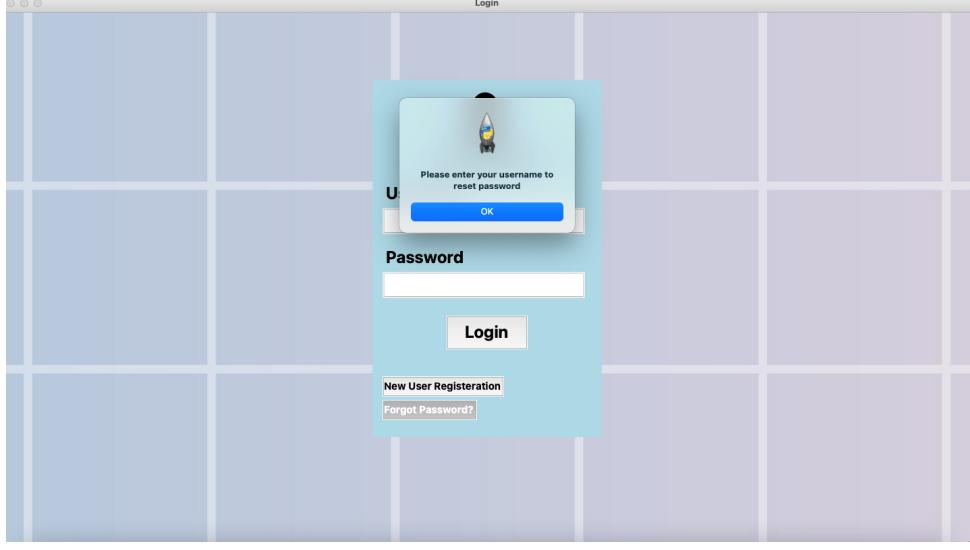
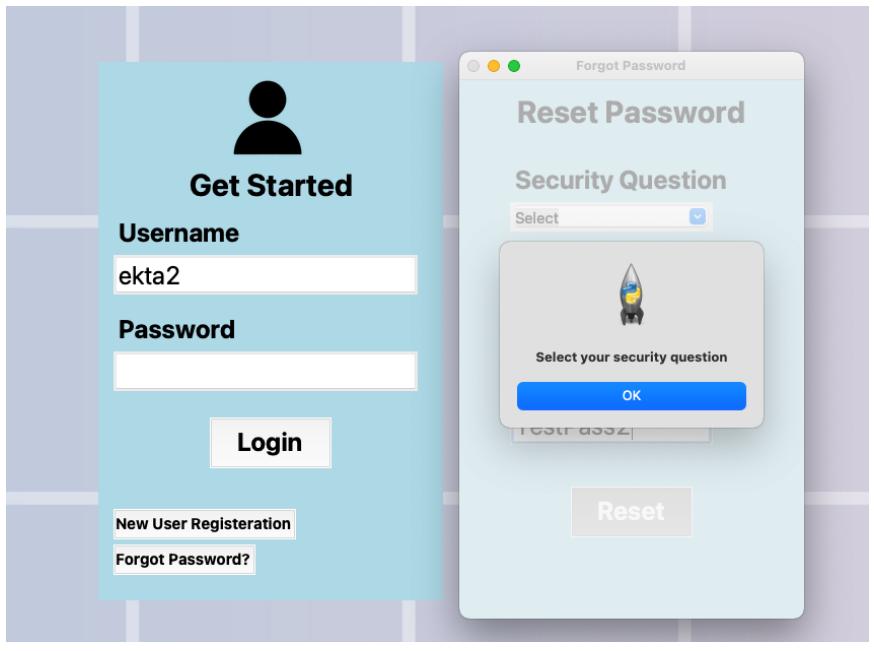
Database before password reset:

Ekta	Nair	ekta2	7Feb	In what city were you born?
Ekta	Nair	ektaanair	Test@123	In what city were you born?
Hello	World	HelloWorld	tX14g#	What was your favourite colour?

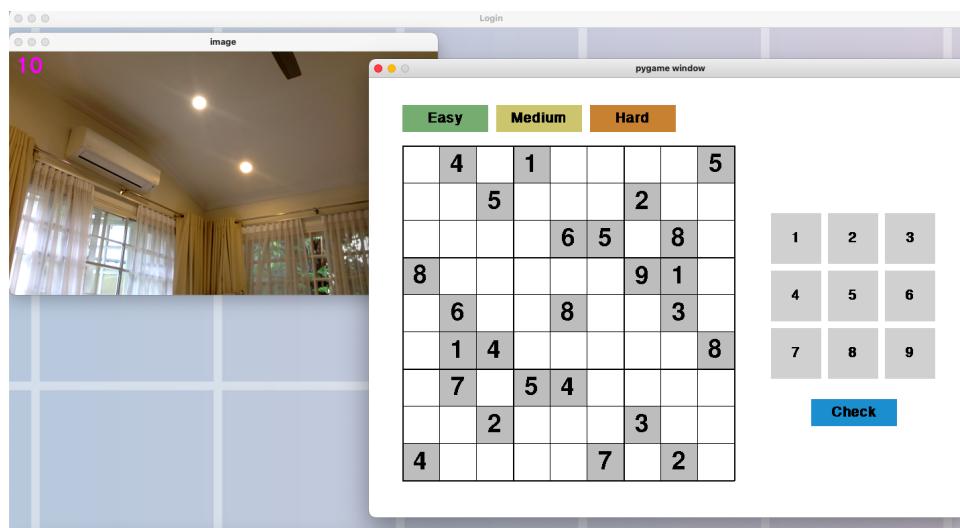
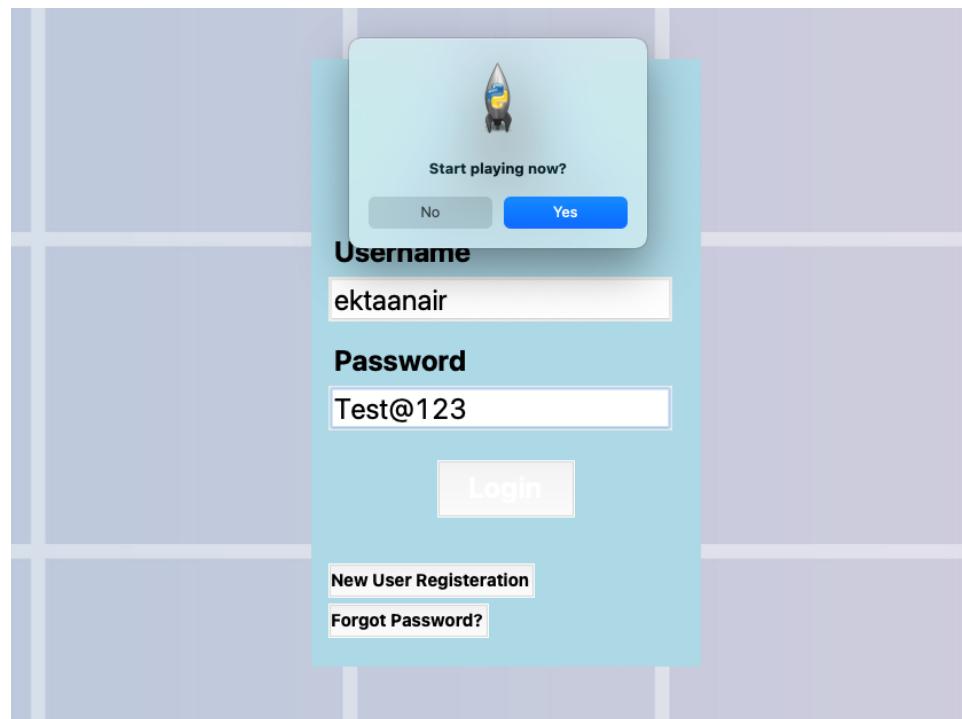
Database after password reset:

Ekta	Nair	ekta2	TestPass2	In what city were you born?
Ekta	Nair	ektaanair	Test@123	In what city were you born?
Hello	World	HelloWorld	tX14g#	What was your favourite colour?
John	Smith	Sm1th.l	tFSn4SS	What was the name of your first school?

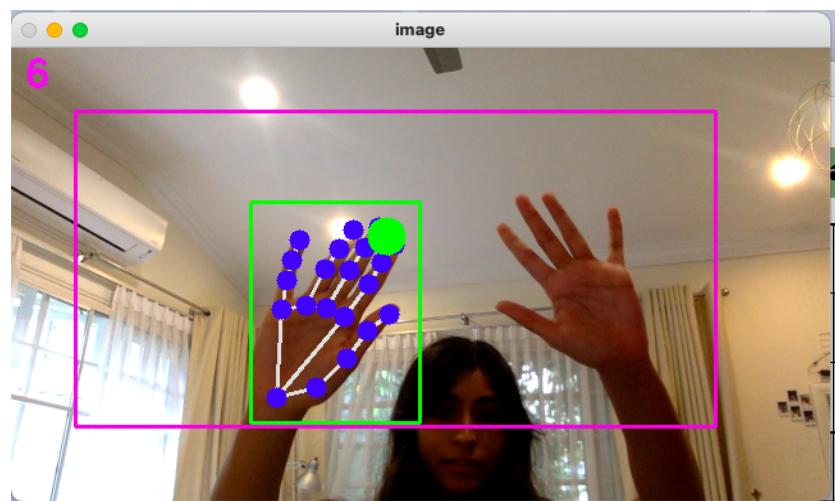
6.

	
7.	
8.	

9.

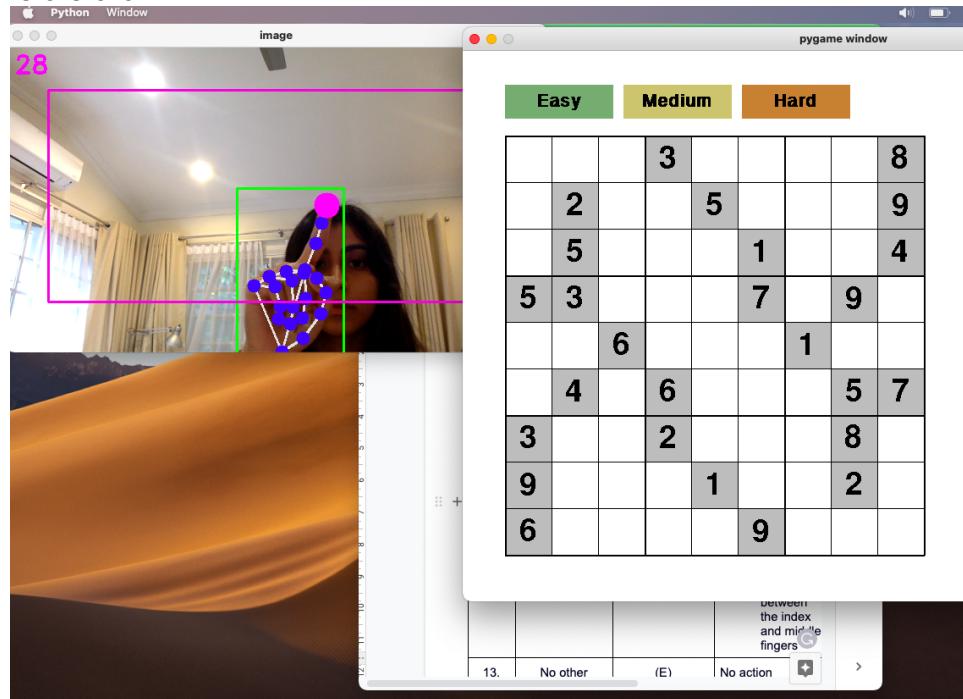


10.

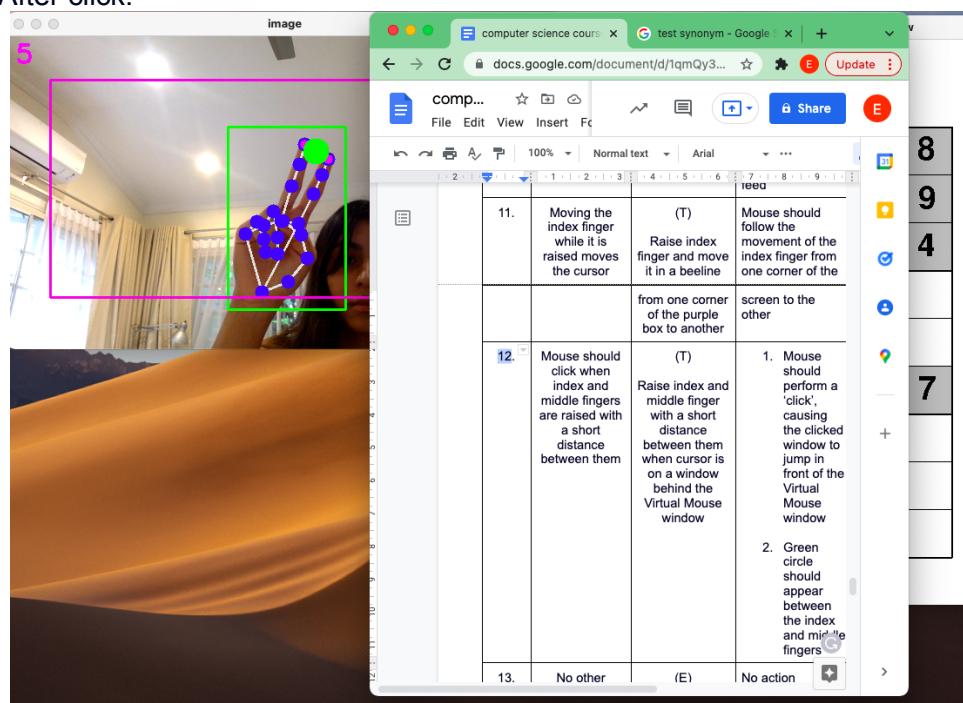


12.

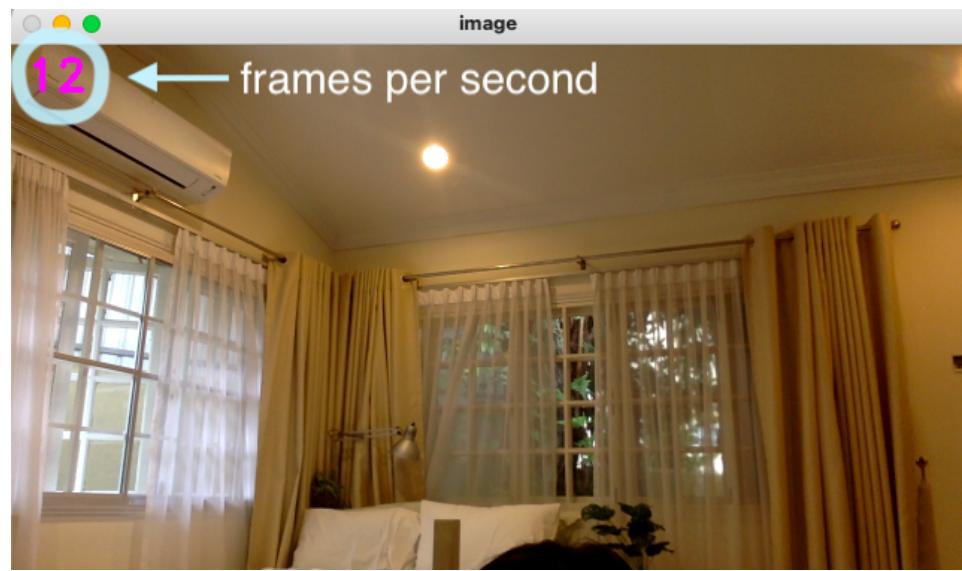
Before click:



After click:



14.



15.

image

pygame window

Easy Medium Hard

2	6			9	4	3
			8			
7			5	9		
		3			4	1
	2				7	
1	8				6	
		7	1			5
				8		
5		3	7		4	8

Check

16.

pygame window

Easy Medium Hard

6	9			4		7		2
		7						
2	1			5	9			
1			9		5		3	
7	2		3	6		8	4	
4		5	8				7	
		2	5			3	6	
					8			
9		4	3			1	5	

Check

1 2 3
4 5 6
7 8 9

```
Hello from the pygame community. https://www.pygame.org/contribute.html
/Users/rakesharavind/NEA/app_class.py:183: GuessedAtParserWarning: No parser was expli
The code that caused this warning is on line 183 of the file /Users/rakesharavind/NEA/
[some code]
soup = BeautifulSoup(html_doc)
[[0, 3, 4, 6, 7, 8, 9, 1, 2], [6, 7, 2, 1, 9, 5, 3, 4, 8], [1, 9, 8, 3, 4, 2, 5, 6, 7]
[[0, 3, 4, 6, 7, 8, 9, 1, 2], [6, 7, 2, 1, 9, 5, 3, 4, 8], [1, 9, 8, 3, 4, 2, 5, 6, 7]
[[0, 3, 4, 6, 7, 8, 9, 1, 2], [6, 7, 2, 1, 9, 5, 3, 4, 8], [1, 9, 8, 3, 4, 2, 5, 6, 7]
[[0, 3, 4, 6, 7, 8, 9, 1, 2], [6, 7, 2, 1, 9, 5, 3, 4, 8], [1, 9, 8, 3, 4, 2, 5, 6, 7]
[[0, 3, 4, 6, 7, 8, 9, 1, 2], [6, 7, 2, 1, 9, 5, 3, 4, 8], [1, 9, 8, 3, 4, 2, 5, 6, 7]

Run TODO Problems Terminal Python Packages Python Console Event Log
```

16. RETEST

pygame window

	Easy	Medium	Hard
	6	2	1 7
		6	4
7	8 2		
8			7
3	5 7		8 2 6
	4		1
			6 3 9
	5	6	
	6 7	9	1

1	2	3
4	5	6
7	8	9

Check

pygame window

	Easy	Medium	Hard
	9	7	1 4
	6		1
	3 4		
3			4 9 5
	8 9		7 2
	6 7 5		
6			7 4
	1		5
	5	6	2 1

1	2	3
4	5	6
7	8	9

Check

```

/c/Users/rakesharavind/NEA/venv/bin/python /Users/rakesharavind/NEA/app_class.py
pygame 2.1.0 (SDL 2.0.16, Python 3.8.0)
Hello from the pygame community. https://www.pygame.org/contribute.html
/Users/rakesharavind/NEA/app_class.py:183: GuessedAtParserWarning: No parser was expl
The code that caused this warning is on line 183 of the file /Users/rakesharavind/NEA

soup = BeautifulSoup(html_doc)

```

18.

pygame window

Easy **Medium** **Hard**

	6		2		1	7		
			6			4		
7	8	2						
8				1			7	
3	5	7			8	2	6	
	4							1
					6	3	9	
		5			6			
		6	7		9		1	

1	2	3
4	5	6
7	8	9

Check

19. (a)

pygame window

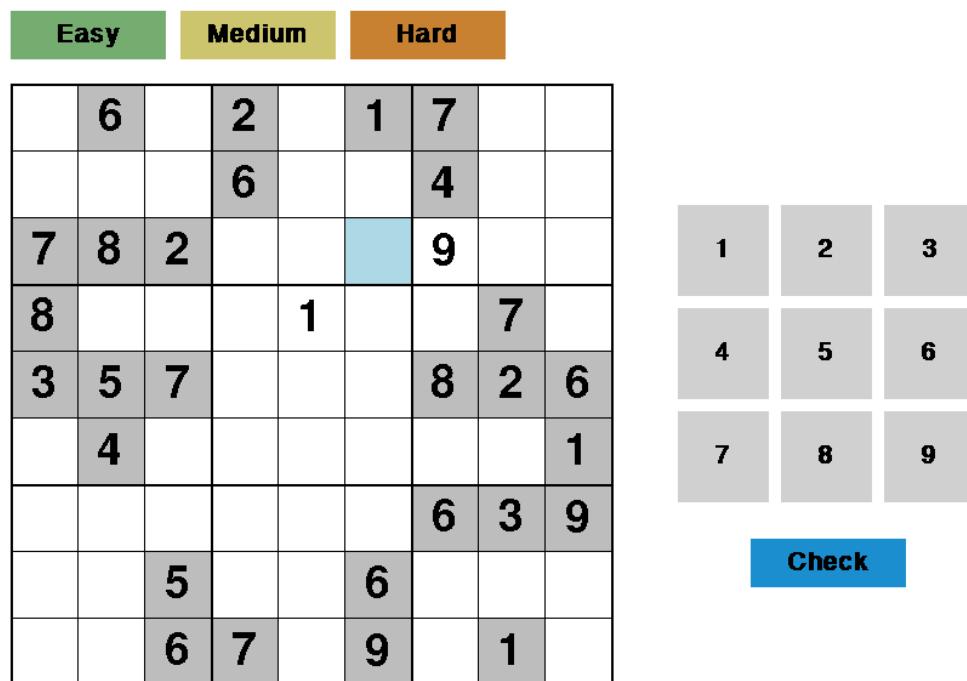
Easy **Medium** **Hard**

	6		2		1	7		
			6			4		
7	8	2				9		
8				1			7	
3	5	7			8	2	6	
	4							1
					6	3	9	
		5			6			
		6	7		9		1	

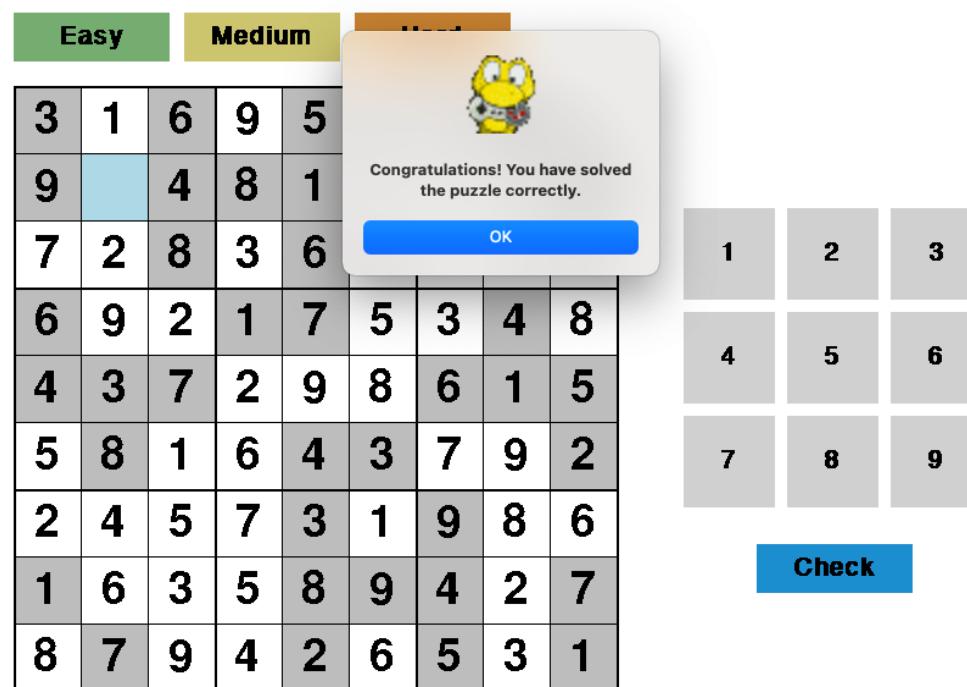
1	2	3
4	5	6
7	8	9

Check

19. (b)



20. (b)

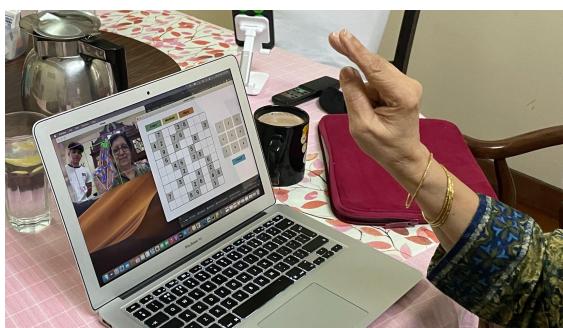


Beta Test

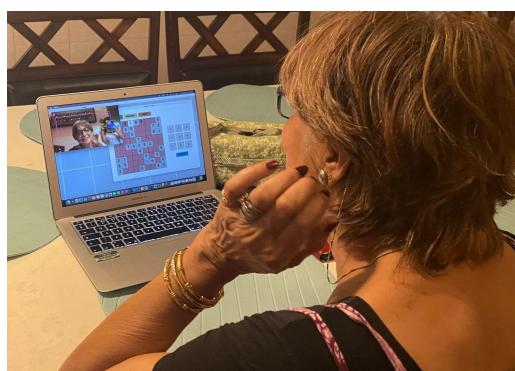
During beta testing the system was trialled by a small group of prospective users to test its robustness and completeness. I chose the 3 individuals I interviewed at the beginning of my project to trial the system as they are familiar with the purpose and objectives of the system.



Aravind registering an account



Rekha and Ambika using the Virtual Mouse



Ambika checking her puzzle

User Feedback

In order to obtain objective feedback of how users rated the system after the initial use, I created a google form for testers to fill out after trialling the system. They were asked to rate the success of the features of the system on a scale from 1 to 5 (1 being poor and 5 being excellent). I then asked for any additional feedback on any aspect of the system or points of improvement for the system.

The feedback form distributed to Ambika, Aravind, and Rekha after they trialled the system was:

System Evaluation

I really appreciate you taking some time to evaluate your experience using my system.

Please rate the following aspects of the system out of 5 (1- poor & 5 - excellent)

User interface

1
 2
 3
 4
 5

Registering a new account

1
 2
 3

Resetting account passwords

1
 2
 3
 4
 5

Ease of using Virtual Mouse

- 1
- 2
- 3
- 4
- 5

Generating new Sudoku puzzles of varying difficulties

- 1
- 2
- 3
- 4
- 5

Checking your Sudoku puzzle for errors

- 1
- 2
- 3
- 4
- 5

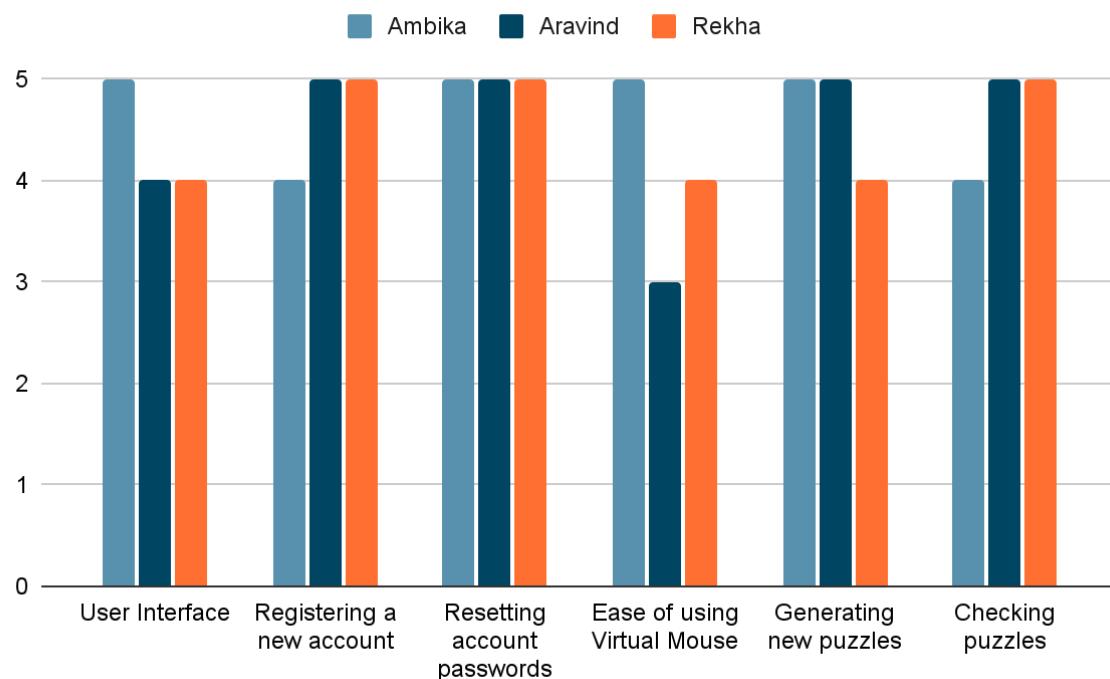
Any additional comments or possible improvements to the system?

Your answer

Submit

Clear form

I have summarised the feedback received below.



Desirable improvements:

Ambika	Aravind	Rekha
<ul style="list-style-type: none"> Rename 'Login' button on Registration window to avoid confusion of which button to press to register account Checking system should also give correct answers to puzzle 	<ul style="list-style-type: none"> Avoid having too many windows open at the same time Too many overlays on the hand make it difficult to know which part of the hand is being tracked Needed more of an explanation that finger controlling Virtual Mouse should stay within the purple box on the screen 	<ul style="list-style-type: none"> To be able to revisit completed puzzles

Evaluation

Evaluation of Objectives

Key:

- Objective not met
- Objective partially met
- Objective met

Objective	Evaluation
<p>Username and password must be validated before allowing the user to access the gesture-controlled game</p> <ul style="list-style-type: none"> • Compare user input to data stored in database • If invalid, show error message • If valid, open gesture-controlled game window on top of login window 	<p>The username and password entered by the user are compared to those already stored in the database, using select queries, when the 'Login' button is clicked. Error messages are shown in the invalid usernames or passwords are entered into the fields, or if they are left blank.</p> <p>If a valid username and password are entered a prompt appears asking the user if they are ready to start playing. If the user clicks the affirmative, the Sudoku Puzzle and Virtual Mouse windows appear above the Login window.</p> <p>Users found the login process straightforward and self-explanatory and were able to create their accounts and log in with ease.</p>
<p>When new users are registering their accounts, user data must be validated before it is stored in the database</p> <ul style="list-style-type: none"> • If invalid, show error message • If valid, store data in the correct columns in the database and show success message 	<p>User inputs are validated for presence and length before they can be successfully stored in the database. If they are unable to be stored in the database, an error message will notify the user. Similarly, if a username entered by the user registering is identical to the username of an existing user, an error message will inform the user that they must change their username to be registered.</p> <p>Each field in the registration form corresponds to a column in the database. Hence, when a user registers successfully, the user inputs in each field are copied into the corresponding column of the table in the database.</p>
Feature to allow users to reset passwords	Users are able to reset their passwords by

<p>to their account</p> <ul style="list-style-type: none"> ● Validate username, security question, and answer by comparing it to data stored in the database ● If valid, delete the original password from the database and replace it with the user's input for a new password ● If invalid, show an error message 	<p>selecting their security question and answering it correctly (as they did when they registered). Although, they must first enter their username. This is something many may forget to do as it must occur in the Login window instead of the Forgot Password window. However, an error message will remind them to enter their username if they try to access the Forgot Password window without having entered it.</p> <p>If any fields are left blank or filled incorrectly, an error message notifies the user of this. Otherwise, the password in the user's record of the database is replaced with the new password set by the user.</p>
<p>Virtual Mouse must be able to detect hand from camera feed</p> <ul style="list-style-type: none"> ● Camera feed must be shown to user ● Drawings are overlaid on the hand in the feed to convey to the user that the hand has been identified 	<p>The Virtual Mouse window shows a live feed of what the camera 'sees' and is able to detect the user's hand using this camera feed.</p> <p>One minor issue is that the camera feed is not flipped although it is seemingly facing the user. This means that the image of the user's finger in the camera feed travels in the opposite direction to the user's actual finger and the mouse pointer. However, this did not appear to phase users as they were more focused on the position of the mouse rather than the image of themselves.</p> <p>A boundary box surrounds the hand and circles are overlaid over hand landmarks to convey to the user that their hand is being tracked.</p>
<p>Hand gestures must be identified and appropriate action must be taken</p> <ul style="list-style-type: none"> ● Move index finger to move mouse ● Raise middle and index fingers to click 	<p>Virtual Mouse identifies when the index and middle fingers are raised and either moves the mouse pointer or clicks, depending on the gesture made. Drawings are overlaid on the fingers to convey to the user which gestures the system is recognising.</p>
<p>Virtual Mouse must be able to interact with the Sudoku puzzle (ie. be able to select buttons and cells on the Sudoku grid)</p>	<p>The Virtual Mouse is able to fully interact with the Sudoku Puzzle window and select all buttons and cells on the window.</p> <p>The Virtual Mouse is not specifically part of the Sudoku Puzzle window, hence it is able to interact with all windows and applications on the computer; not just the Sudoku Puzzle window. This can cause some disruptions as the user may accidentally</p>

	open another window or application over top of my system when using the Virtual Mouse.
The onscreen keypad must be able to be used to enter values into the Sudoku grid to solve puzzles	The onscreen keypad consists of 9 buttons with the numbers 1 to 9 on them. If a button on the onscreen keypad is clicked when a non-locked cell on the Sudoku grid is selected, it will enter the corresponding number into that cell.
The gesture-controlled game must be able to generate new Sudoku puzzles of varying difficulties	There are 3 buttons (Easy, Medium, and Hard) that when pressed will scrape a Sudoku puzzle of the corresponding difficulty from a website and display it on the grid of the Sudoku Puzzle window.
If the user does not wish to use the onscreen keypad, valid values can be entered into the Sudoku grid by pressing the appropriate keys on the keyboard	If a non-locked cell is selected, the user will be able to type in a digit between 1 and 9. This digit will be entered into the selected cell and displayed on the screen. If a value other than the digits 1 to 9 is input using the keyboard, nothing will be entered into the selected cell.
A feature that will check if the Sudoku puzzle completed correctly by the user is correct	<p>The checkAllCells() function is triggered when the user clicks the 'Check' button or when they complete the puzzle. This function is able to check if there are any repeated numbers in each row, column, or 3 by 3 grid and shade the cells of the repeated values red to notify the user of this.</p> <p>However, the function is unable to recognise if values in the same row, column, or 3 by 3 grid are swapped (eg. in the correct row but incorrect location) as it does not actually find the correct solution for the puzzle.</p>
<p>The different windows of the system must be interconnected and open and close smoothly when needed</p> <ul style="list-style-type: none"> • Users should be able to navigate between windows easily and independently 	<p>Windows are able to open and close when the appropriate buttons are clicked. However, the Login window remains open constantly as it is the root window, and closing it would terminate the program.</p> <p>Users managed to open all necessary windows to manoeuvre around the system, but some users were confused at the presence of the Login window when it was not necessary. They found that having it open cluttered the screen and made for an unpleasant user interface.</p>

General Evaluation

Overall, the system achieved its objective of creating a simple game that could be played using gesture controls. It was not intended to suit intense full-body rehabilitation, the simple gestures successfully detected by the Virtual Mouse window using AI image processing algorithms provide an accessible simple game to pass the time while simultaneously executing strengthening gestures and also challenging the mind. My program, however, focuses on hand gestures and so would not be suitable for rehabilitating other parts of the body. Even if users are not going through rehabilitation, they will be able to use the system as a form of entertainment; to avoid feeling lonely or depressed while not tiring themselves out with constant strenuous activities.

The system does not require any external inputs or companion devices other than a webcam if the device the client is using does not have a camera. This significantly drives down the costs of the system, setting it apart from other similar systems in its field (as researched on page 8).

Having met the majority of its objectives, the system seems generally successful. Nevertheless, a number of improvements could still be made in future iterations.

Potential for Future Developments

Making the Virtual Mouse a part of the Sudoku Puzzle window as per my original plan would clean up the look of the GUI of the system. There would be fewer windows open, making for a less overwhelming screen and an overall more pleasant user experience. Closing the Login window if another window opens on top of it would add to the impact. This probably could have been executed, had I known about threading earlier on in my project and still had enough time to execute this idea. However, the only solution I could think of at the time was to create two separate windows.

While the login system is not the main focus, it can still be improved upon by adding more validation checks (rather than just length and presence) specifically on the password field. This is to ensure users select a secure password to improve the security of the system. These validations could include, making it compulsory for there to be at least one capital letter, number, and symbol in the password. The password amongst other fields that contain personal user data could also be encrypted before it is stored to improve the overall security of data in the system.

To make the system more versatile and accessible, a calibration feature could be added so that users can personalise their gesture controls. While the gestures, I chose in my system a simple enough for most, personalised gesture controls would make the system more accessible as not all people have the same level of dexterity; therefore they cannot all make the same gestures. Furthermore, individuals in rehabilitation would be more likely to use the system if they could incorporate it into their rehabilitation plans by assigning compulsory hand exercises as gestures to control the system. Similar computer vision and image processing techniques that were used in this system would be used to identify the positions of the hand landmarks of the user's hands in order to detect the gesture that the user is making. This gesture would then be assigned an action (eg. move the mouse pointer).

Analysis of User Feedback

Taking into account the feedback received from test users, changes like renaming buttons to provide more clarity of their purposes and removing overlays that indicate hand landmarks that are not being tracked, can be implemented in future iterations of the system.

A user guide could be provided alongside the system so that users have a better understanding of how the system works. Some users found it challenging to learn to use the Virtual Mouse as it is a feature that does not have mainstream attention yet and so many users had never used something similar to this. A user guide may provide some clarity as to what the purpose of the different coloured rectangles and circles overlaid onto the hand is, as well as a general explanation of what the Virtual Mouse is. This might help users be more comfortable using the system, increasing their enjoyment. An instruction menu added to the system could have the same effect. However, the ability to play the game using peripheral input provides a safety blanket for the user as they can always revert back to more familiar methods of handling a computer should they want to.

Rekha suggested the possibility of being able to revisit old puzzles. This could be done by having loaded puzzles be stored in a dictionary or database with unique IDs that could be called if the user wanted to view that puzzle again.

In hindsight, it is evident that the checking algorithm I decided to use was not the most efficient. Having the puzzle stored permanently would also allow for the use of a backtracking algorithm to be used to solve the puzzle. This would give users the option to view the correct answer to the puzzle - as Ambika wanted. This would also optimise the checking algorithm used, as the algorithm could simply compare values against the correctly solved puzzle to identify errors. This would method of checking the puzzle would not disregard swapped values, hence improving the accuracy of the puzzle checking feature.

The improvements listed above could be the objectives for the second pass through the system's development life cycle.

Bibliography

¹(2017, September 18). Boredom could extend your hospital stay | Daily Mail Online.

Retrieved February 26, 2022, from

<https://www.dailymail.co.uk/health/article-4896710/Boredom-EXTEND-hospital-stay.html>

²(2020, December 25). Cooroy hospital robotic rehab program uses virtual-reality ... - ABC.

Retrieved February 26, 2022, from

<https://www.abc.net.au/news/2020-12-26/robotics-rehab-program-at-cooroy-hospital-helps-stroke-victims/13006646>

³(2020, September 30). Playing video games helps stroke recovery | Brunel University London. Retrieved February 26, 2022, from

<https://www.brunel.ac.uk/news-and-events/news/articles/Playing-video-games-helps-stroke-survivors-recover-soonered2>

⁴(2019, December 16). Video Games Rise in Popularity Among Older Adults - AARP.

Retrieved February 26, 2022, from

<https://www.aarp.org/home-family/personal-technology/info-2019/report-video-games.html>

⁵(n.d.). Top 5 Memory Games for Senior Adults. Retrieved February 26, 2022, from

<https://www.caringseniorservice.com/blog/top-5-memory-games-for-seniors>

⁶(n.d.). Analysis of movement and gesture recognition using Leap Motion Retrieved February 28, 2022, from

<https://www.sciencedirect.com/science/article/pii/S1877050918307403>

⁷(n.d.). The Sudoku 4ALL Interface. | Download Scientific Diagram. Retrieved February 28, 2022, from

https://www.researchgate.net/figure/The-Sudoku-4ALL-Interface_fig2_220807028

⁸(n.d.). Hand Exercises - Physiopedia. Retrieved February 28, 2022, from

https://www.physio-pedia.com/Hand_Exercises

⁹(n.d.). Design decoded: The top 12 easy to read fonts - Vistaprint. Retrieved February 28,

2022, from <https://www.vistaprint.com/hub/design-decoded-top-12-easy-read-fonts>

¹⁰(2020, May 26). Understanding color theory: the color wheel and finding - InVision.

Retrieved February 28, 2022, from

<https://www.invisionapp.com/inside-design/understanding-color-theory-the-color-wheel-and-finding-complementary-colors/>

¹¹(n.d.). Mediapipe hand landmarks. | Download Scientific Diagram. Retrieved February 28, 2022, from https://www.researchgate.net/figure/Mediapipe-hand-landmarks_fig1_355402809

¹²(n.d.). Parse tree - Wikipedia. Retrieved February 28, 2022, from

https://en.wikipedia.org/wiki/Parse_tree