**Section 1 : Analysis**

**Problem identification**

People play games very often as a hobby or even a job and really want to improve at the games, a big aspect in the majority of games is the ability to aim, this skill needs quick reactions and fast reflexes. So, the problem I am trying to solve is how to improve at games but specifically at the aim, reactions and reflex aspect. Many people who play games want to improve but struggle due to a lack of specific practice in-game so a third-party program that would help u improve your reflexes and reactions would be widely attractive.

I will make a program that improves your reactions and reflexes by having multiple mini-game modes that target the improvement of reflexes and reactions which would, in turn, improve your aim in games and make u improve at your game and maybe even other aspects in life.

This program is solvable by computational methods as the problem itself is based on games that are only computational and the people that would play this would already be on their computers so it would make it much easier to improve aim (reactions and reflexes). This program will require the players to sign in or register to make an account, this may take up a lot of data which will need a database then a series of the game would appear, and u will then play being able to get a high score and then put the score on the leaderboard to have a competitive aspect to this game. You will also be able to change a certain setting to make the game your own.

The main problem I am trying to solve with this project is the lack of training for controllers. As gaming and esports are becoming an actual job pathway, being able to practice can increase the chances of it being a reality, and due to the gap in the market for the controller, aim trainers see a great opportunity to make a solution that fills the gap.

**Why a computational solution is needed.**

A computational problem is needed due to many reasons, the main one being that video games (the ones that I am targeting) are only computational so a real-life non-computational solution would not work. Also, it is very easy and convenient for gamers who game on their computers anyway to also practice and train on their computer as well. The software I am making needs a controller or a mouse which also needs to be run on a computer as these are peripherals made for different types of computers. A computational solution is needed.

**The problems I might run into**

The problems that I might run into when making this is the need for a database to store things like scores, account usernames and passwords, a working and functional menu that can change in-game settings like timers, colours, crosshairs and much more, support for both controller and mouse The other problem that I will try to solve is how I’m going to make the settings menu and how I will be able to change things like timers and crosshairs which change the way you will play.

**Computational methods**

Problem decomposition:

Problem decomposition is the action of breaking down a complex problem into smaller parts that are more manageable and easier to understand, this is what I’m going to use to try to plan and help me keep track of my project.

1. Login, register and guest menu
2. The Game
3. The Database
4. The Settings

Each of these problems can be divided even more into subproblems to make it easier to solve each one. For example for the game the subproblems would be :

1. The controls/mouse movements
2. The scoring/timing system
3. How different game modes work and function like object movement

This makes the overall problem less intimidating and complex and makes it seem very simple and easy to complete.

Divide and conquer:

Divide-and-conquer is an algorithm design paradigm based on multi-branched recursion. A Divide-and-conquer algorithm works by recursively breaking down a problem into two or more sub-problems of the same or related type until these become simple enough to be solved directly.

Now that my problem is broken down, I will then make each of my subproblems different modules and add them all together in my final main python file.

**Stakeholders**

The main demographic I’m aiming at is gamers looking to improve at games or even strive to be a professional gamer, these people are usually male and aged 14-20

Even though this program is aimed for gamers it can also benefit other people very well, for example, this can help drivers with their reaction time and focus during driving

Gamers can use this program in between queue times during free time or even just for fun and challenge themselves.

I don’t think there’s a need any accessibility requirements as if someone is able to play games then they will be able to play this game may be colour blind for people so I can research a different type of colour blindness and colours that suit their a condition for deaf people this won’t be problem maybe I have a disable sound option (if there are sound effects for shooting) or subtitles but I don’t think I’ll need it.

**Interview**

I will interview 2 people with different opinions and perspectives on the program I will make hoping to understand how people with different views on gaming will see this program.

The first person I will interview is my friend Rueben who plays games pretty often but I want to get a deeper look at the types of games he plays and if an aim trainer will suit him.

These are the questions I will be asking him:

1. Do u play games if so what is it pc or console and what time of games do you play?

Too see if his perspective view and experience in games

1. Do you want to improve at the game you play or do you just play just for fun?

Too see his aspirations and what they want to achieve so if they play just for fun or if they find improving fun

1. Do u think aim/reactions/reflexes are important in games?

Too see if the main purpose of this program lines up with what skills he values in games

1. Do you think you can improve your aim by playing aim trainers?

Too see if this game actually attracts an average gamer

1. Do you know any aim training games and if so can u offer any aspects that u enjoy?

Too see if he can offer any new suggestions that I can implement in my program

1. Would you play this game to practice your aim and get better in games?

Too see if he can be a potential customer

1. Do u think this game can benefit people in any other aspect other than games?

Too see how I can broaden the demographic of this project

**Interview 2**

The second interview is for another gamer named Rayane but one that is a more hardcore and competative and is a proir user of other aim trainers and can give a better and more indepth view .

These are the questions i will ask:

1. What games do you play competitively?

Too see the games that relate with aim trainers

1. What aim trainers do you use?

Too see my main competitors

1. Do you see any potential in this?

Too see if his experienced view can see a future in my solution

1. Do you see playing games as a job pathway?

Too see if the main reason for my solution is valid

1. What are your favourite parts of the aim trainers you have used?

Too see what i can implement from the competitors he has used

1. Would you use this as your future aim trainer?

Too see if rayane is a future customer

(people are busy)

**Current solutions**

There are many aim training games/programs out there, but they are either web-based so needs a stable connection or cost money to play. There is also no aim trainer for the controller as they are all for keyboard and mouse games so I will be trying to target console/controller gamers rather than pc gamers that these programs often target.

**Kovaaks**

Kovaaks is an aim trainer on steam developed by a company called the meta. It is an aim trainer designed for mainly FPS games like CSGO and. It is the most popular aim trainer out right now and averages about 1700 concurrent players all the time. Kovaaks has lots of mini-games and game modes and even lets you make your game mode to satisfy the thing you are trying to improve at.

One of the most popular game modes is called tile frenzy and is the one on the left. This is one of the game modes which consists of 2 boxes being on screen and clicking them as quick as possible, and as soon as you click 1 another pops up so there is always 3 on the screen at once that I will improve and implement into my program.

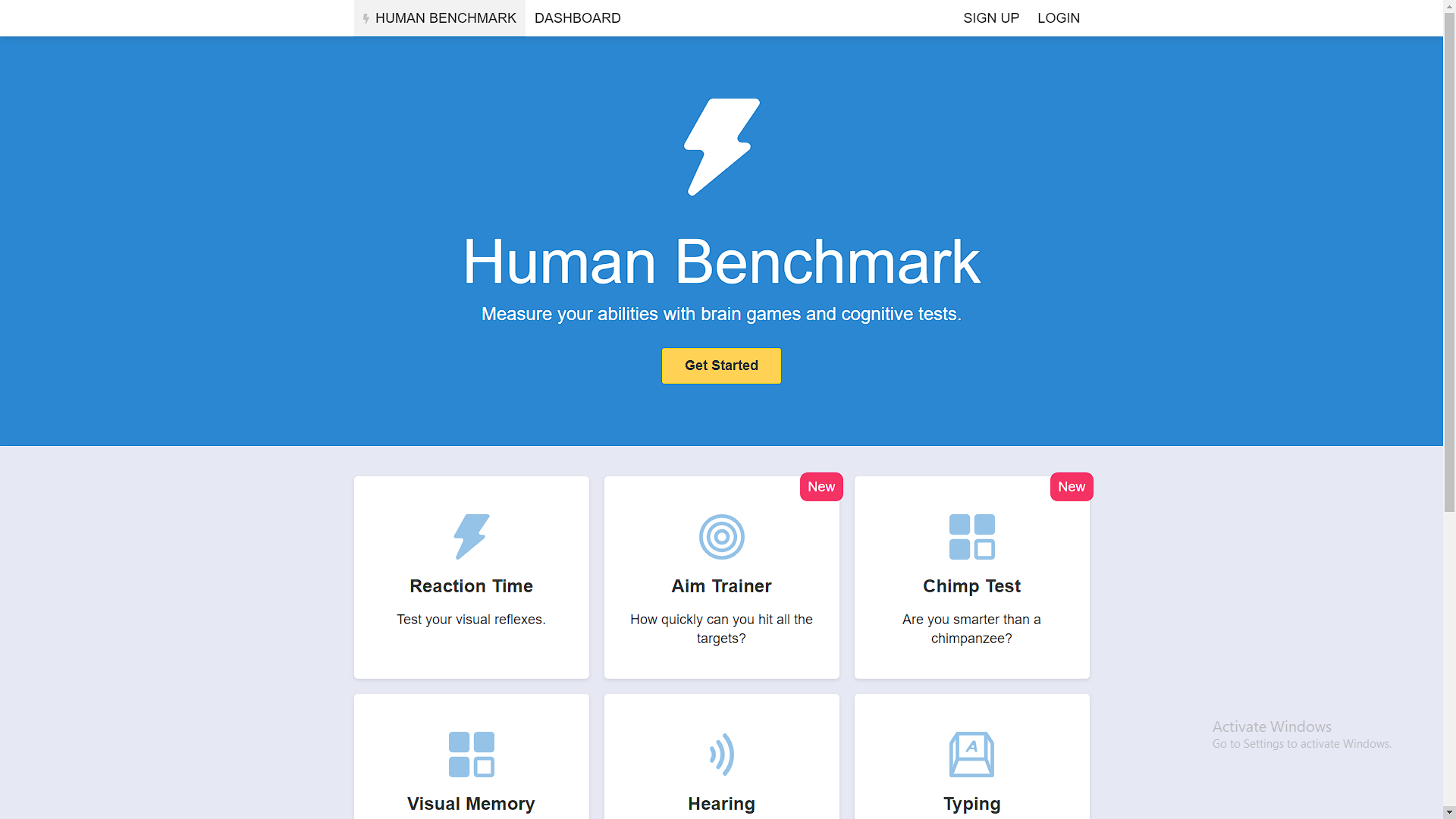
Pro:

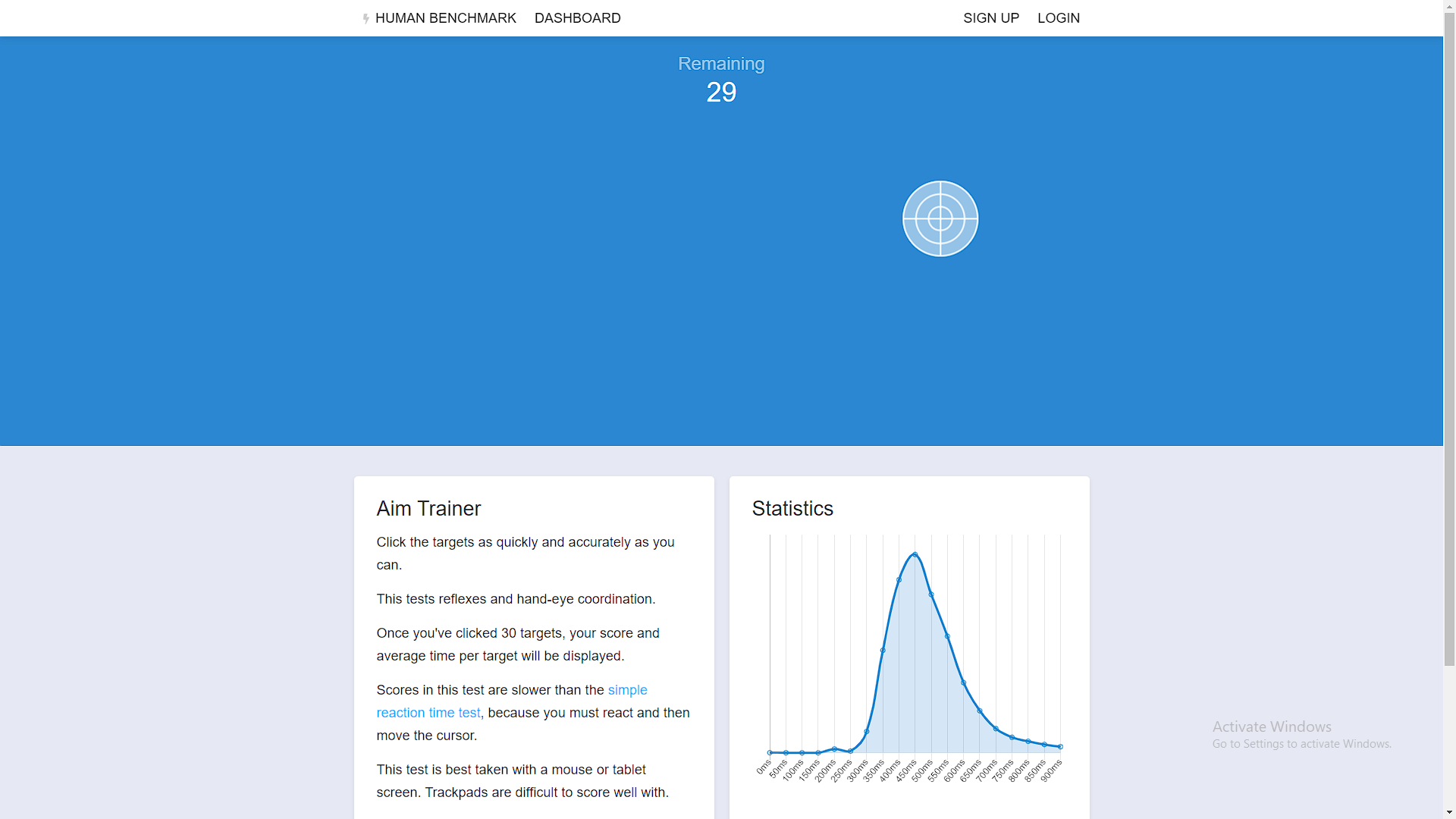
One thing that I want to adapt and put into my project from this program is that on each mini game there is a leaderboard that shows all the top scores and the names of the people that got it. There is also a range of settings you can change on each game mode like if there is any timer you can change that and you can change the number of objects you can hit and even the difficulty you can also change your sensitivity.

Cons:

This aim trainer is not compatible with the controller and too overwhelming and complicated as there are too many options. Too complicated, overwhelming. You also need a steam account to play this game and is not independent.

**Human benchmark**



Humanbenchmark.com is a website that claims to Measure your abilities with brain games and cognitive tests. This includes an aim trainer, a reaction time test and many memory games and typing speed test. This is a more broad competitor than other aim trainers and as a consequence only has 1 aim training game. The website has a very simplistic design with a nice cold colour palette which does not over complicate the website. It also shows the stats on all the games to show you how good you are at them, due to it being a website its very accessible and you must download anything to make it work all u need is a browser. The aim trainer itself is very basic and does not even have any settings to change. All it is a series of circles you have to click on then it tells how long it took you to click on 30 and it also tells you how long on average it took you to click on each one.

Pros:

Some of the things that I like about this website is how clean and simple the design is and how easy it is to just go on the website and practice without even having an account. There’s also a reaction test on the website and a broader amount of tests which can bring in a wider audience or a second demographic

Cons:

This website also does not support controllers and does not even let you change your sensitivity or even any settings. It also only offers one aim trainer which is very weird as all other aim trainers have at least 3 different games you can practice on. The website is also very limited due to the fact that it isn't a program but just a website so it can only be played with a stable internet connection. This is one of the problems I am overcoming but making it a downloadable program.

**Things I will implement:**

**Requirements**

**Limitations:**

One of the limitations is that the lack of a player base this solution will have will make it easy to be a top scorer on leaderboards which then makes it less challenging and engaging for the customers and the lack of minigames will narrow the skills you can improve.

**Hardware and Software:**

|  |  |
| --- | --- |
| **Requirements** | **Explanation** |
| Peripherals: Mouse, Keyboard, monitor,  Controller (DS4) | This is just normal peripherals just to interact with the program and pc, but the controller is just optional but recommended for a good user experience |
| Computer specs: 1.5 Mhz  512 MB RAM  1 GB storage  (change this when u finish code)  (but like raspberry pi can prob run it) | These are the computer specs needed to run Python and all the modules that I will use. |
| System Software:  Any OS preferably Windows | Any OS that can run python can be used |
| App software: Python with Pygame, Tkinter  maybe more stuff or less | Python and the modules stated are going to be the only things used |

**Design:**

|  |  |
| --- | --- |
| **Requirements** | **Explanation** |
| Have a quick way of instantly exiting or opening program | This is so people can open and play this game in between games or in queues and can take advantage of any small amounts of time |
| Simple layout that is easy to navigate | This is to not complicate this game as it is pretty simple |
| Clear settings and instructions to put the controller and calibrate settings | To not overcomplicate it and hopefully make it just connect instantly without calibration |
| Have a nice colour palette and be able to change it | To satisfy the customers’ colourful needs and in case of any colour blindness |

**Success Criteria:**

|  |  |
| --- | --- |
| **Requirements/want:** | **Evidence** |
| Have a login and registration | Screenshot of both the code and the window when the program is running |
| Be able to play as a guest | Screenshot of both the code and the window when the program is running |
| Be able to play with keyboard and mouse and controller | Screenshot of the code and a video of someone using both inputs |
| Have multiple game modes | Screenshots only game |
| Have a leaderboard that stores the top scores and the people/account that got the high score next to each of the games/mini games | Screenshots of the game leaderboard and the external file holding the data |
| Have a database to store stuff like account information, high scores, and user settings | Screenshot of the database and files and the code connecting the game to the database |
| Easy to run even extremely bad/old PC’s can run it | Screenshot the CPU, GPU load and how much RAM it takes, and storage needed to save the program |
| To be able to change certain in-game settings like timers and about of things | Screenshot of the setting menu and the proof of it affecting the game |
| Have instructions that are easy to follow | Screenshots of the instructions in-game |
| To be able to change colour in case of colour blindness | Screenshots of the setting to change the colours and the changing of colours |
| Have settings for sound volume | Video showing the change of volume |
| Be able to change controller and mouse sensitivity | Screenshots of the code, settings in-game and video evidence of the change of sensitivity |
| An information menu so the customers can understand how to navigate in the software | Screenshots of the menu and the code |
| Have a graph that can your scores over time | This is so that the user can see their improvement over time |
| Easy and simple to use GUI |  |

**Design of the solution:**

In this design segment, I will be mainly breaking down the problem into 5 smaller and easier to handle parts.

1. The Login, register and guest GUI

This will be the first step to the problem as this is the first thing the user will have to do, as every user must ever make an account, login or play as a guest. I have also made this a separate part as I will be making the login, register and guest GUI using Tkinter which a different python library to the rest of the program

1. The database

The making of the database will be my second step as this is going to be needed to complete the login, register and guest part of the solution as the database combined with the GUI is what I want as the final product. This is also needed to be a separate step as the database I will be making will not be using python but using external software instead. The database will not only be used for the accounts but will also be in the game itself, used to save scores and display scores the customers

1. Game menus

This is the start of the program I will be creating, and this is going to be the first part due to the fact of how simple and easy this will be compared to the rest of the program. This part of the program will be mostly design and little to no problem solving needed. This part of the program will also be needed to connect the game modes to the settings and to the other game modes.

1. All the different game modes

This is going to be the main part of the program. This is going to be the part with the most problem-solving skills required and the most time will be dedicated. In this section i will not only make the game modes but I will also integrate the already made database into the main program so scores can be viewed, added or manipulated.

1. The settings

The settings are where the user can change specific things about the game modes or rest of the program. The main reason for this is to create the best user experience designed for the specific user depending on their personal preferences.

The Login, register and guest GUI

The initial user interface will be coded using tkinter then as soon as you log in/register it will use pygame as user interface

Login

Register

Guest

**Welcome**

This is the main initial interface of my program and allows the user to play instantly as a ‘Guest’, make an account with ‘Register’ or ‘Login’ into their already made account. The benefit of making an account or logging in is that you can then save your high scores and the game will recognize you.

This also links to the success criteria:

* Easy and simple GUI
* Be able to play as a guest

If the guest option is clicked you will then be sent straight into the pygame game, but if register or login is clicked you will be sent to 2 different yet similar screens asking you to either register or login.

Login

Username:

Password:

**Game**

Back

This is going to be the screen u access when u click on the login option, the back button allows you to go back to the menu before this, in case you miss clicked, and the login button allows you to then go in the game if the username and password inputted was correct. There are 2 input boxes, one for username and one for password

Register

Username:

Password:

**Game**

Back

This is the registration menu which is very similar to the login menu but allows you to register and make a new account instead of using an old account or guest, back button allows you to go back to the initial menu and the register makes the inputted data an account

The Database

|  |  |
| --- | --- |
| users |  |
| User\_id | int |
| Username | varchar |
| password | varchar |

|  |  |
| --- | --- |
| score |  |
| Score\_id | int |
| user | int |
| game | int |
| score | int |

|  |  |
| --- | --- |
| games |  |
| Game\_id | int |
| name | varchar |

This is going to be the relational data base I will be planning on making

The Game Menus

Setting

Play Game

Log out

**Game**

This is the basic version of what will be able to be seen when you login, register or play as a guest. The play game option then allows you to be able to select a game mode to play. The setting option allows you to change the game setting like controls, sensitivity, colours, volume and much more and finally the log out or exit option will just do as said, logout or exit the game completely. The colour of this screen will be able to be changed in the settings.

The Game Modes

3

00:18

This is one of the mini-games that will be on my program and it consists of a series of boxes that u have to click on each giving a certain amount of points, whenever 1 box is clicked another comes up randomly (but not too close to the old boxes/ one shown(so some dumb algorithm)). There is also a score on the top of the screen so the player can see where he is and how well he is doing

**Variables used in main game program:**

|  |  |  |
| --- | --- | --- |
| **Variable name** | **Variable data type** | **Explanation** |
| Screen\_height | integer | Variable used to assign the height of the window and other objects are limited based on the this variable |
| Screen\_width | integer | Variable used to assign the width of the window and other objects are limited based on the this variable |
| FPS | integer | How many times the main game loop runs per second |
| timer | float | How long game has been running, this is based on fps.will be used for a timer on the screen and to test for times in between presses |
| loop | Boolean | Variable that controls the game loop |
| score | integer | Variable to keep track of the score |

**Variables needed for game 1**

|  |  |  |
| --- | --- | --- |
| **Variable** | **Variable data type** | **Explanation** |
| object\_height | integer | The height of the object used for the hit box of the object and drawing of the object |
| objest\_width | integer | The width of the object used for the hit box of the object and drawing of the object |
| colour | tuple | Colour of the object and this can be changed depending on the settings |
| coords | list | Used to know the location where the object is drawn on the screen |
| health | integer | The health of the object used on specific games, for most objects this value will be 1 |

**Variables needed for game 2**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| top\_speed | integer |  |
| Acceleration |  |  |
| current\_speed\_x |  |  |
| current\_speed\_y |  |  |
| Object\_height |  |  |
| object\_width |  |  |
| colour |  |  |
| coords |  |  |
| health |  |  |
|  |  |  |

**Game 3**

**Pseudocode for one of my games**

This is a function that generates a new box that is a set distance away from the previous box

Function new\_box(old\_x,old\_y,object\_height,object\_width):

loop = True

While loop == True:

x = random.number(1,1280 - object\_width)

y = random.number(1,720 - object\_height)

If x > 200 and y > 300 : #this is to check if the boxes are over the score/time

X\_diff = old\_x - x

Y\_diff = old\_y - y

If ((x\_diff\*\*2)+(y\_diff\*\*2))\*\*0.5 > 50: # so box can't be in a radius around

loop = False

List = [x,y]

Return list

Game\_loop = True

coords = [720/2 , 1280/2] # this is going to be the first box

Object\_height = 50

Object\_width = 50

Score = 0

While Game\_loop == True: # loops 50 times per second

FPS = 50

Draw.rectangle(object\_height, object\_width, coords[0], coords[1])

If rectangle is clicked:

Score = score + 1

Coords = new\_box(x,y,object\_height,object\_width)

Score : 75

Play Again

Exit

Menu

After a game is finished, the score is displayed alongside any other statistics that are specific to the game. Also, u will be able to see a leaderboard of all the top scores of all the players that played the game and saved the scores.

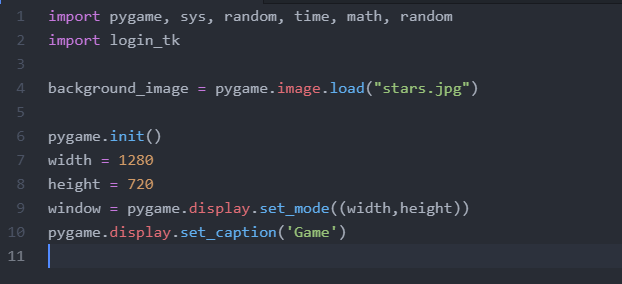
User thoughts:

Rayane: ‘I like it but it's a bit basic, maybe u can add other stats like how long it takes to click each box or accuracy (how many clicks were off the objects)’

The Settings

Development:

In the development of this project, I will be using two different files to simplify the coding process, the first file I will be making is the game which includes the GUI for the main game, the settings and the games themselves all written using the python module pygame. The other will be the login and registration file which will be written using the python module tkinter.



This is the start of my program, this is where all the python modules I am using are imported, where the image I will be using will be loaded and where the main pygame variables will be made

The first thing I did is make a simple yet useful function that draws text on the screen

