# GAMING ARCADE

**A Project Report**

***Submitted by***

## VAISHNAVI RATHOD (E016)

## RIYA TENDULKAR (E047)

***Under the Guidance of***

## PROF. KAMAL MISTRY

***in partial fulfillment for the award of the degree of***

## BACHELOR OF TECHNOLOGY

**COMPUTER SCIENCE**

At



**MUKESH PATEL SCHOOL OF TECHNOLOGY AND MANAGEMENT, MUMBAI**

**APRIL, 2020**

## DECLARATION

I, **Vaishnavi Rathod and Riya Tendulkar**, Roll No. **E016 and E047**, B.Tech (Computer Engineering), IV semester understand that plagiarism is defined as anyone or combination of the following:

1. Un-credited verbatim copying of individual sentences, paragraphs or illustration (such as graphs, diagrams, etc.) from any source, published or unpublished, including the internet.
2. Un-credited improper paraphrasing of pages paragraphs (changing a few words phrases, or rearranging the original sentence order)
3. Credited verbatim copying of a major portion of a paper (or thesis chapter) without clear delineation of who did wrote what. ( Source: IEEE, The institute, Dec. 2004)
4. I have made sure that all the ideas, expressions, graphs, diagrams, etc., that are not a result of my work, are properly credited. Long phrases or sentences that had to be used verbatim from published literature have been clearly identified using quotation marks.
5. I affirm that no portion of my work can be considered as plagiarism and I take full responsibility if such a complaint occurs. I understand fully well that the guide of the seminar/ project report may not be in a position to check for the possibility of such incidences of plagiarism in this body of work.

Signature of the Students: \_\_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Names: Vaishnavi Rathod, Riya Tendulkar

Roll Nos. : E016, E047

Place: Mumbai

Date:

## CERTIFICATE

This is to certify that the project entitled “**Gaming Arcade**” is the bonafide work carried out by **Vaishnavi Rathod** and **Riya Tendulkar** of B.Tech, MPSTME (NMIMS), Mumbai, during the IV semester of the academic year 201-20, in partial fulfillment of the requirements for the Course Programming Language.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Prof. Kamal Mistry**

Internal Mentor

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Examiner 1 Examiner 2

## TABLE OF CONTENTS

**CHAPTER NO. TITLE PAGE NO.**

1. INTRODUCTION 5
2. SOFTWARES AND API USED 6

WITH DESCRIPTION

1. METHODS IMPLEMENTED 8
2. SCREENSHOTS 16
3. CONCLUSION & FUTURE SCOPE 33
4. SOCIETAL APPLICATION 33

**INTRODUCTION**

A big reason why we love games so much is because it is an escape from our boring and monotonous lives. They also give us sense of accomplishment. The feeling you get after playing a good game and winning it is unbeatable. Games are such recreational activities of our lives that make us forget everything else. They leave us wanting more.

With our knowledge of the python programming language we tried to make some of our favorite games for everyone to enjoy. We have attempted to give our little spin of these wonderful games, some of which have been a part of our childhoods.

These games are for everyone to enjoy, they guide those who are new to the game too.

You can easily login and retrieve your previous scores and play on. Even if you are not a registered user, you can register and play. We have included 5 games in our project: Guess the Logo, Tic Tac Toe, Hangman, Snake and the Color Game.

A brief description of the games is:

**Guess the logo**: Logos of famous companies will be displayed and you will have to guess the names of the companies.

**Tic Tac Toe**: The classic X and O game to play with your friend.

**Hangman**: The person needs to guess all the letters of the word before the stick figure is hanged.

**Snake**: The player needs to navigate the snake to its food on the screen.

**Color game**: It is mind tricking game where the player must answer the colour of the word and not the text of the word.

Member details of the project are:

|  |  |  |
| --- | --- | --- |
| **ROLL NO.** | **NAME** | **CONTRIBUTION** |
| E016 | Vaishnavi Rathod | * Hangman * Color Game * Guess the Logo |
| E047 | Riya Tendulkar | * Snake * Tic Tac Toe * Guess the Logo |

Using this project, we have tried to learn and implement GUI using Tkinter, Database connectivity using sqlite3 and PyGame.

**SOFTWARE AND API USED**

**Python IDLE**

We have used Python IDLE and SQLite3 for our project. A brief description of them is:

The**Python IDLE** (**Integrated DeveLopment Environment**) **editor** is a graphical user interface for Python development. This GUI is free and installed automatically during the Python installation. It enables us to edit, run, and debug Python programs in a simple GUI environment.

IDLE is actually a Python program that uses the standard library’s **tkinter** **GUI toolkit** to build its windows. It is portable and can be run on all major platforms. It supports the following features:

* command history and syntax colorization
* auto-indent and unindent for Python code
* word auto-completion
* support for multiple windows
* integrated debugger

In IDLE we have used the following main libraries:

* **Tkinter**- Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications.
* **Sqlite3**- SQLite is a C library that provides a lightweight disk-based database that doesn’t require a separate server process and allows accessing the database using a nonstandard variant of the SQL query language.
* **Pygame**- Pygame is an open source module in python which can be used to develop games. PyGame makes it really easy to develop games as it tracks all the moments on the display screen of the game. It easy to understand and implement as well.
* **MessageBox**- The messagebox module is used to display the message boxes in the python applications. There are the various functions which are used to display the relevant messages depending upon the application requirements.
* **Random**- This module is used to generate random numbers from the given range.
* **PIL**-Python Imaging Library (abbreviated as PIL) is a free library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats.

**SQLITE 3**

SQLite is a very popular database which has been successfully used with on disk file format for desktop applications. **SQLite** is a [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS) contained in a [C](https://en.wikipedia.org/wiki/C_(programming_language)) [library](https://en.wikipedia.org/wiki/Library_(computer_science)). In contrast to many other database management systems, SQLite is not a [client–server](https://en.wikipedia.org/wiki/Client%E2%80%93server) database engine. Rather, it is embedded into the end program. SQLite is [ACID](https://en.wikipedia.org/wiki/ACID)-compliant and implements most of the [SQL](https://en.wikipedia.org/wiki/SQL) standard, generally following [PostgreSQL](https://en.wikipedia.org/wiki/PostgreSQL) syntax.

Normally, an RDBMS such as MySQL, PostgreSQL, etc., requires a separate server process to operate. The applications that want to access the database server use TCP/IP protocol to send and receive requests. This is called client/server architecture. SQLite does not work this way and does not require a server to run. SQLite database is integrated with the application that accesses the database. The applications interact with the SQLite database read and write directly from the database files stored on disk.There are a lot of advantages to use SQLite as an application file format:

1. Lightweight
2. Better Performance
3. No Installation Needed
4. Reliable
5. Portable
6. Accessible
7. Reduce Cost and Complexity

**DB BROWSER**

It is not possible to read the data from SQLite directly as the format in which the data is stored is not understandable. Hence, we need a browser to help us to read the data stored. For this purpose we use the DB Browser.DB Browser for SQLite is a high quality, visual, open-source tool made for creating, designing, and editing database files that are compatible with SQLite. It is for users and developers who want to create, search, design and edit [databases](https://www.edureka.co/blog/what-is-a-database/). SQLite browser uses a general spreadsheet-like interface, and there is no need to learn complicated [SQL commands](https://www.edureka.co/blog/sql-commands). It is a tool that is used by both developers and end-users, and for that reason, it has to remain as simple as possible.

## ****Uses of SQLite Browser:**** It is a tool that lets us view the data that is stored in an SQLite Database. Depending on the format and type of data in the database it may or may not be readable by a human. This is generally used for debugging or other development tasks where the developer needs to read the data that has been stored but does not have a built-in system to access it through the program.

**METHODS IMPLEMENTED**

**GAME 1: TIC TAC TOE**

**GAME INTRODUCTION**

This is our basic X and O, played with two players.

One player is X and the other is O and they need to play the game alternately. The game structure of Tic Tac Toe is:

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

We have a 3x3 grid. The users will have to alternately select which box they want their symbol to go. The player who first gets his symbol in 3 continuous boxes wins.

**WINNING CONDITION**

The winning condition is that the player should get his symbol in three consecutive boxes. For that to happen there should be ‘X’ or ‘O’ for the following boxes:

Rows Columns Diagonals

1=2=3 1=4=7 1=5=9

4=5=6 2=5=8 3=5=7

7=8=9 7=8=9

After 9 turns( as there are 9 boxes) if there is no winner, then it is a tie.

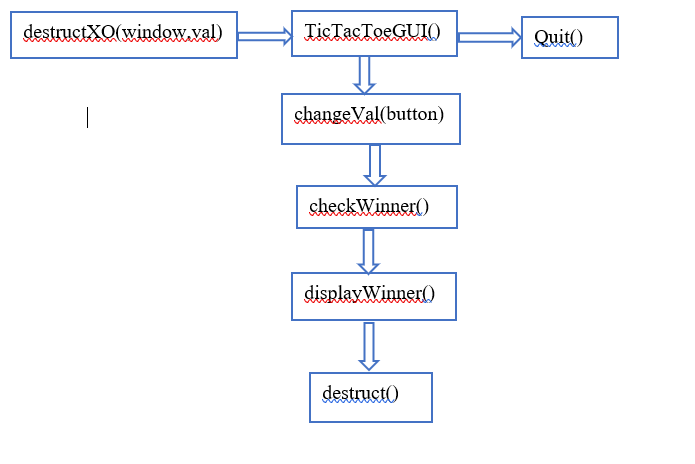
**LOGIC BEHIND THE CODE:**

There are 9 buttons. Every time any button is clicked, it changes its text value to the symbol of the player. The symbol is tracked by the alternative nature of the game. After every click the winning conditions are checked.

**CODE FUNCTIONS AND THEIR DESCRIPTION**

|  |  |
| --- | --- |
| **FUNCTION NAME** | **DESCRIPTION** |
| destructXO(window,val) | This function destructs the window of the previous function and retrieves and stores the ID of the player, received from the previous function. |
| Quit() | If the player clicks the quit button then this displays a warning message confirming if the user wants to quit despite having chances. If the user wants to quit, the window and game will be destroyed. If he decides against it then the game resumes. |
| Destruct() | After the final score and data is displayed and the user proceeds to exit, this function destructs the entire game and the windows. |
| displayWinner() | This function displays who the winner is. It also displays the high score from the database and the players current score. If the player from the database wins then his score is updated in the database. |
| checkWinner() | This function checks who the winner is. If the above winning conditions are satisfied by ‘X’ then player 1 and if are satisfied by ‘Y’ then player 2 wins. If both the players don’t win and 9 turns are over, then it is a tie. |
| changeVal(button) | This function changes the values of the 9 buttons which are the grid. If the player clicks on the button then the value of the button changes to ‘X’ and ‘O’ depending on which player selects the button. |
| TicTacToeGUI() | This is the main function the game which displays the entire grid composed of 9 buttons. |

**FLOW OF FUNCTIONS**



**GAME 2: THE SNAKE GAME**

**GAME INTRODUCTION**

This game has two elements-snake and the food. The snake needs to eat the food. The player can control the snake using the arrows(up, down, left, right). After the snake eats one box of food, another one is creating on the screen. Every time the snake eats food, its length and speed increases. The game goes on till the snake doesn’t die.

Game over conditions:

The snake hits his own body

The snake hits the screen boundaries.

**LOGIC BEHIND THE CODE**

* **Food**

The food is rectangle box placed on the screen using the random function. the x and y coordinates of the food are generated by the random function. Every time the snake eats one rectangle, another one is created.

* **Snake**

The snake is a list of rectangles. We have a list of rectangles which is the body of the snake. Every time the snake eats food, we append a new rectangle to the list. This list keeps a track of the x and y coordinates of each rectangle. So it is a list in list where the first list keeps tracks of the rectangles and these rectangles keep a track of the x and y coordinate.

* **Changing the speed of the snake**

Every time the snake eats food, its speed increases. This is done by changing the clock.tick() parameter. The clock tick controls the frames per second of the screen of the game. Whenever the snake eats, we increase the speed by 1 and then pass that value to clock.tick()

* **Losing conditions**

1. Hitting the boundaries.

We check the x and y coordinate of the rectangle from the snake list. If it is more than or equal to the boundary of the screen then the game gets over because it means that the snake has hit the screen.

1. Hitting itself

Before displaying the snake, every time the user presses a key, we check if the new coordinates are equal to the coordinates of the rectangles already present in the snake.

**CODE FUNCTIONS AND THEIR DESCRIPTION**

|  |  |
| --- | --- |
| **FUNCTION NAME** | **DESCRIPTION** |
| destructMain(window,val) | This function destructs the window of the previous function and retrieves and stores the ID of the player, received from the previous function. |
| gameOver() | This function displays the score of the player, high score, previous total score and current score of the player. |
| increaseSnake(a,b,snake) | The function appends a rectangle to the snake list, i.e., it increases the body of the snake. ‘a’ and ‘b’ are the coordinates of the rectangle and snake is the list. |
| displaySnake(snake) | This function is used to draw the rectangles on the screen and display the snake. we display the snake in 2 colors, every alternate rectangle is displayed in a different color. |
| SnakeMainGame() | This is the main snake game where the main pygame module is implemented. |

**GAME 3: GUESS THE LOGO**

**GAME INTRODUCTION**

The logo game is a game where you will be displayed logos of famous companies and you need to enter the names of the company they belong to. Upon entering the correct name for the respective logo, you get points.

**LOGIC BEHIND THE GAME**

The images of the logos are stored in a list and their respective names are stored in another list. One by one the image is displayed to the player and he must enter the correct full name of the logo to score points. The pictures of the dictionary are displayed one by one and the user needs to enter the name. The word entered by player is compared with the name of the image displayed.

When he clicks the submit button, the name is checked, if it is correct then the player gets 1 points otherwise 0.

You win:

* If you enter the correct name of the logo with perfect formatting.
* One point is scored for every correct answer

You lose:

* You enter an incorrect answer, the game ends.

**FLOW OF FUNCTIONS**

Checklogo()

logogameGUI()

destructLogoGame()

exit

newgame

**FUNCTIONS AND THEIR DESCRIPTION**

|  |  |
| --- | --- |
| **Function name** | **Description** |
| DestructLogoGame  (window,val) | This function destructs the window of the previous function and retrieves and stores the ID of the player, received from the previous function. |
| Score() | This function updates and displays the score on the screen when a correct answer is entered. |
| new\_game() | This function is called after the player enters a correct answer. He is then displayed another logo which he must guess. It also updates the score of the game. |
| exit() | This function displays the score of the player, high score, previous total score and current score of the player. |
| checkLogo() | This function checks if the name of the logo entered by the user is the same as the name of the logo stored in the list. A message box after comparing displays whether the answer entered by player was correct or wrong. |
| logogameGUI() | It is the main function that displays all the instructions, label, numbers and entry for user of the program. It sets the window for the logo game and calls the function to start the game. |

**GAME 4: HANGMAN**

**GAME INTRODUCTION**

The hangman game is a game where you will be displayed the word in the form of blank dashes. Since the no of letters of the words are the no of dashes, the player must guess all the letters of the word. The player has advantage if the word contains double letters. If he guesses one correct, the rest duplicate letters are also displayed. If he guesses incorrect letter, one life is lost. With every life that is lost, the stick figure moves a step closer to being hanged. We can see one by one his body parts start to appear with every wrong guess. The player must guess all the correct letters before the lives run out.

**LOGIC BEHIND THE GAME**

In this game, I have implemented the words that belong to python language. A list is created to store all the words. A random function is used to shuffle the words in the list. The dashes are displayed on the window equal to the length of the word. The letter entered by the player is checked in a function. If it is present in the word, all the positions where it is present are displayed. Then it keeps on checking whether the word formed is equal to the word chosen from the list.

You win:

* If all the letters entered constitute the word and the player is left with one life or more. The player scores one point if he wins

You lose:

* If he guesses incorrect letter, one life is lost.
* If all 11 lives are lost, the game ends for the player

**FLOW OF FUNCTIONS**

guess()

hangmanGUI()

destructHangmanGame()

exit

newgame

**FUNCTIONS AND THEIR DESCRIPTION**

|  |  |
| --- | --- |
| **Function name** | **Description** |
| DestructHangmanGame  (window,val) | This function destructs the window of the previous function and retrieves and stores the ID of the player, received from the previous function. |
| new\_game() | This function is called after the player enters a correct answer. He is then displayed another word which he must guess. It also updates the score of the game. |
| exit() | This function displays the score of the player, high score, previous total score and current score of the player. |
| guess() | This function checks if the letter of the word entered by the user is the same as the letter of the word stored in the list. A message box appears after comparing, displays whether the answer entered by player was correct or wrong. It updates the correctly guessed letters if any. |
| HangmanGUI() | It is the main function that displays all the instructions, label, numbers and entry for user of the program. It sets the window for the hangman game and calls the function to start the game. |

**GAME 5: COLOR GAME**

**GAME INTRODUCTION**

The game is a simple experiment of the stroop effect. It refers to the delayed reaction times when the colour of the word doesn’t match the name of the word. It is easier to say the colour of a word if it matches the semantic meaning of the word. For example, if someone asked you to say the colour of the word “blue” if it is printed in blue would be much easier to say than if it was printed in “green”.

**LOGIC BEHIND THE GAME**

A list contains all the names of the words that will be displayed on screen and they are shuffled randomly. There is button present to start the timer. The player must enter maximum correct answers in the give time period i.e before the clock runs out. For every correct answer the score increases by one. The word displayed keeps on shuffling so does the colour to ensure the player remains of his toes throughout the game.

The game ends when timer turns 0. The score till then is the final score of the game. No answer is entertained after the time is over.

**FUNCTIONS AND THEIR DESCRIPTION**

|  |  |
| --- | --- |
| **Function name** | **Description** |
| DestructColourGame  (window,val) | This function destructs the window of the previous function and retrieves and stores the ID of the player, received from the previous function. |
| new\_game() | This function starts the timer for the game and calls the function for the player to enter the answer he deems right. |
| exit() | This function displays the score of the player, high score, previous total score and current score of the player. |
| guess() | This function checks if the letter of the word entered by the user is the same as the letter of the word stored in the list. A message box appears after comparing, displays whether the answer entered by player was correct or wrong. It updates the correctly guessed letters if any. |
| Count() | If the timer is more than 0 seconds, this function keeps of decreasing the timer for the game. If timer is 0, it exits the game |
| ColourGUI() | It is the main function that displays all the instructions, label, numbers and entry for user of the program. It sets the window for the colour game and calls the function to start the game. |

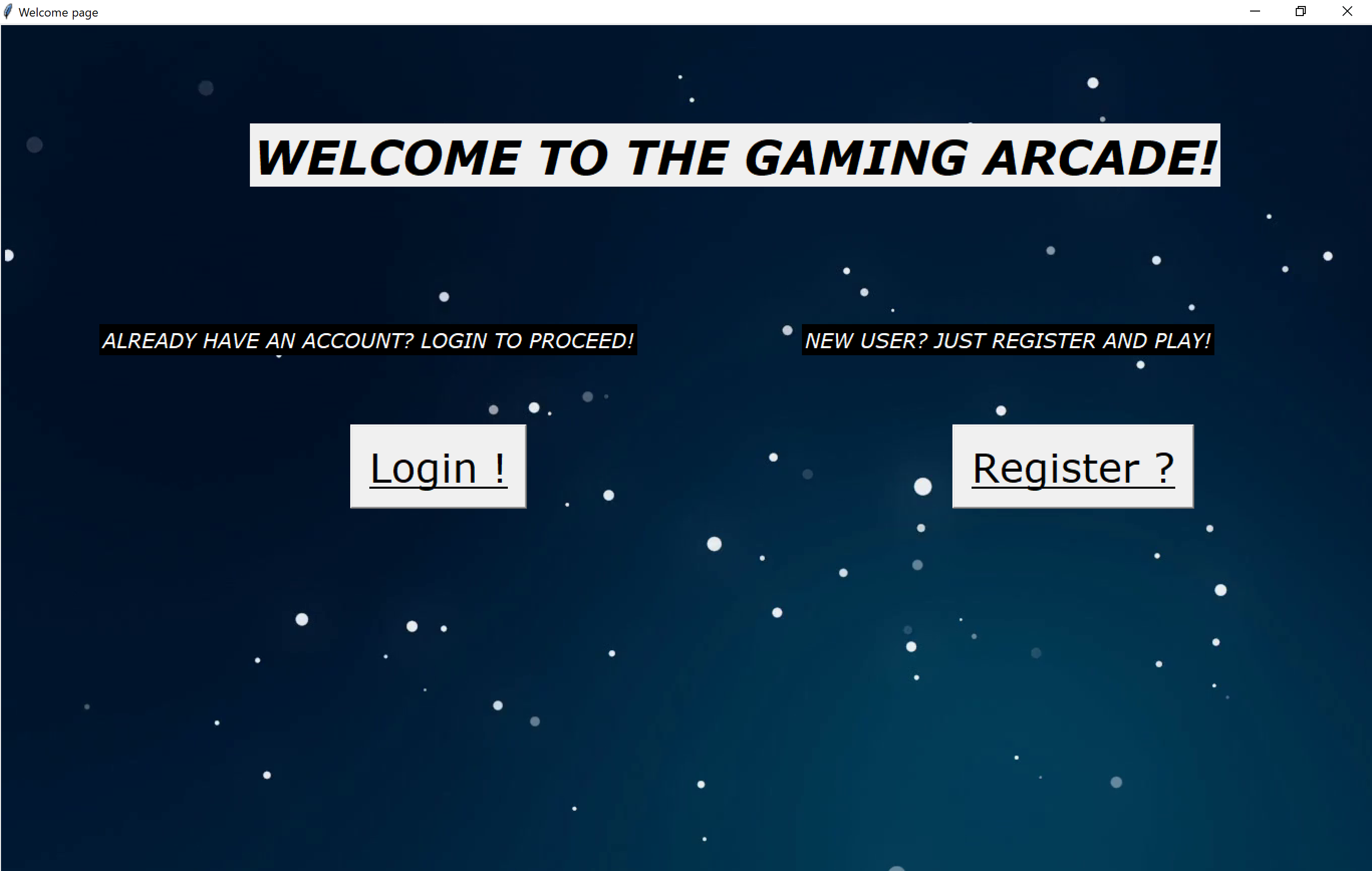
**FLOW OF**

**FUNCTIONS:** destructColourGame 🡪 colourGUI 🡪 new\_game 🡪 enter, count 🡪exit

**SCREENSHOTS**

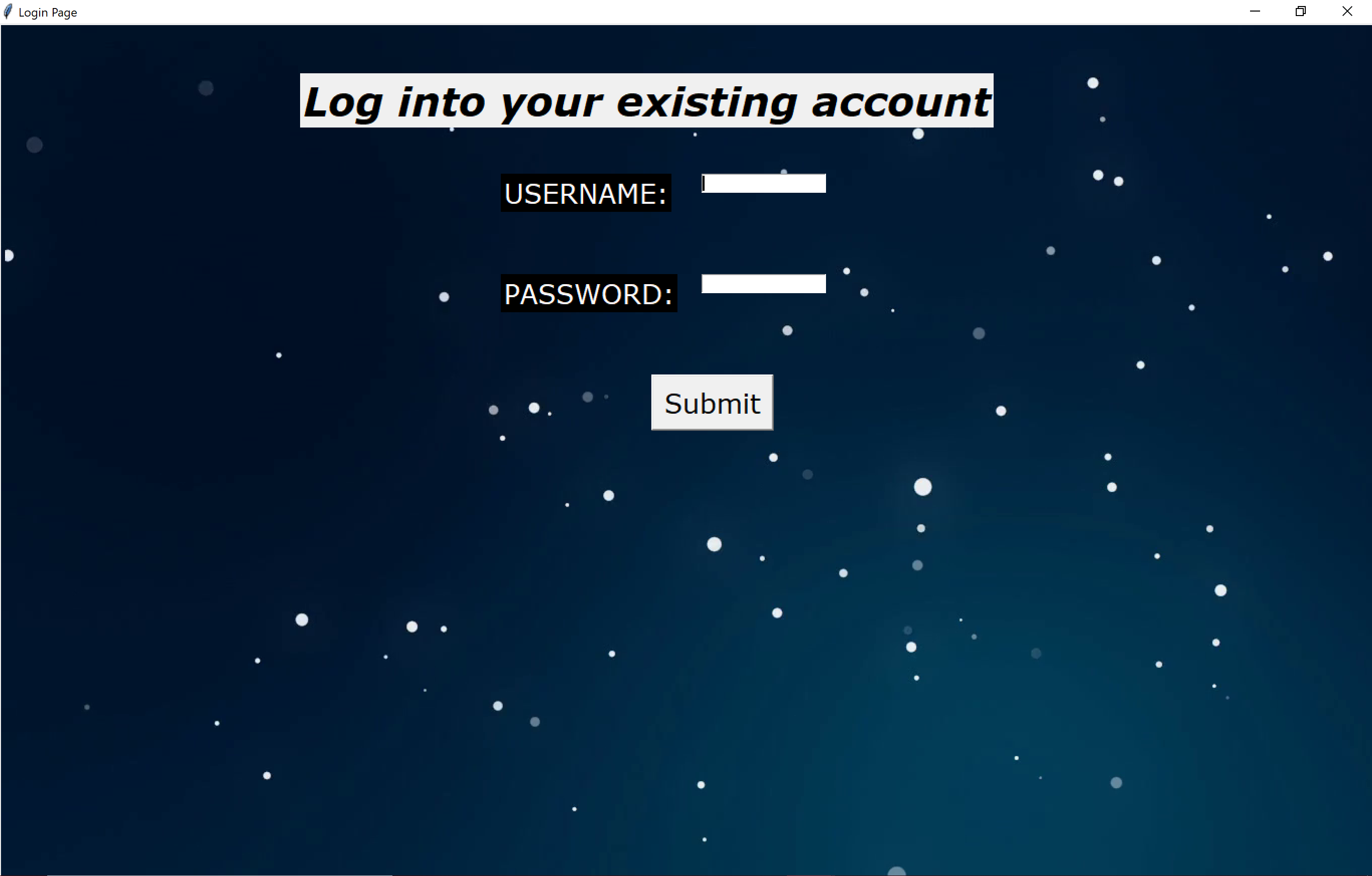
**1. WELCOME**

In the start of the program we have the welcome page that serves at the beginning point for the player. If the player is already an user, they might just login. If they aren’t they might have to register. Two buttons are available with the respective functions.

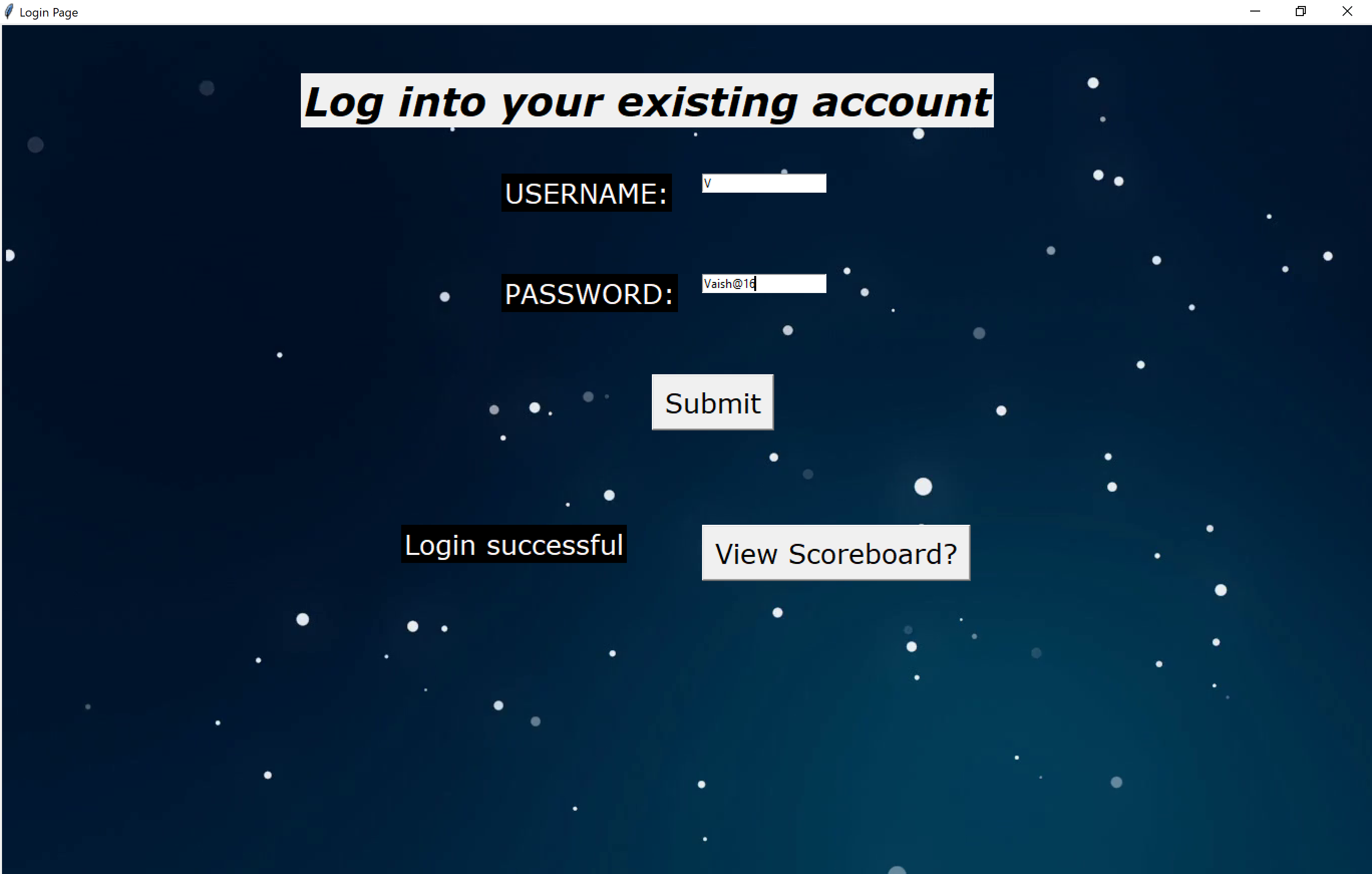


**2. LOGIN**

After the person is led to the login page, they will have to enter their login details, ie the username and password to login.



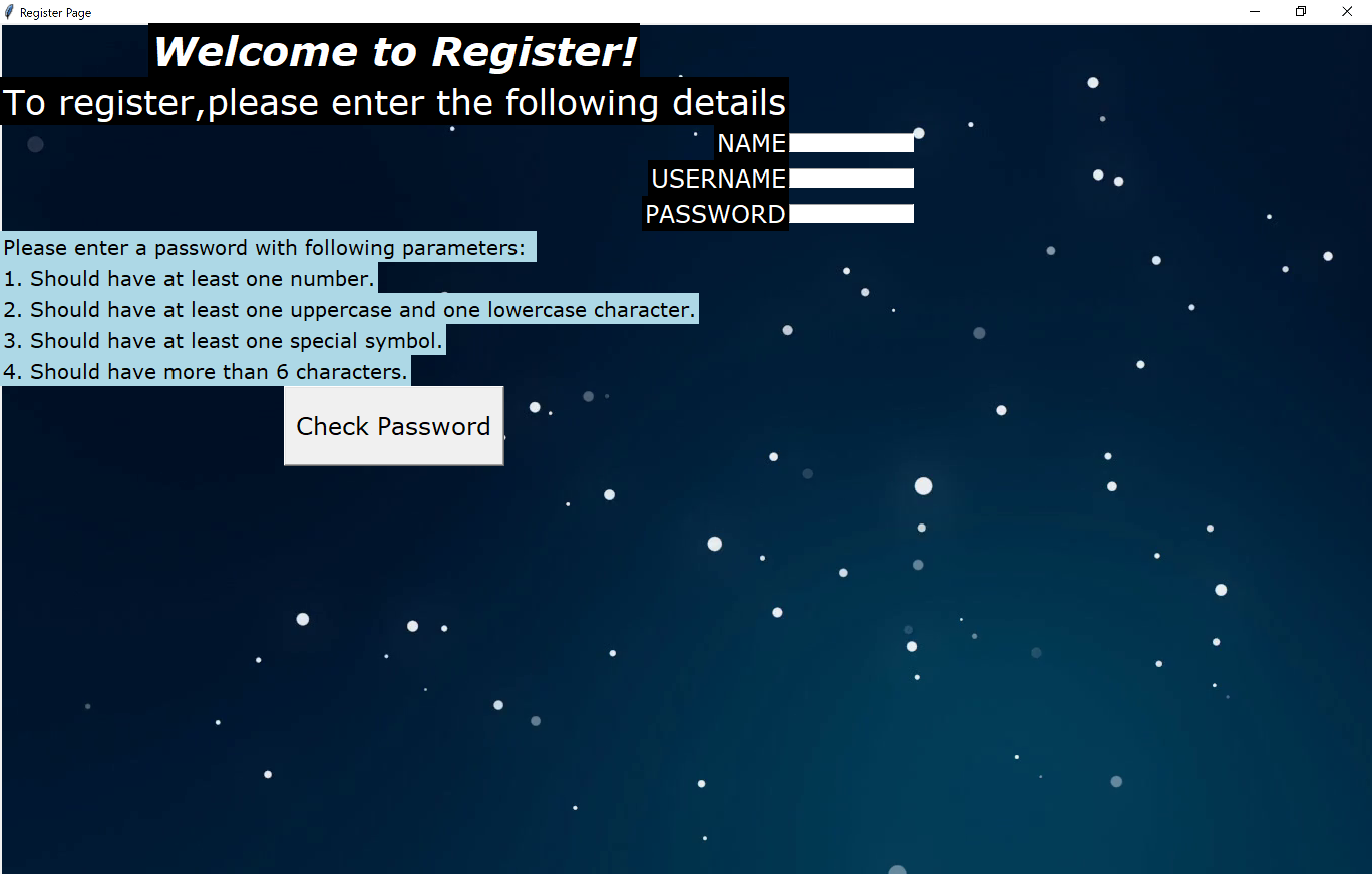
After the person enters the correct login information,



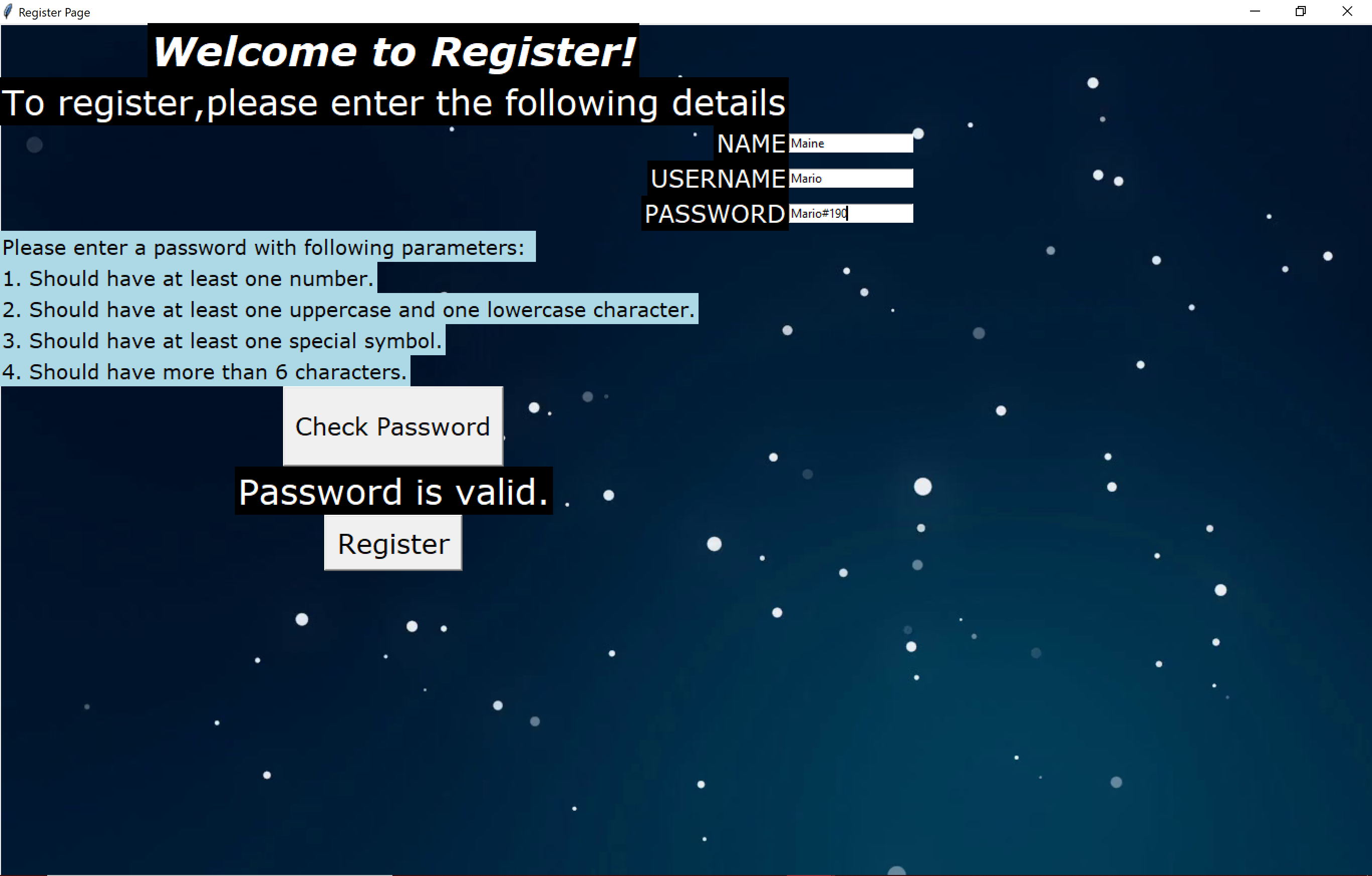
They are displayed the message that login was successful and they get the option to view the scoreboard containing their scores from previous games.

**3.REGISTER**

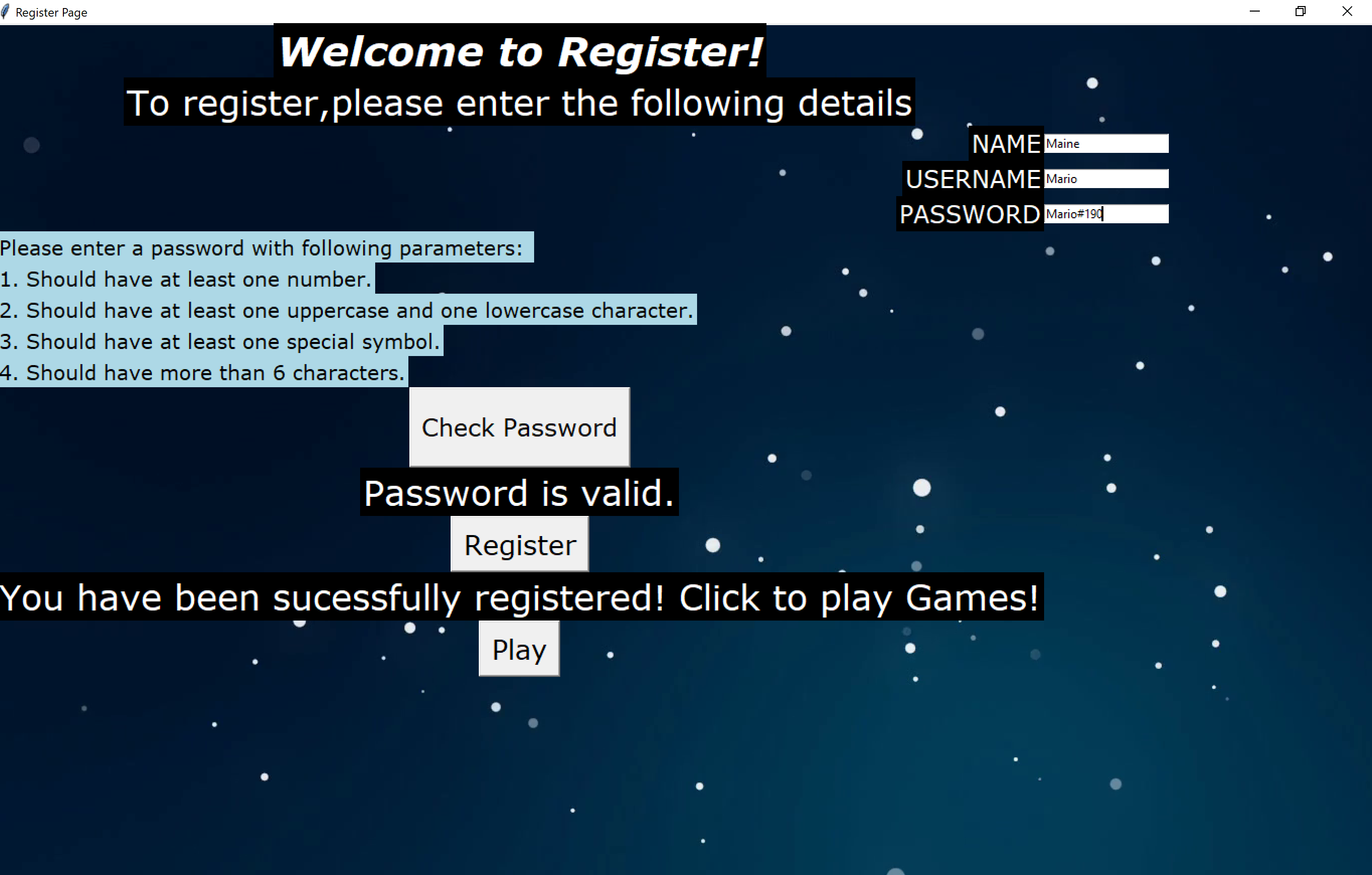
For those who haven’t registered in the system, they must click on the register button. After they click they are displayed,



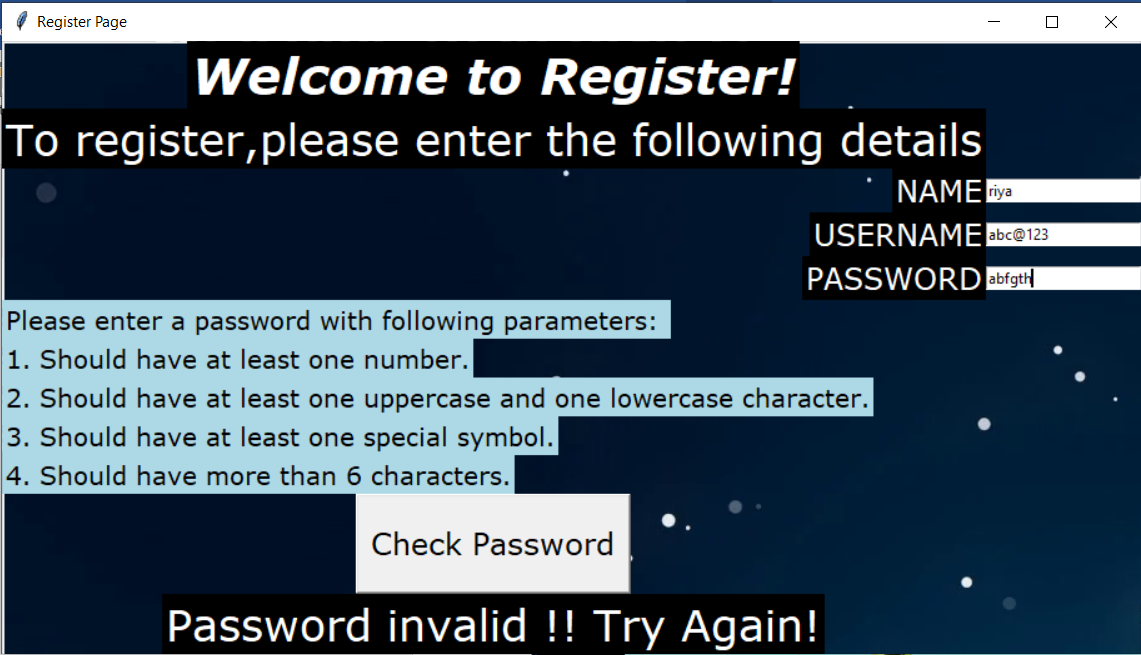
The user is prompted to enter the password with certain rules to make it more secure and not easily accessible to anyone.



Since the password requirements are met, the password is valid and the person can register.



If the user enters an incorrect password he gets the following error:



**4. SCOREBOARD**

This is displayed to only the players that login. Their previous total scores in all the games are shown.



**5.HOME PAGE**

After both the ways, the login and register, when the player clicks on the play button he is led to the home page that displays all the games that are available to be played.

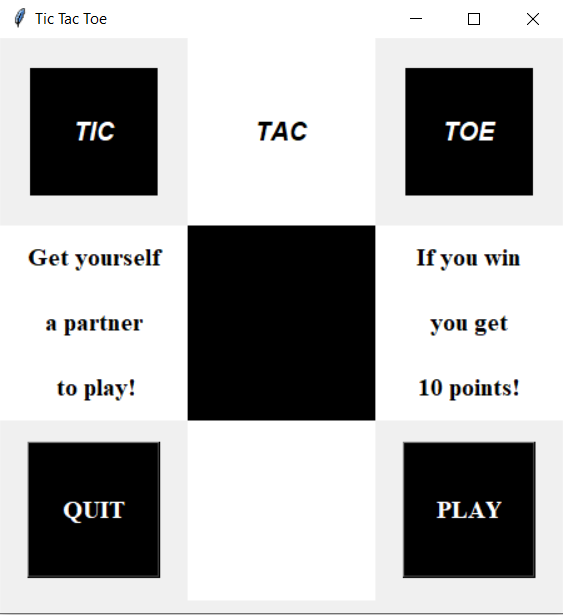


The player can choose which game to play out of the five by clicking on the button of the game to be played.

***Tic tac toe***

This is the main page of the logo game:

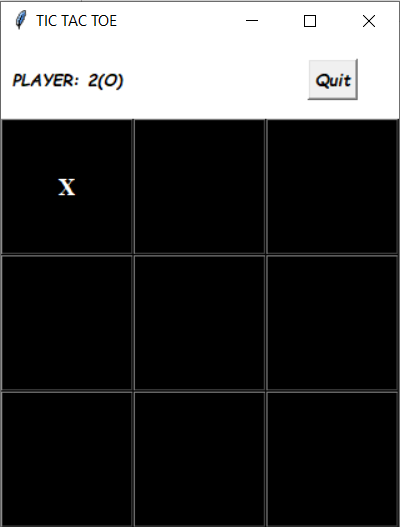
Upon clicking on play you are directed to the game and on clicking quit, the game is destructed.



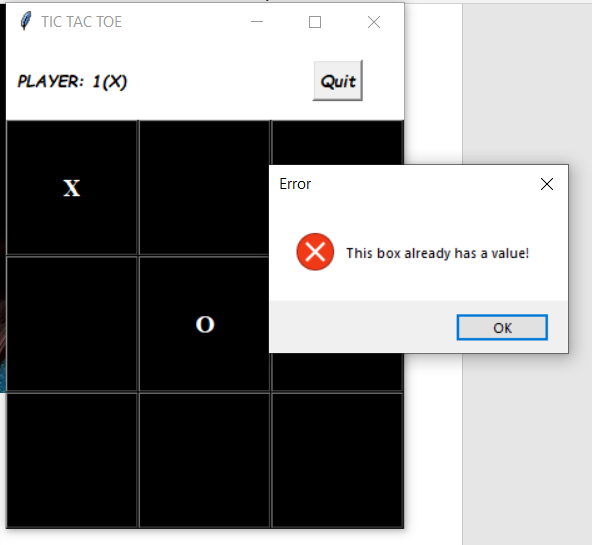
This is the main page of the game. The top left corner displays whose turn it is to play.



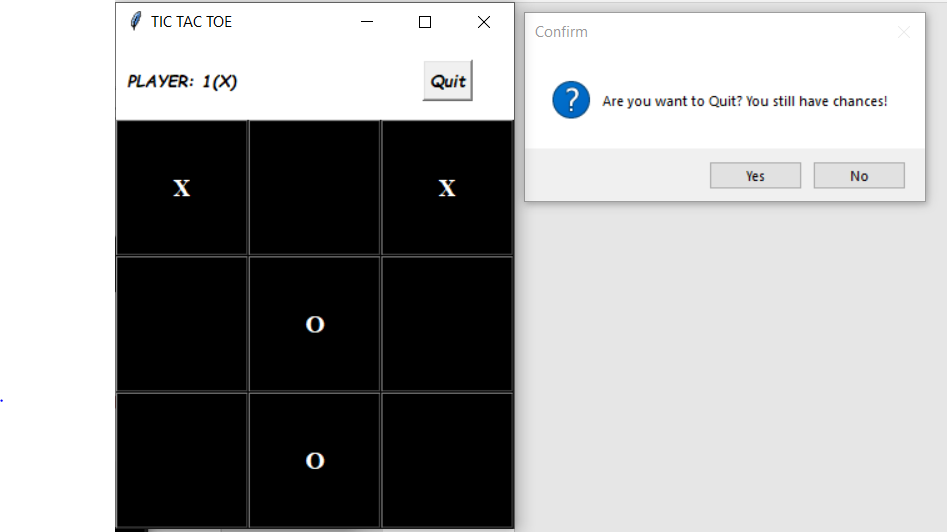
Upon selecting a box, that player’s symbol appears on the game.



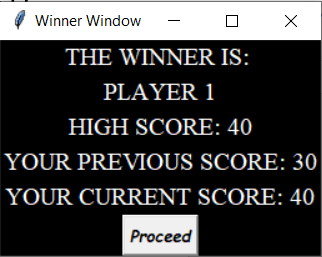
If the players tries to select the box which already has a symbol it gets an error:



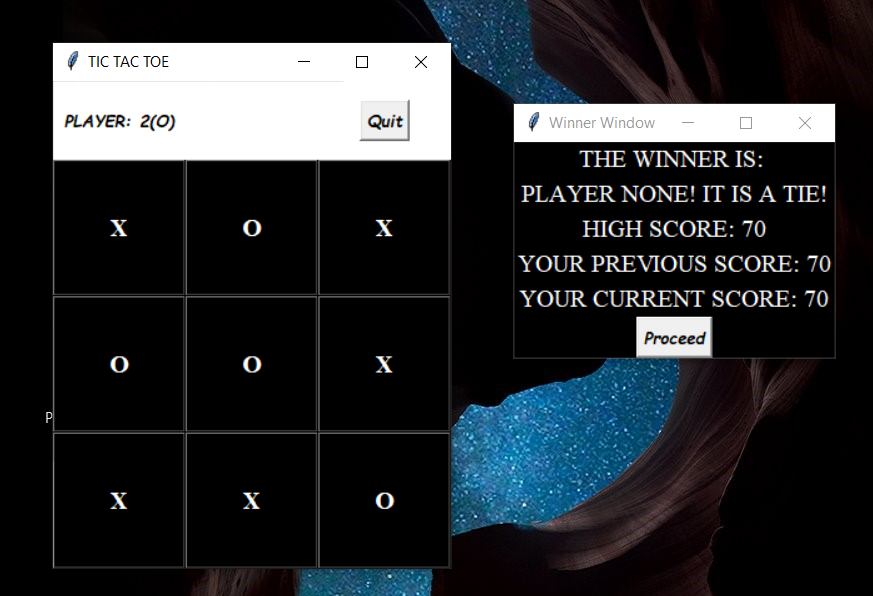
If the user tries to quit when he already has chances then a message is displayed:

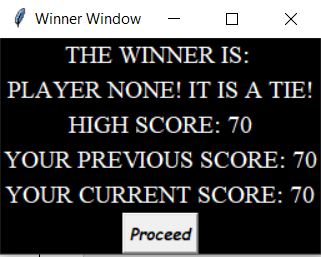


When any player wins, his score is displayed:



In case of a tie:



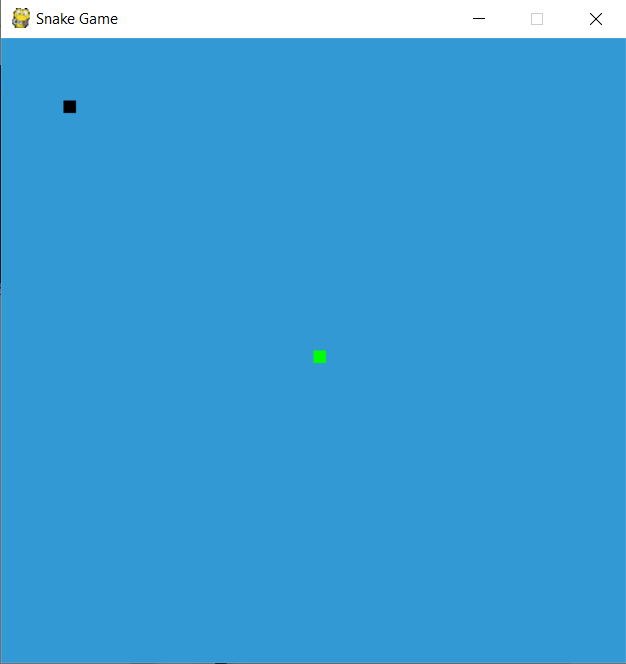


***Snake Game***

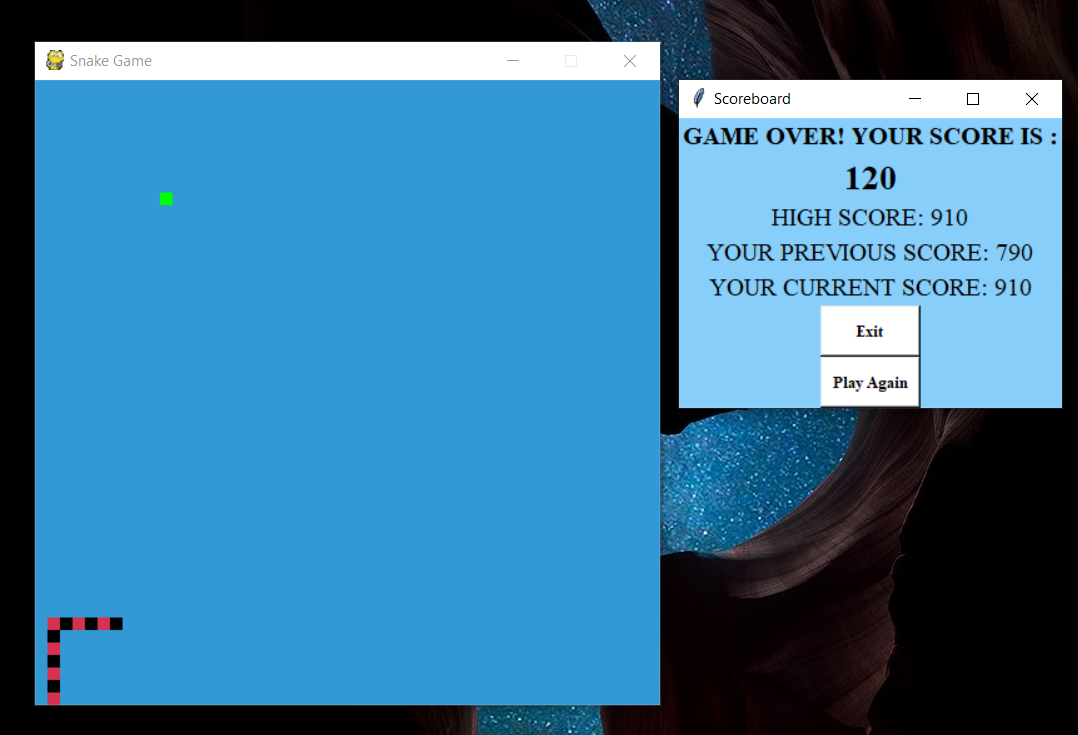
This is the main page of the snake game, it gives a description of the rules and regulations and the winning conditions. Upon clicking play, you will be directed to the main page.



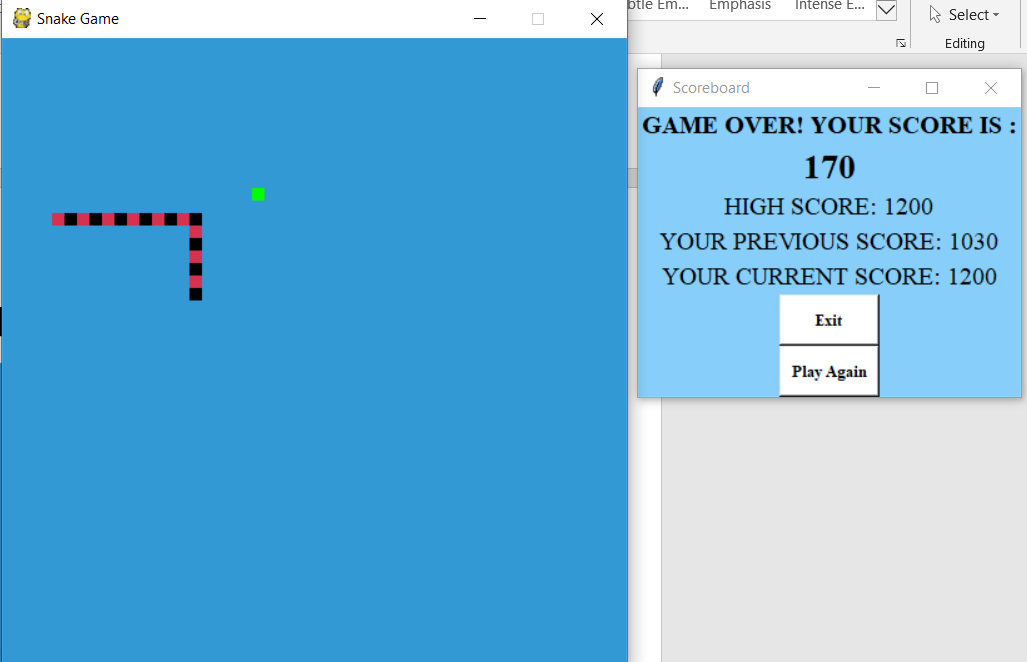
This is the main window of the snake, the black rectangle is the snake and the green box is the food.



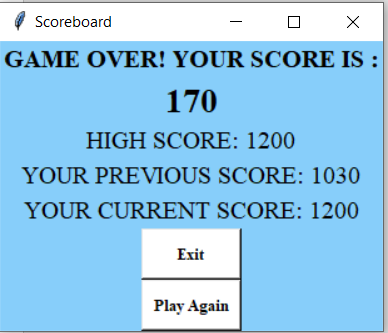
When the snake hits the screen of the window, the game gets over:



When the snake hits itself, the game gets over too:



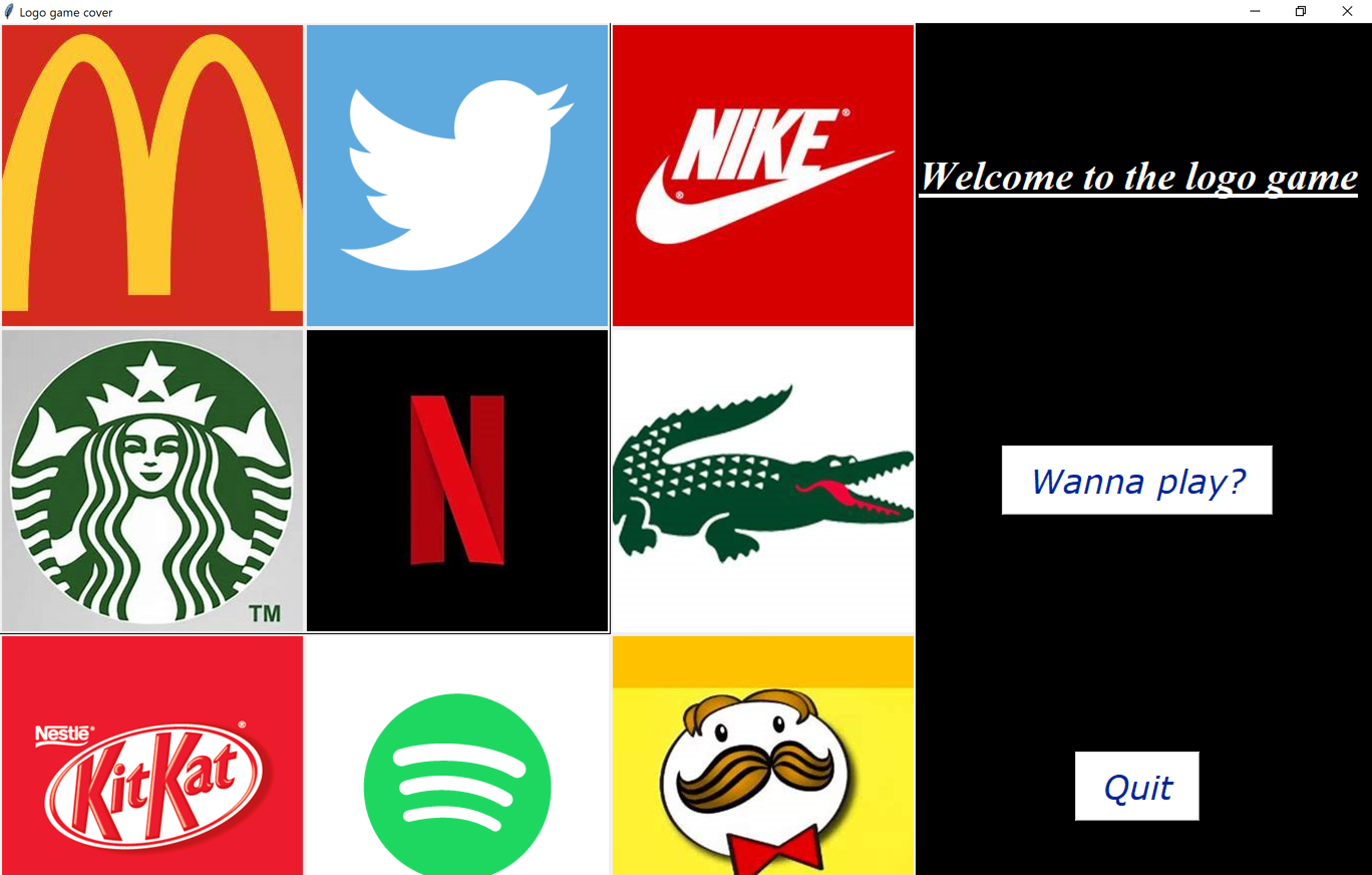
This is the winning window:



Upon clicking play again, the game starts again.

***Logo Game***

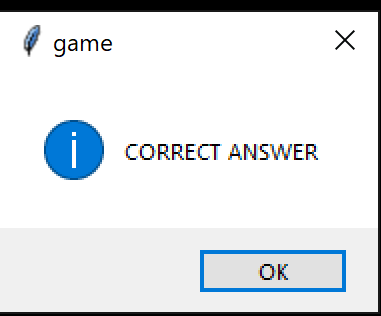
After clicking on the button, the person is led to this page which gives him a preview of how the game is.



If he clicks on quit, the game ends and he is led to the homepage. If the player clicks on the play button, he is led to the guess the logo game to play.

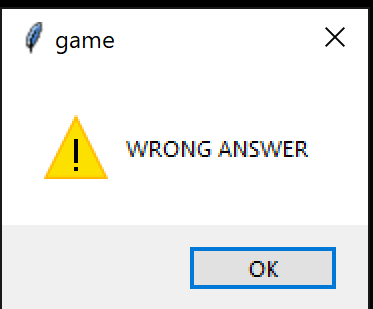


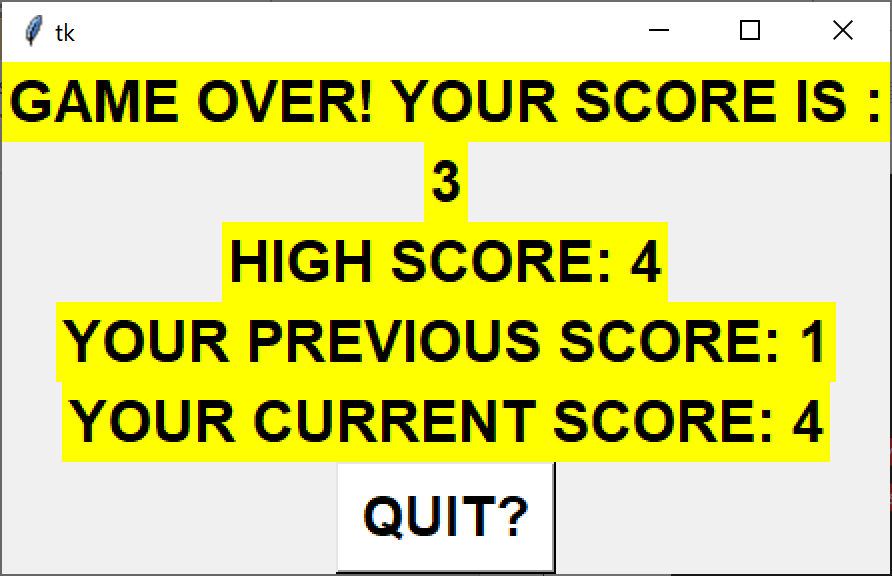
This is the main game. The player must guess the game of the logo displayed correctly to score a point. If the answer is correct, the player scores a point and is displayed another image. If the answer is wrong the game ends.





Since the answer entered is incorrect, the game ends here. The game can also be ended by clicking on the quit button on the left corner top.

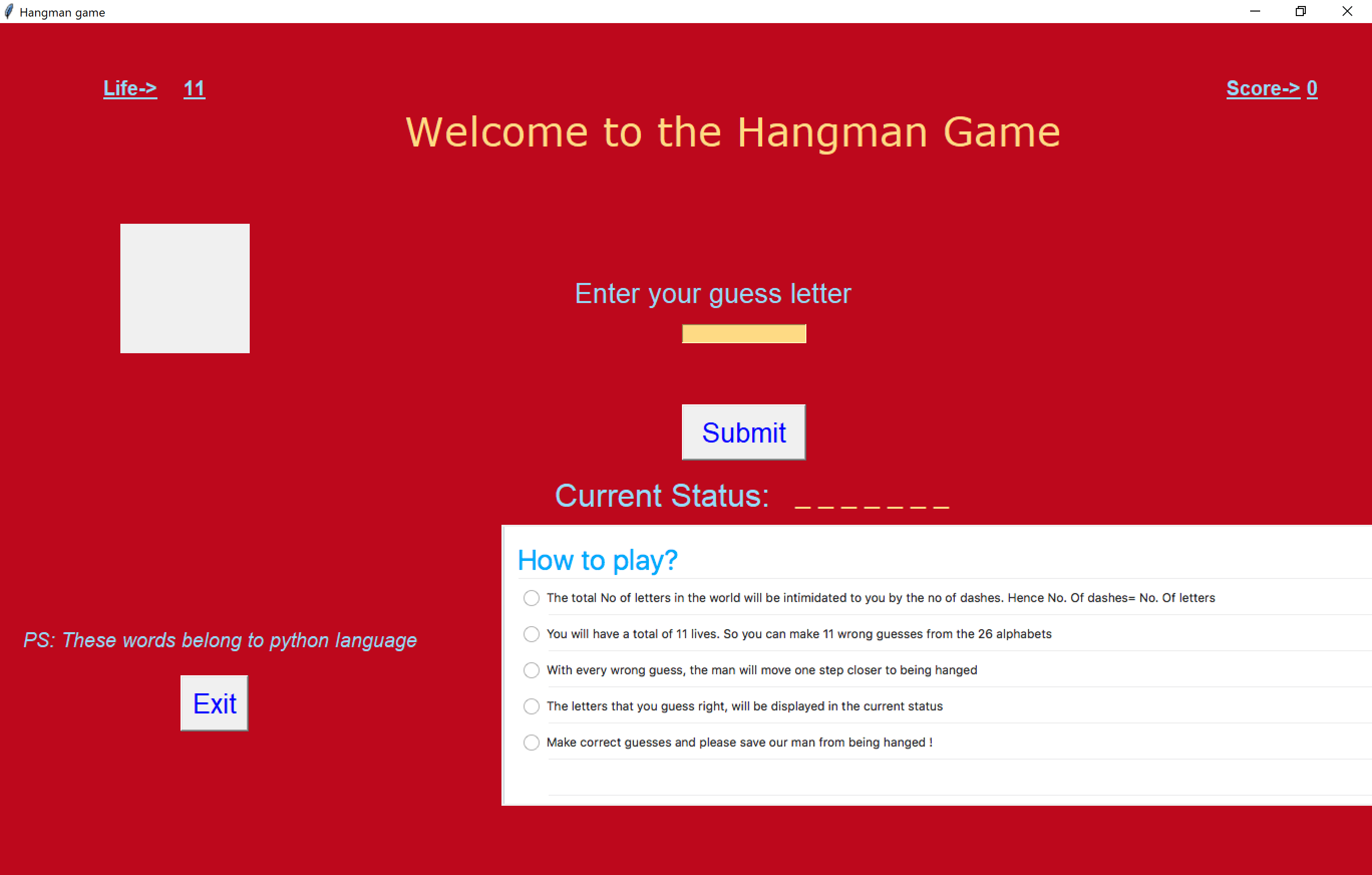




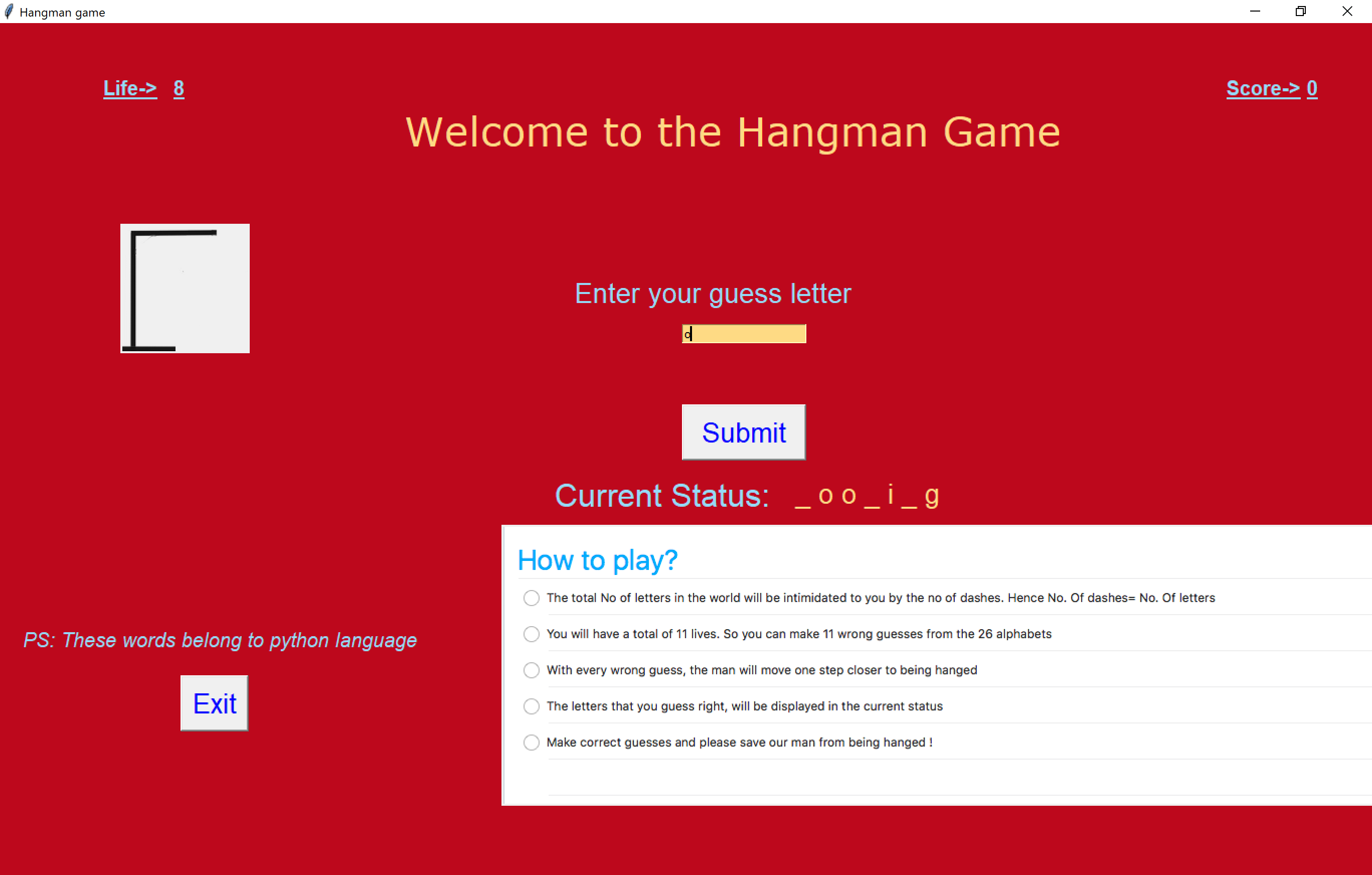
This window is displayed after the game ends. It displays the score of the game, the high score, the previous and current score for the game. These scores get stored in the database to be referred again.

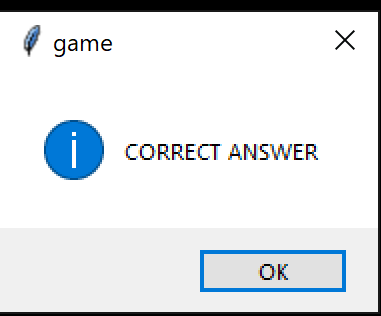
***Hangman Game***

The game starts when the hangman button is clicked in the homepage.

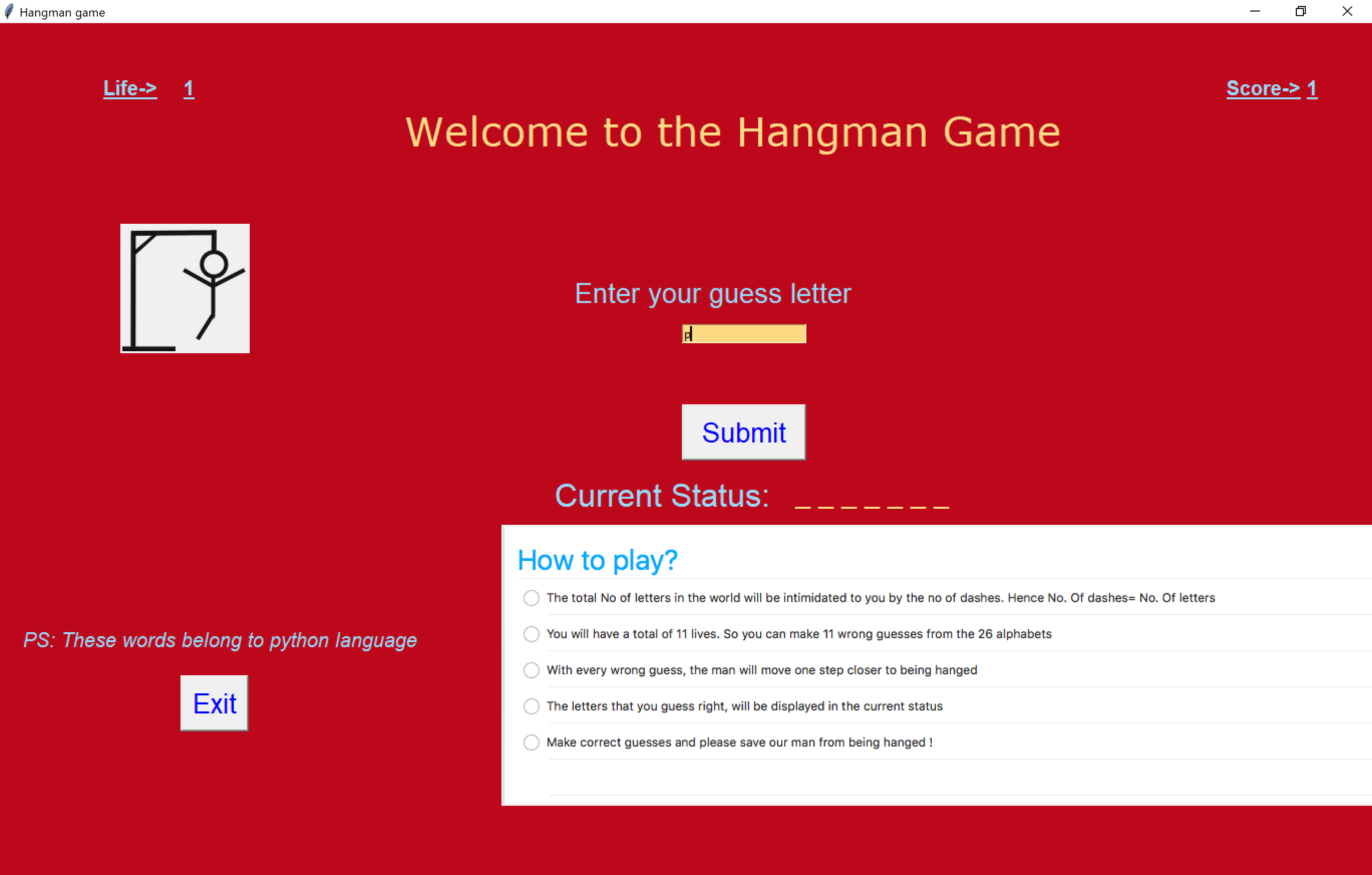


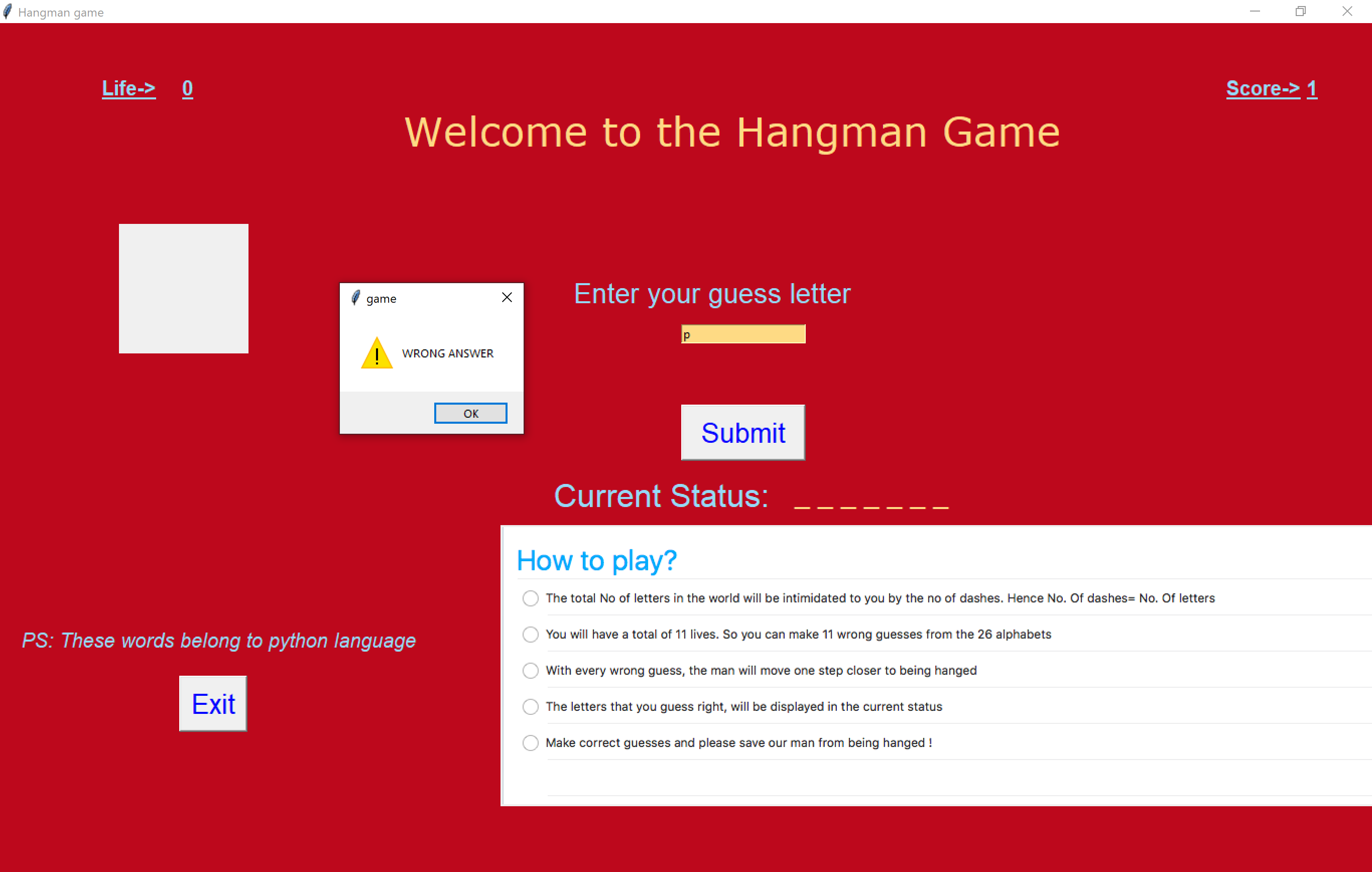
The rules of the game are also displayed to help the user play the game. For every right guess, the letter appears in its position in the word. For every wrong guess one life is lost and body parts of the hangman start appearing.

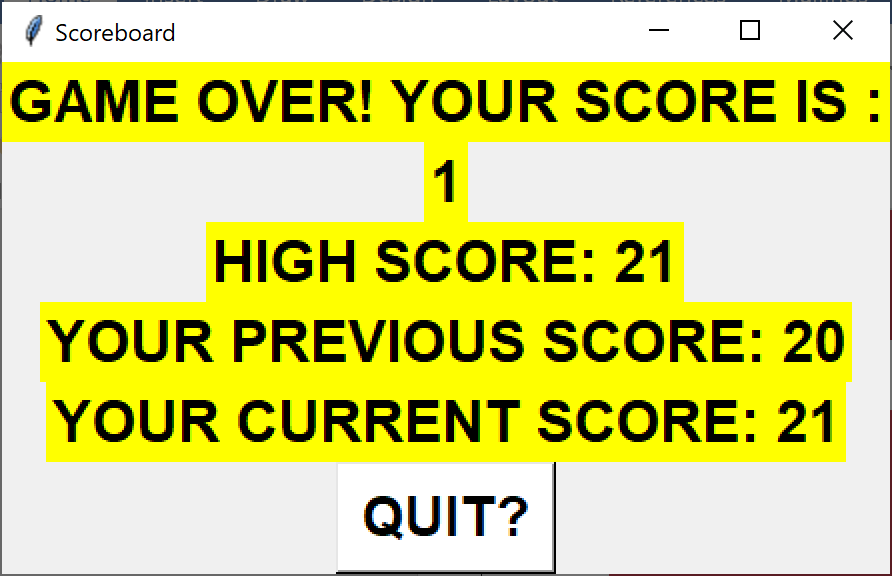




The game ends when all lives are lost and the word isn’t guessed properly. It can also end if the user selects the exit button.







***Colour Game***

This game too starts with the click of the button and this window is displayed.

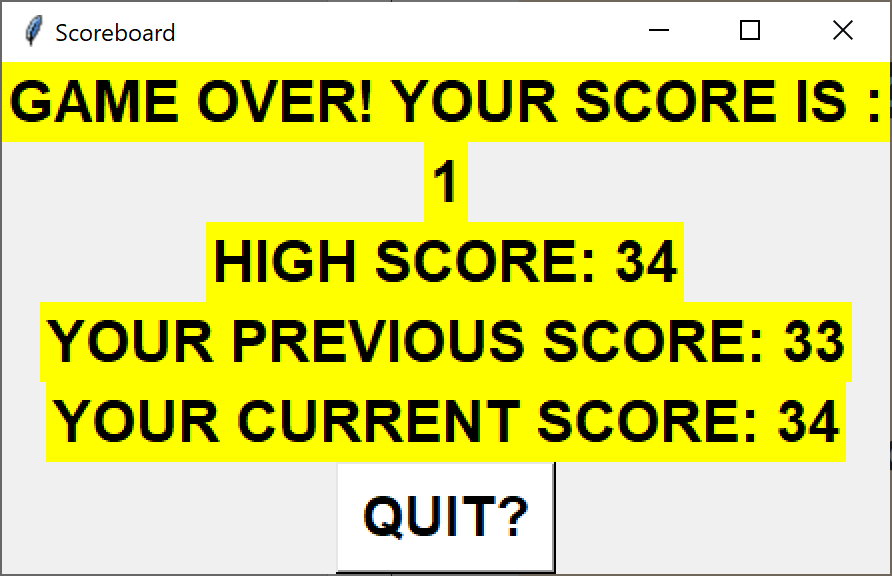


The user has to type the name of the colour and not the word. So blue would be a correct answer and red will be the wrong answer.

After the enter button is clicked the timer starts and the user must click the button to submit the answer too.



Enter maximum correct answers before the timer runs out and ends the game.



**CONCLUSION**

We have successfully made a project Gaming Arcade for our project this semester for the Python subject. Doing this project helped us understand the various concepts of Python and helped us compile all the knowledge we learnt in the course of the semester and apply it to make a front end and back end project. We have tried incorporate and merge the important concepts like GUI, database connectivity, PyGame, regular expressions in a very fun way. We have tried to make the project as self-understanding as possible to make it easy for the players to play. We are thankful for the opportunity we received to make this gaming arcade and learn to implement the concepts we learnt in class.

The project was as fun for us to make as it is for anyone to play.

**FUTURE SCOPE**

We aim to add more games to our arcade. Currently we have just 5 games and we wish to add more. We also wish to add levels to our game. Currently our games don’t have levels and it is something we would like to explore. We would also like to record the time taken for the player to complete a game and then add it to the scoreboard. We would like to add multiplayer games as well. Also in the current arcade we have only one game which uses PyGame; we would like to add more games using PyGame.

**SOCIETAL APPLICATION**

Games can be played by anyone. Then be it a small child or an office going adult who is bored of his work. The games in our arcade are normal simple games which everyone has played one time or the other in their life. They are simple and easy to understand and play. We hope that people have fun playing at our arcade and are able to easily navigate through the games.