

Python for Computational Problem Solving

UE23CS151A

Mini-Project

SEMESTER-1



HANGMAN

Shakirth Anisha-PES2UG23AM095

Saijyoti Panda -PES2UG23CS512

UNDER THE GUIDANCE OF,
PROF. DIVYA EBENEZER NATHANIEL
ASST. PROF
DEPT. OF CSE
PES UNIVERSITY

PROBLEM STATEMENT

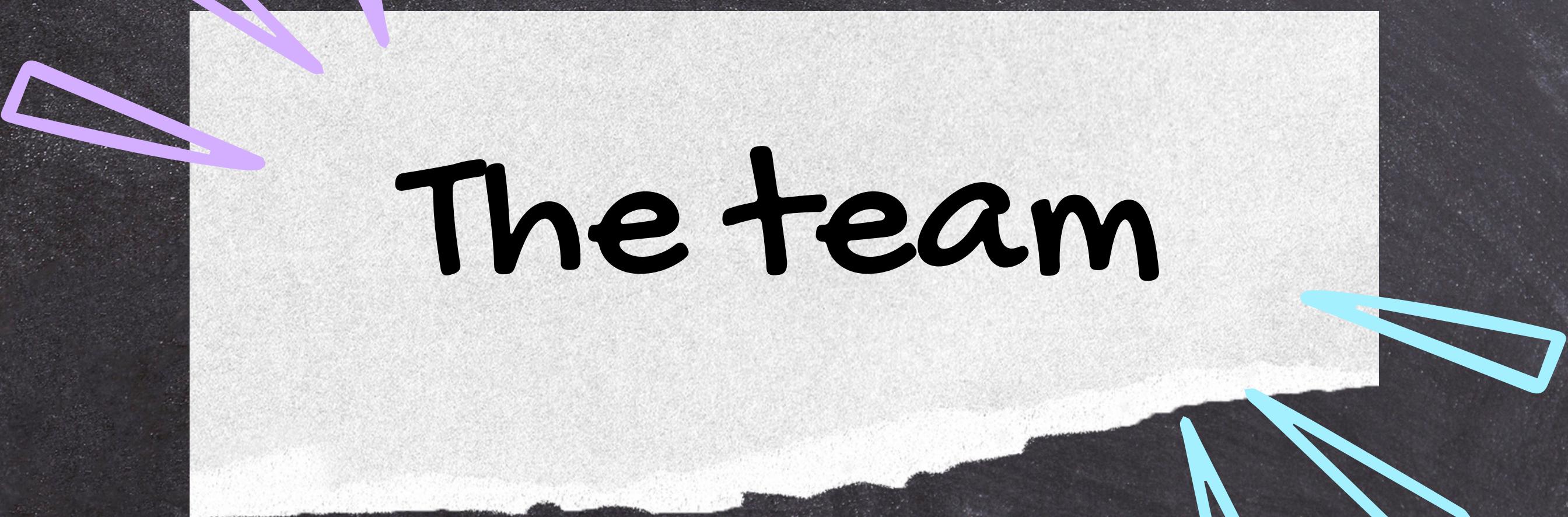


Crafting an interactive game based on the traditional word guessing game, Hangman in Python, by using Tkinter for a polished GUI. Players can immerse themselves in their desired game mode.

The single player mode involves various categories, the player can guess letters within a limited number of attempts to unveil the mystery word.

Enhanced with captivating animations and immersive sound effects, our aim is to provide users with an unparalleled gaming experience.





The team

TEAM ROLES

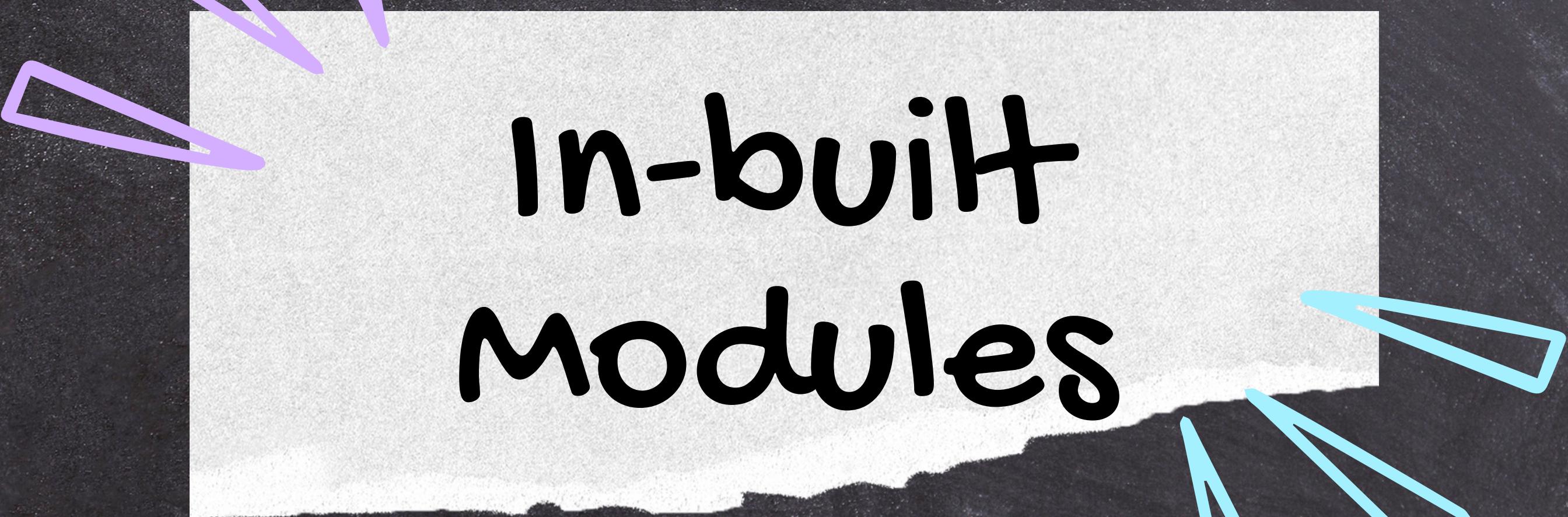
- START SCREEN
- HOME SCREEN
- INSTRUCTIONS SCREEN
- MULTIPLAYER SCREEN
- CATEGORY SCREEN
- GAME SCREEN
- KEYBOARD INTERFACE
- GUI DESIGNING
- DESIGNING COMPONENT
- WORD LIST
- PPT

Shakirth Anisha

- INSTRUCTIONS SCREEN
- WIN AND LOSE SCREEN
- SOUND INSERTION
- ANIMATION INSERTION
- GUI DESIGNING
- WORD LIST
- PPT

Saijyoti Panda





In-built
modules

In Built Modules



1

Start Screen

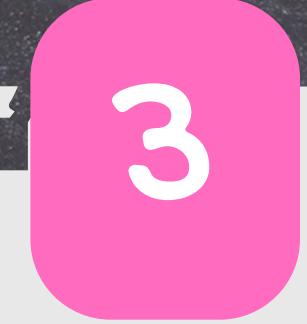
Within the start screen module resides an initiation Start button. When activated by a user click, this button commences the game and seamlessly progresses to the subsequent Home Screen interface.



2

Home Screen

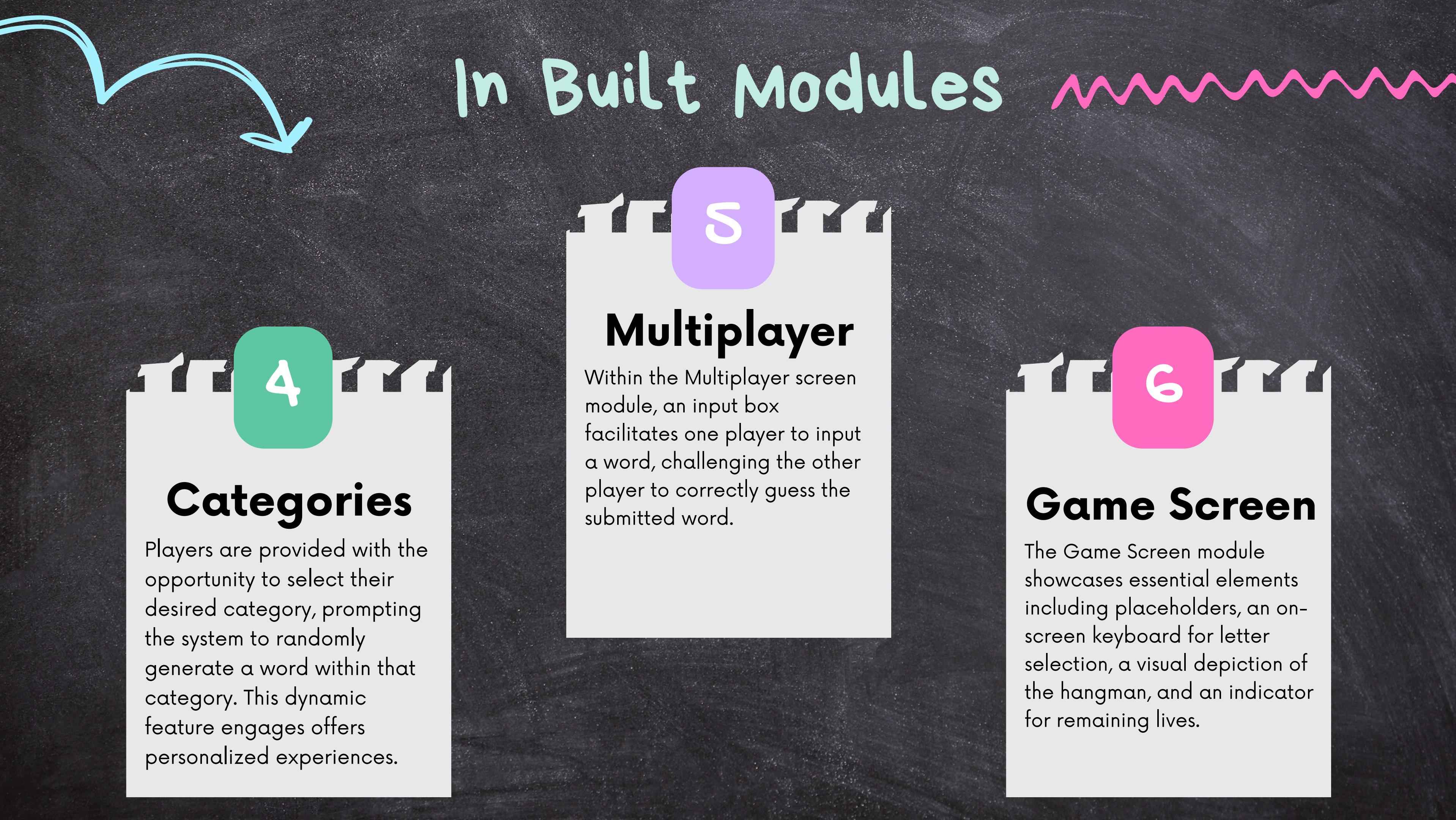
The Home Screen module showcases distinct buttons—Single Player, Multiplayer, and Instructions—that directs the users to their designated screens upon selection. This design ensures a smooth transition for every interaction.



3

Instructions

The Instructions screen module provides comprehensive guidance on how to play. Additionally, the presence of Single Player and Multiplayer buttons within this interface ensures user accessibility and engagement.



Categories

Players are provided with the opportunity to select their desired category, prompting the system to randomly generate a word within that category. This dynamic feature offers personalized experiences.

Multiplayer

Within the Multiplayer screen module, an input box facilitates one player to input a word, challenging the other player to correctly guess the submitted word.

Game Screen

The Game Screen module showcases essential elements including placeholders, an on-screen keyboard for letter selection, a visual depiction of the hangman, and an indicator for remaining lives.

In Built Modules



Keyboard

The on-screen keyboard within the game interface presents functional alphabet keys, ensuring intuitive letter selection. Its integration with the physical keyboard enhances user interaction.



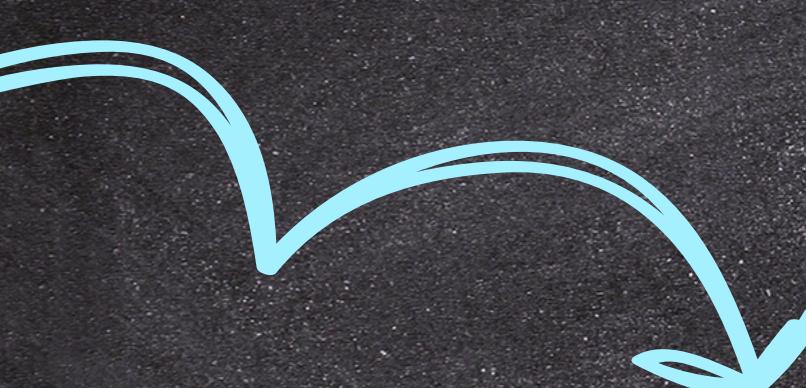
Win Screen

Upon successful guessing of the word, the Win Screen activates, showcasing a congratulatory message along with the revealed word. Additionally, a "Play Again" button is available to navigate back to Home Screen.



Lose Screen

Upon unsuccessful guessing of the word, the Lose Screen activates, the correct word is revealed. Additionally, a "Play Again" button is available to navigate back to Home Screen.



In Built Modules



10

Sound

Sound modules are integrated into the Win and Lose Screens. These audio cues accompany the respective outcomes, providing an engaging audio element that enhances the overall gaming experience.

11

Animation

Animations have been incorporated into our project for the Win and Lose Screens, enriching the user experience with engaging visual elements. These videos accompany the respective outcomes, adding a dynamic dimension to the overall gaming interface.

12

Extra Buttons

The interface includes the Back button, enabling users to navigate to the previous screen for enhanced flexibility. Additionally, a Close button integrated into the end screen which offers players a convenient option to exit the game.

Module Libraries

Tkinter

01

Pygame

03

OS

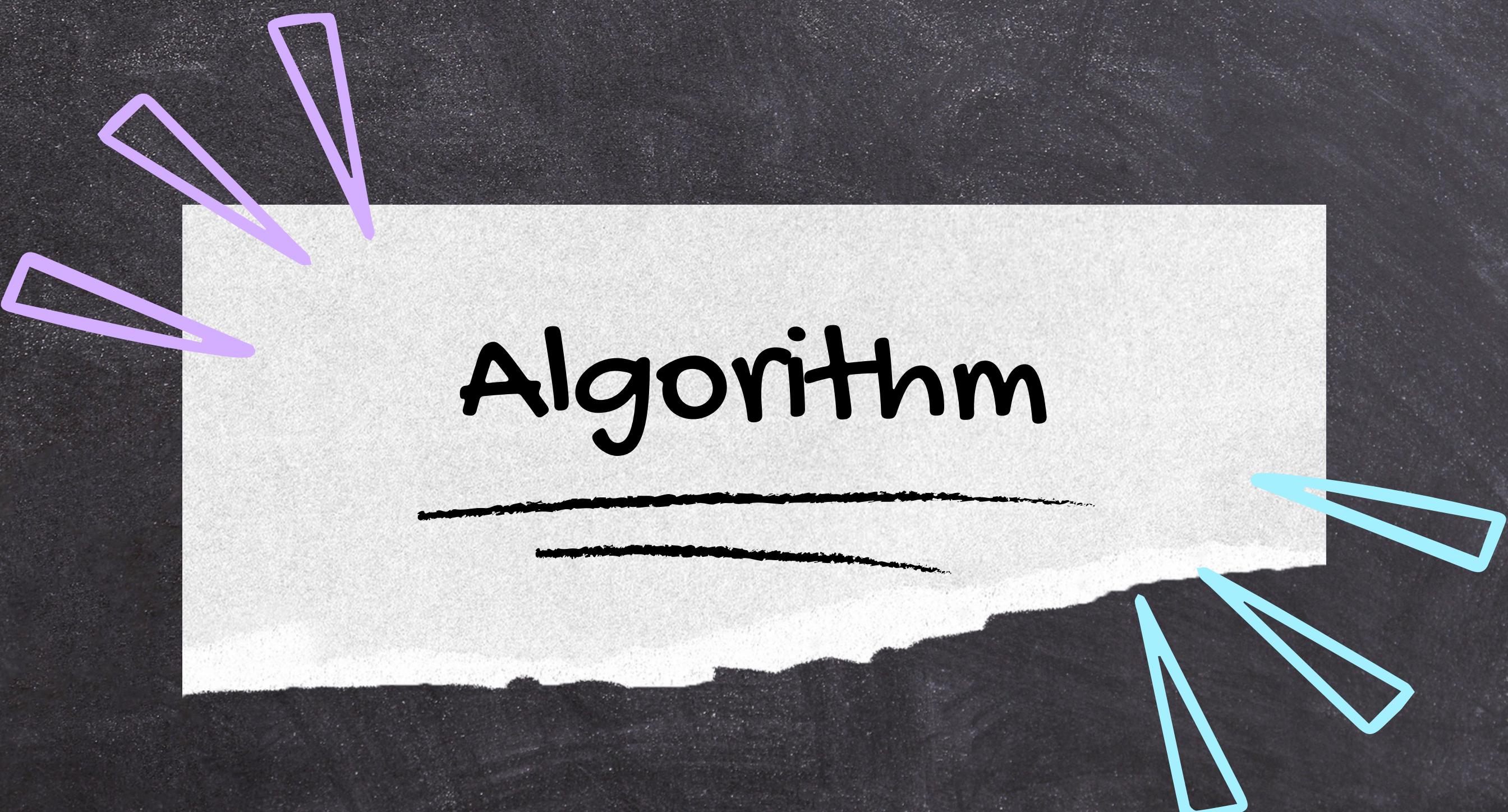
05

Tkvideo

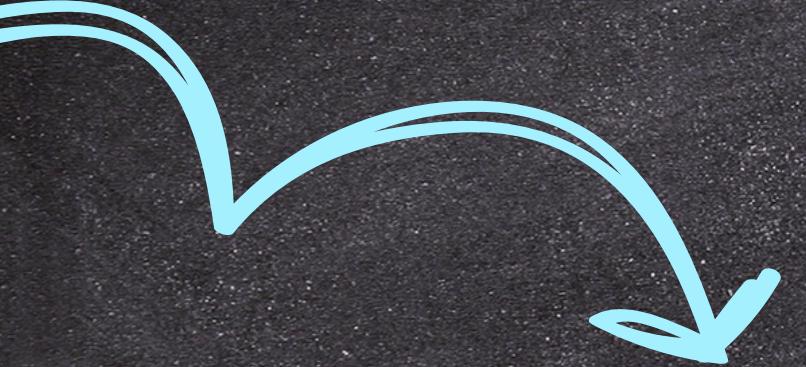
02

Random

04



Algorithm



Algorithm



```
# Import the necessary libraries.
```

```
import tkinter as tk
```

```
import random
```

```
# Define a list of words for the game words.
```

```
# Select a random word from the list.
```

```
chosen_word = random.choice(words)
```

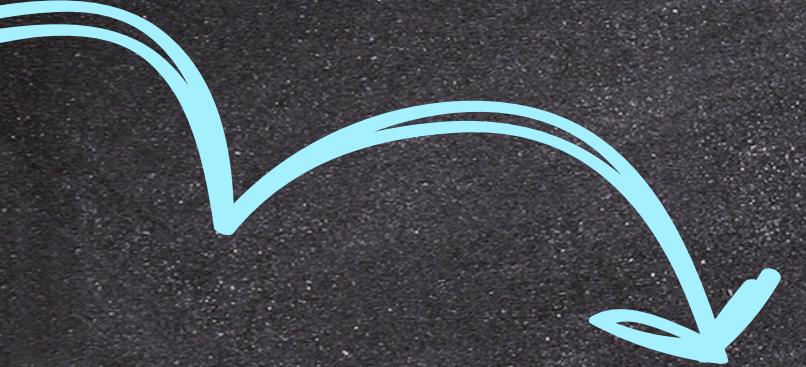
```
word_length = len(chosen_word)
```

```
# Create a Tkinter window
```

```
hangman = tk.Tk()
```

```
hangman.title("Hangman")
```





Algorithm



```
# Let's assume the chosen_word is "COFFEE"  
# Function to update word display  
def replace():  
  
# Update the label to show current progress  
- O _ _ _ -  
  
# Function to handle letter selection  
def letter_click(letter):  
    # Check if the selected letter is in the chosen  
    # word and update the screen accordingly.  
    - O _ _ E E
```



Algorithm

```
# Function to check game status (win/loss)
def check_game_status():
    # Check if the player has guessed the right word
    or not and proceed to the next screen.

# Function to start a new game
def play_again():

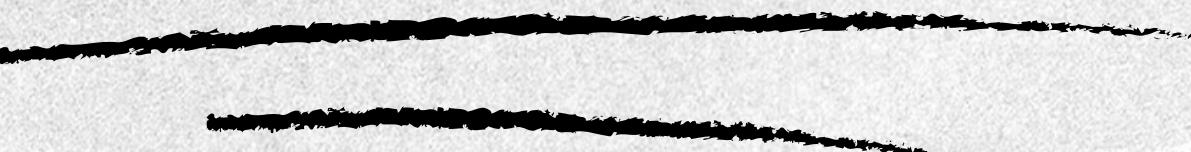
    # Reset all the variables and go to the Home
    Screen.

# Start the main loop
hangman.mainloop()
```



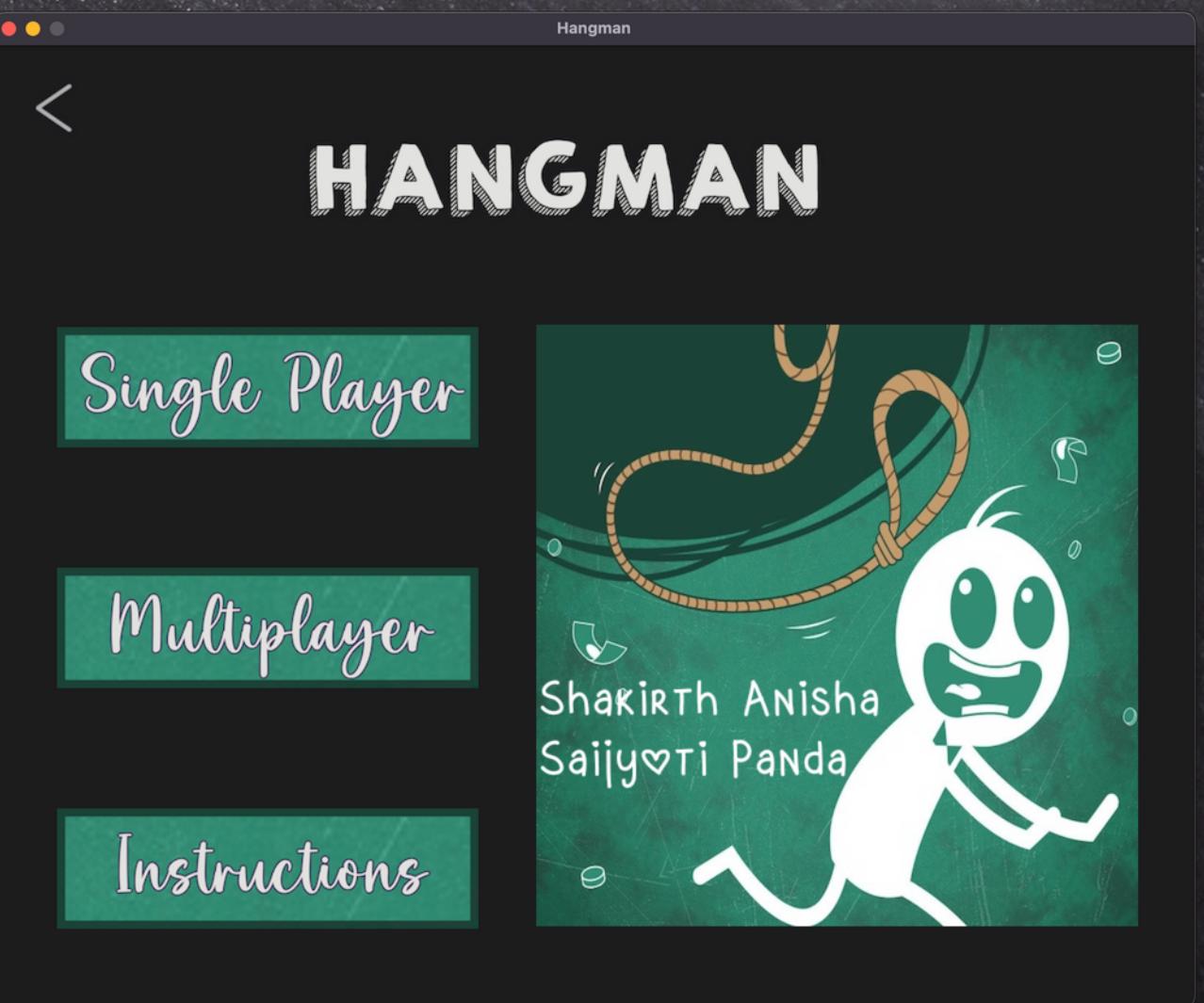


Results





START
SCREEN



<

Categories



SINGLE PLAYER

GAME SCREEN

Lives: XXXX

A B C D E F G H I J
K L M N O P Q R
S T U V W X Y Z

Hangman

o - A - E
- O U

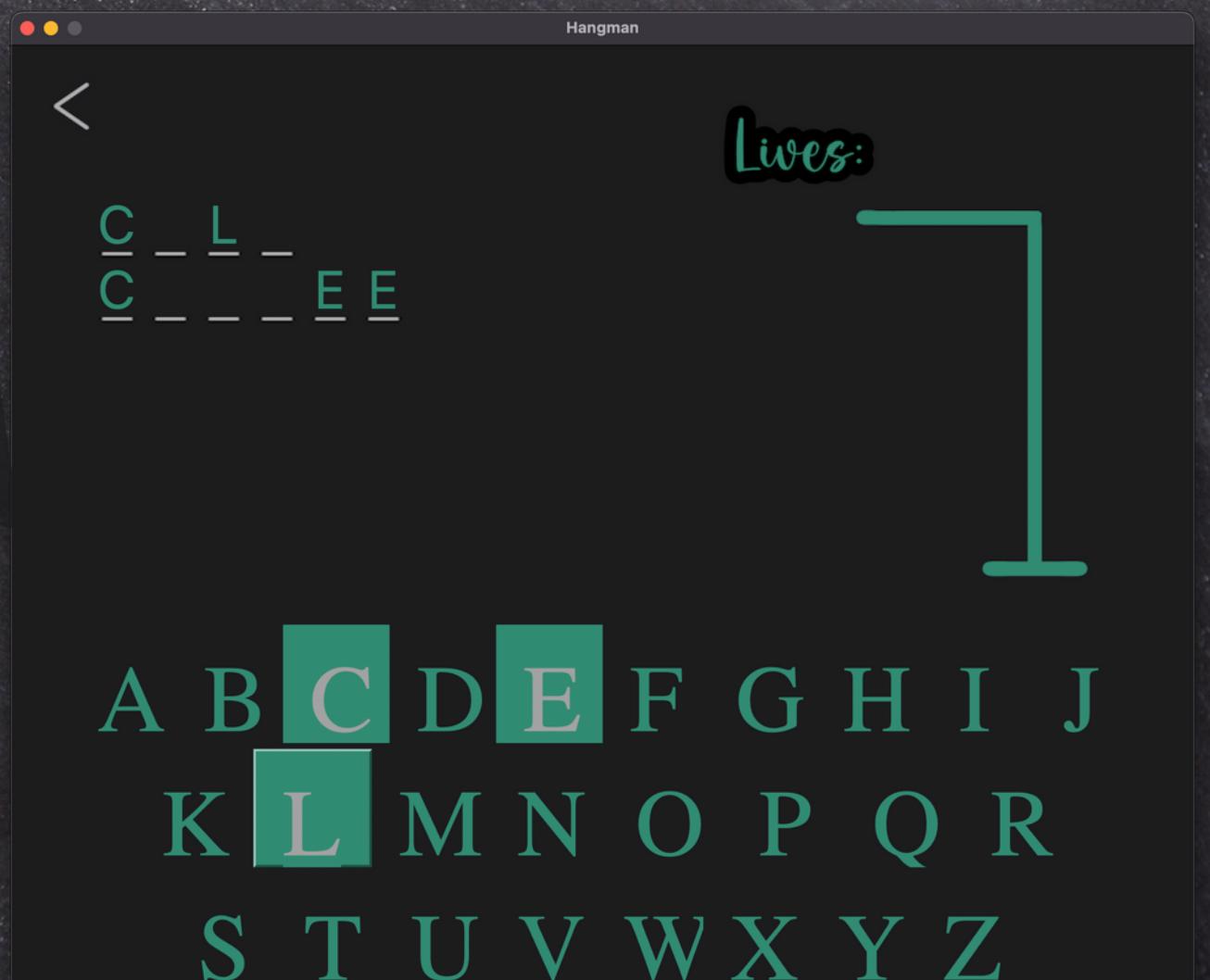
The game screen displays the word 'HANGMAN' partially猜出 (H_A_N_G_M_A_N) with underscores. The letter 'M' is highlighted in green. The player has four lives remaining, indicated by a stick figure hanging from a gallows. A grid of letters A through Z is provided for guesses.



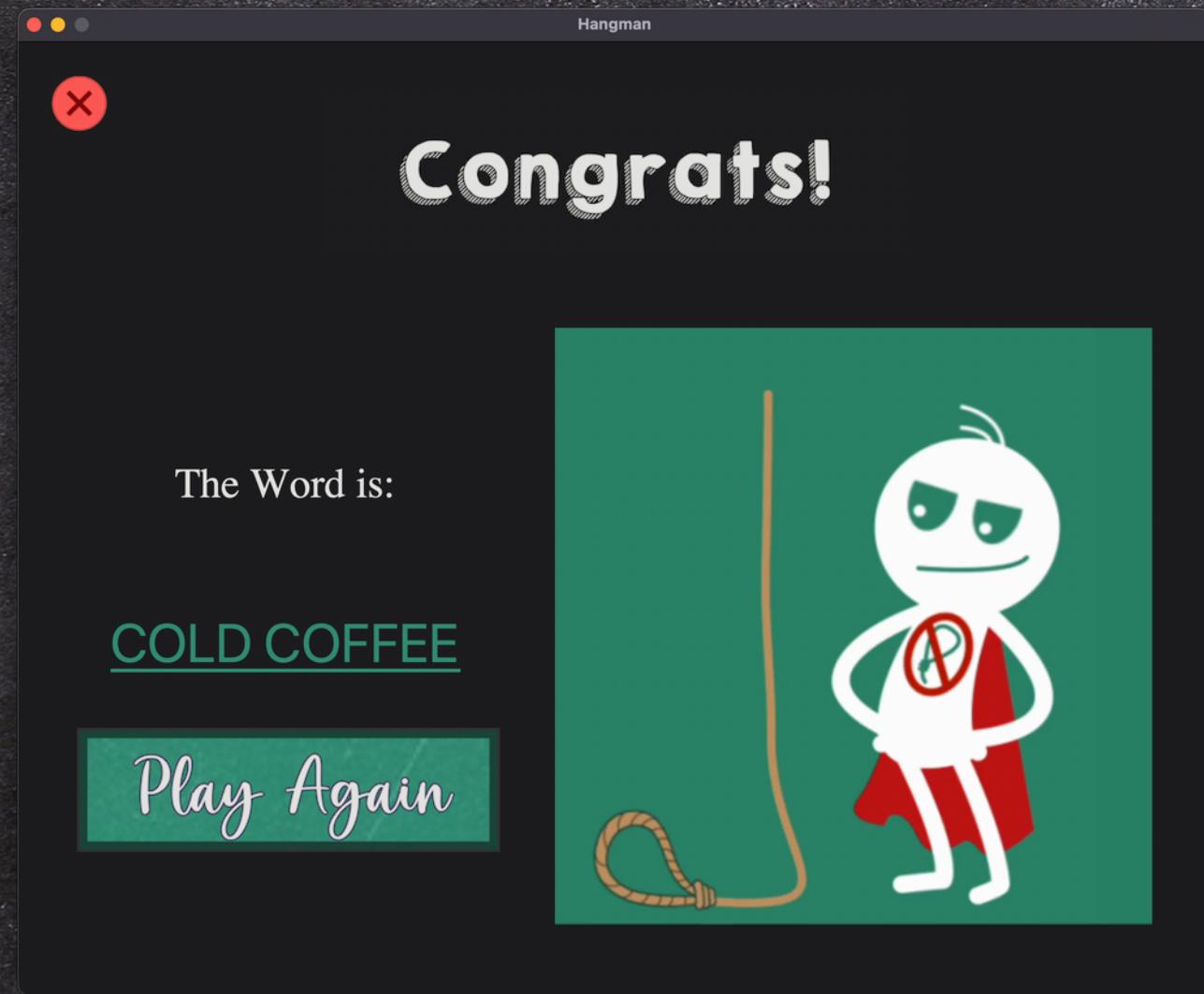
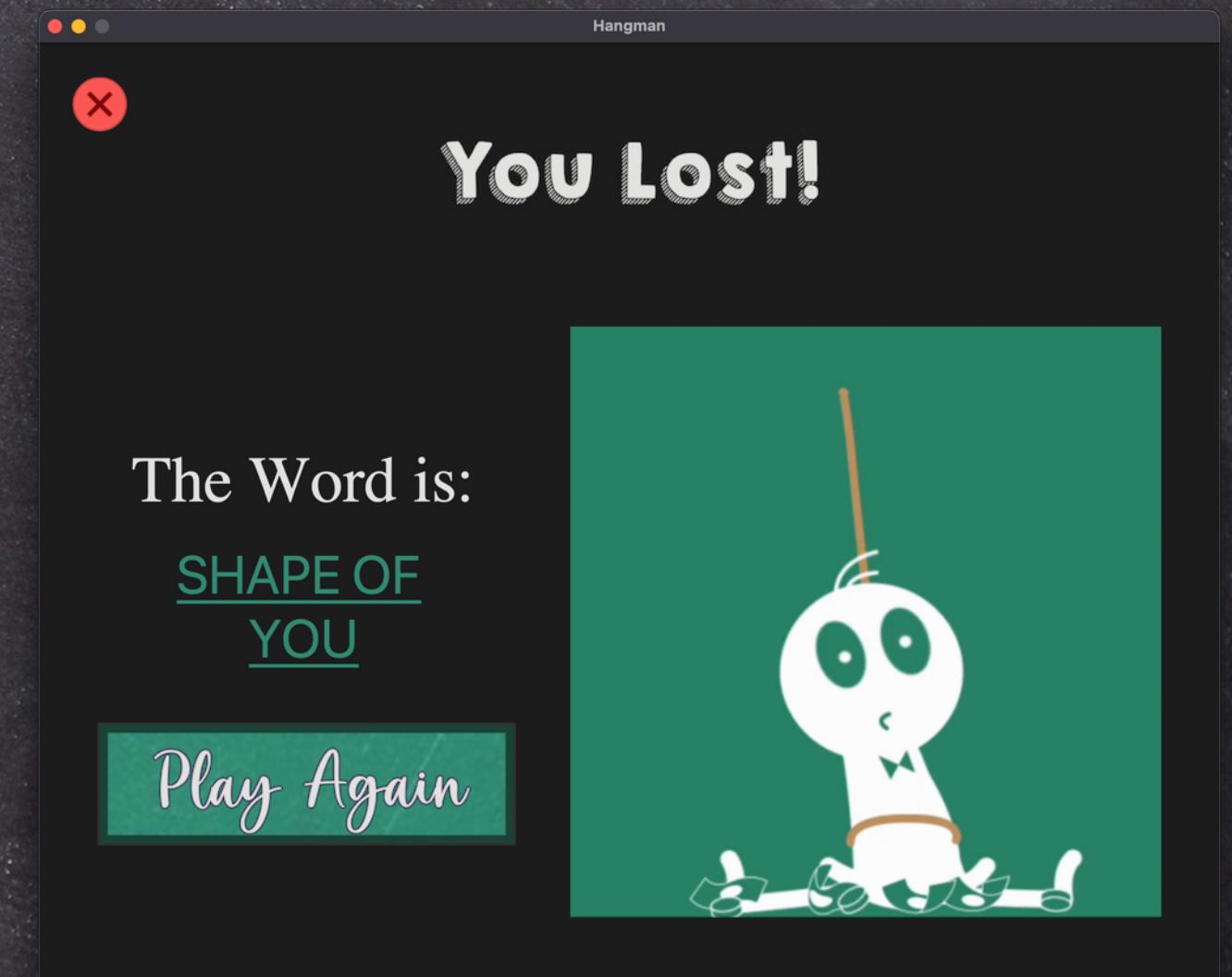
MULTI
PLAYER



GAME SCREEN



LOSE SCREEN



WIN PLAYER



Learning

Learning from the project

01.

Concepts of control statements- selection statement and looping statements are used in many different parts of our code.



03.

Using tkinter and it's widgets to create the GUI for the game.
Concepts of modules have been used to create the project
Importing python libraries.

02.

List, set and string concepts are have been used in the game screen and category screen.
Concepts of creating functions and callback has been used throughout the project.

04.

Concepts of lambda, exceptions like try-except have been used in the game screen

CONCLUSION

Our Hangman game in Python, that was created by utilising the Tkinter interface, presents an innovative twist on word-guessing entertainment.

Featuring diverse play modes like single player and multiplayer, along with an array of categories, complemented by animations and sounds, our goal was to deliver an immersive and personalised gaming adventure.

We aspire to provide a gaming experience that's not only enjoyable but also leaves a lasting impression of fun and excitement!



REFERENCES

- 1 [GEEKS FOR GEEKS](#)
- 2 [DOCUMENTATION](#)
- 3 [REAL PYTHON](#)
- 4 [PESUACADEMY](#)
- 5 [TUTORIALS POINT](#)

THANK
YOU