

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ACKNOWLEDGEMENT	3
	ABSTRACT	4
	LIST OF FIGURES	6
1	INTRODUCTION	
	1.1 Overview	7
	1.2 Technology	7
2	RECOMMENDATION SYSTEM	
	2.1 General Pipeline	9
	2.2 Application Of Word Guesser	10
3	PROPOSED SYSTEM	
	3.1 Flow Diagram	11
	3.2 Proposed Solution	12
4	PROJECT REQUIREMENT	
	4.1 Hardware Requirement	14
	4.2 Software Requirement	14
5	IMPLEMENTATION	
	5.1 Program Code	15
	5.2 Output	22
6	CONCLUSION	26
	REFERENCES	27

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
1.1	Visual Studio	7
1.2	Tkinter	8
2.1	Pipeline	9
2.2	Wordle	10
2.3	Scrabble go	10
3.1	Flow Diagram	11
5.1	Getting Input	22
5.2	Searching in level 0	22
5.3	Searching in level 1	23
5.4	Searching in level 2	23
5.5	Found Character	24
5.6	Main Interface	25
5.7	Closing Interface	25

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

"Word Guesser" is an engaging and intellectually stimulating word-based game that challenges players to decipher a hidden word using clues provided by either their opponent or the game system. The primary objective of "Word Guesser" is for the player to correctly identify a hidden word within a limited number of attempts based on the clues provided. One player, or the game system, selects a word and provides clues, while the other player attempts to guess the word using these clues. Clues can range from synonyms and antonyms to descriptive associations and contextual hints, and the guesser has a set number of attempts to identify the word correctly.

The game features a user-friendly interface designed to enhance the gaming experience. Key elements include input fields where players can easily input their guesses, feedback mechanisms that provide immediate feedback on each guess indicating whether it is correct or incorrect, and a clear display of clues to aid the guessing process. The game incorporates visual and auditory feedback to keep players engaged and motivated, and clear instructions and interactive elements ensure an immersive experience. For example, after each guess, players receive hints such as "Warmer" or "Colder" to indicate their proximity to the correct word, further engaging them in the deduction process.

1.2 TECHNOLOGY

VISUAL STUDIO

- Visual Studio Code is a free, lightweight but powerful source code editor that runs on your desktop and on the web and is available for Windows, macOS, Linux, and Raspberry Pi OS. It comes with built-in support for Javascript, Typescript and node js and has a rich ecosystem of extensions for other programming languages



Figure 1.1₇—Visual Studio

TKINTER

- Tkinter is a standard Python GUI library that provides a set of tools and widgets to create desktop applications with graphical interfaces. Tkinter is included with most Python installations, making it easily accessible for developers who want to build GUI applications without requiring additional installations or libraries.



Figure 1.2 – Tkinter

CHAPTER 2

WORD GUESSING SYSTEM

2.1 GENERAL PIPELINE

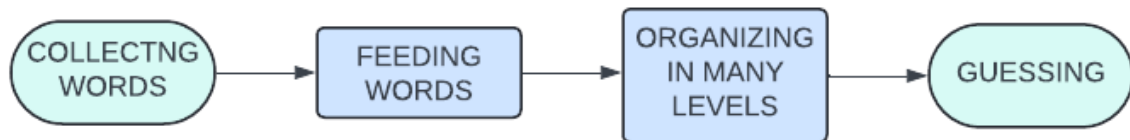


Figure 2.1 - Pipeline

To create a word guesser, start by conceptualizing the game's purpose and identifying the target audience. Design the game mechanics, including the number of attempts, feedback system, and word length, while also crafting the user interface. Collect a diverse word list and categorize it by difficulty. Choose a development platform (web, mobile, or desktop) and select appropriate programming languages and frameworks, such as HTML/CSS/JavaScript for web or Swift/Kotlin for mobile. Implement the game logic and UI, then conduct unit, integration, and user testing. Deploy the application on a suitable platform, set up continuous integration and continuous deployment (CI/CD) pipelines, and finally, launch the game. Gather user feedback and provide regular updates to improve the experience.

2.2 APPLICATION OF WORD GUESSER

WORDLE:

A popular online game where players have six attempts to guess a five-letter word. It provides feedback on which letters are correct and in the correct position or are in the word but in the wrong position.



Figure 2.2-Wordle

SCRABBLE GO:

An online version of the classic Scrabble game that includes various word guessing and word formation challenges against other players. Scrabble is a beloved word game where two to four players compete to score the highest points by creating words on a game board. Each player draws seven letter tiles from a pool and takes turns forming words horizontally or vertically on the 15x15 grid board. Players earn points based on the letters' values and any bonus squares they utilize. The game continues until all letter tiles are used, or players cannot make any more valid words. The player with the highest score at the end wins. Scrabble is not only entertaining but also challenges players' vocabulary and strategic thinking skills.



Figure 2.3 Scrabble Go

CHAPTER 3

PROPOSED SYSTEM

3.1 FLOW DIAGRAM

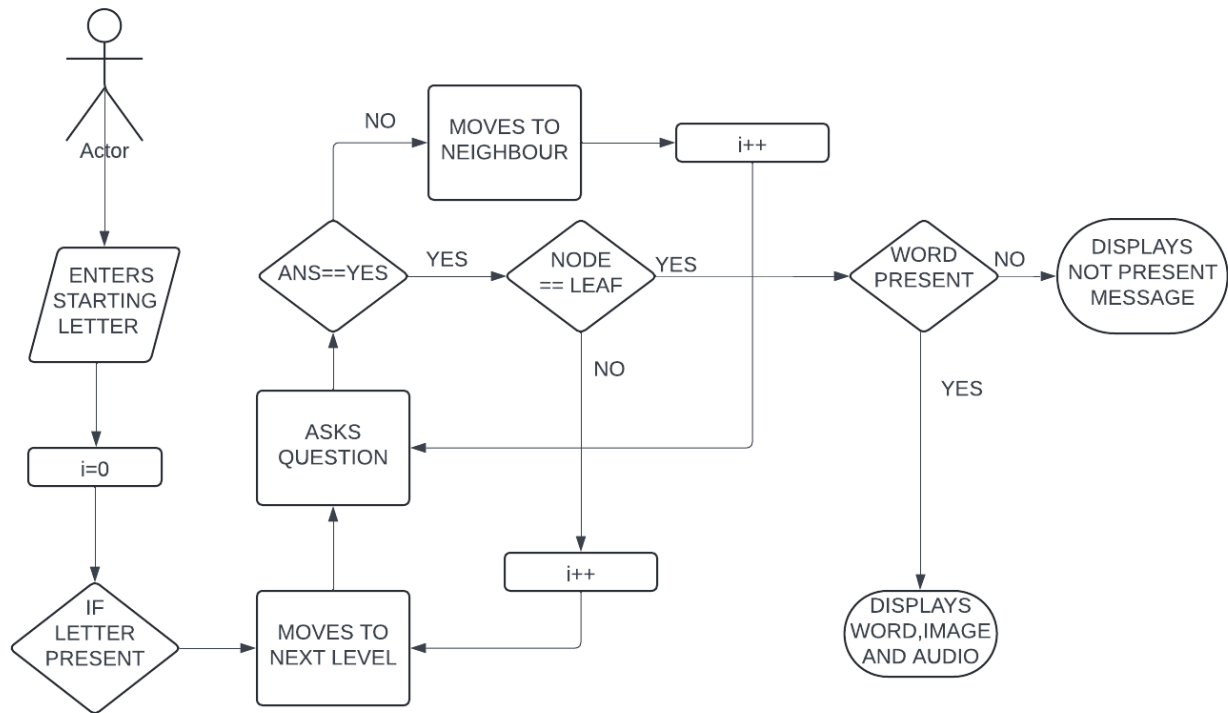


Figure 3.1. Flow Diagram

3.2 PROPOSED SOLUTION

Imported Packages:

1. Tkinter:

- This is the standard GUI (Graphical User Interface) toolkit for Python. It provides a set of tools for creating graphical user interfaces.
- Used for creating windows, labels, buttons, entry fields, and other GUI elements.

2. PIL (Python Imaging Library):

- PIL is a library for adding image processing capabilities to Python.
- Specifically, Image and ImageTk modules are used for loading and displaying images within tkinter windows.

3. pygame.mixer:

- Pygame is a set of Python modules designed for writing video games.
- pygame.mixer is used for handling sound playback. It's initialized with pygame.mixer.init() and used to play audio files.

4. os:

- The os module provides a portable way to use operating system-dependent functionality.
- Although it's not explicitly imported in the code, it's commonly used for file path operations.

5. tkinter.messagebox:

- This submodule of tkinter provides methods to create standard modal dialog boxes.
- Used for displaying informational messages and yes/no prompts.

6. Image:

- An image processing library imported from the PIL package.
- Used for loading and manipulating image files.

7. ImageTk:

- Another module from the PIL package.
- Used to convert PIL images to Tkinter-compatible photo images for display.

Searching technique used:

- The search technique used in word guesser is a simplified form of a Breadth-First Search (BFS) algorithm. BFS is a graph traversal algorithm that explores all the neighbor nodes at the present depth before moving on to the nodes at the next depth level.
- In this context, the search technique is applied to the task of guessing a character based on the user's input (the first letter of the character). Here's how it works:
- **Filtering Characters:** The program starts by filtering the list of characters based on the first letter provided by the user. This step narrows down the search space to only those characters whose names begin with the specified letter.
- **Breadth-First Search:** The program then iteratively dequeues characters from the filtered list and checks if they match the user's input through a series of yes/no questions. This process mimics the breadth-first traversal of a tree or graph, where each character represents a node, and the questions represent edges.
- **Guessing Process:** For each character, the program asks a predefined set of questions to determine if it matches the user's description. If a character is guessed correctly based on the user's responses to the questions, the program displays information about the character and ends the search.
- **Feedback and Iteration:** If the program fails to guess the character correctly, it continues to iterate through the remaining characters in the filtered list, asking questions until either a character is guessed correctly or the list is exhausted.
- Overall, this search technique efficiently narrows down the list of characters based on the user's input and systematically explores each possibility until a match is found or the search space is exhausted. It's a simple yet effective approach for guessing characters in a game-like setting

CHAPTER 4

PROJECT REQUIREMENT

4.1 HARWARE REQUIREMENTS:

- Operation System WINDOWS 7 AND ABOVE
- Any Processor Corei3 and above
- RAM of 512MB+
- Monitor of 14.1 or 15 -17 inch
- Keyboard and Mouse.

4.2 SOFTWARE REQUIREMENTS:

- Windows OS
- Python version 3.0.8 and above
- Visual Studio

CHAPTER 5

IMPLEMENTATION

5.1 PROGRAM CODE

```
import tkinter as tk
from tkinter import messagebox
from PIL import Image, ImageTk
import pygame.mixer

# Initialize the pygame mixer
pygame.mixer.init()

class Character:
    def __init__(self, name, questions, info, image_path, audio_path):
        self.name = name
        self.questions = questions
        self.info = info
        self.image_path = image_path
        self.audio_path = audio_path

def guess_character(characters, first_letter):
    if not first_letter:
        show_info("Akinator", "Please enter a letter.")
        return

    # Convert to lowercase
    first_letter = first_letter.lower()

    # Filter characters based on the first letter
    filtered_characters = [char for char in characters if char.name.lower().startswith(first_letter)]

    if not filtered_characters:
        show_info("Akinator", "No characters found with the specified letter.")
        return

    # Perform BFS
    while filtered_characters:
        # Dequeue the character
        current = filtered_characters.pop(0)

        if guess_ask_question(current):
            # Ask if the guessed character is correct
            answer = ask_yes_no("Akinator", f"I guessed it! Your character is {current.name}. \nIs this correct?")
            if answer:
                show_character_info(current)
```

```

        return

    # If unable to guess the character
    show_info("Akinator", "Sorry, I couldn't guess your character.")

def guess_ask_question(character):
    """
    Function to ask questions about the character.
    """
    for q in character.questions:
        answer = ask_yes_no("Question", q)
        if not answer:
            return False
    return True

def show_character_info(character):
    dialog = tk.Toplevel()
    dialog.title(character.name)
    dialog.geometry("600x400")
    dialog.configure(bg="#FFD700") # Gold

    # Load and display character image
    try:
        image = Image.open(character.image_path)
        photo = ImageTk.PhotoImage(image)
        label = tk.Label(dialog, image=photo)
        label.image = photo # Keep a reference
        label.pack()
    except Exception as e:
        print("Error loading image:", e)

    # Display character information
    label_info = tk.Label(dialog, text=character.info, font=("Arial", 12), bg="#FFD700", fg="#2E8B57")
    # SeaGreen
    label_info.pack(pady=10)

    ok_button = tk.Button(dialog, text="OK", command=lambda: on_ok_button_click(dialog),
font=("Arial", 12, "bold", "italic"), bg="#2E8B57", fg="white") # SeaGreen
    ok_button.pack(pady=5)

    # Play audio
    pygame.mixer.Sound(character.audio_path).play()

    dialog.transient(root)
    dialog.grab_set()
    root.wait_window(dialog)

def on_ok_button_click(dialog):
    # Stop the currently playing music
    pygame.mixer.stop()
    dialog.destroy()

```

```

def show_info(title, message):
    dialog = tk.Toplevel()
    dialog.title(title)
    dialog.geometry("500x300")
    dialog.configure(bg="#FFD700") # Gold
    label = tk.Label(dialog, text=message, font=("Arial", 12), bg="#FFD700", fg="#2E8B57") # SeaGreen
    label.pack(pady=10)
    ok_button = tk.Button(dialog, text="OK", command=dialog.destroy, font=("Arial", 12, "bold",
"italic"), bg="#2E8B57", fg="white") # SeaGreen
    ok_button.pack(pady=5)
    dialog.transient(root)
    dialog.grab_set()
    root.wait_window(dialog)

def ask_yes_no(title, message):
    dialog = tk.Toplevel()
    dialog.title(title)
    dialog.geometry("600x400")
    dialog.configure(bg="#FFD700") # Gold
    label = tk.Label(dialog, text=message, font=("Arial", 12), bg="#FFD700", fg="#2E8B57") # SeaGreen
    label.pack(pady=5)
    yes_button = tk.Button(dialog, text="Yes", command=lambda: set_answer(dialog, True),
font=("Arial", 12, "bold", "italic"), bg="#2E8B57", fg="white") # SeaGreen
    yes_button.pack(side="left", padx=10)
    no_button = tk.Button(dialog, text="No", command=lambda: set_answer(dialog, False), font=("Arial",
12, "bold", "italic"), bg="#2E8B57", fg="white") # SeaGreen
    no_button.pack(side="right", padx=10)
    dialog.transient(root)
    dialog.grab_set()
    root.wait_window(dialog)
    return dialog.answer

def set_answer(dialog, value):
    dialog.answer = value
    dialog.destroy()

def akinator(characters):
    """
    Main function to play the Akinator game.
    """
    global root
    # Create Tkinter window
    root = tk.Tk()
    root.title("Akinator")

    # Get screen width and height
    screen_width = root.winfo_screenwidth()
    screen_height = root.winfo_screenheight()

    # Styling

```

```

root.geometry(f"{screen_width}x{screen_height}")
root.configure(bg="#FFD700") # Gold

# Load and display image
try:
    image = Image.open("C:/Users/Steady/Downloads/aip.gif") # Change path accordingly
    image = image.resize((screen_width, screen_height))
    photo = ImageTk.PhotoImage(image)
    label = tk.Label(root, image=photo)
    label.image = photo # Keep a reference
    label.pack(fill="both", expand=True)

    # Create a frame for buttons and text fields
    frame = tk.Frame(root, bg="#FFD700", bd=5)
    frame.place(relx=0.5, rely=0.2, relwidth=0.6, relheight=0.2, anchor="n")

    # Create GUI elements
    label = tk.Label(frame, text="Welcome to Akinator!", font=("Arial", 18, "bold", "italic"),
bg="#FFD700", fg="#2E8B57") # SeaGreen
    label.pack(pady=10)

    entry_label = tk.Label(frame, text="Enter the first letter of the character: ", font=("Arial", 12, "bold",
"italic"), bg="#FFD700", fg="#2E8B57") # SeaGreen
    entry_label.pack()

    entry = tk.Entry(frame, font=("Arial", 12))
    entry.pack()

    def start_game():
        guess_character(characters, entry.get())

    # Button to start the game
    start_button = tk.Button(frame, text="Start", command=start_game, font=("Arial", 12, "bold",
"italic"), bg="#2E8B57", fg="white") # SeaGreen
    start_button.pack(pady=10)

    except Exception as e:
        print("Error loading image:", e)

    root.mainloop()

# Define characters and their questions
mario = Character("Mario", ["Does your character wear a hat?", "Is your character associated with
Nintendo?"], "Mario is a fictional character in the Mario video game franchise, created by Nintendo.",
"C:/amirdhasuba/sem4/AI/ai.jpg", "C:/amirdhasuba/music/mario.mp3")
mickey_mouse = Character("Mickey Mouse", ["Does your character have big round ears?", "Is your
character associated with Disney?"], "Mickey Mouse is an animated character created by Walt Disney
and Ub Iwerks at The Walt Disney Company.", "C:/Users/Steady/OneDrive/Desktop/aiproj.jpg",
"C:/amirdhasuba/sem4/AI/AIPROJECT/audio\Mickey Mouse Clubhouse Theme Song
disneyjunior.mp3")
spongebob = Character("SpongeBob SquarePants", ["Is your character a sea creature?", "Is your character

```

associated with Nickelodeon?"]], "SpongeBob SquarePants is a fictional character and the titular character of the American animated television series of the same name.",
"C:/amirdhasuba/sem4/AI/AIPROJECT/sponge.jpg", "path/to/spongebob_audio.wav")
superman = Character("Superman", ["Does your character wear a cape?", "Is your character associated with DC Comics?"]], "Superman is a fictional superhero appearing in American comic books published by DC Comics.", "C:/amirdhasuba/sem4/AI/AIPROJECT/superman.jpg", "path/to/superman_audio.wav")
spiderman = Character("Spider-Man", ["Does your character shoot webs?", "Is your character associated with Marvel Comics?"]], "Spider-Man is a fictional superhero created by writer Stan Lee and artist Steve Ditko for Marvel Comics.", "C:/amirdhasuba/sem4/AI/AIPROJECT/spider.jpg",
"C:/amirdhasuba/sem4/AI/AIPROJECT/audio/Spider Man Song Original [Remastered].mp3")
scooby_doo = Character("Scooby-Doo", ["Is your character a talking dog?", "Does your character solve mysteries?"]], "Scooby-Doo is a fictional character and the titular character of the animated television series of the same name.", "C:/amirdhasuba/sem4/AI/AIPROJECT/scooby.jpg",
"C:/amirdhasuba/sem4/AI/AIPROJECT/audio/Scooby Doo Theme Song – Best Coast from Scoob The Album [Official Audio].mp3")
batman = Character("Batman", ["Does your character wear a mask?", "Is your character associated with DC Comics?"]], "Batman is a fictional superhero appearing in American comic books published by DC Comics.", "C:/amirdhasuba/sem4/AI/AIPROJECT/batman.jpg",
"C:/amirdhasuba/sem4/AI/AIPROJECT/audio/Batman Opening and Closing Theme 1966 - 1968 With Snippets.mp3")
ironman = Character("Iron Man", ["Does your character wear an armored suit?", "Is your character associated with Marvel Comics?"]], "Iron Man is a fictional superhero appearing in American comic books published by Marvel Comics.", "C:/amirdhasuba/sem4/AI/AIPROJECT/iron man.jpg",
"C:/amirdhasuba/sem4/AI/AIPROJECT/audio/IRON MAN - Opening.mp3")
homer_simpson = Character("Homer Simpson", ["Is your character associated with The Simpsons?", "Does your character work at a nuclear power plant?"]], "Homer Simpson is a fictional character and one of the main characters of the American animated sitcom The Simpsons.", "C:/amirdhasuba/sem4/AI/AIPROJECT/homer simpson.jpg",
"path/to/homer_simpson_audio.wav")
pikachu = Character("Pikachu", ["Is your character a yellow electric rodent?", "Is your character associated with Pokémon?"]], "Pikachu is a species of Pokémon, fictional creatures that appear in an assortment of media of the Pokémon franchise by Nintendo and Game
Freak.", "C:/amirdhasuba/sem4/AI/AIPROJECT/pikachu.jpg",
"C:/amirdhasuba/sem4/AI/AIPROJECT/audio/Pikachu Song - Pokemon Go Dance Pokemon Song
Remix.mp3")
hello_kitty = Character("Hello Kitty", ["Is your character a white cat with a red bow?", "Is your character associated with Sanrio?"]], "Hello Kitty is a fictional character produced by the Japanese company Sanrio.", "C:/amirdhasuba/sem4/AI/AIPROJECT/hello kitty.jpg",
"C:/amirdhasuba/sem4/AI/AIPROJECT/audio/Hello Kitty Greatest Hits Song Medley.mp3")
scooby_dum = Character("Scooby-Dum", ["Is your character Scooby-Doo's cousin?", "Is your character less intelligent than Scooby-Doo?"]], "Scooby-Dum is a fictional Great Dane created by Hanna-Barbera Productions.", "C:/amirdhasuba/sem4/AI/AIPROJECT/scooby dum.gif",
"C:/amirdhasuba/sem4/AI/AIPROJECT/audio/Scooby Doo Theme Song – Best Coast from Scoob The Album [Official Audio].mp3")
spongegar = Character("SpongeGar", ["Is your character a prehistoric sponge?", "Is your character associated with SpongeBob SquarePants?"]], "SpongeGar is a prehistoric ancestor of SpongeBob SquarePants.", "C:/amirdhasuba/sem4/AI/AIPROJECT/spongegar.gif", "path/to/spongegar_audio.wav")
tiger = Character("Tigger", ["Is your character a striped tiger?", "Is your character associated with Winnie the Pooh?"]], "Tigger is a fictional tiger character originally introduced in the A. A. Milne book The House at Pooh Corner.", "C:/amirdhasuba/sem4/AI/AIPROJECT/tigger.jpg",
"C:/amirdhasuba/sem4/AI/AIPROJECT/audio/The Wonderful Thing About Tiggers Sing Along

Songs.mp3")

popeye = Character("Popeye", ["Does your character love spinach?", "Is your character associated with Popeye the Sailor?"], "Popeye the Sailor is a fictional cartoon character created by Elzie Crisler Segar.", "C:/amirdhasuba/sem4/AI/AIPROJECT/poppeye.jpg",

"C:/amirdhasuba/sem4/AI/AIPROJECT/audio/Popeye The Sailor Man Intro Theme Song.mp3")

daffy_duck = Character("Daffy Duck", ["Is your character associated with Looney Tunes?", "Is your character a black duck?"], "Daffy Duck is an animated cartoon character produced by Warner Bros. Animation studios.", "C:/amirdhasuba/sem4/AI/AIPROJECT/duffy duck.jpg",

"C:/amirdhasuba/sem4/AI/AIPROJECT/audio/Duck Dodgers intro.mp3")

pikmin = Character("Pikmin", ["Is your character small and plant-like?", "Is your character associated with Nintendo?"], "Pikmin are fictional plant-like creatures in the Pikmin video game series created by Shigeru Miyamoto.", "C:/amirdhasuba/sem4/AI/AIPROJECT/pimin.jpg",

"C:/amirdhasuba/sem4/AI/AIPROJECT/audio/Pikmin song cover.mp3")

kermit = Character("Kermit the Frog", ["Is your character associated with The Muppets?", "Is your character green?"], "Kermit the Frog is a Muppet character and the protagonist of the Muppet Show.", "C:/amirdhasuba/sem4/AI/AIPROJECT/kermit.jpg",

"C:/amirdhasuba/sem4/AI/AIPROJECT/audio/Original Theme Song The Muppet Show Disney.mp3")

homer = Character("Homer Simpson", ["Is your character associated with The Simpsons?", "Is your character overweight?"], "Homer Simpson is a fictional character and one of the main characters of the American animated sitcom The Simpsons.", "C:/amirdhasuba/sem4/AI/AIPROJECT/homer simpson.jpg", "path/to/homer_audio.wav")

bugs_bunny = Character("Bugs Bunny", ["Is your character associated with Looney Tunes?", "Is your character a gray rabbit?"], "Bugs Bunny is an animated cartoon character created in 1940 by Leon Schlesinger Productions.", "C:/amirdhasuba/sem4/AI/AIPROJECT/bugs bunny.gif",

"C:/amirdhasuba/sem4/AI/AIPROJECT/audio/Bugs Bunny Theme - This Is It.mp3")

chhota_bheem = Character("Chhota Bheem", ["Is your character a young boy?", "Is your character associated with Pogo TV?"], "Chhota Bheem is an Indian animated comedy adventure television series created by Rajiv Chilaka.", "C:/amirdhasuba/sem4/AI/AIPROJECT/Chhota-Bheem-GIF-Png.gif",

"C:/amirdhasuba/sem4/AI/AIPROJECT/audio/Chotta bheem title song in tamil (1).mp3")

shinchan = Character("Shinchan", ["Is your character a mischievous boy?", "Is your character associated with Japanese anime?"], "Shin Chan is a Japanese manga series written and illustrated by Yoshito Usui.", "C:/amirdhasuba/sem4/AI/AIPROJECT/shinchan.gif",

"C:/amirdhasuba/sem4/AI/AIPROJECT/audio/Shin chan opening theme song in tamilTamil cartoon world.mp3")

chutki = Character("Chutki", ["Is your character a cheerful girl?", "Is your character associated with Chhota Bheem series?"], "Chutki is one of the main characters in the Indian animated television series Chhota Bheem.", "C:/amirdhasuba/sem4/AI/AIPROJECT/chutki.jpg",

"C:/amirdhasuba/sem4/AI/AIPROJECT/audio/Chotta bheem title song in tamil (1).mp3")

raju = Character("Raju", ["Is your character a mischievous young boy?", "Is your character associated with Chhota Bheem series?"], "Raju is one of the main characters in the Indian animated television series Chhota Bheem.", "C:/amirdhasuba/sem4/AI/AIPROJECT/raju.jpg",

"C:/amirdhasuba/sem4/AI/AIPROJECT/audio/Chotta bheem title song in tamil (1).mp3")

kalia = Character("Kalia", ["Is your character a strong and powerful boy?", "Is your character associated with Chhota Bheem series?"], "Kalia is one of the main characters in the Indian animated television series Chhota Bheem.", "C:/amirdhasuba/sem4/AI/AIPROJECT/kalai.jpg",

"C:/amirdhasuba/sem4/AI/AIPROJECT/audio/Chotta bheem title song in tamil (1).mp3")

jaggu = Character("Jaggu", ["Is your character a talking monkey?", "Is your character associated with Chhota Bheem series?"], "Jaggu is one of the main characters in the Indian animated television series Chhota Bheem.", "C:/amirdhasuba/sem4/AI/AIPROJECT/jaggu.jpg",

"C:/amirdhasuba/sem4/AI/AIPROJECT/audio/Chotta bheem title song in tamil (1).mp3")

doraemon = Character("Doraemon", ["Is your character a robot cat?", "Is your character associated with


```
Japanese manga?"]], "Doraemon is a Japanese manga series written and illustrated by Fujiko F.
Fujio.", "C:/amirdhasuba/sem4/AI/AIPROJECT/doreamon.jpg",
"C:/amirdhasuba/sem4/AI/AIPROJECT/audio/Doraemon title song in tamil.mp3")
```

```
# List of characters
```

```
characters = [mario, mickey_mouse, spongebob, superman, spiderman, scooby_doo, batman, ironman,
homer_simpson, pikachu,
```

```
hello_kitty, scooby_dum, spongegar, tigger, popeye, daffy_duck, pikmin, kermit, homer,
bugs_bunny,
```

```
chhota_bheem, shinchan, chutki, raju, kalia, jaggu, doraemon] # Add more characters here
```

```
# Start the Akinator game
```

```
akinator(characters)
```

```
# List of characters
```

```
# Add more characters here
```

```
# Start the Akinator game
```

```
akinator(characters)
```

5.2 OUTPUT

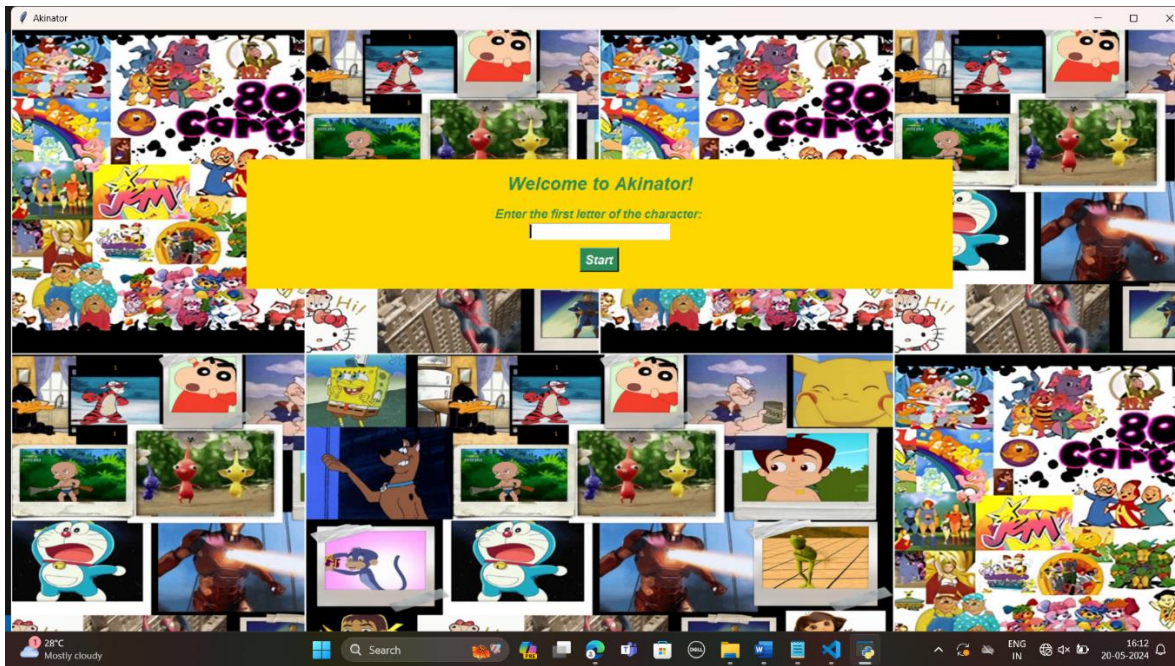


Figure 5.1 – Getting Input

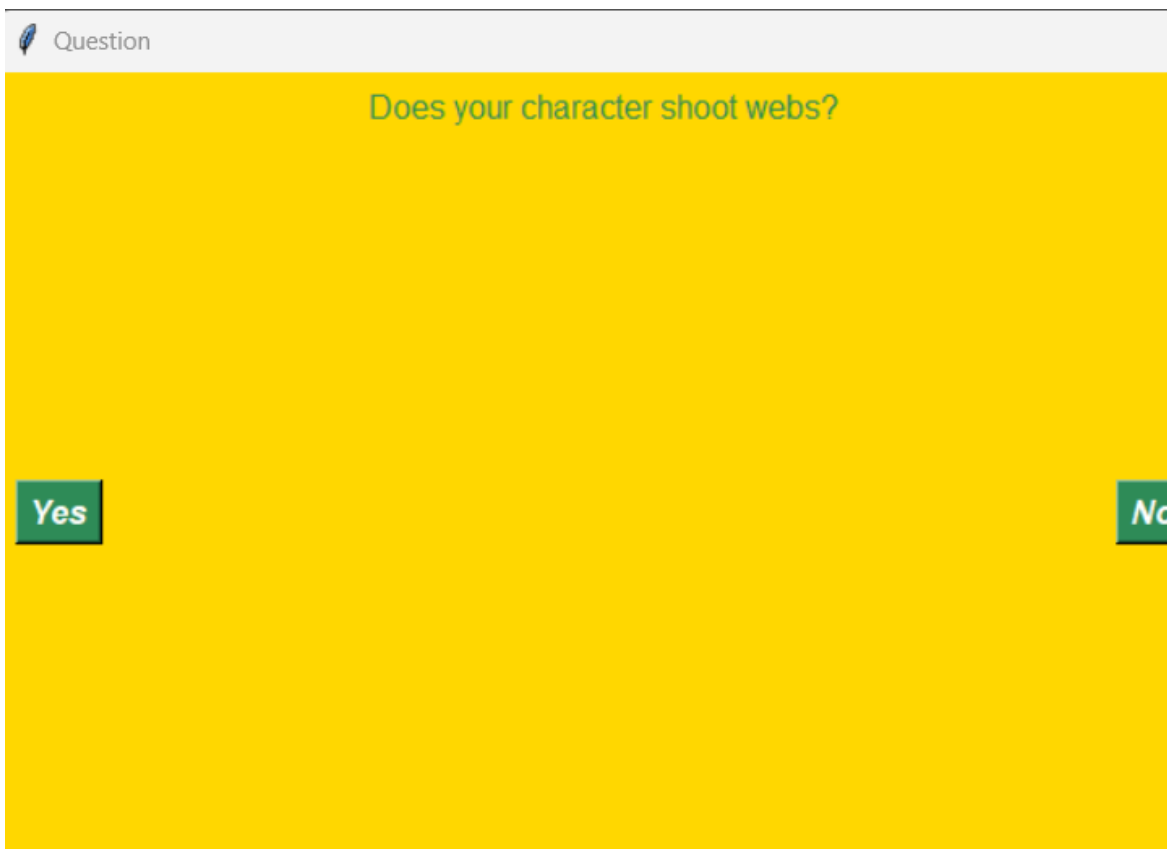


Figure 5.2 – Searching in level 0

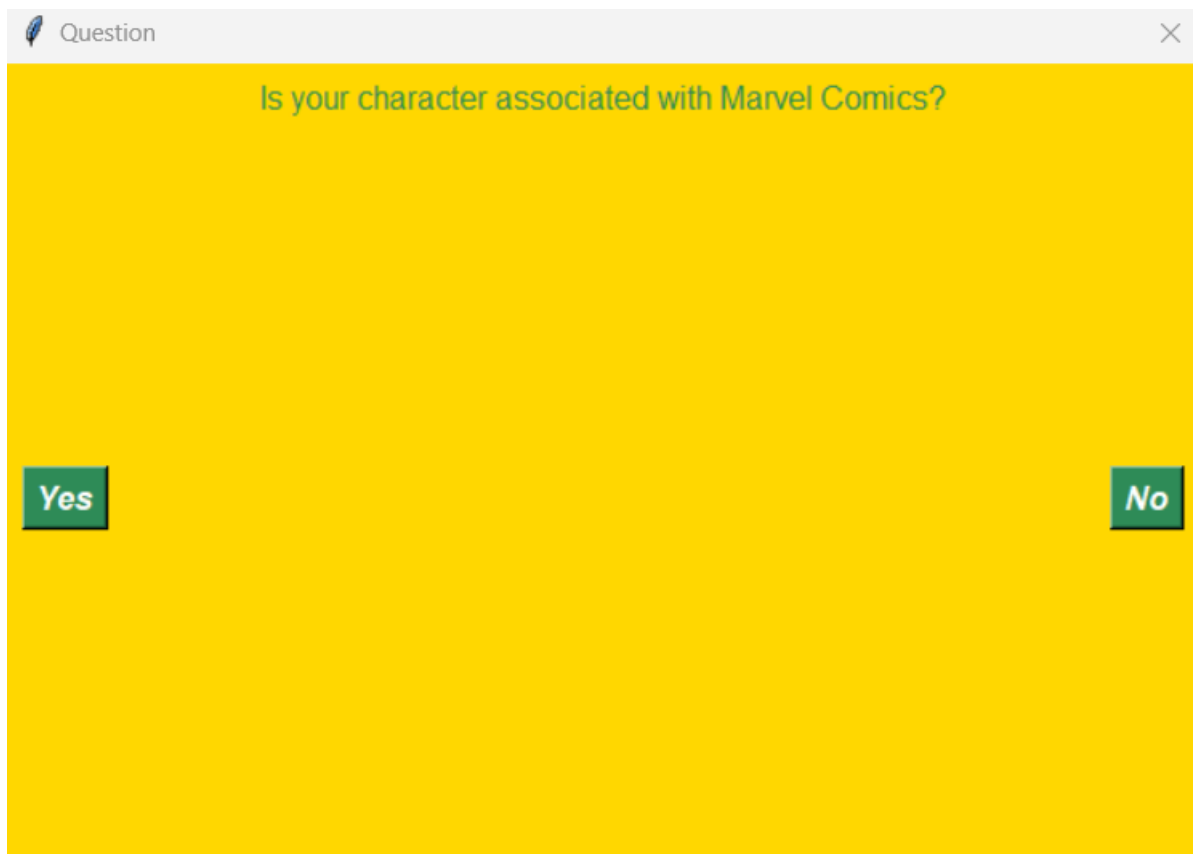


Figure 5.3 – Searching in level 1

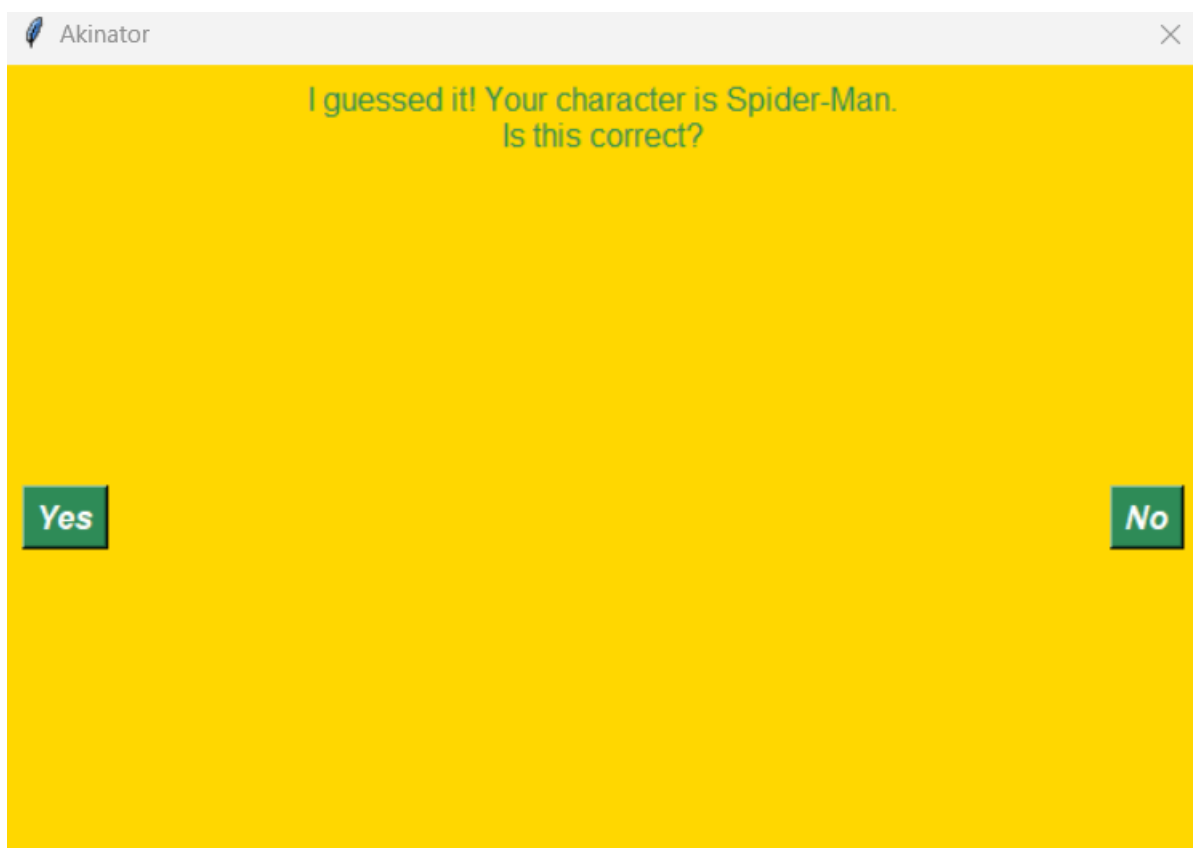


Figure 5.4 – Searching in level 2

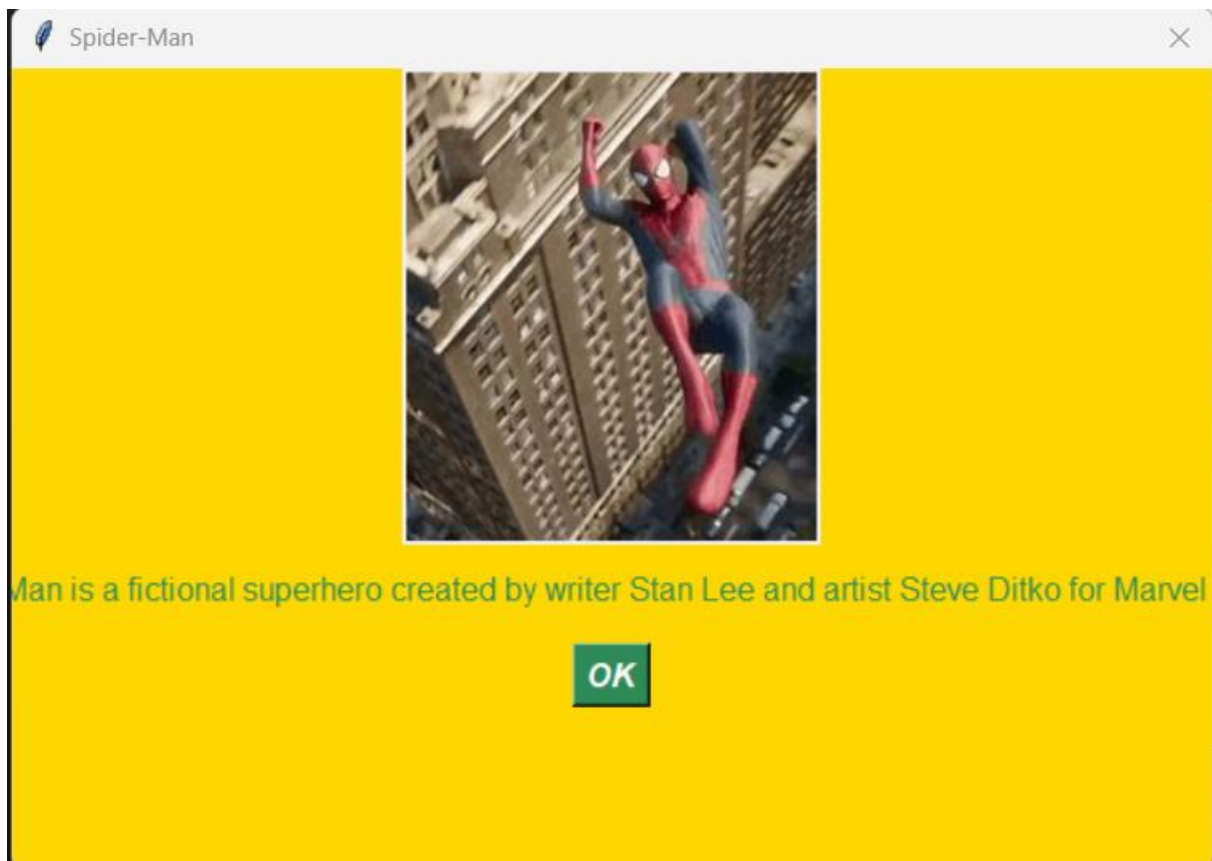


Figure 5.5 – Found Character

USER INTERFACE

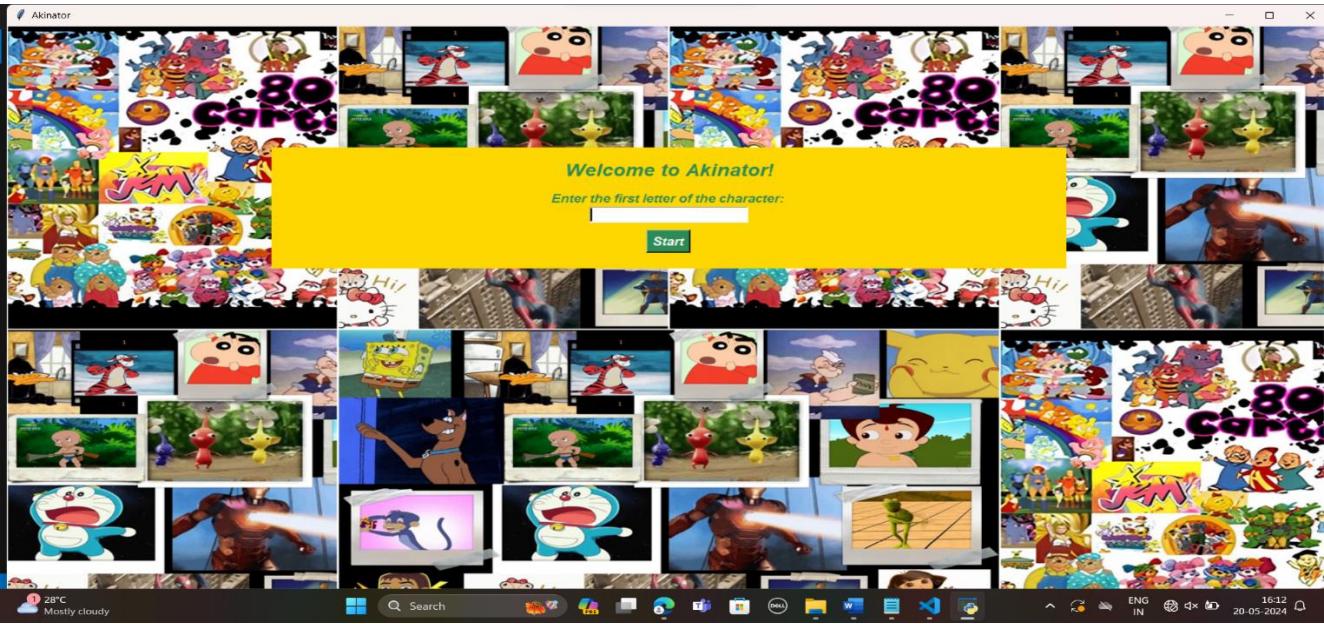


Figure 5.6 -Main Interface

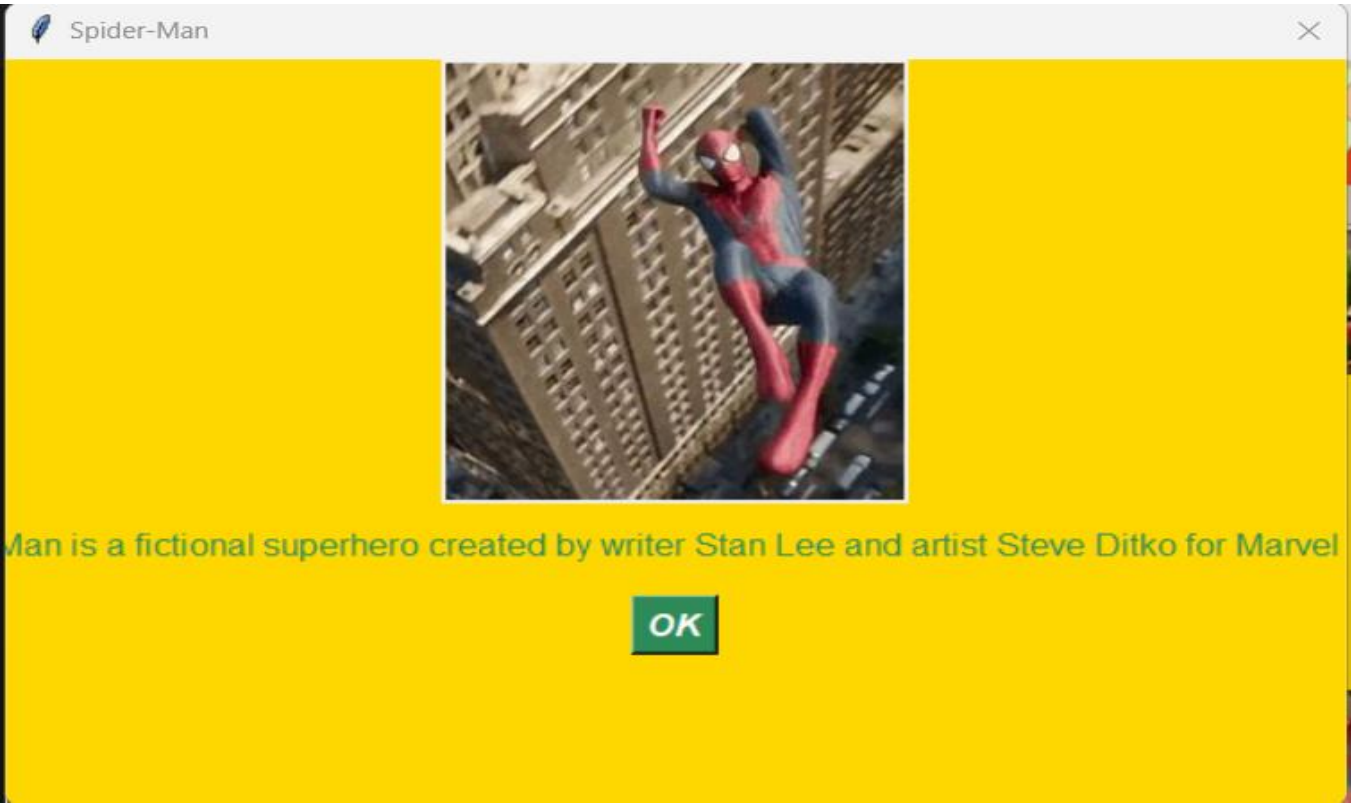


Figure 5.7 – Closing Interface

CHAPTER 6

CONCLUSION

The Word Guesser project successfully demonstrated the development and implementation of an engaging and educational game using programming principles. Through this project, we achieved several key objectives: enhancing programming skills through practical application, including string manipulation, control structures, and user input handling; reinforcing fundamental programming concepts by implementing the game logic; and creating an intuitive, user-friendly interface that ensured players could easily interact with the game. Additionally, feedback mechanisms, such as indicating correct and incorrect guesses, enhanced the overall user experience.

The project also involved solving various challenges, such as handling edge cases and optimizing the guessing algorithm, which honed problem-solving and critical thinking skills. The iterative development process, including testing and debugging, further solidified these skills. Moreover, the game provided an enjoyable way to improve vocabulary and spelling skills, serving as an effective tool for learning and reinforcing language concepts through interactive play.

Overall, the Word Guesser project not only achieved its primary goal of creating a functional and entertaining game but also provided valuable insights into software development practices. The successful completion of this project underscores the importance of combining creativity with technical skills to develop innovative solutions. Future enhancements could include expanding the word database, adding difficulty levels, and incorporating multiplayer functionality to further enrich the gameplay experience.

REFERENCES

- 1) Tkinter GUI Application Development Blueprints by Bhaskar Chaudhary
- 2) Making Games with Python & Pygame by Al Sweigart John Riedl.
- 3) Python Game Programming By Example by Alejandro Rodas de Paz and Joseph Howse.
- 4) Python GUI Programming Cookbook: Develop functional and responsive user interfaces with tkinter and PyQt5, 3rd Edition by Burkhard A. Meier
- 5) Pygame documentation:
<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwj0xdnGypyGAxWjwTgGHYgqA0QQFnoECAYQAQ&url=https%3A%2F%2Fwww.pygame.org%2Fdocs%2F&usg=AOvVaw01eKMOV8jFlEssbvMRH48sm&opi=89978449>
- 6) <https://www.reddit.com/r/learnpython/?rdt=43957>
- 7) <https://www.geeksforgeeks.org/python-program-for-word-guessing-game/>