Final Project Report (CSM216)

On

"Wordle – A Daily Word Game"

Submitted By:

Meghana E

12303998

Submitted on:

22nd November 2024.

"School of Computer Science and Engineering"

Lovely Professional University

Phagwara, Punjab.

Acknowledgement

The successful completion of this project would not have been possible without the invaluable support, guidance, and encouragement of several individuals. I would like to express my sincere gratitude to each of them for their contributions, assistance, and insights, which were instrumental in realizing the Family Tree Management System.

First and foremost, I extend my heartfelt appreciation to my project advisor, Aman Kumar, whose expertise and patient guidance were indispensable throughout this journey. Their commitment to nurturing a thorough understanding of the subject matter provided a solid foundation upon which this project was built. Their constructive feedback, comprehensive review, and insightful suggestions were pivotal in refining this system and addressing challenges effectively.

A special note of gratitude goes to my colleagues and classmates who offered their encouragement and insights, engaging in fruitful discussions that often-sparked new ideas and improvements to the system. Their camaraderie and support fostered a collaborative atmosphere that enriched the development process.

Finally, I am immensely grateful to my family and friends for their understanding, patience, and encouragement throughout this endeavour. Their unwavering support has been a source of motivation and strength, allowing me to focus fully on the successful completion of this project.

To all those mentioned above and many others who contributed indirectly, I extend my deepest thanks. This project has been an enriching learning experience, made possible by your guidance and support.

Table of Contents

Introduction

The Wordle project is an engaging and interactive Python-based word puzzle game inspired by the popular online game. Its purpose is to create an immersive and user-friendly experience where players can guess a daily five-letter word within a limited number of attempts. This project blends logic, creativity, and programming skills to provide an entertaining yet intellectually stimulating game.

The project's primary objective is to simulate a Wordle-like environment, allowing users to enjoy a personalized version of the game in an offline setting. This implementation provides an opportunity to understand essential programming concepts such as graphical user interface (GUI) development, event handling, and logic-building in Python.

Additionally, the project emphasizes sound integration for a more engaging user experience and highlights the utility of features such as hints, feedback mechanisms, and visual indicators to enhance usability. It serves as a creative way to learn and apply programming concepts while also offering an enjoyable product for end users.

This project showcases programming versatility, from managing graphical elements and event-driven logic to creating an enjoyable user experience. It serves as a practical application for developing and understanding software design patterns, GUI handling, and game mechanics. A new "Word of the Day" generated using the current date ensures replay ability and keeps the game fresh. The game includes real-time feedback with color-coded responses to indicate correct, misplaced, and incorrect letters. The implementation offers a clean and interactive interface, leveraging tkinter for GUI development. Features like a hint system, sound effects, and a virtual keyboard make the game more engaging.

Objectives and Scope of the Project

Project Objectives:

1. **Create an Engaging Word Puzzle Game:** Developed a Wordle-inspired game where players guess a daily five-letter word within six attempts.

2. **Provide an Intuitive and Fun User Interface**: Design a user-friendly graphical interface using Python's tkinter, making the game accessible and easy to play.

3. **Data Offer Feedback and Guidance**: Include a feedback system with color-coded hints (green, yellow, and grey) to help players refine their guesses.

4. **Introduce Replay ability with Daily Words:** Implement a feature that generates a new "Word of the Day" daily using a hash-based system.

5. **Enhance User Experience with Sound Effects**: Add audio feedback, including congratulatory sounds, to make the game more enjoyable.

6. **Promote Logical Thinking and Entertainment**: Provide an interactive game that challenges players' problem-solving skills in a fun way.

Scope of the Project:

The scope of the project defines the functionalities and features that the Wordle game will provide to create an enjoyable and interactive experience for users. Here's a more detailed overview:

1. **Core Game Functionality**:

- Players enter guesses in a grid and receive feedback for each letter.
- The game provides six attempts to guess the word correctly.

2. **Graphical User Interface (GUI)**:

- A visually appealing interface with an input grid, buttons, and a virtual keyboard.
- Highlight active rows and provide clear messages (e.g., hints or game results).

3. **Sound Integration**:

- Play a congratulatory sound when players win.
- Include keypress sound effects to make interactions engaging.

4. **Hint System**:

- Players can use a one-time hint to reveal the first letter of the word.

5. **Input Validation**:

- Restrict inputs to valid letters and ensure guesses match the word length.
- Display appropriate warnings for incomplete or invalid guesses.

6. **Daily Word Feature**:

- Use the current date to generate a unique daily word from a fixed list.

By achieving these goals, the project delivers an interactive and rewarding experience for users while showcasing programming capabilities.

Application Tools

1. **Programming Language**

- **Python**:
  The primary programming language used to develop the Wordle game, chosen for its simplicity and versatility in GUI development and game logic.

2. **Integrated Development Environment (IDE)**

- **PyCharm**:
  For structured project management, debugging, and code completion features.
- **Visual Studio Code**:
  Lightweight and versatile, often used for writing and testing the code.
- **Jupyter Notebook** *(Optional)*:
  Used for testing specific functions or experimenting with code snippets during development.

3. **Python Libraries and Packages**

- **tkinter**:
  For creating the graphical user interface (GUI), including buttons, text fields, and visual feedback.
- **hashlib**:
  For generating a daily unique "Word of the Day" using date-based hashing.
- **datetime**:
  To fetch the current date for daily word generation.
- **pygame**:
  For integrating and playing sound effects (e.g., congratulatory fanfare).
- **winsound**:
  To provide instant audio feedback for user interactions (e.g., keypress sounds).

4. **Version Control**

- **Git**:
  Used for version control to track changes in the project, collaborate effectively, and maintain a clean project history.
- **GitHub** *(if applicable)*:
  For storing the project's code repository, ensuring backups, and sharing with collaborators or showcasing publicly.

5. **Additional Tools**

- **Text Editor**:
  Notepad++ or Sublime Text for quick edits and viewing text-based resources, like the word list.
- **Audio Files**:
  Custom or royalty-free audio files (e.g., "tada-fanfare-a-6313.mp3") for sound effects, sourced from trusted platforms like Freesound.org.
- **Image Resources**:
  Any icons or images used for buttons or the GUI, sourced from design tools like Canva or Adobe Illustrator.

6. **Operating System**

- **Windows OS**:
  The game is primarily designed and tested on Windows due to dependencies like winsound.

7. **Testing Tools**

- **Python Debugger (pdb)**:
  For debugging logic errors and ensuring smooth gameplay functionality.
- **Unit Tests** *(Optional)*:
  For testing individual functions like word validation and feedback generation.

8. **Documentation and Learning Resources**

- **Python Official Documentation**:
  Used to understand library functionalities and syntax.

- **Online Tutorials and Forums**:
  Platforms like Stack Overflow, W3Schools, and GeeksforGeeks were helpful for troubleshooting and learning new concepts.

  By leveraging these tools and resources, the project combines robust functionality with a seamless user experience.

Project Design

The project is designed in a modular way, with clear separation of responsibilities between various functions and components. Below is a detailed description of the structure:

1. Main Components

1. **Launcher Window (root)**:
   - The main application window that provides an entry point to the game.
   - Includes a "Begin Wordle" button to launch the game.
2. **Game Window (game_window)**:
   - The primary interface where players interact with the Wordle game.
   - Contains the grid for guesses, virtual keyboard, feedback messages, and other interactive elements.
3. **Word of the Day System**:
   - Generates a unique daily word using a hash-based algorithm and a fixed list of words.
   - Ensures replay ability by changing the word daily.
4. **Feedback System**:
   - Evaluates player guesses against the target word and provides feedback (green, yellow, grey) based on the correctness of each letter.
   - Updates the input grid and virtual keyboard colours dynamically.
5. **Sound System**:
   - Plays sounds for keypresses and congratulatory messages to enhance user engagement.

---

2. Functions and Their Roles

a. Word Generation and Validation

- **get_daily_word ()**:
    - Uses the current date to generate a daily word based on a hash value.
    - Ensures that the same word is used for all players on a given day.

- **get_word_from_grid(row)**:
    - Combines user inputs from a specific row to form the guessed word.

b. Input Handling and Feedback

- **limit_input (var, row, col)**:
    - Restricts input to one uppercase letter per box.
    - Automatically moves the focus to the next input box.
- **handle_backspace ()**:
    - Deletes the last entered character in the active row and adjusts the focus.
- **check_word ()**:
    - Validates the guessed word for length and character restrictions.
    - Compares the guessed word to the target word and provides feedback.
    - Handles winning, losing, or progressing to the next attempt.
- **color_grid (row, feedback)**:
    - Updates the grid's cell colours based on the feedback for each letter.
- **update_keyboard_colors (letter_feedback)**:
    - Updates the virtual keyboard button colours dynamically based on feedback for each letter.

c. User Experience Features

- **highlight_active_row ()**:
    - Highlights the current row in the grid to guide players.
    - Resets previous rows to their default state.
- **show_hint ()**:

- Displays a one-time hint (the first letter of the word) to assist players.

d. Sound Functions

- **play_key_sound ()**:
  - Plays a short beep sound when a key is pressed.
- **play_congratulatory_sound ()**:
  - Plays a celebratory sound when the player wins the game.

---

3. GUI Components

a. Input Grid

- A 6x5 grid where players enter their guesses.
- Each cell is represented by a tk.Entry widget linked to a tk.StringVar.
- Dynamically updated with feedback (e.g., green, yellow, grey) after each guess.

b. Virtual Keyboard

- Composed of buttons for all alphabet letters, divided into three rows (QWERTY layout).
- Button colours update based on feedback, and players can interact using the virtual or physical keyboard.

c. Feedback Label

- Displays messages (e.g., congratulations, warnings, game over) to guide players during the game.

d. Control Buttons

- **Enter**: Submits the current guess.
- **Backspace**: Deletes the last input in the current row.
- **Hint**: Provides a one-time hint for the current word.

---

4. Interaction Between Components

1. **Game Flow**:
   - The launcher starts the game by opening the game window.
   - Players input guesses through the grid or virtual keyboard.

- Feedback is displayed, and the game progresses based on the results of each guess.

2. **Feedback and Visual Updates**:
    - Each guess triggers the feedback mechanism to evaluate correctness.
    - The input grid and virtual keyboard colours are updated based on the feedback.
3. **Daily Word System**:
    - The daily word is retrieved at the start of the game and remains consistent for all players on that day.
    - Provides replay ability with a new word each day.
4. **Sound and Hints**:
    - Audio feedback is triggered for keypresses and wins, while the hint system supports players struggling with the word.

---

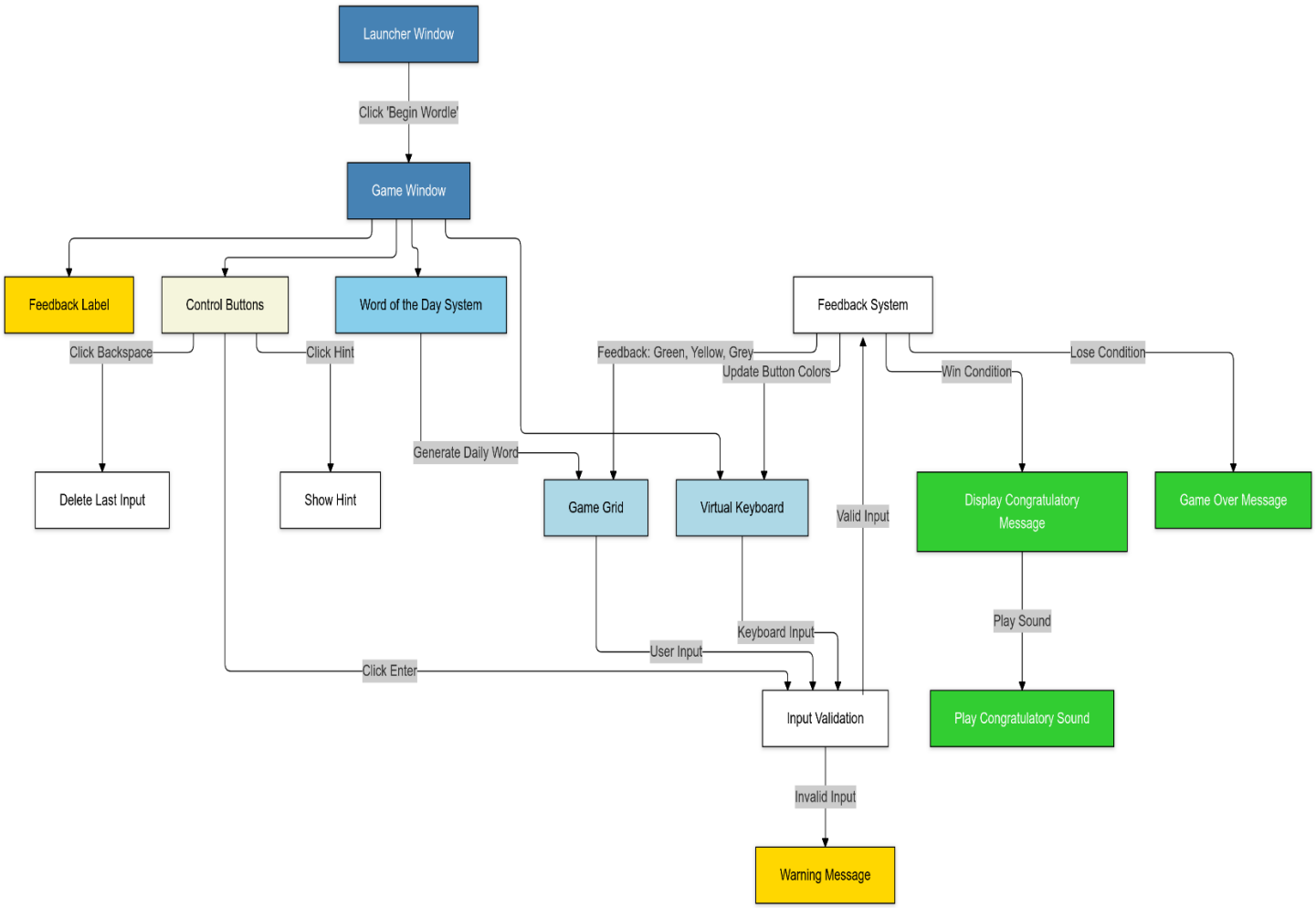
5. Scalability

The modular design makes it easy to:

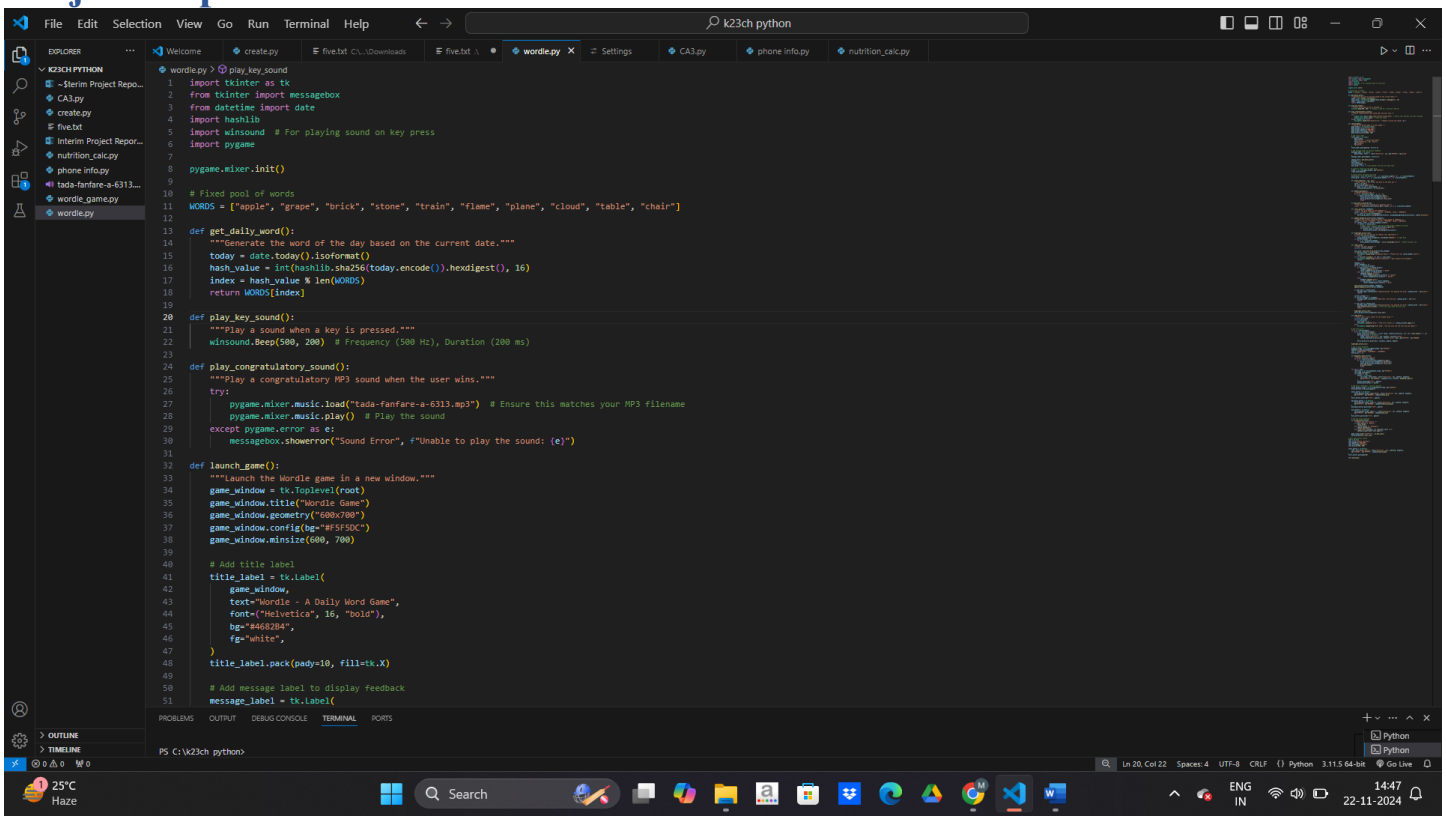
- Add new features, such as leaderboard tracking or multiplayer modes.
- Enhance the word list for a broader range of challenges.

By organizing the project into these components and functions, the Wordle game achieves its objectives of being an engaging, educational, and fun experience for players.

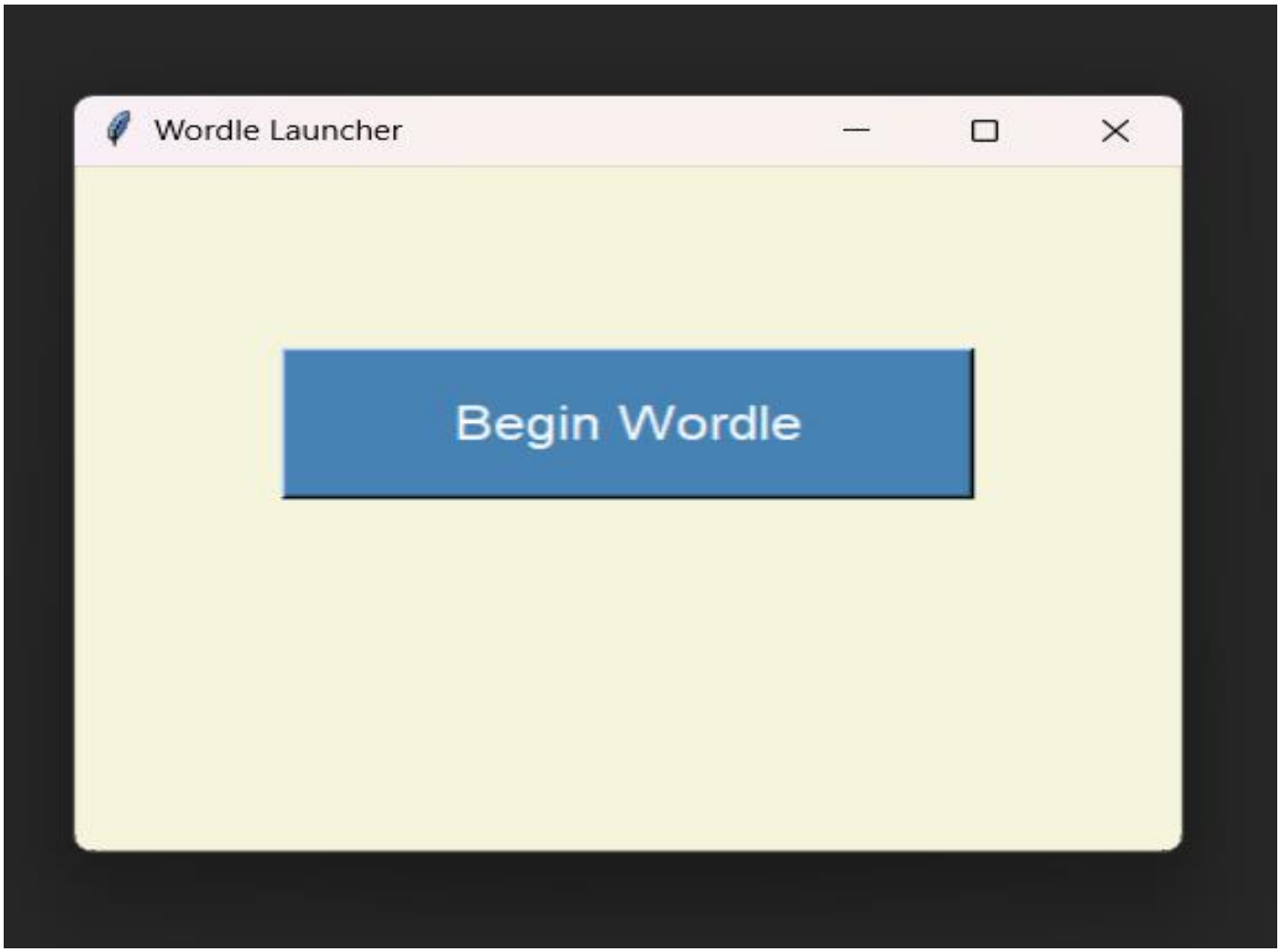
Flowchart

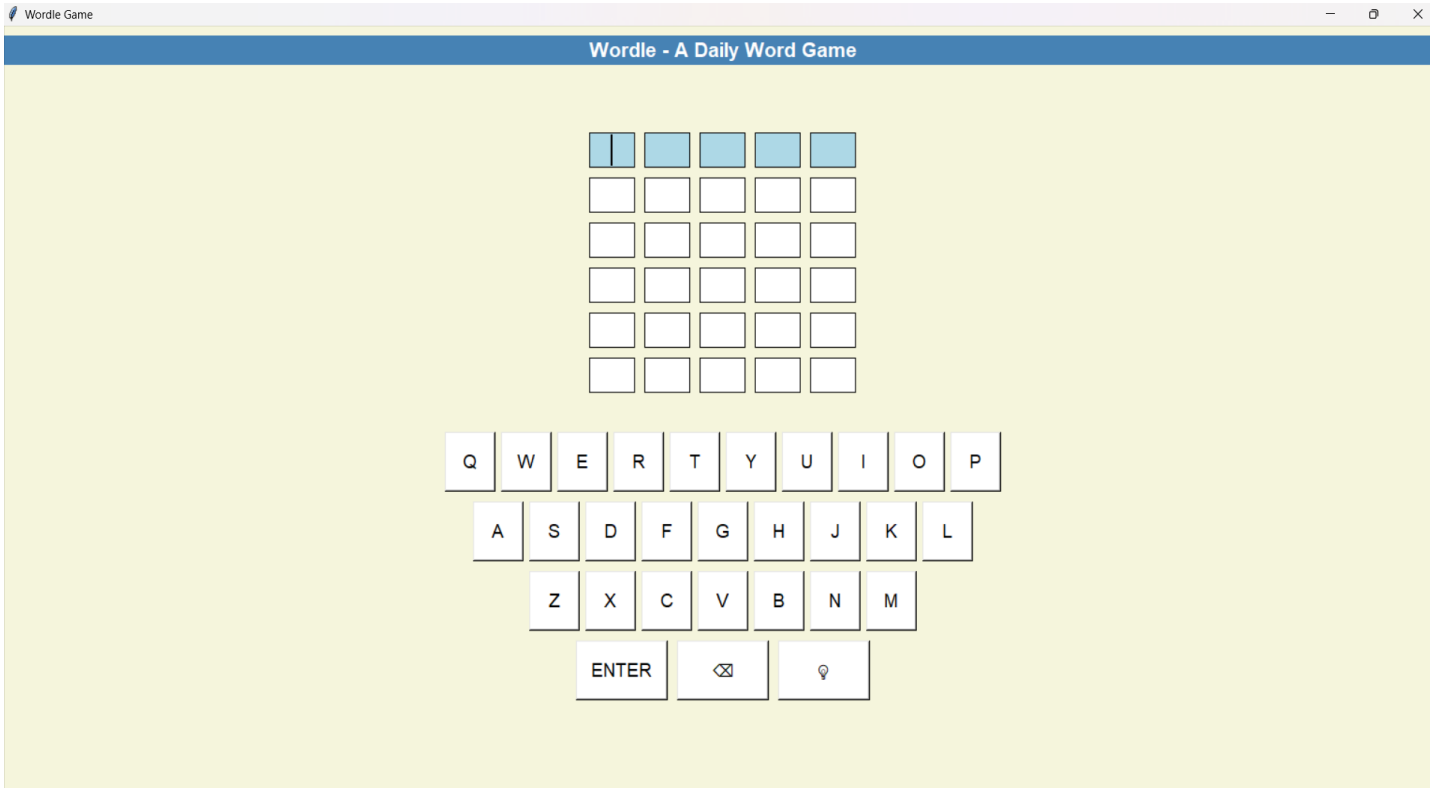

Launcher Window
Click 'Begin Wordle'
Game Window
Feedback Label
Control Buttons
Word of the Day System
Feedback System
Click Backspace
Click Hint
Feedback: Green, Yellow, Grey
Update Button Colors
Win Condition
Lose Condition
Delete Last Input
Show Hint
Generate Daily Word
Game Grid
Virtual Keyboard
Valid Input
Display Congratulatory Message
Game Over Message
Play Sound
Keyboard Input
User Input
Click Enter
Input Validation
Play Congratulatory Sound
Invalid Input
Warning Message

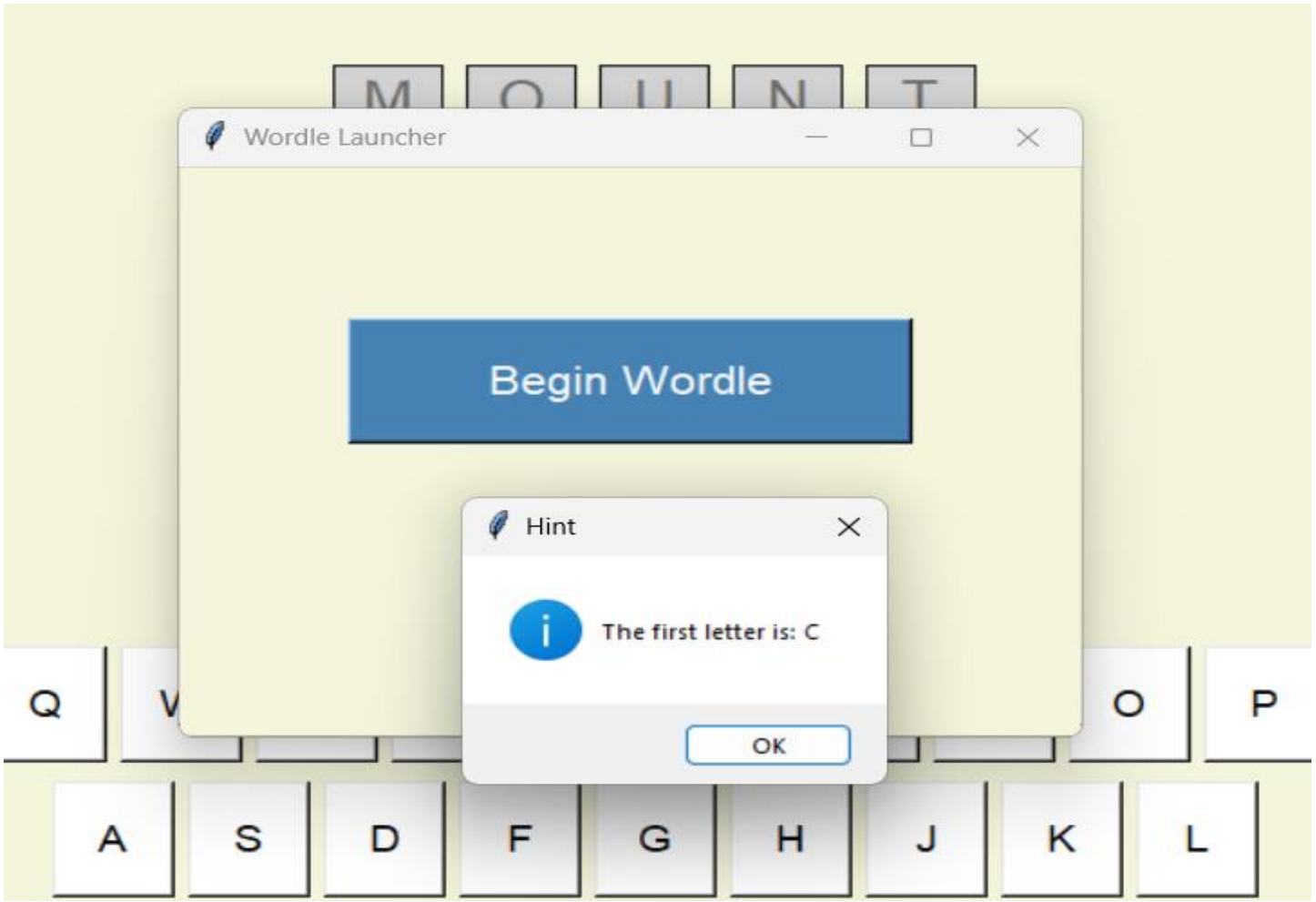
Project Implementation

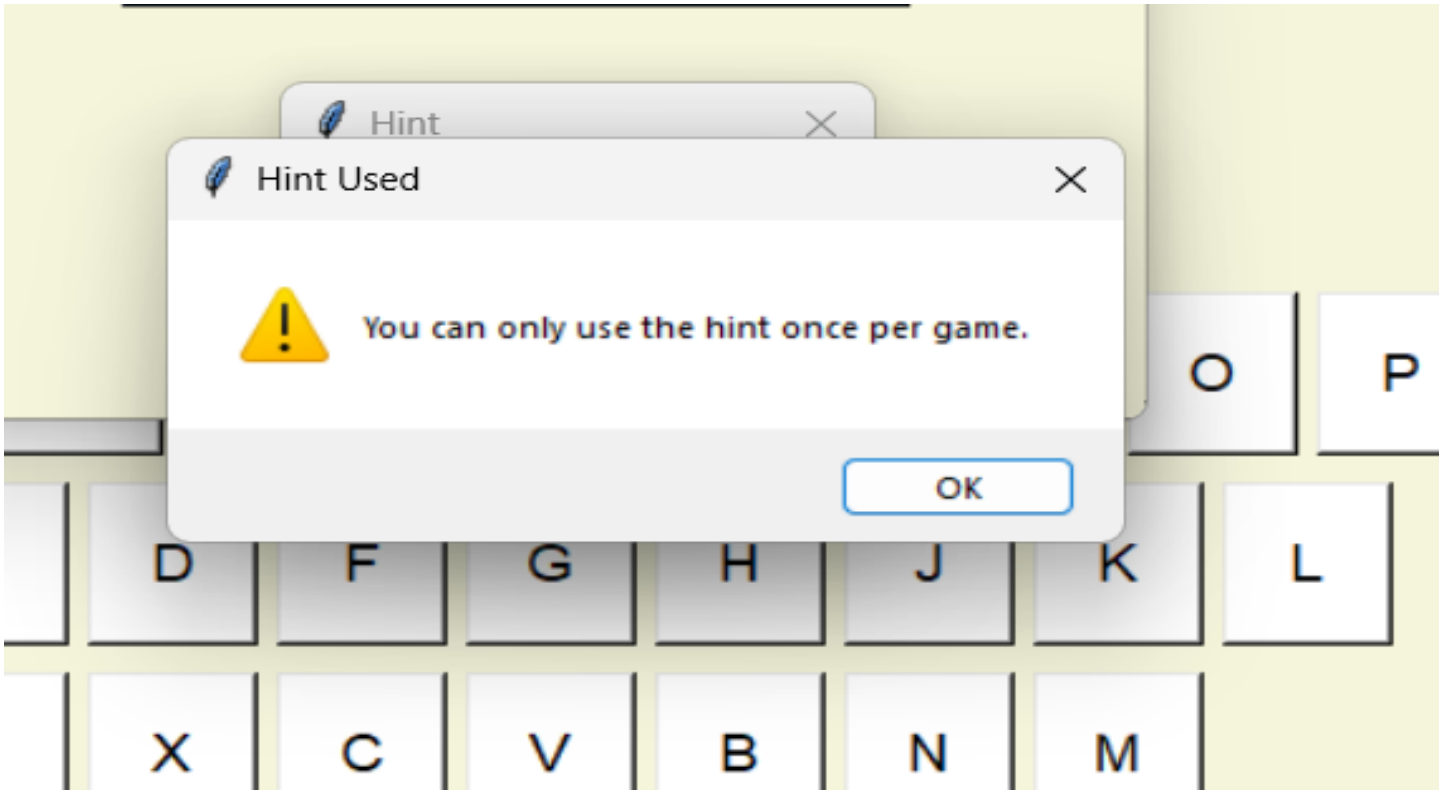


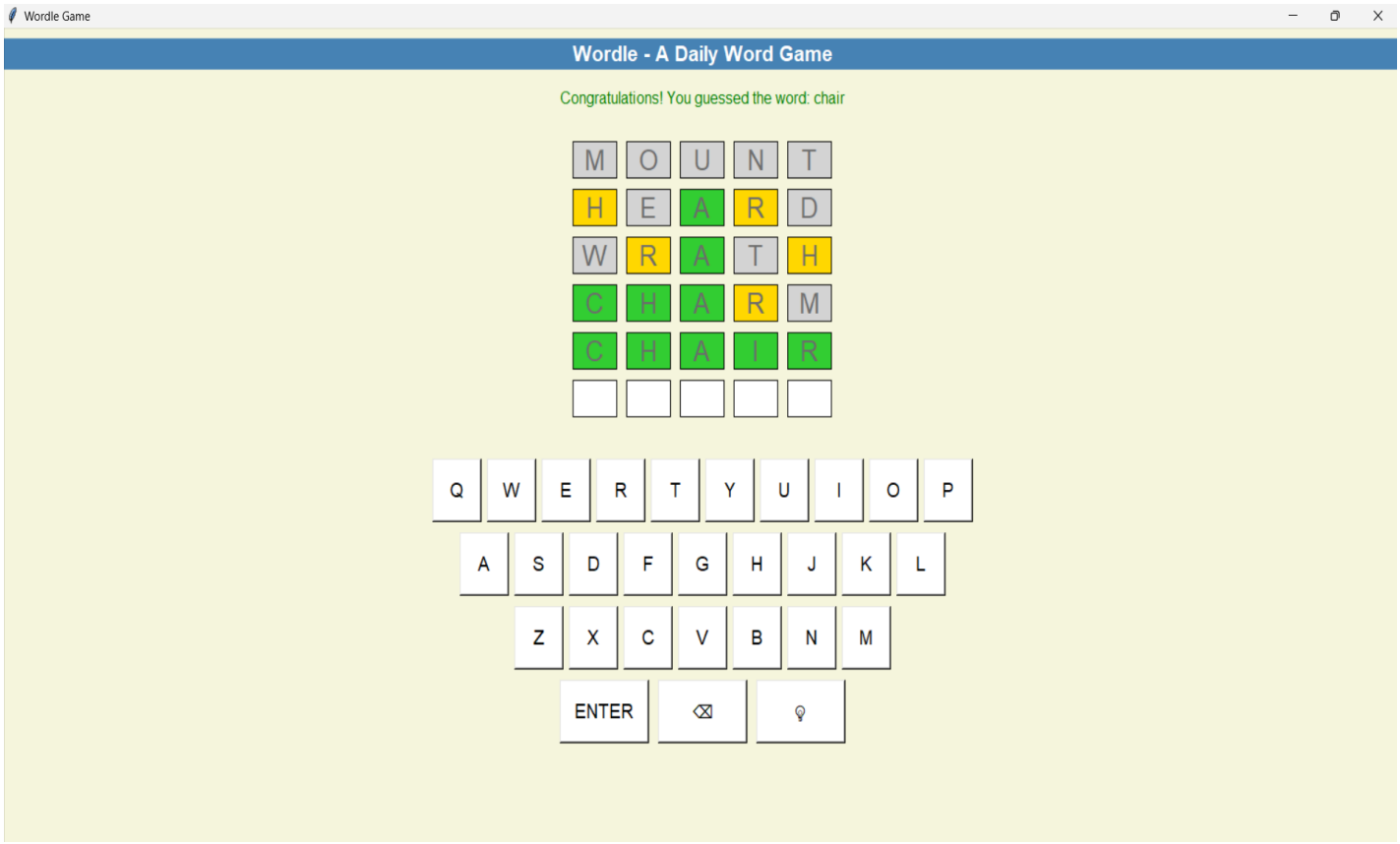
```python
import tkinter as tk
from tkinter import messagebox
from datetime import date
import hashlib
import winsound  # For playing sound on key press
import pygame

pygame.mixer.init()

# Fixed pool of words
WORDS = ["apple", "grape", "brick", "stone", "train", "flame", "plane", "cloud", "table", "chair"]

def get_daily_word():
    """Generate the word of the day based on the current date."""
    today = date.today().isoformat()
    hash_value = int(hashlib.sha256(today.encode()).hexdigest(), 16)
    index = hash_value % len(WORDS)
    return WORDS[index]

def play_key_sound():
    """Play a sound when a key is pressed."""
    winsound.Beep(500, 200)  # Frequency (500 Hz), Duration (200 ms)

def play_congratulatory_sound():
    """Play a congratulatory MP3 sound when the user wins."""
    try:
        pygame.mixer.music.load("tada-fanfare-a-6313.mp3")  # Ensure this matches your MP3 filename
        pygame.mixer.music.play()  # Play the sound
    except pygame.error as e:
        messagebox.showerror("Sound Error", f"Unable to play the sound: {e}")

def launch_game():
    """Launch the Wordle game in a new window."""
    game_window = tk.Toplevel(root)
    game_window.title("Wordle Game")
    game_window.geometry("600x700")
    game_window.config(bg="#F5F5DC")
    game_window.minsize(600, 700)

    # Add title label
    title_label = tk.Label(
        game_window,
        text="Wordle - A Daily Word Game",
        font=("Helvetica", 16, "bold"),
        bg="#468284",
        fg="white",
    )
    title_label.pack(pady=10, fill=tk.X)

    # Add message label to display feedback
    message_label = tk.Label(
```

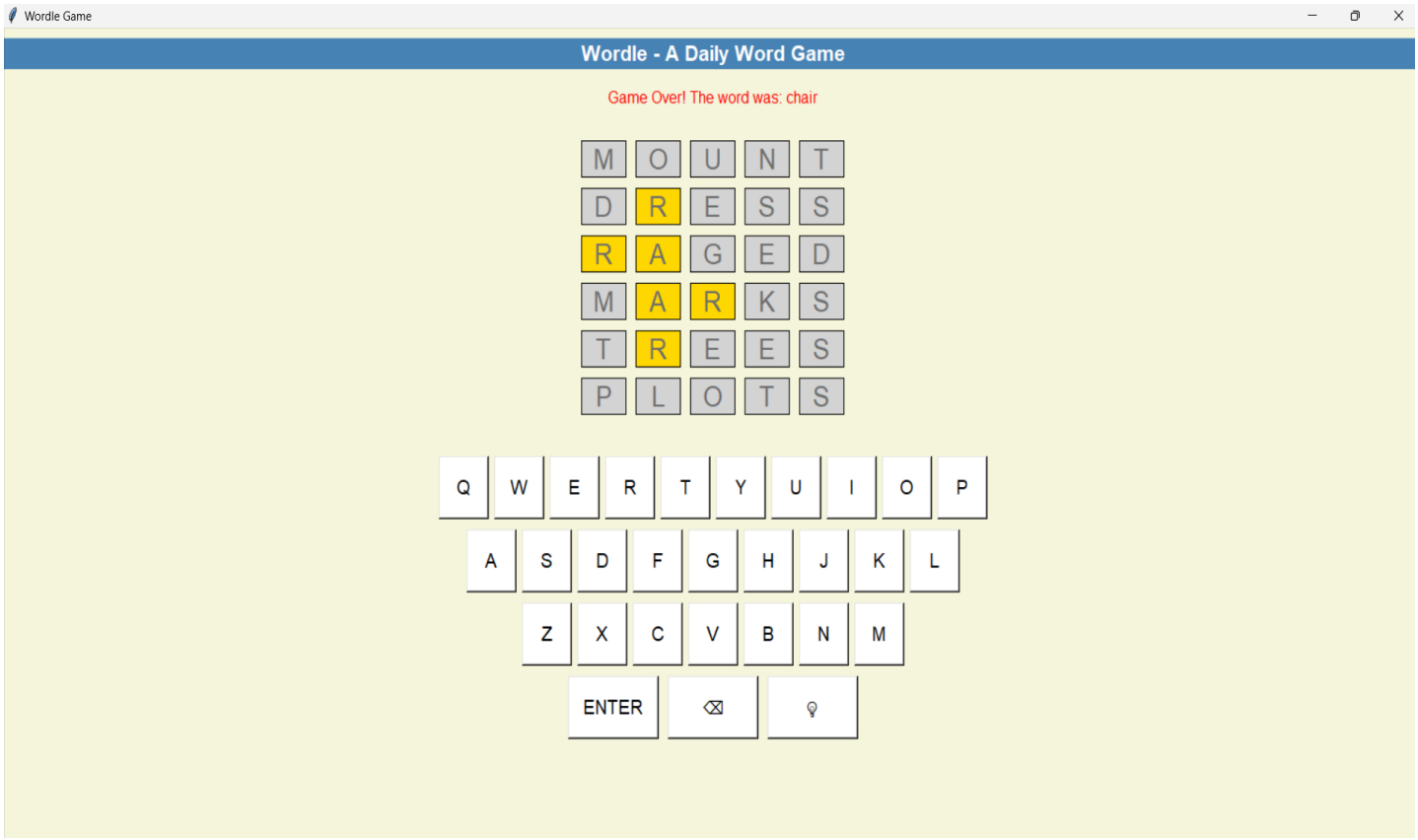



Wordle Game
Wordle - A Daily Word Game
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
ENTER


Wordle Launcher
Begin Wordle
Hint
The first letter is: C
OK


Hint
Hint Used
You can only use the hint once per game.
OK
O
P
D
F
G
H
J
K
L
X
C
V
B
N
M


Wordle Game
Wordle - A Daily Word Game
Congratulations! You guessed the word: chair
M O U N T
H E A R D
W R A T H
C H A R M
C H A I R
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
ENTER



Testing and Validation

Unit Testing Test Cases

Test Case ID: UT_01

- Test Designed by: Meghana E
- Test Designed date: 22/11/24
- Test Title: **Starting the game**
- Test Execution date: 22/11/24

Description: Test if the game launches correctly when the user clicks the "Begin Wordle" button.

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Open the application.	**N/A**	The application gets started.	Wordle Launcher starts with "Begin Wordle" button.	Pass	
2	Click on the "Begin	N/A	A new game window opens with the title "Wordle - A Daily Word	New game window opened	Pass	Test successful, game window launches.

	Wordle" button.		Game", entry grid, and virtual keyboard.	with title, grid, and keyboard		

Test Case ID: UT_02

- Test Designed by: Meghana E
- Test Designed date: 22/11/24
- Test Title: **Entering a word**
- Test Execution date: 22/11/24

Description: Test if the user can input a word using the virtual keyboard and receive feedback.

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Click on a letter (e.g., "A") from the virtual keyboard. Enter additional letters to complete the word, such as "apple".	"apple"	The entered letters ("A", "P", "P", "L", "E") should appear in the grid in their respective boxes. After clicking "ENTER", the game should provide feedback on the entered word: - "A" should be marked **green** (correct letter in the correct position). - "P" should be marked **green** (correct letter in the correct position). - "P" should be marked **green** (correct letter in the correct position). - "L" should be marked **green** (correct letter in the correct position). - "E" should be marked **green** (correct letter in the correct position).	Letters "A", "P", "P", "L", "E" appear in the grid when clicked.	Pass	
2	After entering all letters, click the "ENTER" button to submit the guess.	N/A	The "ENTER" button should submit the word, and feedback colours should update to reflect the correctness of each letter.	Upon pressing "ENTER", the grid shows all letters in green, indicating the word "apple" is correctly guessed.	Pass	The feedback mechanism works correctly, the entered word is displayed, and all feedback colours are shown as expected.

3	Enter a word with invalid characters (e.g., "app3e").	app3e	A warning message should appear: "Only letters are allowed!"	Warning message is shown as expected	Pass	Error message is displayed for invalid characters.

Test Case ID: UT_03

- Test Designed by: Meghana E
- Test Designed date: 22/11/24
- Test Title: **Use of Hint**
- Test Execution date: 22/11/24

Description: Test the hint functionality, ensuring the user can only use the hint once per game.

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Click on the "💡" hint button.	**N/A**	The first letter displays.	Hint is displayed by showing first letter.	Pass	Hint functionality works as expected.
2	Click the "💡" button again.	N/A	The hint should be displayed again.	The second click shows the warning: "You can only use the hint once per game."	Fail	You can see the hint only once.

System Testing Test Cases

Test Case ID: ST_01

- Test Designed by: Meghana E
- Test Designed date: 22/11/24
- Module Name: Game Start and Initialization
- Test Executed by: Meghana E
- Test Title: Verify the game starts successfully with all the functionalities.
- Test Execution date: 22/11/24
- Description: Test if the game starts correctly and displays all required elements.

Step	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes
1	Launch the Wordle application.	N/A	The game window opens and displays the grid and virtual keyboard.	The game window opened successfully.	Pass	
2	Enter the word "grape" using the virtual keyboard (click on letters one by one).	“grape”	After entering "grape" and pressing "ENTER", the grid should provide feedback: green for "G", "R", "A", "P", "E" if correct.	The word "grape" was correctly processed, with each letter showing green (correct position).	Pass	
3	Click the "ENTER" button to submit the word.	N/A	The word should be submitted.	“Enter” functionality worked, word submitted successfully.	Pass	
4	Use the "💡" hint button and check the hint.	N/A	The hint button should display the first letter of the word after a click. The hint should be displayed as many times as the user wants.	The hint button showed the first letter correctly on the first click and displayed the warning on the second click.	Fail	
5	After 6 attempts, check the "Game Over" message with the correct word displayed.	N/A	After the 6th attempt, a "Game Over" message should appear with the correct word revealed.	After 6 attempts, "Game Over" was displayed with the correct word "grape" revealed.	Pass	All system features tested: game launch, word input, feedback system, hint system, and game over condition.

Conclusion

The Wordle game project successfully achieves its primary goal of recreating a popular word puzzle game while adding personalized and interactive features to enhance the user experience. The application is designed to be both fun and challenging, helping users to improve their vocabulary and problem-solving skills. Below are the key accomplishments, limitations, and insights from this project :

Key Accomplishments:

1. **Core Functionality**: The game provides a user-friendly interface that allows players to guess a five-letter word within six attempts. The system gives immediate feedback on each guess by color-coding letters based on their correctness, similar to the original Wordle game.

2. **Hint System**: A hint feature was added, which provides the first letter of the word to help players if they are stuck. This improves accessibility and ensures a more enjoyable experience for users at all levels.
3. **Virtual Keyboard**: The project includes a virtual keyboard that allows players to input letters using mouse clicks, as well as by pressing physical keys on their keyboard. This ensures versatility for different user preferences.
4. **Sound Feedback**: The integration of sound effects (e.g., key press sounds and a congratulatory tone when the user wins) adds an extra layer of interactivity, making the experience more immersive.

Limitations:

1. **Limited Word Pool**: The game uses a fixed set of words, which can lead to repetitive gameplay over time. Expanding the word pool or incorporating an external word database could add variety and increase replay ability.
2. **Lack of Difficulty Settings**: The game does not currently offer adjustable difficulty settings. A future improvement could include varying the word length or complexity or implementing additional hints for more advanced players.
3. **No Multiplayer Mode**: The project does not include any multiplayer functionality, limiting it to a single-player experience. Implementing a multiplayer option would increase the game's engagement and competitiveness.

IMPACT

The Wordle game provides an enjoyable and educational experience for users, helping them develop their cognitive skills in a gamified environment. It also serves as a showcase for the application of Python programming in real-world projects. The successful integration of user interface components, feedback systems, and multimedia elements showcases how a simple game can be made interactive, engaging, and fun.

Overall, the project is a significant step in the creation of interactive and accessible gaming applications, and its foundation offers great potential for future enhancements and broader applications in educational technology.

References:

1. Tkinter Documentation: https://docs.python.org/3/library/tkinter.html
2. pygame Documentation: https://www.pygame.org/docs/
3. "How to Use Tkinter in Python" – Real Python. Available at: https://realpython.com/python-gui-tkinter/
4. “Building Games with Python and Pygame” – FreeCodeCamp. Available at: https://www.freecodecamp.org/news/python-game-development-with-pygame/