**Interim Project Report**

On

**“Wordle – A Daily Word Game”**

Submitted By:

**Meghana E**

**12303998**



**“**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**

[**Acknowledgement** 2](#_Toc183176966)

[**Introduction** 2](#_Toc183176967)

[**Objectives and Scope of the Project** 4](#_Toc183176968)

[**Application Tools** 5](#_Toc183176969)

[**Project Design** 7](#_Toc183176970)

[**Flowchart** 11](#_Toc183176971)

# **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.

1. **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).

1. **Sound Integration**:

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

1. **Hint System**:

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

1. **Input Validation**:

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

1. **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.

1. **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.

1. **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).

1. **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.

1. **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.

1. **Operating System**

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

1. **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.

1. **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**

