



Excelssior Education Society's
K. C. College of Engineering and Management Studies and Research
(Affiliated to the University of Mumbai)
Mith Bunder Road, Near Hume Pipe, Kopri, Thane (E)-400603

HANGMAN GAME
submitted in partial fulfillment of the requirement of
BACHELOR
IN
COMPUTER ENGINEERING
BY

Armaan Nakhuda	02
Sushant Navle	05
Samay Pandey	09

Supervisor
Prof. Irin Solomon



Department Of Computer Engineering
K.C. College of Engineering and Management Studies and Research
Mith Bunder Road, Kopri, thane (E) - 400603
University of Mumbai (AY 2023-24)

CERTIFICATE

This is to certify that the skill lab mini project entitled “HANGMAN GAME” is Bonafide work of Armaan Nakhuda (B-02), Sushant Navle (B-05), Samay Pandey (B-09) submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of “Bachelor of Engineering in Computer Engineering”.

Supervisor

Prof. Irin Solomon

Head of Department

Prof. Mahesh Maurya

Principal

Dr. Vilas Nitnavare

Mini Project Approval

This mini project entitled HANGMAN GAME by Armaan Nakhuda (B-02), Sushant Navle (B-05), Samay Pandey (B-09) is approved for the degree of Bachelor of engineering in Computer Engineering.

Examiners

1.....

(Internal Examiner Name & sign)

2.....

(External Examiner name & sign)

Date: -

Place: -

Contents

Abstract

Acknowledgement

List of abbreviations

List of figures

1. Introduction

1.1 Introduction

1.2 Motivation

1.3 Problem Statement & Objectives

2 . mini project contribution

3 . Proposed System

3.1 Introduction

3.2 Architecture or Framework

3.3 Details of hardware and software

3.4 Experiment and Results

3.5 Conclusion and future work

4 . References

List Of Figure

Figure no.	Figure Name	Page no.
1.		
2.		
3.		
4.		
5.		
6.		
7.		

Abstract

This project introduces a Hangman game developed in Python with the PyQt library for the user interface and Firebase as the backend database for word storage. Hangman is a classic word-guessing game where players attempt to guess a word by suggesting letters within a limited number of attempts.

PyQt facilitates the creation of an interactive graphical user interface for seamless gameplay. Players are presented with a blank word to guess and a set of letters to choose from. As they make guesses, the interface dynamically updates to reveal correct letters and display the hangman figure for incorrect ones.

Firebase serves as the database backend, storing a diverse collection of words for the game. This integration ensures a varied selection of words, enhancing the game's replayability across different categories and difficulties.

The project demonstrates the integration of Python, PyQt, and Firebase to create an engaging gaming experience. By leveraging cloud-based word storage, players can enjoy a continuously updated pool of words, adding depth and challenge to each gameplay session. This implementation highlights the synergy between modern technologies in game development, offering a glimpse into the potential of combining cloud services with graphical interfaces for enhanced user experiences in gaming applications.

Acknowledgement

We would like to express our sincere gratitude to all those who contributed to the development and success of this HANGMAN GAME project.

First and foremost, we extend our appreciation to our team members for their dedication and hard work in designing, programming, and testing the management system. Their collective efforts played a vital role in bringing this project to fruition.

We are also thankful for the guidance and support of our project supervisor Prof. Irin Solomon whose expertise and insights were invaluable throughout the development process. Your mentorship and encouragement significantly enriched our project.

We acknowledge the numerous individuals who provided feedback, participated in beta testing, and contributed to refining the HANGMAN GAME. Your input was essential in enhancing the system's overall user experience, and we are grateful for your valuable contributions.

We extend our gratitude to the open-source software community, whose contributions laid the foundation for many tools and libraries incorporated into this project. The collaborative spirit of the community has been a driving force behind the success of the HANGMAN GAME.

Finally, we want to express our heartfelt thanks to our friends and family for their unwavering support and understanding during the project's development period. Your encouragement and patience were essential in helping us overcome challenges and stay motivated.

This HANGMAN GAME project would not have been possible without the collaborative efforts and support of all these individuals and groups. Thank you for being an integral part of this transformative journey.

List Of Abbreviation

List of Abbreviations for HANGMAN GAME:

1. GUI: Graphical User Interface
2. CPU: Central Processing Unit
3. UI: User Interface
4. SP: Single Player
5. DB: Database

This list provides a quick reference to the abbreviations used in the HANGMAN GAME

INTRODUCTION

1.1 Introduction

The Hangman Game with Python, PyQt, and Firebase Integration is a modern rendition of the classic word-guessing game, designed to offer an engaging and interactive gaming experience. Developed using Python, PyQt, and Firebase technologies, this project brings together a blend of programming prowess and contemporary database management.

With PyQt as the framework for the graphical user interface (GUI), players can enjoy a visually appealing and intuitive interface, allowing seamless interaction with game elements such as masked words and letter selection buttons. Firebase integration serves as the backbone of the game, providing a dynamic repository of words for gameplay. This integration ensures a constantly evolving word pool, enriching the gaming experience with a diverse range of words across various categories.

1.2 Motivation for Creating A HANGMAN GAME

The motivation behind creating the Hangman Game in Python stems from a desire to combine classic entertainment with modern technology. By implementing the game using Python, we aimed to showcase the language's versatility and ease of development, making it accessible to both novice and experienced programmers. Additionally, integrating PyQt for the graphical user interface and Firebase for word storage adds depth and dynamism to the gaming experience, aligning with contemporary trends in software development. Ultimately, this project serves as a testament to the creative potential of Python and its ability to breathe new life into timeless pastimes like Hangman.

1.3 Problem Statements and Objectives

Problem Statements: -

In contemporary gaming environments, the Hangman project tackles several key challenges prevalent in traditional game development practices, impeding efficiency and hindering optimal gameplay experiences. Below are some problem statements specific to the Hangman project:

1. **Manual and Tedious Gameplay Mechanisms:** The Hangman game relies on manual processes for tasks such as word selection, letter masking, and user interaction, leading to inefficiencies and sluggish gameplay. Manual handling of these tasks not only increases the likelihood of errors but also diminishes the overall enjoyment and engagement of players.
2. **Fragmented Data Management:** Existing Hangman implementations often lack centralized data management, resulting in scattered word repositories and disjointed gameplay experiences. This fragmentation leads to difficulties in maintaining a cohesive collection of words and extracting meaningful insights for enhancing gameplay dynamics.
3. **Limited Word Diversity:** Many Hangman implementations suffer from a limited pool of words, leading to repetitive gameplay experiences and reduced player engagement. The absence of a diverse range of words hampers the game's replay value and fails to cater to the varied preferences of players.

Objective: -

The objectives while creating the Hangman game were:

1. **Entertainment:** The primary goal is to provide players with an enjoyable and engaging gaming experience. The game should be entertaining, immersive, and capable of captivating players' attention for extended periods.
2. **Education:** Hangman can serve as an educational tool for improving vocabulary, spelling, and language skills. Incorporating diverse word categories and difficulty levels helps players learn new words and concepts while having fun.
3. **Challenge:** The game should offer an appropriate level of challenge to keep players motivated and engaged. Balancing difficulty levels, providing incentives for progression, and offering hints or assistance when needed are crucial aspects of maintaining player interest.

2. Mini Project Contribution

The workflow was divided and done by all the members of the project group, the work done by each member is as follows:

1. Armaan Nakhuda: Complete Implementation and connection of DB
Documentation- Final Version
PPT
2. Sushant Navle: Basic GUI
Documentation- First draft
Activity Diagram- First version
3. Samay Pandey: working of the project(backend) along with basic retrieval of words from a text file
Dynamic word add/remove from database through the application

3. Proposed System

3.1 Introduction

The Hangman project is a Python-based game tackling challenges in traditional game development. It automates gameplay, centralizes data management, enhances word diversity, and optimizes user experience. By addressing these issues, it aims to offer an engaging and seamless Hangman gaming experience for players across different platforms.

3.2 Architecture

The architecture behind a Hangman game typically involves several key components:

1. **User Interface (UI):** The UI is responsible for presenting the game to the player and facilitating interactions. It includes elements such as the game board, letter buttons for guessing and visual representation of the hangman.
2. **Word Database:** A word database stores the words that players will guess during the game such as a predefined list of words, a pre-generated list based on difficulty levels.
3. **Game Logic:** The game logic handles the core functionality of the game, including word selection, letter validation, updating the hangman's visual representation, tracking the player's progress, and determining the outcome of the game (win, loss, or ongoing).
4. **visual Assets:** Graphics, animations, and contribute to the overall aesthetic and atmosphere of the game. These assets are integrated into the UI and triggered based on game events or player actions.

Flowchart:-

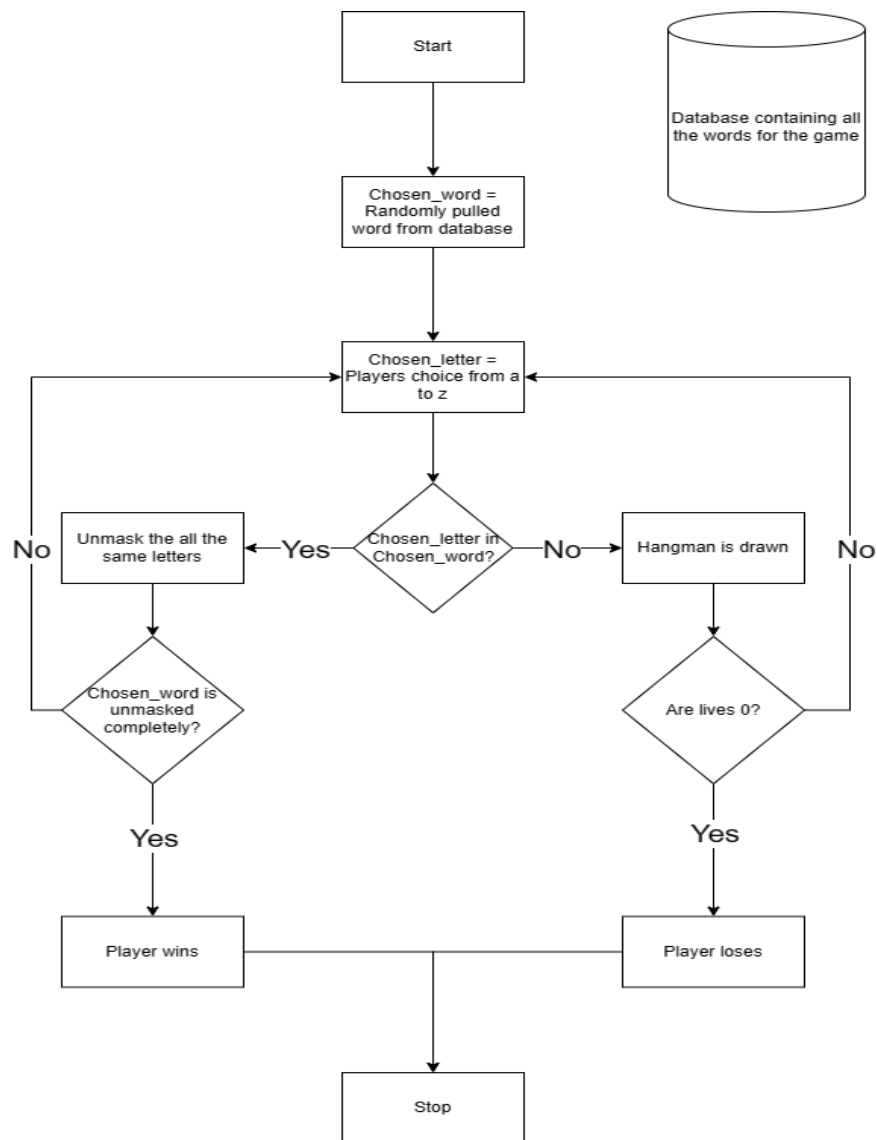


Figure 1

Figure 1 is a flowchart of the basic showcase of HANGMAN GAME.

1. Initialization: When the player starts the application, the game is started and the first word is automatically pulled from the database.
2. Database: It holds all the words used in the game.
3. Buttons: Letters from A to Z are provided so the user can select any letter they want, Extra buttons are also provided to add/remove words from the database, to reset the game or give up.
4. Outcome: Based on the winning or losing of the player the toast is displayed.

3.3 Hardware And Software Requirements:-

Hardware :-

1. Minimum Requirements:

- a) Processor:- Dual core processor @2.4Ghz
- b) Ram:- 4GB Ram
- c) Storage:- 2GB free space

2. Recommended Requirements:

- a) Processor:- Quad core processor @2.8Ghz
- b) Ram:- 8GB Ram
- c) Storage:- 4GB free space

Software :-

1. Minimum Requirements:

- a) OS: Windows 10 22H2
- b) Python version 3.11 with firebase and PyQt5 modules installed.

2. Recommended Requirements:

- a) OS: Windows 11 22H2
- b) Python version 3.12.2 with firebase and PyQt5 modules installed.

3.4 Experimental Result:-

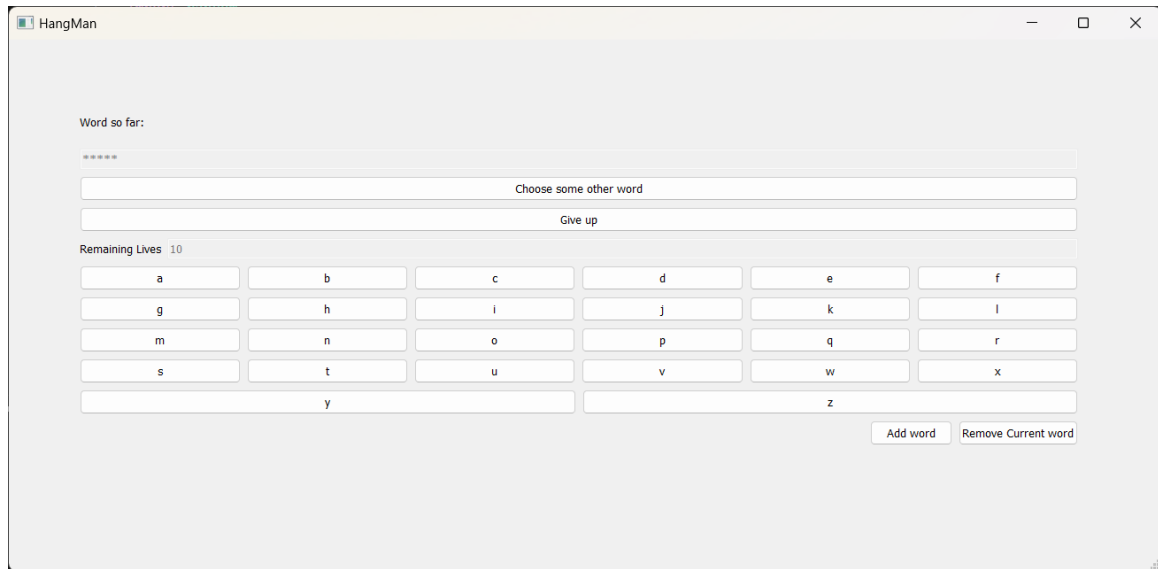


Figure 2

Figure 2 :- The main game screen shown to the user, which can be played instantly. The word is completely hidden and no letters are shown. The player starts with 10 lives and the lives reduce as the incorrect letters are clicked.

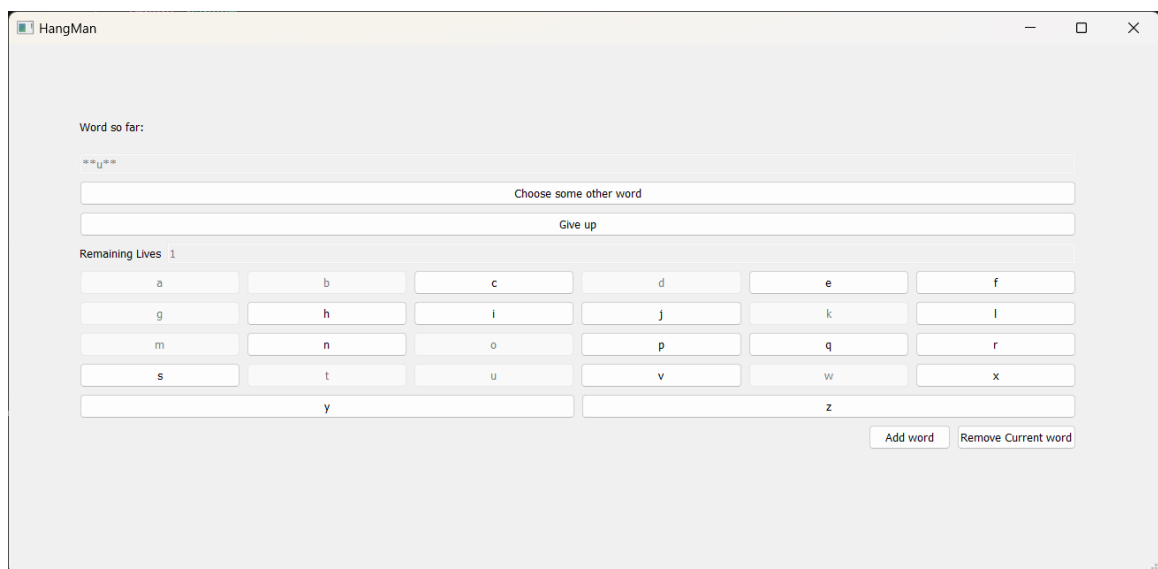


Figure 3

Figure 3 :- As the game is played and the letters are clicked, the buttons get disabled. When the correct letter is clicked the letter unhides itself to reveal part of the word.

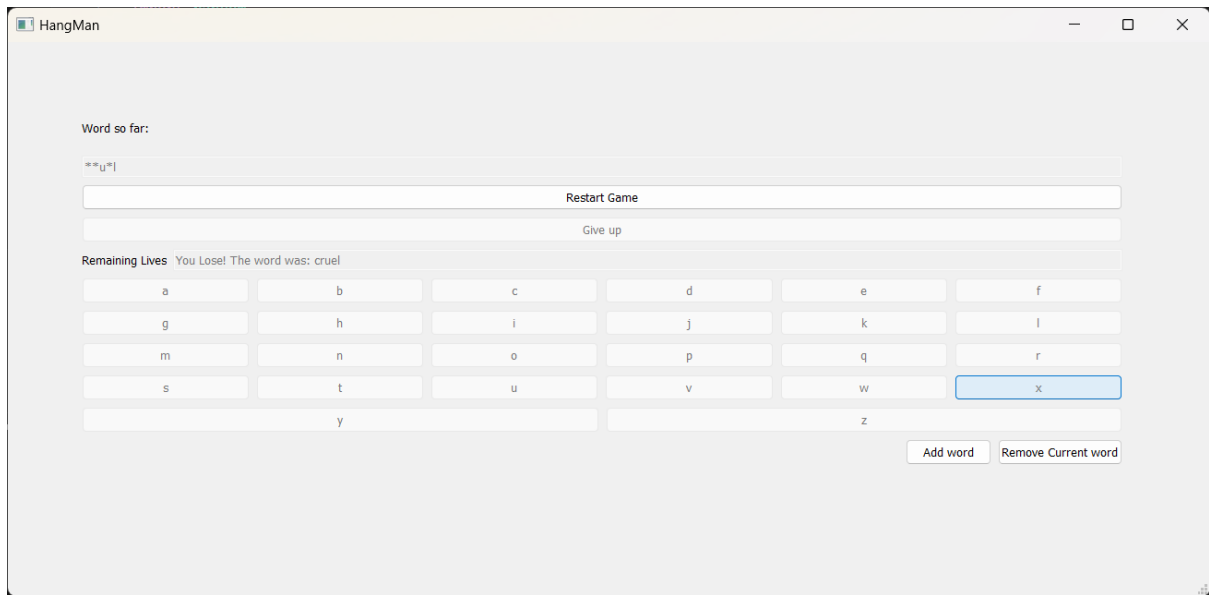


Figure 4

Figure 4 :- When the lives go to 0 for the player, the game ends and the whole word is revealed. The give up button is also disabled as the game is over.

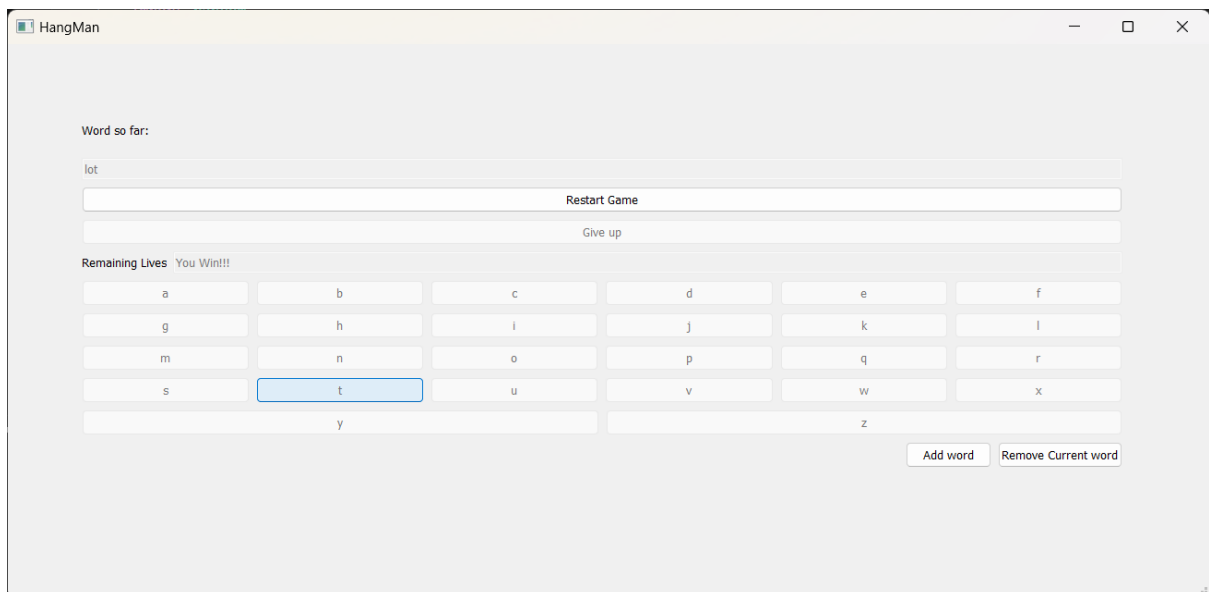


Figure 5

Figure 5 :- If the player wins the game the complete word is shown along with the winning toast and the give up button is disabled as the game has ended.

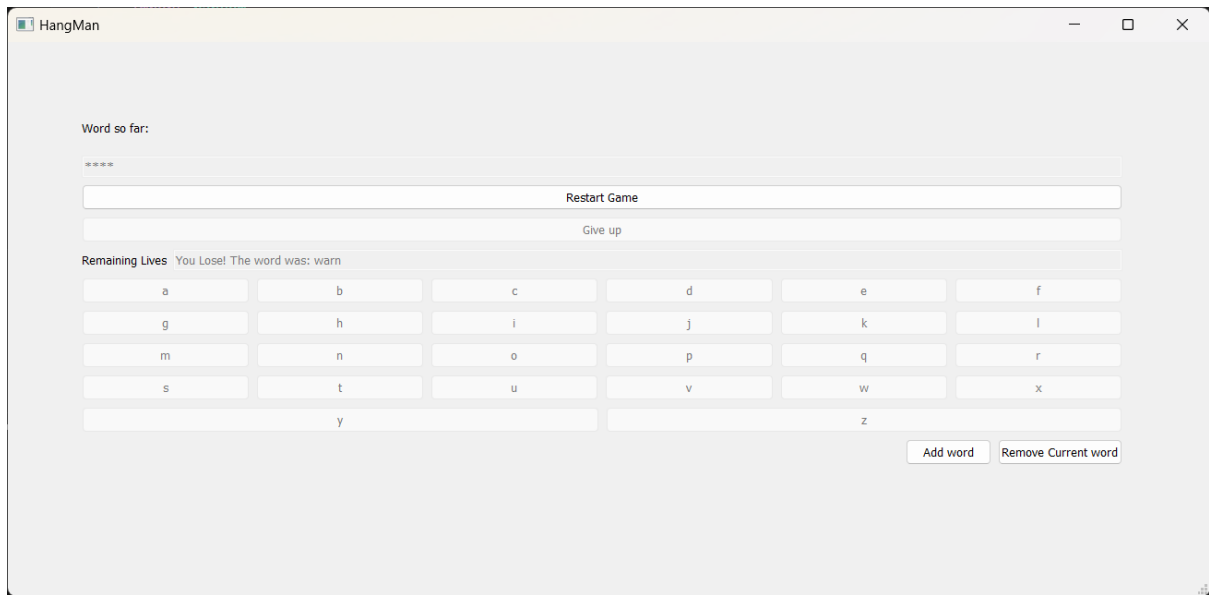


Figure 6

Figure 6 :- When the give up button is used, the losing toast is shown and the word is revealed to the player.

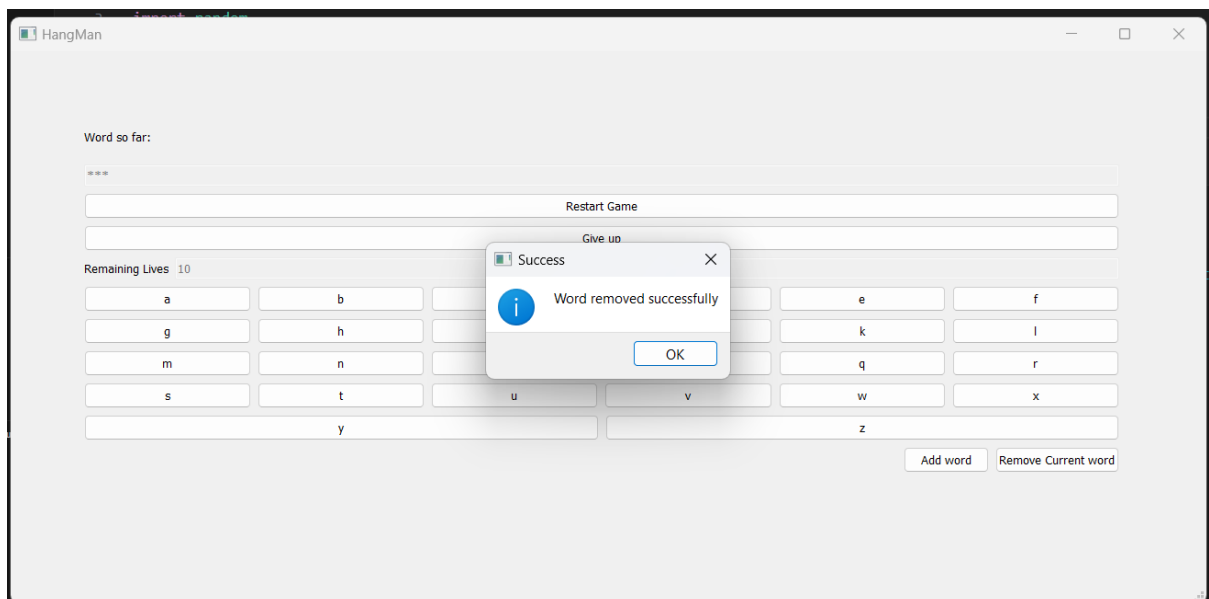


Figure 7

Figure 7 :- Using the remove current word button, completely removes the word from the database so that it wont be used in any client in a new instance.

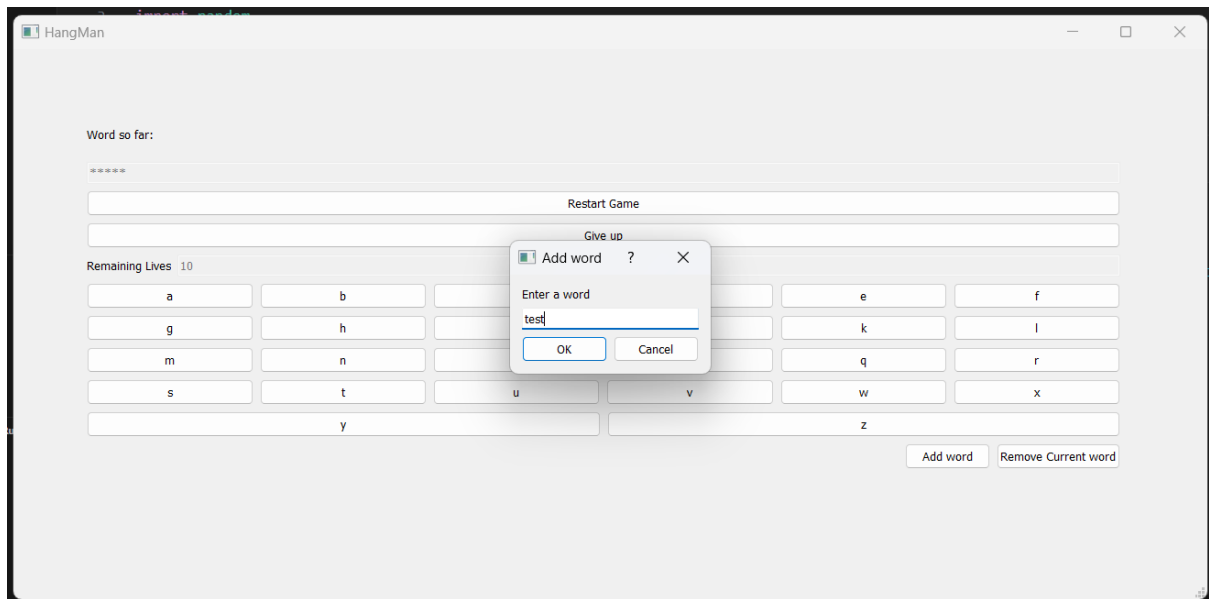


Figure 8

Figure 8 :- When the user clicks on the add word button a new pop up is shown which asks the word to be entered in the dialog box along with the option to finalize the word or to cancel the process.

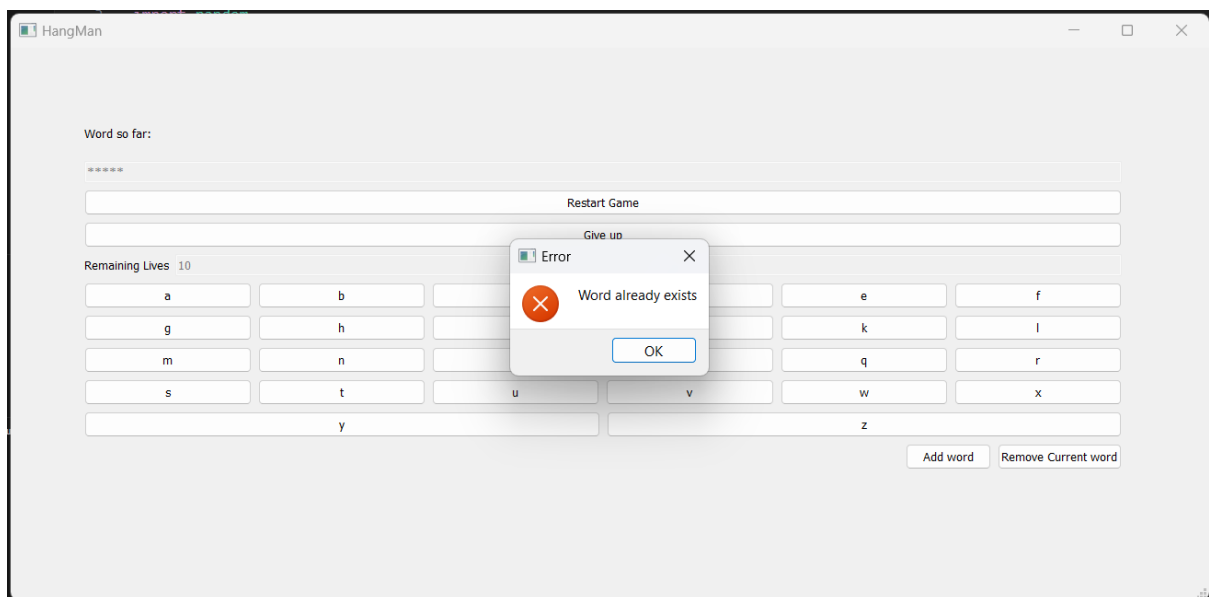


Figure 9

Figure 9 :- The Word which needs to be added is checked against the database to check if it's a repeated word or not. If the word is repeated then this error is shown.

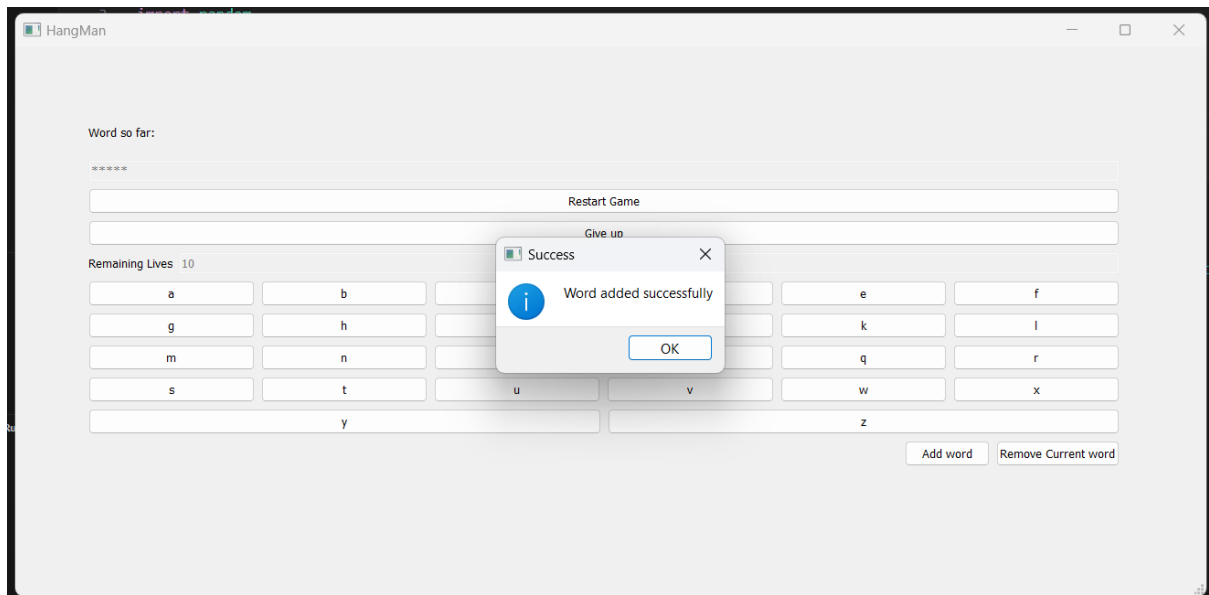


Figure 10

Figure 10 :- If the new word which needs to be added into the database is not found then it is added in the first empty spot found in the database.

3.5 Conclusion and Future Scope :-

Conclusion:-

In conclusion, Hangman remains a timeless favourite, blending simplicity with challenge and educational value. It fosters language skills and strategic thinking. With its enduring appeal and innovative design, Hangman stands as a testament to the power of classic entertainment in the digital era, bringing joy.

Future Scope:-

The future scope of the Hangman game is promising, with opportunities for expansion, innovation, and adaptation to emerging technologies and trends. Here are some potential areas for growth:

1. **Enhanced Gameplay Features:** Introducing new gameplay mechanics, such as power-ups, special abilities, or alternative game modes, can add depth and variety to the Hangman experience. These features could offer players additional challenges or strategic options to explore.
2. **Advanced AI and Personalization:** Implementing AI-powered opponents or personalized game experiences tailored to individual players' preferences and skill levels could elevate the gameplay experience. Adaptive difficulty levels, intelligent hints, and dynamic word selection algorithms can make the game more engaging and immersive.
3. **Educational Applications:** Further exploring the educational potential of Hangman by partnering with educational institutions or integrating curriculum-aligned content can position the game as a valuable learning tool. Incorporating language learning exercises, vocabulary quizzes, or themed word packs can make Hangman more appealing to educators and students alike.
4. **Monetization and Business Models:** Exploring alternative monetization strategies such as subscription services, in-game purchases for cosmetic items or premium features, or ad-supported free-to-play models can diversify revenue streams and sustain the game's development and growth.

REFERENCE

1. ChatGPT (OpenAI):
 - i. Useful for resolving technical issues and bug fixes during the project development process.
 - ii. Website: [OpenAI](#)
2. GeeksforGeeks:
 - a. A valuable resource for generating ideas, solutions to coding challenges, and technical insights.

Website: [GeeksforGeeks](#)
3. YouTube:
 - a. A platform to explore a wide range of video tutorials, demonstrations, and insights related to mobile game development and user interface design.

Website: [YouTube](#)
4. Overstack:

A community-based space to find and contribute answers to technical challenges, and one of the most popular websites in the world.
5. GitHub:

It is a developer platform that allows developers to create, store, manage and share their code. It uses Git software, providing the distributed version control of Git plus access control, bug tracking, software feature requests, task management, continuous integration, and wikis for every project.