



**Vilasrao Deshmukh Foundation, Group
of Institutions**

VDF School of Engineering and Technology, Latur

New M.I.D.C Airport Road, Latur-413531

A Report On
“Hangman Game Using Python”

Submitted By:

MD Owais
Garad Ganesh

2122541245556
2122541245557

Under the Guidance of
Prof. Bondge K.S



CERTIFICATE

This is to certify that Mr. Junaidi MD Owais Ateeque, Mr. Garad Ganesh Vishnu of Sem (VI) class TE(CE) has successfully completed the mini-project on “Hangman Game” as prescribed by Dr. Babasaheb Ambedkar Technological University, Lonere during the year 2022-2023 in the Department of Computer Engineering.

Guide
Prof.Bondge K.S

H.O.D
Prof.Wakhare.Y.R.

Principal
Vilasrao Deshmukh Foundation
Group of Institution,
Latur

ACKNOWLEDGEMENT

We would like to thank **Prof.Bondge K.S**, for helping / guiding as in selecting the topic and contents, giving valuable suggestion in preparation of seminar report and presentation.

We are graceful to **Prof.Wakhare.Y.R.**, Head of the Computer Department for providing the environment and facilitate the Department Goal makes us to work, vision is more important than Goal which makes us to do work in the best way to make equally the best. thanks to principal **Dr. Mohan Buke.** for this support.

Thanks to all colleagues for their extended support and valuable guidance, we would like to be graceful to my all friends for their constant support, help and guidance.

INDEX

1. Introduction
2. Objective of project
3. Requirement and analysis.
4. Feasibility study.
5. Hardware and software requirement.
6. Implementation details.
7. Future and scope.
8. Conclusion.
9. References.

ABSTRACT

This is a simple Hangman game using Python programming language. We can use this as a small project to boost their programming skills and understanding logic. The Hangman program randomly selects a secret word from a list of secret words. The random module will provide this ability, so line 1 in program imports it. Hangman is a popular word game in which one player (the "chooser") chooses a secret word and another player (the "guesser") attempts to guess the word one letter at a time. If a guessed letter appears in the word, all instances of it are revealed. If not, the guesser loses a chance. If the guesser figures out the secret word before he or she runs out of chances, he or she wins. If not, the player who chose the word wins.

Functionalities of this project include:

1. The game should randomly select a hidden word from a predefined list or database based on the chosen word category.
2. Players should be able to input their guesses by selecting individual letters or typing them.
3. The game should display the partially revealed word to the player, with blanks representing unguessed letters.
4. The display should update as the player makes correct guesses.
5. The system should track and display the number of attempts remaining for the player to guess the word.
6. The number of attempts should decrease when the player makes an incorrect guess.

Chapter 1

INTRODUCTION

Hangman is a popular word guessing game where the player attempts to build a missing word by guessing one letter at a time. After a certain number of incorrect guesses, the game ends and the player lose. The game ends and the player lose. The game also ends if the player correctly identifies all the letters of missing word. The code is written in c++ language. Visual Studio is used to compile the code. The code carries out all the functions that we see in a hangman game.

Background of project. The idea for the game was to make this game easy to operate without the problem of thinking a new word every time. You just click a button and a random word is guessed for you. And every time you win you level-up and anew word is waiting for you to be guessed. With every win you get a score of 5 added. And when you lose a game, your name is asked and your score is saved with your name in a text file. You can view your high score in main menu where we have given an option for high score.

Objectives and scope: We are designing the world popular game hangman. Its purpose is to turn this paper game into a computerized version. In it User has option to play game. The hangman will be drawn automatically if your chose an incorrect word. If you finish the first level, you will be taken to the next level with a new word to be guessed. And every time you clear a level a score of 5 five will be added to your overall score.

Chapter 2

Objective of Project

The objective of this project is to develop a Hangman game that provides an entertaining and engaging experience for users. The game will challenge players to guess a hidden word by guessing individual letters within a limited number of attempts. The project aims to implement intuitive user interfaces, interactive gameplay mechanics, and a variety of word categories to enhance user enjoyment. Additionally, the project will focus on incorporating features such as hints, scoring systems, and difficulty levels to cater to players of different skill levels. By developing a high-quality Hangman game, the objective is to deliver a fun and addictive gaming experience that promotes critical thinking, vocabulary expansion, and overall user satisfaction.

The project aims to serve as an educational tool by incorporating a wide range of words from various domains, such as vocabulary relevant to academic subjects, common idioms, or specialized terminology. The objective is to provide an enjoyable way for players to expand their knowledge and enhance their language skills. The project objective includes developing the Hangman game to be compatible with multiple platforms, such as desktop computers, mobile devices (both iOS and Android), and web browsers. This will ensure accessibility and allow users to enjoy the game on their preferred devices. The project aims to prioritize an intuitive and visually appealing user interface that enhances the overall gaming experience. This includes designing clear and user-friendly menus, aesthetically pleasing graphics, and engaging animations or sound effects to make the gameplay more immersive.

3.1 PRESENT SYSTEM

The present system consists of a basic implementation of the Hangman game, where players can guess letters to reveal a hidden word. The game may have a simple command-line interface or a basic graphical user interface (GUI). The present system offers a limited number of word categories or themes for players to choose from. This may include generic categories such as animals, countries, or fruits, but lacks a diverse range of specialized or educational categories.

The present system includes only essential gameplay features, such as guessing individual letters, tracking incorrect attempts, and displaying the partially revealed word. However, it may lack additional features like hints, scoring systems, or difficulty levels.

The present system is limited to a specific platform or environment, such as a specific operating system or a particular programming language. It may not be easily portable or compatible with different devices or platforms. The present system may have a rudimentary user interface with limited visual elements and lacks appealing graphics or animations. The overall user experience may be simplistic and not engaging enough. The present system only supports a single-player mode, where users can play the game individually without any multiplayer or social interaction features. The present system may lack robust error handling mechanisms, resulting in potential crashes or unexpected behavior when users input incorrect or invalid characters.

3.2 PROPOSED SYSTEM

The proposed system will feature a visually appealing and user-friendly interface. The game will have vibrant graphics, engaging animations, and intuitive menus, creating an immersive and enjoyable gaming experience for players. The proposed system will include a wide range of word categories to choose from. This will involve incorporating not only generic categories but also specialized domains, academic subjects, popular culture references, or even user-generated categories. The expanded word categories will add variety and cater to different interests and knowledge levels.

The proposed system will introduce various gameplay enhancements to make the Hangman game more engaging and challenging. This may include the implementation of difficulty levels, where players can choose between easy, medium, and hard modes, each offering a different level of complexity and word length. Additionally, the system may incorporate a scoring system to track and display players' progress, encouraging competition and motivation.

The proposed system will introduce multiplayer functionality, allowing players to compete against their friends or other online players. This will involve implementing real-time or turn-based multiplayer modes, where players can challenge each other, compare scores, and engage in friendly competition. The proposed system will include a hint feature to assist players in guessing the hidden word. Players will have the option to use hints that provide clues about the word, helping them make educated guesses and increasing their chances of success. The hint system may have limited uses to maintain the challenge and balance of the game.

The proposed system will be designed to be compatible with various platforms and devices, including desktop computers, mobile devices (both iOS and Android), and web browsers. This will allow players to enjoy the Hangman game seamlessly on their preferred devices.

By implementing these enhancements and new features, the proposed system aims to deliver an upgraded Hangman game that offers a visually appealing interface, diverse word categories, engaging gameplay mechanics, multiplayer interactions, and an overall improved user experience.

Chapter 3

Requirement and Analysis

System Analysis is about complete understanding of existing systems and finding where the existing system fail. The solution is determined to resolve issues in the proposed system. It defines the system. The system is divided into smaller parts. Their functions and inter relation of these modules are studied in system analysis. The complete analysis is followed below.

4.1 Problem definition

1. Lack of Engaging and Educational Word Game: The problem is the absence of an entertaining and educational word game that provides an engaging experience for players. Existing Hangman games may be limited in terms of word categories, gameplay features, and overall user experience, resulting in a less enjoyable and immersive gaming experience.
2. Limited Word Categories and Customization Options: The problem lies in the restricted number of word categories available in Hangman games. Players may not have access to a wide range of categories that cater to their interests or provide educational value. Furthermore, the lack of customization options prevents players from adding or creating their own word categories, limiting the game's flexibility and personalization.
3. Minimal Gameplay Features: Existing Hangman games may offer only basic gameplay features, such as guessing individual letters and tracking incorrect attempts. This results in a relatively simplistic experience that may not sufficiently challenge or engage players. Additional gameplay features, such as difficulty levels, scoring systems, hints, and multiplayer functionality, are lacking and contribute to a less dynamic and immersive gaming experience.

4. Limited User Interface and Experience: The problem is the lack of a visually appealing and user-friendly interface in current Hangman games. The absence of engaging graphics, animations, and sound effects may diminish the overall user experience. Inadequate feedback and guidance throughout the game may also hinder the player's progress and enjoyment.

5. Incompatibility with Multiple Platforms: Existing Hangman games may be restricted to specific platforms or environments, limiting accessibility and hindering the game's reach. The lack of compatibility with various platforms, such as desktop computers, mobile devices (iOS and Android), and web browsers, prevents players from enjoying the game on their preferred devices.

6. Insufficient Error Handling and Resilience: The problem lies in the inadequate error handling mechanisms within Hangman games. Users may encounter crashes or unexpected behaviour when inputting invalid characters or actions. This undermines the overall user experience and may lead to frustration or disengagement.

By addressing these problems through the development of an enhanced Hangman game, the project aims to provide an entertaining, educational, and immersive gaming experience. The game will offer diverse word categories, customizable options, engaging gameplay features, an appealing user interface, platform compatibility, and robust error handling mechanisms to ensure a seamless and enjoyable experience for players.

4.2 Requirement specification

Requirement Specification for a Hangman Game:

1. Functional Requirements:

- a. Word Selection: The game should randomly select a hidden word from a predefined list or database based on the chosen word category.
- b. Guessing Mechanism: Players should be able to input their guesses by selecting individual letters.
- c. Word Display: The game should display the partially revealed word to the player, with blanks representing unguessed letters.
- d. Attempt Tracking: The system should track and display the number of attempts remaining for the player to guess the word.
- e. Correct Guess Handling: When a player guesses a letter correctly, the game should update the partially revealed word to reveal the correctly guessed letters.

2. Word Categories and Customization:

- a. Diverse Word Categories: The game should offer a wide range of word categories, including generic categories (e.g., animals, countries) and specialized categories (e.g., academic subjects, pop culture references).
- b. Category Customization: Players should have the ability to customize word categories, adding or removing categories according to their preferences. They should also have the option to create their own custom word categories.

3. User Interface and Experience:

- a. Intuitive Interface: The game should have a user-friendly interface with clear instructions and easily navigable menus.
- b. Visual Elements: The game should feature visually appealing graphics, animations, and sound effects to enhance the user experience and engagement.
- c. Feedback Mechanisms: The system should provide immediate and clear feedback to the player on the correctness of their guesses, highlighting correct and incorrect letters.

4. Difficulty Levels and Progression:

a. Difficulty Levels: The game should offer different difficulty levels (easy, medium, hard), each with varying word lengths and complexities.

b. Progression System: The game should include a progression system that allows players to unlock higher difficulty levels as they succeed in lower levels.

5. Hint System:

a. Hint Feature: The game should provide a hint system to assist players in guessing the hidden word by offering clues or suggestions.

b. Limited Hints: The hint system should have a limited number of uses to maintain the challenge and balance of the game.

These requirements form the basis for the development of a comprehensive and enjoyable Hangman game that offers diverse word categories, customizable options, engaging gameplay, an appealing user interface, platform compatibility, and effective error handling.

Chapter 4

Feasibility Study

A feasibility study evaluates the practicality and viability of a project. In the case of a Hangman game, here are some key aspects to consider:

1. Technical Feasibility:

- Resources: Assess the availability of the necessary technical resources, such as hardware, software development tools, and programming languages, to develop the Hangman game.
- Expertise: Evaluate the technical skills and expertise required to develop and maintain the game. Determine if the development team possesses the necessary knowledge or if external assistance is needed.
- Compatibility: Ensure the chosen development platform supports the desired features and can handle the game's requirements. Check compatibility with target platforms (desktop, mobile) and browsers.

2. Economic Feasibility:

- Budget: Evaluate the financial resources required for the development, including hardware, software licenses, development tools, artwork, and potential marketing expenses.
- Revenue Model: Identify potential revenue streams for the Hangman game, such as advertisements, in-app purchases, or a premium version. Assess if the projected revenue is sufficient to cover development costs and generate profit.
- Market Analysis: Analyze the target market and competition for Hangman games. Determine if there is sufficient demand and potential market share to make the project economically viable.

3. Time Feasibility:

- **Development Timeframe:** Estimate the time required for development, testing, bug fixing, and deployment of the Hangman game. Consider factors like the complexity of features, team size, and availability of resources.
- **Deadlines:** Identify any project constraints, such as desired launch dates, marketing campaigns, or seasonal considerations, and assess if the development timeline aligns with those deadlines.

4. Legal and Ethical Feasibility:

- **Intellectual Property:** Ensure that the Hangman game does not infringe upon copyrighted material, including word categories or graphics used in the game.
- **Privacy and Data Protection:** Evaluate if the game complies with relevant privacy regulations and collects and handles user data responsibly.
- **Ethical Considerations:** Assess any potential ethical implications related to the game's content or features, ensuring it aligns with ethical standards and values.

5. Operational Feasibility:

- **User Support:** Determine the resources and processes needed to provide user support, address inquiries, and handle potential issues or bug's post-launch.
- **Scalability:** Consider the potential for future updates, additional features, or scalability as the user base grows. Assess if the game's architecture and design allow for easy expansion.

Based on the findings of the feasibility study, decisions can be made regarding the viability of the Hangman game project. This includes determining if the project is technically feasible, economically sustainable, can be completed within the desired timeframe, complies with legal and ethical considerations, and is operationally viable.

Chapter 5

HARDWARE AND SOFTWARE REQUIREMENTS

Hardware and software requirements for a Hangman game can vary depending on the specific platform and development approach. Here are some general hardware and software requirements to consider:

Hardware:

Processor: i5 Generation.

RAM: 8GB.

Hard disk: 1TB.

Software:

Windows 10.

Python 3.8.9.

Visual Studio.

Chapter 6

Implementation Details

Implementation details for a Hangman game can vary depending on the chosen development platform, programming language, and frameworks. However, here are some general implementation considerations for developing a Hangman game:

1. Game Structure and Flow:

- Design the overall structure of the game, including menus, game screens, and transitions between them.
- Implement a main game loop that handles user input, updates the game state, and renders the graphics.
- Define the flow of the game, including starting a new game, displaying the word to guess, processing player guesses, and handling game end conditions (winning or losing).

2. Word Selection and Display:

- Create a list or database of words for each category.
- Implement a mechanism to randomly select a word from the chosen category.
- Display the partially revealed word to the player, with blanks representing unguessed letters. Update the display as the player makes correct guesses.

3. User Input and Validation:

- Implement a mechanism to receive and process user input for letter guesses.
- Validate the input to ensure it meets the game rules (single alphabetical character, not previously guessed).
- Handle invalid inputs and provide appropriate feedback to the player.

4. Game Logic and Progression:

- Implement game logic to track the number of attempts remaining and update the hangman graphic accordingly.
- Update the partially revealed word when the player makes correct guesses, revealing the correctly guessed letters in their respective positions.

5. Difficulty Levels and Progression:

- Define different difficulty levels, each with varying word lengths or complexity.
- Implement mechanisms to switch between difficulty levels and track the player's progress.
- Adjust the number of attempts, word selection, or other factors based on the chosen difficulty level.

6. Hint System:

- Design and implement a hint system to assist players in guessing the hidden word.
- Determine the conditions under which a hint is provided (e.g., after a certain number of incorrect guesses or upon player request).
- Provide hints in the form of clues or suggestions to guide the player.

7. User Interface and Graphics:

- Design and implement the user interface elements, including menus, buttons, and game screens.
- Create visually appealing graphics, such as the hangman graphic, partially revealed word display, and any additional visual elements to enhance the user experience.
- Implement animations and transitions to provide smooth and engaging user interactions.

These implementation details provide a general guide for developing a Hangman game. The specific implementation steps and considerations may vary based on the chosen development platform, programming language, and frameworks. It is recommended to refer to the documentation and resources provided by the selected tools for detailed implementation guidelines.

Libraries and packages

Pygame: A popular library for developing 2D games in Python, providing essential game development functionalities.

tkinter or PyQt: Libraries for building the graphical user interface and handling user input.

Pytest or unittest: Testing frameworks for unit testing Python code.

Pygame. Mixer: A module for playing sound effects and background music in the game.

Datetime: This library provides us the actual date and time.

Wikipedia: It is a python module for searching anything on Wikipedia.

Smtplib: Simple mail transfer protocol that allows us to send mails and route mails between mail servers.

Pyjokes: It is a python library which contents lots of interesting jokes in it.

Web browser: It provides interface for displaying web based document to user.

Pyautogui: It is a python library for graphical user interface.

Os: It is representing operating system related functionality.

Sys: It allows operating on the interpreter as it provides access to the variable and function that usually interact strongly with the interpreter.

Functionality:

In this we check the functionality of the system whether the system performs the task which it was intend to do. To check functionality each function was checked and run, it is able to execute the required task correctly then the system passes in the functionality tests.

Chapter 7

FUTURE SCOPE

1. Implement multiplayer capabilities, allowing players to compete against each other in real-time or take turns guessing words.
2. Introduce online leaderboards to track and display high scores or fastest completion times, fostering competition among players.
3. Enable social features such as sharing game progress, inviting friends to play, or challenging others through social media integration.
4. Add more customization options, such as choosing different themes, backgrounds, or difficulty levels, to cater to individual player preferences.

Chapter 8

CONCLUSION

In conclusion, the Hangman game is a classic and popular word-guessing game that offers entertainment and challenges players' vocabulary skills. Throughout the development process, various aspects need to be considered, such as the objective, present system, proposed system, problem definition, requirement analysis, feasibility study, hardware and software requirements, implementation details, and functionalities.

By leveraging appropriate libraries and packages based on the chosen programming language and development platform, developers can streamline the development process and enhance the game's functionality. The Hangman game requires features such as word selection, guessing mechanism, word display, attempt tracking, correct guess handling, end conditions, game management, hint system, difficulty levels, user interface, and customization.

REFERENCES

1. Shibwabo, B. K. Omyonga, K.(2015).The application of real-time voice recognition to control mobile operations.
2. IOSR Journal of Engineering Mar.2012, Vol. 2(3) pp: 420-423 ISSN: 2250-3021 www.iosrjen.org 420.
“Android
Speech to Text Converter for SMS Application” Ms.
AnujaJadhav* Prof. Arvind patil.
3. Anwani, R., Santuramani, U., Raina, D. RL, P.
Vmail: voice Based Email Application. International
Journal of Computer Science and Information
Technologies, Vol. 6(3),2015