06/11/2023

Game Project

The hangman

**Project Proposal**

Object Oriented Programming

A picture containing text, symbol, emblem, logo

Description automatically generated

* **Project Title:**

**THE HANGMAN**

* **Group Members:**
* **Muhammad Noman (2022-SE-31)**
* **Tayyaba Tariq (2022-SE-01)**

\

## Figure 1 Main Page

* **Project Title:**

**THE HANGMAN**

* **Group Members:**
* **Muhammad Noman (2022-SE-31)**
* **Tayyaba Tariq (2022-SE-01)**

\

**Table of Contents**

[0](file:///D:\1.STUDY\Hangman%20Proposal.docx#_Toc137767622)

[ **Introduction:** 3](#_Toc137767623)

[ **Features:** 3](#_Toc137767624)

[o **Random Word Selection** 3](#_Toc137767625)

[o **User Input:** 3](#_Toc137767626)

[o **Word Display:** 3](#_Toc137767627)

[o **Guess Count:** 3](#_Toc137767628)

[o **Visual Feedback:** 3](#_Toc137767629)

[o **Win/Loss Condition:** 3](#_Toc137767630)

[ **Implementation Steps:** 3](#_Toc137767631)

[ **Timeline:** 3](#_Toc137767632)

[o **Week 1:** 3](#_Toc137767633)

[o **Week 2:** 3](#_Toc137767634)

[o **Week 3:** 3](#_Toc137767635)

[o **Week 4:** 3](#_Toc137767636)

[ **User Interface:** 4](#_Toc137767637)

[ **Classes and their data members and member functions** 5](#_Toc137767638)

[o **Data Members of class hangman:** 5](#_Toc137767639)

[ **word\_length:** 5](#_Toc137767640)

[ **word\_array:** 5](#_Toc137767641)

[ **wrong\_check:** 5](#_Toc137767642)

[ **win\_check:** 5](#_Toc137767643)

[ **flags:** 5](#_Toc137767644)

[o **Member Functions class hangman:** 5](#_Toc137767645)

[ **display\_hangman():** 5](#_Toc137767646)

[ **give\_words():** 5](#_Toc137767647)

[ **display\_word():** 5](#_Toc137767648)

[ **check():** 5](#_Toc137767649)

[ **display\_win():** 5](#_Toc137767650)

[ **admin\_panel():** 5](#_Toc137767651)

[ **Conclusion:** 5](#_Toc137767652)

**Table of Figures**

[Figure 1 Main Page 4](file:///D:\1.STUDY\SECOND%20SEMESTER\OOP\Hangman%20(Game%20Project)\Hangman%20Proposal.docx#_Toc137807604)

[Figure 2 Guessing the Word 4](file:///D:\1.STUDY\SECOND%20SEMESTER\OOP\Hangman%20(Game%20Project)\Hangman%20Proposal.docx#_Toc137807605)

[Figure 3 Win and Lose Interface 5](file:///D:\1.STUDY\SECOND%20SEMESTER\OOP\Hangman%20(Game%20Project)\Hangman%20Proposal.docx#_Toc137807606)

# **Introduction:**

We propose the development of a Hangman game in C++. Hangman is a classic word-guessing game where players attempt to guess a hidden word by suggesting letters. The game will provide an entertaining and interactive experience for players while helping to enhance their vocabulary and word-guessing skills.

# **Features:**

* **Random Word Selection**: The game will select a random word from a predefined list of words.
* **User Input:** Players will be able to input their guesses by entering letters one at a time.
* Letter Validation: The game will validate the input to ensure it is a single alphabetical character.
* **Word Display:** The game will display the partially revealed word with correctly guessed letters in their respective positions and placeholder characters for unknown letters.
* **Guess Count:** The game will keep track of the number of incorrect guesses made by the player.
* **Visual Feedback:** The game will provide visual feedback to represent the hangman's gallows and the incorrect guesses made.
* **Win/Loss Condition:** The game will determine if the player has won by guessing the word correctly or lost by exceeding the maximum number of allowed incorrect guesses.

# **Implementation Steps:**

* Design the main game loop and initialize necessary variables.
* Create a function to randomly select a word from a predefined list.
* Implement input validation to ensure the user's guess is a single alphabetical character.
* Update the partially revealed word based on the correct guesses.
* Implement the visual representation of the hangman's gallows and incorrect guesses.
* Define win and loss conditions and display appropriate messages.
* Provide an option to play again or exit the game.

# **Timeline:**

* **Week 1:** Designing the game structure, implementing the word selection functionality, and develop the user input validation.
* **Week 2:** Building the word display logic, implementing the visual feedback, and handle win/loss conditions.
* **Week 3:** Refining the user interface and performing comprehensive testing.
* **Week 4:** Finalizing the game, reviewing the code for optimizations, and preparing for the release.
* **User Interface:**

A screenshot of a video game

Description automatically generated with medium confidenceIt is a single player game. The user interface provides the visual representation of the hangman and choice to login as admin to add, remove or display the words been there to guess as a player. The user interface will include prompts for user input, a visual representation of the hangman's gallows, the partially revealed word, and feedback on incorrect guesses.

Figure 2 Main Page

A picture containing text, screenshot, font

Description automatically generated

Figure 3 Guessing the Word

Figure 4 Win and Lose Interface

# **Classes and their data members and member functions**

## **Data Members of class hangman:**

* **word\_length:** To get length of the word to be guessed.
* **word\_array:** To store the word which is randomly picked from file.
* **wrong\_check:** To have track of wrong inputs.
* **win\_check:** To have track of correct inputs.
* **flags:** Flags being used to draw/display the hangman by making flags true or false by having track of correct or wrong inputs.

## **Member Functions class hangman:**

* **display\_hangman():** To display the creation of gallows of hangman when user enters a wrong input.
* **give\_word():** To fetch a random word from file to be guessed.
* **display\_word():** Function to display word with dashes which the player has to guess.
* **check():** Function to check for correct or wrong input by player,
* **display\_win():** If all inputs are correct from player win interface of the hangman will display.
* **admin\_panel():** Through admin panel, admin can add, remove, or display the words being saved.

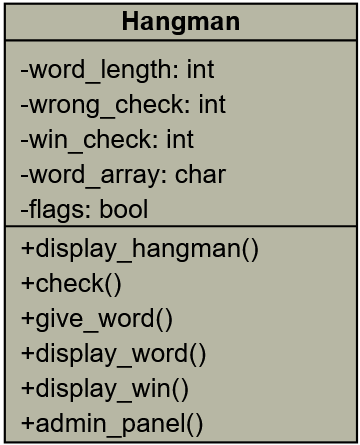
****

Figure 4 class diagram

* **Conclusion:**

The proposed Hangman game will provide an engaging and challenging experience for players, allowing them to enhance their vocabulary while having fun. The project will be implemented in C++, providing an opportunity to showcase programming skills and logical problem-solving abilities.

We believe that the Hangman game will be well-received and enjoyable for players of all ages. It will serve as a great opportunity to delve into the world of game development using C++.