

Programming Project / C++ – Hangman Game – Word Guessing

Student 1: Bacalete Benjamin

Student 2: Bene Cristian

I. Task Description

Student 1 is responsible for game logic:

- Loading and storing words from file
- Choosing a random word
- Checking player guesses
- Updating word display with correct letters
- Tracking number of incorrect attempts

Student 2 is responsible for player interaction:

- Reading and validating player input
- Showing current state of guessed word
- Drawing hangman figure based on lives
- Handling game restart and end conditions

II. Data Structures Used by the Team

The following classes will be used:

- `Word`: string actualWord, string guessedWord, vector<char> usedLetters
- `Hangman`: int lives, int maxLives, Word word

III. File Structure

The following files will be used:

words.txt

A file with a list of words used for guessing:

<word1>

<word2>

<word3>

...

For example:

computer

banana

mystery

hangman

holiday

scores.txt

A file to optionally track high scores (name, word, attempts):

<name> <word> <attempts>

...

IV. Interacting with Executables

Application 1 will offer the following options (Game Logic):

`./hangman_logic.exe load_words`

→ Loads and verifies the word list

`./hangman_logic.exe choose_word`

→ Picks a random word for the current game

`./hangman_logic.exe check_guess <letter>`

→ Checks a guessed letter and updates guessed word

Application 2 will offer the following options (Player Interaction):

`./hangman_ui.exe start`

→ Starts a new game

`./hangman_ui.exe guess <letter>`

→ Sends a guess to the logic app

`./hangman_ui.exe view_state`

→ Displays the current word, lives left, and used letters

`./hangman_ui.exe restart`

→ Starts a new game after win/loss

V. Optional Features

- Levels: Easy, Medium, Hard (word length or lives)
- Scoreboard from scores.txt
- Word hint or category support (e.g., Animals, Tech)
- Timer (bonus challenge)