Hangman

How to Play

Objective
Guess the hidden word by suggesting letters within a certain number of attempts. The game ends when you either guess the word correctly or run out of lives.

Getting Started
Start: Launch the app to see the game interface.
• Controls:
- Guess a Letter: Click a letter on the virtual keyboard to make a guess.
- Restart Game: Click the "Restart" button to start a new game with a new word.
- Quit: Click the "Quit" button to exit the application.

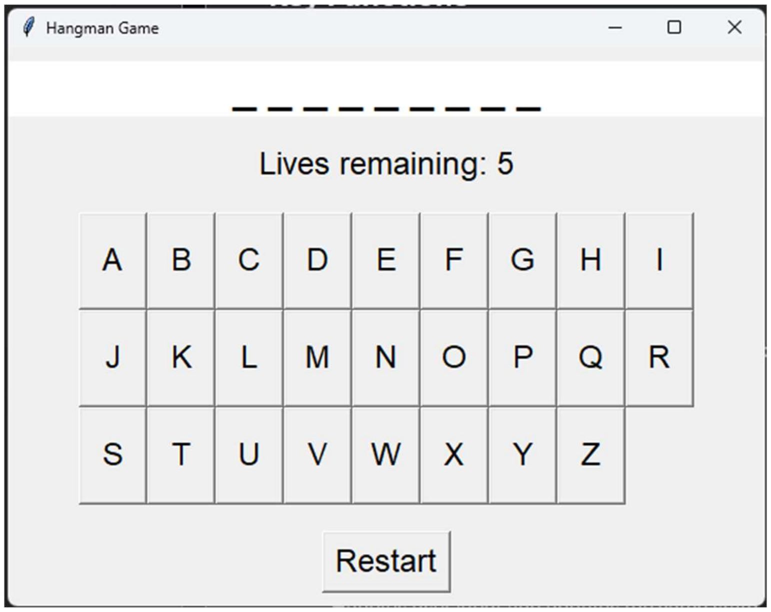
Gameplay
• Making a Guess: Click a letter on the virtual keyboard to guess it.
• Winning: Guess all letters of the word before running out of lives.
• Losing: If you use up all your lives without guessing the word, you lose.
• New Game: The game automatically restarts after each round, with a new word.
• Ending Game: Click "Quit" to exit the game anytime.

Enjoy playing Hangman!

Hangman Demo Link:
https://drive.google.com/file/d/1f6Ey9Cg0i-GjQvzxUDokKuyJyC4q8k97/view?usp=sharing

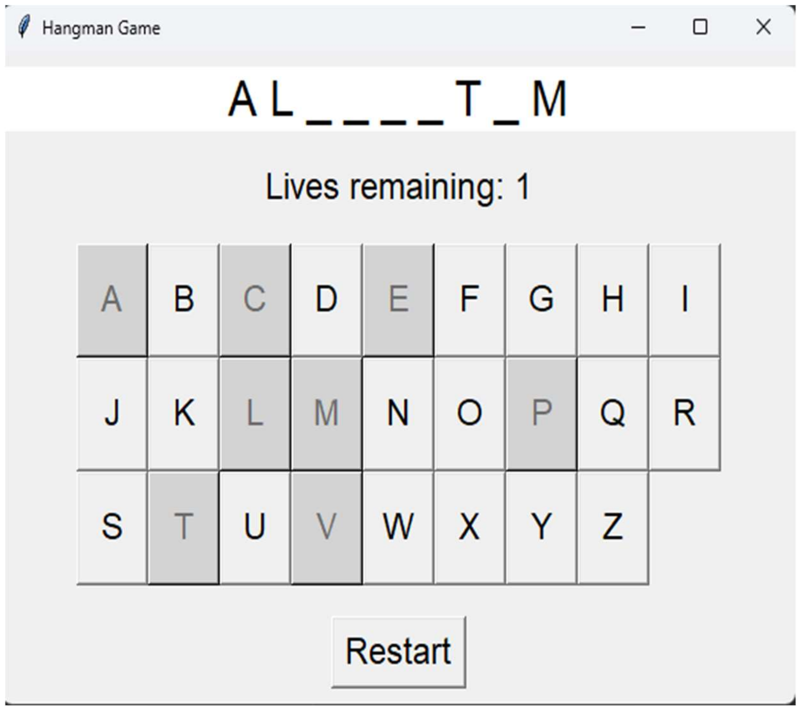
Screenshots

1. Initial Game Screen



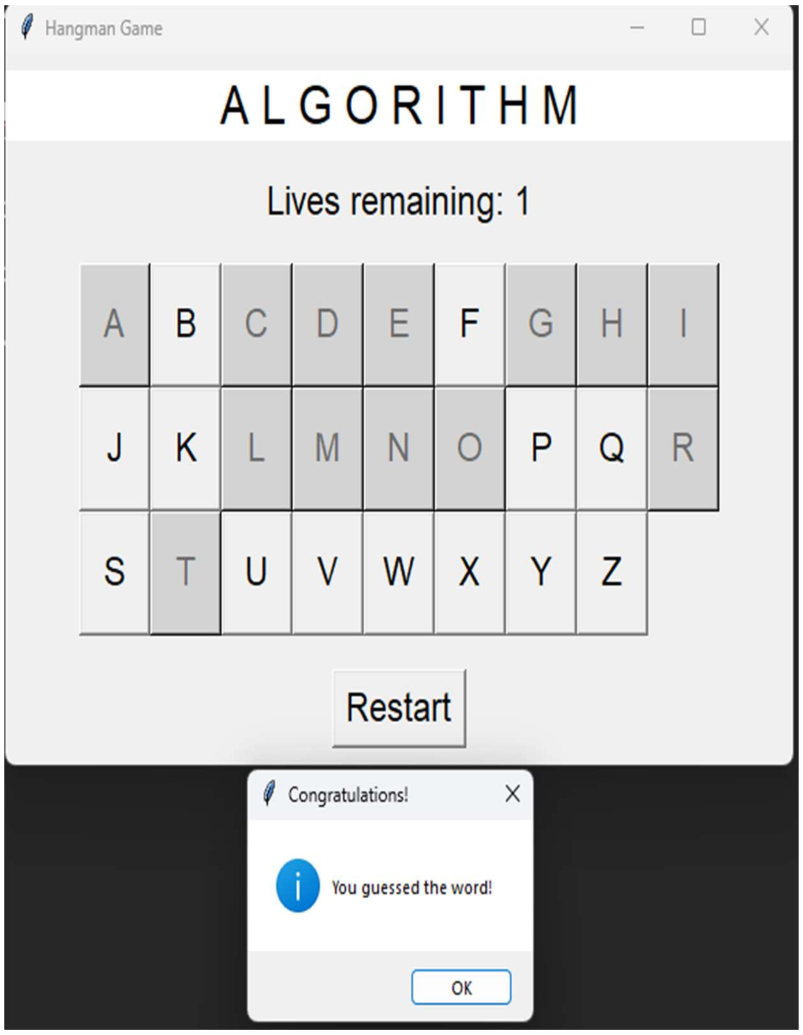
○ Description: Displays the game board with placeholders for the word, number of remaining lives, and the virtual keyboard.

2. Game in Progress



- Description*:* Shows the board with the current state of the word and the player's guesses. The virtual keyboard shows which letters have been used.

3. Game Ended



- Description*:* Displays the result (win or lose), the guessed word, and the option to restart or quit.

Key Functions

HangmanGame Class:

1. reset_game()
   Initializes or resets the game state with a new word, empty guesses, and full lives.
2. make_guess(letter)
   Processes a player's guess, updates the game state, and checks for a win or loss.
3. get_word_display()
   Returns the current display of the word with guessed letters and underscores.
4. get_lives()
   Returns the number of lives remaining.
5. is_game_over()
   Checks if the game is over (win or loss).
6. get_log()
   Retrieves the log of game events (guesses, lives, game status).

GLogger Class:

1. log_entry(entry)
   Appends a log entry to the game log file.
2. start_new_game(word)
   Logs the start of a new game, including the word to be guessed.
3. log_guess(letter, lives_remaining)
   Logs a player's guess and the number of remaining lives.
4. log_restart()
   Logs when the game is restarted.
5. log_game_over(word_was_guessed)
   Logs the outcome of the game (win or loss).

GUI Class:

1. create_widgets()
   Creates and places all widgets in the main window (word display, lives, keyboard, restart button).
2. create_keyboard()
   Creates buttons for each letter of the alphabet and places them in a grid.
3. process_guess(letter)
   Handles a letter guess from the player, updates the game state, and logs the guess.
4. update_ui()
   Updates the UI to reflect the current game state.
5. update_keyboard()
   Updates the virtual keyboard buttons to reflect guessed letters.
6. handle_end_of_game()
   Handles the end of the game, shows a message box with the result, and logs the outcome.
7. restart_game()
   Restarts the game by resetting the game state and logging the restart.
8. save_game_log()
   Placeholder method for saving game logs. (The GameLogger class handles logging.)