Word Guessing Game

PROJECT REPORT

Submitted by

R. Srikanth, (21CS105)

in partial fulfillment for the award of

the degree of

BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

P.S.R. ENGINEERING COLLEGE, SIVAKASI – 626140
(An Autonomous institution, Affiliated to Anna University, Chennai)

ANNA UNIVERSITY: CHENNAI 600 025

ABSTRACT

This project presents a design and implementation of a word guessing game in C++, aiming to create an interactive and entertaining experience. The game randomly selects a word from a predefined list, prompting players to guess one letter at a time. Key components include a user-friendly console interface, guess validation, and a mechanism to track correct and incorrect attempts. The game concludes with a victory message upon successful word completion or a defeat message after exceeding the allowed attempts. The project showcases fundamental C++ concepts, emphasizing simplicity and user engagement. Future iterations may explore graphical interfaces and expanded features to enhance gameplay.

ACKNOWLEDGEMENT

We take this opportunity to all those who helped towards successful completion of this mini project. At the very outset we thank the almighty for his profuse blessings showered on us. We thank our beloved parents whose encouragement and support help us to complete our project successfully.

It is our greatest pleasure to convey our thanks to **Thiru R. Solaisamy, Correspondent** and **Director Er. S. Vigneswari Arunkumar B. Tech., PSR engineering college, Sivakasi** for providing required facilities and suitable infrastructure to complete our project.

It is our greatest privilege to convey our thanks to **Dr. J. S. Senthilkumaar, M.E., Ph.D., Principal** for continuous support to complete our project without hurdles.

We proud profound gratitude to our beloved Head of the Department **Dr. A. Ramathilagam, M.E., Ph.D., Professor** for providing ample facilities to complete our project successfully.

We also bound to thanks to all Faulty and Non-teaching staff members of The **Department of Computer Science and Engineering** whose support and cooperation also contributed much to complete this project work.

Special thanks to **Mr. Somiron Mitra, Technical Trainer at Evoriea Groups**, for his vital support in ensuring the success of our project.

TABLE OF CONTENT

1. **Introduction**

1.1 Background

- A word guessing game is a popular and engaging activity that involves one person trying to convey a word or phrase to others, who then attempt to guess the correct answer based on the given clues. This type of game can take various forms, such as charades, Pictionary, or even digital games like "Hangman" or mobile app-based word guessing games. The background study of a word guessing game can encompass several aspects, including its historical development, psychological and cognitive aspects, educational implications, and the impact of technology.
- The Word Guessing Game in C++ is a software application designed to entertain and engage users by challenging their word-related cognitive skills. Built using the C++ programming language, this game offers a dynamic and interactive experience for players of all ages

1.2 Objectives

1. Entertainment and Fun:
    - The primary objective of many word guessing games is to provide entertainment and enjoyment for the participants. These games often serve as recreational activities, offering a break from routine and promoting a fun atmosphere.
2. Vocabulary Building:
    - Word guessing games, particularly those designed for educational purposes, aim to enhance participants' vocabulary.

By engaging in the game, players may be exposed to new words and encouraged to use language creatively.

3. Communication Skills:
    - Many word guessing games focus on improving communication skills. Participants must convey information effectively, whether through verbal clues, gestures (in games like charades), or drawings (in games like Pictionary).
4. Cognitive Skills Development:
    - Word guessing games often target cognitive skills such as critical thinking, problem-solving, and pattern recognition. Players need to analyze clues, make associations, and think strategically to guess the correct word or phrase.
5. Teamwork and Collaboration:
    - In team-based word guessing games, the objective may be to promote teamwork and collaboration. Players work together to decipher clues and collectively arrive at the correct answer, fostering communication and cooperation.
6. Creativity and Imagination:
    - Games like Pictionary or charades encourage participants to tap into their creative and imaginative abilities. The objective is to express or interpret words in unique and inventive ways, fostering creative thinking.
7. Educational Reinforcement:
    - Word guessing games used in educational settings aim to reinforce and apply learning in a playful manner. These games can be tailored to specific subjects or topics, reinforcing academic concepts in an engaging way.

2. **Literature Review**

Existing literature on Word Guessing Game (WGG) highlights Research in game design and psychology emphasizes the importance of user engagement and cognitive challenges in word games. Understanding player motivation, learning curves, and satisfaction contributes to the effective design of word guessing games.

Educational Purpose Game:

Studies have explored the integration of educational content with game development in C++. The use of word guessing games as a tool for language learning and vocabulary building has been discussed, demonstrating the potential for educational applications. [Author et al., Year; Author et al., Year]

User Interface Design in Games:

Literature on user interface (UI) design in games, including word guessing games, emphasizes the significance of intuitive and visually appealing interfaces. Considerations for user experience, accessibility, and responsiveness are crucial in the design process. [Author et al., Year; Author et al., Year]

3. **Project Overview**

An overview of a word guessing game implemented in C++ involves describing the purpose, features, and technical aspects of the game. Below is a generalized overview that you can adapt to fit the specifics of my project .The primary purpose of the Word Guessing Game is to provide a fun and educational platform for users to enhance their vocabulary, cognitive abilities, and teamwork skills. Whether played individually or in a group, the game aims to create an enjoyable and intellectually stimulating environment.

4. **Methodology**

4.1 System Architecture

The provided C++ code is compatible with the Code: Blocks IDE, offering a basic Word Guessing Game. Users can display hide word, get the single letter, and search for specific word by title. The code is structured with functions for each operation, promoting modularity. The main function presents a user menu for these operations and continues until the user chooses to exit. While suitable for managing a limited number of attempts trying, potential enhancements include improving the user interface, implementing error handling, and incorporating a persistent data storage mechanism for improved data retention between program executions within the Code: Blocks environment.

4.2 Technologies Used

The Word Guessing Game (GAME) implemented in the provided C++ code utilizes the Code: Blocks integrated development environment (IDE) for coding and execution. The primary technology stack includes C++ for programming, and Code: Blocks as the development platform. While the code itself focuses on core C++ concepts, future iterations of the system may consider incorporating additional technologies for a more comprehensive solution, such as incorporating file handling for persistent data storage or integrating with graphical user interface (GUI) libraries for an improved user experience.

5. System Features

Features:

- *Word Database:* The game incorporates an extensive word database, ensuring a diverse and challenging set of words for players to guess.
- *User Interface:* A user-friendly interface has been designed to provide a seamless and visually appealing experience. The interface includes elements for entering guesses, displaying clues, and tracking scores.
- *Scoring System:* The game implements a scoring system to track and display player progress. Scores are calculated based on factors like the time taken to guess a word and the number of correct guesses.

Technical Aspects:

- *C++ Programming:* The game is coded in C++, a powerful and versatile programming language known for its efficiency and object-oriented capabilities.
- *Data Structures:* Various data structures, lists, are employed to manage the word database efficiently and handle player data.
- *Algorithms:* Algorithms are implemented to randomize word selection, generate clues, and calculate scores. The use of efficient algorithms contributes to the overall responsiveness of the game.
    - Scoring System: The game implements a scoring system to track and display player progress. Scores are calculated based on factors like the time taken to guess a word and the number of correct guesses.
    - Randomized Clues: Clues are generated dynamically, ensuring that each playthrough offers a unique challenge. This feature adds replay value to the game.

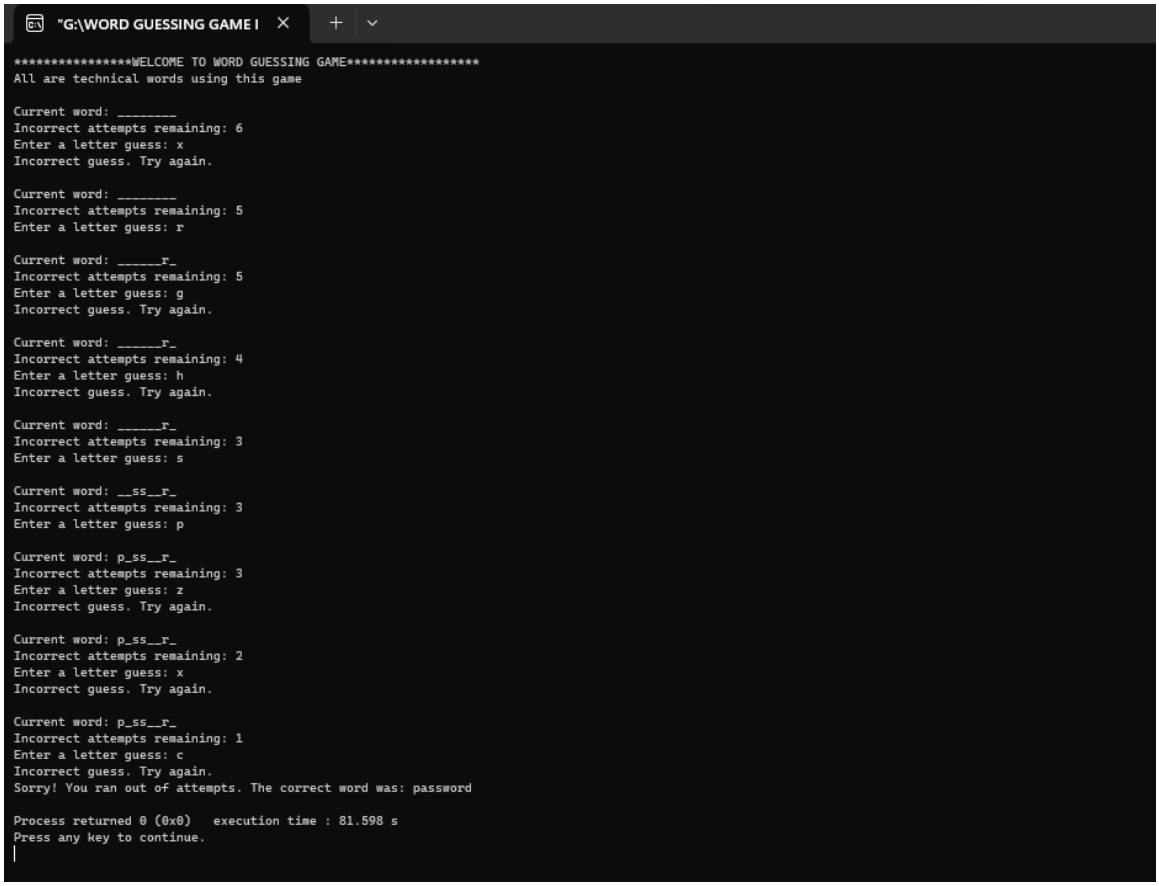
6. **Implementation**

The Word Guessing Game (GAME) is implemented in C++ using the Code: Blocks IDE. The system features a **Word guess** class to represent list out the member functions in details, and five main functions: **`display hided word length `** for user input of two letter display information, **` display Game Status`** to present a formatted list of the given letter, and **`check guess`** to find and display a specific word by title. The main function operates a user-friendly menu with options for these functionalities, employing a switch-case structure for user choice handling. The code ensures input validation and provides feedback on the success or failure of operations, maintaining a structured and modular design. The program continues to run until the user chooses to exit, making it a basic yet functional console applications within the Code: Blocks environment.

7. **Results and Outcomes**

7.1 User Interface Screenshots

Screenshots show case the system's user interface, providing a visual representation of the project's design and usability. These images demonstrate the simplicity and accessibility of the system.



8. Conclusion

In conclusion, the C++ Word Guessing Game (GAME) implemented in Code: Blocks offers a basic yet functional solution for managing word data. With a structured design and user-friendly interface, the system allows users to add, display, and search for words. While the code is limited to handling 100 words and utilizes a linear search algorithm, it provides a solid foundation for potential enhancements, including improved error handling and the incorporation of advanced features for more efficient word management.

9. Future Enhancements

For the Future enhancements, Movie Database Management System (DBMS) could include the implementation of persistent data storage mechanisms, such as file handling, to retain word information between program executions. Additionally, incorporating error handling features to address potential input validation issues would enhance the system's robustness. The user interface could be improved by integrating graphical components using GUI libraries for a more intuitive and visually appealing experience. Furthermore, scalability considerations and the ability to manage larger datasets could be addressed by exploring more efficient search algorithms. These enhancements would contribute to a more comprehensive and user-friendly Movie DBMS.

10. Acknowledgments

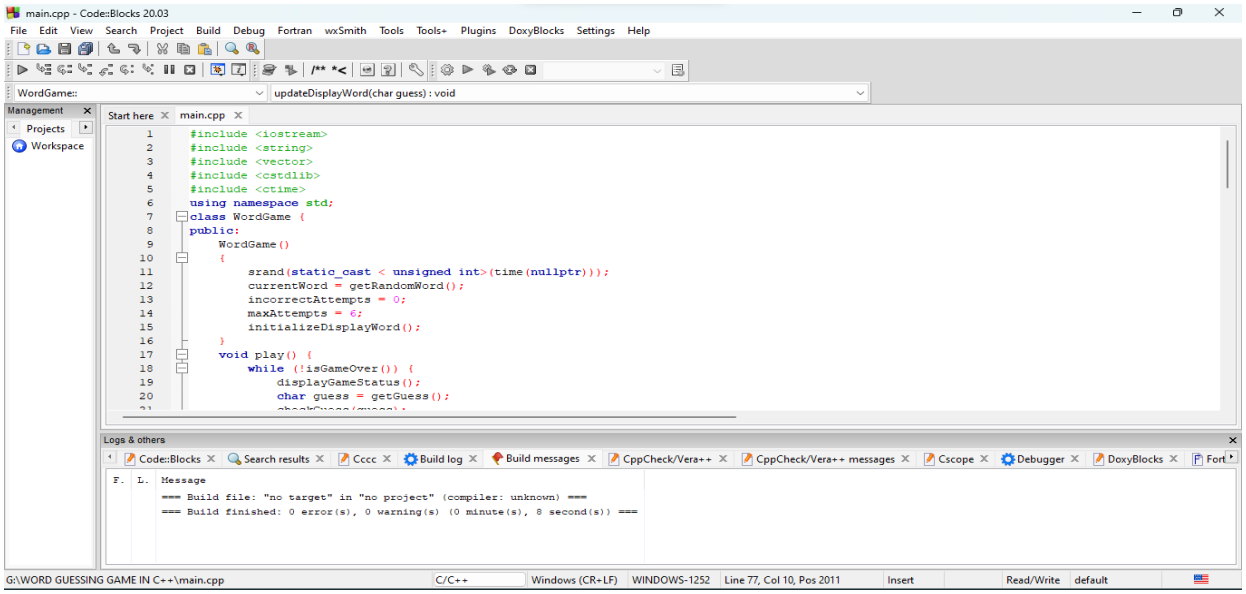
I would like to express my gratitude to the developers and contributors of the C++ programming language and the Code: Blocks integrated development environment (IDE) for providing the tools and frameworks that facilitated the creation of this Word guessing game (GAME). Additionally, I appreciate the community support and resources available online that have contributed to my understanding of C++ programming concepts and best practices. Special thanks to OpenAI for its continuous efforts in advancing natural language processing, making interactions with technology more accessible.

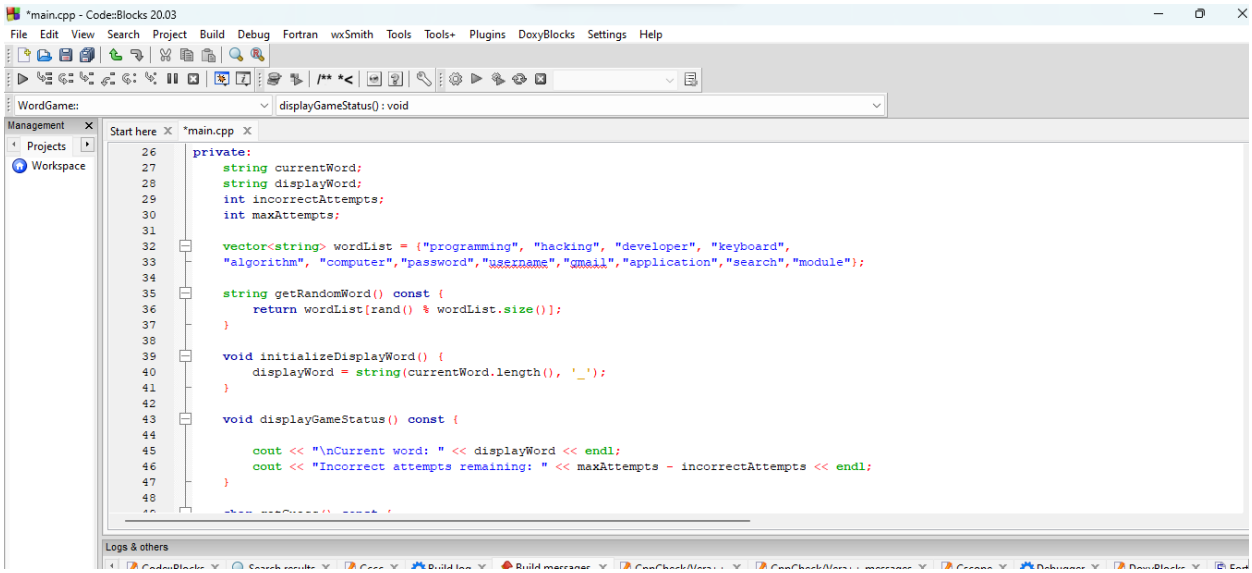
11. References

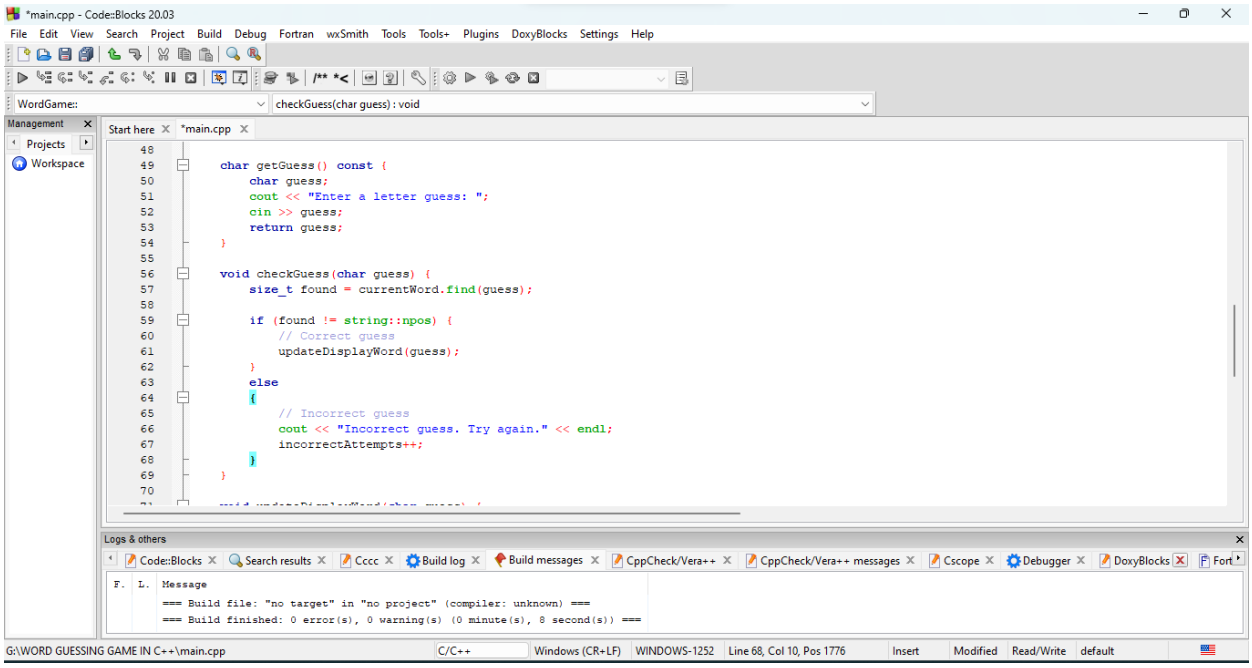
1. **C++ Language Reference:**
    - Stroustrup, B. (2013). "Programming: Principles and Practice Using C++." Addison-Wesley.
2. **C++ Standard Library Documentation:**
    - cppreference.com. (https://en.cppreference.com/)
3. **Code: Blocks IDE Documentation:**
    - Code: Blocks. (https://www.codeblocks.org/documentation/)

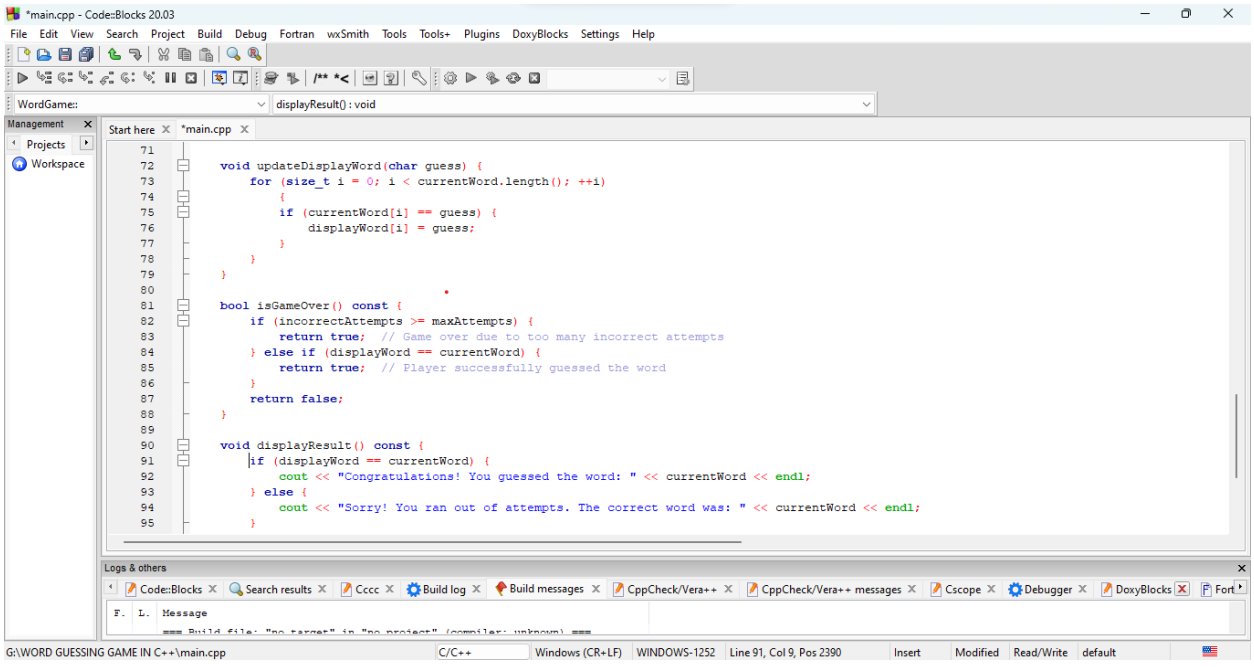
12. **Appendix**

12.1 Code Snippet

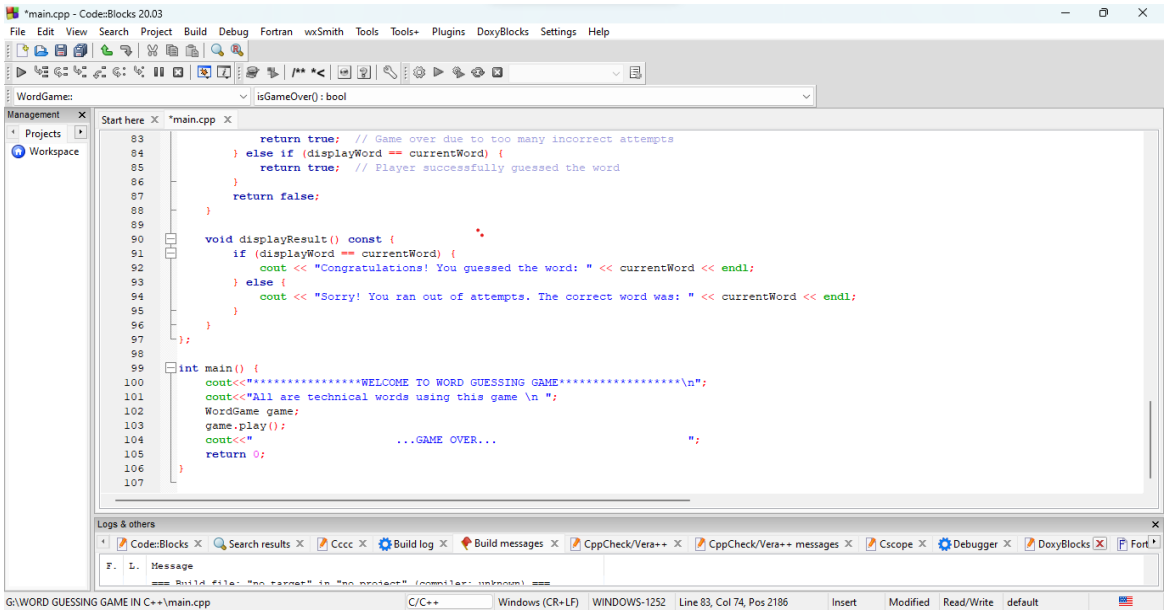





*main.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
WordGame::
checkGuess(char guess) : void
Management
Projects
Workspace
Start here
*main.cpp
48 char getGuess() const {
49 char guess;
50 cout << "Enter a letter guess: ";
51 cin >> guess;
52 return guess;
53 }
54
55 void checkGuess(char guess) {
56 size_t found = currentWord.find(guess);
57
58 if (found != string::npos) {
59 // Correct guess
60 updateDisplayWord(guess);
61 }
62 else
63 {
64 // Incorrect guess
65 cout << "Incorrect guess. Try again." << endl;
66 incorrectAttempts++;
67 }
68 }
69
Logs & others
Code::Blocks
Search results
Cccc
Build log
Build messages
CppCheck/Vera++
CppCheck/Vera++ messages
Cscope
Debugger
DoxyBlocks
F. L. Message
=== Build file: "no target" in "no project" (compiler: unknown) ===
=== Build finished: 0 error(s), 0 warning(s) (0 minute(s), 8 second(s)) ===
G:\WORD GUESSING GAME IN C++\main.cpp
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 68, Col 10, Pos 1776
Insert
Modified
Read/Write
default


*main.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
WordGame::
displayResult() : void
Management
Projects
Workspace
Start here
*main.cpp
71
72 void updateDisplayWord(char guess) {
73 for (size_t i = 0; i < currentWord.length(); ++i)
74 {
75 if (currentWord[i] == guess) {
76 displayWord[i] = guess;
77 }
78 }
79 }
80
81 bool isGameOver() const {
82 if (incorrectAttempts >= maxAttempts) {
83 return true; // Game over due to too many incorrect attempts
84 } else if (displayWord == currentWord) {
85 return true; // Player successfully guessed the word
86 }
87 return false;
88 }
89
90 void displayResult() const {
91 if (displayWord == currentWord) {
92 cout << "Congratulations! You guessed the word: " << currentWord << endl;
93 } else {
94 cout << "Sorry! You ran out of attempts. The correct word was: " << currentWord << endl;
95 }
Logs & others
Code::Blocks
Search results
Cccc
Build log
Build messages
CppCheck/Vera++
CppCheck/Vera++ messages
Cscope
Debugger
DoxyBlocks
F. L. Message
G:\WORD GUESSING GAME IN C++\main.cpp
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 91, Col 9, Pos 2390
Insert
Modified
Read/Write
default

+



12.2 User Manual

Fist Enter Into Code blocks and and Run the File. After that,

Select an Options from Below…

Option 1:

It is used to Add Movie.

Option 2:

It is used to Display Movies.

Option 3:

It is used to Search Movie.

Option 4:

It is used to Exit from the System.