

**MINI PROJECT REPORT ON**

**WORDLE**

**B. TECH IN ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING**





**KARTHICK SRIRAM 22AM241 GURU VEDESH 22AM136**

**GODWIN SAM PAUL 22AM130**

# Abstract

This abstract presents the implementation of Wordle, a popular word-guessing game, using the Java programming language. The objective was to develop a console-based version of Wordle where players guess a hidden word by submitting guesses and receiving feedback. The implementation involved generating a random word from a dictionary, validating user input, comparing guesses, and providing feedback on correctness. The game utilizes arrays, strings, loops, and conditional statements to manipulate data and make logical decisions. The Java implementation showcases the language's efficiency and simplicity, providing players with an engaging word-guessing experience without graphical dependencies. It serves as an example of using fundamental programming concepts to create an interactive and entertaining game while enhancing vocabulary and logical thinking skills.

Table of Contents:

|  |  |  |
| --- | --- | --- |
| **S. No** | **Title** | **Page Number** |
| **1** | **1.1** Introduction | 4 |
| **1.2** Scope of the work | 4 |
| **1.3** Purpose of the wordle game | 4 |
| **2** | Implementation | 5 |
| **3** | Design | 9 |
| **3.1** Hardware and Software Requirements | 9 |
| **4** | **4.1** Testing Result Analysis | 10 |
| **4.2** Test Cases | 10 |
| **4.3** Sample Input and Output | 11 |
| **4.4** Analysis of Results | 13 |
| **5** | Conclusion | 14 |
| **6** | References | 15 |

# Introduction:

In this project, we present the development of a Wordle game implemented in the Java programming language. Wordle is a popular word-guessing game where players attempt to guess a hidden word by submitting their guesses and receiving feedback. The objective of this implementation is to create a console-based version of Wordle, providing an interactive and challenging gaming experience for players. By leveraging the power and simplicity of the Java language, we aim to develop an efficient and lightweight implementation that allows users to enjoy the game across different platforms without the need for graphical interfaces. Through the use of arrays, strings, loops, and conditional statements, we will design an algorithm that generates random words, validates user input, compares guesses, and provides feedback on the correctness of the guesses. By completing this project, we aim to showcase the versatility of the Java language and demonstrate the application of fundamental programming concepts in creating an engaging and entertaining word-guessing game.

# Scope of the Work:

The scope of implementing Wordle in Java involves designing and developing a console-based game that allows players to guess a hidden word by submitting their guesses. The program will generate a random word from a predefined dictionary and provide feedback on the correctness of each guess. The implementation will include handling user input, validating guesses, comparing letters and positions, and determining game outcomes.

# Purpose Of the Wordle Game:

The purpose of the Wordle game is to provide an entertaining and engaging word-guessing experience for players. It offers a challenging yet accessible gameplay format that tests players' vocabulary, deductive reasoning, and problem-solving skills. The game aims to stimulate mental agility and linguistic abilities by encouraging players to think critically and strategically to uncover the hidden word within a limited number of guesses. Additionally, Wordle can serve as a source of enjoyment and relaxation, offering a fun diversion from daily routines and providing an opportunity for players to unwind and engage in a stimulating word-based activity.

# Implementation:

The code for the program is given below:

import java.util.Random;

import java.util.Scanner;

public class WordleGame {

private static final String[] WORDS = { "apple", "beach", "chair", "dance", eagle", "field", "ghost", "heart", "igloo", "joker", "knife", "lemon", "music", "night", "ocean", "piano", "queen", "river", "shark", "tiger", "uncle", "virus", "whale", "xerox", "yacht", "zebra", "alarm", "blink", "cloud", "dream", "fairy", "grape", "honey", "ivory", "jolly", "kayak", "lucky", "mango", "nudge", "orbit", "peach", "quick", "storm", "table", "unity", "venom", "wagon", "xenon", "yodel", "zesty", "amber", "blaze", "chime", ”ember", "flame", "glide", "hyena", "icing", "jewel", "koala", "maple", "novel", "olive", "quirk", "roset", "snail", "tulip", "unity", "waltz", "xenon", "zebra", "alloy", "black", "charm", "delta", "eagle", "flora", "grail", "happy", "ivory", "jolly", "karma", "lemon", "magic", "nexus", "opera", "prism", "quick", "raven", "smile", "tiger", "unity", "vocal", "waltz" };

private static final int MAX\_ATTEMPTS = 6;

public static void main(String[] args) {

Random random = new Random();

String hiddenWord = WORDS[random.nextInt(WORDS.length)];

char[] wordToGuess = hiddenWord.toCharArray();

boolean[] guessed = new boolean[5];

System.out.println("Welcome to Wordle!");

System.out.println("You have 6 attempts to guess a 5-letter word.");

System.out.println("If the letters are in correct position then it is in \033[0;32m green \033[0m ");

System.out

.println("If the letters matches\033[0;33m yellow \033[0m if the letter matches with the final word");

System.out.println("Good luck!");

Scanner scanner = new Scanner(System.in);

int attempts = 0;

while (attempts < MAX\_ATTEMPTS) {

System.out.println("Attempt " + (attempts + 1));

System.out.print("Enter your guess (5 letters): ");

String guess = scanner.nextLine();

if (guess.length() != 5) {

System.out.println("Please enter a 5-letter word.");

continue;

}

char[] guessedWord = guess.toCharArray();

boolean[] correctLetters = new boolean[5];

for (int i = 0; i < 5; i++) {

if (wordToGuess[i] == guessedWord[i]) {

correctLetters[i] = true;

guessed[i] = true;

}

}

for (int i = 0; i < 5; i++) {

if (correctLetters[i]) {

System.out.print("\033[0;32m");

} else if (containsLetter(guessedWord[i], wordToGuess)) {

System.out.print("\033[0;33m");

}

System.out.print(guessedWord[i]);

System.out.print("\033[0m");

}

System.out.println();

if (isWordGuessed(guessed)) {

System.out.println("Congratulations! You guessed the word: " + hiddenWord);

break;

} else {

System.out.println("Try again.");

attempts++;

}

}

if (attempts == MAX\_ATTEMPTS) {

System.out.println("Out of attempts! The word was: " + hiddenWord);

}

scanner.close();

}

private static boolean containsLetter(char letter, char[] word) {

for (char c : word) {

if (c == letter) {

return true;

}

}

return false;

}

private static boolean isWordGuessed(boolean[] guessed) {

for (boolean b : guessed) {

if (!b) {

return false;

}

}

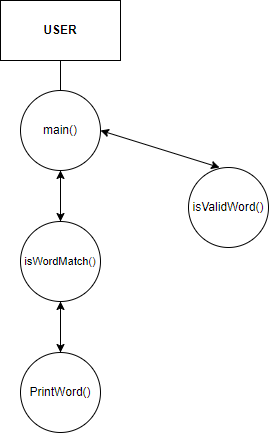
return true;

}

}

# Design:

The basic simple data flow diagram (DFD) for the program is given below:



# 3.1 Hardware and Software Requirements:

Operating System: Windows, Linux No of Systems: 1

Processor: i3 or higher RAM: 1GB minimum

Hard Disk: 500 mb minimum

# Testing/Results and Analysis:

To ensure the functionality of the working of the wordle game, it underwent testing with various test cases. The test cases covered different characters and different possibilities. The program was tested to verify the accuracy of the wordle game and the display of the correct answer.

# Test Cases:

Test Case 1:

Guess #1: Enter your guess: heart

Not the correct word Position and letter matches: h e a r t

Test Case 2:

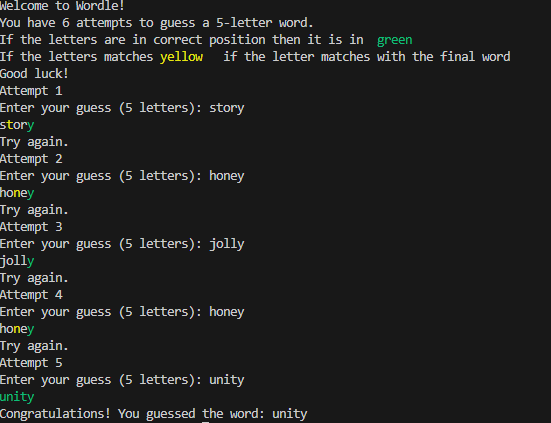
Guess #1: Enter your guess:lemon

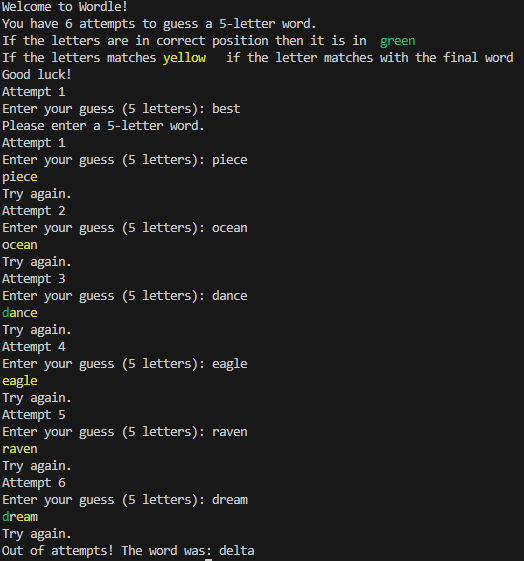
Not the correct word.

Position and letter matches:

l e m o n

# Sample Input and Output:





* 1. Analysis of Results:

The program for the game of wordle of finding the five-letter word within six guesses was written, executed and the output was found successfully.

# Conclusion:

In conclusion, the game of wordle of finding the five-letter word within six guesses has been successfully implemented, providing an interactive and engaging word-guessing experience. The project involved the implementation of key components, including user input, guess validation, and game progress tracking. Through the application of fundamental programming concepts, such as string manipulation and conditional logic, the game was developed to be robust and user- friendly. While the game is fully functional, potential future enhancements could include additional gameplay modes and graphical elements. Overall, the project served as a valuable exercise in applying programming principles and provided a solid foundation for further exploration in game development and software engineering.

# References:

* + **Github**: https://github.com/leonardocrociani/Wordle-in-Java.git
  + **OpenAI**: <https://chat.openai.com/>