Tools and Tech Stack: Programming Language:  
  
**C++:**

This language is used to create the logic for the game, manage the leaderboard, and handle user input and output.

**Common Libraries:**  
  
**<iostream>:** For functions involving input and output.

**<cstdlib>:** For general utilities and the creation of random numbers.

**<ctime>:** To start the random number generator.

**<vector>:** For dynamic array management of the leaderboard.

**<algorithm>:**  Used to arrange the entries on the leaderboard.

**Instruction and Ideas:**

**C++ Code Development:**  
  
Knowledge of the syntax and semantics of C++.

Knowledge of common input and output procedures.

Familiarity with control structures (conditionals, loops).

Use of functions to promote code reuse and modularity.

**Information Organizations:**  
Usage of std::vector for dynamic array management.

Sorting operations are performed using std::sort.

**Project Overview:**

A C++ console program called Random Number Guessing Game asks users to predict a randomly generated number inside a given range. A menu for simple navigation, many difficulty settings, and a leaderboard to keep track of high scores are all included in the game.  
  
**Characteristics and Capabilities**

Levels of Difficulty:  
  
Simple: 1–20 range, 5 tries.

Medium: 10 tries in the 1–50 range.

Hard: 15 tries in the 1-100 range.

**Game mechanics:**  
  
Generates random numbers within the specified range of difficulty.  
Player guesses and name input.  
Please let us know if the estimate is accurate or not.  
calculating the score using the number of attempts left.

**LeaderBoard:**  
  
Top 10 scores will be stored.  
Names and scores are shown in descending order.

**CODE:**

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <vector>

#include <algorithm>

using namespace std;

struct LeaderboardEntry {

string name;

int score;

};

vector<LeaderboardEntry> leaderboard;

int generateRandomNumber(int min, int max) {

return min + rand() % (max - min + 1);

}

void updateLeaderboard(const string& playerName, int score) {

leaderboard.push\_back({ playerName, score });

sort(leaderboard.begin(), leaderboard.end(), [](const LeaderboardEntry& a, const LeaderboardEntry& b) {

return a.score > b.score;

});

if (leaderboard.size() > 10) {

leaderboard.pop\_back();

}

}

void displayLeaderboard() {

cout << "Leaderboard:" << endl;

cout << "Name\tScore" << endl;

for (const auto& entry : leaderboard) {

cout << entry.name << "\t" << entry.score << endl;

}

cout << endl;

}

void playGame(int maxNumber, int maxAttempts) {

int secretNumber = generateRandomNumber(1, maxNumber);

int attempts = 0;

int guess;

string playerName;

cout << "Welcome to the Random Number Guessing Game!" << endl;

cout << "Enter your name: ";

cin >> playerName;

cout << "I have chosen a number between 1 and " << maxNumber << ". You have " << maxAttempts << " attempts to guess it." << endl;

while (attempts < maxAttempts) {

cout << "Enter your guess: ";

cin >> guess;

attempts++;

if (guess == secretNumber) {

int score = (maxAttempts - attempts + 1) \* 10;

cout << "Congratulations, " << playerName << "! You guessed the number in " << attempts << " attempts." << endl;

cout << "Your score is: " << score << endl;

updateLeaderboard(playerName, score);

displayLeaderboard();

return;

} else if (guess < secretNumber) {

cout << "Too low. Try again." << endl;

} else {

cout << "Too high. Try again." << endl;

}

}

cout << "Sorry, " << playerName << ", you've used all " << maxAttempts << " attempts. The number was " << secretNumber << "." << endl;

}

int main() {

srand(static\_cast<unsigned int>(time(0))); // Seed the random number generator

int choice;

do {

cout << "Menu:" << endl;

cout << "1. Play Game" << endl;

cout << "2. View Leaderboard" << endl;

cout << "3. Exit" << endl;

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1: {

int difficulty;

cout << "Select the difficulty level:\n1. Easy (1-20, 5 attempts)\n2. Medium (1-50, 10 attempts)\n3. Hard (1-100, 15 attempts)\nEnter your choice: ";

cin >> difficulty;

switch (difficulty) {

case 1:

playGame(20, 5);

break;

case 2:

playGame(50, 10);

break;

case 3:

playGame(100, 15);

break;

default:

cout << "Invalid difficulty level." << endl;

}

break;

}

case 2:

displayLeaderboard();

break;

case 3:

cout << "Exiting game." << endl;

break;

default:

cout << "Invalid choice." << endl;

}

} while (choice != 3);

return 0;

}

