**YAHTZEE**

**A DICE GAME**

**Ireoluwa Dairo**

**CSC/CIS 5**

**45744**

**Summer 2024**

**INTRODUCTION**

This documentation provides an overview of a Yahtzee game implemented in C++. It covers the game's mechanics and programming approach.

HOW THE GAME WORKS

Yahtzee is a dice game where players roll five dice and attempt to achieve specific combinations to score points. The game involves rolling dice, holding some dice while re-rolling others, and calculating scores based on various combinations.

OBJECT OF THE GAME

The objective of the game is to accumulate the highest possible score by rolling five dice and achieving various scoring combinations over a series of rounds.

RULES OF THE GAME

1.Roll five dice and attempt to achieve the best combinations for points.

2.You can roll the dice up to three times per turn.

3.Specific combinations like Yahtzee, Large Straight, and Four of a Kind have different point values.

GAMEPLAY MECHANICS

•Rolling Dice: The player rolls dice that are not held.

•Holding Dice: The player can choose to hold specific dice between rolls.

•Scoring: Points are awarded based on the combinations rolled.

MY APPROACH TO THE GAME

The game was implemented with a focus on simplicity and user interaction. The program allows for saving and loading game results.

SIMILARITIES TO THE ORIGINAL GAME

•Dice rolling and holding mechanics are similar.

•Scoring combinations and their point values match the traditional Yahtzee game.

DIFFERENCES FROM THE ORIGINAL GAME

•This implementation does not include all Yahtzee combinations or scoring features.

•The game does not support multiplayer.

THE LOGIC OF IT ALL

Declare Variables

Initialize Variables

Display welcome message and prompt for username

Ask if player wants to load a previous game file

IF player wants to load previous game

Prompt for filename

IF file opens successfully

Display previous game results

Display "Try to beat your previous score"

ELSE

Display "File not found. Starting a new game."

END IF

END IF

Ask if player wants to use username as part of filename

Determine filename based on user input

Open file for writing game results

FOR each round (1 to 13)

Reset dice hold status

Output round information

REPEAT

Get user's choice (Roll, Hold, or Quit)

SWITCH user's choice

CASE Roll:

Roll dice not on hold

Display rolled dice values

Write rolled dice values to file

Check for Yahtzee

Check for Large Straight

Decrement number of rolls left

CASE Hold:

For each die

Ask if player wants to hold the die

Update hold status based on player's choice

CASE Quit:

Set round to 13 to end the game

DEFAULT:

Display "Invalid Choice!"

END SWITCH

Notify user of remaining rolls

UNTIL out of rolls or player quits

Calculate and display round score

Write round results to file

IF last round

Calculate and display final score and average

Write final results to file

END IF

END FOR

Close the file

PROOF OF WORKING CODE

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**A screenshot of a computer

Description automatically generated**

**THE CODE**

/\*

\* Author: Ireoluwa

\* Created on July 21, 4:18 pm

\* Purpose: a game of Yahtzee

\*/

// System Libraries

#include <iostream> // I/O library for input and output operations

#include <ctime> // Library to work with time functions

#include <cstdlib> // Random number generation

#include <fstream> // File input and output

#include <iomanip> // I/O manipulator

#include <string> // string library

#include <cmath> // Math library

using namespace std;

//User Libraries

//Global Constants - Mathematical, Scientific, Conversions

//Higher Dimensions go here. No Variables

//Function Prototypes

//Execution Begins here

int main(int argc, char\*\* argv) {

// Setting the random number seed

srand(static\_cast<unsigned int>(time(0)));

// Declaring Variables

fstream in, out; // File streams for input and output

const int Md = 6, mnD = 1; // Max and min value for a dice

char GChce, Chce, sChce, plymode, shwRuls, ldgFile; // game Choices

int d1, d2, d3, d4, d5, rllsLft, ttlScre, y;

bool h1, h2, h3, h4, h5; // Boolean to track dice status

float ascore;

string Yname, fname, line;

// Initialize Variables

out.open(fname, ios::out);

in.open(fname, ios::in);

ttlScre = 0;

// Welcome message and prompt for username

cout << "Welcome to Yahtzee!" << endl;

cout << "Enter Username: ";

getline (cin, Yname);

cout << endl << "Welcome " << Yname <<"!" << endl;

// Ask if the player wants to load a previous game file

cout << "Do you want to load a previous game file? (Y/N): ";

cin >> ldgFile;

cin.ignore();

while(ldgFile != 'Y' && ldgFile != 'y' && ldgFile != 'N' && ldgFile != 'n'){

cout << "Invalid Input. Input 'Y' or 'N': ";

cin >> ldgFile;

cin.ignore();

}

// Load game file if the user chooses to

if (ldgFile == 'Y' || ldgFile == 'y') {

cout << "Enter the filename to load the game results: ";

getline(cin, fname);

in.open(fname, ios::in);

if (in.is\_open()) {

while (getline(in, line)) {

cout << line << endl << endl;

}

cout << "Try to beat your previous score" << endl;

cout << "Starting New Game!" << endl << endl;

in.close();

} else {

cout << "File not found. Starting a new game." << endl << endl;

}

}

// Ask if the player wants to use their username as part of the filename

cout << "Do you want to use your username as part of the filename? (Y/N): ";

cin >> sChce;

cin.ignore();

while ( sChce != 'Y' && sChce != 'y' && sChce != 'n' &&

sChce != 'N'){

cout << "Invalid Input. Input 'Y' or 'N': ";

cin >> sChce;

cin.ignore();

}

// Determine the filename based on user input

if (sChce == 'Y' || sChce == 'y') {

fname = Yname + " game results.txt";

}

else {

cout << "Enter the filename to save the game results:";

getline (cin, fname);

}

// save file - Open file for writing game results

out.open(fname, ios::out);

if (out.is\_open()) {

out << "Welcome to Yahtzee!" << endl;

}

// Game loop to manage multiple rounds

for (int rnd = 1; rnd <= 13; rnd++) {

h1 = h2 = h3 = h4 = h5 = false; // Reset dice hold status

rllsLft = 3; // Number of rolls in the current round

// Output round information

cout << "Round " << rnd << endl;

out << "Round " << rnd << endl; // Write round to file

// Get the user's Choice for the next action

do {

cout << "Do you want to (R)oll, (H)old, or (Q)uit?" << endl;

cin >> GChce;

cin.ignore();

switch (GChce) { // Switch statement to handle the user's Choice

case 'R': // If the user chooses to roll the dice

case 'r':

if (!h1) d1 = rand() % Md + mnD;

if (!h2) d2 = rand() % Md + mnD;

if (!h3) d3 = rand() % Md + mnD;

if (!h4) d4 = rand() % Md + mnD;

if (!h5) d5 = rand() % Md + mnD;

// Display rolled dice values

cout << "You rolled: " << d1 << " " << d2 << " ";

cout << d3 << " " << d4 << " " << d5 << endl;

// Write rolled dice values to file

out << "Rolled: " << d1 << " " << d2 << " ";

out << d3 << " " << d4 << " " << d5 << endl;

// Check for Yahtzee

if (d1 == d2 && d2 == d3 && d3 == d4 && d4 == d5) {

cout << "Yahtzee!" << endl;

out << "Yahtzee!" << endl;

}

// Check for Large Straight

else if ((d1==1 && d2==2 && d3==3 && d4 == 4 && d5 == 5) ||

(d1 == 2 && d2 == 3 && d3 == 4 &&d4 == 5 && d5 == 6)) {

cout << "Large Straight!" << endl;

out << "Large Straight!" << endl;

}

--rllsLft; // Decrement the number of rolls left

break;

case 'H': // If the user chooses to hold certain dice

case 'h':

// Ask the user if they want to hold a certain dice

cout << "Hold dice 1 (" << d1 << ")? (Y/N): ";

cin >> Chce;

cin.ignore();

if (Chce == 'Y' || Chce == 'y') {h1 = true;}

else if (Chce == 'N' || Chce == 'n') {h1 = false;}

else {cout << "Invalid Choice dice will reroll!" << endl;}

cout << "Hold dice 2 (" << d2 << ")? (Y/N): ";

cin >> Chce;

cin.ignore();

if (Chce == 'Y' || Chce == 'y') {h2 = true;}

else if (Chce == 'N' || Chce == 'n') {h2 = false;}

else {cout << "Invalid Choice dice will reroll!" << endl;}

cout << "Hold dice 3 (" << d3 << ")? (Y/N): ";

cin >> Chce;

cin.ignore();

if (Chce == 'Y' || Chce == 'y') {h3 = true;}

else if (Chce == 'N' || Chce == 'n') {h3 = false;}

else {cout << "Invalid Choice dice will reroll!" << endl;}

cout << "Hold dice 4 (" << d4 << ")? (Y/N): ";

cin >> Chce;

cin.ignore();

if (Chce == 'Y' || Chce == 'y') {h4 = true;}

else if (Chce == 'N' || Chce == 'n') {h4 = false;}

else {cout << "Invalid Choice dice will reroll!" << endl;}

cout << "Hold dice 5 (" << d5 << ")? (Y/N): ";

cin >> Chce;

cin.ignore();

while (Chce != 'Y' &&Chce !='y' &&Chce !='n' &&Chce != 'N'){

cout << "Invalid Input. Input 'Y' or 'N': ";

cin >> Chce; cin.ignore();

}

h5 = (Chce == 'Y' || Chce == 'y') ? true : false;

break;

case 'Q': // If the user chooses to quit the game

case 'q':

cout << "You Quit!" << endl;

rnd = 13;

break;

default: // If the user enters an invalid Choice

cout << "Invalid Choice!" << endl;

}

// Notify the user of remaining rolls

if (rllsLft > 0 && GChce != 'Q' && GChce != 'q') {

cout << "You have " << rllsLft << " rolls left." << endl<<endl;

}

} while (rllsLft > 0 && GChce != 'Q' && GChce != 'q');

cout << endl;

//Displaying Input/Output Information

int rndScre = d1 + d2 + d3 + d4 + d5;

ttlScre += rndScre;

cout <<"End of round "<<rnd<<endl;

cout <<"Score for this round: "<<rndScre<<" points"<<endl<<endl;

// Write round results to file

out <<"Score for this round: "<< rndScre << " points"<<endl << endl;

if (rnd == 13){ // Display final score after the last round

ascore = (ttlScre / 13.0f);

y = round(ascore);

cout<< "You finished with a score of " << setw(3) << setfill('0')<<

ttlScre<<" points with an average of "<<y<<" point(s) per round"<<endl;

// Write final results to file

out << "You finished with a score of " << setw(3) << setfill('0')<<

ttlScre<<" points with an average of "<<y<<" point(s) per round"<<endl;

}

}

// Close the file

out.close();

//Exiting stage left/right

return 0;

}