Project 1

**Hangman Game**

By: Minh Truong

CIS-5-46332

July 25, 2022

Tables of Content

Introduction……………………………………………………………………………………..…3

Description………………………………………………………………………………………...3

Word\_Lists.txt……………………………………………………….……………………………7

Cross Reference for Project………………….…………………………………………………..10

Flow Chart……………………………………………………………………………………….11

How to play the game……………………..……………………………………………………..12

Introduction

Hangman is a quick and easy game for at least two people but in this case, it would be between the player and the console. The console will randomly pick a word, while you the player tries to guess the word by typing any letter you want. Every wrong guess brings you one step closer to losing. You are limited to certain number of tries to guess the word so the objective is for you to win, if not you will be hanged!

Description:

This project contains 230 lines of code.

/\*

\* File: main.cpp

\* Author: Minh Truong

\* Created on: July 24, 2022 11:36 PM

\* Purpose: Hangman Game

\*/

//The standard library files used

#include<iostream> //for basic input and output

#include<cstdlib> //for exit() and system(“cls”) functions

#include<fstream> //for handling the file "words.txt"

#include<string> // for string

#include<cmath>

using namespace std;

/// <summary>

/// Hangman word guessing game

/// - Enter letters until you have guessed the entire word

/// </summary>

/// <returns></returns>

int main() //The main function

{

// show a start menu to the user

char choice; //to store choice

int tries = 7; // number of tries default = 7

int level = 1; // difficulty level

do {

system("cls");

cout << endl << "+-----------------------------------------------------------------------------+";

cout << endl << "| Hangman Game |";

cout << endl << "+-----------------------------------------------------------------------------+";

cout << endl << endl;

cout << "Select an Option: ";

cout << "\n1. Start Game";

cout << "\n2. Change Difficulty level";

cout << "\n3. How to play";

cout << "\n4. Exit";

cout << endl << "\t\tChoice: ";

cin >> choice; //input value

if (isnan(choice))

{

choice = '1';

}

if (choice == '1') //start game

{

// load the file containing a list of words - words.txt

char ch;

int i = 0;

ifstream fin("Word\_Lists.txt", ios::in); //open file

fin.seekg(0); //start at beginning of the file

if (!fin) //if the file is not found there

{

system("cls");

cout << endl << "+-----------------------------------------------------------------------------+";

cout << endl << "| Hangman Game |";

cout << endl << "+-----------------------------------------------------------------------------+";

cout << endl << endl;

cout << "\nError 701. Unable to open file!";

cout << "\n\n - This program needs the accompanying file \"Word\_Lists.txt\" to run.";

cout << "\n - Please check that the file exists with program and restart again.";

cout << "\n - If you can\'t find the file then create a new text file with the same name.";

cout << "\n - Type the words to be asked in the hangman game in separate lines.";

exit(1); //exit with 1 status

}

string content = "";

bool noteof = true;

while (noteof)

{

fin.get(ch);

if (fin.eof()) break;

content = content + ch;

i++;

}

fin.close();

// 1. start game

// pick a word to use in the game

string word = "game";

char letter; //user's guessed letter

int wrong = 0; //no of wrong guesses

int right = 0; //no of right guesses

int length = word.length();

string unknown = word;

for (i = 0; i < length; i++) // to generate the secret word

{

unknown[i] = '\*';

}

unknown[i] = '\0';

cout << "====================";

cout << "================== Level: "; cout << level;

cout << "== No of Tries: " << tries << " " << endl;

cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~";

while (wrong < tries) //condition for running program

{

cout << endl << endl << unknown;

cout << endl << "\tYour Guess: ";

while (!(cin >> letter)) //if user provides invalid input

{

cout << "\tInvalid input! Try again: ";

}

if (word.find(letter) != string::npos) //else he will find the letter

{

cout << "\tYou found a letter!";

int ipos = word.find(letter);

unknown[ipos] = letter;

right++;

}

else //if user provides wrong guess

{

cout << "\tOops! Your guess is wrong!";

wrong++;

}

cout << " Now You have " << tries - wrong << " tries left.";

if (word == unknown && word.length() > 0) //checking if he guessed the word

{

cout << endl << endl << "Yeah! You got it! The word was: " << word;

cout << endl << endl;

cin.get();

cin.get();

break;

}

}

bool lost = wrong == tries ? true : false;

if (lost) //condition for user to lose

{

float percent = (float)right / (float)word.length();

cout << endl << endl << "\nSorry, you lose...you've been hanged.";

cout << " The word was: " << word << endl << endl;

cout << " You got " << percent \* 100 << "% correct." << endl;

cin.get();

cin.get();

}

}

else if (choice == '2') //change difficulty level

{

system("cls");

cout << endl << "+-----------------------------------------------------------------------------+";

cout << endl << "| Hangman Game |";

cout << endl << "+-----------------------------------------------------------------------------+";

cout << endl << endl;

char level;

cout << "Select a level: ";

cout << "\n1. Very Easy";

cout << "\n2. Easy";

cout << "\n3. Moderate";

cout << "\n4. Hard";

cout << "\n5. Extremely Hard";

cout << "\n\t\tChoice: ";

cin >> level;

switch (level)

{

case ('1'):

tries = 15;

break;

case ('2'):

tries = 10;

break;

case ('3'):

tries = 7;

break;

case ('4'):

tries = 5;

break;

case ('5'):

tries = 3;

break;

default:

cout << "\t\tInvalid Input! Try again.";

cin.get();

// show the options again

break;

}

cout << "\t\tLevel Changed !\n";

if (tries == 15) cout << "\t\tVery Easy =====";

else if (tries == 10)cout << "\t\tEasy ==========";

else if (tries == 7) cout << "\t\tModerate ======";

else if (tries == 5) cout << "\t\tHard ==========";

else if (tries == 3) cout << "\t\tExtremely Hard ";

cin.get();

cin.get();

}

else if (choice == '3') //how to play option

{

// show the rules to the user

system("cls");

cout << endl << "+-----------------------------------------------------------------------------+";

cout << endl << "| Hangman Game |";

cout << endl << "+-----------------------------------------------------------------------------+";

cout << endl << endl;

cout << "\n\nHere's how to play: ";

cout << "\n - Try to guess the word before your chances run out.";

cout << "\n - Type in the letters you wish to guess. Careful! Wrong letters lose chances.";

cout << "\n - You can change the level from the Change Difficulty Level option.";

cout << "\n\n\t\t\t\t Press any key to go back to main menu...";

cin.get();

cin.get();

}

else if (choice == '4') //exit program

{

cout << "\t\tExited successfully!";

exit(0);

}

else

{

cout << "\t\tInvalid Input! Try again.";

cin.get();

// show the menu

}

} while (true);

return 0;

}

/\*End of program.\*/

Word\_List.txt

Fruit:

Papaya

Jambolan

Dewberries

Yangmei

Tamarind

Avocados

Kumquat

Orange

Melon

Bananas

Hackberry

Entawak

Eggfruit

Raspberries

Tomato

Wolfberry

Persimmon

Imbe

Fig

Rambutan

Pomegranate

Mulberry

Apples

Uniq Fruit

Cucumbers

Zucchini

Cantaloupe

Xigua

Olive

Dragon

Elderberry

Clementine

Lychee

Lime

Mango

Kiwi

Guava

Apricots

Ugni

Longan

Dates

Grapefruit

Jackfruit

Tangerine

Mandarin

Cherries

Blueberries

Watermelon

Nectarine

Grapes

Strawberries

Quince

Evergreen

Pineapple

Peach

Gooseberries

Boysenberries

Voavanga

Loquat

Oranges

Indonesian Lime

Huckleberry

Honeydew melon

Farkleberry

Animal:

Bear

Cat

Cow

Dog

Elephant

Fox

Giraffe

Horse

Kangaroo

Lion

Tiger

Wolf

Wolverine

Cat

Chameleon

Dog

Fish

Hamster

Horse

Mouse

Parrot

Pig

Pigeon

Puppy

Rabbit

Snake

Turtle

Crocodile

Deer

Gorilla

Hippopotamus

Jaguar

Kangaroo

Lion

Monkey

Panda

Rhinoceros

Squirrel

Tiger

Wolf

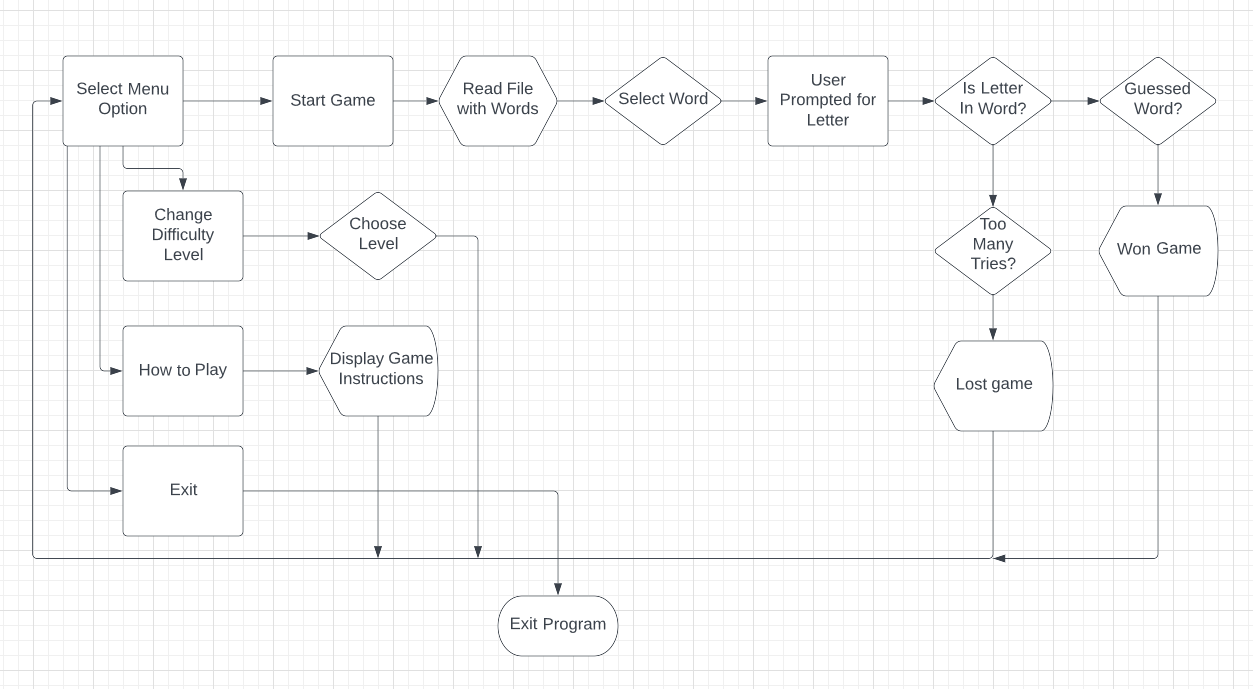
Yak

Zebra

Table

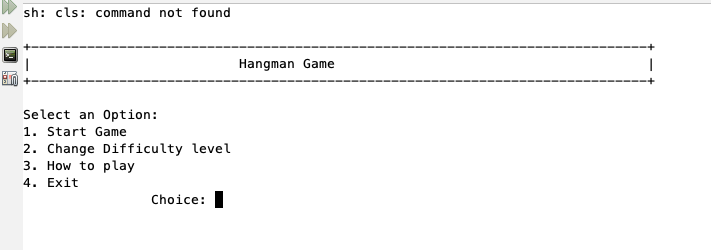
Description automatically generated

Flow Chart



How to play the game:

1. **The beginning of the game starts with a menu option.**



1. **Choosing option 1 starts your game and you can guess your first letter. Your word is displayed on the left of the screen with asterisks.**

Text, table

Description automatically generated

1. **If you guessed the wrong letter, you would get an error message that lets you know how many tries you have left.**

Table

Description automatically generated

1. **If you don’t guess any correct letters, you will get a message “You’ve been hanged” and reveals the word you were supposed to guess. Below that, it displays the percentage of how many letters you guessed correctly.**



1. **If you guess the correct letter, the letter will reveal itself on the left of the screen. And if you guess it correctly it will display a message “Yeah! You got it!”**

Table

Description automatically generated