Project 2

**Hangman Game**

By: Minh Truong

CIS-5-46332

July 31, 2022

Tables of Content

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

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

Hangman Code……………………………………………………….……………………………7

Word Lists………………...………………….…………………………………………………..10

Flow Chart……………………………………………………………………………………….13

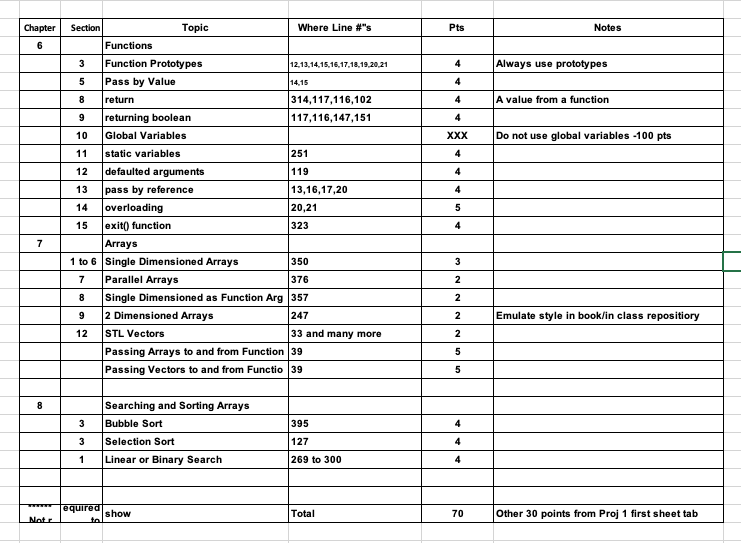
How to play the game……………………..……………………………………………………..14

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 420 lines of code.



/\*

\* Author: Minh Truong

\* Created on July 31, 2022 2:15PM

\* Purpose: Project 2 Hangman

\*/

//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<vector>// for Vectors

using namespace std;

void Display(int\* arr, int size);

int\* Sort(int\* arr, int size);

int\* ScoreHandler(int\* arr, int score, int size);

int Menu();

int\* IntRegrow(int\* arr, int size);

vector<string>& ReadDictionary(vector<string>& ptr);// To Read The File For Words

string RandomWord(vector<string> dic);// To select the Random Word from the vector

string UnknownGenerator(string str);// To Generate the unknown form of the random searched word i.e \*\*\*\*\*

bool MatchLetter(string org, string& unkown, char ch);// To search and to display the letter in the \*\*\* form of random word

bool GuessWord(vector<string>& dic,int& score, int tries = 15, int level = 1);// To Validate if the word was correct or not

int SelectLevel(int& level);// To Select word

void HowToPlay();// TO SHOW INSTRUCTIONS ON HOW TO PLAY

void Startgame(vector<string>& dic, int& tries, int& level);// TO mainpulate other functions and to start the game.

void Startgame();

vector<string> LevelBalancer(int level, vector<string>& dic);// To Linearly search for the word equal or less

//then the lives in length of the word based on the levels.

/// <summary>

/// Hangman word guessing game

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

/// </summary>

/// <returns></returns>

int main() //The main function

{

int\* arr = nullptr;

int score = 0;

int size = 0;

vector<string>dic;

bool flag;

ReadDictionary(dic);

int tries = 0, level = 0;

tries = SelectLevel(level);

vector<string>d;

d = LevelBalancer(level, dic);

Startgame(dic, tries, level);

}

/\*End of program.\*/

int Menu()

{

int choice;

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

return choice;

}

string RandomWord(vector<string> dic)

{

bool flag = true;

string temp;

srand(time(NULL));

int ind;

while (flag != false || ind % 2 != 0)

{

flag = false;

ind = (rand() % 10);

temp = dic[ind];//using random function to search random word.

for (int x = 0;x < temp.size();x++)

{

if (temp[x] == ' ')

{

flag = true;

}

}

}

return temp;

}

vector<string>& ReadDictionary(vector<string>& ptr)

{

ifstream fin("Word\_Lists.txt");

if (fin.is\_open())

{

char ch;

string temp = "";

while (!fin.eof())

{

getline(fin, temp);

if (temp != " " || temp != "\n")

{

ptr.push\_back(temp);

}

}

fin.close();

}

else

{

Startgame();

}

return ptr;

}

string UnknownGenerator(string str)

{

string temp = "";

for (int x = 0; x < str.length(); x++)// Make aand return the unknown form of the random word.

{

temp += "\*";

}

temp[1] = str[1];

temp[temp.length() - 1] = str[str.length() - 1];

return temp;

}

bool MatchLetter(string org, string& unknown, char ch)

{

bool flag = false;

for (int x = 0; x < org.length(); x++)// searches and convert the letter to the orgnal one from \*

{

if (org[x] == ch)

{

unknown[x] = ch;

flag = true;

}

}

if (flag == true)

return true;

return false;

}

bool GuessWord(vector<string>& dic, int& score, int tries, int level)

{

string org = RandomWord(dic);

string unknown = UnknownGenerator(org);

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

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

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

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

int count = 0;

char ch;

bool flag = false;

while (flag != true || count + 2 != org.length())

{

cout << endl << endl << unknown;

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

cin >> ch;

if (MatchLetter(org, unknown, ch))

{

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

tries--;

score += 5;

count++;

}

else

{

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

tries--;

}

if (org == unknown)

return true;

if (tries == 0)

{

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

return false;

}

cout << " Now You have " << tries << " tries left.\n";

}

}

int SelectLevel(int& level)

{

int tries = 0;

system("cls");

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

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

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

cout << endl << endl;

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();

return tries;

}

void HowToPlay()

{

// 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();

}

void Startgame(vector<string>& dic, int& tries, int& level)

{

int choice;

int\* arr = nullptr;

int size = 0;

while (1)

{

static int score = 0;

score = 0;

system("cls");

choice = Menu();

if (choice == 1)

{

if (GuessWord(dic, score, tries, level))

{

cout << endl << endl << "Yeah! You got it! Amazing Job The man was save";

cout << endl << endl;

arr=ScoreHandler(arr, score, size);

size++;

cin.get();

cin.get();

}

else

{

cout << "Sorry Out Of Lives The Man Was Hanged\n\n";

arr = ScoreHandler(arr, score, size);

size++;

cin.get();

cin.get();

}

}

else if (choice == 2)

{

tries = SelectLevel(level);

}

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

{

HowToPlay();

}

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

{

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

system("cls");

arr = Sort(arr, size);

cout << "High Score: ";

Display(arr, size);

exit(0);

}

else

{

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

cin.get();

// show the menu

}

}

}

vector<string> LevelBalancer(int level, vector<string>& dic)

{

vector<string> temp;

if (level == 1)

return dic;

else if (level == 2)

{

for (int x = 0; x < dic.size(); x++)

{

if (dic[x].length() <= 10)

{

temp.push\_back(dic[x]);

}

}

}

else if (level == 3)

{

for (int x = 0; x < dic.size(); x++)

{

if (dic[x].length() <= 7)

{

temp.push\_back(dic[x]);

}

}

}

else if (level == 4)

{

for (int x = 0; x < dic.size(); x++)

{

if (dic[x].length() <= 5)

{

temp.push\_back(dic[x]);

}

}

}

else if (level == 5)

{

for (int x = 0; x < dic.size(); x++)

{

if (dic[x].length() <= 3)

{

temp.push\_back(dic[x]);

}

}

}

return temp;

}

void Startgame()

{

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 \"Words.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

}

int\* IntRegrow(int\* arr, int size)

{

if (size == 0)

{

arr = new int[size + 1];

return arr;

}

int\* temp = new int[size + 1];

for (int x = 0;x < size;x++)

{

temp[x] = arr[x];

}

delete[]arr;

arr = temp;

temp = nullptr;

return arr;

}

int\* ScoreHandler(int\* arr, int score, int size)

{

arr = IntRegrow(arr, size);

arr[size] = score;

return arr;

}

int\* Sort(int\* arr, int size)

{

for (int x = 0;x < size - 1;x++)

{

static int temp = 0;

for (int y = x + 1;y < size - x - 1;y++)

{

if (arr[y] < arr[y + 1])

{

temp = arr[y];

arr[y] = arr[y + 1];

arr[y + 1] = temp;

}

}

}

return arr;

}

void Display(int\* arr, int size)

{

for (int x = 0;x < size;x++)

{

cout << arr[x] << " ";

}

}

Words for Hangman

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

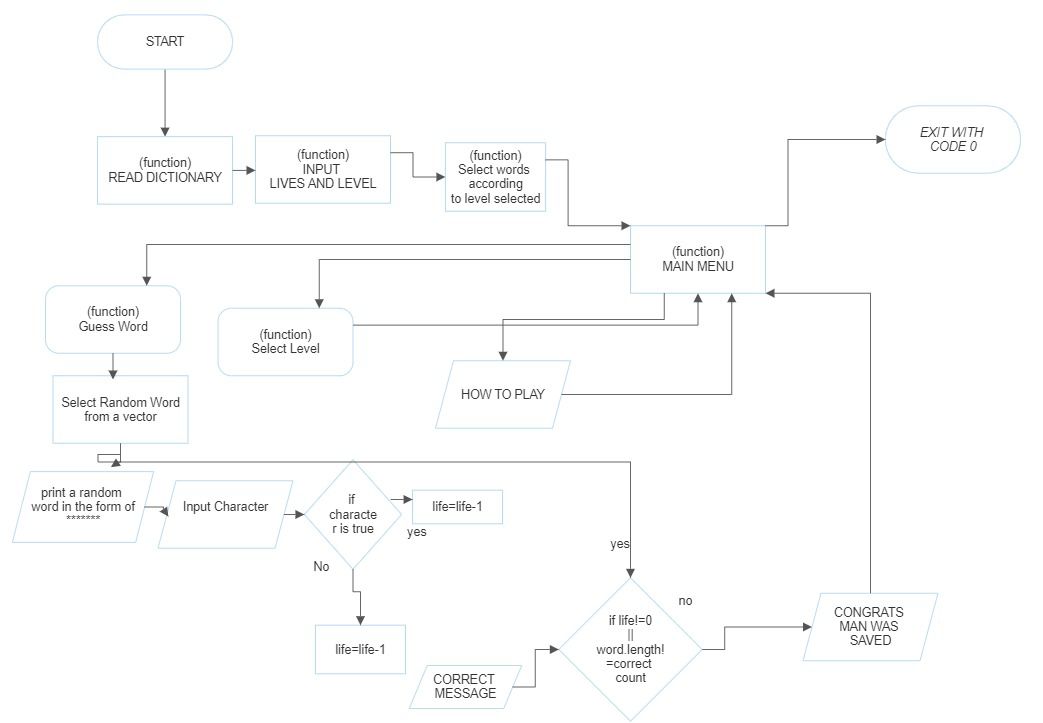
Tiger

Wolf

Yak

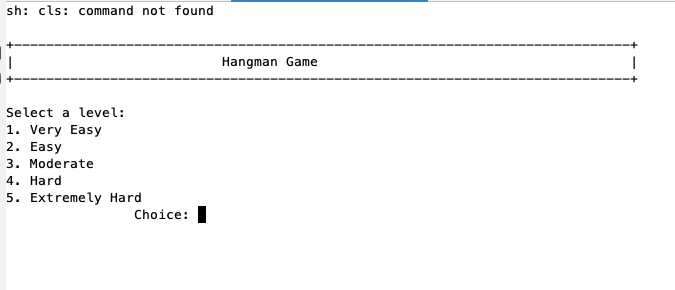
Zebra

Flow Chart



How to play the game:

1. **The beginning of the game starts with a option to choose your level.**



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

Table

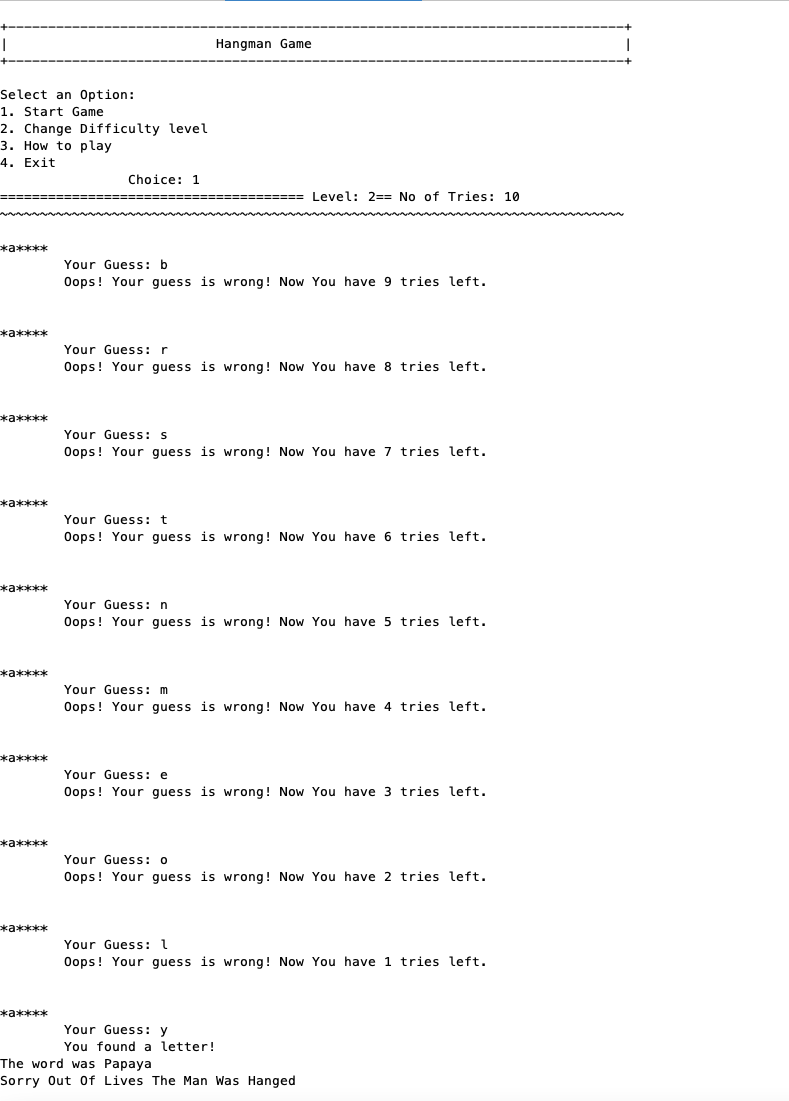
Description automatically generated

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

Text, email

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!”**

