Laquon Hamilton

ITS-240

Group 11

Final Project Deliverable

I chose to make a ﻿﻿﻿﻿﻿Hangman game for this project. The ﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿program ﻿operates as a simple ﻿hangman game. The user will be presented with the main menu with three options, being Begin Game, View Scores, and Exit Program, all doing just as you'd expect. ﻿﻿﻿﻿The user﻿ will enter a nickname to be displayed in-game and﻿﻿﻿ stored alongside their ﻿score on the scoreboard. Once that is done, ﻿﻿﻿﻿the game can begin. Once the user either guesses all letters correctly or runs out of lives, the program will conclude, and their nickname and score will be sent to the scoreboard. ﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿The user can also type quit to end the game early, and their score will still be sent to the scoreboard. Line 121 is where the person giving the words ﻿﻿﻿﻿can enter or edit the words to be randomly chosen in the game.

Source code:

/\*

  Laquon Hamilton

  Final Project

  2020-04-29 - 2020-05-02

  Hangman Game

\*/

#include <stdio.h>        //user input and output

#include <stdlib.h>       //for conversions such as rand and srand

#include <string.h>       //for reading the length of strings

#include <time.h>         //for time function

#define charSIZE 100      //for constant character sizes

void gameFunction(char\*, FILE \*inFile); //function prototype for game function

void resultsFile(FILE \*inFile); //function prototype for results file

int main()

{

  system("clear");  //for clearing the console to make things neater, but for visual studio it is "cls" instead of "clear"

  printf("\nProgram that operates as a hangman game. You will enter a nickname to be stored alongside your score in the scoreboard. Line 121 is where the game master can enter or edit the words to be randomly chosen in the game.\n");

  int menuOption;           //for choosing menu option

  char nickName[charSIZE];      //for setting a nickname

  FILE \*inFile;

  char welcomeMsg[] = "\nThank you for playing the Hangman Game! \n";

  char \*msgPtr;             //pointer variable

  msgPtr = welcomeMsg;      //stores the address of the array in msgPtr

  //menu options

  printf("\n1-- Begin Game\n");

  printf("\n2-- View Scores\n");

  printf("\n3-- Exit Program\n\n: ");

  scanf("%d", &menuOption);

  //input validation for menu

  if (menuOption == 1)

  {

    while (getchar() != '\n') //ensure only the enter/return key can be input

    {

      printf("\nYou must press enter in order continue.\n");

      getchar();

    }

    system("clear");

    for (int i = 0; i < strlen(welcomeMsg); i++)  //for printing welcome message

    {

      printf("%c", \*(msgPtr + i));

    }

    printf("\nEnter a nickname with no spaces: ");

    fgets(nickName, charSIZE, stdin);   //gets the string entered

    while ((\*nickName == ' ') || (\*nickName == '\t') || (\*nickName == '\n'))              //for preventing spaces from being entered

    {

      printf("\nYou must enter a name with no spaces.\n");

      printf("\nEnter a nickname: ");

      fgets(nickName, charSIZE, stdin);

    }

    //for removing spaces from the nickname

    for(int i = 0, j = 0; i < strlen(nickName); i++)

    {

      nickName[i-j] = nickName[i];

      if(nickName[i] == ' ')

      {

        j++;

      }

    }

    printf("\nYour nickname is: %s", nickName);

    printf("\nPress enter or return to continue... ");

    while (getchar() != '\n') //for pausing the loops

    {

      printf("\nYou must press enter in order continue.\n");

      getchar();

    }

    system("clear");

    gameFunction(nickName, inFile); //call the game function

  }

  else if (menuOption == 2)

  {

    system("clear");

    resultsFile(inFile);      //call the results function and sends inFile to it

  }

  else if (menuOption == 3)

  {

    exit(1); //exits the program

  }

  else if (menuOption != 1 || menuOption != 2 || menuOption != 3)

  { //input validation for menu choices

    printf("\nPlease enter a number only between 1 - 3\n");

    printf("\nPress enter or return to continue...\n");

    getchar();

    while (getchar() != '\n')

    {

      getchar();

    }

    system("clear");

    main();

  }

  return 0;

}

void gameFunction(char \*nickName, FILE \*inFile)

{

  srand(time(NULL)); //sets the seed of the random number generator algorithm used by rand()

  //here the game master can enter 6 words to be randomly chosen for the game

  char guessWords[][16] = {               "break",

                            "void",

                            "struct",

                            "continue",

                            "union",

                            "quarantine"

  }; //2D array where first index is for choosing a random word and the second is for specifying a max length for the strings

  printf("\n\*For testing\* The word bank to be randomly chosen from is:\n");

  for (int i = 0; i < strlen(guessWords[i]); i++) //for testing the array, can be removed

  {

    printf("\n%s ", guessWords[i]);

  }

  printf("\n\nPress enter or return to begin the game... ");

  while (getchar() != '\n') //for pausing the loops

  {

    printf("\nYou must press enter in order continue.\n");

    getchar();

  }

  system("clear");

  int randIndex = rand() % 6; //generates a random number which will be used to choose a randon number in the array index

  int numbLives = 5; //total lives

  int totalPoints = 0; //points awarded for correct guesses

  int wrongGuess = 0; //for deducting points for wrong guesses

  int letterAccu = 0; //for ending the game when all letters have been guessed

  int wordLength = strlen(guessWords[randIndex]); //reads the length of the chosen word

  int guessedLetter[16] = {0, 0, 0, 0, 0, 0, 0, 0}; //where guessed letters will be stored

  int userQuit = 0; //used for when the user quits the game early

  int loopIndex = 0; //used for looping through the array index

  char guessLetters[16]; //letters the user enters

  char letterGuessed; //for storing the user entered letter

  while (totalPoints < wordLength) //loops the game

  {

    if (totalPoints < 0)  //keeps score from going below zero

    {

      totalPoints = 0;

    }

    printf("\nChosen Word: %s | Index Number: %d | Word Length: %d\n", guessWords[randIndex], randIndex, wordLength); //for testing the program, can be commented out/removed

    printf("\nNote: The above is for testing the game. It can be removed on line 155 for a serious game.");

    printf("\n------------------------------------------------------\n");

    printf("\n|New Turn|\n\nCurrent Player: %s\nHangman Word: ", nickName);

    for (loopIndex = 0; loopIndex < wordLength; loopIndex++) //loops randomly through the first array index

    {

      if(guessedLetter[loopIndex] == 1) //if the array index for the chosen word has been activated

      {

        printf("%c", guessWords[randIndex][loopIndex]); //print character that was correctly guessed

      }

      else

      {

        printf("-");  //prints hyphens until a letter is correctly guessed

      }

      if (letterGuessed == guessWords[randIndex][loopIndex])

      { //if user entered a correct letter

        guessedLetter[loopIndex] = 1; //activates the array index of the correct letter

      }

    }

    printf("\n");

    printf("\nLives left: %d\n", numbLives);

    printf("\nTotal Points: %d\n", totalPoints);

    printf("\nCurrent letter number: %d\n", letterAccu);

    printf("\nEnter a letter to be guessed: ");

    fgets(guessLetters, 16, stdin); //gets the user entered letter

    if (strncmp(guessLetters, "quit", 4) == 0)//if user enters quit

    {

      userQuit = 1;

      break;  //ends the loop

    }

    letterGuessed = guessLetters[0]; //stores the guessed letter in the array

    printf("\nLetter entered: %c", letterGuessed); //displays the entered letter

    wrongGuess = totalPoints; //for subtracting lives and points if user entered wrong letter

    //loops for when the correct letter is guessed until word length is reached

    for (loopIndex = 0; loopIndex < wordLength; loopIndex++)

    {

      if(guessedLetter[loopIndex] == 1)

      {

        continue; //continues with the loop

      }

      if (letterGuessed == guessWords[randIndex][loopIndex])

      { //add points when correct letter guessed

        guessedLetter[loopIndex] = 1; //activate array which holds the letter

        totalPoints++;    //add one point

        letterAccu++;   //accumulate letter counter

      }

    }

    if (wrongGuess == totalPoints) //if wrong letter is entered

    {

      numbLives--;    //deduct 1 life

      totalPoints--;    //deduct 1 point

      printf("\n-------------------------------\n");

      printf("\nSorry, wrong guess!\n");

      printf("\n-------------------------------\n");

      printf("\nPress enter or return to continue... ");

      getchar();

      system("clear");

      if (numbLives == 0) //ends game once all lives are gone

      {

        break;

      }

    }

    else

    {

      printf("\n-------------------------------\n");

      printf("\nCorrect guess!\n");

      printf("\n-------------------------------\n");

      printf("\nPress enter or return to continue... ");

      getchar();

      system("clear");

    }

    system("clear");

  if (letterAccu == wordLength) //ends the loop, ending the game once all letters have been guessed

  {

    break;

  }

  }

  if (userQuit == 1) //if user quits the game early

  {

    printf("\n-------------------------------\n");

    printf("\nYou have quit the game. The chosen word was: %s\n", guessWords[randIndex]);

    printf("\nYou scored: %d\n", totalPoints);

    printf("\n-------------------------------\n");

    printf("\nPress enter or return to return to the main menu... ");

    getchar();

    system("clear");

  }

  else if (numbLives == 0)  //when all lives are lost

  {

    if (totalPoints < 0)  //keeps score from going below zero

    {

      totalPoints = 0;

    }

    printf("\n---------------------------------------------------------\n");

    printf("\nSorry, you lose. The chosen word was: %s\n", guessWords[randIndex]);

    printf("\nYou scored: %d\n", totalPoints);

    printf("\n---------------------------------------------------------\n");

    printf("\nPress enter or return to return to the main menu... ");

    getchar();

    system("clear");

  }

  else

  {

    printf("\n---------------------------------------------------------\n");

    printf("\nCongratulations, you've won! The chosen word was: %s\n",guessWords[randIndex]);

    printf("\nYou scored: %d\n", totalPoints);

    printf("\n---------------------------------------------------------\n");

    printf("\nPress enter or return to return to the main menu... ");

    getchar();

    system("clear");

  }

  inFile = fopen("results.txt", "a+");    //creates the file

  fprintf(inFile, "%s\t%d\n", nickName, totalPoints); //prints nickname and total points to the file

  fclose(inFile);               //closes the file

  main();

return;

}

struct scores   //structure for displaying scores

{

  char dispName[charSIZE];          //display name

  int dispScore;                //display score

} displayScores;

void resultsFile(FILE \*inFile)

{

  struct scores displayScores; //allows this function to read from the structure

  inFile = fopen("results.txt", "r");     //opens file for reading

  if (inFile == NULL)             //check if file exists

  {

    printf("\n---------------------------------------------------------\n");

    printf("\nNo scores to display yet!\n");

    printf("\n---------------------------------------------------------\n");

    printf("\nPress enter or return to return to the main menu... ");

    getchar();

    while (getchar() != '\n')

    {

      getchar();

    }

    system("clear");

    main();

  }

  printf("\n---------------------------------------------------------\n");

  printf("\nAll scores\n");

  printf("\n---------------------------------------------------------\n");

  while (fscanf(inFile, "%s\r%d", displayScores.dispName, &displayScores.dispScore) != EOF) //reads values from the file for the listed arguments until the end of the file is reached

  {

    printf("\nNickname: %s | Score: %d", displayScores.dispName, displayScores.dispScore);

    printf("\n\n---------------------------------------------------------\n");

  }

  fclose(inFile);

  printf("\nPress enter or return to return to the main menu... ");

  getchar();

  while (getchar() != '\n')

  {

    getchar();

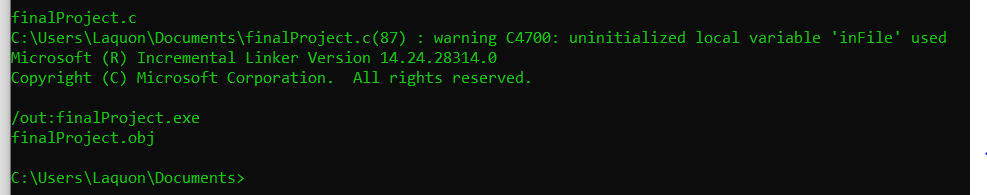
  }

  system("clear");

  main();

}

Compilation



Execution Examples

