HangMang Game

Introduction:

The game developed is hangman. It randomly selects a word from a binary file and the user guessing letter by letter must fill out the word. The user knows the number of letters in the word. Each failed attempt draws a body part on the man. When the man is fully drawn the game is lost. When the word is filled out the user wins.

I choose hangman as I dislike cardgames, and puzzle games. Hangman is a light puzzle game, and I liked the idea of using text files to upload the words, then store in a permanent binary file. The project demonstrates my understanding of input/output files, conditional statements, pointers, and creating an interface for the user that is not an eyesore.

Summary:

The program occupies 808 lines, but with whitespace for readability it is probably around 300-400 lines. The program meets the requirements of the first project. Pointers, structures, pointers to structures, pointers to string arrays, strings, character arrays, binary file input/output, conditional statements, functions with structs/pointers, are utilized in development of the project.

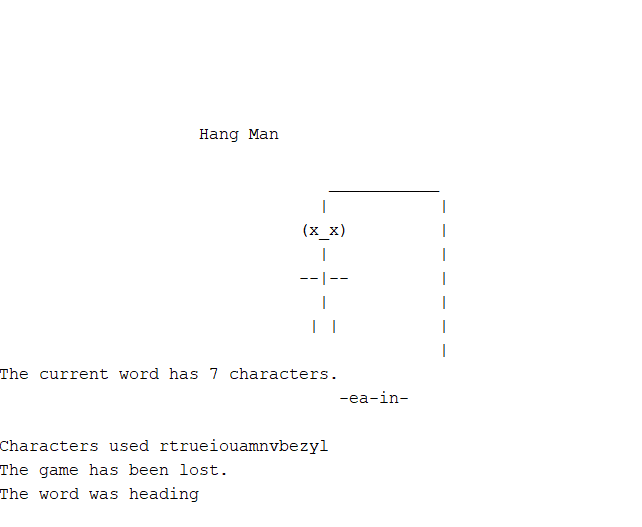
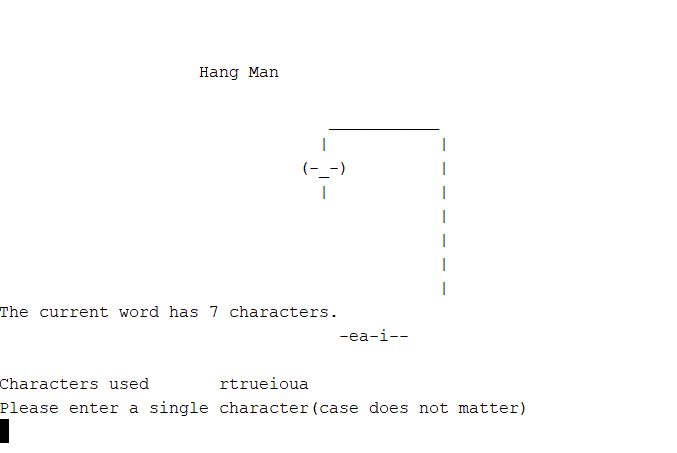
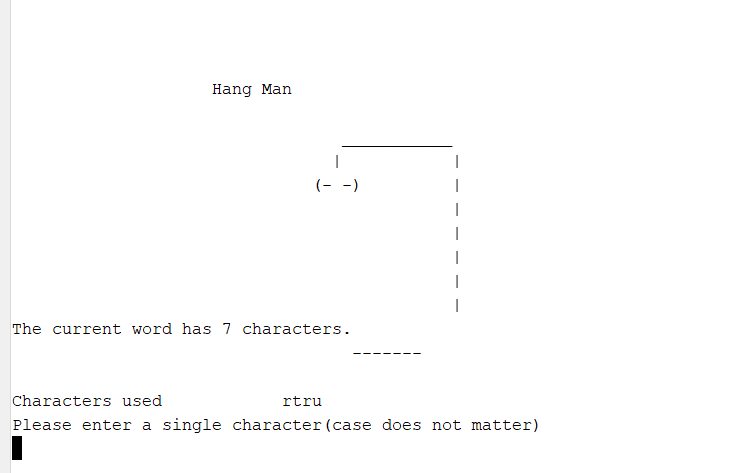
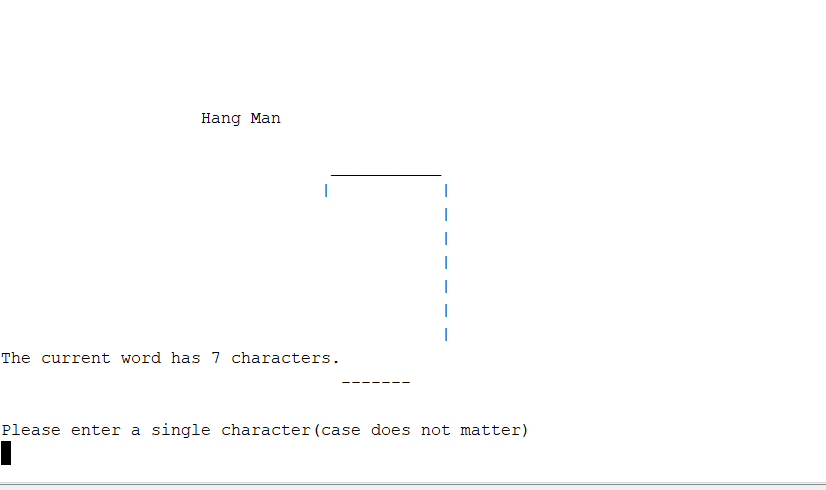
Project was not necessarily challenging, but was time consuming. One area I had some struggle was utilizing the data that I extracted from a bin file into a character array. I resolved the issue by copying the char array up to a null terminator into another char array, then use that character array as a argument to the string constructor. The time consuming came from the needed functions, and excessive conditional statements during the display of the HangMan.

Project took about 6 hours to complete in development, and another 3 hours to complete the documentation process.

Variables #: 24 aprox

Structs # 2 used thru program, some pointers, some not

Description:

Sample input/output 

Flowchart and pseudocode:

Flowchart included in documentation older/ pseudocode of program ideas included as well

Major variables were the 2 struct pointers

word hangMe

used to point to an array of strings that represent words(2d array)

used loading from the bin file, and then transferring one word randomly to the inputW input1.

inputW input1

Used thru all functions except the function to read a bin file, holds various struct info.

A char array ptr to an array of characters with the chosen word.

An int with length of word, blk array to hold only dashes, it is compared to the ptr word character, and the dashes are changed as correct letters are guessed.

Int nGuess which keeps track of attempted guesses

Bool gameStart which records whether the disp function has ran once.

Character array called guess which holds all attempted characters guessed with

Int misses which tracks failed guesses

Concepts:

Structs are used thru the program, conditional statements, loops, dynamic memory, strings,character arrays, pointers.

Input and output of files is also utilized ,but in 3 functions.

inputF

Reads a text file of words for the hang man game. It is a utility function for the program, ran only one time as writeF takes the words and stores them into a binary file.

writeF

Takes the words from the inputF function that were read in from the text file. This function will write those words into a binary file for the inputF2 function to utilize. This also is a utility function and is only ran to change the binary file.

inputF2

This function will read in a binary file which contains the words for the hangman game. These words are then stored into a string array.

References:

I used <http://www.cplusplus.com/reference/> for reference on various functions and their operation.

I obtain my word list from <http://www.manythings.org/vocabulary/lists/l/words.php?f=noll15>