CISC 1610
Louis Boulas
Final Project
Hangman README.txt

1. PROGRAM DESIGN

I started by designing the basic program and tried to include as many functions as I could. To display the hidden word i decided to create an array of size x, where x is the length of the secret word and stored "-" hyphens inside of it. As the player guessed correct letters I replaced the hyphens with that letter in the corresponding place.

For some parts i created a seperate .cpp file in order to test out the functions before incorporating it in the main hangman.cpp file.

for example: when it came to checking the letter inputed vs the secret word they were trying to guess. And i used this process to iterate most of my code.

I also decided to leave the "using namespace std;" in the header (global scope) because the majority of my functions and parts of int main() needed it.

2. TESTING AND DEBUGGING

One major bug i ran into was that i was trying to compare a vector of type string to a string. And got thrown at a long list of compiler errors. Turns out i was comparing a string to a char and therefore it didnt work. So i changed my vector<string>hidden_word into an array of type string. And i could proceed in peace and everything worked.

Another bug that arised was when i tried reading into an array a list of possible hangman words. My vector initially only took the last word of the .txt file and copied it 15 times. after a few modifications it ended up only registering the last word of the file in one instance. And finally i was able to store all 15 words in a 2d array (5 words for the 3 difficulties) through trial and error.

3. EXTRA CREDIT

I have 2 extra features:

a). A scoreboard
I start off the program by letting the user know how the scoring system works. The player can get 1, 2, or 3 points for correctly guessing a word based on the difficulty and gets extra points for any outstanding lives that were unused. However they lose a point if they run out of lives but have the option to reset their lives and try the word again. If they decide not to the score will be displayed and they have the option of trying again with a new word/new difficulty if they would like

b). I wrote 15 possible secret words in a .txt file and prompt the program to read those words in a 2d array before the game begins. So no words were hardcoded into the program. Once the player chooses the difficulty, I called a rand function to random a number between 0-4 (5 words) and and that was the criteria to determine which word in the 2d array to choose and have the player guess.