Raymond Lin

Smallberg

CS31

20 November 2017

Project 5 Report

2 a) There were many obstacles that I overcame throughout the course of this project. One of the obstacles include setting up a multi-file C++ project. Since I had never done that before, I had trouble figuring out how to format the project into multiple files, such that each file could use code from the other files. Additionally, it was challenging to consider all of the requirements in the project. Sometimes, when I implemented one requirement of the project, another part that was already implemented would become silence. So I had to change my code so that both cases could be invoked.

b) This is the pseudocode for the main method:

int main()

{

//declare array of cstrings and populate it using the getwords array

//check if getWords is greater than MaxWords

if (getwords is greater than maxwords) {

//terminate program

}

//check if getWords returns a number less than 1

if (getwords is less than 1) {

//throw an error

//terminate program

}

//asks user how many rounds to play; store it in numrounds

int numRounds = 0;

//check if it is positive

if (numRounds is less than or equal to 0) {

//throw error

//terminate program

}

while (numRounds is not reached) {

//tell user how long word is

// run the function runoneround with the target selected from a random integer

//calculates average

//calculates max and min tries

//report the number of tries taken to get the target word

if (tries is equal to 1) {

//print something

}

else {

//print something else

}

}

}

This is the pseudocode for the runOneRound function:

//runs one round of the game until the user guesses the correct word

int runOneRound(const char words[][7], int nWords, int wordnum) {

//nwords is then number of words in the array 'words'

//wordnum is the position of the target word, in the array 'words'

//preconditions

if (nwords is not positive or wordnum is less than zero or wordnum is greater than or equal to nwords) {

//terminate program

}

while (probe and target are not equal) {

//put probe word into 'probe' cstring

//if probe word length is not between 4 and 6, return an error

if (length of probe is not between 4 and 6) {

//throw an error }

else {

//check if the probe word is in the array

for (through the length of the array) {

if (probe matches a word in array) {

//set a variable to true and break

}

}

if (not in array) {

//throw an error

}

else {

//make a copy of target cstring so that we can modify it

//loop through the cstrings and count stars

for (through length of probe word) {

//count stars

if (character in target matches character in probe) {

//add to star counter

//change the star in probe and target to different characters

if (if all letters match) {

//return number of tries

}

}

}

//count planets

for (//loop through the letters of the probe word) {

for (loop through the letters of the target word) {

if (if the letters are the same and they are not at the same position) {

//plus one planet

//change planet in target to a different character

break;

}

}

}

}

}

}

return 1;

}