Jason Tran

CS31 Project 5

11/14/22

A)

The most challenging obstacle I faced was figuring out how to count the silvers for the probe word. When I first tried to do it, it kept adding too many silvers to the final score. So basically, I needed to find a way to make sure only one silver was given to the letter. This was hard to do, but basically I just made checking-functions and arrays that accounted for the positions that were already used.

Another obstacle I faced was figuring out what the spec wanted us to do, as in what to write and where to write it. It was confusing in the beginning because I didn’t know what getwords did or how to use it in my program. It was also hard figuring out how to implement multiple files into 1 program. Also I had to consider the many specific cout statements the spec wanted to make sure I didn’t write any wrong or even put any in the wrong place. Such as what PlayOneRound would write or what the function caller would write.

One other notable obstacle was figuring out how to loop asking the probe word again if the user inputs it correctly. It kept looping repeatedly, despite the word being in the correct format. I basically had to make sure that once the probe word was in the correct format, to turn off the loop asking for the word, which was a quick solution with Boolean statements.

B) My main fucntion basically just asks for rounds, checks if it is valid, then does a loop for how many rounds there are calling the PlayOneRound function each time. Then it does some calculations for the score with what playoneround returns.

My PlayOneRound function repeatedly asks for a word until they input a correctly formatted one. With the correctly formatted probe word it checks if it is in the list and calls gold and silver count functions.

My goldCount funciton is pretty simple. It loops through the probe word and checks if the letter and position are both the same in the words. Then replace the letter with a space. My silverCount loops through both words and checks if the letter is in any position. If it is then added to the silver count.

IN PSEUDOCODE:

**goldCount:**

Repeatedly:

if position and character matches answer:

Add 1 to gold

replace both characters with a space

return gold

**silverCount:**

Repeatedly through probe:

Repeatedly through answer:

If letter of probe is in answer and position are not equal:

Change the letter of answer to space

Add one to silver count

Return silver

**playOneRound:**

Ask for input:

Check if input is valid

If valid:

Check if input is in list

If in list:

Add to score

Call goldCount

Call silverCount

If probe is same as answer:

Return score

Else:

Output golds/silvers

Reset answer,probe strings

Set to invalid

**main:**

Ask for rounds

Repeatedly for # of rounds:

Get random number for word

Call playOneRound

If score is a max or min:

Change max or mix total

Add score to a running count

Calculate average

Output statistics