Nora Riches

Sparsh Rawlani

CS 172-1

Final Project Documentation

**Description**

We implemented a typing tutor that showed a user a file for 60 seconds, asked them to copy it and then calculated statistics that were output to another file. We took the final project that Kara and Nora completed for the fall semester and implemented more complex structures that were learned in CS 2.

The entire project is located in: \\cs1\CS\_Projects\2019\_spring\_CS172-1\cs172\_final\_nriches21\_srawlani22

**Outline of the Project**

The majority of the project was spent in the student class that was created. The UML diagram for the student class can be found below. The member variables of the student class are a string for the name, a vector to keep track of the accuracy of each attempt, a vector to keep track of the words per minute calculation of each attempt, and an integer for the level of difficulty. There were also a number of methods that interacted with these various variables. The first two methods were student constructors. There is an empty constructor that sets the name to be empty and the level to be 1. The other constructor takes in a string that is set to be the name and sets the level to be 1. The next methods are interface methods of getName, getLastAccuracy, and getLastWPM. Each of these methods simply allow access of the variables outside of the class.

There are more complicated methods that were implemented in the student class as well. The levelDecider function asks the user to choose what level they want to do and assigns the level and a new filename. There is also an openFile function. This is a simple function to try to open the file using a file stream and an input string, which is assigned in the level decider function, and showing an error if it failed to open. The showUserFile function shows the file that was opened using the openFile function and then asks the user to copy the file. This is the most complex method in the class because we has to time and grab user input at the same time. This required the use of the GetAsyncKeyState method. This method tells if there is a key being pressed. By iterateing through a while loop for 60 seconds and constantly checking if there is a key being pressed, we were able to push a letter into a string each time a specific key was pressed. This took a lot of research and trial in order to operate the method correctly. After the user has copied the contents of the file for 60 seconds and it is pushed into a string, we were able to compare the two inputs and complete calculations. We calculated accuracy and words per minute each time a student completes the test. Once those things are calculated they are pushed into their respective vectors. The playGame function calls the level decider, open file, show user file, calc accuracy and calc wpm functions at the same time because that is what needs to occur each time the player wants to play a round of the typing tutor. The last method in the student class is a print function to print each round that the student has played as well as the average statistics.

Since most of the content is in the student class, the main function simply asks for the name of a student and then calls the constructor, play game, and print functions.

**Assumptions**

NOT SURE WHAT ASSUMPTIONS WE HAVE

**UML Diagrams**

The Student class UML diagram is in the CS Projects Folder

**Use Case and Manual**

When you run the code, it will ask for a name. The user must input their name. After that, it will ask the user to input which level they would like to be at. There is an option to push 1, 2, or 3 where 1 is easy, 2 is intermediate, and 3 is hard. Immediately following a paragraph will appear in the terminal and the user will be promoted to copy the paragraph for 60 seconds. After the time has run out, the turn will end and the user will be asked if they want to play again. If they push yes, the code will run again. If they push no, a file will appear with the accuracy and statistics stats for each try and an average will also appear.

**Works Cited**

These websites instructed on how to use the GetAsyncKeyState method: <https://stackoverflow.com/questions/32031368/c-how-do-i-read-in-input-for-a-certain-amount-of-time-only-can-i-do-so-withou>

Website to decide how to measure time:

<https://stackoverflow.com/questions/22387586/measuring-execution-time-of-a-function-in-c>

Reference for the string append method:

<http://www.cplusplus.com/reference/string/string/append/>

Reference for UML Diagram:

<https://online.visual-paradigm.com/diagrams.jsp#diagram:type=ClassDiagram&gallery=/repository/e3676c7c-2bb5-44a1-a600-4502fd09e99e.xml&name=Car>