**Overall Summary**

Our objective for this game was to create a unique, easily comprehensive speed typing game that will track how many words the player types correctly and display the player's wpm speed. The player will be able to choose a difficulty level and as they play the typing game, based on the sentences the player types correctly, the frog will move across the screen to his home. The goal of the game will be to get the animal to the end as quickly as possible while aiming for a high wpm. Then, the player will be given an option to restart the game which will bring them back to the page that they saw when they first opened the game.

**Front-End:**

Instructions on how the game works are directly displayed on the screen which allows the player to easily grasp how to play. The user would have to click on the “Start Game” button and choose a difficulty level ranging from “Easy,” “Medium,” and “Hard.” Based on the user input, the length of the game changes to accommodate for the user’s choice and the user would input sentences that are displayed on the screen.

The text changes to a teal when typed correctly; however, the text will be red and underlined when typed incorrectly. If there is even one mistake in the input, the player will not be able to continue; instead, they would have to go back and fix it. If one was to type the sentence wrong, the game will not proceed as a mechanism to limit the behavior of the player. This also prevents crashes and any strange elements that could be on the screen. And to prevent the user from cheating by copy and pasting the sentence, we disabled the pasting function into the input box.

The entire game is frog themed. So, everything is green-blue based to emulate a pond and frogs jumping. Along with that, the sentences were created based on random facts about frogs. When the input text box is selected, the box is outlined to ensure the user that they have the input text box selected. When the input text box is not selected, the outline is not there. Everything is centered and has the necessary space to not overlap and have a bit of space between each element. The box for the sentences generated changes to fit the sentence generated for a more sleek look.

**Back-End:**

When the sentence is correctly typed into the input text box, the sentence will disappear and a new sentence will appear. This happens while the score increases by 1. There will be a timer running in the background while the game is in progress and will stop once the player inputs the amount of sentences that relate to the level difficulty. All the loops/intervals are cleared by the end of the program which prevents potential crashes from infinite loops. Our program can be read mostly linearly and we do not have any overlapping functions or infinite loops that our program can run more smoothly. Most of our functions and id have comments next to them which give brief descriptions of their task.

We used a conjunction of HTML, CSS, and Javascript to create this game with HTML being the “base” file that we used to compile the CSS and Javascript together. The CSS file helped us design the interface that the user would see when they start and progress through the game while the Javascript file was the main file that we coded all the functions in. Having these different files helped differentiate the objectives and functions we wanted to accomplish while also having an aesthetic interface that the user could interact with.

**Notable Functions**

One objective we wanted to accomplish with this game was the color-correcting function through which we would display to the user that they had input an incorrect letter by changing the color of the letter (red and underlined) while also displaying the correct characters with a different color (green). We provide the player with the option to go back and change their answer if typed incorrectly but the player would not be able to proceed to the next sentence unless their sentence was typed exactly identical to the sentence displayed on the screen.

Another vital function of the game is that as the player gains a score from completing a sentence correctly, we wanted to have a graphic of a frog jumping over a rock to indicate that George is on his way home. With the conjunction of HTML, CSS and Javascript, we were able to display our frog George on the screen that interacted with the player’s score increasing. We created elements for the frog and the rock in HTML that we could code to simulate the jumping and sliding in CSS and then used a function in Javascript to edit those elements to make our graphics interact with the player’s score.

**Marketability:**

Yes, people would use this project. This is because typing is a very important skill in the current age. Most professional writings and communication is done through a keyboard online. Thus, this would be a good way to introduce someone to typing. The user of this game would learn how to type accurately and fast overtime. Our project also stresses the importance of accuracy over speed which isn’t done in most typing games. This forces the user to adapt to fixing their mistakes as fast as possible rather than skipping over them. Along with that, this game also tests endurance as the player would be trying to type as fast as they can while being accurate for multiple sentences. This is done so that the player can also get a general idea of how long they can last at a certain speed. There are different levels that can test this endurance; based on what level the player chooses at the beginning of the game, the game will progress longer as it will take more correct sentences to get George from one end to the other and.

**Challenges**

We decided to make a web browser game using HTML, CSS, and Javascript with the rationale that it would be easier to manipulate the user interface as these three languages are compatible with each other. However, no one in our group had experience with coding languages other than C++ (which we learned throughout the course of this class) and MATLAB (through the EK125 prerequisite). So quite literally, over the span of the past half week, we learned to code in these three languages with specific objectives and functions in mind for the game we wanted to create.