Journal Report 9
11/11/19-11/14/19
Jialin Tso
Computer Systems Research Lab
Period 2, White

Daily Log

Monday November 11

Continued researching tkinter and how to make GUI for graphing calculator.

Tuesday November 12

Installed tkinter and realized that ”tkinter” is for Python 3 while ”Tkinter” is for Python 2. Experimented with different types of graphing calc GUIs after becoming inspired by others’ codes on public repositories. Watched videos to gain idea on how to create an effective calculator GUI.

Thursday November 14

Discussed with Dr. White the best way to create calculator GUI. Decided that calculator should be designed for physically impaired people and later, functions for blind people can be added. Chose GUI with buttons rather than command prompt. Decided to change color of button to show which buttons on the calculator get ”pushed”. Used four-function calculator program to create large dictionary of buttons that would enable color changes.

Timeline

Date	Goal	Met
Oct 14	Convert all calculator-recognized function answers to words	Success!
Oct 28	Add more functionalities to program	Program can generate random numbers n times, convert most decimals to fractions, evaluate functions, graph many equations, find intersection of two functions, and find a zero of a function.
Nov 11	Begin creating GUI for calculator	Researched layout for calculator buttons and began creating dictionary of buttons to enable changing their colors.
Nov 18	Add all calculator-recognized buttons to a dictionary and correctly format on calculator.	
Winter goal	Create GUI for calculator and if possible, use SpeechRecognition to ensure program will recognize user's voice and correctly transcribe sound to text.	

Reflection

Now that I have most of the essential calculator functions added to my code, I've decided to take steps in the direction of making the calculator into a more ”concrete” product. For instance, I spent much of this week brainstorming and researching how to create a calculator GUI and debating between whether the calculator would be an emulator (with buttons) or just a command prompt. With advice from Dr. White, I decided that creating buttons would be most beneficial because it would be visually similar as a TI-84 and it would allow the user to easily edit their command (since I envision the program will be dynamic and will output what the user has said to the screen a few seconds behind what the user is currently saying). If I use a command prompt and the user makes a mistake, the only way to edit it would be to say the entire phrase again.

I initially wanted to use a simulated clicker that would press the buttons corresponding to what the user says, but I couldn't find anything about it online and Dr. White recommended that I briefly change the colors of the necessary buttons.