Journal Report 1

9/3/19-9/5/19
Jialin Tso
Computer Systems Research Lab
Period 2, White

Daily Log

Tuesday September 3

Learned about Python SpeechRecognition through ”The Ultimate Guide To Speech Recognition with Python” by David Amos. Had a conversation with Mr. White about how to approach project without calculator emulator. Learned how to divide project into steps and began research on Python expression compilers. Remembered that Python has a handy dandy eval() function that can compute nearly all numerical expressions. Obtained ROM image for Wabbitemu and was able to get emulator to work.

Thursday September 5

Discovered a website called https://mathtalk.com/ and watched a video on how MathTalk works to graph functions by voice and uses a simulated clicker. Began work on transforming text to math expression to correct answer. Learned about word2number Python library that converts positive number words to numberic digits up to 999,999,999,999. Hard coded elif statements to transform each word (e.g. ”plus” to ”+”, decimal ”point” to ”.”, ”power of” to ”**”, ”square root of” to ”math.sqrt”, ”left parentheses” to ”(”) to something recognized by eval(). Due to current inaccessibility to word2number, I did not input numbers as corresponding words.

Timeline

Date	Goal	Met
Sept 3	Install Wabbitemu and convert text to expression for basic functions	Emulator works and program can calculate many functions (w/o number words)
Sept 9	Install SpeechRecognition and word2number; text to expression for factorial, logarithms, permutations and combinations, and other remaining useful functions	
Sept 16	Solve integrals, derivatives, and matrices; find 1-var stats for user-inputted list	
Sept 23	Begin experimenting with SpeechRecognition and implementing speech to text	
Sept 30	Find 2-var stats for user-inputted lists and find regression equations from lists	

Reflection

I was successful in converting many of the common function words to their Python symbols. To account for the user inputting phrases with filler words like ″of,″ ″the,″ and ″to,″ I hard coded the program to ignore each of them. I hope to add more functions for the next few weeks, but I'm wondering if there's a more efficient way (e.g. a library) to do so instead of having to manually convert all of them.

I am currently trying to solve the problem of installing pyCharm or another comparable IDE. However, my license has expired so I'm working around the issue by using repl.it, rendering me unable to install libraries.

Example of input and output of current program:

Input: absolute value of left parentheses negative 4 point 5 plus tangent of left parentheses pi divided by 4 right parentheses right parentheses to the power of 2.

Expression output: math.fabs(-4.5+math.tan(math.pi/4))**2, Answer output: 12.25.

Although it may seem burdensome and unnecessary to frequently say ″left parentheses″ and ″right parentheses,″ where the parentheses falls does matter! It is of the utmost importance that the program doesn't confuse math.fabs(-4.5+math.tan(math.pi/4)) with math.fabs(-4.5)+math.tan(math.pi/4)), which are respectively 3.5 and 5.5.