Alex Chen

CS 31

Project 3 Writeup

1. One obstacle I faced was editing the plotHorizontalLine and plotVerticalLine methods that I wrote in the warmup in order for them to function properly for phase 3. The original forms of the methods did not have to deal with negative distance values, which was more confusing than hard to fix – thus, I created two new functions, plotHorizontalLine2 and plotVerticalLine2, which could deal with negative distance values, in addition to a few other adjustments. In addition to writing these new methods, another obstacle I faced was correctly parsing through the characters following ‘h’ or ‘v’. I spliced commandString using brackets following commandString, yet it would not splice correctly. This obstacle trumped me for a while, but I resolved it by splicing commandString with the substring method of the <cctype> library. For some reason, this fixed the issue. One last obstacle that stumped me for a while was when I would enter a syntactically incorrect command string that contained an ‘f’ or ‘b’. The mistake was that in the blocks within the conditional (if character == ‘b’ / ‘f’) (🡨 pseudocode), I forgot to change the badPos variable accordingly. Setting it equal to i+1, where i represents the index in the string at which the “cursor” is at solved the issue.

Overall, the largest difficulty in this project was understanding all the minute details in the spec. There were a lot of small things to incorporate and keep track of, but definitely taking a incremental approach to solving each portion of this project was extremely helpful.

1. My code begins with the inclusion of all libraries and the grid.hpp header, then follows with the declaration and initialization of constant integers and the prototypes of used methods. The programming logic begins with the main function, in which the user can enter a command string. If the string is empty, then the user essentially breaks out of the function and exits. Else, performCommands is called with the inputted string as one of the arguments, and the already initialized plotChar, mode, and position, as the other three arguments. performCommands can either return a 0, 1, or 2, and the following switch statement deals with each case.

If 0 returned (commandString is of proper syntax, graphing string will work)

Draw grid and break out of outside loop

If 1 returned (commandString has syntax error)

Print out error, break out of outside loop

If 2 returned (commandString is of proper syntax, but graphing string leaves grid)

As for the actual performCommands method, here is pseudocode to explain it. Note that the pseudocode does not include the incremented i values, which I utilized to parse through the command string, and also does not keep track of the badPos variable, both for the sake of simplicity.

Row and column = 1, 1

Loop through command string, one character at a time

If char is c

Reset row, column, plotted character, and mode

Clear grid

Else if char is f

Look for following character

Set plotted character to following character, set mode to FG

Else if char is b

Look for following character

Set plotted character to following character, set mode to FG

Else if char is h

If following character is ‘-‘

If following two characters are digits

Convert two characters to int

Make int negative

Call plotLine to plot the horizontal line, passing int as one of arguments

Else

If following character is a digit

Convert character to int

Call plotLine to plot the horizontal line, passing int as one of arguments

Else if char is v

Follow same protocol as if char is h, except use plotLine to plot vertical line

Else there must be a syntax error

Return 0, because if this point is reached, everything was successful.

1. 1. H11v4h5v10
      1. Ensures h and v commands work properly, when both one or two following characters are digits, and also ensures the final (r,c) value is retained after each subcommand of the command string.
   2. C
      1. Will show if c properly clears grid.
   3. V2f^h3v4
      1. Shows if previous c command properly reset plot character to ‘\*,’ and if f command works properly.
   4. H-3
      1. Shows if method can properly return index of leftmost part of error in the command string – error is in command string requires plotting to leave grid.
   5. H5v2h4b^h-5
      1. Shows if method can properly deal with negative symbols after string and if the ‘b’ command works as required.