a. Obstacles

One of the major things I had to deal with when doing this project was keeping track of what exactly everything did, and where everything was, because it was a much longer program than our previous projects. I made sure to keep cleaner comments as I programmed, and that really helped me.

Another problem that took me some time was translating an inputted integer in a string into the actual integer, and I eventually solved that by creating a separate method to translate the character into an integer.

Lastly, when parsing through the string, it was difficult to keep the iterator at the correct position throughout the iteration of the entire string, and that frustrated me for a little while, but I managed to work through it and get my program working properly.

b. Description of the design of my program

main method:

* Set up the default values for the grid
* Repeatedly prompt the user for a command string
* If empty, exit
* Otherwise, call the performCommands method to perform the function and get a value back stating what happened
  + If returns 0, it plotted properly and draws the grid
  + If returns 1, there is a syntax error
  + If returns 2, there was a command that wasn’t able to be plotted

performCommands method:

* Set the starting position in the top left at (1, 1)
* Repeatedly:
  + Get the first letter of the command
  + If it’s a bad letter or command, return that error
  + Otherwise, for v and h:
    - Get up to the next three characters, with two digits maximum after a possible negative sign
    - Plot the line using the plotline function
    - Increment to the right index based on how many characters were parsed
  + For c:
    - Clears the grid and resets to defaults
    - Increment the iterator
  + For b and g
    - Set the mode to either background or foreground
    - Set the plot character to the following character
    - Iterate a total of twice

plotLine method:

* Checks if the inputs for plotting a line is valid
* Plots an either horizontal or vertical line depending on what it is told from the inputted coordinates and length of the line using the character inputted

c. Various Test Data

Syntax error testing (similar to the example syntax errors)

|  |  |
| --- | --- |
| B#v+25H?V3! | Syntax error at position 4 with the + symbol |
| F&h | Syntax error at position 4 because it expects a digit or – after h |
| 10 | Syntax error at position 1 because no command |
| c12 | Syntax error at position 2 because it expects a letter for the command. You can’t start a command with a number |
| Tysm123 | Syntax error at position 1 because T is not a specified command |
| V-1-2 | Syntax error at position 4 because - can’t start a command |

Other test cases

|  |  |
| --- | --- |
| h12V3H-1B@v-3 | Test upper and lower case v and h, symbol switching, change to background plotting, negative inputs (part of the example test code) |
| v2b h12fHh1fih0 | Another part of example test code to see if it retains the previous drawing and switches back to foreground mode. Also test to see if it accepts the space character |
| H30 | Test to see if the program recognizes that the line goes out of the grid |
| v20 | Test to see if the program recognizes the line goes out of the grid |
| c | Just test if basic clearing works |
| H8v10b@h-2v-2F3h-3V10 | Longer string with all commands except clear in it, both upper and lowercase |