**Program Design**

*Problem*: The problem to be solved is implementing a program of Langston’s Ant. Users will choose if they want to play or quit, and then choose the size of the board (rows and columns) for the ant to move on, and the number of steps for the ant to take. Users will then choose if they want the ant to start at a random location. If the user wants to choose the location, then the user will enter the starting row and starting column.

*Structure*:

Ant class header file

Ant class header file will include an enumerated data type for the different directions an ant can take (up, down, right and left); a character data type for color (black and white), and integer data types for the rows and columns on the board. The header file will also include getter and setter functions for collecting the row and column inputs from the user (provided they don’t choose the random generation for starting point) and returning them back to the program.

Ant class implementation file

The Ant class implementation file gets and sets the columns and rows based on input from the user.

Main file

Mainfile includes following inputs from the user.

*Test Table*:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Input Values | Driver Functions | Expected Outcomes | Observed Outcomes |
| First input: Value not 1 or 2 | 3 | main()  while loop | Prints error message and loops back to the original question | Prints error message and loops back to the original question |
| Choosing start row: value outside margins (5 – row) | 6 | main()  while loop  getStartingLocation1 | Prints error message and loops back to the original question | Prints error message and loops back to the original question |
| Choosing start column: value outside margins (5 – row) | 6 | main()  while loop  getStartingLocation1 | Prints error message and loops back to the original question | Prints error message and loops back to the original question |
| Ant takes steps and hits edge of the board (6 steps right/left, or up/down on a 5x5 board) | 6 | main()  getStartingLocation1 | ant goes up or down depending on location on the board (row, column, - 1) sends ant backwards | ant goes up or down depending on location on the board (row, column, - 1) sends ant backwards |

**Reflection**

I started with up, down, right, and left for my ant to be directed but for some reason I was receiving errors that said this was too ambiguous. On github, I found that I should use north, south, east, and west instead.

I originally started with a menu class header and implementation file, but had trouble implementing these in my main so I included the menu inside main() function.

I also initially started with having the display print out the physical board “\_” “|” but kept getting segmentation faults, so removed them. I believe it has something to do with the way I moved my ant on the board but despite numerous research attempts online, I couldn’t find out how to include a representation of the physical board and also avoid segmentation errors.

I found this project extremely difficult. Even after a week, I still didn’t have everything figured out. This definitely gave me a good idea of what I need to do to prepare for projects in the future since they are so in depth and long. This is definitely the most difficult thing I have put together so far.

I had to google how to do the random number generation as I didn’t remember how to do it.