CS_162_400_S2019
Project 1
Susan Hibbert
ID# 933913975

Langton's Ant Simulation

Design Description

Display menu

Ask user to press 1 to run program or press 2 to quit

While user enters invalid input

Re-prompt user to enter valid input

If select 1

Ask user to input information for ant simulation

While user enters invalid input

Re-prompt user to enter valid input

Initialize member variables of Board and Ant objects

While user enters invalid input

Re-prompt user to enter valid data

Run ant simulation for specified number of steps

Initialize loop counter to keep track of steps

While count is less than steps

If ant is on white space

Turn ant's orientation right 90 degrees

Change space to black

Move one step forward

Place '*' in square

Display board

Increment count

If ant is on black space

Turn ant's orientation right 90 degrees

Change space to black

CS_162_400_S2019
Project 1
Susan Hibbert
ID# 933913975

Move one step forward

Place '*' in square

Display board

Increment count

Ask user if want to run ant simulation again

If select 2

Quit Program

Display goodbye message

Test table (test plan, results)

Test Case	Input Values	Driver Functions	Expected Outcomes	Observed Outcomes
User does not enter an integer at beginning of the program or enters a value other than 1 or 2	Input is a not an integer - character, float, mix of characters and integers Input is an integer other than 1 or 2	Display_menu() While loop prompts user to enter 1 or 2 Int_input_val() function Do-while loop prompts user to enter an integer	Loop back to the question prompting user for correct input	Program loops back to the question prompting user for correct input, as expected. Does not do this for float values in int_input_val() function
User does not input integers when entering information for the ant simulation	Input is a not an integer - character, float, mix of characters and integers	Int_input_val() function Do-while loop prompts user to enter an integer	Loop back to the question prompting user for correct input	User is prompted to re-enter an integer. Does not do this for float values
Board size too low	Input ≤ 1 for row and/or column	Initialize() function of Board class If-else statements	Board defaults to 10 for row and/or column	Board defaults to 10 for row and/or column and program continues as expected
Board size too high	Input > 100	Initialize() function of Board class If-else statements	Board defaults to 100 for row and/or column	Board defaults to 100 for row and/or column and

CS_162_400_S2019
Project 1
Susan Hibbert
ID# 933913975

				program continues as expected
Ant placed off board	Ant's placement row and/or column is not within the confines of the board	Place_ant() function of Board class If-else statements	Ant placed in centre of the board	Ant placed in centre of the board and program continues as expected
Ant moves off board during simulation	Ant moves off edge of board during ant simulation	Move_ant() function of Board class If-else statements	Board wraps and ant will appear on other side	When ant moves off any edge of the board the board wraps correctly. The ant appears on other side of the board as expected
Steps are too high	Input > 10000	Set_steps() function of Board class If-else statements	Steps are limited to 100000	Steps set to 100000 and program runs as expected
Steps are too low	Input ≤ 0	Set_steps() function of Board class If-else statements	Steps are defaulted to 10	Steps set to 10 and program runs as expected
User runs program again	User presses '1' to run program again when menu prompts at end of program	Main() While loop prompts user with menu options while choice is 1	Program will run again as normal	Segmentation fault – will run valgrind to find problem
User quits program after running program once	User presses '2' to run program again when menu prompts at end of program	Main() Exits while loop	Program will exit and display goodbye message	Segmentation fault -will run valgrind to find problem

<u>Reflection</u>

I encountered several issues which prompted several design changes.

I left out several important details from my initial design, such as how to keep track of the ant's orientation as it progressed through the simulation. I also implemented my input validation functions differently to what I described. Some user input was validated in the Board class member functions instead of my input validation functions as I found it easier to implement them in this way.

CS_162_400_S2019
Project 1
Susan Hibbert
ID# 933913975

As per my test table above, I ran into a few issues with segmentation faults. These occurred after the program ran once and the user either selected to run the simulation again or quit. I realized that I needed to create a new Board object for each new simulation and direct my pointer to this new Board object each time. I was getting segmentation faults as my pointer was pointing to a Board object that had been deleted after the simulation had completed. I also added a default constructor for the Board class which initialized the data member of the Board object as I was having issues using an uninitialized pointer. When I fixed these issues I no longer got segmentation faults which I checked using valgrind.

I also ran into issues trying to validate float input values. Initially, if the user entered a float value it would become truncated and I would use this value in my ant simulation which I recognize is bad practice. After I saw the Piazza post that float values input by the user had to be rejected, I had to re-write my entire input validation function. After re-writing it, it now rejects float values.

I initially felt quite overwhelmed when I read through the specifications for Project 1, however by breaking down the assignment into several parts and addressing each one in turn I started to feel more in control. I feel this approach will help me with future assignments.

Additionally, I learned that although it feels like you have a long time to complete a project, you really don't! I encountered a lot of debugging issues which took me much longer to fix than I had planned. The take home message is to start projects early!