Nick Roy

Project 1 Reflection

4/6/2017

Before writing the code for the Langton's Ant program, I started out designing some requirements for the program, and the functions/classes that would be needed. Although the instructions for project 1 provided some general requirements, there was a lot more flexibility with the way the program was implemented. The program would also require some additional planning to implement functions for a menu that can be used throughout the course.

There were four main components to the program including the ant class, the board functions, the menu functions, and a main file that would bring everything together. I started out by writing out the steps for what would be required for the program. The user would need to be prompted for information, the board would be created, an ant would be placed either randomly or per the user input, and then the program would loop through the logic moving the ant.

Creating the board was easy enough after reading through the posted material about initializing a 2d array pointer. After creating the board an initializing the values to all zero's the other function would display the board and convert the 0's and 2's to black and white squares and the 1 to the ant character. This would be one of the only struggles related to creating the board is the count by 0 when working with the arrays. At first I tried keeping up with what the user entered for the and showing the appropriate values, but ended up showing the information based on the count by 0 that the array values were located at.

Most of the time was spent on the ant class and figuring out the best way to have the ant move, and then deciding how to handle the cases of the ant turning into the wall. The ant class would also need to handle tracking the color of the square that the ant was on, and swapping out the square after it moved off of it's square. Part of my design of the project involved listing more in depth steps for writing the program and listing out some functions of the ant class. Looking over the project design after completing the project there are not any functions to deal with the color of the square that the ant is on, and although there was some notes about a default constructor, I ended up needing more items in my constructor.

The other problem was coming up with a way to track the color of the square the ant was on and how to swap out the color. The design document does not have a lot of notes about how the program would track this, The caused some problems as originally my program was trying to get the color of the square the ant was on, which was always returning 1 as it was looking at the ant's value. Part of this was solved by adding an item to the constructor to set the initial square to white, moving the ant's position by 1, getting the color of the square, and then setting the ant in it's place.

In regards to creating the menu functions and input validation, I wasn't sure exactly what would be needed going forward, so I wanted to keep things as high level as possible. Each function is passed a single string that acts as a prompt, and then is passed either two char's or two int's. These two char's or int's are used to determine if the number the user entered falls within a specified range, as well as it then performs validation that the user is entering a char or int.

Testing started out with basic items to ensure that everything was functioning properly. One original issue that I ran into was generating the random numbers if the user does not want to decide where to place the ant, or they enter incorrect coordinates. My original random number function would occasionally cause a segmentation fault as a result of it using the values a user entered for the max size of the table, and not the actual count by 0 value that the 2d array was. Other tests involved adding in additional debug statements into the switch statements to ensure that the ant was moving in the proper direction based on the color of the square that it was on. The final test included comparing results from my program's outputs to the sample animation on the Wikipedia article for Langton's Ant to make sure that the rules of the program were being followed correctly.