Program name : Project1_Design_Reflection_Schaefer_Kristin.pdf
Author : Kristin Schaefer
Date : 04-14-2019

source of formatting and document structure : ***JasonLong_Final_ReflectionJournal.pdf***

Design

Looking back at my design plan after completing the Langton's Ant program, my design plan was fairly accurate, however, it was definitely missing important 'pieces of the puzzle', which I later added during the programming process. Regarding functions, I was correct in that I would need a constructor, ant location, ant orientation and ant simulation function. As I was designing the Ant class though, I realized the importance of adding functions for the ant's movement, such as moveForward(), turnLeft(), and turnRight(). These functions not only made the code more legible, but they also helped me to understand and to break down the movement of the ant. I think in the future I will definitely try to break apart more my functions based on action verbs like move, set, rotate, place.

One part of my design plan that I consider too vague was my description of the menu function. In designing the menuStart() and menuReplay() functions I struggled to understand how to allow the user to quit the program. After asking for help on Piazza, I received a great response from a teaching assistant that I should not try to exit the program from within the menu function, but instead use a boolean variable and return it to the main function. Then in the main function I used the boolean variable in combination with do-while and while loops, with all routes of the program feeding in to return 0. So ultimately I learned that it is best practice to structure your program to return 0!

Another area that my design plan did not cover well was the structure of the input validation necessary for testing all of the integer inputs in the program. Setting up the input validation took a surprising amount of time to understand! It was only after browsing through Piazza, Stack Overflow and Youtube that I came up with a solution that was working. I think this program provided a good first exercise in trying to understand input validation though, and I will reuse and improve the code in future assignments.

Testing

One of the errors I encountered while trying to compile my program was related to my Ant constructor and Ant destructor. I tried to set up a destructor with a loop to delete the pointers of the dynamic 2d array. I noticed that it was not necessary to call a destructor like a normal function, but instead to create a pointer to the object and then delete it at the end of the program. This should have in theory called the destructor. I was not able to get this to work though, so instead I created a separate function with the same for loop to delete the pointers. This seemed to work okay, as I was able to use the same object for the replay simulation loop without any problems, however, I don't think this is best practice, so I would like to understand in the future how to better set up destructors.

The second part of testing I struggled with was understanding the order and process of the ant steps. After sketching out the simulation I understood the order of the steps, but then organizing these steps within different functions was incredibly challenging! I went through several iterations of testing where to place the ant '*' and where to print the board. It simply took a lot of trial and error in combination with meticulously tracing every step of the program.

In general, I had good results with input validation testing. I used a combination of cin.good(), cin.clear() and cin.ignore(). For my testing plan, most of the inputs passed the validation test and returned the anticipated results, however, there were a few small errors I could not resolve. The main issue being when I entered floats, the error output message always printed twice.

Improvement

Looking back, completing Lab1 prior to Project1 was critical to my understanding of dynamic 2d arrays. I now feel more comfortable working with dynamic 2d arrays and with dynamic memory allocation within a basic program. I still feel that I need more practice with this subject, however, so it's something I'll devote more time to in the readings and lectures.

As previously mentioned, I would like to understand how to implement constructors in combination with destructors. This is quite a basic concept, that I am still not completely comfortable with. Regarding my testing plan, there were a few inputs that I could not figure out how to resolve, such as floats, so I will work on finding better solutions for the next assignment.