

Chermaine Cheang

Assignment 1 Reflection

My original design was able to move the ant successfully for four steps, on the fifth steps, the ant which is now standing on a black cell should turn left 90°, but instead the ant turn right. This is because the variable to hold the cell's color that the ant is going to move to is not updated correctly. To resolve this, after determining which cell the ant is going to move to, I store the color of that cell to my color variable in Ant class. To get the color of the cell the ant is going to move to, I will call the `oriColor()` function of Grid class, which is a function I added after discovering this error, and pass in the x and y coordinate where the ant will move to. By storing the color of the cell the ant is moving to before updating the ant location, the program was able to run successfully.

Another problem that I encounter is that my design does not take in account the possibility that the ant will move out of the grid boundaries. To resolve this, I revised my `moveAnt()`, `whiteMove()` and `blackMove()` function from void functions to bool functions. This is to determine whether the propose movement of the ant is valid. Hence, before I update the location of ant in `whiteMove()` and `blackMove()`, I checked whether the propose x and y coordinates are smaller than 0 to prevent ant from moving beyond the top and left border of the grid, and I checked whether the proposed x and y coordinates are greater than the number of columns and rows available in the grid to prevent the ant from moving out of the right and bottom border of the grid. Then, in my main function where I have a for loop that will loop through and call the `moveAnt()` function x number of times based on user's inputted number of steps, I added a condition in which if `moveAnt()` return false, I will force exit the loop. This is to stop the system from running when the ant is already out of the grid.

Another error in my design is that I used string instead of char by mistake. To resolve this, I changed all string variables and functions to char. The reason they should be char is because my 2D array is an array of char, not an array of string. In addition, I ran into problem creating my Ant object. Originally I thought of creating them after I received all the input necessary from the user that is why I have different constructors for each case. However, this did

not work because my ant object is declared inside a loop which makes it inaccessible outside the loop where it was originally declared. In order to resolve this, I have to declare my ant object first before asking user for starting location or direction. Hence, Ant(Grid* g) is the only constructor I need, and I will use the setter functions to set ant's x-coordinate, y-coordinate and direction according to user preference.

I also added a time delay of 0.5s and cleared the screen after each step to animate the movement of the ant.

Test Plans and Results

Test Case	Input Values	Driver functions	Expected Outcomes	Test Results
1	Row = 0 Col = 0	intValidate(input)	Invalid input. Prompt user to re-enter number of rows and columns	Invalid input. Number must be greater than 0. Enter number of rows and columns.
2	Row = 0 Col = 10	intValidate(input)	Invalid row input. Prompt user to re-enter number of rows and columns	Invalid input. Number must be greater than 0. Enter number of rows and columns.
3	Row = 10 Col = 0	intValidate(input)	Invalid column input. Prompt user to re-enter number of rows and columns	Invalid input. Number must be greater than 0. Enter number of rows and columns.
4	Row = 10 Col = 10 User's choice for starting location of ant = 9	intValidate(input)	Invalid choice input. Prompt user to re-enter choices. 1 for specifying starting location 2 for randomly generate a starting location	Invalid choice. Enter 1 or 2 only. Successfully display menu again and ask user for choices.
5	Row = 10 Col = 10	intValidate(input) updateLocation(x,y)	Screen will display current location of ant at position (x,y)	Screen successfully display ant's

	User's choice = 1 User's x and y coordinates			location according to user's input.
6	User's choice = 0 (randomly generate a starting a position)	randStart()	Screen will display location of ant	Screen successfully display ant's location.
7	User's choice for direction of ant facing (1, 2, 3, or 4)	intValidate(input) setCurFace()	If user entered a valid choices, then ant should be facing the selected direction	Successfully print out ant's current direction and output matched user's input
8	Number of steps entered by user	intValidate(input) moveAnt() whiteMove() blackMove()	If user entered a valid number, the ant should move the set amount times according to the rules, and the movement will be display on screen.	Screen displayed movement of ant successfully.
9	Current cell color is white	whiteMove()	The ant should turn right 90 degree and the cell should be now black.	Ant successfully move to the right cell.
10	Current cell color is black	blackMove()	The ant should turn left 90 degree and the cell should be now white	Ant successfully move to the right cell
11	Row = 10 Column = 10 Random start. Random direction. Steps = 50	randStart() randFace() antMove()	The screen should display ant's moving 50 steps	Screen successfully displayed ant moving 50 steps

12	Row = 10 Column = 10 x = 15	intValidate()	Invalid input because x-coordinate is out of bound. Prompt user to reenter x-coordinate.	X-coordinate is out of bound. Enter a value smaller than 10. Enter starting x-coordinate.
13	Row = 10 Column = 10 X = 5 Y = 20	intValidate()	Invalid input because y-coordinate is out of bound. Prompt user to re-enter y-coordinate	Y-coordinate is out of bound. Enter a value smaller than 10. Enter starting y-coordinate.
14	Row = 10 Column = 10 X = 10 Y = 10 Direction = Up Steps = 10	antMove()	Ant moves out of bound. Program should terminate.	Screen displayed ant's starting location. Screen printed out "Ant moved out of bound. Program terminates now." Program successfully terminated.
15	Row = 10 Column = 10 X = 8 Y = 10 Direction = up Steps = 10	antMove()	Ant moves one step and program terminates	Screen's displayed ant first step and "Ant moved out of bound. Program terminates now.". Program terminated.