Christian Roccanova

CS 162 – Project1

4/16/2017

The purpose of this program is to implement a simulation of Langton's Ant. It will contain the classes Board and Ant. The purpose of the Board function will be to create and manage the board. It will create a 2D character array to represent the board itself with each coordinate in the array representing a single space. The black spaces will be represented by "#" and the white spaces by blank spaces " ".

The Ant class will contain the functions position, move, holdSpace and direction. The position function will hold the ant's current position on the board. It will be initialized with the starting point values given by the user. The holdSpace function will retain the position and color of the board space that the ant currently occupies by placing the x and y coordinates, as well as the color of the space into private the variables x, y and color. The direction function will contain a private variable "heading" which will represent the direction the ant is facing (up, down, left, right). The move function will turn the ant and move it to the next space on the board.

The main function, will contain an ant object, int variables for rows, columns, steps, and starting coordinates for x and y. It will first call the Menu function which will ask the user to select from the options Play, Play with random start and Quit. If Quit is selected, the function will cause the program to exit. If the user selects one of the play options, the user will be prompted for integers for the size of the array and the number of steps the ant will take as well as a starting position if random start was not selected. If random start was selected, the program will generate a random seed using the time function and will use that seed to create a random starting position. The Board function will then be called to build the initial board based on the user's input. The functions of the Ant class, as well as a loop to print the board will be contained within a for loop that repeats for the number of steps input by the user. The main function will then loop back to call the Menu function again until the user chooses to exit.

Testing

Input	**Expected Result**
Input a non-integer value for each user input	The program will loop until all user inputs are integers.
Request a board size that is non-existent (0 or negative rows/columns)	The program will loop until an appropriate size is entered.
Input a starting point that is outside the board.	The program will loop until an appropriate starting point is entered.
Place the ant in the center of the board and run for more than 4 steps.	The first four steps should create a square after which the ant will begin to turn away from the center following the right on white, left on black rule. It should also be changing the color of the

	spaces as it goes.
Place and run the ant such that it will move off of the board edge.	The ant should reverse its heading and move back into the board, continuing with the standard rules.
Run the ant a number of steps such that it will step onto spaces it had previously stepped on.	The ant should replace the space as normal (black to white or vice versa)
Select option 1 – play and repeat several times	The program should run as normal, including asking for starting coordinates. It should bring up the menu again after the last step. It should repeat this until the user selects option 3 - Quit
Select option 3 - Quit	The program should immediately exit without asking for any further input.
Select option 2 – Random start and play several times.	The program should run as normal, but without asking the user to input starting coordinates. The starting point should be different for each run.
Select option 1, then option 2 when the menu comes back up.	The program should run normally, asking the player for starting coordinates the first time and then running without starting coordinates the second time.
Select option 2, then option 1 when the menu comes back up.	The program should run normally, without asking the player for starting coordinates the first time and then running again, but this time asking for starting coordinates.

Reflections

Several changes were made to the initial design as issues arose within the program. First white spaces were changed to now be represented by a "." as it made reading the board significantly easier during testing. Several function names were changed; move became two functions, step and turn. Step performed the actual motion of the and had a loop added which tests if the ant will move off the board and if so, it reverses the ant's facing so that it moves back in. This as well as the need to determine the color of the next space necessitated the addition of two private variables, xfacing and yfacing, which held the value corresponding to the direction the ant would move (i.e. yfacing = 1 moves up). These facing variables were also incorporated into the direction function to give a numerical value to the string variable antDirection. The turn function was added to separate changing the ant's direction and its actual movement. holdSpace became saveSpace, but this change was merely cosmetic, though it does now make use of the facing variables to look back at the previous color. A reset function was also added to put all the relevant values in the class back to their original values in the case that the user chose to play again after the current game ended. Finally, the input validation class was added to take all user inputs and test if they were integers using the sscanf() function before returning the values to the main function. If this were not included, the program would go into an infinite loop if an invalid input was entered.