Justin Hammel
CS 162
Spring 2017
Project 1 Design Plan and Reflection

1. Define what the program should do.
    a. Purpose: Implement a simulation of Langton's ant. "Ant" moves for a set number of rounds within a 2d array with the following rules:
        i. In a white square, turn right 90^{o} and change the square to black.
        ii. In a black square, turn left 90^{o} and change the square to white.
    b. Input:
        i. size of 2d array: row – int, col - int
        ii. number of rounds: rounds – int
        iii. menu options using Menu class: run simulation with user selected start, run simulation with random start, possible menu options to set parameters from main menu (grid size, starting point, rounds), exit
    c. Processing:
        i. Loops for controlling ant movement within ant class
        ii. Loops to generate and fill board within a board class
        iii. Ant object orientation is determined by creating enumerated type, Direction { UP, DOWN, RIGHT, LEFT }
    d. Output:
        i. Output menu on start of program
        ii. Output prompts following menu selections (asking for board size and round input)
        iii. Print board using method within board class
        iv. Use clear screen method after each board print?
2. Program design
    - Output menu from Menu class: input array size/generate board, select number of rounds, run simulation with user chosen starting point, run simulation with random starting point, or exit program.
        - Use input validation to prevent simulation from running if the parameters have not been generated first (array size and rounds)
            - While no Board object made and rounds == 0 return to main menu with message indicating board not created or rounds have not been entered
        - After menu options selected use clearScreen method prior to start of simulation or prior to returning to main menu
    - Selecting array size creates Board object
        - Fill board initially with white spaces (' ').
        - If Ant position known place that using placeAnt(@)
        - Ant object, ant1, is created when user selects pick starting location, or selects random starting location
            - If user selected pass starting ant row (x) and col (y) to ant object.
    - Selecting option to input number of rounds allows user input for that parameter and stores it with set method.

- Run simulation with user selected start point chosen
    - Prompt user for row and column for ant starting position
        - Remind user about the board size constraints
    - Create Ant object to represent board starting location
    - Use Board print method for each round to reprint board
    - After first board print use Board clearScreen method to prepare screen for new board print
    -
- Require user input to exit simulation prior to returning to menu (e.g. press any key following final round)

3. Class Diagram

Board
int row int col char** boardLayout
Board(int, int) fill() printBoard() clearScreen() setRow(int) setCol(int) getRow() getCol() placeAnt (Ant) runSimulation ()

Ant
int antX int antY Direction antOrientation
ant (int x, int y, Direction) setAntX(int) setAntY(int) setOrientation (Direction) getAntX() getAntY() getOrientation() randomStart() whiteMove() blackMove()

Justin Hammel
CS 162
Spring 2017
Project 1 Design Plan and Reflection

Menu
vector <string> options int selections
addOption (string) printMenu()

4. Testing Plan

TEST	INPUT VALUES	DRIVER FUNCTIONS	EXPECTED RESULTS	OBSERVED OUTCOMES
No input1	None	main() while loop for sim	Testing input validation to prevent simulation from running with no data ([0][0] array, or 0 rounds)	
Integer input validation	spaces "asdf" 1234asdf after array creation test ant placement for values out of bounds	main() InputValidation function	Should report error message that only integer input is valid for all options in the simulation. Also test that bounds will not be exceeded by the program.	
Run with hard coded values	10x10 array 10 rounds ant placed at 5x5	main() simulationLoop() antMovment()	With a smaller dataset I expect it to be easier to track that the ant is moving as expected.	
Max/High Values	80x80 array increasing number of rounds starting with 100 Ant placed in various configurations	testing dynamic memory allocation, ensure that large numbers do not cause out of bounds issues	I expect the simulation to reach a point where running might prove difficult based on my internet connection to the flip server.	

5. Reflection

This assignment has helped me to realize that my design and implementation process needs some work. This project ended up taking much longer than I thought it would due to not having a clear picture in my head of how to properly implement the program.

There are two issues that I kept seeing pop up throughout this process. My biggest problem seems to come from not having a good enough strategy for developing my pseudocode, or being able to properly implement a flowchart. I find myself constantly overthinking things when it comes time to try and create a basic starting point to work from.

Finally, I have noticed that my level of knowledge when it comes to working with pointers is lacking. This is especially the case when working with arrays and dynamic memory allocation. I should have spent a little more time in the beginning trying to understand some of the concepts from the text prior to just diving in.