Brendan Jang
CS 16 2Project 1
1/19/2019

1. Program design
    a. The problem to be solved
        i. We put an ant on board, the ant will turn left and move forward if it is on a white cell, it will turn right and move forward if it is on a black cell. The cell lips its color after the ant leaves it
        ii. The board is a 2D array of characters
        iii. The ant should be an object of a class, attributes of the class cannot be accessed directly from the outside of the class
    b. Structure of the program
        i. There are 2 classes, Ant and Menu
        ii. Ant represents an ant, its position on the board and its orientation
        iii. Menu stores list of options and lets the user to enter the legal selection
    c. Test table
        i. Test int Menu::getSelection()

Test case	Input value	Driver function	Expected outcomes	Observed Outcome
Invalid input	e	int Menu::getSelection() do … while selection <= 0 || selection > size	Loop back to the question prompting the user for input	Loop back to the question prompting the user for input
Input too low	-1	int Menu::getSelection() do … while selection <= 0 || selection > size	Loop back to the question prompting user for input	Loop back to the question prompting user for input
Input too high	size + 1	int Menu::getSelection() do … while selection <= 0 || selection > size	Loop back to the question prompting user for input	Loop back to the question prompting user for input

        ii. Test int getValue(string message)

Test case	Input value	Driver function	Expected outcomes	Observed Outcome
Invalid input	e	int getValue(string message) while (val <= 0)	Loop back to the question prompting the user for input	Loop back to the question prompting the user for input
Input too low	-1	int getValue(string message) while (val <= 0)	Loop back to the question prompting the user for input	Loop back to the question prompting the user for input

iii. Test int getValueBelow(string message, int above)

Test case	Input value	Driver function	Expected outcomes	Observed Outcome
Invalid input	e	int getValueBelow(string message, int above) while (val <= 0 || val > above)	Loop back to the question prompting the user for input	Loop back to the question prompting the user for input
Input too low	-1	int getValueBelow(string message, int above) while (val <= 0 || val > above)	Loop back to the question prompting the user for input	Loop back to the question prompting the user for input
Input too high	above + 1	int getValueBelow(string message, int above) while (val <= 0 || val > above)	Loop back to the question prompting the user for input	Loop back to the question prompting the user for input

iv. Test int main(int argc, char** argv)

Test case	Input value	Driver function	Expected outcomes	Observed Outcome
Quit game	2	int main(int argc, char** argv) if (m1.getSelection() == 1)	The game ends	The game ends
Start game	1	int main(int argc, char** argv) if (m1.getSelection() == 1)	The game starts	The game starts
Continue the simulation	1	int main(int argc, char** argv) do while (m2.getSelection() != 2);	Ask for new inputs	Ask for new inputs
Stop the simulation	2	int main(int argc, char** argv) do while (m2.getSelection() != 2);	Stop the simulation	Stop the simulation
Ask for Random starting location	1	int main(int argc, char** argv) if (m3.getSelection() == 1)	No new input required	No new input required
Ask for Random starting location	2	int main(int argc, char** argv) if (m3.getSelection() == 1)	Ask for ant starting position	Ask for ant starting position

2. Reflection
    a. Ant and the board
        i. I don’t really put the ant on board,
        ii. I store the position of the ant in Ant object
        iii. I put it on the board when I print the board on screen then take it out
    b. I use constant value to note the orientation of the ant so that we could calculate its orientation swiftly without using if else structures
    c. I use try catch to get legal input from the user
    d. I gained a lot of things while trying to get legal input from the user and be sure that there isn't any memory leaking after the program ends