Stephen Townsend

Langton's Ant Design

The design of my Langton's Ant will have a menu that will prompt the user for a choice 1 or 2.

The first choice will be to select a location. The user will then be prompted to choose how many rows are in their grid. The input will go to an input validation function that will determine if the input is between 20 and 60. 60 and 20 have been chosen as a max and min because I want to try to keep the program on the screen output. If the grid is too large then there is risk of not being able to see the ant depending on the location. If the input for both row and col is acceptable, then the user will be prompted for the starting X and Y values. The X and Y values will also go through a validation check.

The validation will make sure that the location is at least 20% less than the parameter on each coordinate. This will make it so that the user has less of a risk of running the ant out of bounds. I chose 20% because I did not want to limit the users option to much. They have a plethora of options to choose from still. After the Row, Colum, X, and Y values are validated then the user can put in the amount of steps they would like the ant to go.

The second option will be to run with a random location. This function does not use an input validation because the user is limited. I needed to find a way to create a random starting point that would also be within bounds. To do this, I have decided to set a default grid of 40x40. With this grid, I made a lowRange and an upRange. These are integers that will make it so the random algorithm is within bounds of my grid.

> I started using rand(), but this is not random enough. Rand() has a very predictable pattern. As a result, I decided to use srand(time(0)). The srand() function is used to make a seed for rand(). To add the randomness I put time(0) in srand. Time(0) uses the system clock to generate a number. With this I am able to have something that is more random then the rand() function.

Once the random number is generated then the user will be prompted for the amount of steps they would like to run.

After the menu, the program will send the inputs into my AntGrind cpp through an AntGrid object. The object will first initialize a dynamic 2D array with row and col to make the grid. The program will then send this to the paraX and paraY variables to hold the grid size. Then the ants location is next followed by the steps. Once these are finished the program will now really start and go into the AntGrid move function.

The move function is where the ants movement locations are set. The program will go through a while loop that will end when the steps have been meet. While in the loop, the Ant cpp will be called to get the ants icon and the ants starting direction. There are a series of if statements that will use this information to know where to start. The loop will first check the square color, then it will call for the ants icon. The program will then go-ahead and output the grid with the ant in its starting location. From here the looping will start. The if statements will first split based on the square color that was obtained previously. Then the grid location will be updated to the opposite character. If black, then it will lace a # sign. If the square was white, then the spot will be assigned with a blank. Now the if statement will compare the direction. The direction will determine how the ant moves and the direction will be updated to account for the movement.

Test Plan & Test Result				
Row	Col	Location	Steps	Output
50	50	(25, 25)	52	The screen will display a heart design
>60	>60	-	-	The user will be prompted to input a new value within the min and max
20<X>60	20<Y>60	> 20%	-	The User will be prompted that the location values are not valid and they need to enter a new value
20<X>60	20<Y>60	Within 20%	2000	The Users values are valid and the program will run the requested desired steps of (500)
20<X>60	20<Y>60	Within 20%	Large Step	The ant may run out of bounds and the program will crash. Unable to find a way to dynamically allocate the grid.

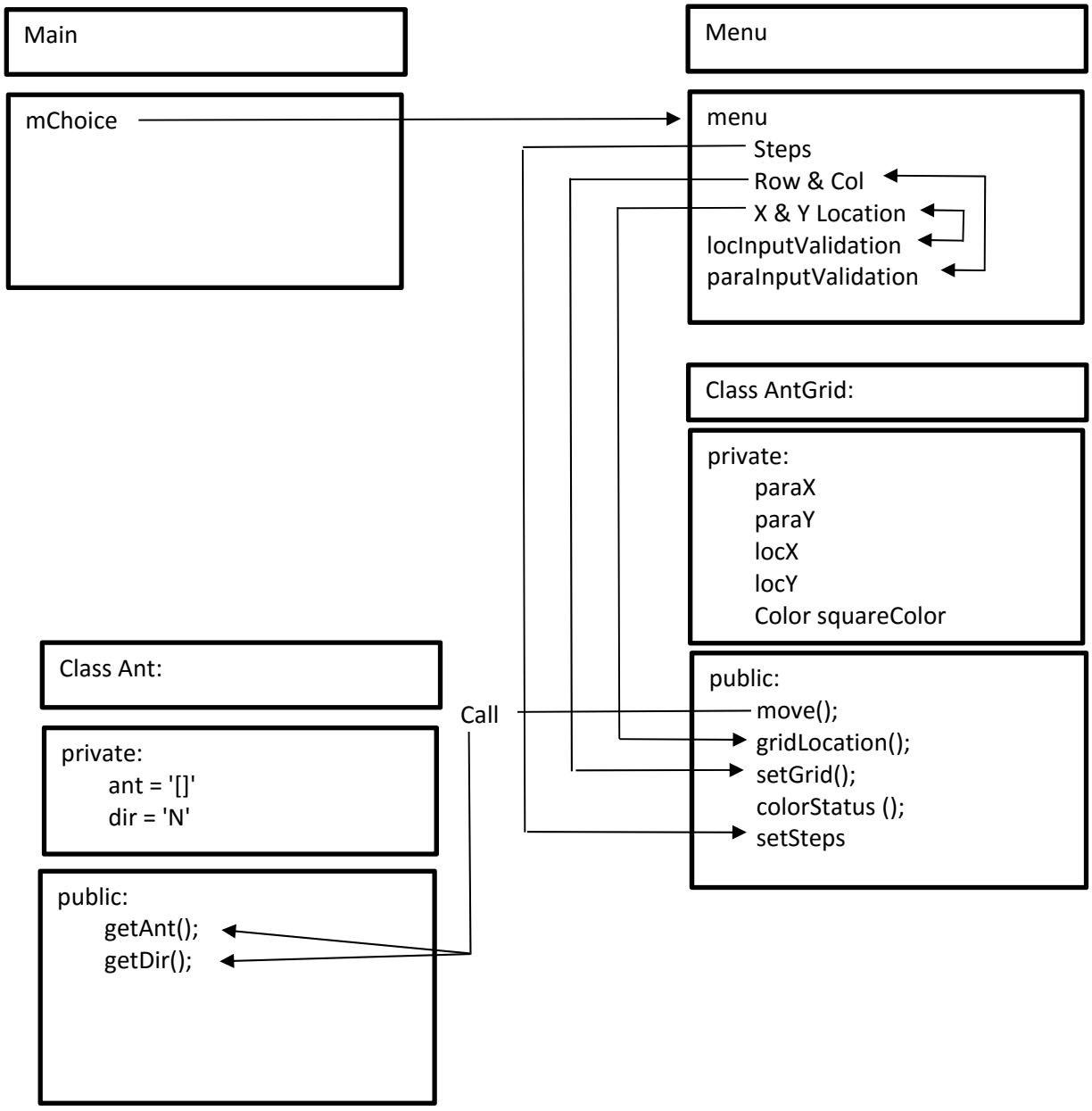
Stephen Townsend

I had issues trying to dynamically allocate the grid to make it so that the ant could run for extended periods of time. To combat this, I set up input validations that I knew would not allow for many out of bounds problems.

I had a hard time trying to figure out the movement for the ant. I was trying to map it out and determine a pattern through by how it would move. I realized that I was making it a lot harder than it was. The mapping took a lot of my time with the project. Once I looked at the smaller picture, I saw how I could make the ant move. As a result, I coded and mapped out the movement at work one day.

At the end of the assignment when I was cleaning up my code. I was getting error where the Ant object in my antGrid class was not recognizing and the computer would not compile. I still do not know what made that error. After a few hours of changing headers and constructors I got the program to accept what I originally had.

Design Logic



Langton's Ant BrainStorm

ANT.HPP & ANT.CPP

Need an Ant Class

What will the Ant Class have???

Need to keep track of ants current position (locX, locY)

Need to keep track of ants current direction (dir)

Ant will be detonated with '[]'

GRID.HPP & GRID.CPP

Need to have a Grid

Ant will walk along the grid

#' marks will be White

Foreground will be considered Black

if Ant hits a black square the turns left and changes to '#'

If Ant hits white square it will turn right and change to ' '

THIS WILL CAUSE THE MOVEMENT TO BE REVERSED THEN IF STARTING WHITE

How will Ant Turn???

Use dir variable to set direction and then change corresponding x or y value

Grid Boundires???

MENU.CPP

```
int main()
{
```

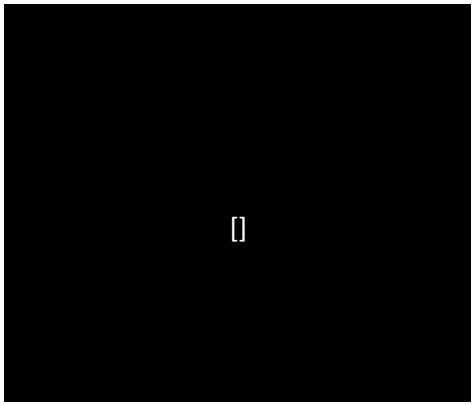
Pre-Menu

ask the user about the dimensions of the grid row & col

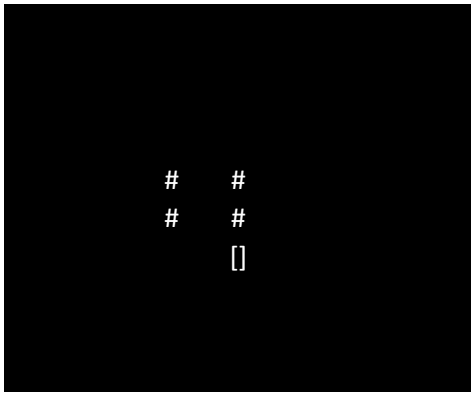
set the 2D array to the user dimensions

Or should I pass an object and have 2D in GRID???

Layout



5 Steps Layout



Need to have a User Menu.

Choice 1 : Where will Ant Start?

Choice 2: Random Starting Location?

if choice > 2 Error check for valid choice

Outside of menu : How many steps?

return 0;
};

Option 2. How do I make random and keep into boundires if user defines gride size.

Take away grid Sizer for option 2.

Random number genertaor:

Rand() is predicatble.

maybe use a seed with time(0) -- Needs to have include <ctime>

Allow user to define steps still.

MOVEMENT

	Row	Col	Move to		Dir	Color	Set Dir
start at	4	4	3	4	Dir == N	WHITE	Dir = E

Direction	Color					
North	W	East	if (Dir == N)	if(color == WHITE)	col +1	New Dir = E
	B	West	if (Dir == N)	if(color == BLACK)	col -1	New Dir = W
East	W	South	if (Dir == E)	if(color == WHITE)	row +1	New Dir = S
	B	Noth	if (Dir == E)	if(color == BLACK)	row -1	New Dir = N
South	W	West	if (Dir == S)	if(color == WHITE)	col -1	New Dir = W
	B	East	if (Dir == S)	if(color == BLACK)	col +1	New Dir = E
West	W	Noth	if (Dir == S)	if(color == WHITE)	row -1	New Dir = N
	B	South	if (Dir == S)	if(color == BLACK)	row +1	New Dir = S