Thanhbinh Truong
CS162

Reflection

As I read the instructions on what the program would pertain and what would need to be done, I began writing down what type of files I would need and such. The first that came to mind was an ant class which means I would need two files, a .cpp and .hpp. The private and public variables and functions were the simpler parts to get together, such as the set and get functions. Furthermore, creating the definition for the functions was also a task that I found easy. What became difficult during the process of creating the program was where to implement the board for the ant to be moving across. I first tried creating a board class itself, with its own hpp and cpp files, but creating this mess just made things longer and harder. Having a board class (which would hold an array) made it difficult to access those members. After deciding to drop the board class idea I began to read the cs162 piazza to see if others were able to solve my problem. After reading, I decided to just implement the board array itself into the main function where I figured that I was thinking too hard towards creating this program that I did not think of the simple solutions.

Another road block that I ran into that ended up being a simple solution was implementing the menu function. After creating most of the program I felt myself fatiguing in the mind and felt that I was working too hard in implementing a menu function. At first I tried to have the function return a int, but forgot about it being out of scope and being able to be read by something outside the function. After contemplating how to set up the menu, I decided to just have a void function for the menu to only display it, and create the cin >> after the function to get the answer from the user.

What I've learned about this program in a broader aspect is to not implement complicated things when it is not necessary. Also, I learned to implement more design before writing the code to be more organized and be able to break down steps of the program into simpler steps.

(Langston's Ant)

Design

Program purpose: to mimmick langston's Ant design

needed: - Ant class (.cpp/.hpp)

- Functions
  - location (get/set)
    - * 2 seperate, 1 for current location
      1 for new location
  - direction (n, s, w, e)
- Board array to display game
- Main
  includes: Menu function to
  1) start game
  2) exit game
  - Prompt user for board size: (row) (column)
  - Prompt user for # of steps
  - Prompt user for start location

  (run program)
  - Deallocate

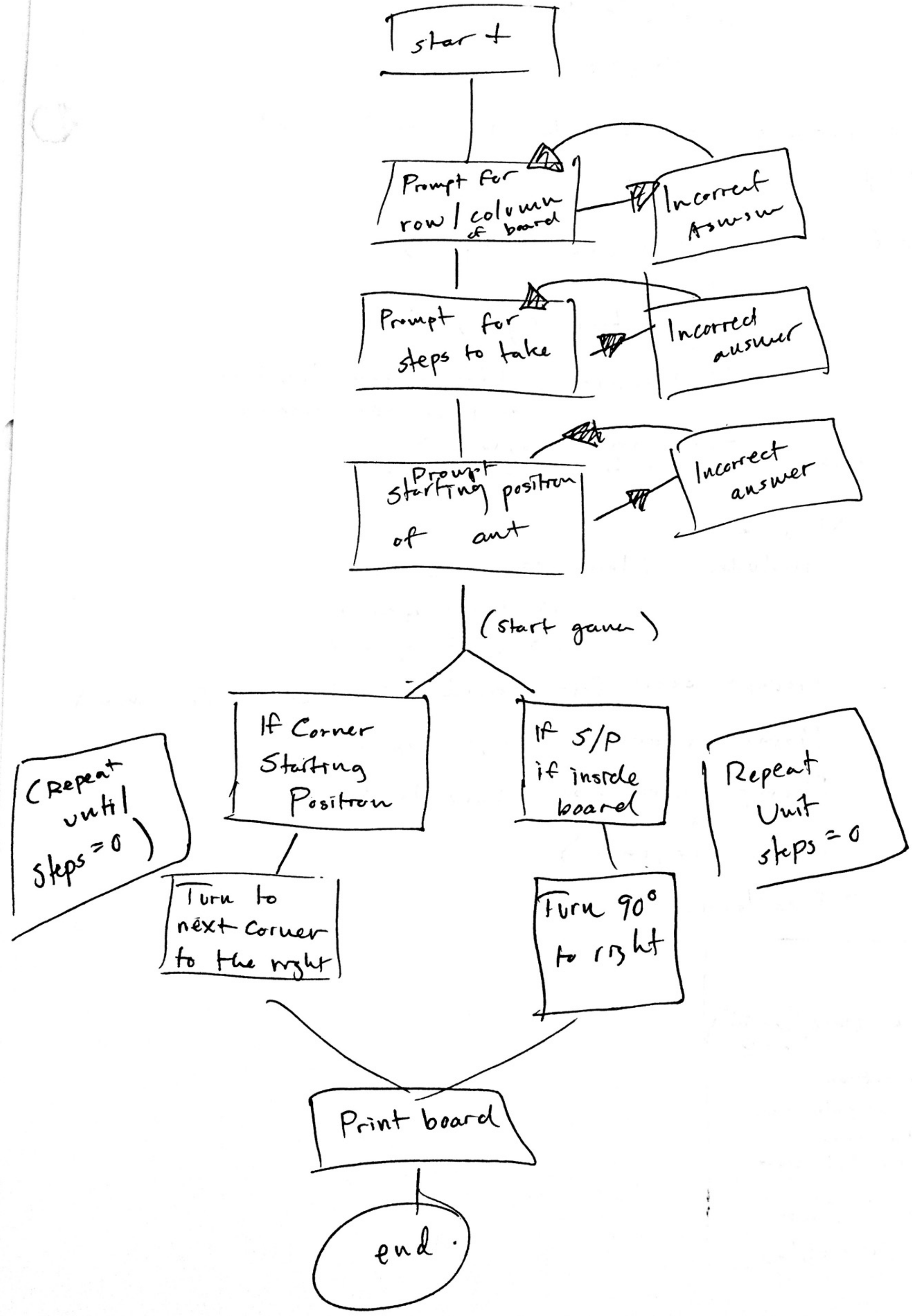
Ant Class:

private: char space (for # or @)

Currentrow
Currentcolumn
New row
New Column

Public: Default Const

get/set for variables


start
Prompt for row / column of board
Incorrect Answer
Prompt for steps to take
Incorrect answer
Prompt starting position of ant
Incorrect answer
(start game)
If Corner Starting Position
If S/P if inside board
(Repeat until steps = 0)
Turn to next corner to the right
Turn 90° to right
Repeat Unit steps = 0
Print board
end.