RobotState Class

Methods

* Method that updates stack with input string
* Method that pops off the top of the stack
* ? Method that reads from the stack ?

Main code functionality:

Initialize variables/vectors to be used

Inputs setup using Maze class methods

Main BFS loop section:

While(cPos != gPos)

movedPosition = false;

// example for ‘north’

if (currentPosition.at(0) > 0 && movedPosition != true) {

newPosition = [wheeled/tracked]Robot::Up(currentPosition)

bool a = Maze:: isObstacle(newPosition)

bool b = (**function that checks if past position. Must decide if vector in main.cpp or attribute within Maze class)**

if (newPosition != previousPosition && a == false && b == false) {

-add to pastPositions array and update size

Maze:changeSpace(newposition, unique robot char)

RobotState Up pushed to top of stack

previousPosition = currentPosition

currentPosition = newPosition

movedPosition = true;

}

}

// Backtracking

if (movedPosition != true) {

vector<int> temp = currentPosition

add currentPosition to wrongPath vector (**decide if vector in main.cpp or attribute within Maze class)**

currentPosition = [last row from PastPositions]

erase last row from PastPositions

previousPosition = temp

pop out top of the stack

}

Considerations:

How are we running both robots within the code?

Do we set pastPositions & wrongPath 2D vectors as Maze attributes or leave them in main.cpp ?

If so, do we need to account for both robots?

How do we check pastpositions? Do we include “|” and “-” inside Maze::isObstacle ?