Phuong, Qudsia, Monika

**Final Lab - Design Document**

**Section 1: Project Overview**

Our robot is a hybrid agent in terms of planning. It is deliberative because it has a map of the world which is a grid in its memory. It is reactive because it changes its path when it hits an obstacle.

The robot's aim is to find a goal within a maze grid. The robot knows only the gird layout and the starting cell but has no idea where the obstacle and goal cells are. The robot has sensors that can detect obstacles and the goal once it reaches them.

For our robot, we use the bump sensors to detect obstacles. Every time the robot bumps in to an obstacle it will update its knowledge base, turn to its left, and try the next cell in its path.

We use the light sensor to detect the goal, which in this case is a white colored cell. Once the robot detects the goal cell it plays some tones and makes a 180 degree turn. It then walks back to the starting cell using a shorter path.

**Section 2: The algorithm**

Pseudo code:

If goal

Go back to start cell

If current cell is not a dead end

Add previous cell to stack

If current cell is an obstacle or border

remove from path

remove from stack

make previous cell the current cell

update the current cell value

For all adjacent cells

if not an obstacle or dead end and has not been visited

add to stack

If no adjacent cell has been added it is a dead end

set the current cell value to one

If stack is empty and current cell value is one

stop, it is a dead end

While the value of the cell on top of stack is one

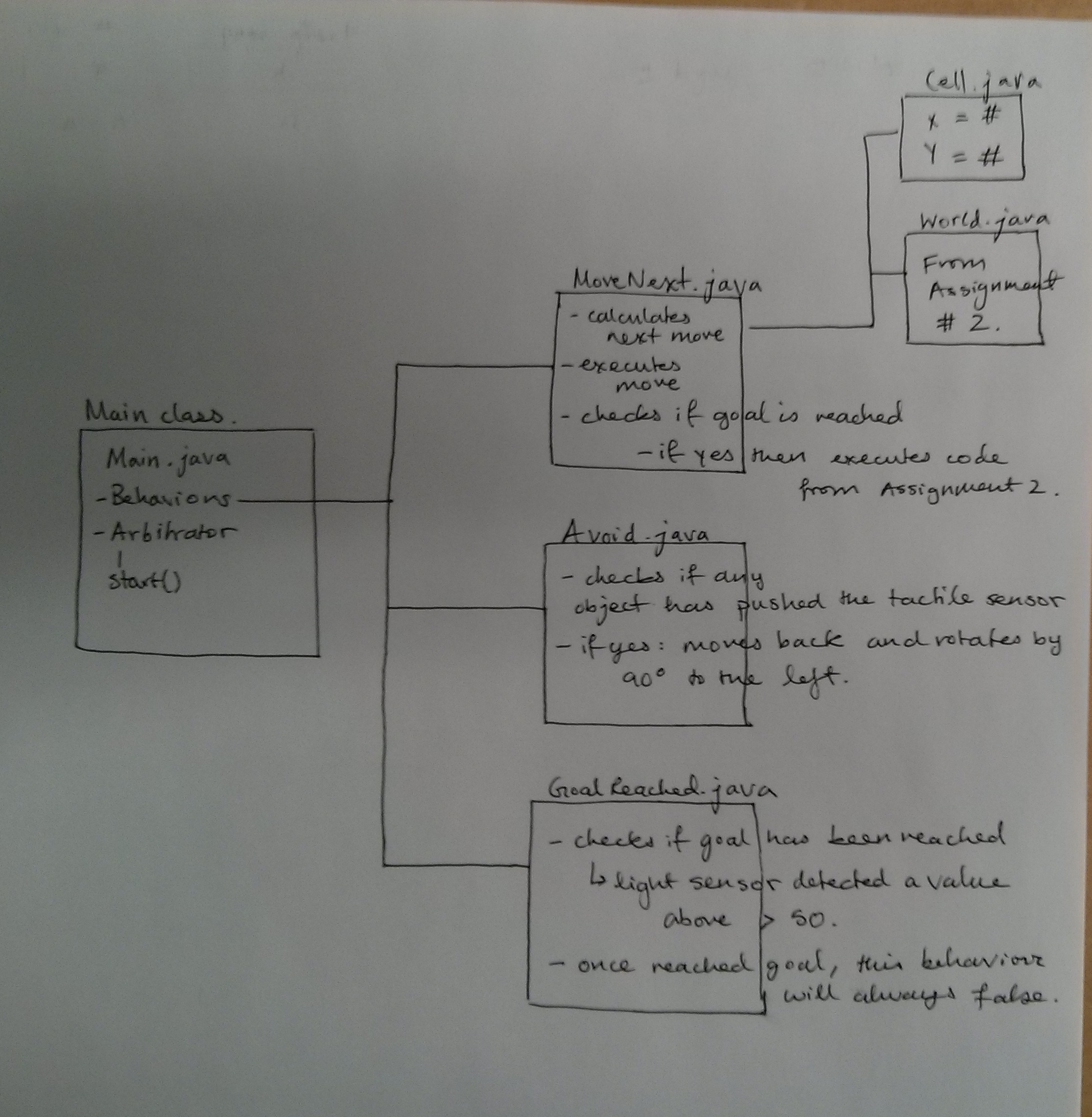
Pop it

Pop the next cell

Move to popped cell

We have used DFS with backtracking. The two main advantages of our traversal algorithm are: First, every single cell will be visited if needed until the goal is found. Second, If the robot reaches a dead end it backtracks. One disadvantage of the algorithm is that goal might be at a shallow level and the robot will take a longer path to reach the goal.

**Section 3: High level design**



**Section 4: Low level design**

Main : Contains the code for the behaviors and the arbitrator. It also passes in the required variables into the various classes as needed, such as goalReached and currentCell. Execution is done in this class.

MoveNext : Is the default behavior for this robot. Contains the code for:

* Travelling to the next cell
* Checking the adjacent cells and pushing them into the stack for the DFS search algorithm
* Avoiding behaviors that have already been explored
* Backtracking if needed
* Executing the code from Assignment 2 when the goal is reached
* Once the goal is reached, a closed world assumption is made where any cells not explored would be considered to be an obstacle and the shortest path would be charted.

Avoid : Is the next behavior in the arbitrator, which avoids the obstacles that push the tactile sensor’s button. This behavior simply turns the robot to the left/right (subjective).

Goal : This high priority behavior simply lets the MoveNext class if the goal has been reached, meaning if a while surface has been observed. Once this goal has been reached, the takeAction() method will always return false.

Cell : This is an object to keep track of each individual square from the grid.

World : This is the class from Assignment #2 to execute the path from the goal to the start position once the goal has been reached.

**Section 5: Benefits / Risks**

Benefits:

1- Our robot will explore each and every cell until it finds the goal. In other words if there is a goal and there is a path to the goal from the start position the robot will eventually find the goal.

2- After it has found the goal it take a shorter path to the start cell based what it knows about the grid. It ignore the cells that have not been visited.

3- The goal can be any where on the grid. The robot does not look for the goal in the other of end of the grid.

4- Because it is a hybrid agent the robot is flexible and can adopt to a changing environment by updating its map every time it visits a cell. So it will have a memory of what it has explored.

5- Once the robot reaches a dead end it can backtrack.

Risks:

1 – If the robot has visited a cell and has updated its map it assumes the world will not change again so it might behave unexpectedly if the world does change.

2- The robot has not been tested for world that doesn't have a goal or a world that doesn't have a path to goal.

Assumptions:

1- The world will not change after it has been explored.

2- There is a goal somewhere in the world.

3- There is a path to the goal.

**Section 6: Future Work**

One future improvement would be to make the robot a more robust hybrid agent so that it is capable of dealing with an unpredictable world. For example if an obstacle is placed on cells that had been visited and didn't have obstacles before the robot must be able to update its map of the world again.

Another improvement would be to test the robot for a world that doesn't have a goal or doesn't have a path to the goal and make adjustments to the design if needed.