**AI: Final Project Robot Maze Navigation**

**Wallie 2.0 Design Document**

**Fall 2013**

**Tai-Lan Hirabayashi & Jacqueline Oberlander**

**Project Overview:**

1. Is a particular search algorithm employed as the foundation for your solution:
   1. In designing a maze navigation robot we used Best First Search (BFS) as the foundation

**Section 2:**

1. How is the state in the search space defined
2. How is the next state chosen
   1. The next state is selected using BFS. Whenever the robot finds an obstacle it is added to the known map. Whenever an obstacle is added all cells neighboring it lose 1 move option thus the cell with the most move options is considered to be the best cell to enter. Cell move selection is preffered in the following order: forwards, right, back, left.
3. What happens when a dead-end is reached
   1. When a dead end is reached in the robot backtracks. The robot tracks the number of possible paths left to explore for each cell, thus whenever it reaches a place where there is only 1 option, to return to the previous cell the robot backtracks until there are other viable paths that it can select from that have not been selected before. If the entire puzzle dead ends without a possible move the robot stops and the search is ended.
4. Is a particular algorithm employed as the foundation for your solution (?repeated)
5. What are the advantages of your approach
6. What are the disadvantages

**Section 3:**

High level entities: groups of objects that constitute major constructs of design. Ex controller object, set of behavior objects. Makes sense to include block diagram to illustrate relationships between objects. (ex uml) explain in a few sentences what each entity does. Describe reasoning for defining entities in your diagram and what their roles are.

**Section 4:**

Describe low level design for each entity: where individual object and relationships are defined.

Define:

Describe in a paragraph how the object is used, and what function it serves. Ex show the interface, describe thought process for defining the object as you did, list benefits and risk if an object provides an encapsulation describe in a sentence why the encapsulation adds value. Give diagrams meaning with description.

Configuration if the object needs special configuration or initialization.

**Section 5:**

Benefits, assumptions, risks/issues. List 5-6 top benefits of the design, all known risks/issues and all assumptions. May rehash, important so the reader doesn’t need to read the entire doc to understand benefits, assumptions, risks/issues.

Benefits include:

Risks/Issues:

The robot must be told where it is located as it’s design is based on a grid approach. We used a grid approach because the robot must accurately find it’s way back to the original cell and in order to improve the path that it took to the goal cell a definite path with obstacles must be held in some form or another. Thus the robot if misplaced can easily wander off of it’s search space.

There are technical difficulties in using Lejos and any other type of sensor as environmental changes can cause false sensor readings. Examples of this might be accidental sound causing the robot to believe that it is going in the wrong direction.

In addition another risk is that the path that the robot finds may be incredibly inefficient as it is **BFS**???.

Assumed:

When developing the project we took the liberty to assume a few things. Firstly we assumed the following: that the goal is unknown, that there is no expectation for the robot to “find” itself within the board rather it will know it’s original location and it’s boundaries, that additional cost(s) of a map and monitored path back were negligible considering that the robot’s movement takes most of it’s time, we also assumed that sound corrections would indicate to the robot that it was going in the wrong direction entirely and that it was thus appropriate to section off the board. We also assumed that the world was solvable in cases where claps, or accidentally setting off the sensors would lead the robot to find no solution we simply shut down.

When designing the algorithm we assumed that it was just as efficient to use an **? BFS??**. We assumed that **...** We decided against using a greedy search based the fact that the goal state is between (3,0) and (4,7) and that we start between (0,0) and (1,3) as we felt that designing a search that was more general would be more customizable to different situations and trials.

**Section 6:**

Future: what changes would you make given time to work on this project (bug fixes, optimizations, changes in design, etc. Explain why they should be made and how the project would benefit.

1. Notes:
   1. Robot navigates towards unknown marked goal.
   2. Learns max to find optimal way back to the starting point.
   3. Robot given a starting point, and final destination point, no other knowledge other than information gathered during navigation.
   4. Ability to navigate & memory as a map
   5. Recognize intersection and mark branches already explored, when you reach a place you have already located take an unvisited branch.
   6. Robot starts somewhere between (0,0 and 1,3)
   7. Goal cell is white, within (3,0) and (4,7)
2. Goal: Develop a strategy and implement the corresponding algorithm so robot will make its way back to start without making a wrong turn.
   1. Robot begins at center of start cell.
   2. Must use atleast 2 different types of sensors.
   3. Play song @ goal.
3. Presentation describing how you solved the problem and lessons learned. Demo robot.
4. Extras:
   1. Sound corrections
      1. Ex clap hands to tell the robot it is headed in the wrong direction
   2. Remote commands using Bluetooth.