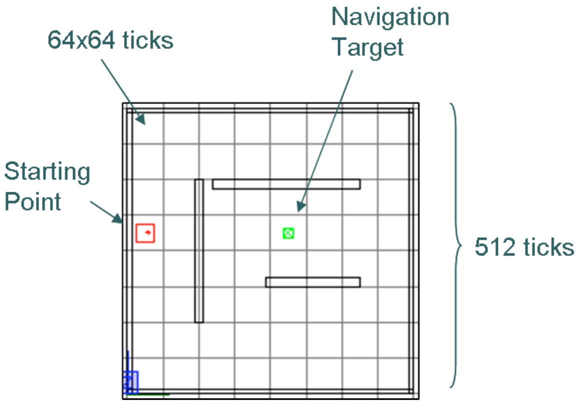
The Shortest Path to the Disk

Programming Assignment #5
CSC 481 – Spring '10
This is **NOT** a Team Assignment

Problem Statement

Your quagent will be placed in a room with obstacles and internal walls (the 'Obstacle Room'). In the room, at the global coordinates (288,288,128), you will find a rotating disk (data 288,288,128). If you are using the quagent.config file that came with the system you can simply uncomment the relevant lines in the files to see the disk (you should also delete the previously inserted tofu statements).

The goal is to develop a Java program based on the A* algorithm which searches for the shortest path between the quagent spawn point on the rotating disk and then plans the quagents commands necessary to move the quagent from the spawn point to the disk. Given that this is a searching and planning algorithm you can assume that your quagent brain has complete knowledge of its environment. The following map is a layout of the 'Obstacle Room' given in 64x64 tick tiles.



For your own program you should probably choose 32x32 or 16x16 ticks tiles to be able to compute smoother paths. **This assignment must be done individually.**

Deliverables

(1) a domain analysis document describing how you implemented the A* algorithm (major data structures, algorithm trade-offs, assumptions about the global knowledge etc)
(2) your Java source code

Submitting your Project

Submit your work by email to hamel@cs.uri.edu by **Tuesday March 30th 10pm.**

Grading

20% - domain analysis report & lists
20% - structure of the code/implementation of the A* algorithm
60% - correct execution of the program

Train Your Quagent

Programming Assignment #6
CSC 481 – Spring '10
This is **NOT** a Team Assignment

Problem Statement

Up to now we as developers were responsible for coding the AI for the quagents explicitly, either as scripted Java code, rule based Prolog code, or as an explicitly coded search procedure as in the A* algorithm. In this assignment we hand over control to machine learning. The goal is to have the quagent acquire the knowledge it needs to perform its task from a set of explicit examples using the ID3 tree builder.

The task at hand is to walk along the walls of the Empty Room in a counter clockwise fashion (it would be ok to cross the room the first time from the spawn point). There is no stopping criterion, the quagent simply keeps on walking along the walls until it dies of old age.

In order to accomplish this assignment you should download the learner.zip and the id3.zip files that contain the learner framework and the ID3 tree builder, respectively. **You are not allowed to change these files or the file that these programs generate** for this assignment, the only control you have over the quagent is via the training examples.

Note, please restore your quagent.config file to a state such that there are no more obstacles in the room. DO NOT use the soldier. **This assignment must be done individually.**

Deliverables

The table with the training instances that teach your quagent the desired behavior.

Submitting your Project

Submit your work by email to hamel@cs.uri.edu by **Tuesday March 30th 10pm**.