Quiz Review

Here are some topics you should review:

Overview

- Understand the main differences between different algorithms.
- Be able to identify which algorithms run faster / slower and how they use resources.
- Know which searches are complete, informed, and systematic.
- Understand which algorithms are optimal and in what ways.

Breadth-First Searches

- Understand the difference between different measurements of cost.
- Know in detail how each of the breadth-first family of algorithms operates.
- Be able to step through code for each of the breadth-first family algorithms.

Informed Searches

- Know how to estimate the distance to the goal.
- Be ready to examine the effect of weights (terrain.)
- Understand the difference in cost between different types of steps.
- Know how to calculate the actual cost from the source (given cost.)

Search Review

There are numerous planning algorithms and each one has its own advantages and disadvantages. The most important criteria for evaluating a planning algorithm are:

- ***CPU Time*** – How much processing time does the algorithm require?
- ***Memory Use*** – How much memory does the algorithm need?
- ***Optimality*** – How "good" is the solution the algorithm finds?
- ***Completeness*** – Does the algorithm guarantee a solution if one exists?

Planning Algorithms

Breadth-First Search (BFS)

Breadth first is one of the fundamental methods of traversing/searching a tree or a graph. It conducts a brute-force style search, checking nodes one ply (step) at a time.

Characteristics:

- Uninformed search
- CPU intensive
- Memory intensive
- Optimal in *steps*
- Complete
- Exhaustive

Greedy Best-First Search (GS)

Greedy Best-First Search is a derivative of BFS. Instead of ordering successors by steps, it considers them by their estimated distance from the goal.

Characteristics:

- Informed Search
- Not very CPU intensive
- Not very Memory intensive
- Not Optimal
- Complete* (with some heuristic estimates)
- Exhaustive

Uniform Cost Search (UCS, or Dijkstra)

Uniform Cost Search is a derivative of BFS. Instead of ordering successors by steps, it considers them by their cost from the start. Lowest cost nodes are considered first.

Characteristics:

- Uninformed search
- CPU intensive
- Memory intensive
- Optimal in *cost*
- Complete
- Exhaustive