Graph Theory Deep Dive Proposal: Minimax Algorithm With Alpha-Beta Pruning For Competitive Path Planning

Irene Hong, Ivy Mahncke, Vivian Mak

Project Overview

For this project, we plan to implement the minimax algorithm with alpha-beta pruning to optimize two competing path planners in a pursuer-evader game.

In the game, two robotic agents traverse an obstacle-dense graph world: the first seeks to minimize the distance between the two, while the second seeks to maximize that distance. Their actions (up, down, left, and right) will be chosen by the minimax algorithm, which alternately selects actions that minimize or maximize the distance between the agents. The minimax algorithm models the game state as an m-ary tree (m = 4), in which each node represents a game state and its children are the next possible game states. The alpha-beta pruning optimization method simplifies the tree by "looking ahead" to future game states and pre-emptively eliminating branches that guarantee a loss for the current agent.

The world itself is discretized as an occupancy grid, structured as a two-dimensional array. Every call is adjacent to at most four other cells (up, down, left, right), but can be adjacent to less if on the edge of the world. Cells containing obstacles contain a value of -1 and are impassable. Cells occupied by an agent contain a 1. All other cells contain a zero. We will explore how obstacle density in the world influences the results in the game.

To win the game, the pursuant must be in a node adjacent to the evader's current node. If the evader successfully avoids the pursuant for a certain number of turns, it wins instead.

Deliverables

- Simulation GIF
- Benchmarking visualizations (speed of win/loss vs obstacle density)
- Poster

Project Timeline

Day 1 (Monday): Algorithm Research & Project Proposal
Day 2 (Tuesday): Build Graph Environment + Agents + Game Logic
Day 3 (Wednesday): Implement Minimax w/ Alpha-Beta Pruning
Day 4 - 5 (Thurs-Fri): Asynchronous Unit Testing + Debugging
Day 6 (Sat): Synchronous Testing + Debugging Minimax Algorithm
Day 7 (Sun): Create Visualization Code
Day 8 (Mon): Code Lock & Begin Poster
Day 9 - 10 (Tues-Wed): Develop Poster + Rehearse
Day 11: Presentation Day!

Date	**Pre Meeting**	**Meeting**	**Goal**
Mon 10/20	Ideate	After class	Finish project proposal
Tues 10/21	Project proposal finished	Divide up tasks for world/agent model	Model world, game logic, start agents
Weds 10/22	Understand minimax and alpha beta pruning World model unit tested	Review + discuss algorithm pseudocode	Implement Minimax w/ Alpha-Beta Pruning
Thurs 10/23	Draft of algorithm with some unit tests complete (can be broken)	In class (V) - Check in with sarah SWE (I)	Test minimax algorithm
Fri 10/24		SWE (I)	Test minimax algorithm
Sat 10/25	Async work done	SWE (I)	Codebase working
Sun 10/26			Create Visualizations
Mon 10/27		In Class - Check in with sarah	Code Lock + Begin Poster

Tues 10/28			
Wed 10/29			
Thurs 10/30	- Poster printed - Rehearse	PRESENTATION DAY	

Resources / Notes

Min Max

- Minimax w/ Alpha-Beta Pruning primer and pseudocode

Alpha Beta Pruning

-

World Simulation

- https://stackoverflow.com/questions/56614725/generate-grid-cells-occupancy-grid-color-cells-and-remove-xlabels
- Algorithms Explained – minimax and alpha-beta pruning

Graph Project Resources