Project Proposal

Matt Caruano
2/24/2014

For the final project I plan on utilizing Monte Carlo simulations heavily. With a simple 11x11 game board, there are a large number of simulations to run. As such, I want to utilize threading to handle as many simulations in parallel as possible. Parallel processing should make a substantial difference in run time.

Multi threading has become very popular in recent years, so it will be to my benefit to learn how to use multi threading in my code. I plan on using the C++11 <thread> library for my threading purposes. Given a certain game board configuration, my program will pick a legal move to make at random, and then randomly simulate the rest of the moves to determine who would win the game. Many games would be simulated for each individual move, and a probability would then be assigned to that specific move in the form of (number of wins) / (number of simulations). When all legal moves have been simulated and assigned a probability, the move with the highest probability of winning will be executed. With the <thread> library, I should be able to split the simulation workload for each move into smaller chunks, and then sum the number of wins and number of simulations after all threads have run to determine the same probabilities in a fraction of the time.

Lastly to verify the effectiveness of multi threading I plan on doing a time analysis comparing a single AI move determination both with and without threading.