Artificial Intelligence for Traditional Games
Winter, 2013
Course Assignments

For this course you will be required to write a program capable of playing the game Breakthrough. This document outlines the milestones on the course project. For graduate students, all work on the project is to be your own. For undergraduates, you make work in teams of two; please confirm your groups before the first assignment is due. You are free to discuss the project and enhancements to your program with other students, but you may not share source code. List any students outside of your group that you discussed each milestone with on your project report.

Your reports should be written up in a word document and stored in your course SVN repository.

Server instructions will be posted soon on the course web page.

Part 1: Implement Breakthrough Due January 28, 2013 at 12 midnight

Implement the game Breakthrough so that you can get a list of legal moves, apply moves, undo moves, and test for the end of the game. Write a short report providing information about the game gathered from random games. What is the average length of a game? What is the average branching factor throughout the game? Does the game seem to be biased towards the first or second player?

Part 2: Implement minimax and a basic evaluation function Due February 4, 2013 at 12 midnight

Implement the minimax algorithm and a basic evaluation function. Your evaluation function can still be quite simple (eg the difference in branching factor for each player). Write a short report describing your evaluation function.

Part 3: Implement alpha-beta pruning Due February 12, 2013 at 12 midnight

Add alpha-beta pruning to your implementation of minimax. Look at how move ordering influences the number of nodes expanded. Write a short report showing how you verified that your implementation of alpha-beta pruning was working and the reduction in nodes expanded with alpha-beta pruning. Estimate the reduced branching factor after applying alpha-beta pruning. Run on the server and compare your performance to your program from Part 2, the random player from the server, and the program on the web page.

Part 4: Implement iterative deepening and transposition tables Due February 18, 2013 at 12 midnight

Add iterative deepening and transposition tables to your program. Write a short report detailing how these enhancements improved your program (faster search/greater depth). Compare against your previous program and the instructors program on the web page.

Part 5: Implement UCT Due February 25, 2011 at 12 midnight

Implement the UCT algorithm for Breakthrough. Compare the playing strength of your UCT and alpha-beta programs to see which one is stronger.

Part 7: Write the strongest program you can Due March 11, 2013 at 12 midnight

Using any techniques that you would like, improve the strength of your Breakthrough program. As you make each improvement to your program, compare the winning rate against your previous program and the program on the server. You may also want to play against other classmates to compare performance.