CSI 535 – Introduction to Artificial Intelligence : Assignment #1

20% of Final Grade, Due: 04/05/2003 at the start of class
Late Policy: You lose one full grade for every week (including partial weeks) the project is late.

In homework #3 you worked in groups to use minimax to create agents to solve 4-in-a-row Tic-Tac-Toe in a 4x4x4 cube. In this assignment you will build upon this work, but you must work only by yourself. You will design an agent to play crosses (X) and I will tell you what noughts (O) will play. Your agent will be MAX. Read the question clearly, answer all questions succinctly, and make sure you answer addresses the question.

Note, for questions 1 through 4 you will need to turn in the moves that each agent made for each game and the result. You will also need to turn your code in.

**Question 1) (20 points)**

Write a program (any language) that plays Tic-Tac-Toe using the mini-max strategy. Crosses will play a general purpose evaluation function of your choice looking no more than five moves in advance. You cannot use the same function as your opponent. Your opponent (noughts) in this question will be a random player. In subsequent questions when your player is not random, they cannot look more than five moves in advance.

Describe your evaluation function.
Play 100 games (episodes), reporting on how many times your agent won, the average number of moves and the average number of nodes searched in the game tree

**Question 2) (20 points)**

Change your implementation from question 1 to incorporate alpha-beta pruning.

Once again play 100 games (episodes), reporting on how many times your agent won, the average number of moves and the average number of nodes searched in the game tree

**Question 3) (20 points)**

Change your implementation from question 2 so that your opponent (noughts) evaluation function is:

Number of open rows/columns/diagonals for noughts - Number of open rows/columns/diagonals for crosses

Once again play 100 games (episodes), reporting on how many times your agent won, the average number of moves and the average number of nodes searched in the game tree

**Question 4) (25 points)**

Alpha beta pruning of the game tree is most profitable when the moves are ordered in monotonically decreasing or increasing order of the evaluation function. Suggest a principled method of determining the order of how to expand nodes in the game tree. This will depend on your evaluation function.
Play 100 games (episodes), reporting on how many times your agent won, the average number of moves and the average number of nodes searched in the game tree. The expected number of nodes searched in the game tree should be less than Question 3).

**Question 5) (15 points)**

Each time your agent played a new game, it forgot its previous experience. Discuss a principled method of incorporating experience into your evaluation function. Note I do **not** want you to mention particular learning techniques (we haven’t covered any as yet!).