Due: Friday 03/09/05  
Worth: 20% of Final Grade  
Late Policy: You lose one full grade for each week (including partial weeks) you are late.

2D Tic-Tac-Toe (a.k.a. noughts and crosses) is a game involving a 3 x 3 grid on which your player places crosses (X) on the board and your opponent places noughts (O). You win if you get three X’s in a row (linear arrangement) including on a diagonal and your opponent if three O’s are in a row. If the board fills up and no player wins, the game is a draw. In this assignment I will refer to 1 as the top left hand corner position, 2 the top middle position, 3 the top right hand corner position, 4 the middle left hand position etc.

In this version of the game your opposition will attempt the strategy of playing the board in the following order: 5, 3, 7, 6, 4, 2, 8, 1, 9. In this assignment you will use Q-learning to learn to defeat this player! It is a good example where working out the best strategy to defeat this player is not obvious so we can let a machine learn it.

Submission Requirements: Submit answers to the following questions and your code (email code to: davidson@cs.albany.edu). Be prepared to show me your code executing.

**Question 1): Problem Formulation (20 points)**

Formulate the problem of learning a policy to defeat the noughts (O) player as a Q-learning problem by documenting: a) The state space, b) the reward function and c) the set of possible actions. Also d) clearly state the form of the state transition function \( \delta(s,a) = s' \) with a concise example.

*You will get more points the more efficient your problem formulation.*

**Question 2): The Q-Table (20 points)**

a) Describe the form of the Q-table that will be learnt, b) What will the initial table values be, c) Once the Q-table is learnt, how shall you use it to determine what action to play given a state.

**Question 3): Learning the Q-Table (40 points)**

a) Write down the algorithm in pseudo code that you will use to learn the Q-Table. Be sure to state what the termination criterion of the algorithm is.  

Your algorithm should be able to learn to defeat the noughts player (how long this will take depends on many individual factors).  

b) Plot the payoff to your player as a non-overlapping average every 100, 200 … moves.  

c) By examining the Q-Table, efficiently describe the policy your agent has learnt.

**Question 4): (For Graduate students only) – Testing the Theorem (20 points)**

a) Does examining the answer to 3b) satisfy the result that Q-learning will geometrically converge to the optimum Q values. Explain your answer carefully.  

b) The above policy learns an optimal strategy to defeat this particular player. How could you train your algorithm to learn an optimal strategy to defeat any player.