**Super Tic-Tac-Toe**

# Introduction

We will develop a super Tic-Tac-Toe game, which supports two modes, namely, man-vs-AI and AI-vs-AI, where AI means artificial intelligence.

# Project Requirements

1. The game can be configured into stand-alone mode or network mode. Under the stand-alone mode, human beings will play with AI (i.e., man-vs-AI). The architecture for the stand-alone mode is Browser/Server (B/S).
2. Under the network mode, your system can be configured for AI playing with AI.
3. The architecture for the network mode will be peer-to-peer (P2P). Two systems should be paired together before any matches can start.
4. The chess board for this game will be 3 × 3.
5. To prevent a player from thinking for an indefinitely long time, the game should allow users to configure the time-out threshold. If a player does not make a move after the threshold is exceeded, the player loses the game.
6. All the teams should work together to decide a protocol regarding how two systems should communicate with each other.
7. At the end of the semester, we will have a contest among all the teams. The champion will receive the full credits (10 points). The second place will receive 9 points, the third place will receive 8, …
   1. We will draw to decide which two teams will play together.
   2. To avoid ties, we will make the “death match”. Each player starts with 4 counters (like X's and O's) and takes turns placing them on the board as in Tic-Tac-Toe, with the goal of getting a 3-in-a-row. However, if the game is a draw after each has played their 4 counters, they take turns sliding a counter along the lines into the space that is empty. The winner is the first player to get 3-in-a-row. More details can be found from <http://www.math.cornell.edu/~mec/2003-2004/graphtheory/tictactoe/howtoplayttt.html>.
   3. Every two teams will play two games. If each wins a game, they will draw the lottery to decide which team can enter the next level.

# Tools to Use for Design and Development

1. You must use some UML CASE tools to make your analysis and design, such as IBM Software Architect, Rational Rose, or Microsoft Visio.
2. You must use Microsoft Project to work out your workplan and use MS Project to keep track of the progress of your project.
3. You may use Microsoft Visio to draw Entity-Relation diagrams (ERDs), and data flow diagrams (DFDs).
4. You must use some Unit Test tools (e.g., JUnit, CppUnit, etc.) to perform automated testing.
5. You must use some version control system to keep the consistency of your code. Possible choices include CVS, subversion, Microsoft team foundation server, etc.