

Andrew Jacober, Sarah Hand, Mira Kasari

**Describe implemented solutions in a way that it should be possible to understand your solution without having to look at your source code. In particular you should describe how you update the utility for each square as the result of each trial, how you have chosen which moves to make during the learning process and how (if at all) you have chosen to handle special cases like those I listed above.**

We initialize a 8x8 tic-tac-toe board conceptually organized as a 4x4x4 board in 4 separate 4x4 layered boards. All of the squares are numbered with the first square (top left) of the first layer board being square 0 (0,0) all the way to the final square (bottom right) of the fourth layer board being numbered square 63 (7,7).

The algorithm is based on Q-Learning. To play, the player (computer AI) will look at all the positions available and choose a random value between 0 and 1. If the random value is less than the exploration rate, then the player chooses a position at random (exploration). If the utility value is greater than the exploration rate, then it will choose from the learned positions which are stored in a Q-Table with a value assigned to each state (exploitation). Therefore, the resulting policy only has values for states that have been explored. The more games the AI is able to play, the larger the Q-Table is likely to be since it has more opportunities to explore. The first round ultimately becomes random by default since there is nothing learned yet. Then, the board is updated with the action taken. Once the game is over (win, loss, draw), the player is then given a reward based on the outcome (+1 for a win, +0.05 for a draw, -1 for a loss all fed into the utility function), the reward is propagated throughout the Q-Table, and the board is reset. Otherwise, the play continues, alternating between each player using their learned or explored values until there is a win or draw. Due to time and bandwidth limitations, we chose not to handle any special cases.

Andrew did a lot of the heavy lifting when it came to figuring out code, Sarah and Mira did a lot of the understanding and interpreting of the desired goals, proofreading and testing. When it was not possible to meet as a group, we each looked at the code and came up with our own thoughts, then brainstormed our thoughts together when we did meet so that the process would be smoother. However, we all collaborated on all parts of the assignment.