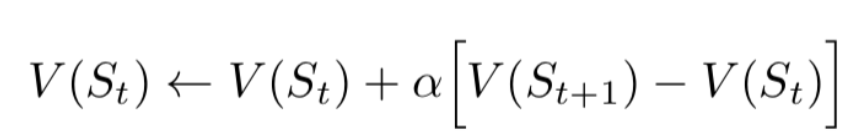
Rupesh Gudipudi, Varun Srivastava, Ayush Jha

14 November 2019

**COMP 560 Assignment 2 – 3D Tic Tac Toe**

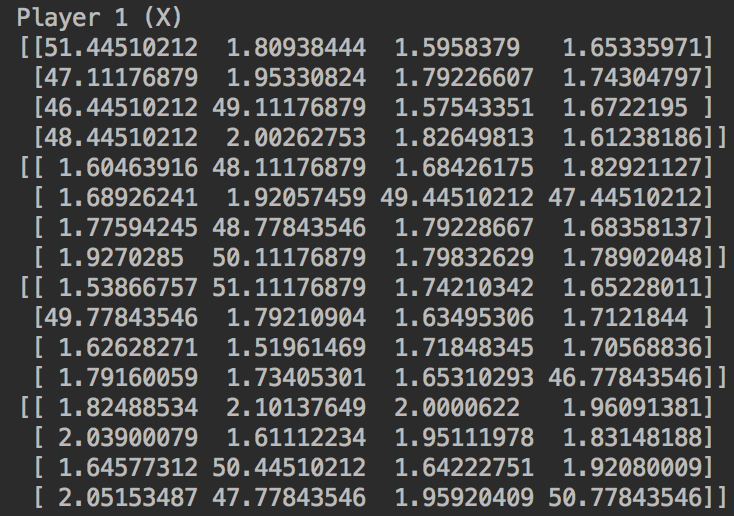
**Implementation**

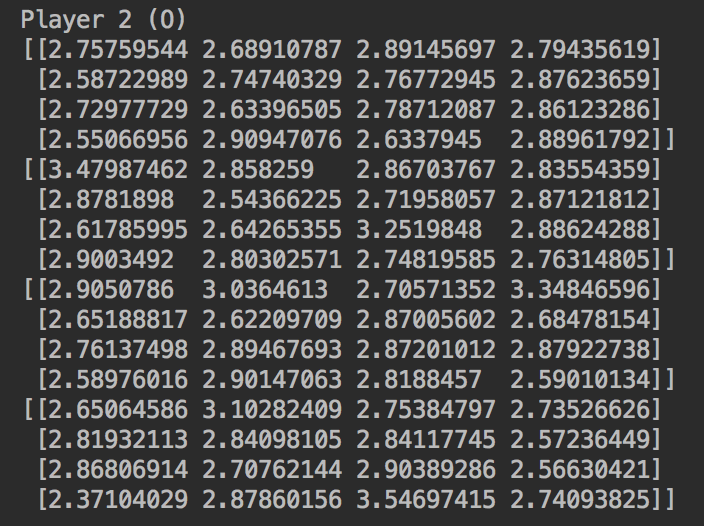
For the first half of the trials, our program performs exploration where we randomly fill the planes to get preliminary awards for both player 1 and player 2. We did this by saving all the plays in coordinate lists, in which we would apply our reward function to depending on the outcome of the game. The player that won would receive a reward of 1, which would be applied to the first move of the tic tac toe game and discounted by 0.75 each move after in the application of this reward function:



A loss would result in no updates of the function and a tie would result in a reward of 0.1 for both players.

After the first half of the trials were completed (exploration), exploitation is used where we used the obtained reward functions for each of the 2 players. Each player would play the spot that corresponded to the maximum value in their reward function thus far. From here, since the random element was eliminated, the player with the more optimal play style would win repeatedly and would cause the rewards for the best spots to be highlighted. We can see an example of this below where Player 1 (X) had the more optimal play style and had the very high corner, center, and middle plane values (it is visualized as 4 4x4 planes stacked on each other). These values came on top since they correspond to numerous ways of winning.



****

Here we see the very large values of Player 1, highlighting the flexibility of winning via corner, center, and middle plane locations (can win via diagonal, vertical, or horizontal). It is important to look at these rewards in relation to one another for move importance.

**Contributions**

Ayush: Implemented the utility function into the 3D Tic Tac Toe game

Rupesh: Worked on creating the 3D Tic Tac Toe game based on the different winning lines

Varun: Worked on creating the 3D Tic Tac Toe game based on the different winning lines

Together: Found what the different winning lines were and ideated what kind of utility function to use