Zi Yan Zhang

For the basic decaying epsilon greedy agent, a Q-Learning agent was the basic agent prepared to be used for the HexGame competition if the Neural-Network based agent did not function properly. The only set of hyper parameters this agent has are the values of epsilon, gamma, and alpha. These values are set to Ɛ=0.1, α=0.02, and γ=0.98, which are the values used for this class for the Q-learning agent. The Q-Learning agent saves the Q-value tables into a JSON file after each game play. When a new game is launched, the old Q-value files are loaded and are used as the Q-values of the current game, and are updated from that point on. Since Q-Learning agents use epsilon-greedy method that relies on Q-values to get the most optimal actions, by saving the Q-values after each successive play and re-using them as the starting point for a new game, the Q-values become more accurate when the agent plays more games and thus, is being trained. The agent handles the difference between two different reward systems by saving to and loading from two separate files based on whether the reward system is sparse or dense. The agent takes an initialization parameter that specifies whether the current game will have dense rewards. If dense reward system is used, then it will save and load the Q-values from dense reward environments. This goes the same for sparse reward systems. So dense reward Q-values and sparse reward Q-values are stored in separate files and are used accordingly