**Reinforcement Learning in Pursuit Evasion Games**

This project simulates a game of tag between two robots in a grid environment using the Q-learning algorithm. One robot acts as the pursuer, trying to catch the other robot, which acts as the evader.

**Environment**

The simulation environment, TurtleBotTag, is a custom environment that follows the OpenAI Gym interface. It consists of a 20x20 grid with obstacles placed in a pattern. The pursuer and evader are placed at random positions on the grid at the start of each episode.

**Q-learning Algorithm**

Both the pursuer and evader use the Q-learning algorithm to learn their respective strategies. The Q-learning parameters are as follows:

* Learning rate (eta): 0.005
* Discount factor (gma): 0.9
* Exploration rate (epsilon): 1.1 (decaying over episodes)
* Maximum number of steps per episode (step\_num): 999
* Number of episodes (epis): 200000

**Simulation and Rendering**

The simulation runs for a specified number of episodes, with each episode having a maximum number of steps. The environment is rendered every 50000 episodes to visualize the progress of the robots. The rendering shows the grid, the robots' positions, and their fields of view.

**Installation**

To run the simulation, you will need to install the following dependencies:

pip install numpy --user

pip install gym --user

pip install matplotlib --user

**Usage**

To start the simulation, run the following command:

python runsim.py