CISC 453/474 Project – Navigation of a robot using Q-learning

Group: Student 1, Student 2, Student 3, Student 4

Description: This project will deal with the problem of a robot that navigates in an environment with obstacles. The obstacles can be stationary (similar to furniture) or dynamic (mostly people moving). The dynamic obstacles are assumed to be identified as such. The environment will be represented as a grid that is assumed not to be known beforehand. The moving objects behave following a stochastic model that is fixed, but unknown to the robot. The stationary obstacles are in fixed positions. The grid is a space of 20x20 cells. The robot's states are represented by its position and orientation (north, south, east or west). The motion is always according to the robot's orientation. The robot can also choose to stay put or turn. The actions available to the robot are:

1. Move one square to the front;
2. Turn 90^o clockwise;
3. Turn 90^o counterclockwise;
4. Stay put.

Rewards: The robot will receive a reward of -1 for the execution of an action. If the robot collides with an obstacle, it will receive a large negative reward (at least one order of magnitude larger than the reward of -1). When the robot reaches the goal, a positive reward will be given.

Objective: The robot will learn how to leave the maze (achieve a goal position) without colliding with stationary or dynamic obstacles as fast as possible.

Learning technique: The algorithm will be a Q-learning algorithm with eligibility traces. It is expected that the use of eligibility traces will speed-up the learning.

Implementation: The implementation will be done in Python 3. Only standard libraries will be used.

Deliverables: The project will generate the following deliverables:

1. A grid world simulation of the task. Obstacles will be represented as occupied cells in the grid. Stationary obstacles will be represented as black cells. The robot will be represented as a blue triangle, pointing to the robot's orientation. Moving obstacles will be represented as red triangles.
2. A learning environment to generate the robot's experience. The motion of the moving obstacles will be implemented as a Normal distribution, with the center of the distribution being a motion following the obstacle' orientation. The obstacles can also move to other neighboring cells (north, south, east or west).
3. Data showing the evolution of learning and plots of the data (in terms of the value of specified states and success rate of solving the maze).
4. A report describing the algorithms used, and results shown as graphs demonstrating learning of the task.