Name: Camilo Girgado & Jason Patel

Class: RBE 550

Assignment: Wildfire

Task

In this assignment, we were tasked setting up a dynamic simulation between two objects of interest (a Wumpus and a firetruck) that both had different goals to achieve in our grid world. The goal of the Wumpus was to start fires and the goal of the firetruck was to put out those fire in an effective manner (closest available).

Approach

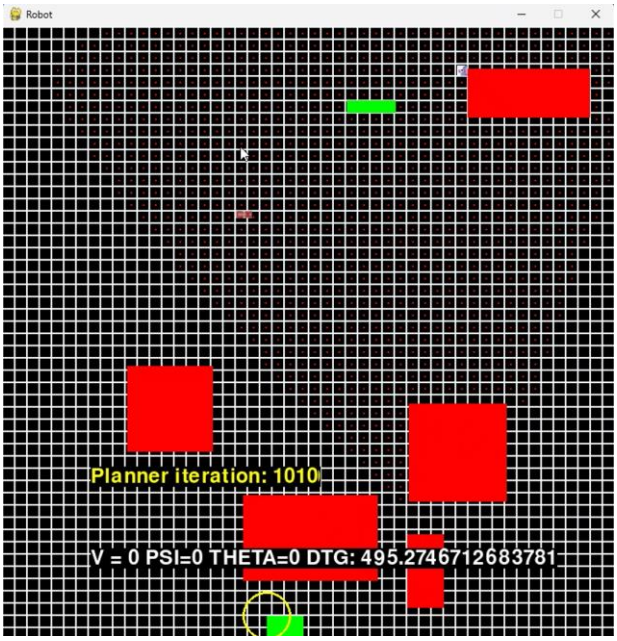
We took the approach of using a typical grid world similar to previous assignments with hardcoded obstacles. We started with randomizing the obstacles but that proves to be time intensive when working with the planning of the Wumpus.

We use pygame to display the relevant robot information as well as the obstacles and goal indicator. We use collision detection to check for collisions between a given rectangle and the existing obstacles in the environment.

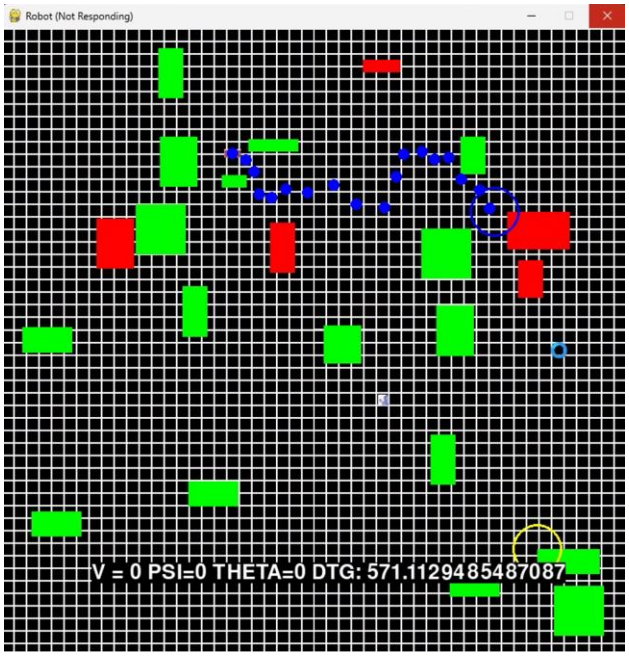
We give the obstacles various states of being, burned and extinguished. This drove the Wumpus to be able to search for the obstacle or re-search for it if it had previously been lit on fire.

Probabilistic Roadmap Planner - Firetruck

For the probabilistic planner that drives our firetruck, we initialize 1000 samples to build a pre-processing base so that queries can be made to move to a goal of interest.



-Figure 1: Wumpus Search Depiction-



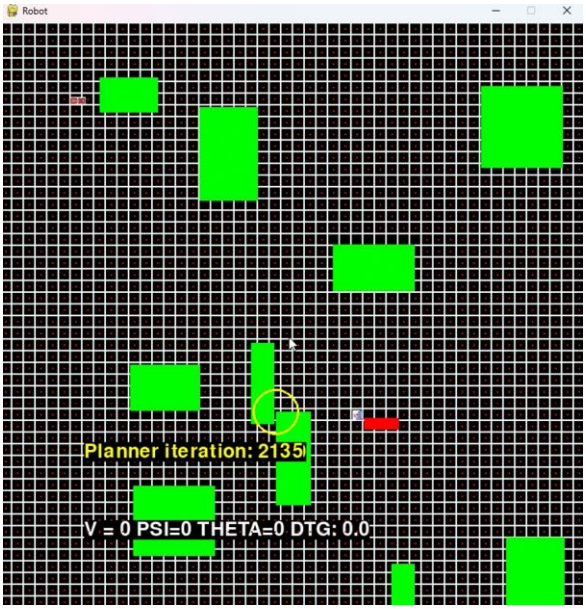
-Figure 2: Firetruck path depiction-

aStar* Planner - Wumpus

Our Wumpus is driven by an astar planner. It takes the start and goal locations, an obstacle grid, a robot, a Pygame display, and a callback function for writing information. The search method iteratively explores potential states, updating the priority queue based on the cost-to-come and cost-to-go. The algorithm considers the robot's neighbors, evaluates their costs, and avoids collisions with obstacles. The step and step2 methods help determine the next state to explore or the final path. The script focuses on finding an optimal path for the robot while adapting to its discrete state space and avoiding obstacles.

We convert the coordinates in our “convert row/column to x/y” lines to give a proper conversion between the grid coordinates and the screen coordinates. The Wumpus uses the get neighbors method to generate a list of neighboring states based on the current location. It uses a simple grid system with up, down, left, and right movements.

We then update the robot's position to match the specified destination state (nextMove). The destination state is an instance of DiscreteState.



-Figure 4: Wumpus lighting a fire -

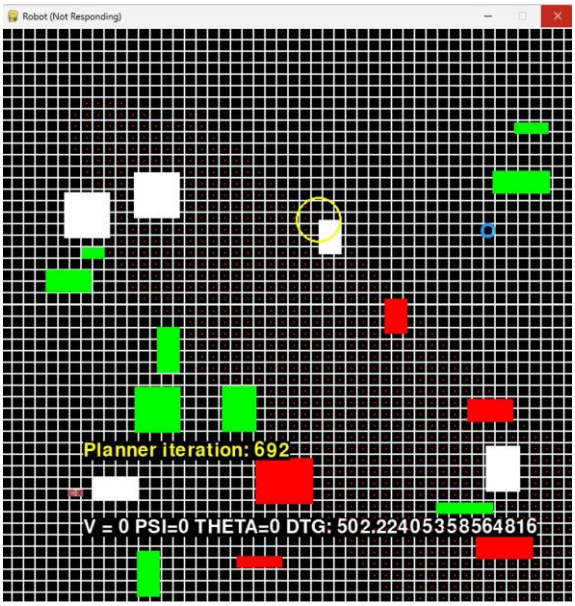


Figure 5: Lit, unlit, and extinguished obstacles

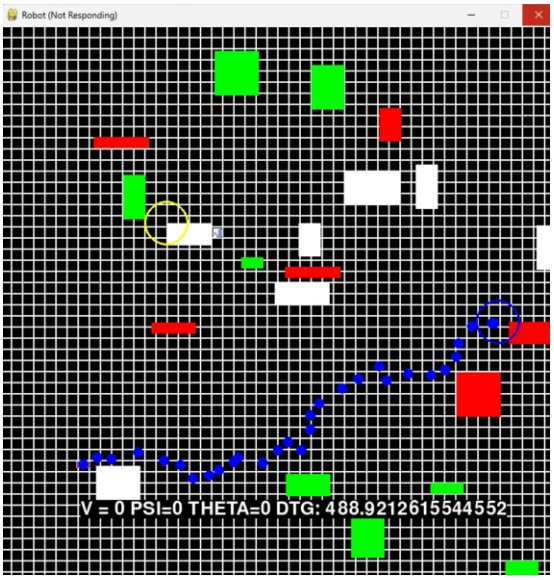


Figure 6: Run 2

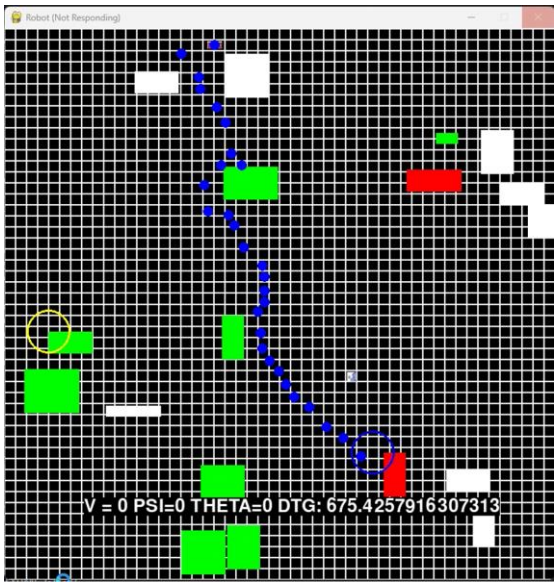
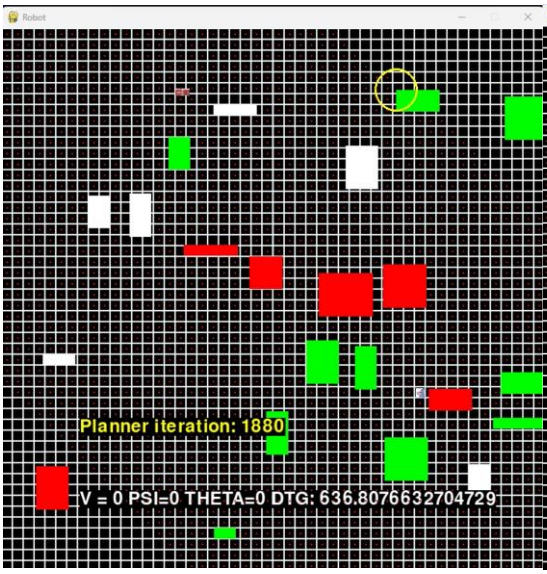
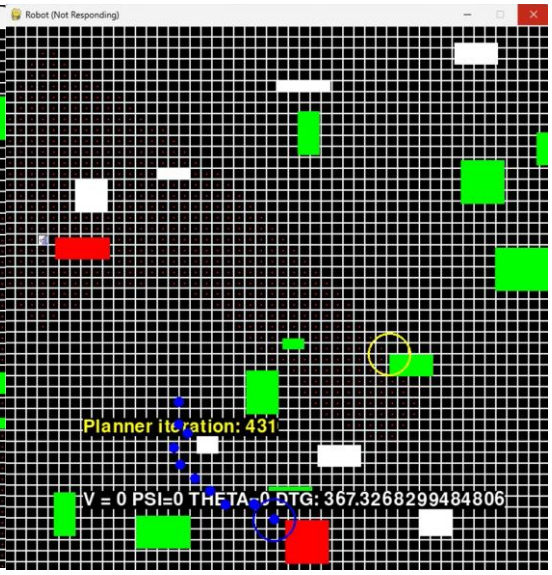


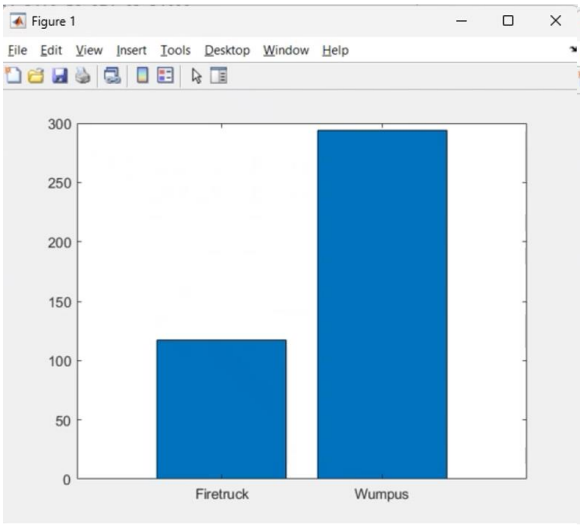
Figure 7: Run 3



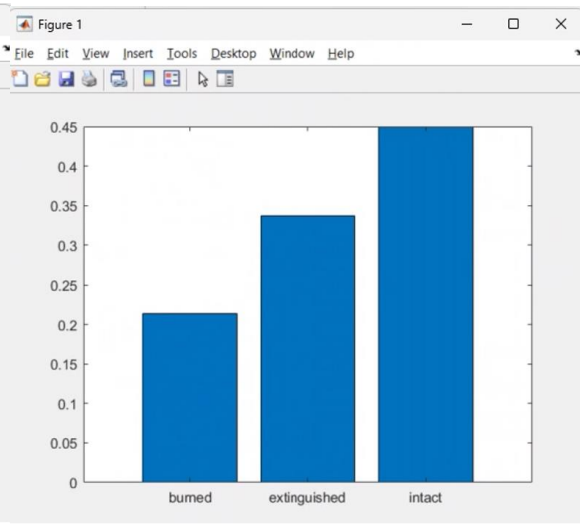
-Figure 8: Run 4-



-Figure 9: Run 5-



-Figure 9: CPU Time-



-Figure 10: Averaged Obstacle States-

Findings:

The probabilistic roadmap planner is more robust to complex and high-dimensional spaces, as it doesn't rely on an explicit grid and can handle continuous state spaces more effectively. We'll hit on a few categories below to compare the two planning methods.

Completeness and Optimality

- A* is complete and optimal when applied to a discrete grid. However, its performance may degrade in high-dimensional or continuous spaces. We saw this when we first had our lattice planner on the firetruck, every iteration would bring the entire grid space to a pause and it would simply take an incredible amount of time per obstacle.

- PRM is probabilistically complete, meaning it tends to find a solution as the number of samples increases. With our number of samples, we found optimality to be accomplished more so than with the sampling based planners. I'm sure as the sample number goes up, the more optimal the results would be.

Computational Efficiency:

- The sampling based planners were exhaustive throughout the entirety of the runtime. This would really only get worse as dimensions and complexity increase.
- Once the roadmap is constructed with our PRM, we saw major efficiency gains as it focused on sampling and connecting relevant points.