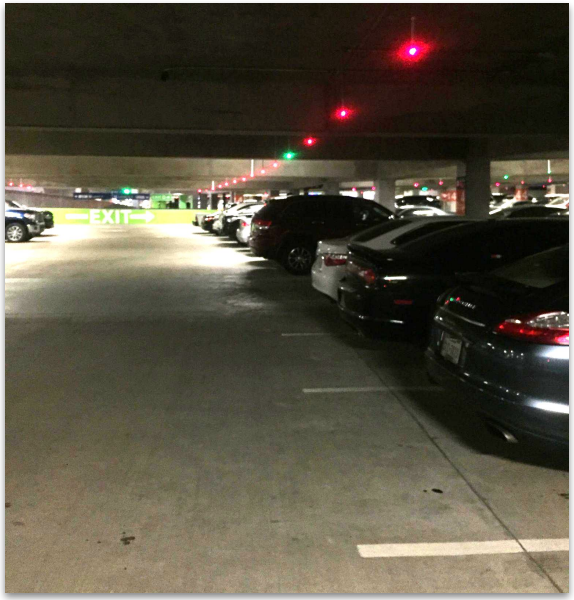
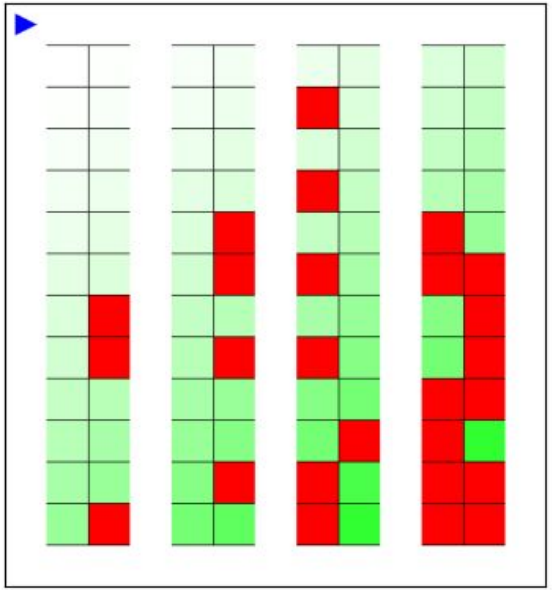
Parking Lot Simulator

Created by: Ethan Lott, Josh Pollack, Pushkar Bhargiri, Patrick Marinich, Asher Anand

Real World Application





Our Goal:

- Use a Q-Learning model to simulate the task of finding a parking spot in a crowded parking lot.
- Make as realistic as possible:
  - Spots semi-randomly filled on each run
  - Driver must enter the lane before pulling into a spot
  - Aim to get a close spot to the exit
- Utilize Python to create the model
- Create a visualization

Technical Approach

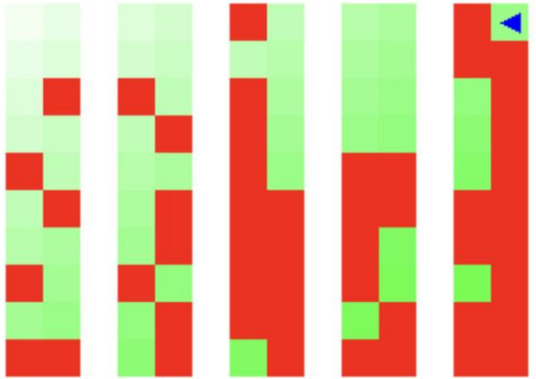
- The structure of the parking lot will not change
- The starting position can be considered constant (for this initial development)
- The positions of cars can be considered static for a sufficiently small period of time
- Q learning can be solved quickly and will converge to an optimal solution
    - This allows it to be re-trained for new cars that enter
    - It has limitations if the state changes (i.e. spots become full or empty after computation)
- This iteration of the project has a low computational cost, works well for smaller lots with less traffic
- The “optimal location” is considered the bottom right spot.

Constraint definition

- The car should follow normal driving conventions when navigating the parking lot
- It should also avoid collision with currently parked cars
- In real life parking lots are different sizes, so we allowed for the parking lot to be defined as different sizes
- Parking lots typically do not fill up uniformly, but the best spots are taken first, so our random spot filing accounts for this.

Development challenges

- We learned that reward functions must be carefully designed
- Our initial agent had some unexpected behavior, since it would enter from the side of a parking spot
- As the parking lot gets larger, the agent may fail to be optimal
    - This is due to the Q-Learning algorithm not finding the optimal solution during the training period and thus the policy never contains the best possible solution
    - To mitigate this effect on the larger parking lots:
        - Training can be longer
        - Q-Learning parameters of: Exploration, Gamma, Alpha can be tuned



Future Directions

- Currently our model requires a static environment
  - Decent assumption for a small parking lot and a short search time
  - For larger parking lots another type of learning may be necessary to account for dynamic events like other cars leaving.
- Update our model to have 3D capabilities for parking garages and designate rewards based on floor and proximity to elevator/stairs
- Allow users to select their preferred sections of the lot when multiple attractions are nearby (Example: Shopping Mall)