Self-Driving Car AI

Group 1

“Self-Driving Car AI” is a combination of a GPS brain and a driver agent AI. The GPS brain picks the shortest path from the source to the destination. The driver agent AI drives the car (accelerator and steering) along the GPS specified path.

Topics/Components of our project

Component 1: GPS Brain - Search algorithms

We will use various search algorithms to decide the shortest path between the source and the destination that the user selects. We will test multiple search algorithms such as A*, BFS, and DFS and compare the results. Our final model will use the algorithm with the best performance and the fewest tradeoffs.

The solution path by the GPS brain (search algorithm) will then be handed to the driver agent AI (component 2) to drive the car along that route.

Component 2: Driver agent AI - Reinforcement Learning Model

The GPS Brain (component 1) only describes what path (roads and turns) the agent should take to reach the destination. It gives no information about how to drive the car. This is where component 2 comes in.

This component of our project is an AI model for the driver of the car. The driver agent AI will be trained using Reinforcement Learning.

The reinforcement learning model will optimize the following driving tasks:

- Stay centered within the lane
- Avoid collisions
- Obey rules at traffic intersection with a signal light

Properties of Environment:

- Partially observable: The agent sees primarily out the front of the car, though it receives limited input from the sides and back.
- Nondeterministic/Stochastic: The car may skid
- Sequential: The previous states/actions affect the future states/actions
- Dynamic: The environment changes while the agent is deliberating
- Continuous: Time is continuous. The exterior distance sensor values are also continuous.
- Multi-agent: Multiple car agents are each trying to maximize their performance measure

PEAS:

- Performance Measure: Least time to reach destination, no collisions
- Environment: Car, other cars, road, road divider lines, traffic lights
- Actions: Steer left, right; Apply accelerator, break; Do nothing
- Sensors: Detect distances from the left and right road divider lines, detect traffic lights and the current traffic light color

Evaluation Criteria for the Agent (reward function for reinforcement learning):

- Time to reach the destination: minimize
- Number of collisions: minimize
- Number of traffic light violations: minimize
- Distance from center of the lane: minimize

Tools Used

- Unity: to create the game environment
- Unity ML bridge: to link the Unity game environment with our Python AI agent
- Python: Our AI agent will be written in Python
- PyTorch: We will likely use a neural network to train our reinforcement learning model. We will use PyTorch to create and manipulate the network.