Team Based Car Racing

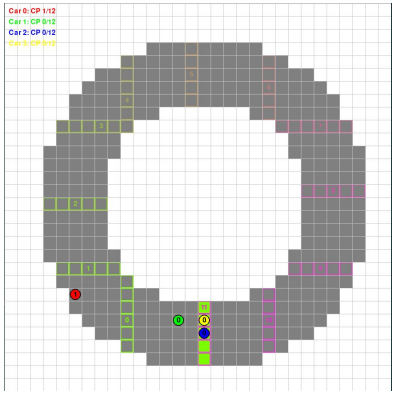
ECS427: Multi Agent Reinforcement Learning
Semester Project Presentation

Akshat Singh
Ananya Gandhi
Chinmay Mundane

Introduction

- In racing sports, manufacturers field multiple vehicles, with overall rankings based on how well their cars perform collectively. This strategy maximizes their chances of winning.
- This creates a cooperative-competitive dynamic, where individual drivers must balance personal performance with team strategy.
- In this project, we try to mirror this idea by having two teams compete against each other, each having two cars.
- The team with at least one agent finishing first wins the race, reflecting the dynamics of team-based competition in racing.

Environment Creation



The Environment is a 30 x 30 grid with a circular track of width 5 cells

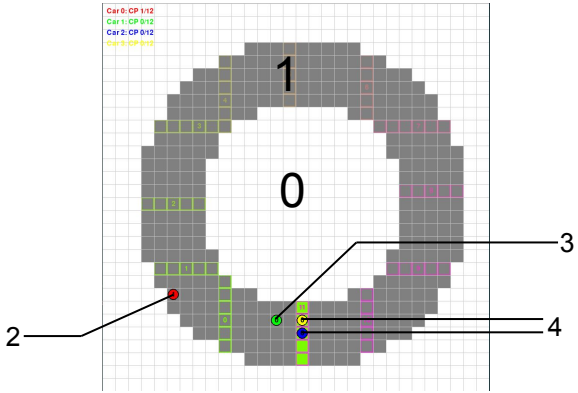
There are 4 agents present in the environment, each start at the starting line and have to complete one clockwise lap of the track

The agents can move left, right, up, down or stay at a place

If they go outside the track, they stay at the same place, if they collide with any other agent, they stay at the same place for the next two steps

The episode is finished if any one agent completes one lap

Action and Observation Space



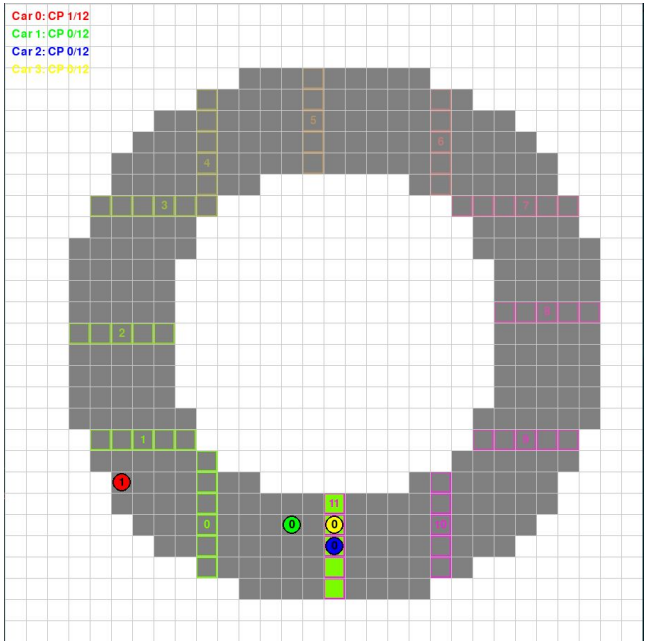
For each agent, we have 5 actions: Left, Right, Up, Down and Stay

Each agent has the following observations:

1. The matrix of the environment, The environment matrix consists of cells labeled as follows: 0 for unwalkable cells, 1 for walkable cells, 2 for the agent, 3 for the teammate, and 4 for enemies.
2. The angle made by the agent with the center for: itself, its teammate and enemies

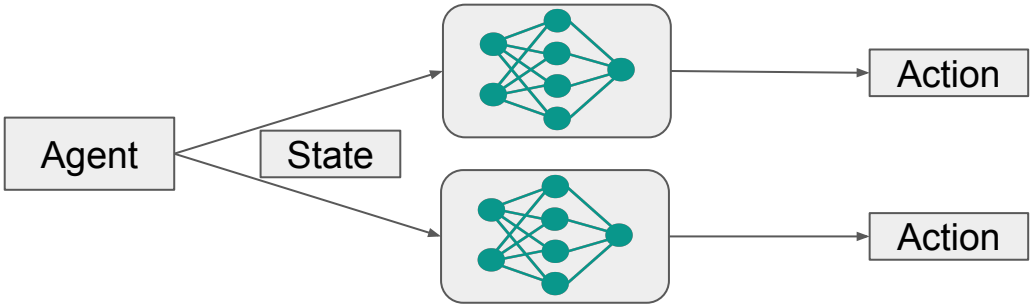
Reward Structure

- **Step Penalty:** To minimize the time taken to complete the lap
- **Angle Reward:** To encouraging clockwise movement.
- **Teammate Completion Reward:** To promote cooperation
- **Angle Comparison Reward/Penalty:** To make sure the agent tries to stay ahead of others.
- **Enemy Completion Penalty:** Penalty when enemy completes the lap



Algorithm Selection

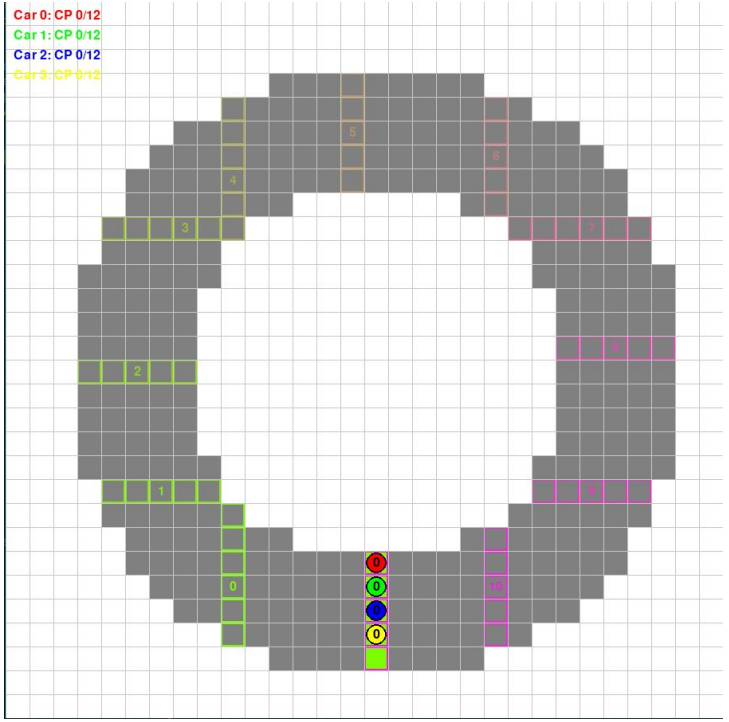
- Multi Agent Deep Q-Network algorithm was selected for training
- A Centralized Training, Centralized Execution (CTCE) approach is used, where both the Q-Network and the Target Network receive states individually, and actions are taken individually for each agent.



- The network has 2 hidden layers with total 904 inputs, 128 nodes in each hidden layer and 5 outputs

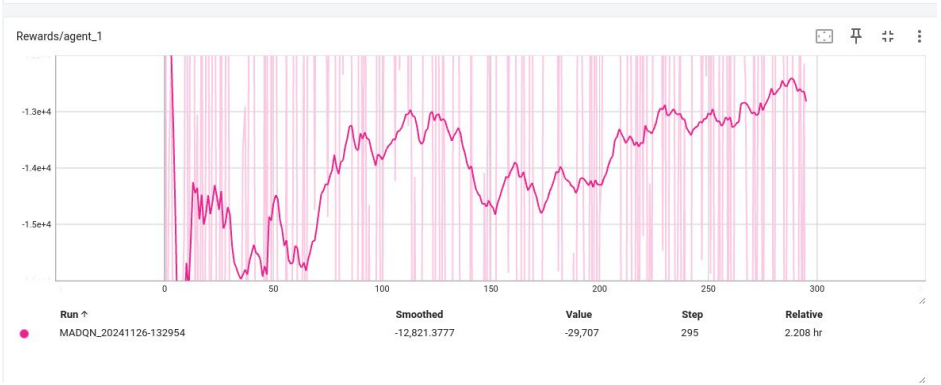
Training Process

- Four agents are placed at the starting line at the start of each episode.
- The two agents that we want to train will have epsilon = 0.1 and discount factor (gamma) = 0.9 training with the MADQN algorithm.
- The rest of the two agents also train on a MADQN network but with epsilon = 0 and gamma = 0, taking a greedy approach



Results

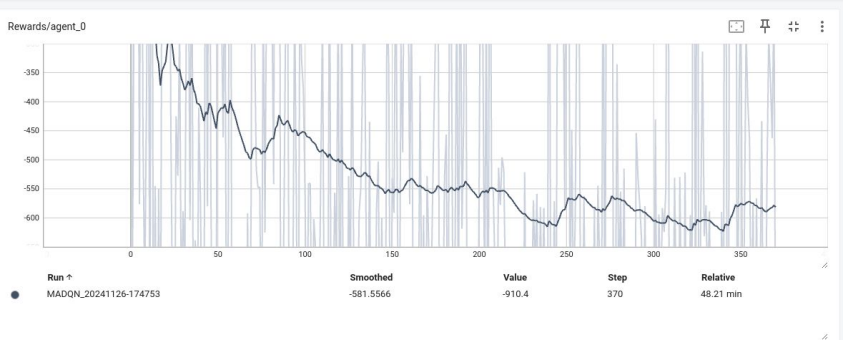
With Only 2 agents: No competitors

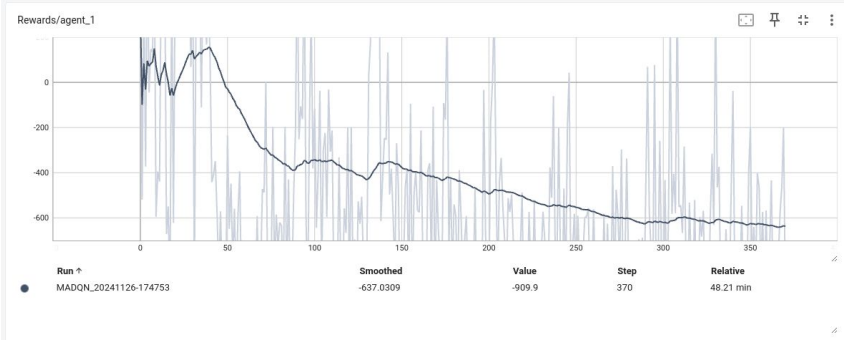


Rewards were increasing but were negative

Results

With 4 agents: Two agents using MADQN, rest two using random policy

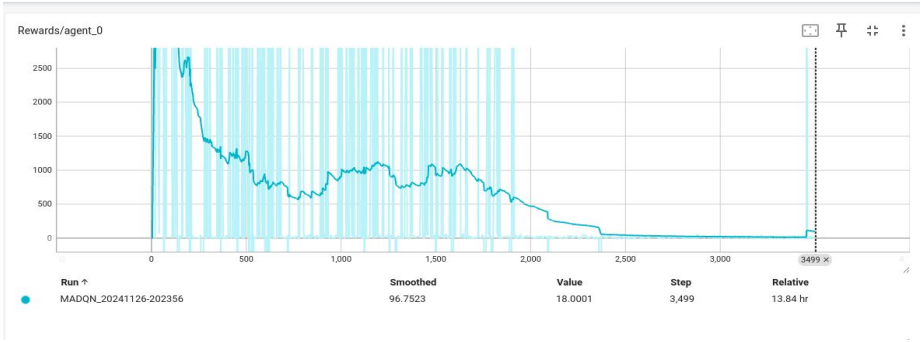


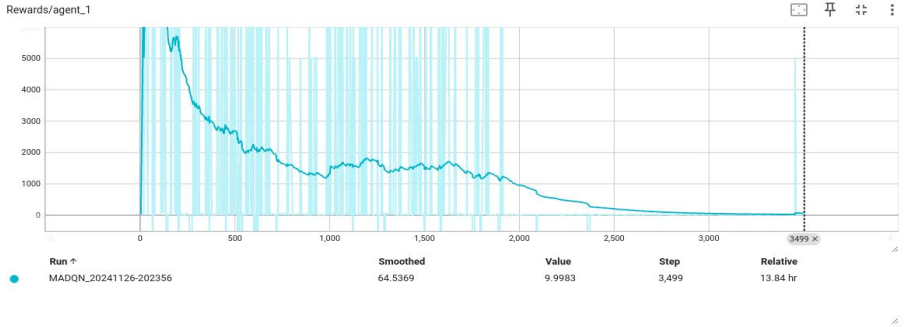


Here rewards and observations were not based on angles, so the agents were going well till halfway point and then stopping there because they could not distinguish between clockwise and counterclockwise movement

Results

With 4 agents: Two agents using MADQN, rest two using greedy MADQN

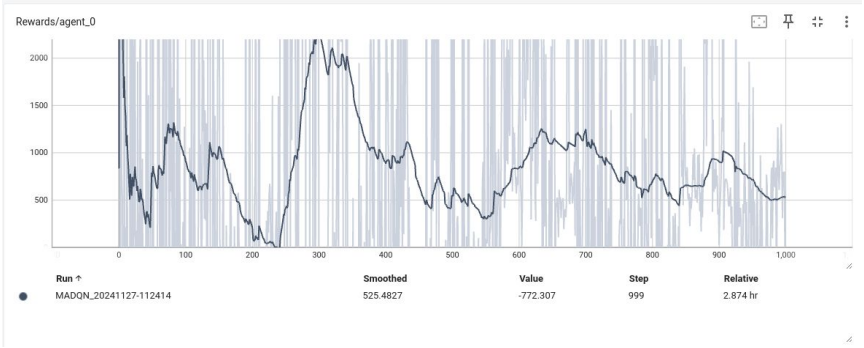


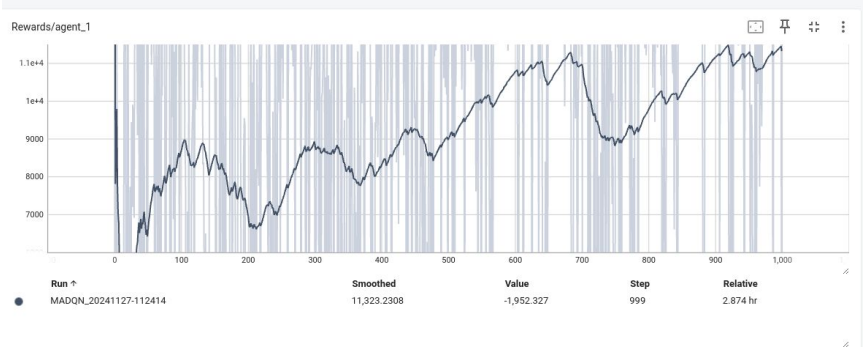


The rewards were as follows: negative for steps, small reward on angle increase and large reward on completion, the agents converging on a reward closer to zero

Results

With 4 agents: Two agents using MADQN, rest two using greedy MADQN





With the final rewards shown earlier, for 1000 episodes this showed potential as the rewards for agent 1 were increasing while for agent 0 it was not very stable, overall increasing the average reward

Thank You!