**AI-Project: Reinforcement Learning Self-Driving Car**

V Semester AI Project made by: Akshath, Sankalpa, Apratim and Raja.

**The project:** A Reinforcement Learning agent that learns to drive around tracks of varying difficulty by using feedback from sensors.

**Table of Contents**

1. Introduction
2. Setup and Installation
3. Usage
4. Simulation Details
5. Presentation
6. Acknowledgements

**Introduction**

This project leverages NEAT (NeuroEvolution of Augmenting Topologies) to train a self-driving car agent. The agent learns by evolving neural networks over generations to navigate a track using feedback from sensors. We use Pygame to visualize the simulation in real-time.

**Setup & Installation**

**Prerequisites**

1. Python 3.12.x
2. Pygame library
3. NEAT-Python library

**Steps**

1. Clone repository
2. Install dependencies: pip install -r requirements.txt

**Usage**

1. Run the model python .\Code\model.py

(change map name in simulation.py to switch maps)

**Simulation details**

**Car Sensors:**

Each car is equipped with sensors that collect data from the environment. This data helps the neural network predict the car’s next move.

**Neural Network Architecture:**

The NEAT algorithm evolves neural networks by mutating topologies and adjusting weights over generations.

**Simulation Logic:**

* Cars are initialized at the start line.
* For each car, the neural network outputs decisions (e.g., increase speed, rotate).
* If a car crashes or stops moving effectively, its fitness score stops increasing.
* The simulation ends when all cars crash or time runs out (30 seconds per generation).

**Presentation**

Presentation slides can be found in the \Presentation folder.

**Acknowledgements**

* NEAT-Python: [https://neat-python.readthedocs.io](https://neat-python.readthedocs.io/)
* Pygame: [https://www.pygame.org](https://www.pygame.org/)