Multi-Agent RL in a competitive Tron light-Cycle Environment

Juan Andrés Jimenez Palomino - 20221020087

ErickSantiago Buitrago Peña - 20221020072

UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS

Introduction

This project explores the design of intelligent agents capable of learning and competing in a dynamic, adversarial environment inspired by the classic Tron game. The challenge lies in training two agents that interact in a constantly evolving system, where every decision influences future states. Previous solutions have focused on passive strategies that maximize space control or obstacle avoidance using proximity sensors. In contrast, this work seeks to develop more strategic and aggressive behaviors, emphasizing intelligent attack decisions over pure survival.

Goal

Can reinforcement learning agents learn to compete strategically in a dynamic, adversarial environment where survival and offense must be balanced?

The goal of this project is to develop and train two AI agents capable of playing against each other in a Tron-inspired game, using a multi-layer observation tensor and a custom reward system to encourage both survival and intelligent attack behaviors.

Proposed solution

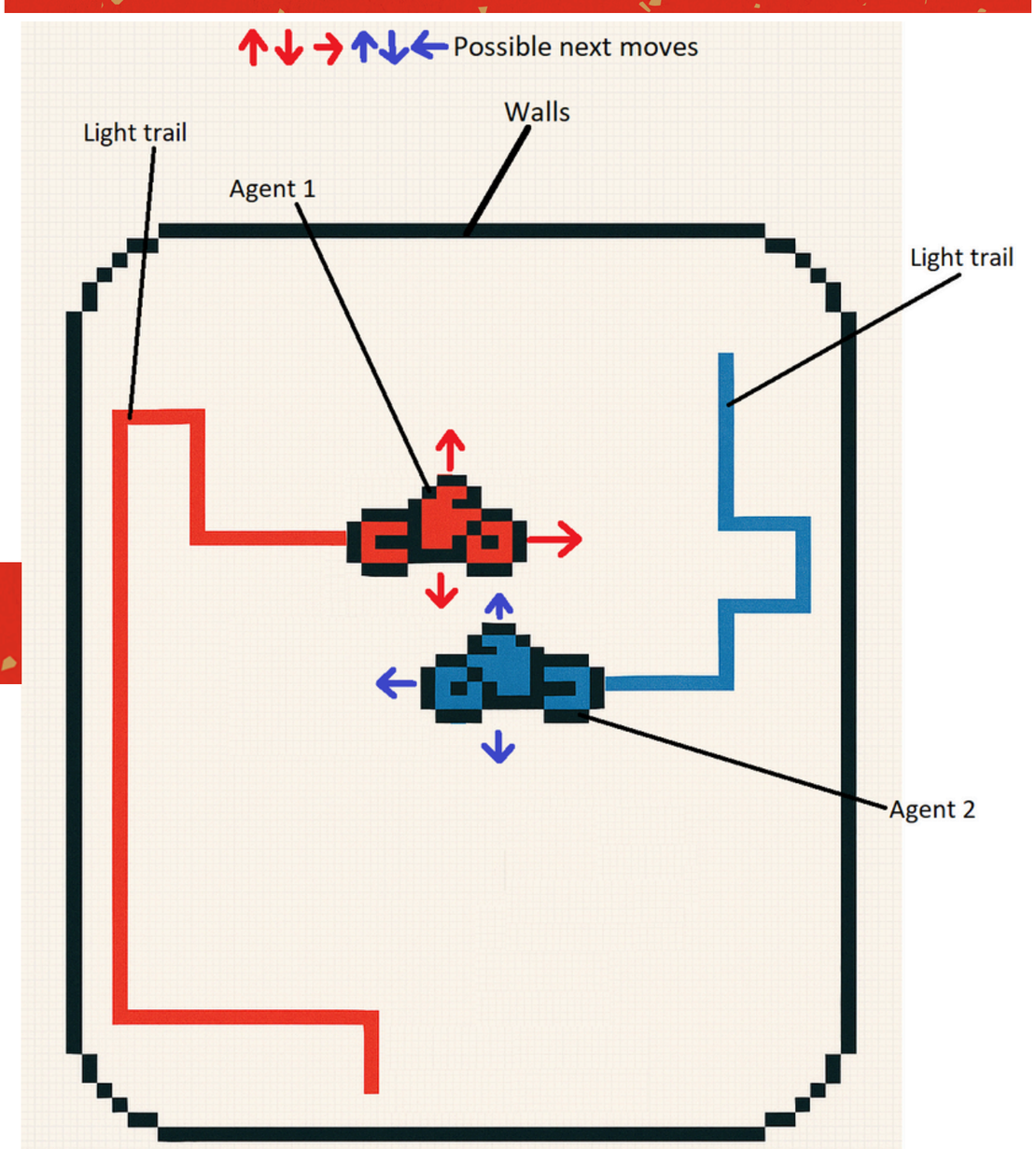
The proposed solution features four reinforcement learning agents, split into two competitive teams, in a custom Tron-inspired environment. Built with PyGame and wrapped using PettingZoo's parallel API, the environment integrates seamlessly with RLlib for multi-agent training.

Each agent receives a 3-channel observation tensor: one for boundaries, one for player positions, and one for light trails. The action space includes a toggle for the light trail, enabling strategic behaviors like feints and ambushes.

Training is performed using PPO in RLlib, adapted to resemble MAPPO by sharing policies across teammates and employing a team-based reward system. Rewards are structured to encourage offensive coordination, rewarding crashes and penalizing inefficiency. This aligns with the Kill and Goal Reward decomposition proposed by Zhang et al. (2021), which promotes faster emergence of aggressive strategies.

PyTorch was chosen over TensorFlow for its compatibility with RLlib and ease of experimentation.

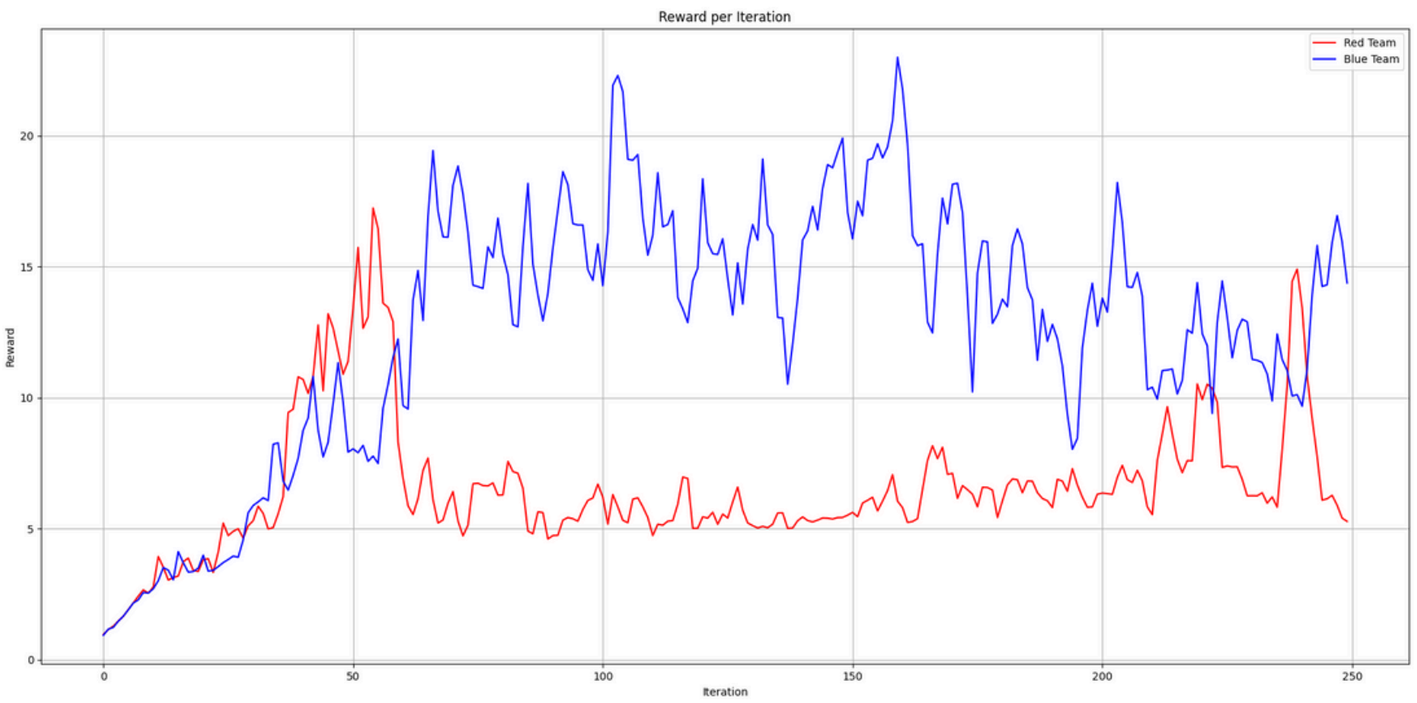
Map, players, trails and walls



Experiments and results

We trained the agents for 250 iterations, each consisting of 2400 steps. To encourage efficient learning, we modified the entropy coefficient schedule: the agent explores more at the beginning and gradually reduces exploration over time. This helps the policy shift from exploration to exploitation as learning progresses.

The following graph shows the average reward per iteration, illustrating the learning trend:



Bibliography

Zhang, S., Zhao, H., & Bachrach, Y. (2021). Multi-Dimensional Distributional DQN. NeurIPS 2021. https://proceedings.neurips.cc/paper_files/paper/2021/file/0b9e57c46de934cee33b0e8d1839bfc2-Paper.pdf

Conclutions

This project establishes a competitive multi-agent reinforcement learning environment inspired by Tron. By designing advanced observation tensors, a tailored reward system, and dynamic interaction mechanics, we set the foundation for agents to develop strategic and aggressive behaviors. While training is still pending, the implemented framework with PyTorch and MARLlib ensures flexibility for future experimentation and evaluation. This approach addresses the research goal by creating a solid platform to explore complex agent dynamics in competitive settings.