DA7400 Term Project Proposal

Crafting a Reinforcement Learning Champion in Cooperative Gameplay

Jashaswimalya Acharjee
DA23D403
Shuvrajeet Das
DA24D402

Faculty: Balaraman Ravindran
Indian Institute of Technology Madras, Tamil Nadu

1 Objective:

The task involves developing and optimizing a reinforcement learning (RL) agent to compete effectively in the `SoccerTwos` and `Dungeon Escape` game environments, which are part of the Unity Machine Learning Agents Toolkit. The main objective is to train an RL agent to outperform baseline agents using advanced multi-agent strategies, particularly focusing on Proximal Policy Optimization (PPO) and Soft Actor-Critic (SAC) as our base methods. The goal is to enhance the agent's decision-making and strategic abilities in a dynamic, competitive, and/or cooperative environment where the agents must collaborate with a teammate to score higher rewards while preventing the opposing team or dragon (the adversary) from scoring. Essentially attaining the nuance of coordination and communication among the agents.

2 Rationale:

Both the environments provides a challenging yet controlled setting for testing and evaluating the effectiveness of Multi-Agent RL algorithms.

The task was selected because:

1. **Complexity and Realism:** The game mimics real-world dynamics where agents must make quick decisions, cooperate with teammates, and adapt to the opponent's strategies.
2. **Research Relevance:** Reinforcement learning in video games, especially in multi-agent settings, is a burgeoning area of research with practical applications in real-world development.
3. **Educational Value:** The environment is designed to be a testbed for MARL algorithms, making it ideal for exploring and understanding algorithms better.

2.1 Possible Evaluation Metric

To evaluate the success of the proposed task, the following metrics will be used:

- **Mean Cumulative Reward:** This metric measures the average performance of the agent over a series of games, serving as a fundamental indicator of overall learning progress. A higher mean cumulative reward signifies that the agent is consistently making decisions that lead to favorable outcomes over time.
- **Mean Episode Length:** Mean episode length refers to the average number of steps per game. A shorter episode length could indicate that the agent is employing a more efficient strategy, effectively achieving its objectives in fewer steps. Conversely, longer episodes might suggest a more cautious or exploratory strategy, which can be beneficial depending on the context.
- **Convergence Speed:** Convergence speed measures how quickly the agent reaches a stable performance level during training. Faster convergence is typically desirable as it indicates that the agent is learning effectively and can adapt to the environment with fewer training iterations. This metric is especially important when considering the computational cost and time associated with training complex agents.

- **Elo Rating *(For SoccerTwos)*:** The Elo rating is a widely used metric in competitive environments to quantify the skill level of an agent relative to others. By evaluating the agent's Elo rating, we can assess the level of expertise it has developed and how it compares to other agents. This metric provides a nuanced understanding of the agent's capabilities, particularly in scenarios where it faces a variety of opponents with different strategies and skill levels.

3 Approach:

The approach will be divided into several stages:

1. **Baseline Comparison:** In the initial stage, we will implement and evaluate baseline reinforcement learning agents using standard algorithms such as Proximal Policy Optimization (PPO) and Soft Actor-Critic (SAC). These algorithms will be deployed without any form of curriculum learning to establish a performance baseline. This step is crucial as it allows us to understand the inherent difficulty of the Soccer Twos environment and the capabilities of our agents without additional training enhancements.

2. **Behavioral Cloning Integration:** To improve the initial performance of the agents, we will integrate behavioral cloning. This technique involves pre-training the agents on a dataset of expert trajectories, enabling them to imitate expert behavior before fully engaging in reinforcement learning. By doing so, the agents start with a better understanding of the environment, which can lead to faster convergence and improved performance in the early stages of training.

3. **Curriculum Learning Integration:** Following the baseline and behavioral cloning stages, we will incorporate curriculum learning to progressively increase the difficulty of the environment. The agent will first be trained on simpler tasks, such as basic movements or strategies, and gradually exposed to more complex scenarios, like coordinated team play and advanced tactics. This progressive learning approach is designed to help the agent build foundational skills before tackling the most challenging aspects of the game, ultimately leading to more robust and adaptable behavior.

4. **Parameter Tuning:** The final stage involves extensive experimentation with various hyperparameters to optimize the agent's performance. This includes adjusting the learning rate, batch size, exploration-exploitation strategies, and other relevant parameters. The goal is to find the optimal configuration that maximizes the agent's ability to learn and perform in the Soccer Twos environment. Careful parameter tuning is essential, as it can significantly impact the efficiency and effectiveness of the training process.

4 Timeline:

- **Initial Proposal:** 01/09/2024
- **Mid-Term Evaluation:** 01/10/2024 (Tentative)
- **End-Term Evaluation:** 15/11/2024 (Tentative)

5 Simulator/Environment:

5.1 SoccerTwos Environment:

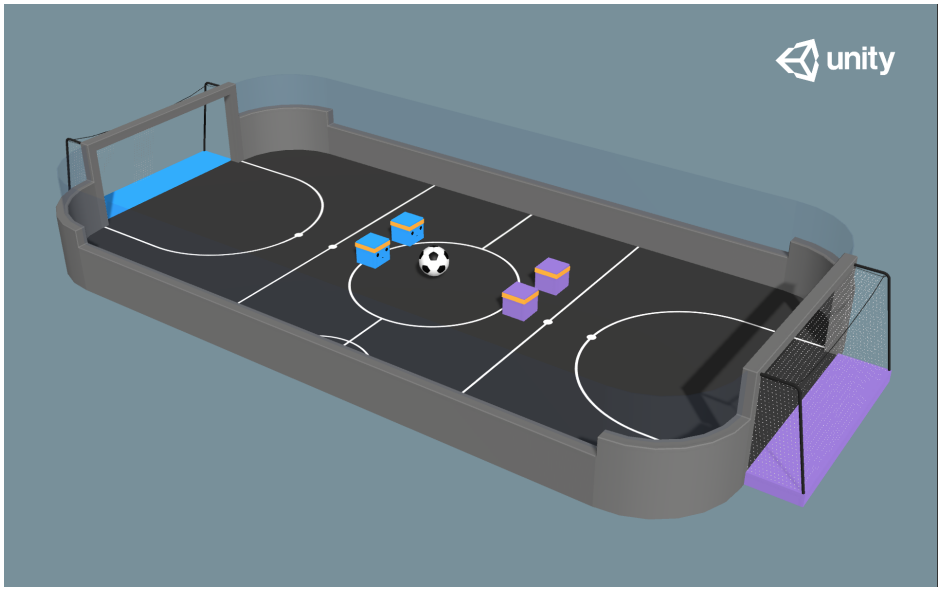

Figure 1: SoccerTwos Environment

- **Set-up:** Environment where four agents compete in a 2 vs 2 toy soccer game.
- **Goal:** Get the ball into the opponent's goal while preventing the ball from entering its own goal.
- **Agents:** The environment contains two different Multi-Agent Groups with two agents in each. Parameters: SoccerTwos.
- **Agent Reward Function (dependent):**
  - (1 - accumulated time penalty) When the ball enters the opponent's goal accumulated time penalty is incremented by (1 / MaxStep) every fixed update and is reset to 0 at the beginning of an episode.
  - -1 When the ball enters the team's goal.
- **Behavior Parameters:**
  - **Vector Observation space:** 336 corresponding to 11 ray-casts forward distributed over 120 degrees and 3 ray-casts backward distributed over 90 degrees each detecting 6 possible object types, along with the object's distance. The forward ray-casts contribute 264 state dimensions and backward 72 state dimensions over three observation stacks.
  - **Actions:** 3 discrete branched actions corresponding to forward, backward, sideways movement, as well as rotation.
  - **Visual Observations:** None
- **Float Properties:** Two
  - **ball_scale:** Specifies the scale of the ball in the 3 dimensions (equal across the three dimensions)
    - *Default:* 7.5
    - *Recommended minimum:* 4
    - *Recommended maximum:* 10
  - **gravity:** Magnitude of the gravity
    - *Default:* 9.81
    - *Recommended minimum:* 6
    - *Recommended maximum:* 20

5.2 Dungeon Escape

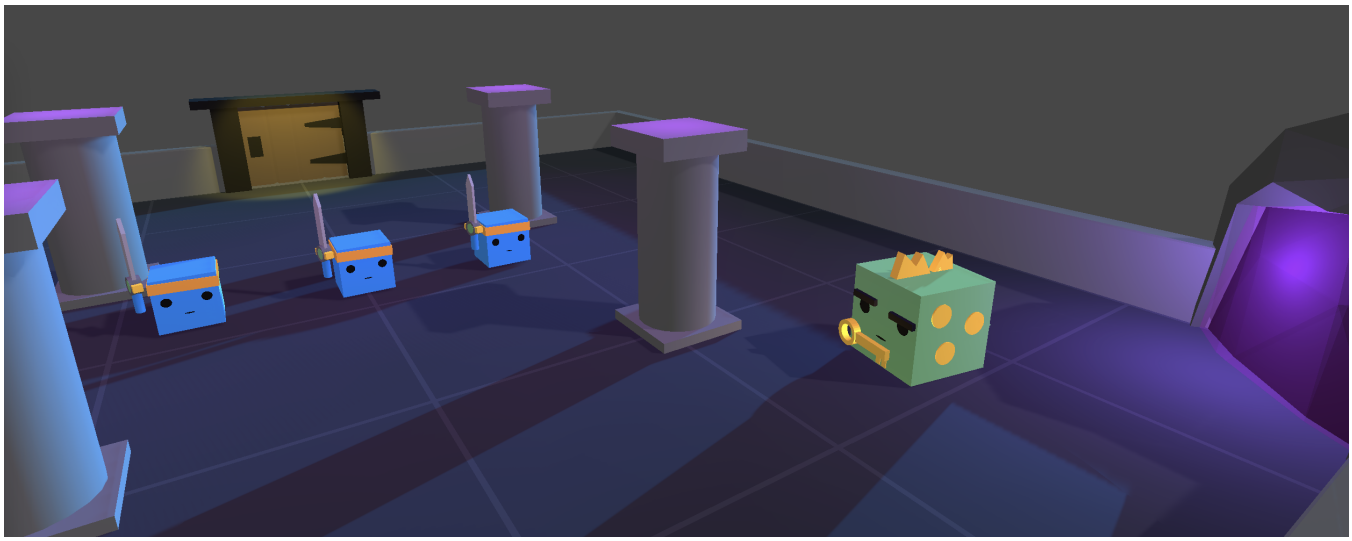

Figure 2: Dungeon Escape

- **Set-up:** Agents are trapped in a dungeon with a dragon, and must work together to escape. To retrieve the key, one of the agents must find and slay the dragon, sacrificing itself to do so. The dragon will drop a key for the others to use. The other agents can then pick up this key and unlock the dungeon door. If the agents take too long, the dragon will escape through a portal and the environment resets.
- **Goal:** Unlock the dungeon door and leave.
- **Agents:** The environment contains three Agents in a Multi Agent Group and one Dragon, which moves in a predetermined pattern.
- **Agent Reward Function:**
  - +1 group reward if any agent successfully unlocks the door and leaves the dungeon.
- **Behavior Parameters:**
  - **Observation space:** A Ray Perception Sensor with separate tags for the walls, other agents, the door, key, the dragon, and the dragon's portal. A single Vector Observation which indicates whether the agent is holding a key.
  - **Actions:** 1 discrete action branch with 7 actions, corresponding to turn clockwise and counterclockwise, move along four different face directions, or do nothing.
- **Float Properties:** None
- **Benchmark Mean Reward:** 1.0 (Group Reward)

References

[1] A. Cohen, E. Teng, V.-P. Berges, R.-P. Dong, H. Henry, M. Mattar, A. Zook, and S. Ganguly, "On the use and misuse of absorbing states in multi-agent reinforcement learning," *RL in Games Workshop AAAI 2022*, 2022.

[2] A. Juliani, V.-P. Berges, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar, and D. Lange, "Unity: A general platform for intelligent agents," *arXiv preprint arXiv:1809.02627*, 2020.