PROJECT PROPOSAL

Soheil Shirvani 862465192

Default Project 1

Introduction:

The Unity Machine Learning Agents Toolkit (ML-Agents) is a pioneering open-source initiative that transforms games and simulations into dynamic environments for training intelligent agents. This project aims to harness the power of ML-Agents to develop and test advanced reinforcement learning (RL) algorithms. By setting up various example environments provided by Unity, I will create a robust simulation platform to evaluate the effectiveness of different RL techniques, including Deep Q-Networks (DQN), Double DQN (DDQN), Soft Actor-Critic (SAC), and Deep Deterministic Policy Gradient (DDPG).

Project Objectives and Setup:

The primary objective of this project is to configure and utilize the Unity ML-Agent environments as testbeds for RL algorithms. The setup process involves three critical steps: downloading Unity ML-Agent examples, building executable scenes from these examples, and wrapping these scenes with the gym-unity to provide a gym-like interface for Python-based RL code. This setup will allow for straightforward integration of our RL algorithms with the Unity environments, facilitating direct interaction and iterative learning processes. Although the current gym-unity wrapper supports only single-agent environments, this constraint will not hinder our ability to reach meaningful conclusions in various test scenarios.

Implementation and Expected Outcomes:

Throughout this project, I will implement and refine several RL algorithms to address the challenges posed by the complex and high-dimensional action spaces of the Unity environments. By focusing on both the algorithmic development and the tuning of hyperparameters, I aim to achieve satisfactory performance across as many environments as possible (Hopefully more than 3). Success will be measured by the agents' ability to earn high rewards and their operational effectiveness within the environments, as observed during rendering phases. The final deliverable will be a comprehensive report detailing the environments tested, the algorithms used, their configurations, codes, and the outcomes, including both successes and areas of difficulty (difficult environments and algorithm comparisons).

Initial Starting Point

In the initial phase of our project on Reinforcement Learning for Unity ML-Agent, we will focus on setting up and interacting with a few key example environments such as the "3D Ball", "Walker", and "Pyramids". These environments offer varied challenges, from balancing and locomotion tasks to complex navigation and object manipulation, providing a broad spectrum of scenarios to test our reinforcement learning algorithms. To facilitate interaction between our Python-based code and these Unity environments, we will utilize the gym-unity wrapper. This tool wraps the Unity environment executables as gym-like environments, which standardizes how agents perceive and interact with the Unity environment through a familiar API used widely in the reinforcement learning community. By implementing this setup, our agents will be able to receive observations from the environment and send back actions in a loop, essentially creating a feedback system crucial for training reinforcement learning models effectively. This process forms the foundation of our project, enabling systematic testing and refinement of our algorithms under controlled yet challenging conditions.