A Curriculum Learning Approach to Coordinated Multi-Agent Drone Delivery Systems

EN.525.637.81.FA23 Final Project

Alec Portelli - 05DEC2023

The Problem

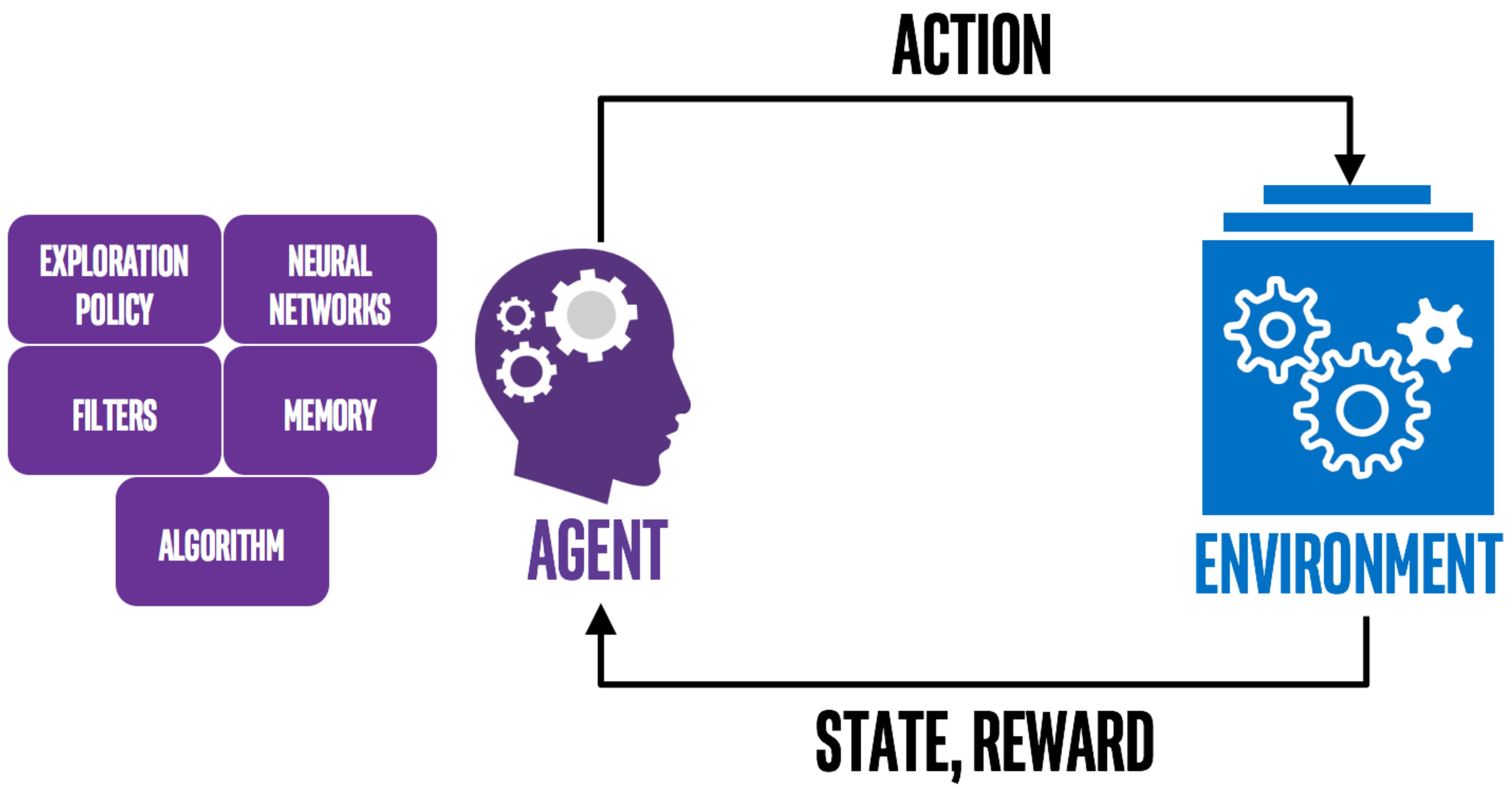
- In a rapidly evolving world marked by e-commerce dominance and global trade, the demand for swift and efficient shipments has reached unprecedented levels.
- Traditional delivery systems struggle to cope with the intricacies of modern logistics, facing challenges in time efficiency and cost-effectiveness.

Why Drones?

- Drones emerge as a transformative solution, offering a paradigm shift in the delivery landscape.
    - **Speed:** Drones can significantly reduce delivery times, meeting the need for rapid shipments.
    - **Agility:** Capable of navigating challenging terrains, drones reach locations inaccessible to conventional vehicles.
    - **Reliability:** Advanced technology ensures consistent drone operation, minimizing the risk of human error.
    - **Environmental Concerns:** Drones contribute to sustainability by reducing carbon emissions compared to traditional delivery methods.

How Does RL Fit Into Everything?

- RL facilitates the creation of a comprehensive autonomy stack for drones.
- Unlike traditional approaches, RL empowers drones to learn and adapt in dynamic, real-world scenarios
- RL enables drones to learn from their experiences, evolving and adapting to diverse and changing conditions.
- This adaptive capability ensures drones can navigate through complex environments with a high degree of autonomy.



Too Tall A Task?

- **Navigating Real-World Challenges:**
  - Training drones in real-world scenarios poses significant challenges due to the inherent complexity of dynamic environments.
- **Unpredictable Environments:**
  - Real-world delivery scenarios are characterized by unpredictability, making it challenging to simulate and train for every possible situation.
- **Dynamic Variables:**
  - Drones encounter diverse variables such as weather changes, traffic, and unexpected obstacles, making real-world training scenarios too intricate.

The Need For Curriculum Learning

- **Navigating the Learning Curve:**
  - Autonomous drone deployment in real-world scenarios demands a thoughtful approach, recognizing the challenges of direct exposure to complex environments.
- **Incremental Skill Development:**
  - Curriculum learning provides a structured approach, allowing drones to incrementally develop skills and capabilities in a controlled manner.
- **Adapting to Complexity:**
  - Real-world scenarios are multifaceted; curriculum learning ensures that drones evolve gradually, adapting to increasing levels of complexity.

Benefits of CL

- **Gradual Exposure:**
  - Drones are exposed to realistic challenges one step at a time, preventing overwhelming scenarios that may hinder learning.
- **Optimizing Learning Efficiency:**
  - By tailoring the learning process, curriculum learning optimizes efficiency, enabling drones to acquire robust skills without being paralyzed by overly complex situations.
- **Realistic Training Environments:**
  - The approach aligns with the need for realistic training environments, ensuring that drones are adequately prepared for the intricacies of real-world deliveries.
- **Enhancing RL Models:**
  - Curriculum learning becomes indispensable in reinforcing RL models, allowing for the effective training of autonomous agents like drones.

Simulation Environment and RL Algorithm Selection

- **Simulation Choice:**
  - The Unity Engine is selected as the simulator for training the drones due to its comprehensive physics engine, incorporating rigid body kinematics, LiDAR, and colliders.
- **ML-Agents Plugin Integration:**
  - Unity's ML-Agents plugin is utilized, providing all essential components for implementing Reinforcement Learning (RL) with custom agents.
- **PPO as the Standard:**
  - The chosen RL algorithm is Proximal Policy Optimization (PPO), recognized as the standard for the Unity ML-Agents package.
- **Success in Multi-Agent RL:**
  - PPO is selected for its demonstrated success in multi-agent reinforcement learning, making it an ideal fit for the project

Methodology - Simple to Complex Environments

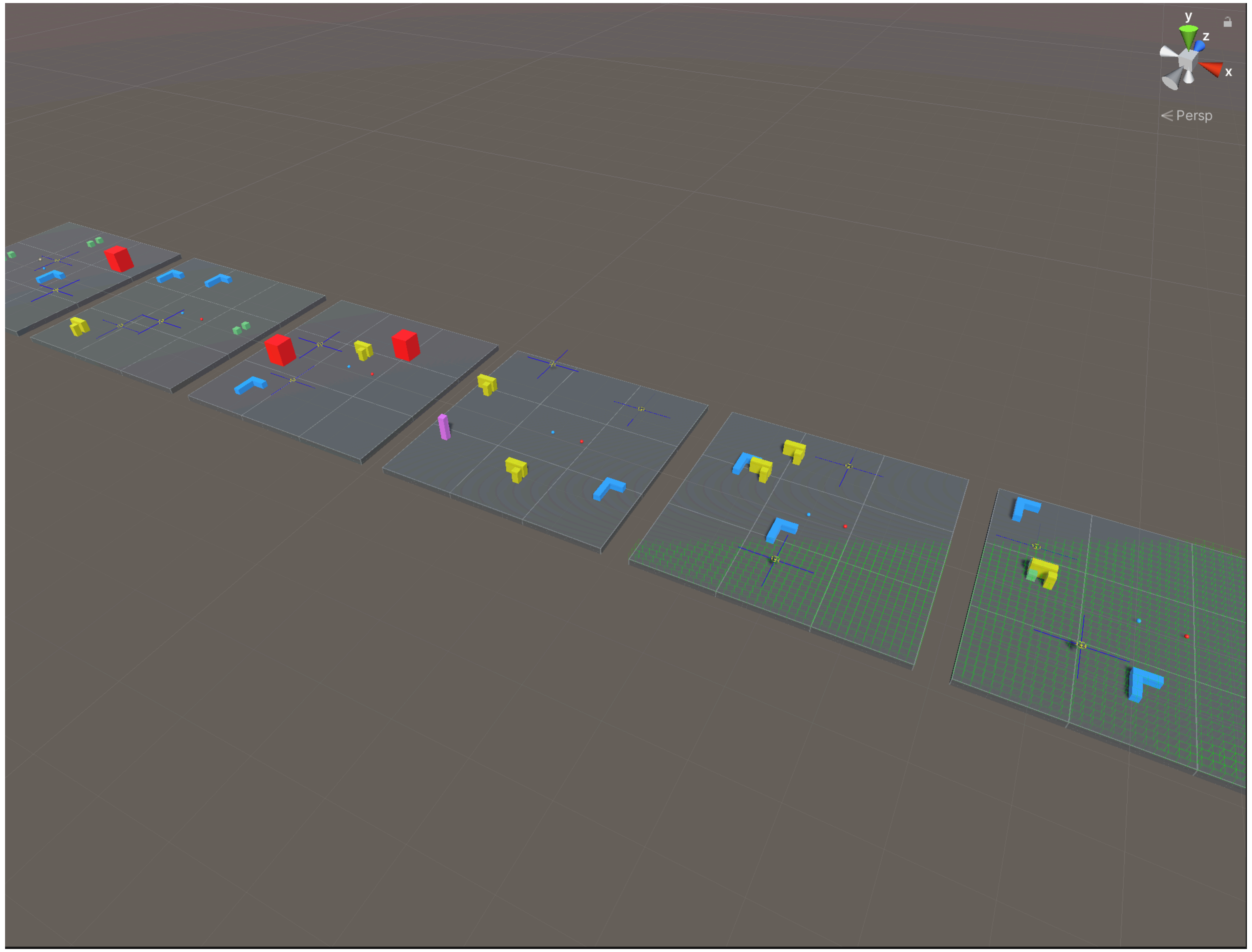
- **Simple 3D Environments:**
  - Unity generates basic 3D environments as the starting point for training the drones.
- **Delivery Goal:**
  - The agent's objective is to deliver the package, regardless of the environmental complexity.
- **Creating Variety:**
  - Domain randomization is employed to introduce variability, generating new training environments every episode.
- **Preventing Overfitting:**
  - This approach ensures a more robust model by preventing overfitting to specific scenarios.
- **Threshold-Based Difficulty Increase:**
  - Curriculum learning is implemented with a YAML file to define thresholds, progressively increasing the difficulty of environments over time.

Monitoring and Parallel Training Environments

Threshold-Based Adjustments:

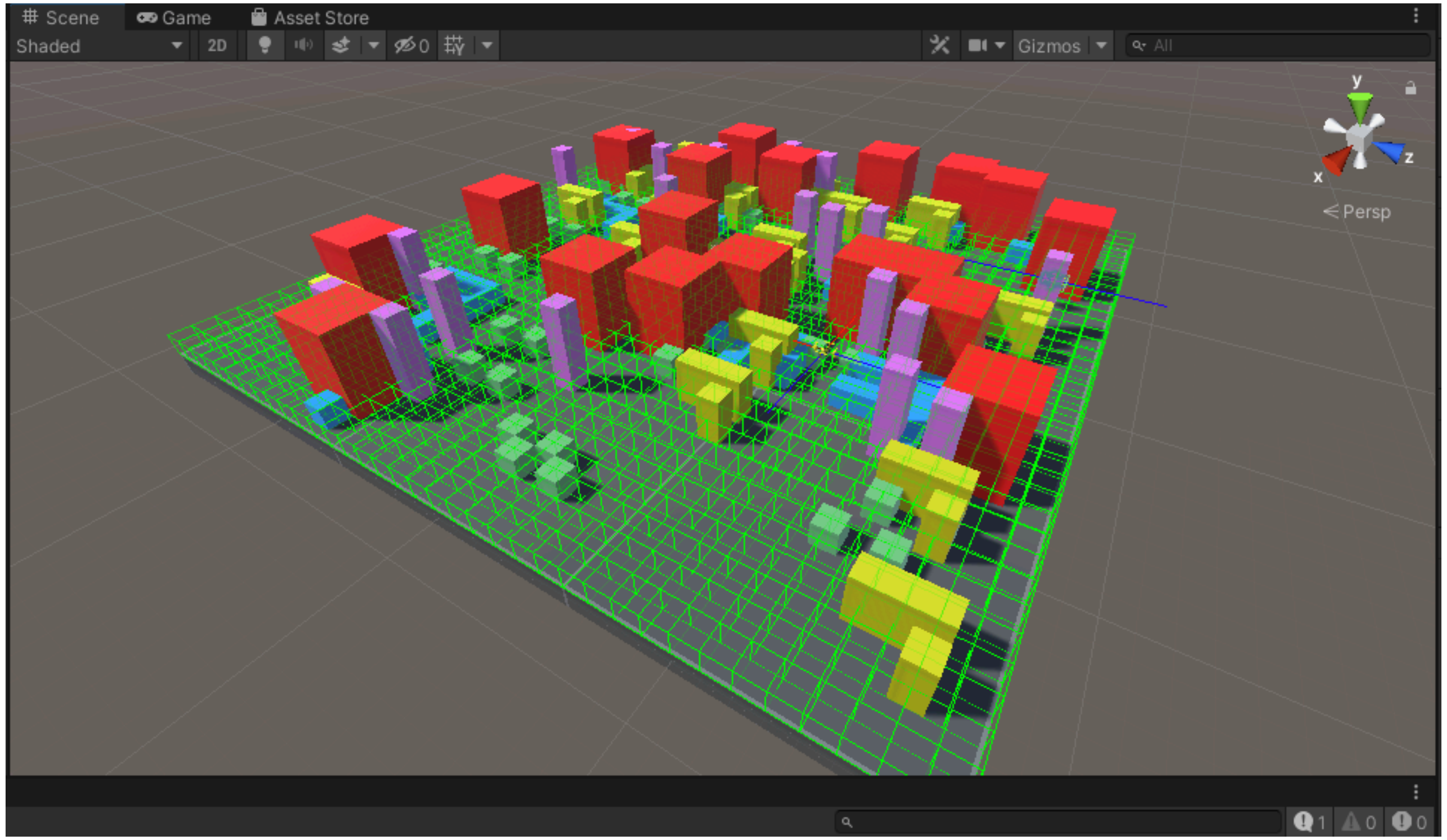
- Difficulty increases are triggered based on the agent's performance, ensuring a challenging but manageable training experience.

- **Unity's Parallel Infrastructure:**
  - Unity's infrastructure for parallel training environments is utilized, allowing the agent to interact with multiple diverse environments simultaneously.
- **Enhancing Model Robustness:**
  - Parallel training enables the agent to encounter various environment configurations concurrently, contributing to the development of a more robust model.
- **Insights through Diversity:**
  - Seeing different environment configurations concurrently provides clearer insights into the meaning of reward totals, enhancing the overall understanding of the training process.



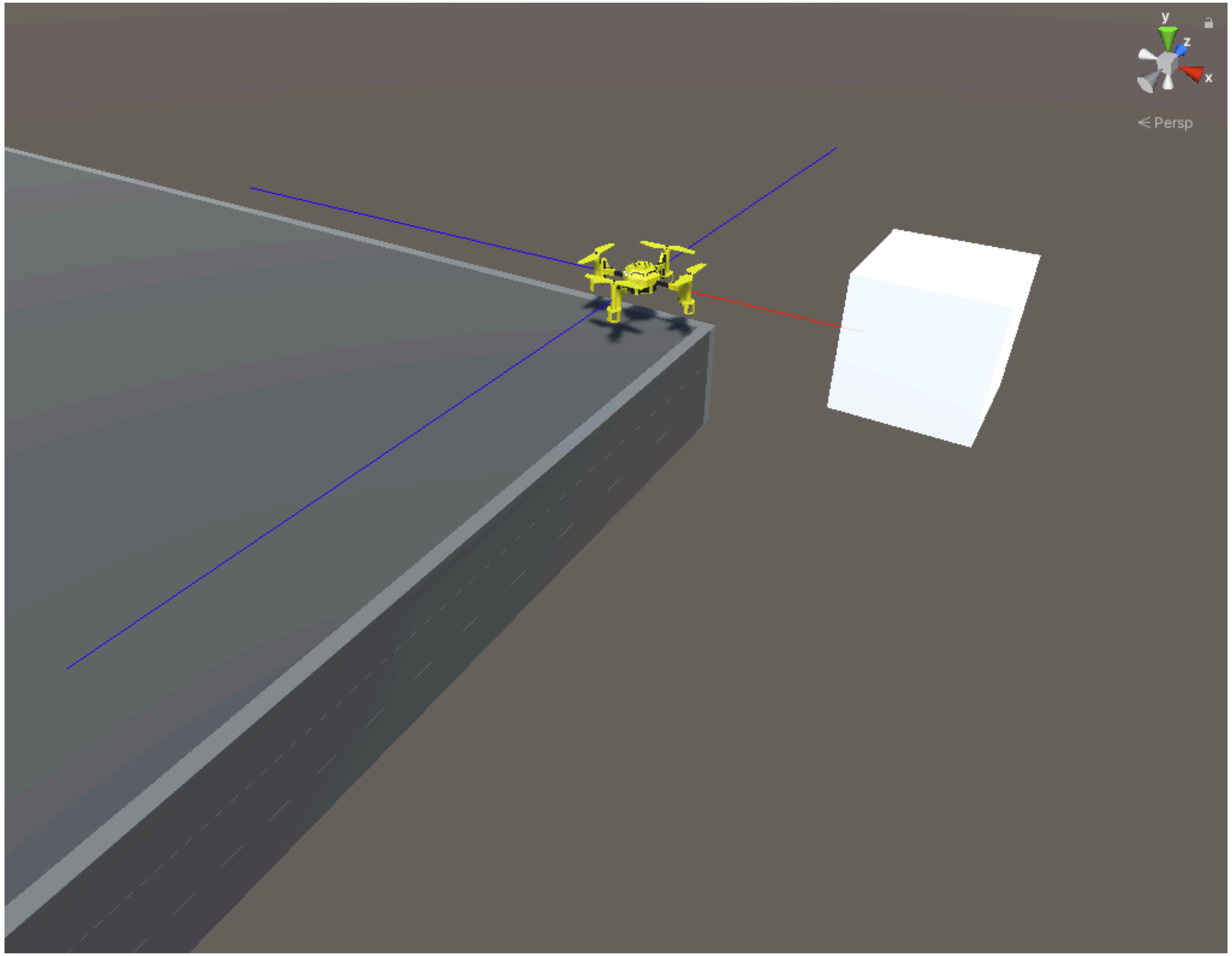
Environment Design:

- 36 by 36 voxel world
- Five building prefabs
- Randomly spawned in
- The amount of buildings determines difficulty
- Drones in random position
- Goals in random position (not in building)
- Buildings have colliders for physics detection



Drone Design

- 3 DoF (up/down, left/right, forward/back)
- LiDAR simulator
    - 4 raycasts going in forward/back and left/right
    - Raycasts fed into the policy



RL Formulation:

- **Goal:**
  - Drone_1 and Drone_2 navigate through world and reach goal destination
- **Episodic:**
  - Episode terminates at going over max steps or failure in mission
  - Failure: Drones crash into one another or crash into building or go out of bounds
- **State (Discrete):**
  - XYZ of drone, XYZ of other drone, 4 raycast values, XYZ of goal
- **Action (Discrete)**
  - Move up, down, left, right, forward, back

Reward Shaping:

$$R_t = \begin{cases} 10 & \text{if the drone reaches the goal} \\ -10 & \text{if the drone goes out of bounds} \\ -100 & \text{if the drone crashes into a building} \\ -\frac{1}{\text{MaxStep}} & \text{for each step taken to encourage quick task completion} \end{cases}$$

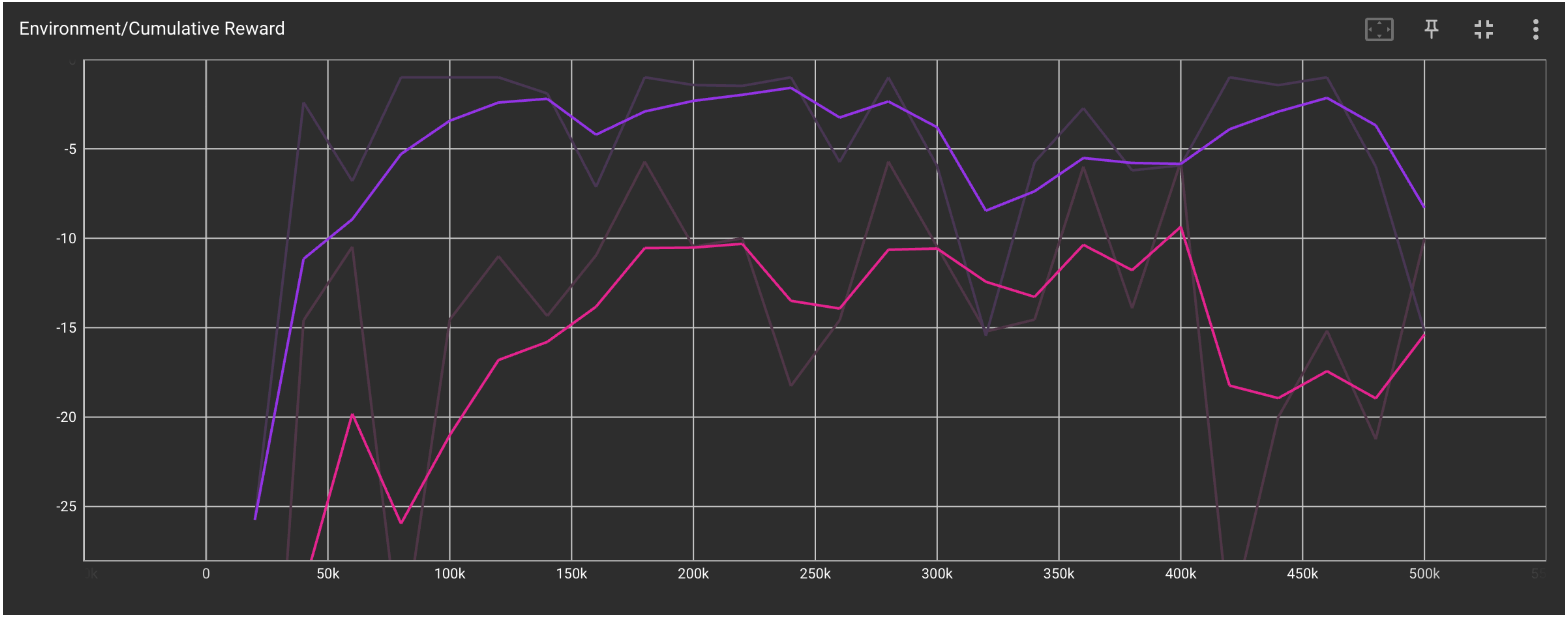
Baseline Experiment:

- **50,000 Steps**
- No curriculum learning
- Hardest difficulty
- Supposed to simulate “going right to the real world”

Added Curriculum Learning:

- **50,000 Steps**
- Curriculum learning
- 5 levels of difficulty
- As the drone learns, the more difficult the environment becomes
- Supposed to ease the drone in

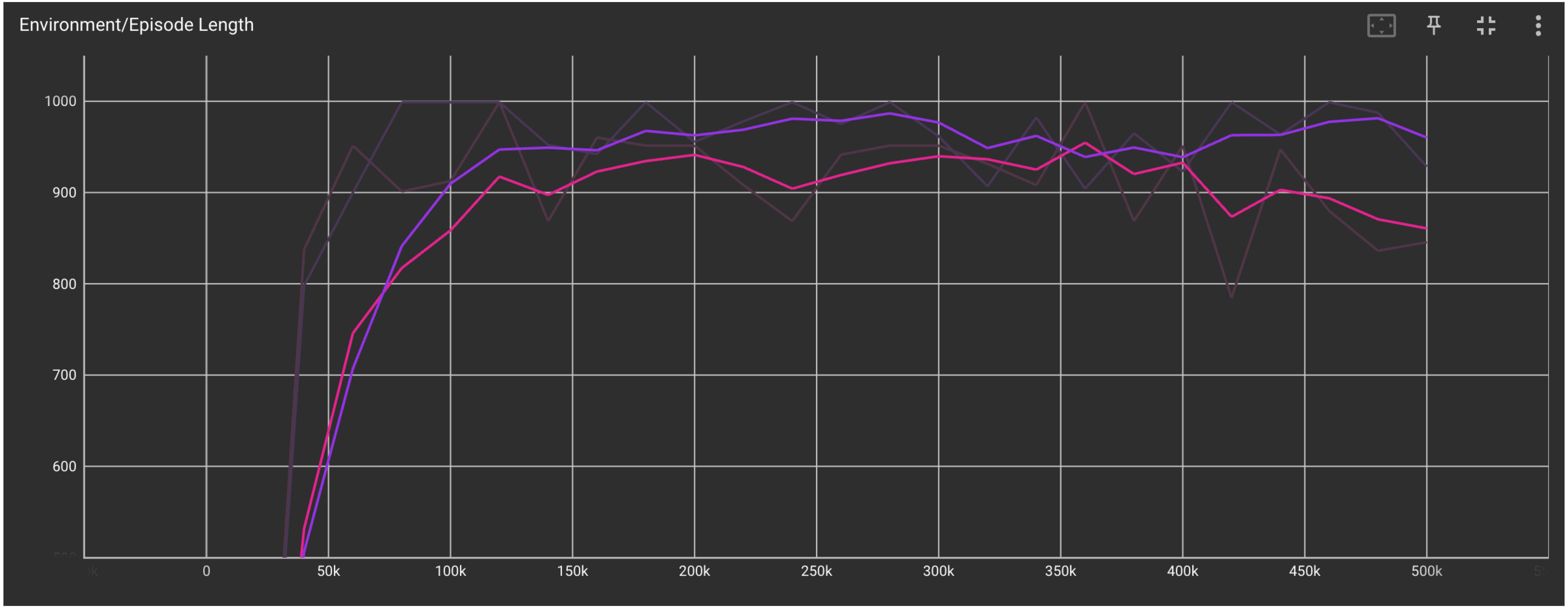
Results



Cumulative Reward Over Time

Purple = Curriculum | Pink = Baseline

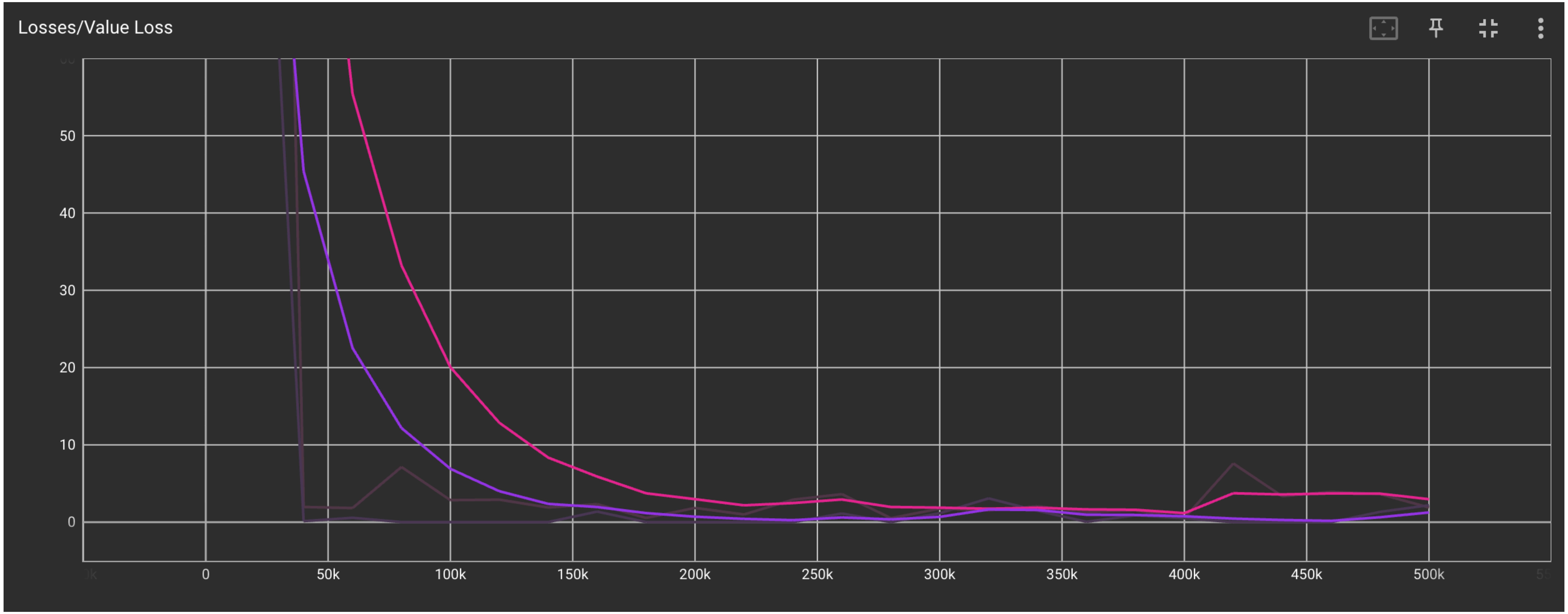
Results



Episode Length Over Time

Purple = Curriculum | Pink = Baseline

Results



Value Loss Over Time

Purple = Curriculum | Pink = Baseline

Explanation

Curriculum

- Curriculum displayed a higher overall score than the baseline
- However, curriculum plateaued, meaning the reward stayed consistent after an initial jump and did not improve
- Got “stuck” meaning the RL algorithm started seeing problems it could not solve
- Saw a wider variety of environments

No Curriculum

- The baseline fluctuated but did not plateau
- More inconsistent
- Lower overall reward score
- Similar episode length converge time
- A very difficult task for a RL algorithm to converge on without prior training

Future Directions

- **Optimizing Learning Progression:**
  - Fine-tune curriculum parameters, including threshold values, to enhance the adaptability of drones in varying environments.
  - Aim to strike a balance between gradual skill development and the complexity of real-world delivery scenarios.
- **Exploration of Hybrid Approaches:**
  - Leveraging the Best of Both Worlds:
  - Investigate hybrid models that integrate curriculum learning with non-curriculum methods.
  - Identify synergies to capitalize on the strengths of each approach, potentially addressing limitations observed in isolated implementations.
- **Scaling to Real-World Environments:**
  - Transitioning from Simulations to Reality:
  - Develop a strategic plan for scaling successful models from simulated environments to real-world drone delivery scenarios.
  - Evaluate the generalization of learned behaviors, considering the challenges and nuances of real-world dynamics, obstacles, and environmental conditions.

Conclusion

- **Curriculum learning breaks the problem down**
  - Curriculum learning gives insight as to how good algorithms are and where they struggle
  - Gives insight on to how close to a real world environment ready they are
  - Less training time in the real environment but help build towards stronger and more robust algorithms
- **The real world is complicated!**
  - Any RL application needs to be run in the real world
  - Taking the real world and making it more manageable is something that will be used to build any AI system
  - A complex task like drone delivery needs to work anywhere, so being able to accurately model where the AI will be is paramount to building systems that are reliable anywhere they are deployed