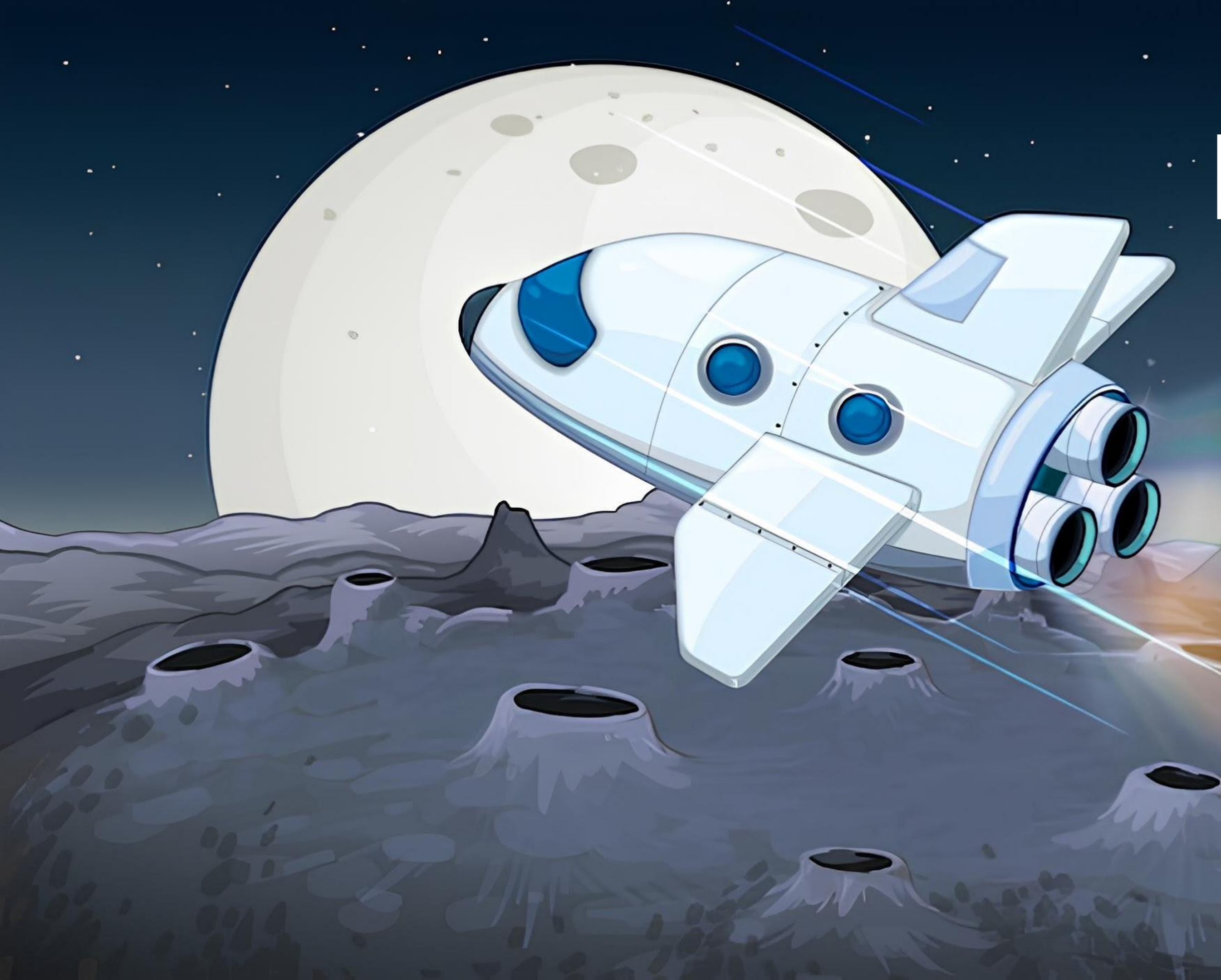


Lunar Lander

FCUP

Introduction to Intelligent Autonomous Systems

1º. Semester 2024/25

Gonçalo Esteves (up202203947)

Nuno Gomes (up202206195)

Pedro Afonseca (up202205394)

Introduction | Project Overview

This project focuses on modifying an **OpenAI Gym environment** to assess the impact of these changes on the performance of a **Reinforcement Learning (RL) agent**.

Therefore, this project includes **two phases**:

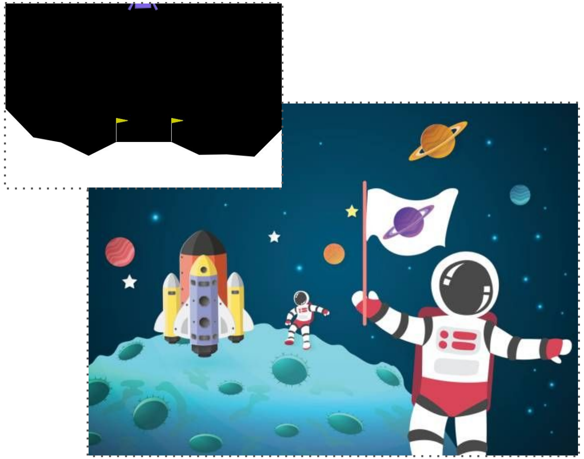
1. **Original Environment Evaluation** → Train and assess the RL agent in the **standard environment** to establish a performance benchmark.
2. **Custom Environment Evaluation** → Introduce **modifications to the environment** and evaluate their influence on the agent's learning overall performance.


Gym Environment
WALL·E

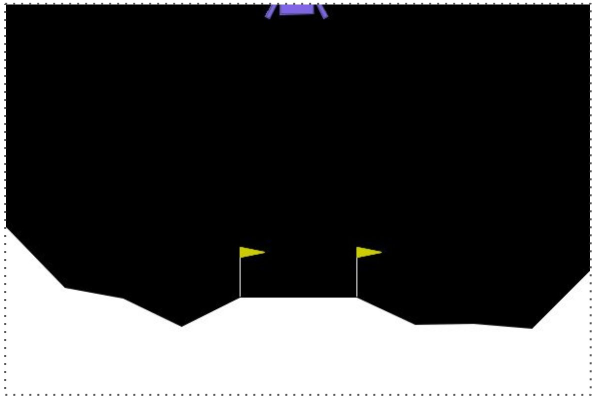
Gym Environment | Selected Environment

After careful consideration upon the available **Open AI gym environments**, we selected:

Lunar Lander Environment



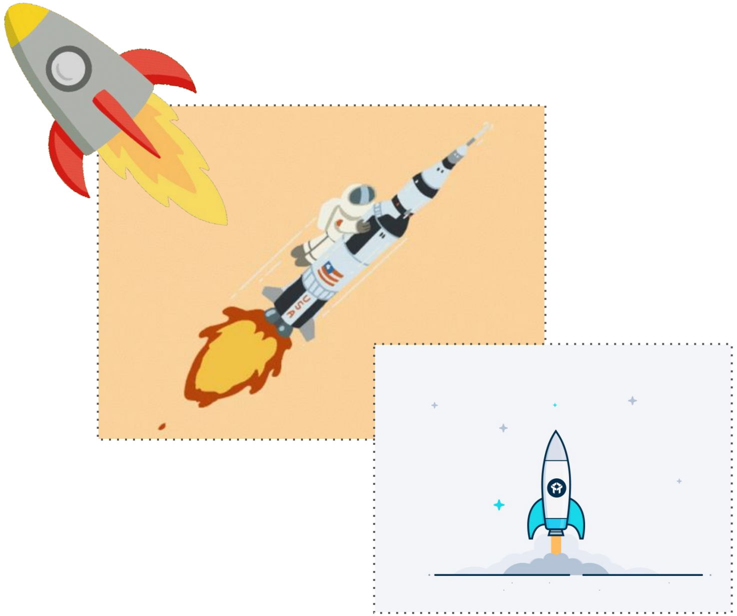
Gym Environment | States and Percepts



The state is represented as an **8-dimensional vector**, which includes the lander's **x and y coordinates**, its **linear velocities along the x and y axes**, its **angle**, **angular velocity**, and two boolean values indicating whether each **leg is in contact** with the ground.

The lander starts at the **top center** of the viewport with a **random initial force** applied to its center of mass. The episode **terminates** if the lander **crashes** (its body makes contact with the moon), **moves outside the viewport** (x-coordinate exceeds 1), or is **no longer awake** - one that is stationary and does not collide with other bodies.

Gym Environment | Actions



The four available **discrete actions** include

→ **No action**

→ **Activate the left orientation engine**

→ **Activate the right orientation engine**

→ **Fire the main engine**

Gym Environment | Rewards

The **reward system** works as follows: **Successfully descending** from the top of the screen to the landing pad and coming to rest earns approximately **100 - 140 points**.

Moving away from the landing pad results in a reward penalty. A **crash** incurs an additional **penalty of -100 points**, while coming to **rest** grants an extra **+100 points**.

Each **landing leg** in contact with the ground provides **+10 points**. Using the **main engine costs -0.3 points per frame**, and using a **side engine costs -0.03 points per frame**. The environment is considered **solved** when a **score of 200 points** is achieved.


Developed Work

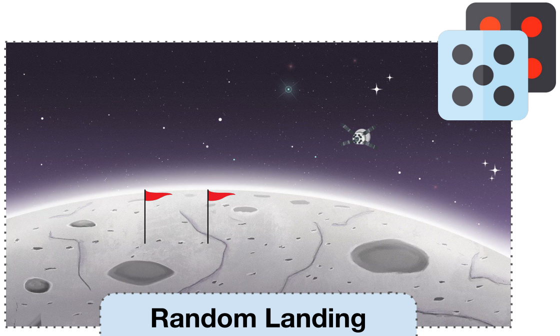
Developed Work | Environment Modifications

For the **custom environment**, we have implemented the following changes:

→ **Randomized Landing Platform Position**

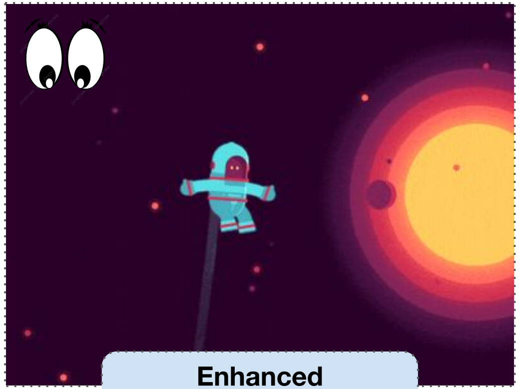
→ **Limited Fuel Tank**

→ **Enhanced Observation Space**



Random Landing Platform Position

Limited Fuel Tank

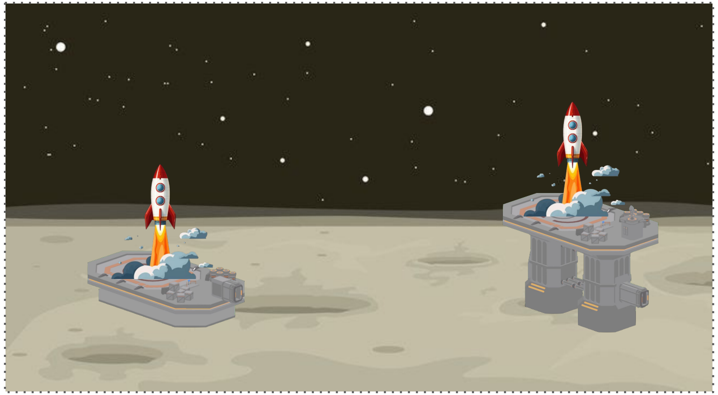


Enhanced Observation Space

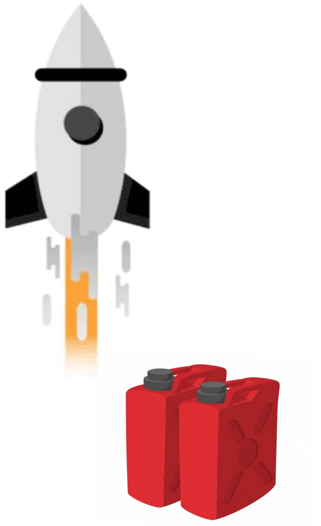
Developed Work | Changes Introduced | Landing Platform

Randomized Landing Platform Position

The landing platform position now **changes randomly** to **add variability** to the spacecraft's approach trajectory.



Developed Work | Changes Introduced | Fuel Tank



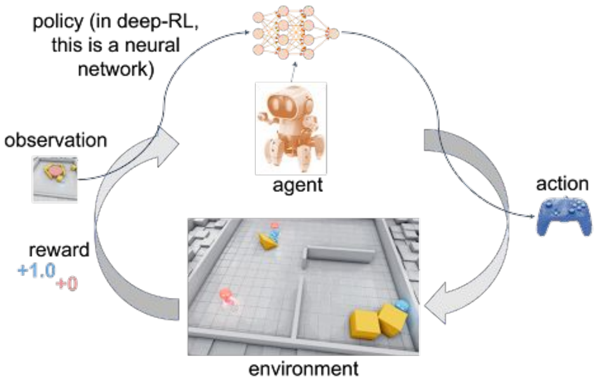
Limited Fuel Tank

The spacecraft is equipped with a **constrained fuel supply**, encouraging **efficient use of propulsion systems**. This not only prevents fuel waste but also adds complexity by factoring in the impact of **fuel weight** on the spacecraft. A larger fuel reserve increases the **overall mass**, thereby intensifying the **gravitational pull** and requiring more precise maneuvering.

Developed Work | Changes Introduced | Observation Space

Enhanced Observation Space

The **observation space** has been augmented to include the **relative position of the spacecraft to the landing platform**, as well as the remaining fuel level.



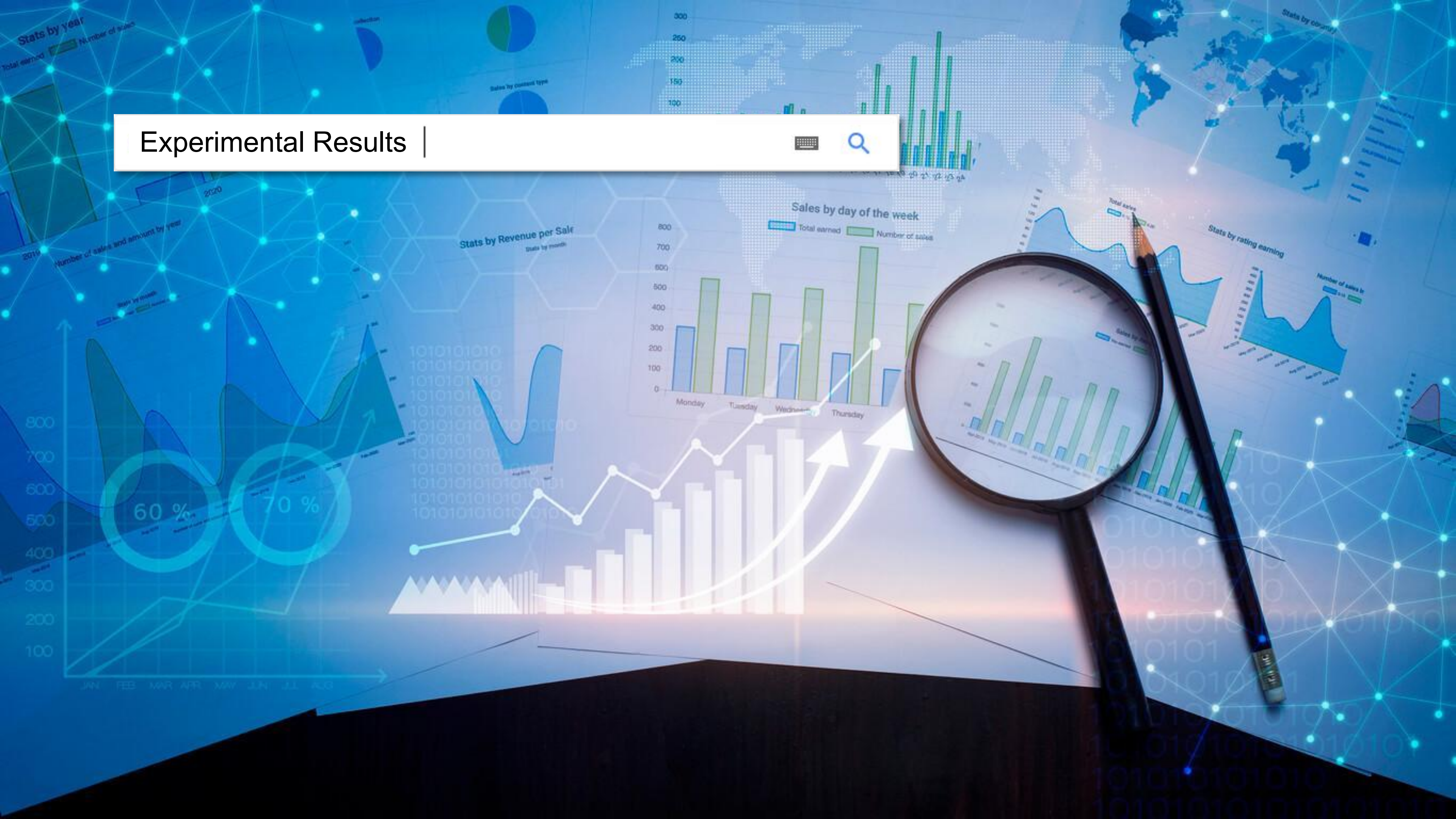
Developed Work | Reinforcement Learning Algorithms

We ended up selecting **2 main algorithms**:

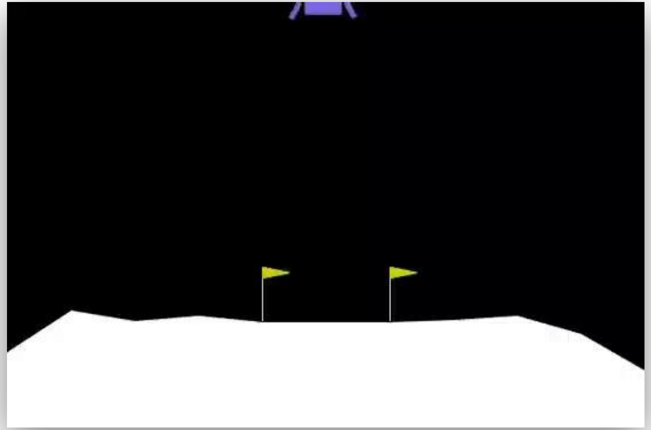
→ **PPO** - focuses on **stability**, **efficiency**, and **exploration**.

→ **DQN** - prioritizes **stability in learning**, **efficient exploration**, and **robust function approximation**.

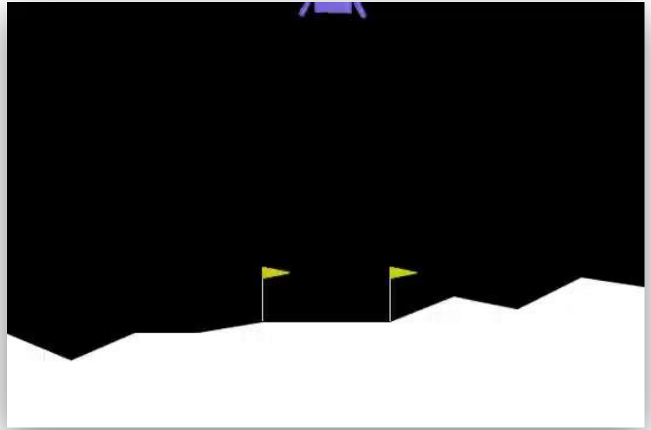
Experimental Results



Experimental Results | Baseline Results



PPO - Settings 1



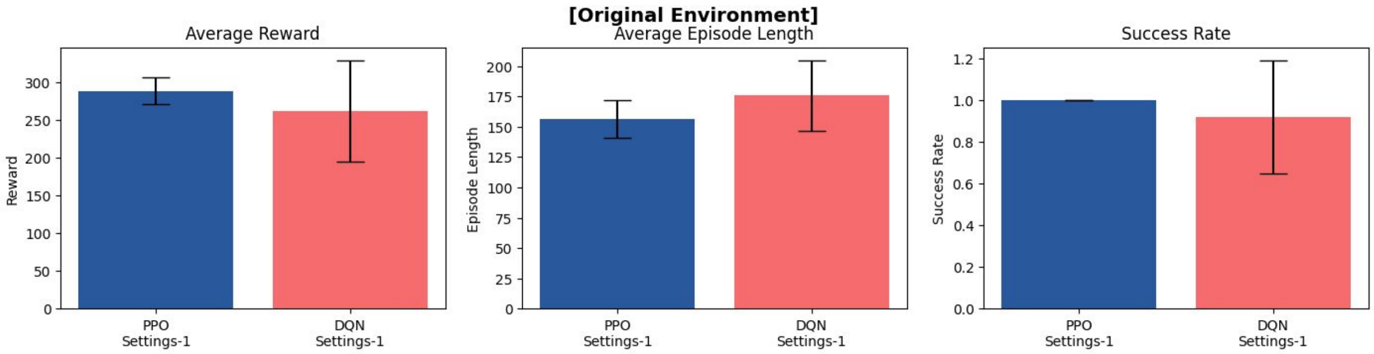
DQN - Settings 1

Experimental Results | Baseline Results

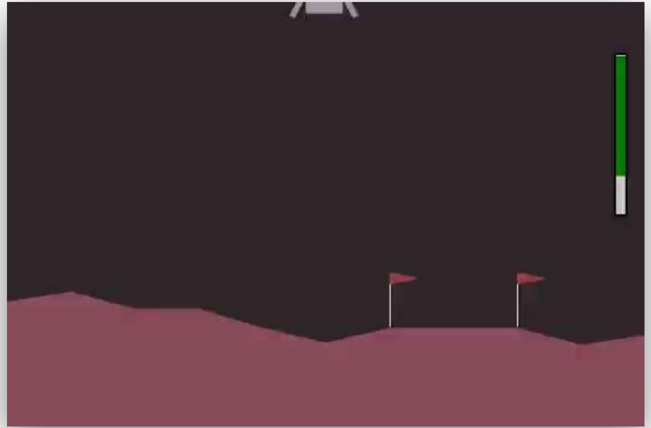
With the original environment, our models performed exceptionally well:

→ **PPO** - Outperformed DQN, achieving a **success rate close to 100%**! Its **average reward** during testing comfortably **exceeded the 200-point** threshold for success.

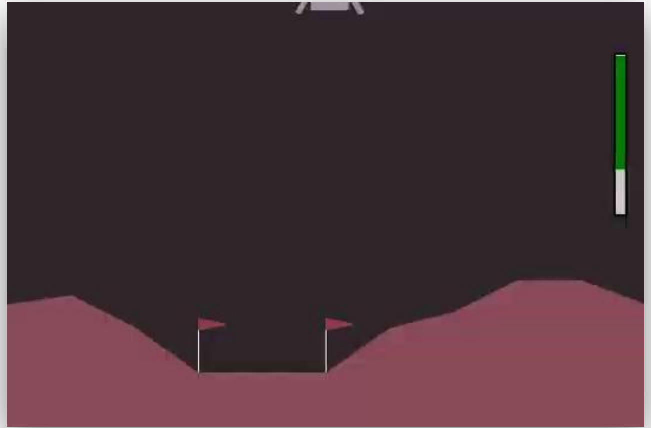
→ **DQN** - While not as strong as PPO, it still delivered **solid results** with an average success rate of around 90%. Its **average reward also surpassed 200 points**, though there were occasional instances of weaker performance.



Experimental Results | Custom Environment Results

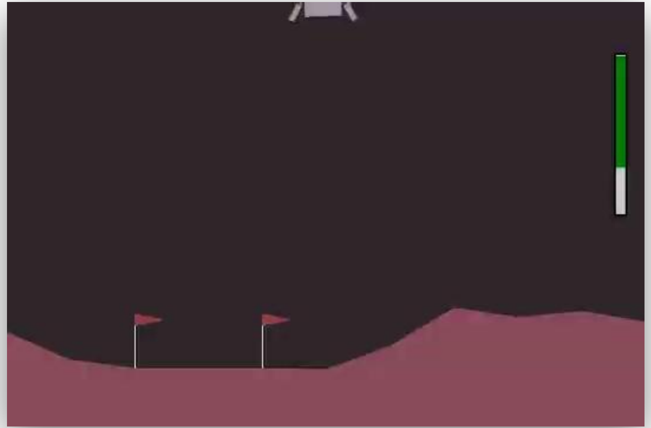


PPO - Settings 1

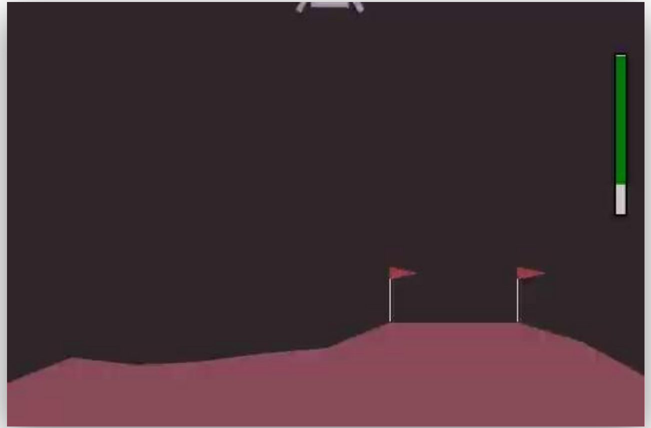


DQN - Settings 1

Experimental Results | Custom Environment Results



PPO - Settings 2



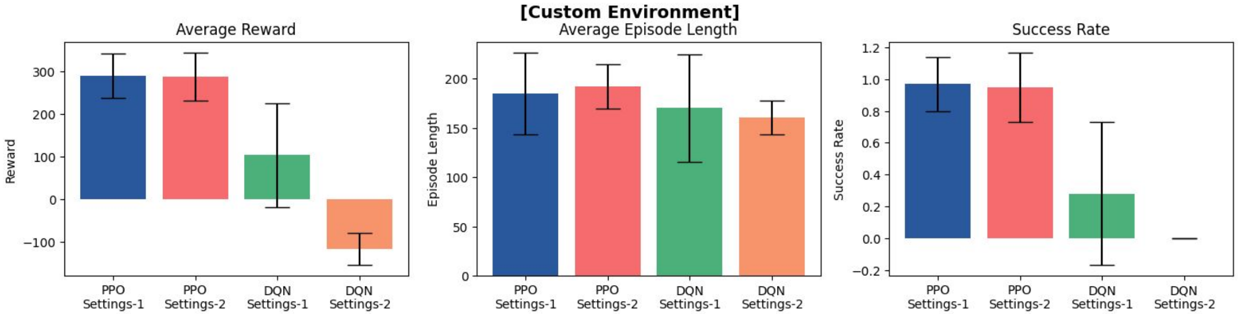
DQN - Settings 2

Experimental Results | Custom Environment Results

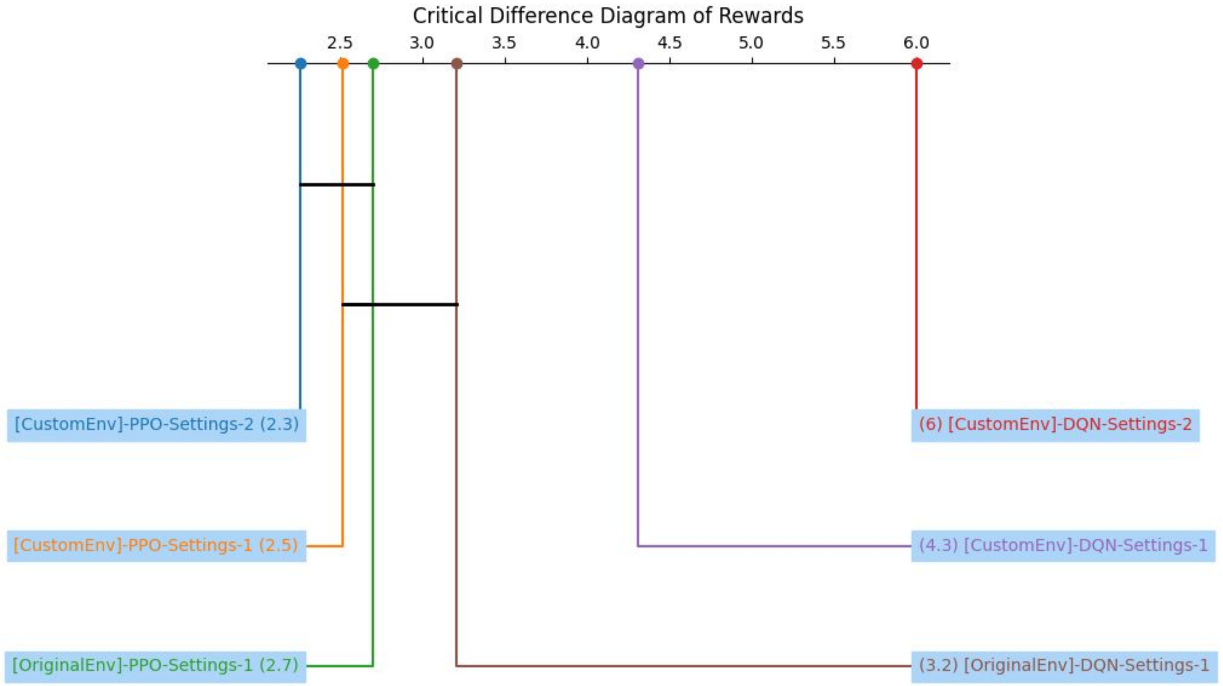
In the custom environment, our results showed significant variation:

→ **PPO** - **Outperformed DQN** by a wide margin, once again achieving a **success rate close to 100%**. The average reward remained consistent across both settings.

→ **DQN** - **Performed poorly** overall. In the first setting, it managed a positive average reward (though below 200) with a 30% success rate. However, in the second setting, its average reward dropped into the negatives, with a success rate of 0%.



Experimental Results | Custom Environment Results




Conclusions

Final Considerations | Conclusions and Future Work

To conclude, the **PPO** algorithm demonstrated **greater stability** across all environments and settings compared to the **DQN** algorithm, as observed over **10 million** timesteps.

For **future work**, we propose **introducing asteroids** to interact with the spacecraft, adding complexity to the environment and **refining the lander's strategies** in **passive-aggressive scenarios**. Additionally, **further hyperparameter** tuning could be performed to enhance the performance of both algorithms.


THANK YOU!