Shuttle Lander

Reinforcement Learning Project

Alessandro Della Siega
Alessandro Minutolo
Stefano Tumino

Objective

Make the shuttle take off from Earth and land on the Moon.

Environment

The system is composed by the Earth and the Moon placed into an image.

Positions are pixels but can be extended with a scale of the real system.

The two entities are fixed in the space:

- No rotation
- No revolution

Representation of the environment

Attributes:

- Position and mass of the Earth
- Position and mass of the Moon
- Position of the flag
- Gravitational field

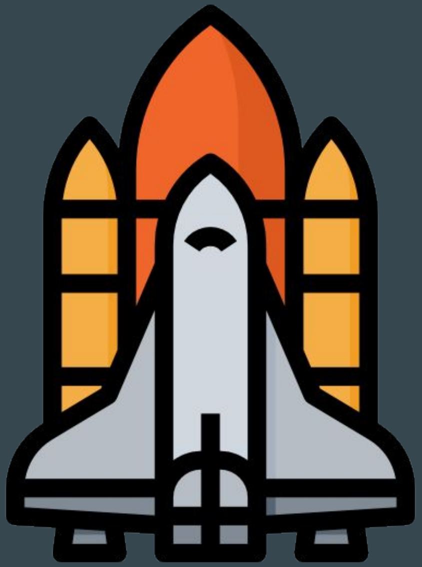
Actions:

- None

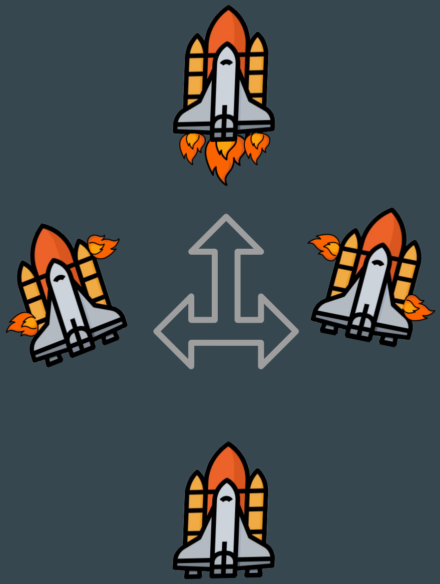
Agent

A space shuttle which starts from the earth and try to reach the moon.

Composed by the main engine and 4 positional engines to rotate.



Representation of the agent



Attributes:

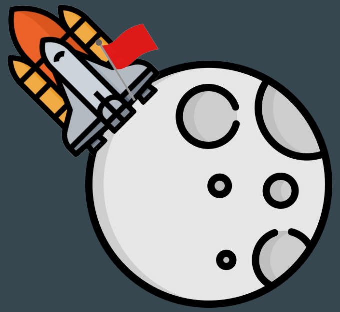
- Height
- Width
- Mass
- Position of the engines
- Fuel
- Speeds and angle

Actions:

- Turn on top-right and bottom-left engines (rotate left)
- Turn on main engine (go straight)
- Turn on top-left and bottom-right engines (rotate right)
- Do nothing

Landing conditions

- The shuttle must land on a given side
- The speed must be low
- The angular speed must be close to 0



Rewards

Positive:

- Small incremental reward based on the distance from the flag
- Small reward if angle similar to the flag one
- Small reward if the speed is small
- Big reward if the shuttle lands

Negative:

- Zero reward if the shuttle has a big angular speed
- Big penalty if the shuttle crashes near the Moon
- Very big penalty if the shuttle crashes near the Earth

Physics

$$\ddot{\boldsymbol{r}} = \frac{\boldsymbol{F}_{\text{net}}}{m}$$

$$\boldsymbol{F}_{\text{net}} = \sum_i \boldsymbol{F}_i$$

Forces, including gravitational field

$$M[\, \boldsymbol{r},\ \boldsymbol{\theta},\ \dot{\boldsymbol{r}},\ \dot{\boldsymbol{\theta}},\ \ddot{\boldsymbol{r}},\ \ddot{\boldsymbol{\theta}}\,](t) = 0$$

$$\boldsymbol{r}(0),\ \boldsymbol{\theta}(0),\ \dot{\boldsymbol{r}}(0),\ \dot{\boldsymbol{\theta}}(0)$$

Equations of motion and initial conditions

$$\ddot{\boldsymbol{\theta}} = \frac{\boldsymbol{\tau}_{\text{net}}}{\boldsymbol{I}}$$

$$\boldsymbol{\tau}_{\text{net}} = \sum_i (\boldsymbol{r}_i - \boldsymbol{r}) \times \boldsymbol{F}_i$$

Torques

Integration

$$y'(t) = f(t, y(t))$$

$$y_{n+1} = y_n + f(t_n,\ y(t_n))\ \Delta t$$

$$y_0 = y(t_0) \quad t_{n+1} = t_n + \Delta t$$

Euler method

Problem Solution

Huge number of states and actions.

Left Engine Power	Main Engine Power	Right Engines Power
[0, 1]	[0, 1]	[0, 1]

We discretized the states showing to the agent only the integer of positions and speeds instead of continuous values

We discretized the actions imposing just to turn on one engine instead of choosing the power for all of them.

Do Nothing	Turn Left	Go Straight	Turn Right

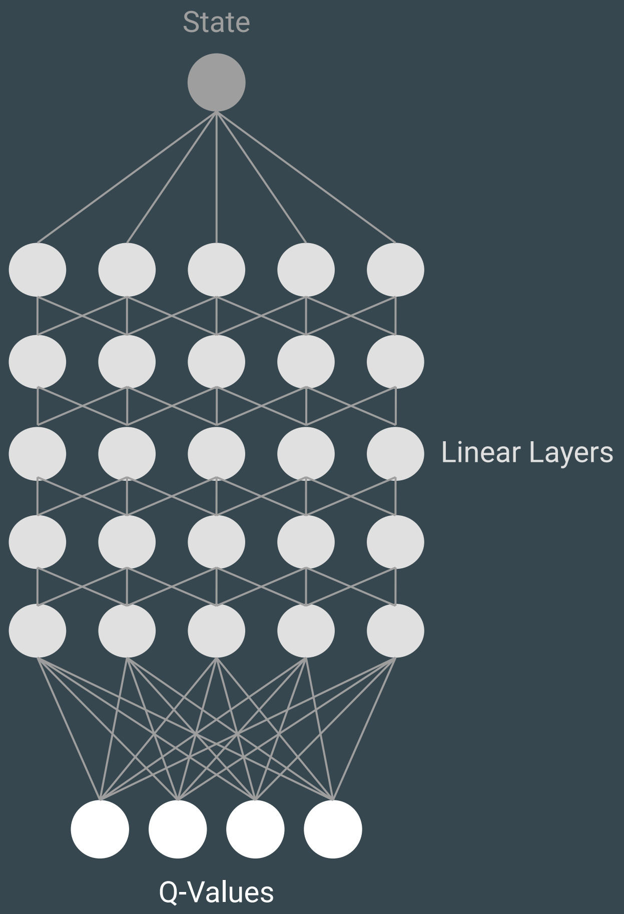
Deep Q-Network

To manage a very large size problem we opted for a function approximation solution.

The easier way to implement this is using a neural network, this allow us to abstract on how to choose the functions which should be derivable.

To train it we exploited the experience replay strategy, keeping a fixed model to make the predictions more stable.

The output will be the expected Q-values for that state.



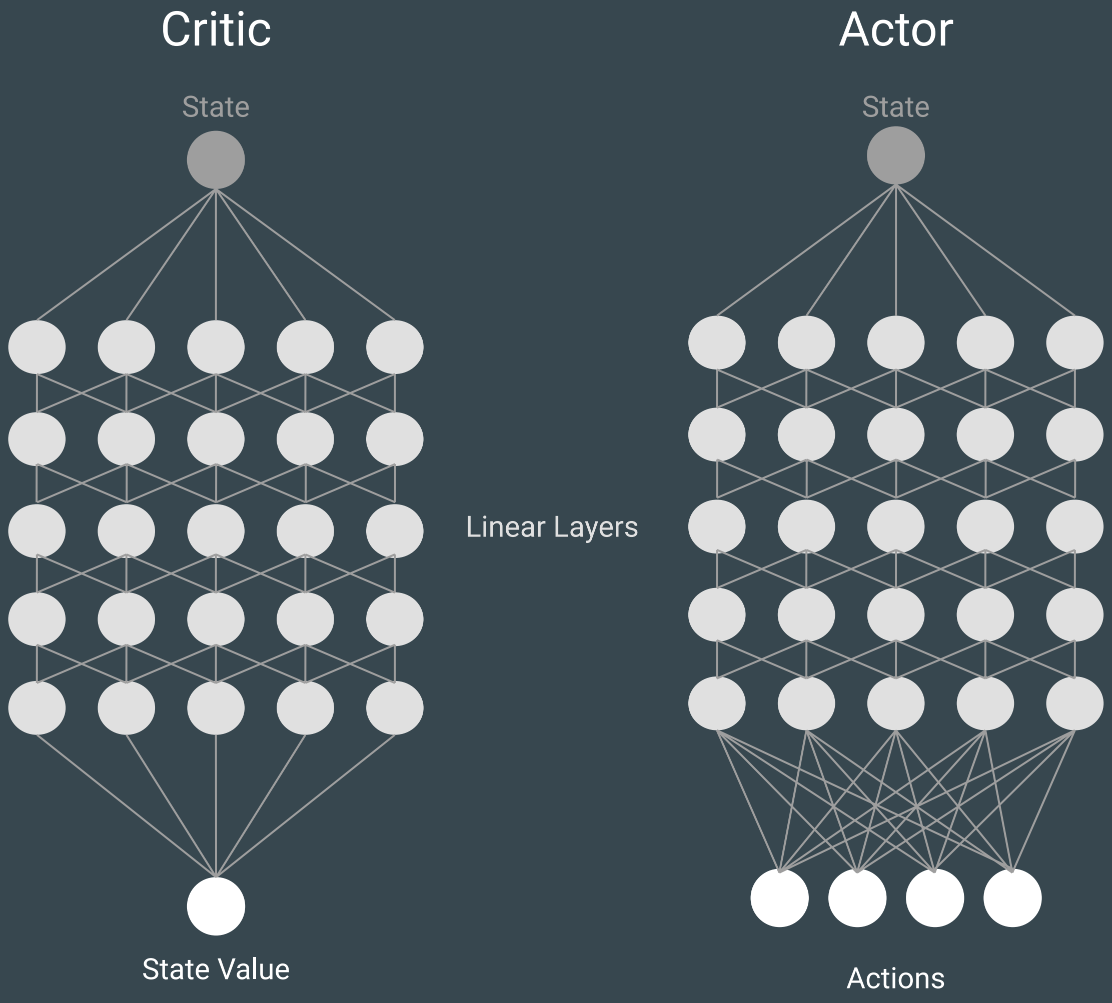
Actor-Critic

We also tried an Actor-Critic approach.

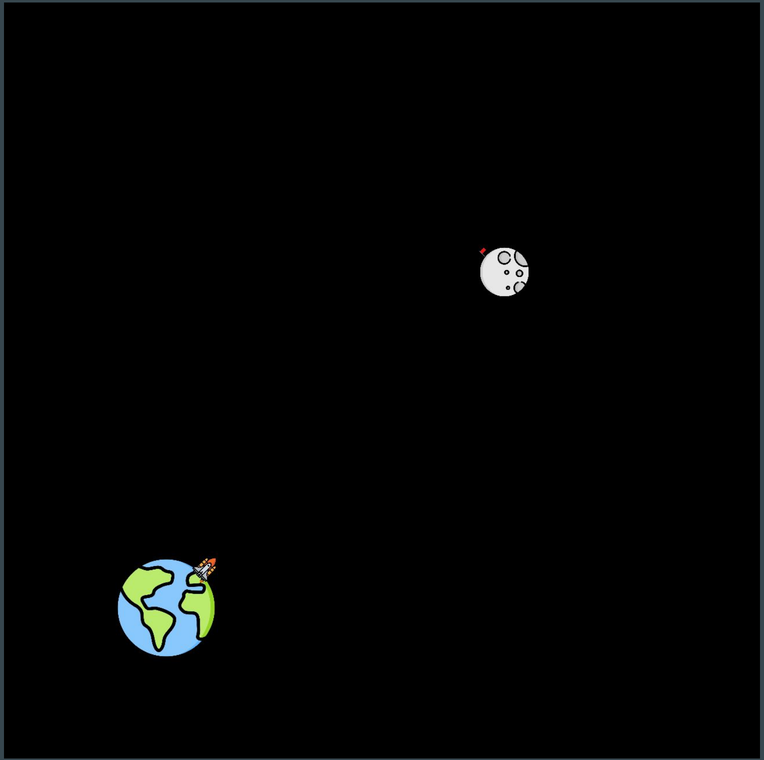
This time we want to approximate the policy directly, and not deriving it from the expected Q-values.

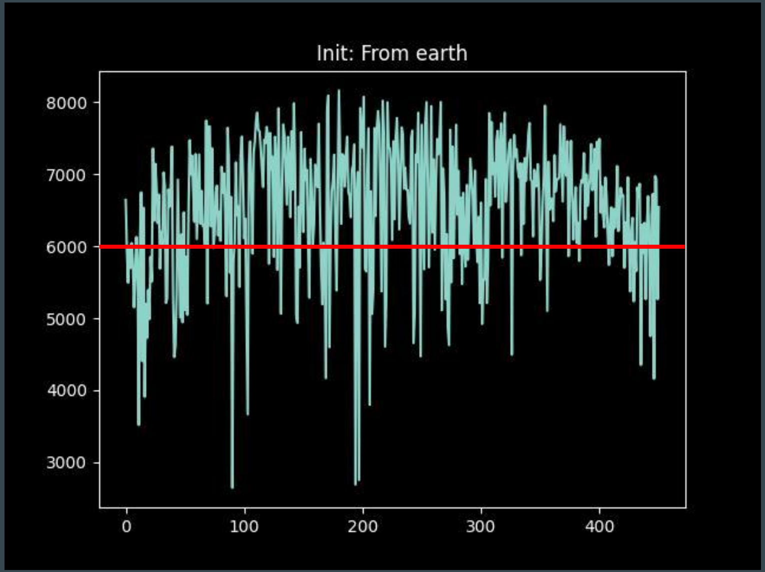
It works with a similar approach to DQN, but this time we approximate both Policy and State Value.

This approach gave us very bad results (return the same action every time)



Results: Shuttle starting from Earth



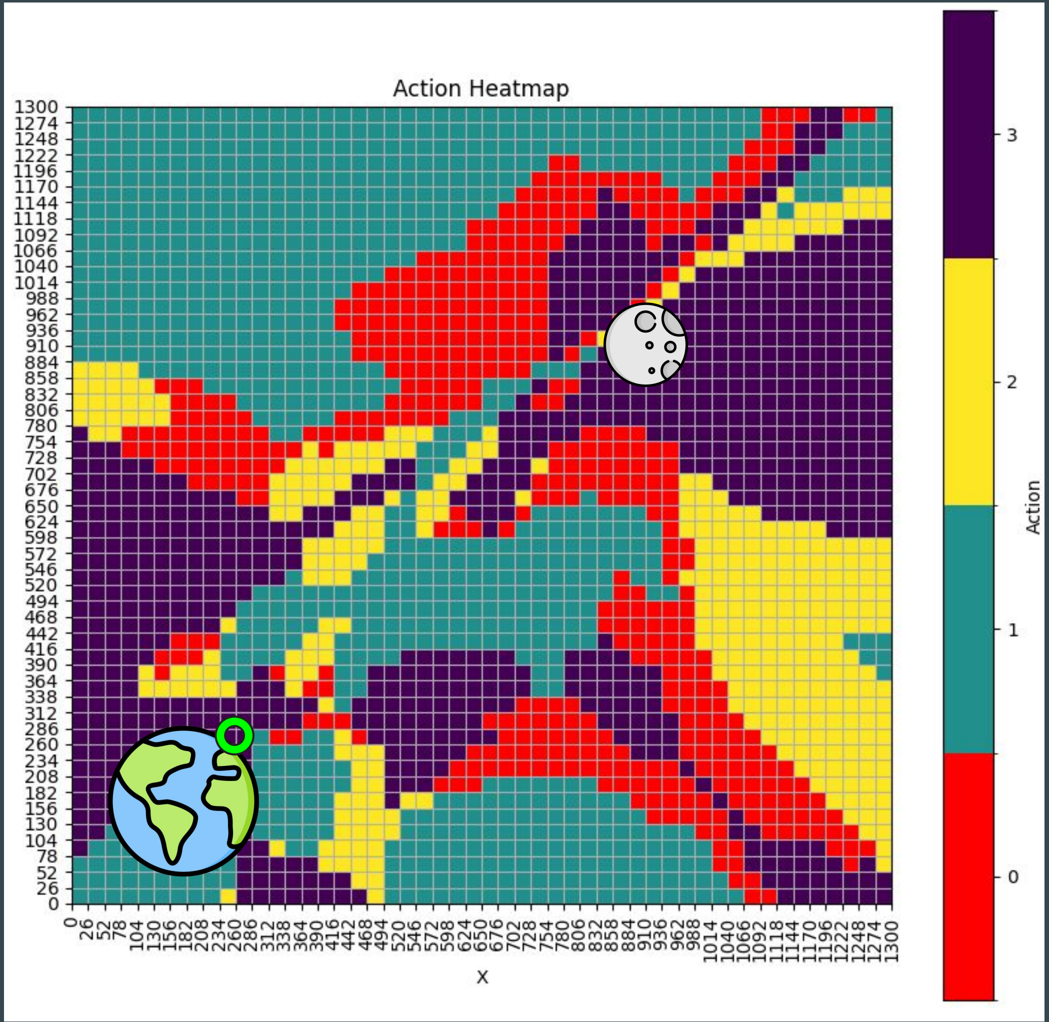


Reward vs Episode

From Earth

Angle: 45°
Speed Module: 0 m/s
Angular Speed: 0 rad/s

Purple: Right
Yellow: Straight
Cyan: Left
Red: Nothing



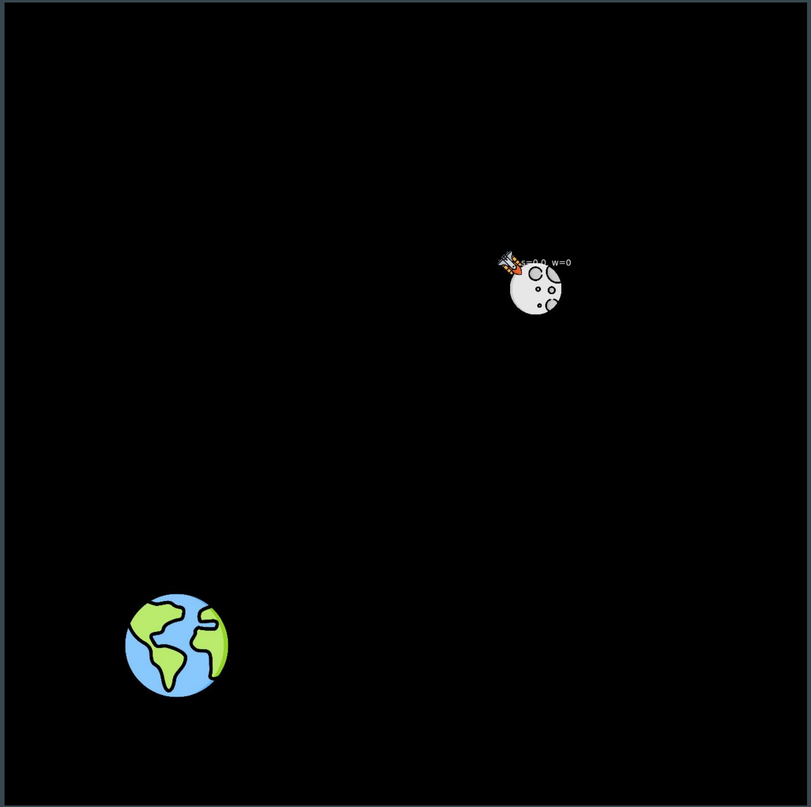
Transfer Learning Technique

Our approach requires too many simulations and resources.

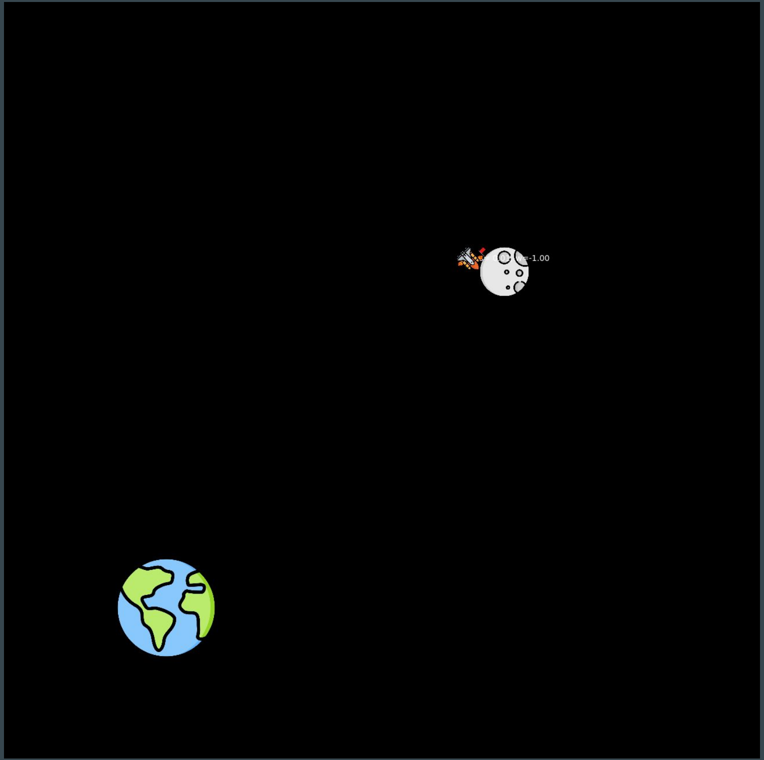
The agent is able to understand where it must go but not to park.

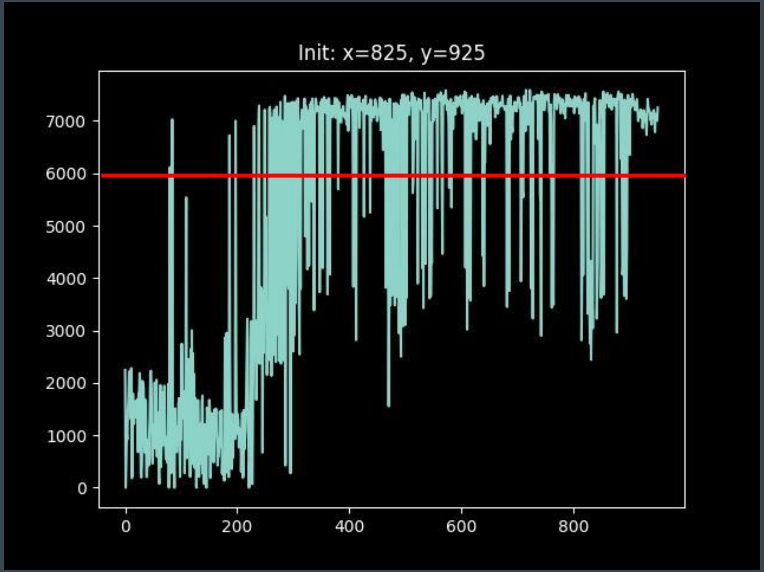
Try to solve simpler problem and propagate these knowledges for harder problem.

It still requires a lot of time but now is able to land.



Results: Shuttle starting near the Moon (x=825, y=925)

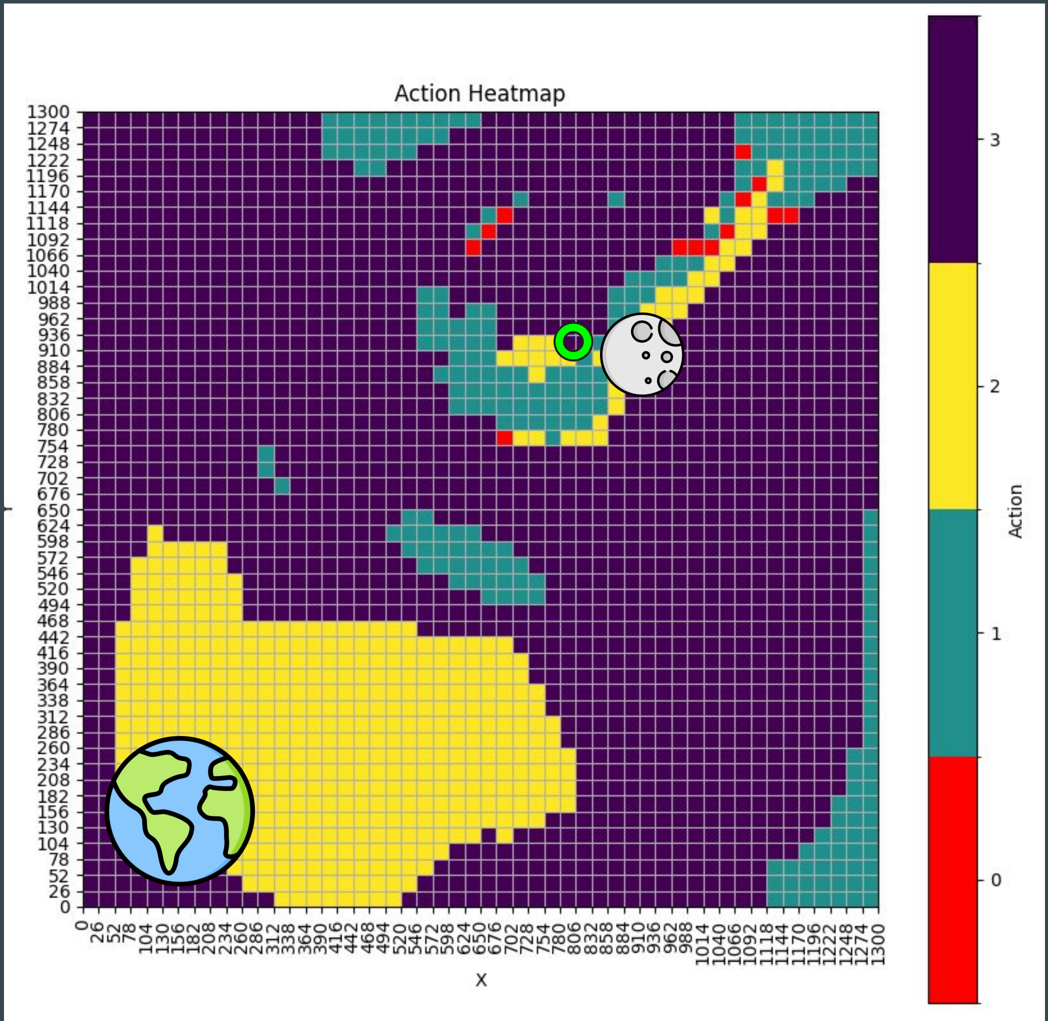




Reward vs Episode

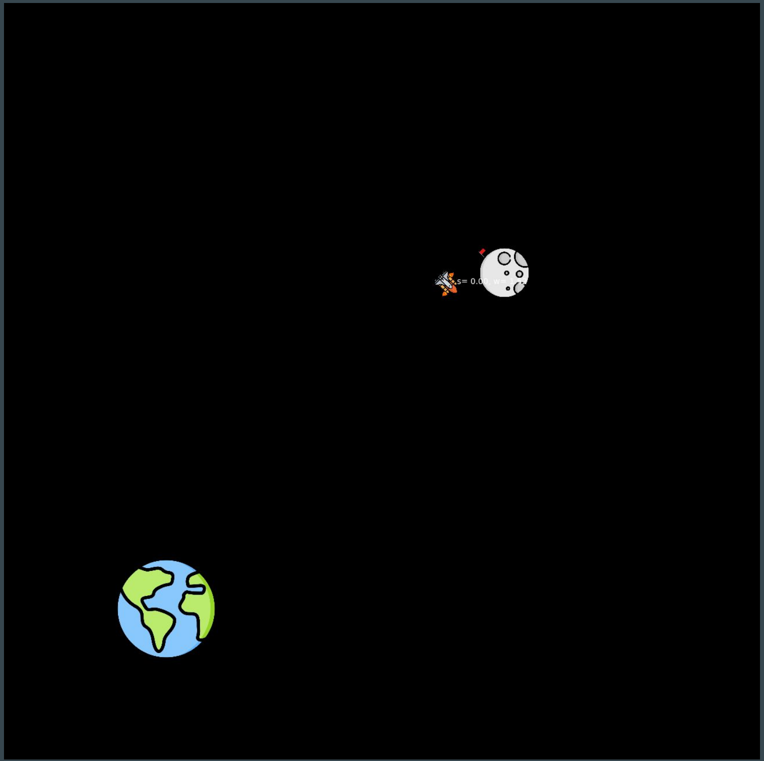
(x=825, y=925)

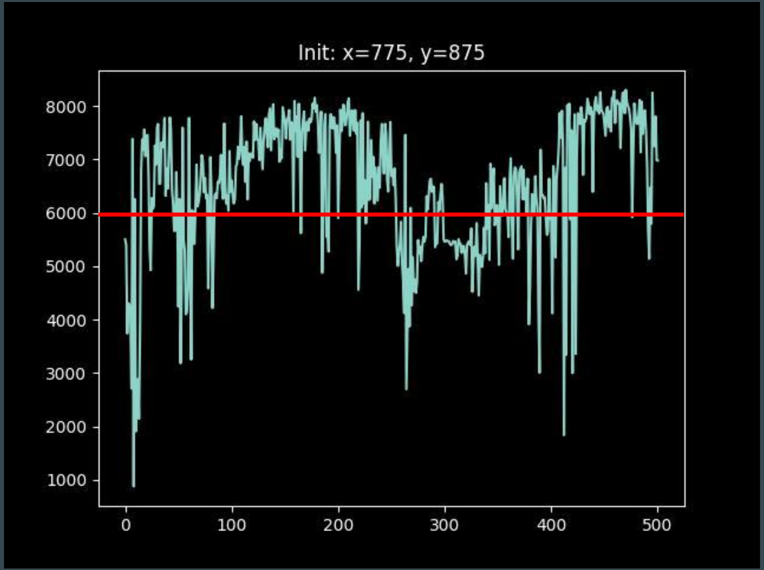
Angle: 45°
Speed Module: 0 m/s
Angular Speed: 0 rad/s



Purple: Right
Yellow: Straight
Cyan: Left
Red: Nothing

Results: Shuttle starting near the Moon (x=775, y=875)



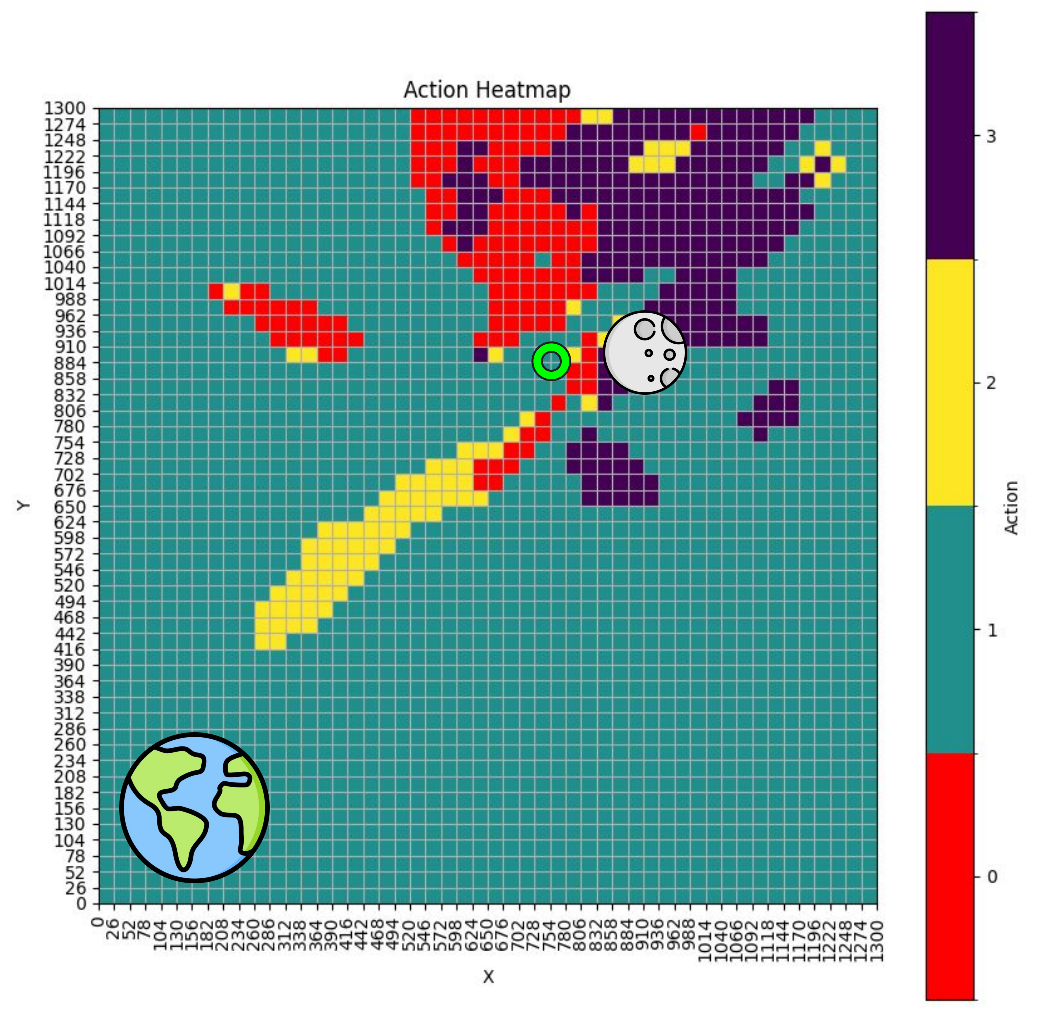


Reward vs Episode

(x=775, y=875)

Angle: 315°
Speed Module: 0 m/s
Angular Speed: 0 rad/s

Purple: Right
Yellow: Straight
Cyan: Left
Red: Nothing



Conclusions

Achievements:

- The problem is solvable using DQN
- The solution is easily interpretable
- The problem is extendable to a more complex scenario

Pending issues:

- The state-action space of the complete problem is huge
- Finding the optimal policy requires computational resources that we do not own

Thank you for your attention!