reinforcement-learning-basic

August 19, 2023

#INTRODUCTION TO REINFORCEMENT LEARNING

Deep RL is a type of Machine Learning where an agent learns how to behave in an environment by performing actions and seeing the results.

In this first unit, you'll learn the foundations of Deep Reinforcement Learning.

Then, you'll train your Deep Reinforcement Learning agent, a lunar lander to land correctly on the Moon using Stable-Baselines3 , a Deep Reinforcement Learning library.

The big picture The idea behind Reinforcement Learning is that an agent (an AI) will learn from the environment by interacting with it (through trial and error) and receiving rewards (negative or positive) as feedback for performing actions.

Learning from interactions with the environment comes from our natural experiences.

For instance, imagine putting your little brother in front of a video game he never played, giving him a controller, and leaving him alone.

By interacting with his environment through trial and error, your little brother understands that he needs to get coins in this environment but avoid the enemies.

Without any supervision, the child will get better and better at playing the game.

That's how humans and animals learn, through interaction. Reinforcement Learning is just a computational approach of learning from actions.

Reinforcement learning is a framework for solving control tasks (also called decision problems) by building agents that learn from the environment by interacting with it through trial and error and receiving rewards (positive or negative) as unique feedback.

Markov Property In papers, you'll see that the RL process is called a Markov Decision Process (MDP).

We'll talk again about the Markov Property in the following units. But if you need to remember something today about it, it's this: the Markov Property implies that our agent needs only the current state to decide what action to take and not the history of all the states and actions they took before.

1. Episodic task In this case, we have a starting point and an ending point (a terminal state). This creates an episode: a list of States, Actions, Rewards, and new States.

For instance, think about Super Mario Bros: an episode begin at the launch of a new Mario Level and ends when you're killed or you reached the end of the level.

2. Continuing tasks These are tasks that continue forever (no terminal state). In this case, the agent must learn how to choose the best actions and simultaneously interact with the environment.

For instance, an agent that does automated stock trading. For this task, there is no starting point and terminal state. The agent keeps running until we decide to stop it.

[1]:
```
%%html
<video controls autoplay><source src="https://huggingface.co/sb3/
↪ppo-LunarLander-v2/resolve/main/replay.mp4" type="video/mp4"></video>
```

```
<IPython.core.display.HTML object>
```

1 Install dependencies and create a virtual screen

[1]:
```
!apt install swig cmake
!pip install -r https://raw.githubusercontent.com/huggingface/deep-rl-class/
↪main/notebooks/unit1/requirements-unit1.txt


#During the notebook, we'll need to generate a replay video. To do so, with␣
↪colab, we need to have a virtual screen to be able to render the environment
!sudo apt-get update
!sudo apt-get install -y python3-opengl
!apt install ffmpeg
!apt install xvfb
!pip3 install pyvirtualdisplay
```

```
Reading package lists… Done
Building dependency tree… Done
Reading state information… Done
cmake is already the newest version (3.22.1-1ubuntu1.22.04.1).
Suggested packages:
  swig-doc swig-examples swig4.0-examples swig4.0-doc
The following NEW packages will be installed:
  swig swig4.0
0 upgraded, 2 newly installed, 0 to remove and 16 not upgraded.
Need to get 1,116 kB of archives.
After this operation, 5,542 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 swig4.0 amd64
4.0.2-1ubuntu1 [1,110 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/universe amd64 swig all
4.0.2-1ubuntu1 [5,632 B]
Fetched 1,116 kB in 1s (829 kB/s)
Selecting previously unselected package swig4.0.
```

```
(Reading database … 120831 files and directories currently installed.)
Preparing to unpack …/swig4.0_4.0.2-1ubuntu1_amd64.deb …
Unpacking swig4.0 (4.0.2-1ubuntu1) …
Selecting previously unselected package swig.
Preparing to unpack …/swig_4.0.2-1ubuntu1_all.deb …
Unpacking swig (4.0.2-1ubuntu1) …
Setting up swig4.0 (4.0.2-1ubuntu1) …
Setting up swig (4.0.2-1ubuntu1) …
Processing triggers for man-db (2.10.2-1) …
Collecting stable-baselines3==2.0.0a5 (from -r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1))
  Downloading stable_baselines3-2.0.0a5-py3-none-any.whl (177 kB)
                             177.5/177.5
kB 3.5 MB/s eta 0:00:00
Collecting swig (from -r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 2))
  Downloading swig-4.1.1-py2.py3-none-manylinux_2_5_x86_64.manylinux1_x86_64.whl
(1.8 MB)
                              1.8/1.8 MB
43.5 MB/s eta 0:00:00
Collecting gymnasium[box2d] (from -r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 3))
  Downloading gymnasium-0.29.0-py3-none-any.whl (953 kB)
                             953.8/953.8 kB
57.8 MB/s eta 0:00:00
Collecting huggingface_sb3 (from -r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 4))
  Downloading huggingface_sb3-2.3-py3-none-any.whl (9.6 kB)
Collecting gymnasium==0.28.1 (from stable-baselines3==2.0.0a5->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1))
  Downloading gymnasium-0.28.1-py3-none-any.whl (925 kB)
                             925.5/925.5 kB
68.3 MB/s eta 0:00:00
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-
packages (from stable-baselines3==2.0.0a5->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (1.23.5)
Requirement already satisfied: torch>=1.11 in /usr/local/lib/python3.10/dist-
packages (from stable-baselines3==2.0.0a5->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (2.0.1+cu118)
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.10/dist-
```

```
packages (from stable-baselines3==2.0.0a5->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (2.2.1)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages
(from stable-baselines3==2.0.0a5->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (1.5.3)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-
packages (from stable-baselines3==2.0.0a5->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (3.7.1)
Collecting jax-jumpy>=1.0.0 (from gymnasium==0.28.1->stable-
baselines3==2.0.0a5->-r https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1))
  Downloading jax_jumpy-1.0.0-py3-none-any.whl (20 kB)
Requirement already satisfied: typing-extensions>=4.3.0 in
/usr/local/lib/python3.10/dist-packages (from gymnasium==0.28.1->stable-
baselines3==2.0.0a5->-r https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (4.7.1)
Collecting farama-notifications>=0.0.1 (from gymnasium==0.28.1->stable-
baselines3==2.0.0a5->-r https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1))
  Downloading Farama_Notifications-0.0.4-py3-none-any.whl (2.5 kB)
INFO: pip is looking at multiple versions of gymnasium[box2d] to determine which
version is compatible with other requirements. This could take a while.
Collecting box2d-py==2.3.5 (from gymnasium==0.28.1->stable-
baselines3==2.0.0a5->-r https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1))
  Downloading box2d-py-2.3.5.tar.gz (374 kB)
                           374.4/374.4 kB
40.6 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Collecting pygame==2.1.3 (from gymnasium==0.28.1->stable-baselines3==2.0.0a5->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1))
  Downloading
pygame-2.1.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (13.7
MB)
                            13.7/13.7 MB
52.1 MB/s eta 0:00:00
Collecting huggingface-hub~=0.8 (from huggingface_sb3->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 4))
  Downloading huggingface_hub-0.16.4-py3-none-any.whl (268 kB)
                           268.8/268.8 kB
31.7 MB/s eta 0:00:00
Requirement already satisfied: pyyaml~=6.0 in
/usr/local/lib/python3.10/dist-packages (from huggingface_sb3->-r
```

```
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 4)) (6.0.1)
Requirement already satisfied: wasabi in /usr/local/lib/python3.10/dist-packages
(from huggingface_sb3->-r https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 4)) (1.1.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-
packages (from huggingface-hub~=0.8->huggingface_sb3->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 4)) (3.12.2)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages
(from huggingface-hub~=0.8->huggingface_sb3->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 4)) (2023.6.0)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-
packages (from huggingface-hub~=0.8->huggingface_sb3->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 4)) (2.31.0)
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.10/dist-
packages (from huggingface-hub~=0.8->huggingface_sb3->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 4)) (4.66.1)
Requirement already satisfied: packaging>=20.9 in
/usr/local/lib/python3.10/dist-packages (from huggingface-
hub~=0.8->huggingface_sb3->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 4)) (23.1)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages
(from torch>=1.11->stable-baselines3==2.0.0a5->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (1.12)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-
packages (from torch>=1.11->stable-baselines3==2.0.0a5->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (3.1)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages
(from torch>=1.11->stable-baselines3==2.0.0a5->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (3.1.2)
Requirement already satisfied: triton==2.0.0 in /usr/local/lib/python3.10/dist-
packages (from torch>=1.11->stable-baselines3==2.0.0a5->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (2.0.0)
Requirement already satisfied: cmake in /usr/local/lib/python3.10/dist-packages
(from triton==2.0.0->torch>=1.11->stable-baselines3==2.0.0a5->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (3.27.2)
Requirement already satisfied: lit in /usr/local/lib/python3.10/dist-packages
(from triton==2.0.0->torch>=1.11->stable-baselines3==2.0.0a5->-r
```

```
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (16.0.6)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->stable-
baselines3==2.0.0a5->-r https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (1.1.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-
packages (from matplotlib->stable-baselines3==2.0.0a5->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->stable-
baselines3==2.0.0a5->-r https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (4.42.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->stable-
baselines3==2.0.0a5->-r https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (1.4.4)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-
packages (from matplotlib->stable-baselines3==2.0.0a5->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->stable-
baselines3==2.0.0a5->-r https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->stable-
baselines3==2.0.0a5->-r https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas->stable-baselines3==2.0.0a5->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (2023.3)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-
packages (from python-dateutil>=2.7->matplotlib->stable-baselines3==2.0.0a5->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (1.16.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.11->stable-
baselines3==2.0.0a5->-r https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (2.1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface-
hub~=0.8->huggingface_sb3->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 4)) (3.2.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
```

```
packages (from requests->huggingface-hub~=0.8->huggingface_sb3->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 4)) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface-
hub~=0.8->huggingface_sb3->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 4)) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests->huggingface-
hub~=0.8->huggingface_sb3->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 4)) (2023.7.22)
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-
packages (from sympy->torch>=1.11->stable-baselines3==2.0.0a5->-r
https://raw.githubusercontent.com/huggingface/deep-rl-
class/main/notebooks/unit1/requirements-unit1.txt (line 1)) (1.3.0)
Building wheels for collected packages: box2d-py
  Building wheel for box2d-py (setup.py) … done
  Created wheel for box2d-py:
filename=box2d_py-2.3.5-cp310-cp310-linux_x86_64.whl size=2349118
sha256=713b51288cf886a70df24cac4c307457e7cf5c8c7a8443559e4d5e2ed53d0fa1
  Stored in directory: /root/.cache/pip/wheels/db/8f/6a/eaaadf056fba10a98d986f6d
ce954e6201ba3126926fc5ad9e
Successfully built box2d-py
Installing collected packages: swig, farama-notifications, box2d-py, pygame,
jax-jumpy, huggingface-hub, gymnasium, huggingface_sb3, stable-baselines3
  Attempting uninstall: pygame
    Found existing installation: pygame 2.5.1
    Uninstalling pygame-2.5.1:
      Successfully uninstalled pygame-2.5.1
Successfully installed box2d-py-2.3.5 farama-notifications-0.0.4
gymnasium-0.28.1 huggingface-hub-0.16.4 huggingface_sb3-2.3 jax-jumpy-1.0.0
pygame-2.1.3 stable-baselines3-2.0.0a5 swig-4.1.1
Get:1 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease
[3,626 B]
Get:2 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64
InRelease [1,581 B]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:4 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64
Packages [458 kB]
Hit:5 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages
[980 kB]
Hit:8 https://ppa.launchpadcontent.net/c2d4u.team/c2d4u4.0+/ubuntu jammy
InRelease
Get:9 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [860
```

```
kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1,136
kB]
Hit:12 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy InRelease
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages
[1,241 kB]
Hit:14 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy
InRelease
Hit:15 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy InRelease
Fetched 5,017 kB in 3s (1,985 kB/s)
Reading package lists… Done
Reading package lists… Done
Building dependency tree… Done
Reading state information… Done
The following additional packages will be installed:
  freeglut3 libglu1-mesa
Suggested packages:
  libgle3 python3-numpy
The following NEW packages will be installed:
  freeglut3 libglu1-mesa python3-opengl
0 upgraded, 3 newly installed, 0 to remove and 16 not upgraded.
Need to get 824 kB of archives.
After this operation, 8,092 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 freeglut3 amd64
2.8.1-6 [74.0 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 libglu1-mesa amd64
9.0.2-1 [145 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/universe amd64 python3-opengl all
3.1.5+dfsg-1 [605 kB]
Fetched 824 kB in 1s (1,214 kB/s)
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based
frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 78,
<> line 3.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Selecting previously unselected package freeglut3:amd64.
(Reading database … 121584 files and directories currently installed.)
Preparing to unpack …/freeglut3_2.8.1-6_amd64.deb …
Unpacking freeglut3:amd64 (2.8.1-6) …
Selecting previously unselected package libglu1-mesa:amd64.
Preparing to unpack …/libglu1-mesa_9.0.2-1_amd64.deb …
Unpacking libglu1-mesa:amd64 (9.0.2-1) …
Selecting previously unselected package python3-opengl.
```

```
Preparing to unpack …/python3-opengl_3.1.5+dfsg-1_all.deb …
Unpacking python3-opengl (3.1.5+dfsg-1) …
Setting up freeglut3:amd64 (2.8.1-6) …
Setting up libglu1-mesa:amd64 (9.0.2-1) …
Setting up python3-opengl (3.1.5+dfsg-1) …
Processing triggers for libc-bin (2.35-0ubuntu3.1) …
/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic
link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link

Reading package lists… Done
Building dependency tree… Done
Reading state information… Done
ffmpeg is already the newest version (7:4.4.2-0ubuntu0.22.04.1).
0 upgraded, 0 newly installed, 0 to remove and 16 not upgraded.
Reading package lists… Done
Building dependency tree… Done
Reading state information… Done
The following additional packages will be installed:
  libfontenc1 libxfont2 libxkbfile1 x11-xkb-utils xfonts-base xfonts-encodings
  xfonts-utils xserver-common
The following NEW packages will be installed:
  libfontenc1 libxfont2 libxkbfile1 x11-xkb-utils xfonts-base xfonts-encodings
  xfonts-utils xserver-common xvfb
0 upgraded, 9 newly installed, 0 to remove and 16 not upgraded.
Need to get 7,812 kB of archives.
After this operation, 11.9 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 libfontenc1 amd64
1:1.1.4-1build3 [14.7 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 libxfont2 amd64
1:2.0.5-1build1 [94.5 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 libxkbfile1 amd64
1:1.1.0-1build3 [71.8 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy/main amd64 x11-xkb-utils amd64
7.7+5build4 [172 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/main amd64 xfonts-encodings all
1:1.0.5-0ubuntu2 [578 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy/main amd64 xfonts-utils amd64
1:7.7+6build2 [94.6 kB]
```

```
Get:7 http://archive.ubuntu.com/ubuntu jammy/main amd64 xfonts-base all 1:1.0.5
[5,896 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 xserver-common
all 2:21.1.4-2ubuntu1.7~22.04.1 [28.0 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 xvfb amd64
2:21.1.4-2ubuntu1.7~22.04.1 [863 kB]
Fetched 7,812 kB in 1s (9,108 kB/s)
Selecting previously unselected package libfontenc1:amd64.
(Reading database … 124668 files and directories currently installed.)
Preparing to unpack …/0-libfontenc1_1%3a1.1.4-1build3_amd64.deb …
Unpacking libfontenc1:amd64 (1:1.1.4-1build3) …
Selecting previously unselected package libxfont2:amd64.
Preparing to unpack …/1-libxfont2_1%3a2.0.5-1build1_amd64.deb …
Unpacking libxfont2:amd64 (1:2.0.5-1build1) …
Selecting previously unselected package libxkbfile1:amd64.
Preparing to unpack …/2-libxkbfile1_1%3a1.1.0-1build3_amd64.deb …
Unpacking libxkbfile1:amd64 (1:1.1.0-1build3) …
Selecting previously unselected package x11-xkb-utils.
Preparing to unpack …/3-x11-xkb-utils_7.7+5build4_amd64.deb …
Unpacking x11-xkb-utils (7.7+5build4) …
Selecting previously unselected package xfonts-encodings.
Preparing to unpack …/4-xfonts-encodings_1%3a1.0.5-0ubuntu2_all.deb …
Unpacking xfonts-encodings (1:1.0.5-0ubuntu2) …
Selecting previously unselected package xfonts-utils.
Preparing to unpack …/5-xfonts-utils_1%3a7.7+6build2_amd64.deb …
Unpacking xfonts-utils (1:7.7+6build2) …
Selecting previously unselected package xfonts-base.
Preparing to unpack …/6-xfonts-base_1%3a1.0.5_all.deb …
Unpacking xfonts-base (1:1.0.5) …
Selecting previously unselected package xserver-common.
Preparing to unpack …/7-xserver-common_2%3a21.1.4-2ubuntu1.7~22.04.1_all.deb
…
Unpacking xserver-common (2:21.1.4-2ubuntu1.7~22.04.1) …
Selecting previously unselected package xvfb.
Preparing to unpack …/8-xvfb_2%3a21.1.4-2ubuntu1.7~22.04.1_amd64.deb …
Unpacking xvfb (2:21.1.4-2ubuntu1.7~22.04.1) …
Setting up libfontenc1:amd64 (1:1.1.4-1build3) …
Setting up xfonts-encodings (1:1.0.5-0ubuntu2) …
Setting up libxkbfile1:amd64 (1:1.1.0-1build3) …
Setting up libxfont2:amd64 (1:2.0.5-1build1) …
Setting up x11-xkb-utils (7.7+5build4) …
Setting up xfonts-utils (1:7.7+6build2) …
Setting up xfonts-base (1:1.0.5) …
Setting up xserver-common (2:21.1.4-2ubuntu1.7~22.04.1) …
Setting up xvfb (2:21.1.4-2ubuntu1.7~22.04.1) …
Processing triggers for man-db (2.10.2-1) …
Processing triggers for fontconfig (2.13.1-4.2ubuntu5) …
Processing triggers for libc-bin (2.35-0ubuntu3.1) …
```

```
/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic
link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link

Collecting pyvirtualdisplay
  Downloading PyVirtualDisplay-3.0-py3-none-any.whl (15 kB)
Installing collected packages: pyvirtualdisplay
Successfully installed pyvirtualdisplay-3.0
```

To make sure the new installed libraries are used, sometimes it's required to restart the notebook runtime. The next cell will force the runtime to crash, so you'll need to connect again and run the code starting from here. Thanks to this trick, we will be able to run our virtual screen.

[ ]:
```
import os

os.kill(os.getpid(), 9)
```

[1]:
```
# Virtual display
from pyvirtualdisplay import Display

virtual_display = Display(visible=0, size=(1400, 900))
virtual_display.start()
```

[1]:
```
<pyvirtualdisplay.display.Display at 0x7a11b82abc40>
```

2 Import the packages

[2]:
```
import gymnasium

from huggingface_sb3 import load_from_hub, package_to_hub
from huggingface_hub import notebook_login # To log to our Hugging Face account␣
 ↪to be able to upload models to the Hub.

from stable_baselines3 import PPO
from stable_baselines3.common.env_util import make_vec_env
from stable_baselines3.common.evaluation import evaluate_policy
from stable_baselines3.common.monitor import Monitor
```

Understand Gymnasium and how it works The library containing our environment is called Gymnasium. You'll use Gymnasium a lot in Deep Reinforcement Learning.

Gymnasium is the new version of Gym library maintained by the Farama Foundation.

The Gymnasium library provides two things:

An interface that allows you to create RL environments. A collection of environments (gym-control, atari, box2D…). Let's look at an example, but first let's recall the RL loop.

At each step: - Our Agent receives a **state (S0)** from the **Environment** — we receive the first frame of our game (Environment). - Based on that **state (S0),** the Agent takes an **action (A0)** — our Agent will move to the right. - The environment transitions to a **new state (S1)** — new frame. - The environment gives some **reward (R1)** to the Agent — we're not dead *(Positive Reward +1)*.

With Gymnasium:

1 We create our environment using `gymnasium.make()`

2 We reset the environment to its initial state with `observation = env.reset()`

At each step:

3 Get an action using our model (in our example we take a random action)

4 Using `env.step(action)`, we perform this action in the environment and get - `observation`: The new state (st+1) - `reward`: The reward we get after executing the action - `terminated`: Indicates if the episode terminated (agent reach the terminal state) - `truncated`: Introduced with this new version, it indicates a timelimit or if an agent go out of bounds of the environment for instance. - `info`: A dictionary that provides additional information (depends on the environment).

For more explanations check this https://gymnasium.farama.org/api/env/#gymnasium.Env.step

If the episode is terminated: - We reset the environment to its initial state with `observation = env.reset()`

Let's look at an example! Make sure to read the code

LunarLander-v2 (Discrete)

Landing pad is always at coordinates (0,0). Coordinates are the first two numbers in state vector. Reward for moving from the top of the screen to landing pad and zero speed is about 100..140 points. If lander moves away from landing pad it loses reward back. Episode finishes if the lander crashes or comes to rest, receiving additional -100 or +100 points. Each leg ground contact is +10. Firing main engine is -0.3 points each frame. Solved is 200 points. Landing outside landing pad is possible. Fuel is infinite, so an agent can learn to fly and then land on its first attempt. Four discrete actions available: do nothing, fire left orientation engine, fire main engine, fire right orientation engine.

LunarLander-v2 is a scenario developed by Oleg Klimov, an engineer at OpenAI, inspired by the original Atari Lunar Lander (https://github.com/olegklimov). In the implementation, you have to take your landing pod to a lunar pad that is always located at coordinates x=0 and y=0. In addition, your actual x and y position is known since their

values are stored in the first two elements of the state vector, the vector that contains all the information for the reinforcement learning algorithm to decide the best action to take at a certain moment. Source: https://learning.oreilly.com/library/view/tensorflow-deep-learning/9781788398060/04be3bfb-74a9-44eb-8ecb-de954e7696fb.xhtml

[3]:
```
import gymnasium as gym

# First, we create our environment called LunarLander-v2
env = gym.make("LunarLander-v2")

# Then we reset this environment
observation, info = env.reset()

for _ in range(20):
  # Take a random action
  action = env.action_space.sample()
  print("Action taken:", action)

  # Do this action in the environment and get
  # next_state, reward, terminated, truncated and info
  observation, reward, terminated, truncated, info = env.step(action)

  # If the game is terminated (in our case we land, crashed) or truncated␣
 ↪(timeout)
  if terminated or truncated:
      # Reset the environment
      print("Environment is reset")
      observation, info = env.reset()

env.close()
```

```
Action taken: 0
Action taken: 2
Action taken: 3
Action taken: 1
Action taken: 3
Action taken: 1
Action taken: 3
Action taken: 3
Action taken: 0
Action taken: 3
Action taken: 1
Action taken: 0
Action taken: 2
Action taken: 1
Action taken: 1
Action taken: 2
Action taken: 0
```

```
Action taken: 1
Action taken: 2
Action taken: 2
```

2.1 Create the LunarLander environment and understand how it works

2.1.1 The environment

In this first tutorial, we're going to train our agent, a Lunar Lander, **to land correctly on the moon**. To do that, the agent needs to learn **to adapt its speed and position (horizontal, vertical, and angular) to land correctly.**

A good habit when you start to use an environment is to check its documentation

https://gymnasium.farama.org/environments/box2d/lunar_lander/

---

[4]:
```
# We create our environment with gym.make("<name_of_the_environment>")
env = gym.make("LunarLander-v2")
env.reset()
print("_____OBSERVATION SPACE_____ \n")
print("Observation Space Shape", env.observation_space.shape)
print("Sample observation", env.observation_space.sample()) # Get a random␣
 ↪observation
```

```
_____OBSERVATION SPACE_____

Observation Space Shape (8,)
Sample observation [-20.141338   -59.90314      4.230214     0.14666176
-1.7034489
  -4.502537     0.61853015   0.20563099]
```

Vectorized Environment

- We create a vectorized environment (a method for stacking multiple independent environments into a single environment) of 16 environments, this way, **we'll have more diverse experiences during the training.**

[5]:
```
# Create the environment
env = make_vec_env('LunarLander-v2', n_envs=16)
```

2.2 Create the Model

- We have studied our environment and we understood the problem: **being able to land the Lunar Lander to the Landing Pad correctly by controlling left, right and main orientation engine**. Now let's build the algorithm we're going to use to solve this Problem .
- To do so, we're going to use our first Deep RL library, Stable Baselines3 (SB3).

- SB3 is a set of **reliable implementations of reinforcement learning algorithms in PyTorch**.

A good habit when using a new library is to dive first on the documentation: https://stable-baselines3.readthedocs.io/en/master/ and then try some tutorials.

To solve this problem, we're going to use SB3 **PPO**. PPO (aka Proximal Policy Optimization) is one of the SOTA (state of the art) Deep Reinforcement Learning algorithms that you'll study during this course.

PPO is a combination of: - *Value-based reinforcement learning method*: learning an action-value function that will tell us the **most valuable action to take given a state and action**. - *Policy-based reinforcement learning method*: learning a policy that will **give us a probability distribution over actions**.

Stable-Baselines3 is easy to set up:

1 You **create your environment** (in our case it was done above)

2 You define the **model you want to use and instantiate this model** `model = PPO("MlpPolicy")`

3 You **train the agent** with `model.learn` and define the number of training timesteps

```
# Create environment
env = gym.make('LunarLander-v2')

# Instantiate the agent
model = PPO('MlpPolicy', env, verbose=1)
# Train the agent
model.learn(total_timesteps=int(2e5))
```

[6]:
```
model = PPO(
    policy = 'MlpPolicy',
    env = env,
    n_steps = 1024,
    batch_size = 64,
    n_epochs = 4,
    gamma = 0.999,
    gae_lambda = 0.98,
    ent_coef = 0.01,
    verbose=1)
```

```
Using cuda device
```

2.3 Train the PPO agent

- Let's train our agent for 1,000,000 timesteps, don't forget to use GPU on Colab. It will take approximately ~20min, but you can use fewer timesteps if you just want to try it out.

- During the training, take a break you deserved it

[7]:
```
# Train it for 1,000,000 timesteps
model.learn(total_timesteps=1000000)
# Save the model
model_name = "ppo-LunarLander-v2"
model.save(model_name)
```

```
---------------------------------
| rollout/           |          |
|    ep_len_mean     | 91.4     |
|    ep_rew_mean     | -173     |
| time/              |          |
|    fps             | 2057     |
|    iterations      | 1        |
|    time_elapsed    | 7        |
|    total_timesteps | 16384    |
---------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 95.4        |
|    ep_rew_mean          | -147        |
| time/                   |             |
|    fps                  | 1994        |
|    iterations           | 2           |
|    time_elapsed         | 16          |
|    total_timesteps      | 32768       |
| train/                  |             |
|    approx_kl            | 0.008592849 |
|    clip_fraction        | 0.0386      |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.38       |
|    explained_variance   | 0.000699    |
|    learning_rate        | 0.0003      |
|    loss                 | 1.98e+03    |
|    n_updates            | 4           |
|    policy_gradient_loss | -0.00511    |
|    value_loss           | 4.64e+03    |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 101         |
|    ep_rew_mean          | -131        |
| time/                   |             |
|    fps                  | 1880        |
|    iterations           | 3           |
|    time_elapsed         | 26          |
|    total_timesteps      | 49152       |
```

```
| train/                  |             |
|    approx_kl            | 0.008314934 |
|    clip_fraction        | 0.0428      |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.36       |
|    explained_variance   | -0.00236    |
|    learning_rate        | 0.0003      |
|    loss                 | 882         |
|    n_updates            | 8           |
|    policy_gradient_loss | -0.00531    |
|    value_loss           | 2.49e+03    |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 108         |
|    ep_rew_mean          | -127        |
| time/                   |             |
|    fps                  | 1782        |
|    iterations           | 4           |
|    time_elapsed         | 36          |
|    total_timesteps      | 65536       |
| train/                  |             |
|    approx_kl            | 0.00751484  |
|    clip_fraction        | 0.0368      |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.33       |
|    explained_variance   | -0.0024     |
|    learning_rate        | 0.0003      |
|    loss                 | 677         |
|    n_updates            | 12          |
|    policy_gradient_loss | -0.00471    |
|    value_loss           | 1.42e+03    |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 109         |
|    ep_rew_mean          | -104        |
| time/                   |             |
|    fps                  | 1713        |
|    iterations           | 5           |
|    time_elapsed         | 47          |
|    total_timesteps      | 81920       |
| train/                  |             |
|    approx_kl            | 0.008963551 |
|    clip_fraction        | 0.116       |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.31       |
|    explained_variance   | -0.000243   |
```

```
|    learning_rate        | 0.0003      |
|    loss                 | 447         |
|    n_updates            | 16          |
|    policy_gradient_loss | -0.00294    |
|    value_loss           | 1e+03       |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 110         |
|    ep_rew_mean          | -96.5       |
| time/                   |             |
|    fps                  | 1704        |
|    iterations           | 6           |
|    time_elapsed         | 57          |
|    total_timesteps      | 98304       |
| train/                  |             |
|    approx_kl            | 0.009971031 |
|    clip_fraction        | 0.114       |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.28       |
|    explained_variance   | -4.43e-05   |
|    learning_rate        | 0.0003      |
|    loss                 | 216         |
|    n_updates            | 20          |
|    policy_gradient_loss | -0.00693    |
|    value_loss           | 610         |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 125         |
|    ep_rew_mean          | -81.8       |
| time/                   |             |
|    fps                  | 1662        |
|    iterations           | 7           |
|    time_elapsed         | 68          |
|    total_timesteps      | 114688      |
| train/                  |             |
|    approx_kl            | 0.009136068 |
|    clip_fraction        | 0.0764      |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.28       |
|    explained_variance   | -5.91e-05   |
|    learning_rate        | 0.0003      |
|    loss                 | 416         |
|    n_updates            | 24          |
|    policy_gradient_loss | -0.00402    |
|    value_loss           | 843         |
-----------------------------------------
```

```
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 117         |
|    ep_rew_mean          | -52.1       |
| time/                   |             |
|    fps                  | 1656        |
|    iterations           | 8           |
|    time_elapsed         | 79          |
|    total_timesteps      | 131072      |
| train/                  |             |
|    approx_kl            | 0.006604066 |
|    clip_fraction        | 0.0534      |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.24       |
|    explained_variance   | 2.84e-05    |
|    learning_rate        | 0.0003      |
|    loss                 | 166         |
|    n_updates            | 28          |
|    policy_gradient_loss | -0.00503    |
|    value_loss           | 512         |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 124          |
|    ep_rew_mean          | -37.8        |
| time/                   |              |
|    fps                  | 1647         |
|    iterations           | 9            |
|    time_elapsed         | 89           |
|    total_timesteps      | 147456       |
| train/                  |              |
|    approx_kl            | 0.0069744745 |
|    clip_fraction        | 0.06         |
|    clip_range           | 0.2          |
|    entropy_loss         | -1.22        |
|    explained_variance   | -5.65e-05    |
|    learning_rate        | 0.0003       |
|    loss                 | 173          |
|    n_updates            | 32           |
|    policy_gradient_loss | -0.00369     |
|    value_loss           | 421          |
-----------------------------------------
----------------------------------------
| rollout/                |            |
|    ep_len_mean          | 136        |
|    ep_rew_mean          | -23.8      |
| time/                   |            |
|    fps                  | 1625       |
```

```
|    iterations           | 10         |
|    time_elapsed         | 100        |
|    total_timesteps      | 163840     |
| train/                  |            |
|    approx_kl            | 0.00981756 |
|    clip_fraction        | 0.0448     |
|    clip_range           | 0.2        |
|    entropy_loss         | -1.21      |
|    explained_variance   | 1.22e-05   |
|    learning_rate        | 0.0003     |
|    loss                 | 207        |
|    n_updates            | 36         |
|    policy_gradient_loss | -0.00282   |
|    value_loss           | 435        |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 159         |
|    ep_rew_mean          | -14.5       |
| time/                   |             |
|    fps                  | 1583        |
|    iterations           | 11          |
|    time_elapsed         | 113         |
|    total_timesteps      | 180224      |
| train/                  |             |
|    approx_kl            | 0.0049166875 |
|    clip_fraction        | 0.0174      |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.17       |
|    explained_variance   | -0.000786   |
|    learning_rate        | 0.0003      |
|    loss                 | 149         |
|    n_updates            | 40          |
|    policy_gradient_loss | -0.00211    |
|    value_loss           | 505         |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 189         |
|    ep_rew_mean          | -12.2       |
| time/                   |             |
|    fps                  | 1486        |
|    iterations           | 12          |
|    time_elapsed         | 132         |
|    total_timesteps      | 196608      |
| train/                  |             |
|    approx_kl            | 0.0028343312 |
|    clip_fraction        | 0.0235      |
```

```
|    clip_range           | 0.2          |
|    entropy_loss         | -1.18        |
|    explained_variance   | 0.000166     |
|    learning_rate        | 0.0003       |
|    loss                 | 172          |
|    n_updates            | 44           |
|    policy_gradient_loss | -0.00187     |
|    value_loss           | 487          |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 299          |
|    ep_rew_mean          | -6.46        |
| time/                   |              |
|    fps                  | 1399         |
|    iterations           | 13           |
|    time_elapsed         | 152          |
|    total_timesteps      | 212992       |
| train/                  |              |
|    approx_kl            | 0.008178342  |
|    clip_fraction        | 0.0506       |
|    clip_range           | 0.2          |
|    entropy_loss         | -1.21        |
|    explained_variance   | 0.00455      |
|    learning_rate        | 0.0003       |
|    loss                 | 162          |
|    n_updates            | 48           |
|    policy_gradient_loss | -0.00297     |
|    value_loss           | 363          |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 356          |
|    ep_rew_mean          | 0.26         |
| time/                   |              |
|    fps                  | 1313         |
|    iterations           | 14           |
|    time_elapsed         | 174          |
|    total_timesteps      | 229376       |
| train/                  |              |
|    approx_kl            | 0.0037551734 |
|    clip_fraction        | 0.00771      |
|    clip_range           | 0.2          |
|    entropy_loss         | -1.22        |
|    explained_variance   | 0.33         |
|    learning_rate        | 0.0003       |
|    loss                 | 139          |
|    n_updates            | 52           |
```

```
|    policy_gradient_loss | -0.00174     |
|    value_loss           | 322          |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 384          |
|    ep_rew_mean          | 6.92         |
| time/                   |              |
|    fps                  | 1238         |
|    iterations           | 15           |
|    time_elapsed         | 198          |
|    total_timesteps      | 245760       |
| train/                  |              |
|    approx_kl            | 0.0061729834 |
|    clip_fraction        | 0.0394       |
|    clip_range           | 0.2          |
|    entropy_loss         | -1.19        |
|    explained_variance   | 0.525        |
|    learning_rate        | 0.0003       |
|    loss                 | 110          |
|    n_updates            | 56           |
|    policy_gradient_loss | -0.00249     |
|    value_loss           | 273          |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 460          |
|    ep_rew_mean          | 14.3         |
| time/                   |              |
|    fps                  | 1190         |
|    iterations           | 16           |
|    time_elapsed         | 220          |
|    total_timesteps      | 262144       |
| train/                  |              |
|    approx_kl            | 0.0047675483 |
|    clip_fraction        | 0.0254       |
|    clip_range           | 0.2          |
|    entropy_loss         | -1.23        |
|    explained_variance   | 0.66         |
|    learning_rate        | 0.0003       |
|    loss                 | 56.5         |
|    n_updates            | 60           |
|    policy_gradient_loss | -0.00289     |
|    value_loss           | 212          |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 518          |
```

```
|    ep_rew_mean          | 21.8         |
| time/                   |              |
|    fps                  | 1156         |
|    iterations           | 17           |
|    time_elapsed         | 240          |
|    total_timesteps      | 278528       |
| train/                  |              |
|    approx_kl            | 0.0057203174 |
|    clip_fraction        | 0.0206       |
|    clip_range           | 0.2          |
|    entropy_loss         | -1.21        |
|    explained_variance   | 0.73         |
|    learning_rate        | 0.0003       |
|    loss                 | 116          |
|    n_updates            | 64           |
|    policy_gradient_loss | -0.00181     |
|    value_loss           | 209          |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 559          |
|    ep_rew_mean          | 24.1         |
| time/                   |              |
|    fps                  | 1118         |
|    iterations           | 18           |
|    time_elapsed         | 263          |
|    total_timesteps      | 294912       |
| train/                  |              |
|    approx_kl            | 0.006825205  |
|    clip_fraction        | 0.0734       |
|    clip_range           | 0.2          |
|    entropy_loss         | -1.18        |
|    explained_variance   | 0.821        |
|    learning_rate        | 0.0003       |
|    loss                 | 109          |
|    n_updates            | 68           |
|    policy_gradient_loss | -0.00244     |
|    value_loss           | 166          |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 583          |
|    ep_rew_mean          | 32.5         |
| time/                   |              |
|    fps                  | 1078         |
|    iterations           | 19           |
|    time_elapsed         | 288          |
|    total_timesteps      | 311296       |
```

```
| train/                  |             |
|    approx_kl            | 0.005868584 |
|    clip_fraction        | 0.0397      |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.2        |
|    explained_variance   | 0.853       |
|    learning_rate        | 0.0003      |
|    loss                 | 70.1        |
|    n_updates            | 72          |
|    policy_gradient_loss | -0.00224    |
|    value_loss           | 106         |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 649          |
|    ep_rew_mean          | 45.6         |
| time/                   |              |
|    fps                  | 1043         |
|    iterations           | 20           |
|    time_elapsed         | 314          |
|    total_timesteps      | 327680       |
| train/                  |              |
|    approx_kl            | 0.0054366277 |
|    clip_fraction        | 0.0382       |
|    clip_range           | 0.2          |
|    entropy_loss         | -1.18        |
|    explained_variance   | 0.796        |
|    learning_rate        | 0.0003       |
|    loss                 | 103          |
|    n_updates            | 76           |
|    policy_gradient_loss | -0.00313     |
|    value_loss           | 182          |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 721         |
|    ep_rew_mean          | 58.4        |
| time/                   |             |
|    fps                  | 1010        |
|    iterations           | 21          |
|    time_elapsed         | 340         |
|    total_timesteps      | 344064      |
| train/                  |             |
|    approx_kl            | 0.004255467 |
|    clip_fraction        | 0.0282      |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.18       |
|    explained_variance   | 0.901       |
```

```
|    learning_rate        | 0.0003      |
|    loss                 | 22.7        |
|    n_updates            | 80          |
|    policy_gradient_loss | -0.00119    |
|    value_loss           | 74.4        |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 784         |
|    ep_rew_mean          | 62.6        |
| time/                   |             |
|    fps                  | 981         |
|    iterations           | 22          |
|    time_elapsed         | 367         |
|    total_timesteps      | 360448      |
| train/                  |             |
|    approx_kl            | 0.0045252834 |
|    clip_fraction        | 0.0338      |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.14       |
|    explained_variance   | 0.926       |
|    learning_rate        | 0.0003      |
|    loss                 | 18.7        |
|    n_updates            | 84          |
|    policy_gradient_loss | -0.00135    |
|    value_loss           | 54.4        |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 766         |
|    ep_rew_mean          | 63.5        |
| time/                   |             |
|    fps                  | 964         |
|    iterations           | 23          |
|    time_elapsed         | 390         |
|    total_timesteps      | 376832      |
| train/                  |             |
|    approx_kl            | 0.0060958676 |
|    clip_fraction        | 0.0592      |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.11       |
|    explained_variance   | 0.886       |
|    learning_rate        | 0.0003      |
|    loss                 | 10.6        |
|    n_updates            | 88          |
|    policy_gradient_loss | -0.00281    |
|    value_loss           | 84.6        |
-----------------------------------------
```

```
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 770         |
|    ep_rew_mean          | 66.4        |
| time/                   |             |
|    fps                  | 948         |
|    iterations           | 24          |
|    time_elapsed         | 414         |
|    total_timesteps      | 393216      |
| train/                  |             |
|    approx_kl            | 0.004979129 |
|    clip_fraction        | 0.0494      |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.08       |
|    explained_variance   | 0.877       |
|    learning_rate        | 0.0003      |
|    loss                 | 103         |
|    n_updates            | 92          |
|    policy_gradient_loss | -0.0019     |
|    value_loss           | 129         |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 720         |
|    ep_rew_mean          | 63.6        |
| time/                   |             |
|    fps                  | 932         |
|    iterations           | 25          |
|    time_elapsed         | 439         |
|    total_timesteps      | 409600      |
| train/                  |             |
|    approx_kl            | 0.006599961 |
|    clip_fraction        | 0.0355      |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.06       |
|    explained_variance   | 0.889       |
|    learning_rate        | 0.0003      |
|    loss                 | 30.4        |
|    n_updates            | 96          |
|    policy_gradient_loss | -0.000833   |
|    value_loss           | 126         |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 684         |
|    ep_rew_mean          | 69.2        |
| time/                   |             |
|    fps                  | 922         |
```

```
|    iterations           | 26           |
|    time_elapsed         | 461          |
|    total_timesteps      | 425984       |
| train/                  |              |
|    approx_kl            | 0.0058977026 |
|    clip_fraction        | 0.0418       |
|    clip_range           | 0.2          |
|    entropy_loss         | -1.06        |
|    explained_variance   | 0.858        |
|    learning_rate        | 0.0003       |
|    loss                 | 166          |
|    n_updates            | 100          |
|    policy_gradient_loss | -0.00285     |
|    value_loss           | 135          |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 711          |
|    ep_rew_mean          | 81           |
| time/                   |              |
|    fps                  | 911          |
|    iterations           | 27           |
|    time_elapsed         | 485          |
|    total_timesteps      | 442368       |
| train/                  |              |
|    approx_kl            | 0.005546063  |
|    clip_fraction        | 0.0259       |
|    clip_range           | 0.2          |
|    entropy_loss         | -1.05        |
|    explained_variance   | 0.882        |
|    learning_rate        | 0.0003       |
|    loss                 | 61.7         |
|    n_updates            | 104          |
|    policy_gradient_loss | -0.000982    |
|    value_loss           | 128          |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 708          |
|    ep_rew_mean          | 85.6         |
| time/                   |              |
|    fps                  | 901          |
|    iterations           | 28           |
|    time_elapsed         | 508          |
|    total_timesteps      | 458752       |
| train/                  |              |
|    approx_kl            | 0.005223057  |
|    clip_fraction        | 0.056        |
```

```
|    clip_range           | 0.2         |
|    entropy_loss         | -1.01       |
|    explained_variance   | 0.871       |
|    learning_rate        | 0.0003      |
|    loss                 | 13.2        |
|    n_updates            | 108         |
|    policy_gradient_loss | -0.000185   |
|    value_loss           | 109         |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 721         |
|    ep_rew_mean          | 92.1        |
| time/                   |             |
|    fps                  | 896         |
|    iterations           | 29          |
|    time_elapsed         | 529         |
|    total_timesteps      | 475136      |
| train/                  |             |
|    approx_kl            | 0.0037519943 |
|    clip_fraction        | 0.0358      |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.03       |
|    explained_variance   | 0.873       |
|    learning_rate        | 0.0003      |
|    loss                 | 139         |
|    n_updates            | 112         |
|    policy_gradient_loss | -0.000845   |
|    value_loss           | 113         |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 737         |
|    ep_rew_mean          | 95.9        |
| time/                   |             |
|    fps                  | 885         |
|    iterations           | 30          |
|    time_elapsed         | 554         |
|    total_timesteps      | 491520      |
| train/                  |             |
|    approx_kl            | 0.0069649937 |
|    clip_fraction        | 0.0464      |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.03       |
|    explained_variance   | 0.856       |
|    learning_rate        | 0.0003      |
|    loss                 | 52.4        |
|    n_updates            | 116         |
```

```
|    policy_gradient_loss | -0.000394    |
|    value_loss           | 137          |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 761          |
|    ep_rew_mean          | 99.7         |
| time/                   |              |
|    fps                  | 877          |
|    iterations           | 31           |
|    time_elapsed         | 579          |
|    total_timesteps      | 507904       |
| train/                  |              |
|    approx_kl            | 0.0041906065 |
|    clip_fraction        | 0.0287       |
|    clip_range           | 0.2          |
|    entropy_loss         | -1.02        |
|    explained_variance   | 0.88         |
|    learning_rate        | 0.0003       |
|    loss                 | 159          |
|    n_updates            | 120          |
|    policy_gradient_loss | -0.000867    |
|    value_loss           | 104          |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 759          |
|    ep_rew_mean          | 96.8         |
| time/                   |              |
|    fps                  | 867          |
|    iterations           | 32           |
|    time_elapsed         | 604          |
|    total_timesteps      | 524288       |
| train/                  |              |
|    approx_kl            | 0.0038534352 |
|    clip_fraction        | 0.0332       |
|    clip_range           | 0.2          |
|    entropy_loss         | -1.05        |
|    explained_variance   | 0.919        |
|    learning_rate        | 0.0003       |
|    loss                 | 45           |
|    n_updates            | 124          |
|    policy_gradient_loss | 1.87e-05     |
|    value_loss           | 70.1         |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 783          |
```

```
|    ep_rew_mean          | 100          |
| time/                   |              |
|    fps                  | 861          |
|    iterations           | 33           |
|    time_elapsed         | 627          |
|    total_timesteps      | 540672       |
| train/                  |              |
|    approx_kl            | 0.0050528254 |
|    clip_fraction        | 0.0399       |
|    clip_range           | 0.2          |
|    entropy_loss         | -1.07        |
|    explained_variance   | 0.929        |
|    learning_rate        | 0.0003       |
|    loss                 | 34.5         |
|    n_updates            | 128          |
|    policy_gradient_loss | -0.000154    |
|    value_loss           | 71.6         |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 820          |
|    ep_rew_mean          | 104          |
| time/                   |              |
|    fps                  | 855          |
|    iterations           | 34           |
|    time_elapsed         | 651          |
|    total_timesteps      | 557056       |
| train/                  |              |
|    approx_kl            | 0.0059309453 |
|    clip_fraction        | 0.0369       |
|    clip_range           | 0.2          |
|    entropy_loss         | -1.06        |
|    explained_variance   | 0.948        |
|    learning_rate        | 0.0003       |
|    loss                 | 10.9         |
|    n_updates            | 132          |
|    policy_gradient_loss | -0.000717    |
|    value_loss           | 54.9         |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 848          |
|    ep_rew_mean          | 108          |
| time/                   |              |
|    fps                  | 846          |
|    iterations           | 35           |
|    time_elapsed         | 677          |
|    total_timesteps      | 573440       |
```

```
| train/                  |              |
|    approx_kl            | 0.0054781875 |
|    clip_fraction        | 0.0353       |
|    clip_range           | 0.2          |
|    entropy_loss         | -1.04        |
|    explained_variance   | 0.97         |
|    learning_rate        | 0.0003       |
|    loss                 | 13           |
|    n_updates            | 136          |
|    policy_gradient_loss | -0.000663    |
|    value_loss           | 27.4         |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 880          |
|    ep_rew_mean          | 115          |
| time/                   |              |
|    fps                  | 836          |
|    iterations           | 36           |
|    time_elapsed         | 704          |
|    total_timesteps      | 589824       |
| train/                  |              |
|    approx_kl            | 0.005830043  |
|    clip_fraction        | 0.0461       |
|    clip_range           | 0.2          |
|    entropy_loss         | -1.1         |
|    explained_variance   | 0.971        |
|    learning_rate        | 0.0003       |
|    loss                 | 3.07         |
|    n_updates            | 140          |
|    policy_gradient_loss | -8.33e-05    |
|    value_loss           | 25.5         |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 902          |
|    ep_rew_mean          | 116          |
| time/                   |              |
|    fps                  | 828          |
|    iterations           | 37           |
|    time_elapsed         | 731          |
|    total_timesteps      | 606208       |
| train/                  |              |
|    approx_kl            | 0.006710346  |
|    clip_fraction        | 0.038        |
|    clip_range           | 0.2          |
|    entropy_loss         | -1.09        |
|    explained_variance   | 0.979        |
```

```
|    learning_rate        | 0.0003      |
|    loss                 | 18.3        |
|    n_updates            | 144         |
|    policy_gradient_loss | -0.000692   |
|    value_loss           | 22.1        |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 924         |
|    ep_rew_mean          | 121         |
| time/                   |             |
|    fps                  | 821         |
|    iterations           | 38          |
|    time_elapsed         | 757         |
|    total_timesteps      | 622592      |
| train/                  |             |
|    approx_kl            | 0.005125195 |
|    clip_fraction        | 0.0336      |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.09       |
|    explained_variance   | 0.986       |
|    learning_rate        | 0.0003      |
|    loss                 | 3.32        |
|    n_updates            | 148         |
|    policy_gradient_loss | -0.000416   |
|    value_loss           | 14.4        |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 946         |
|    ep_rew_mean          | 127         |
| time/                   |             |
|    fps                  | 815         |
|    iterations           | 39          |
|    time_elapsed         | 783         |
|    total_timesteps      | 638976      |
| train/                  |             |
|    approx_kl            | 0.005139123 |
|    clip_fraction        | 0.0434      |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.08       |
|    explained_variance   | 0.973       |
|    learning_rate        | 0.0003      |
|    loss                 | 12.8        |
|    n_updates            | 152         |
|    policy_gradient_loss | -0.00122    |
|    value_loss           | 32.2        |
-----------------------------------------
```

```
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 961         |
|    ep_rew_mean          | 130         |
| time/                   |             |
|    fps                  | 811         |
|    iterations           | 40          |
|    time_elapsed         | 807         |
|    total_timesteps      | 655360      |
| train/                  |             |
|    approx_kl            | 0.009937766 |
|    clip_fraction        | 0.0721      |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.06       |
|    explained_variance   | 0.994       |
|    learning_rate        | 0.0003      |
|    loss                 | 4.59        |
|    n_updates            | 156         |
|    policy_gradient_loss | -0.00251    |
|    value_loss           | 5.8         |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 949         |
|    ep_rew_mean          | 128         |
| time/                   |             |
|    fps                  | 807         |
|    iterations           | 41          |
|    time_elapsed         | 831         |
|    total_timesteps      | 671744      |
| train/                  |             |
|    approx_kl            | 0.005214632 |
|    clip_fraction        | 0.0342      |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.02       |
|    explained_variance   | 0.993       |
|    learning_rate        | 0.0003      |
|    loss                 | 4.09        |
|    n_updates            | 160         |
|    policy_gradient_loss | -0.00183    |
|    value_loss           | 6.35        |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 926         |
|    ep_rew_mean          | 125         |
| time/                   |             |
|    fps                  | 804         |
```

```
|    iterations           | 42          |
|    time_elapsed         | 854         |
|    total_timesteps      | 688128      |
| train/                  |             |
|    approx_kl            | 0.003999071 |
|    clip_fraction        | 0.027       |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.03       |
|    explained_variance   | 0.971       |
|    learning_rate        | 0.0003      |
|    loss                 | 17.9        |
|    n_updates            | 164         |
|    policy_gradient_loss | -0.000946   |
|    value_loss           | 35.9        |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 926         |
|    ep_rew_mean          | 128         |
| time/                   |             |
|    fps                  | 801         |
|    iterations           | 43          |
|    time_elapsed         | 879         |
|    total_timesteps      | 704512      |
| train/                  |             |
|    approx_kl            | 0.0034697542 |
|    clip_fraction        | 0.0162      |
|    clip_range           | 0.2         |
|    entropy_loss         | -0.984      |
|    explained_variance   | 0.961       |
|    learning_rate        | 0.0003      |
|    loss                 | 34          |
|    n_updates            | 168         |
|    policy_gradient_loss | -0.00188    |
|    value_loss           | 53.1        |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 933         |
|    ep_rew_mean          | 129         |
| time/                   |             |
|    fps                  | 798         |
|    iterations           | 44          |
|    time_elapsed         | 902         |
|    total_timesteps      | 720896      |
| train/                  |             |
|    approx_kl            | 0.0065774596 |
|    clip_fraction        | 0.0568      |
```

```
|    clip_range           | 0.2          |
|    entropy_loss         | -0.992       |
|    explained_variance   | 0.995        |
|    learning_rate        | 0.0003       |
|    loss                 | 0.887        |
|    n_updates            | 172          |
|    policy_gradient_loss | -0.000302    |
|    value_loss           | 4.7          |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 933          |
|    ep_rew_mean          | 131          |
| time/                   |              |
|    fps                  | 797          |
|    iterations           | 45           |
|    time_elapsed         | 925          |
|    total_timesteps      | 737280       |
| train/                  |              |
|    approx_kl            | 0.0046285056 |
|    clip_fraction        | 0.0328       |
|    clip_range           | 0.2          |
|    entropy_loss         | -0.97        |
|    explained_variance   | 0.989        |
|    learning_rate        | 0.0003       |
|    loss                 | 1.37         |
|    n_updates            | 176          |
|    policy_gradient_loss | -0.000609    |
|    value_loss           | 13.4         |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 911          |
|    ep_rew_mean          | 130          |
| time/                   |              |
|    fps                  | 796          |
|    iterations           | 46           |
|    time_elapsed         | 946          |
|    total_timesteps      | 753664       |
| train/                  |              |
|    approx_kl            | 0.0037862058 |
|    clip_fraction        | 0.0439       |
|    clip_range           | 0.2          |
|    entropy_loss         | -0.946       |
|    explained_variance   | 0.996        |
|    learning_rate        | 0.0003       |
|    loss                 | 3.77         |
|    n_updates            | 180          |
```

```
|    policy_gradient_loss | -0.000404    |
|    value_loss           | 3.67         |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 940          |
|    ep_rew_mean          | 135          |
| time/                   |              |
|    fps                  | 790          |
|    iterations           | 47           |
|    time_elapsed         | 974          |
|    total_timesteps      | 770048       |
| train/                  |              |
|    approx_kl            | 0.0054154107 |
|    clip_fraction        | 0.0214       |
|    clip_range           | 0.2          |
|    entropy_loss         | -0.935       |
|    explained_variance   | 0.967        |
|    learning_rate        | 0.0003       |
|    loss                 | 11           |
|    n_updates            | 184          |
|    policy_gradient_loss | -0.000244    |
|    value_loss           | 44.4         |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 963          |
|    ep_rew_mean          | 139          |
| time/                   |              |
|    fps                  | 788          |
|    iterations           | 48           |
|    time_elapsed         | 997          |
|    total_timesteps      | 786432       |
| train/                  |              |
|    approx_kl            | 0.0031047282 |
|    clip_fraction        | 0.0409       |
|    clip_range           | 0.2          |
|    entropy_loss         | -0.931       |
|    explained_variance   | 0.995        |
|    learning_rate        | 0.0003       |
|    loss                 | 0.88         |
|    n_updates            | 188          |
|    policy_gradient_loss | -0.000506    |
|    value_loss           | 3.54         |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 963          |
```

```
|    ep_rew_mean          | 140          |
| time/                   |              |
|    fps                  | 784          |
|    iterations           | 49           |
|    time_elapsed         | 1023         |
|    total_timesteps      | 802816       |
| train/                  |              |
|    approx_kl            | 0.0055639897 |
|    clip_fraction        | 0.036        |
|    clip_range           | 0.2          |
|    entropy_loss         | -0.929       |
|    explained_variance   | 0.997        |
|    learning_rate        | 0.0003       |
|    loss                 | 1.05         |
|    n_updates            | 192          |
|    policy_gradient_loss | -0.000335    |
|    value_loss           | 3.08         |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 963         |
|    ep_rew_mean          | 141         |
| time/                   |             |
|    fps                  | 783         |
|    iterations           | 50          |
|    time_elapsed         | 1046        |
|    total_timesteps      | 819200      |
| train/                  |             |
|    approx_kl            | 0.004088245 |
|    clip_fraction        | 0.0379      |
|    clip_range           | 0.2         |
|    entropy_loss         | -0.922      |
|    explained_variance   | 0.998       |
|    learning_rate        | 0.0003      |
|    loss                 | 0.566       |
|    n_updates            | 196         |
|    policy_gradient_loss | -0.000669   |
|    value_loss           | 2.59        |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 963         |
|    ep_rew_mean          | 141         |
| time/                   |             |
|    fps                  | 781         |
|    iterations           | 51          |
|    time_elapsed         | 1069        |
|    total_timesteps      | 835584      |
```

```
| train/                  |             |
|    approx_kl            | 0.005680429 |
|    clip_fraction        | 0.0512      |
|    clip_range           | 0.2         |
|    entropy_loss         | -0.903      |
|    explained_variance   | 0.988       |
|    learning_rate        | 0.0003      |
|    loss                 | 1.16        |
|    n_updates            | 200         |
|    policy_gradient_loss | -0.000512   |
|    value_loss           | 14.4        |
-----------------------------------------
-------------------------------------
| rollout/                |           |
|    ep_len_mean          | 971       |
|    ep_rew_mean          | 143       |
| time/                   |           |
|    fps                  | 779       |
|    iterations           | 52        |
|    time_elapsed         | 1093      |
|    total_timesteps      | 851968    |
| train/                  |           |
|    approx_kl            | 0.0044502 |
|    clip_fraction        | 0.0415    |
|    clip_range           | 0.2       |
|    entropy_loss         | -0.878    |
|    explained_variance   | 0.998     |
|    learning_rate        | 0.0003    |
|    loss                 | 0.71      |
|    n_updates            | 204       |
|    policy_gradient_loss | -0.000632 |
|    value_loss           | 1.83      |
-------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 942         |
|    ep_rew_mean          | 141         |
| time/                   |             |
|    fps                  | 778         |
|    iterations           | 53          |
|    time_elapsed         | 1115        |
|    total_timesteps      | 868352      |
| train/                  |             |
|    approx_kl            | 0.0034386532 |
|    clip_fraction        | 0.0164      |
|    clip_range           | 0.2         |
|    entropy_loss         | -0.867      |
|    explained_variance   | 0.974       |
```

```
|    learning_rate        | 0.0003       |
|    loss                 | 13.9         |
|    n_updates            | 208          |
|    policy_gradient_loss | -0.000543    |
|    value_loss           | 34.8         |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 935          |
|    ep_rew_mean          | 142          |
| time/                   |              |
|    fps                  | 776          |
|    iterations           | 54           |
|    time_elapsed         | 1138         |
|    total_timesteps      | 884736       |
| train/                  |              |
|    approx_kl            | 0.0028421837 |
|    clip_fraction        | 0.0172       |
|    clip_range           | 0.2          |
|    entropy_loss         | -0.841       |
|    explained_variance   | 0.964        |
|    learning_rate        | 0.0003       |
|    loss                 | 4.44         |
|    n_updates            | 212          |
|    policy_gradient_loss | -0.000334    |
|    value_loss           | 55.6         |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 897          |
|    ep_rew_mean          | 138          |
| time/                   |              |
|    fps                  | 776          |
|    iterations           | 55           |
|    time_elapsed         | 1160         |
|    total_timesteps      | 901120       |
| train/                  |              |
|    approx_kl            | 0.0049287863 |
|    clip_fraction        | 0.0509       |
|    clip_range           | 0.2          |
|    entropy_loss         | -0.836       |
|    explained_variance   | 0.989        |
|    learning_rate        | 0.0003       |
|    loss                 | 1.5          |
|    n_updates            | 216          |
|    policy_gradient_loss | -0.000409    |
|    value_loss           | 16.1         |
-----------------------------------------
```

```
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 867         |
|    ep_rew_mean          | 137         |
| time/                   |             |
|    fps                  | 775         |
|    iterations           | 56          |
|    time_elapsed         | 1183        |
|    total_timesteps      | 917504      |
| train/                  |             |
|    approx_kl            | 0.0024348623 |
|    clip_fraction        | 0.0192      |
|    clip_range           | 0.2         |
|    entropy_loss         | -0.828      |
|    explained_variance   | 0.964       |
|    learning_rate        | 0.0003      |
|    loss                 | 1.73        |
|    n_updates            | 220         |
|    policy_gradient_loss | -9.98e-05   |
|    value_loss           | 46.9        |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 842         |
|    ep_rew_mean          | 139         |
| time/                   |             |
|    fps                  | 775         |
|    iterations           | 57          |
|    time_elapsed         | 1204        |
|    total_timesteps      | 933888      |
| train/                  |             |
|    approx_kl            | 0.004851416 |
|    clip_fraction        | 0.0385      |
|    clip_range           | 0.2         |
|    entropy_loss         | -0.844      |
|    explained_variance   | 0.957       |
|    learning_rate        | 0.0003      |
|    loss                 | 40.2        |
|    n_updates            | 224         |
|    policy_gradient_loss | -0.000544   |
|    value_loss           | 68.5        |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 810         |
|    ep_rew_mean          | 142         |
| time/                   |             |
|    fps                  | 775         |
```

```
|    iterations           | 58           |
|    time_elapsed         | 1225         |
|    total_timesteps      | 950272       |
| train/                  |              |
|    approx_kl            | 0.0043038856 |
|    clip_fraction        | 0.0455       |
|    clip_range           | 0.2          |
|    entropy_loss         | -0.788       |
|    explained_variance   | 0.936        |
|    learning_rate        | 0.0003       |
|    loss                 | 52.5         |
|    n_updates            | 228          |
|    policy_gradient_loss | -0.000662    |
|    value_loss           | 99.8         |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 720          |
|    ep_rew_mean          | 153          |
| time/                   |              |
|    fps                  | 776          |
|    iterations           | 59           |
|    time_elapsed         | 1244         |
|    total_timesteps      | 966656       |
| train/                  |              |
|    approx_kl            | 0.006026829  |
|    clip_fraction        | 0.0867       |
|    clip_range           | 0.2          |
|    entropy_loss         | -0.716       |
|    explained_variance   | 0.866        |
|    learning_rate        | 0.0003       |
|    loss                 | 24.9         |
|    n_updates            | 232          |
|    policy_gradient_loss | -0.0025      |
|    value_loss           | 197          |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 608          |
|    ep_rew_mean          | 175          |
| time/                   |              |
|    fps                  | 778          |
|    iterations           | 60           |
|    time_elapsed         | 1262         |
|    total_timesteps      | 983040       |
| train/                  |              |
|    approx_kl            | 0.004601596  |
|    clip_fraction        | 0.0711       |
```

```
|    clip_range           | 0.2         |
|    entropy_loss         | -0.696      |
|    explained_variance   | 0.814       |
|    learning_rate        | 0.0003      |
|    loss                 | 188         |
|    n_updates            | 236         |
|    policy_gradient_loss | -0.00272    |
|    value_loss           | 317         |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 481         |
|    ep_rew_mean          | 207         |
| time/                   |             |
|    fps                  | 781         |
|    iterations           | 61          |
|    time_elapsed         | 1278        |
|    total_timesteps      | 999424      |
| train/                  |             |
|    approx_kl            | 0.006298151 |
|    clip_fraction        | 0.0434      |
|    clip_range           | 0.2         |
|    entropy_loss         | -0.763      |
|    explained_variance   | 0.779       |
|    learning_rate        | 0.0003      |
|    loss                 | 124         |
|    n_updates            | 240         |
|    policy_gradient_loss | -0.00223    |
|    value_loss           | 341         |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 405          |
|    ep_rew_mean          | 216          |
| time/                   |              |
|    fps                  | 784          |
|    iterations           | 62           |
|    time_elapsed         | 1294         |
|    total_timesteps      | 1015808      |
| train/                  |              |
|    approx_kl            | 0.0059260186 |
|    clip_fraction        | 0.0571       |
|    clip_range           | 0.2          |
|    entropy_loss         | -0.745       |
|    explained_variance   | 0.702        |
|    learning_rate        | 0.0003       |
|    loss                 | 79.2         |
|    n_updates            | 244          |
```

```
|    policy_gradient_loss | -0.00215     |
|    value_loss           | 322          |
-----------------------------------------
```

2.4 Evaluate the agent

- Remember to wrap the environment in a Monitor.
- Now that our Lunar Lander agent is trained , we need to **check its performance**.
- Stable-Baselines3 provides a method to do that: `evaluate_policy`.
- To fill that part you need to check the documentation
- In the next step, we'll see **how to automatically evaluate and share your agent to compete in a leaderboard, but for now let's do it ourselves**

When you evaluate your agent, you should not use your training environment but create an evaluation environment.

[8]:
```
#@title
eval_env = Monitor(gym.make("LunarLander-v2"))
mean_reward, std_reward = evaluate_policy(model, eval_env, n_eval_episodes=10,␣
↪deterministic=True)
print(f"mean_reward={mean_reward:.2f} +/- {std_reward}")
```

```
mean_reward=257.92 +/- 19.23536582855511
```

[9]:
```
notebook_login()
!git config --global credential.helper store
```

```
VBox(children=(HTML(value='<center> <img\nsrc=https://huggingface.co/front/
↪assets/huggingface_logo-noborder.sv…
```

Let's fill the `package_to_hub` function: - `model`: our trained model. - `model_name`: the name of the trained model that we defined in `model_save` - `model_architecture`: the model architecture we used, in our case PPO - `env_id`: the name of the environment, in our case `LunarLander-v2` - `eval_env`: the evaluation environment defined in eval_env - `repo_id`: the name of the Hugging Face Hub Repository that will be created/updated `(repo_id = {username}/{repo_name})`

A good name is {username}/{model_architecture}-{env_id}

- `commit_message`: message of the commit

[10]:
```
import gymnasium as gym

from stable_baselines3 import PPO
from stable_baselines3.common.vec_env import DummyVecEnv
from stable_baselines3.common.env_util import make_vec_env

from huggingface_sb3 import package_to_hub

# PLACE the variables you've just defined two cells above
# Define the name of the environment
```

```
env_id = "LunarLander-v2"

# TODO: Define the model architecture we used
model_architecture = "PPO"

## Define a repo_id
## repo_id is the id of the model repository from the Hugging Face Hub (repo_id
 ↪= {organization}/{repo_name} for instance ThomasSimonini/ppo-LunarLander-v2
## CHANGE WITH YOUR REPO ID
repo_id = "Andyrasika/ppo-LunarLander-v2" # Change with your repo id, you can't
 ↪push with mine

## Define the commit message
commit_message = "Upload PPO LunarLander-v2 trained agent"

# Create the evaluation env and set the render_mode="rgb_array"
eval_env = DummyVecEnv([lambda: gym.make(env_id, render_mode="rgb_array")])

# PLACE the package_to_hub function you've just filled here
package_to_hub(model=model, # Our trained model
               model_name=model_name, # The name of our trained model
               model_architecture=model_architecture, # The model architecture
 ↪we used: in our case PPO
               env_id=env_id, # Name of the environment
               eval_env=eval_env, # Evaluation Environment
               repo_id=repo_id, # id of the model repository from the Hugging
 ↪Face Hub (repo_id = {organization}/{repo_name} for instance ThomasSimonini/
 ↪ppo-LunarLander-v2
               commit_message=commit_message)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283:
DeprecationWarning: `should_run_async` will not call `transform_cell`
automatically in the future. Please pass the result to `transformed_cell`
argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

```
 This function will save, evaluate, generate a video of your agent,
create a model card and push everything to the hub. It might take up to 1min.
This is a work in progress: if you encounter a bug, please open an issue.
```

```
/usr/local/lib/python3.10/dist-
packages/stable_baselines3/common/evaluation.py:67: UserWarning: Evaluation
environment is not wrapped with a ``Monitor`` wrapper. This may result in
reporting modified episode lengths and rewards, if other wrappers happen to
modify these. Consider wrapping environment first with ``Monitor`` wrapper.
  warnings.warn(
```

```
Saving video to /tmp/tmpfqzb97d6/-step-0-to-step-1000.mp4
Moviepy - Building video /tmp/tmpfqzb97d6/-step-0-to-step-1000.mp4.
Moviepy - Writing video /tmp/tmpfqzb97d6/-step-0-to-step-1000.mp4



Moviepy - Done !
Moviepy - video ready /tmp/tmpfqzb97d6/-step-0-to-step-1000.mp4
  Pushing repo Andyrasika/ppo-LunarLander-v2 to the Hugging Face
Hub
Upload 4 LFS files:   0%|          | 0/4 [00:00<?, ?it/s]
pytorch_variables.pth:   0%|          | 0.00/431 [00:00<?, ?B/s]
policy.pth:   0%|          | 0.00/43.3k [00:00<?, ?B/s]
policy.optimizer.pth:   0%|          | 0.00/87.9k [00:00<?, ?B/s]
ppo-LunarLander-v2.zip:   0%|          | 0.00/147k [00:00<?, ?B/s]
  Your model is pushed to the Hub. You can view your model here:
https://huggingface.co/Andyrasika/ppo-LunarLander-v2/tree/main/
```

[10]:
```
'https://huggingface.co/Andyrasika/ppo-LunarLander-v2/tree/main/'
```

2.5 Load a saved LunarLander model from the Hub

Thanks to ironbar for the contribution.

Loading a saved model from the Hub is really easy.

You go to https://huggingface.co/models?library=stable-baselines3 to see the list of all the Stable-baselines3 saved models. 1. You select one and copy its repo_id

[11]:
```
!pip install -Uqqq shimmy
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283:
DeprecationWarning: `should_run_async` will not call `transform_cell`
automatically in the future. Please pass the result to `transformed_cell`
argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

[12]:
```
from huggingface_sb3 import load_from_hub
repo_id = "Classroom-workshop/assignment2-omar" # The repo_id
filename = "ppo-LunarLander-v2.zip" # The model filename.zip

# When the model was trained on Python 3.8 the pickle protocol is 5
# But Python 3.6, 3.7 use protocol 4
# In order to get compatibility we need to:
```

```
# 1. Install pickle5 (we done it at the beginning of the colab)
# 2. Create a custom empty object we pass as parameter to PPO.load()
custom_objects = {
            "learning_rate": 0.0,
            "lr_schedule": lambda _: 0.0,
            "clip_range": lambda _: 0.0,
}

checkpoint = load_from_hub(repo_id, filename)
model = PPO.load(checkpoint, custom_objects=custom_objects,␣
 ↪print_system_info=True)
```

```
Downloading ppo-LunarLander-v2.zip:   0%|          | 0.00/146k [00:00<?, ?B/s]
```

```
== CURRENT SYSTEM INFO ==
- OS: Linux-5.15.109+-x86_64-with-glibc2.35 # 1 SMP Fri Jun 9 10:57:30 UTC 2023
- Python: 3.10.12
- Stable-Baselines3: 2.0.0a5
- PyTorch: 2.0.1+cu118
- GPU Enabled: True
- Numpy: 1.23.5
- Cloudpickle: 2.2.1
- Gymnasium: 0.28.1
- OpenAI Gym: 0.25.2

== SAVED MODEL SYSTEM INFO ==
OS: Linux-5.4.188+-x86_64-with-Ubuntu-18.04-bionic #1 SMP Sun Apr 24 10:03:06
PDT 2022
Python: 3.7.13
Stable-Baselines3: 1.5.0
PyTorch: 1.11.0+cu113
GPU Enabled: True
Numpy: 1.21.6
Gym: 0.21.0


/usr/local/lib/python3.10/dist-
packages/stable_baselines3/common/vec_env/patch_gym.py:95: UserWarning: You
loaded a model that was trained using OpenAI Gym. We strongly recommend
transitioning to Gymnasium by saving that model again.
  warnings.warn(
```

[13]:
```
#@title
eval_env = Monitor(gym.make("LunarLander-v2"))
mean_reward, std_reward = evaluate_policy(model, eval_env, n_eval_episodes=10,␣
 ↪deterministic=True)
print(f"mean_reward={mean_reward:.2f} +/- {std_reward}")
```

```
mean_reward=297.34 +/- 15.738909153404379
```

2.6 Some additional challenges

The best way to learn **is to try things by your own**! As you saw, the current agent is not doing great. As a first suggestion, you can train for more steps. With 1,000,000 steps, we saw some great results!

In the Leaderboard you will find your agents. Can you get to the top?

Here are some ideas to achieve so: * Train more steps * Try different hyperparameters for `PPO`. You can see them at https://stable-baselines3.readthedocs.io/en/master/modules/ppo.html#parameters. * Check the Stable-Baselines3 documentation and try another model such as DQN. * **Push your new trained model** on the Hub

Compare the results of your LunarLander-v2 with your classmates using the leaderboard

Is moon landing too boring for you? Try to **change the environment**, why not use MountainCar-v0, CartPole-v1 or CarRacing-v0? Check how they work using the gym documentation and have fun .

[ ]: