Project Containers

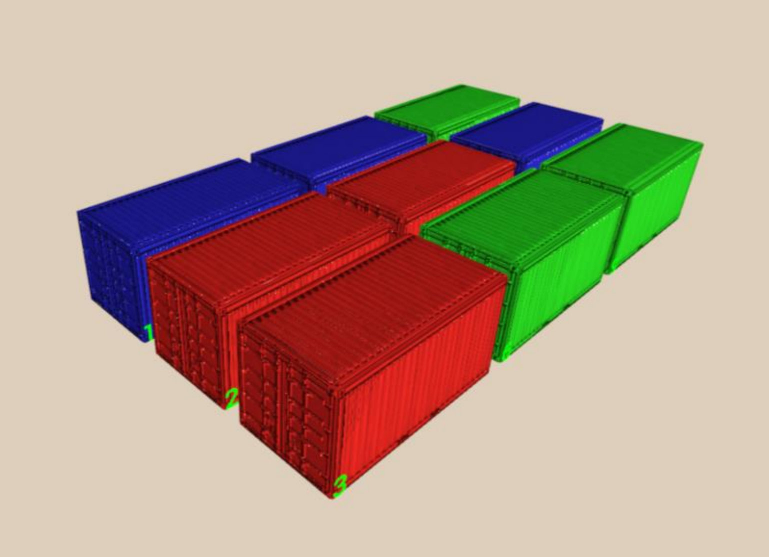
Groep 1

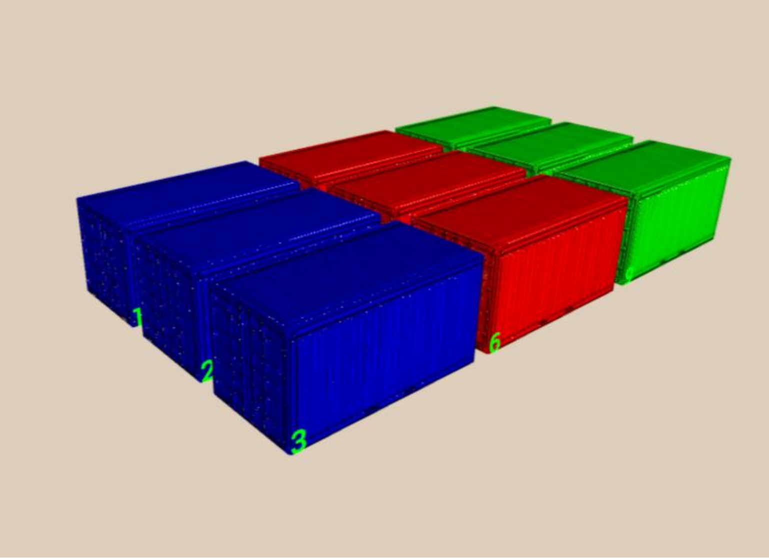
Week 4: 05-12-2022

Herhaling beginprobleem

- 3 bij 3 in 2D
- 3 per schip
- Vanaf beide kanten
- Optimale oplossing

- Per container



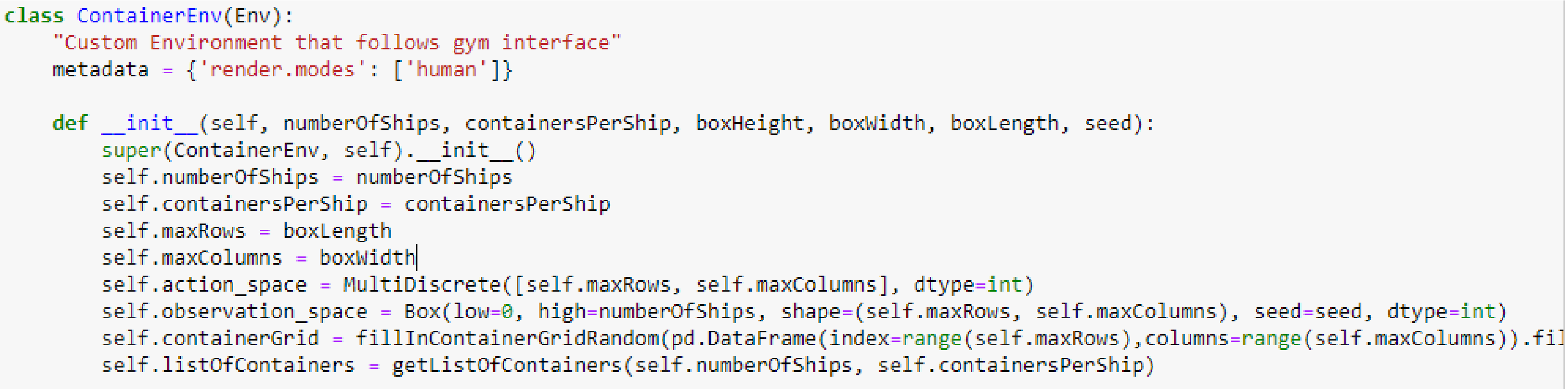


Opsplitsen

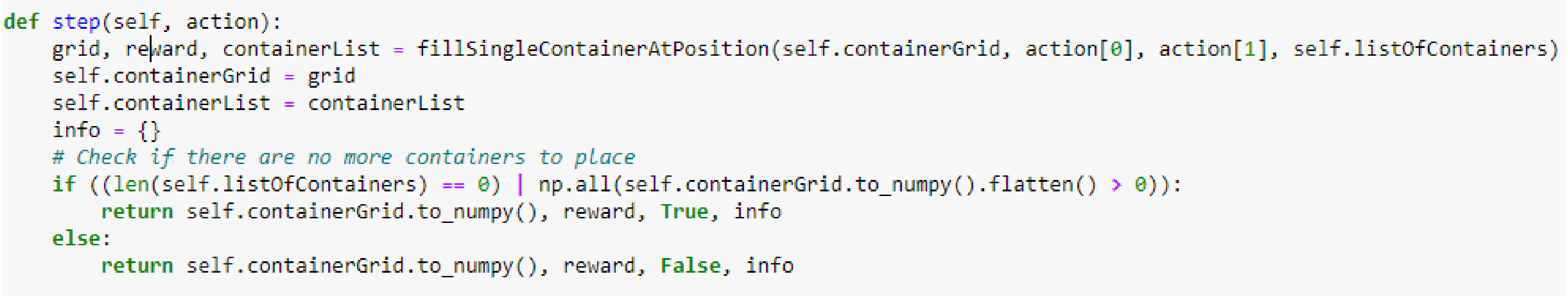
- Apart werken aan modellen
- 3 bij 3
- Rewardfunctie
- Environment
- Trainen

Environment

- Wat houden we bij
- Wat is de stap



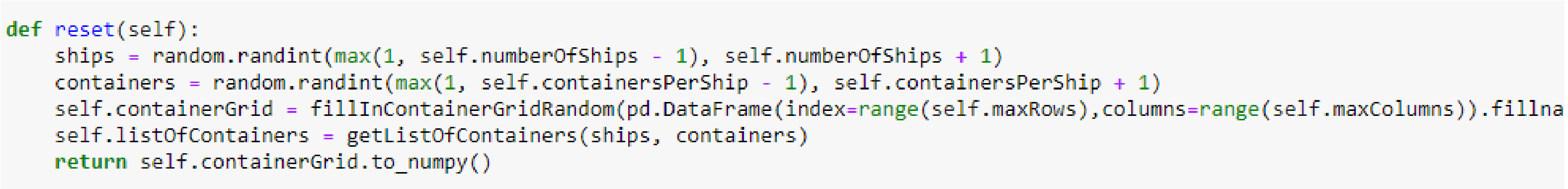
```
class ContainerEnv(Env):
    "Custom Environment that follows gym interface"
    metadata = {'render.modes': ['human']}

    def __init__(self, numberOfShips, containersPerShip, boxHeight, boxWidth, boxLength, seed):
        super(ContainerEnv, self).__init__()
        self.numberOfShips = numberOfShips
        self.containersPerShip = containersPerShip
        self.maxRows = boxLength
        self.maxColumns = boxWidth
        self.action_space = MultiDiscrete([self.maxRows, self.maxColumns], dtype=int)
        self.observation_space = Box(low=0, high=numberOfShips, shape=(self.maxRows, self.maxColumns), seed=seed, dtype=int)
        self.containerGrid = fillInContainerGridRandom(pd.DataFrame(index=range(self.maxRows),columns=range(self.maxColumns)).fil
        self.listOfContainers = getListOfContainers(self.numberOfShips, self.containersPerShip)
```



```
def step(self, action):
    grid, reward, containerList = fillSingleContainerAtPosition(self.containerGrid, action[0], action[1], self.listOfContainers)
    self.containerGrid = grid
    self.containerList = containerList
    info = {}
    # Check if there are no more containers to place
    if ((len(self.listOfContainers) == 0) | np.all(self.containerGrid.to_numpy().flatten() > 0)):
        return self.containerGrid.to_numpy(), reward, True, info
    else:
        return self.containerGrid.to_numpy(), reward, False, info
```

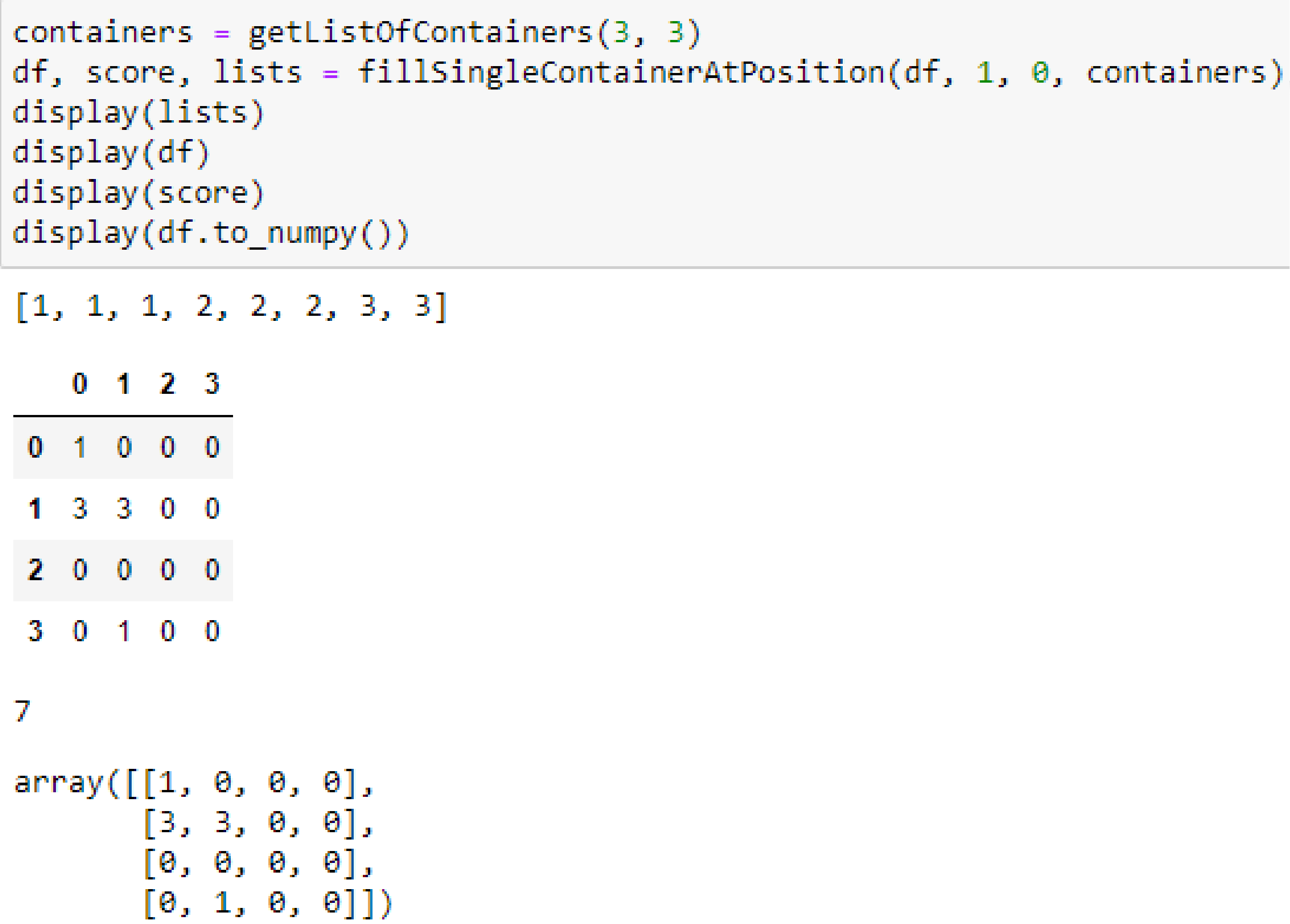
Generaliseren

- Problemen met een grid groter dan 3 bij 3
  - Overfitten
  - Moeite met andere soort lijst van containers
  - Moeite met andere lengte van lijst van containers
- List randomizen
- Grid randomizen



```
def reset(self):
    ships = random.randint(max(1, self.numberOfShips - 1), self.numberOfShips + 1)
    containers = random.randint(max(1, self.containersPerShip - 1), self.containersPerShip + 1)
    self.containerGrid = fillInContainerGridRandom(pd.DataFrame(index=range(self.maxRows),columns=range(self.maxColumns)).fillna
    self.listOfContainers = getListOfContainers(ships, containers)
    return self.containerGrid.to_numpy()
```

Generaliseren voorbeeld



```
containers = getListOfContainers(3, 3)
df, score, lists = fillSingleContainerAtPosition(df, 1, 0, containers)
display(lists)
display(df)
display(score)
display(df.to_numpy())
```

```
[1, 1, 1, 2, 2, 2, 3, 3]
```

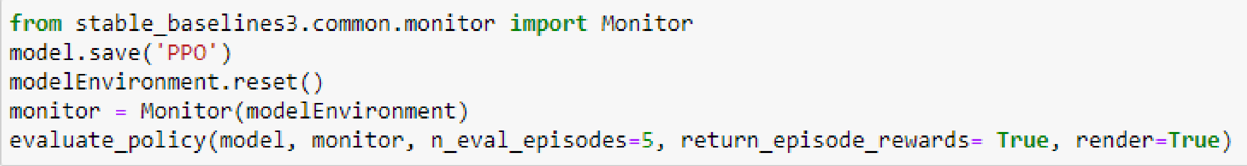
	0	1	2	3
0	1	0	0	0
1	3	3	0	0
2	0	0	0	0
3	0	1	0	0

```
7
```

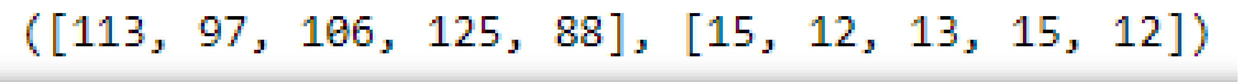
```
array([[1, 0, 0, 0],
       [3, 3, 0, 0],
       [0, 0, 0, 0],
       [0, 1, 0, 0]])
```

Stable baselines 3 trainen

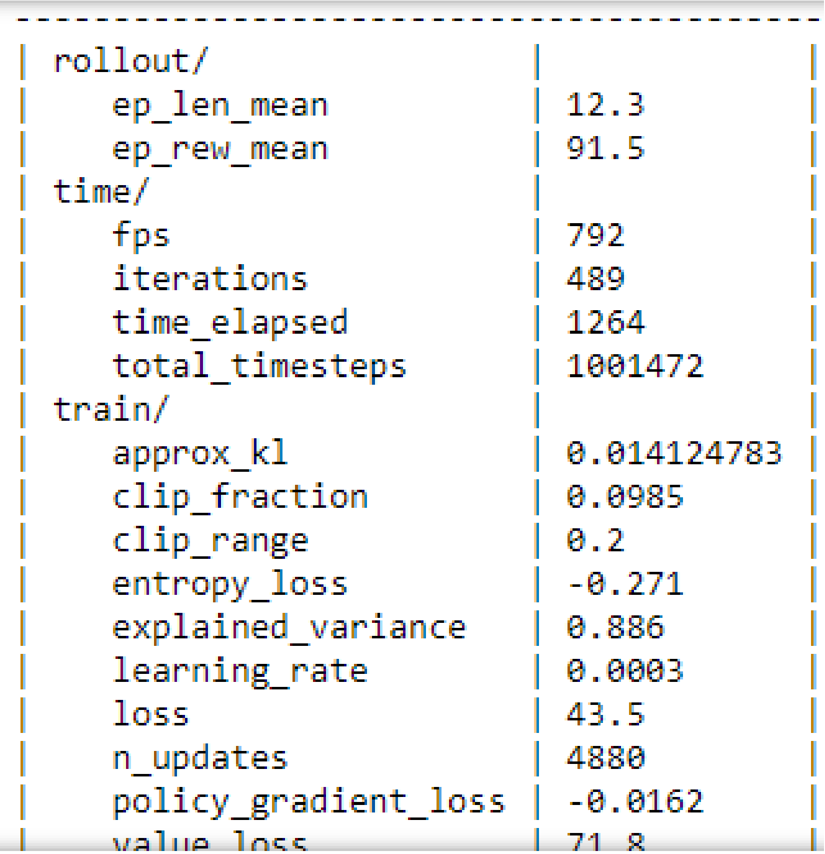
- PPO
- 1 miljoen iteraties in ~20 minuten



```
from stable_baselines3.common.monitor import Monitor
model.save('PPO')
modelEnvironment.reset()
monitor = Monitor(modelEnvironment)
evaluate_policy(model, monitor, n_eval_episodes=5, return_episode_rewards= True, render=True)
```



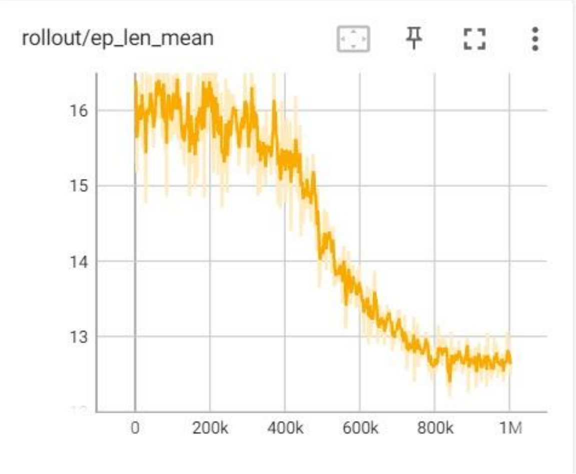
```
([113, 97, 106, 125, 88], [15, 12, 13, 15, 12])
```

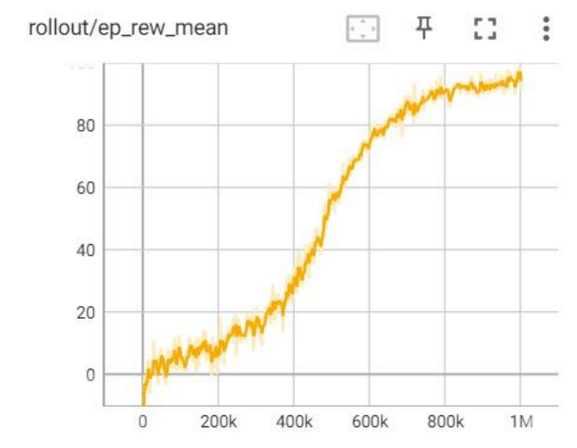


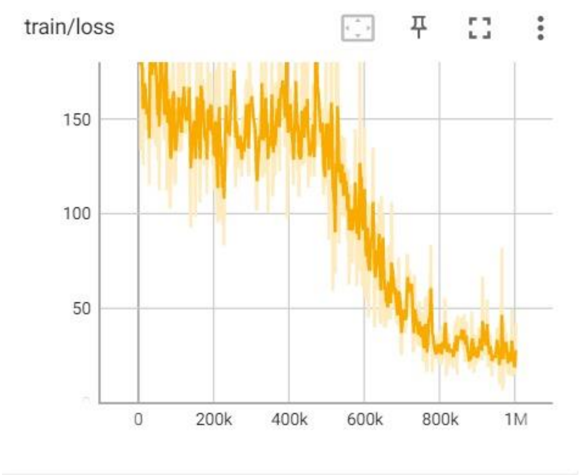
```
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 12.3        |
|    ep_rew_mean          | 91.5        |
| time/                   |             |
|    fps                  | 792         |
|    iterations           | 489         |
|    time_elapsed         | 1264        |
|    total_timesteps      | 1001472     |
| train/                  |             |
|    approx_kl            | 0.014124783 |
|    clip_fraction        | 0.0985      |
|    clip_range           | 0.2         |
|    entropy_loss         | -0.271      |
|    explained_variance   | 0.886       |
|    learning_rate        | 0.0003      |
|    loss                 | 43.5        |
|    n_updates            | 4880        |
|    policy_gradient_loss | -0.0162     |
|    value_loss           | 71.8        |
```

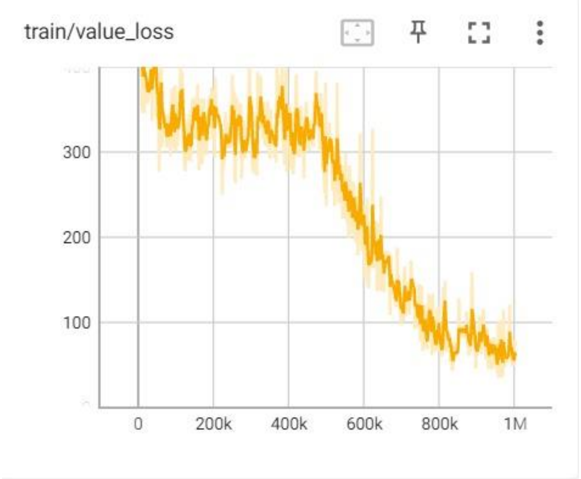
Grafieken van trainen

- X as -> iteraties

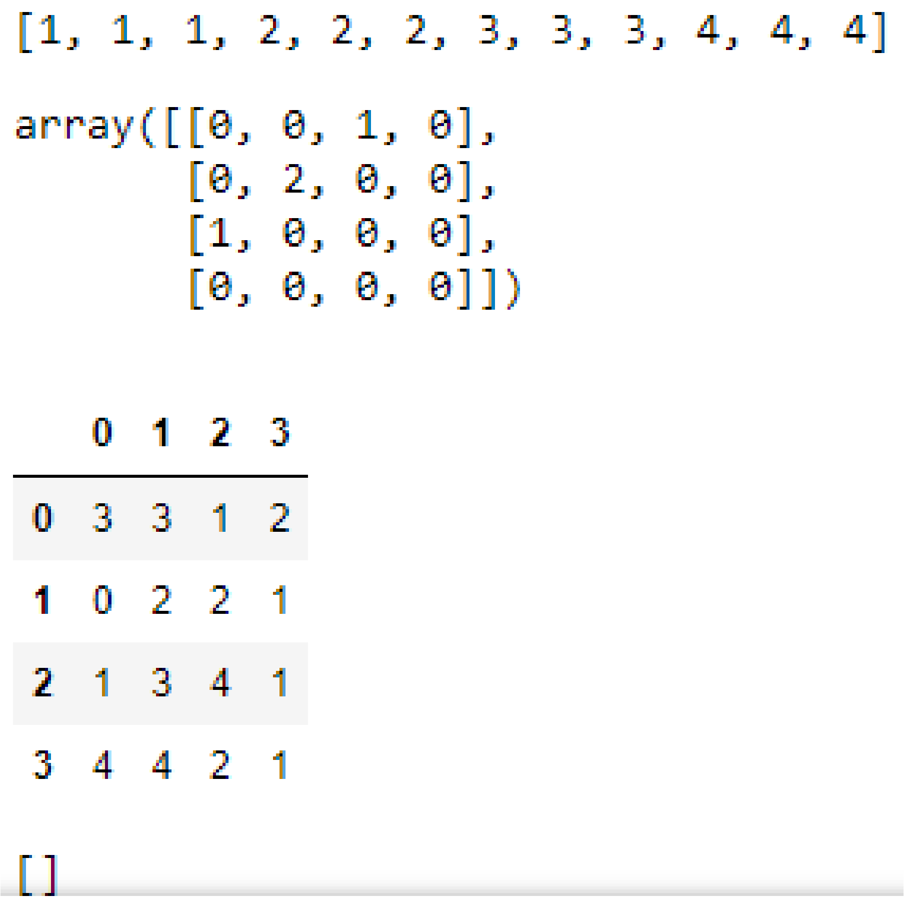








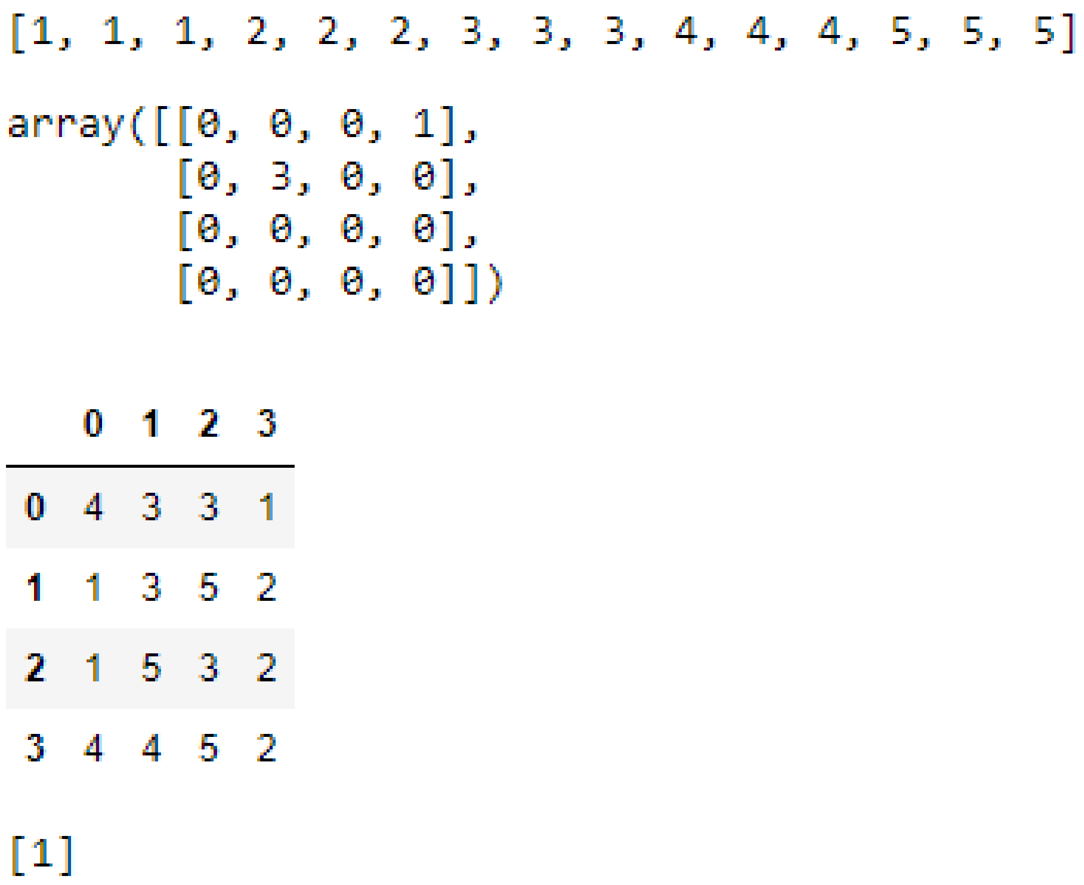
Resultaat van stable baselines 3



```
[1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4]

array([[0, 0, 1, 0],
       [0, 2, 0, 0],
       [1, 0, 0, 0],
       [0, 0, 0, 0]])
```

	0	1	2	3
0	3	3	1	2
1	0	2	2	1
2	1	3	4	1
3	4	4	2	1

```
[]
```



```
[1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 5]

array([[0, 0, 0, 1],
       [0, 3, 0, 0],
       [0, 0, 0, 0],
       [0, 0, 0, 0]])
```

	0	1	2	3
0	4	3	3	1
1	1	3	5	2
2	1	5	3	2
3	4	4	5	2

```
[1]
```

Resultaat van stable baselines 3

- Goede punten
  - Containers worden niet op elkaar gezet
  - Containers komen soort van bij elkaar in de buurt
- Verbeteringen
  - Containers bij elkaar zetten kan nog beter

Vragen/Tips?