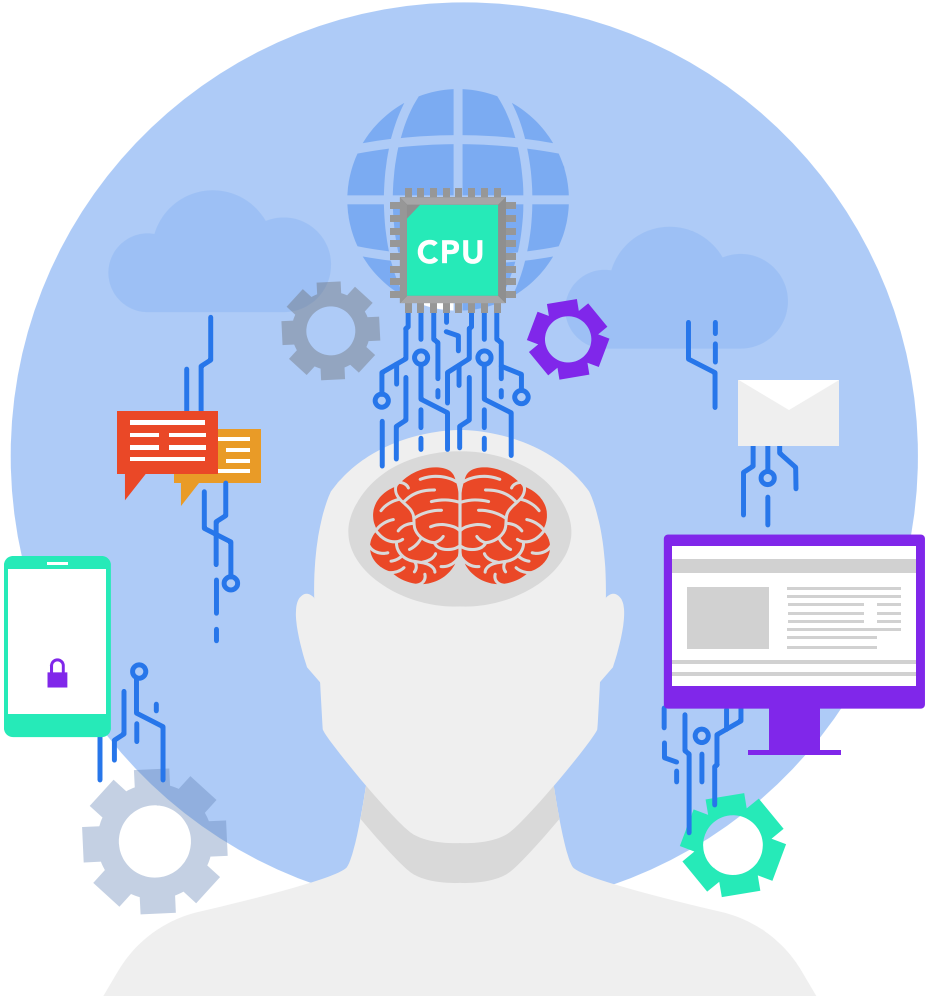
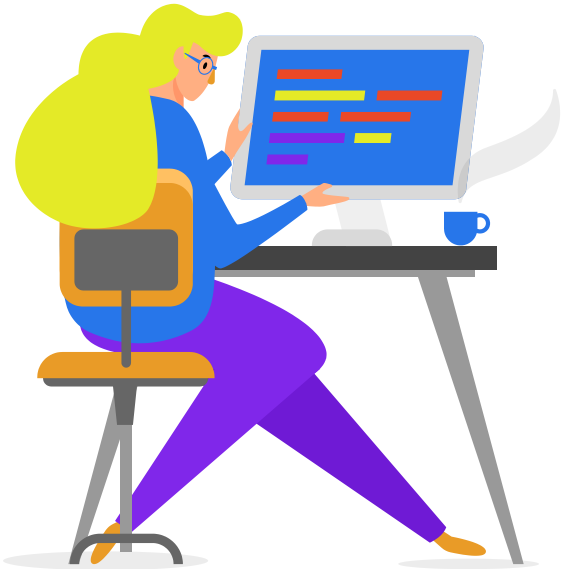


HUMAN MACHINE TEACHERS

JESSE, JOANNE, ERIC,
MARTTI, SEFA & AYRTON

VOORTGANG CONTAINER PROJECT



01 Recap

02 Simulatie Environment

03 Reinforcement Agent

04 Results

Recap: Onderzoeksvragen

Hoofdvraag

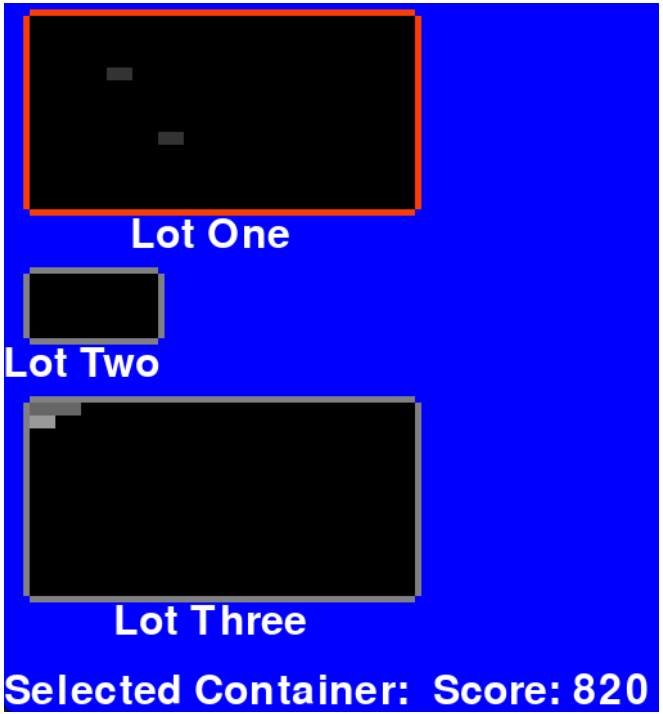
Met welke methode(s) kunnen we het uitladingsdeel van het container stacking probleem optimaal oplossen?

Deelvragen

1. Welke methoden (heuristieken) zijn mogelijk bij het container stacking probleem?
2. Wat is een move en wat zijn de restricties?
3. Welke containers zijn er en welke gaan we gebruiken?
4. Hoe is de haven ingericht en wat zijn de restricties?
5. Hoe kunnen we de container data simuleren?

Simulation Environments: Eerste prototype

- PyGame (GUI)
- Niet in Jupyter Notebook
- Meerdere lots
- Complex



Simulation Environments: Tweede prototype

- In Jupyter Notebook
- Makkelijker concept

```
Speelveld
```

```
array([[['0', '0'],
        ['0', '0'],
        ['0', '0']],

       [['0', '0'],
        ['0', '0'],
        ['0', '0']],

       [['0', '0'],
        ['0', '0'],
        ['0', '0']]], dtype='<U1')
```

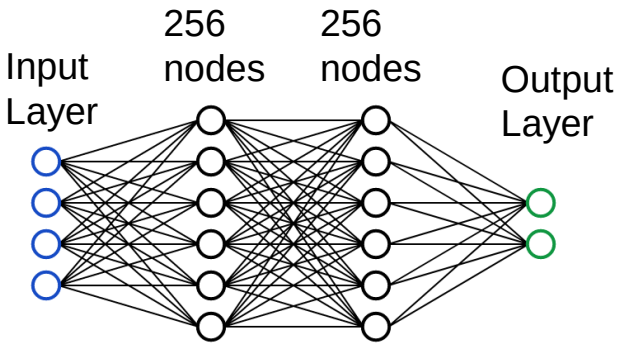
'0' = Leeg
'L' = Low Priority
'H' = High Priority

Reinforcement Agent: DeepQLearning

DeepQLearning classes:
- DeepQNetwork
- Agent

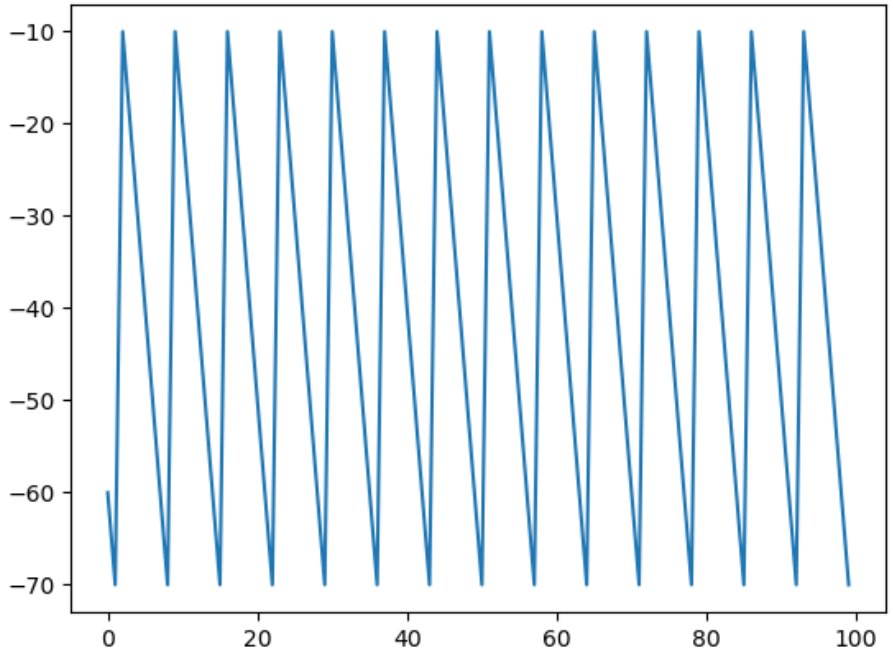
Key Agent Functions:
- choose_action(observation)
- store_transition(old observation, action, new_observation)
- learn()

Observation = Huidige status van het speelveld
Action = De zet die het neuraal netwerk maakt op het speelveld



Results

Positief = beter
Negatief = slechter
Feedback?



Vragen?