REINFORCEMENT LEARNING

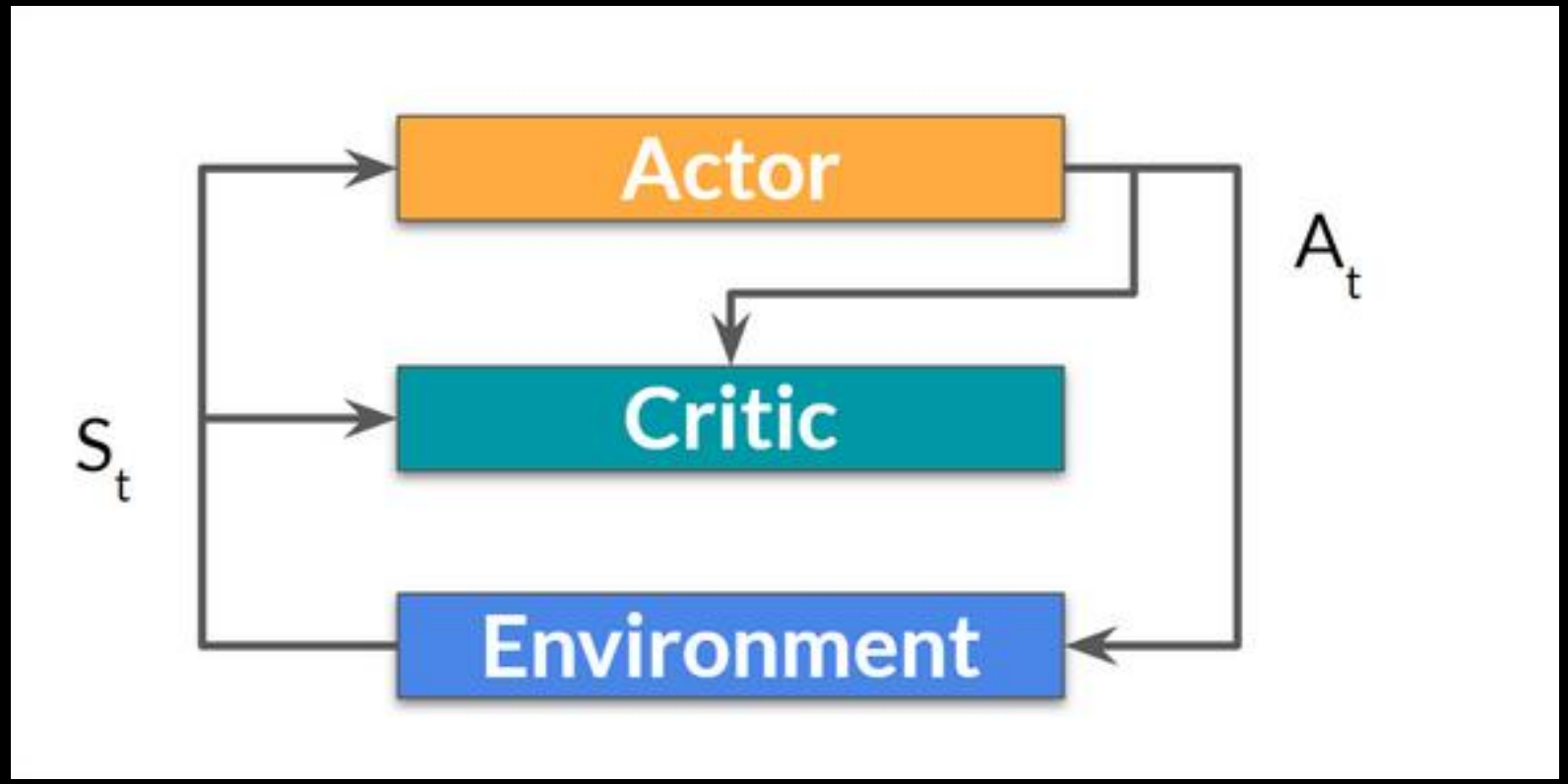
Aleks Kapich
Michał Matejczuk
Jakub Lange
Paweł Świderski
Mateusz Kubita

ADVANTAGE ACTOR-CRITIC (A2C)

$$L^{A2C}(\theta) = \hat{\mathbb{E}}_t \left[\log \pi_\theta (a_t \mid s_t) \hat{A}_t \right]$$

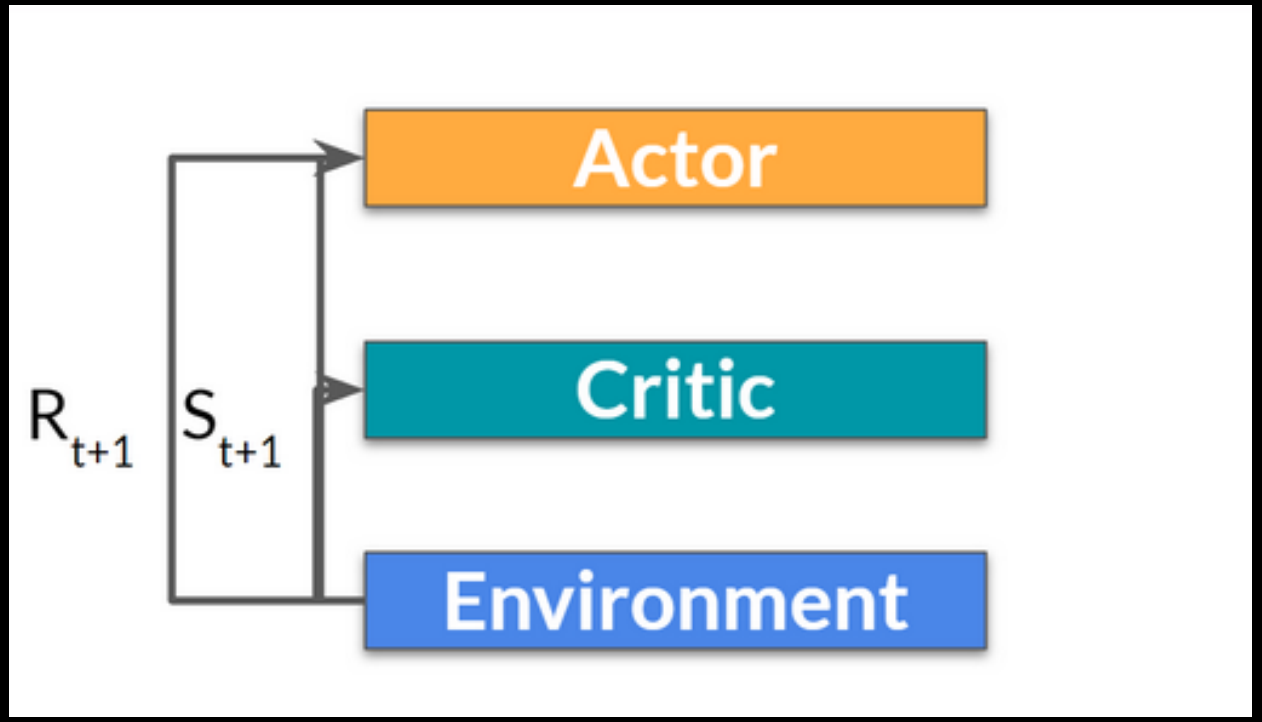
https://arxiv.org/pdf/2205.09123

ADVANTAGE ACTOR-CRITIC (A2C)



https://huggingface.co/learn/deep-rl-course/unit6/advantage-actor-critic

ADVANTAGE ACTOR-CRITIC (A2C)



https://huggingface.co/learn/deep-rl-course/unit6/advantage-actor-critic

ADVANTAGE ACTOR-CRITIC (A2C)

$$A(s,a) = Q(s,a) - V(s)$$

q value for action a in state s

average value of that state

https://huggingface.co/learn/deep-rl-course/unit6/advantage-actor-critic

PROXIMAL POLICY OPTIMIZATION (PPO)

$$L^{A2C}(\theta) = \hat{\mathbb{E}}_t \left[\log \pi_\theta (a_t \mid s_t) \hat{A}_t \right]$$

https://arxiv.org/pdf/2205.09123

PROXIMAL POLICY OPTIMIZATION (PPO)

$$L^{PPO}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(r_t(\theta) \hat{A}_t, \text{clip} \left(r_t(\theta), 1 - \epsilon, 1 + \epsilon \right) \hat{A}_t \right) \right]$$

$$\text{where } r_t(\theta) = \frac{\pi_\theta \left(a_t \mid s_t \right)}{\pi_{\theta_{\text{old}}} \left(a_t \mid s_t \right)} \hat{A}_t$$

https://arxiv.org/pdf/2205.09123

PROXIMAL POLICY OPTIMIZATION (PPO)

```
 git:(main) ✗ python sb3_a2c.py
mlp_extractor.policy_net.0.weight's sum = 3.9289
mlp_extractor.policy_net.0.bias's sum = 0.4128
mlp_extractor.policy_net.2.weight's sum = 2.2437
mlp_extractor.policy_net.2.bias's sum = -0.6634
mlp_extractor.value_net.0.weight's sum = -2.2411
mlp_extractor.value_net.0.bias's sum = -0.4382
mlp_extractor.value_net.2.weight's sum = -0.1973
mlp_extractor.value_net.2.bias's sum = -1.7232
action_net.weight's sum = -0.0139
action_net.bias's sum = -0.0
value_net.weight's sum = -2.1549
value net.bias's sum = 0.297
```

```
 git:(main) ✗ python sb3_ppo.py
mlp_extractor.policy_net.0.weight's sum = 3.9289
mlp_extractor.policy_net.0.bias's sum = 0.4128
mlp_extractor.policy_net.2.weight's sum = 2.2437
mlp_extractor.policy_net.2.bias's sum = -0.6634
mlp_extractor.value_net.0.weight's sum = -2.2411
mlp_extractor.value_net.0.bias's sum = -0.4382
mlp_extractor.value_net.2.weight's sum = -0.1973
mlp_extractor.value_net.2.bias's sum = -1.7232
action_net.weight's sum = -0.0139
action_net.bias's sum = -0.0
value_net.weight's sum = -2.1549
value net.bias's sum = 0.297
```

https://arxiv.org/pdf/2205.09123

BASIC

OPIS SCENARIUSZA

Jest podstawowym, bardzo okrojonym scenariuszem polegającym za zabiciu jednego potwora

Potwór pojawia się w losowym miejscu na szerokości ściany, pozostaje nieruchomy przez cały czas rozgrywki i nie atakuje gracza

Agent ma do dyspozycji jedynie 3 akcje: move_left, move_right oraz attack

Gra kończy się gdy gracz zabije potwora lub gdy minie 300 ticów

FAZA TRENINGU

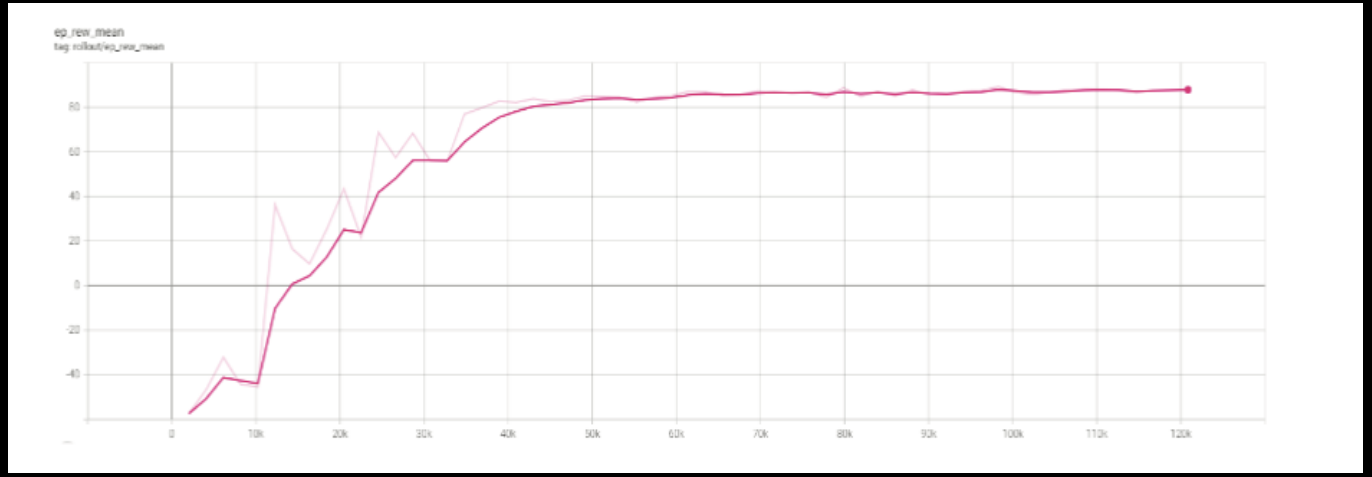
Funkcja nagrody

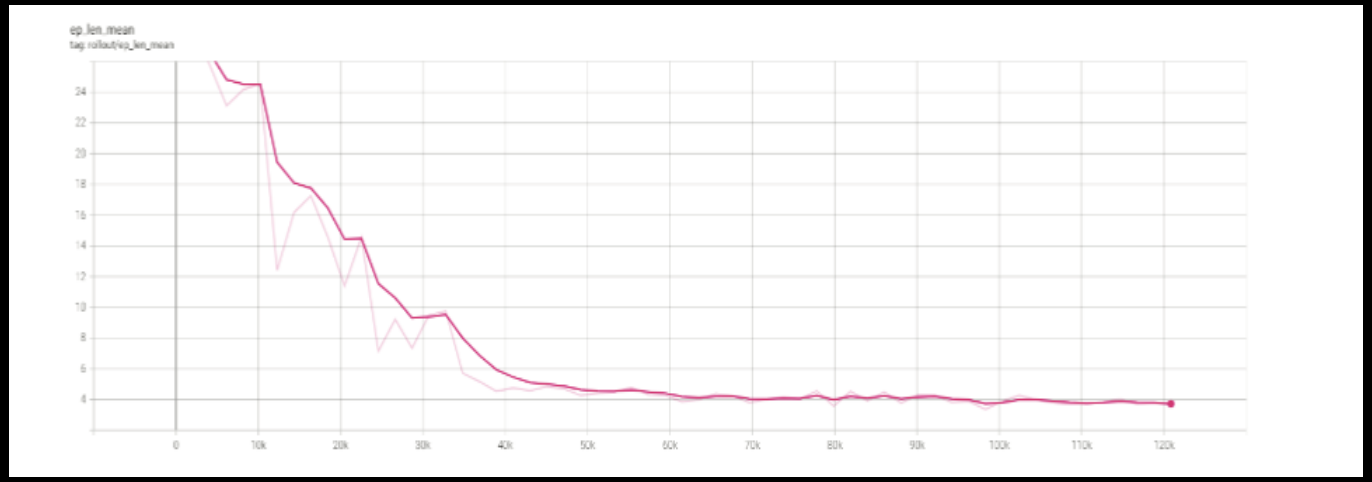
$$\begin{aligned}\textbf{Reward} = &\text{living\_reward} \\ &+ \text{kill\_reward} \\ &+ \text{ammo\_reward}\end{aligned}$$

Gdzie:

- living_reward = kara wartości -1 naliczana co 1 tic, podczas gdy gracz żyje
- kill_reward = nagroda wartości 106 za zabicie potwora (ukończenie gry)
- ammo_reward = kara wartości -5 za wystrzelenie pocisku.

FAZA TRENINGU

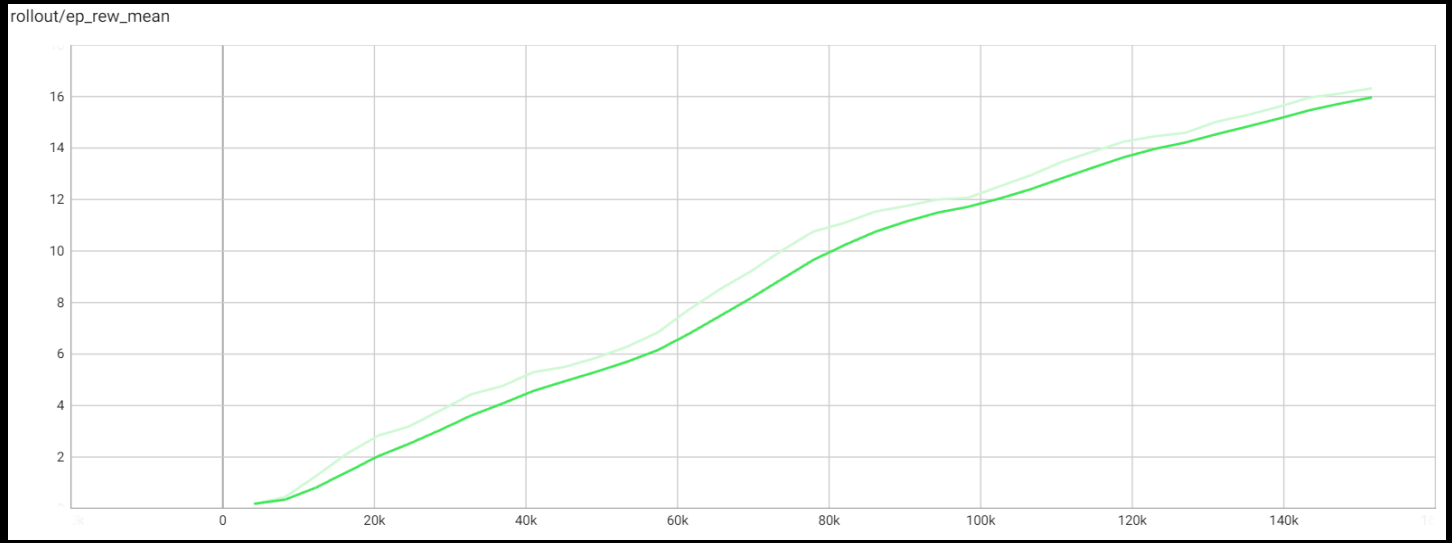


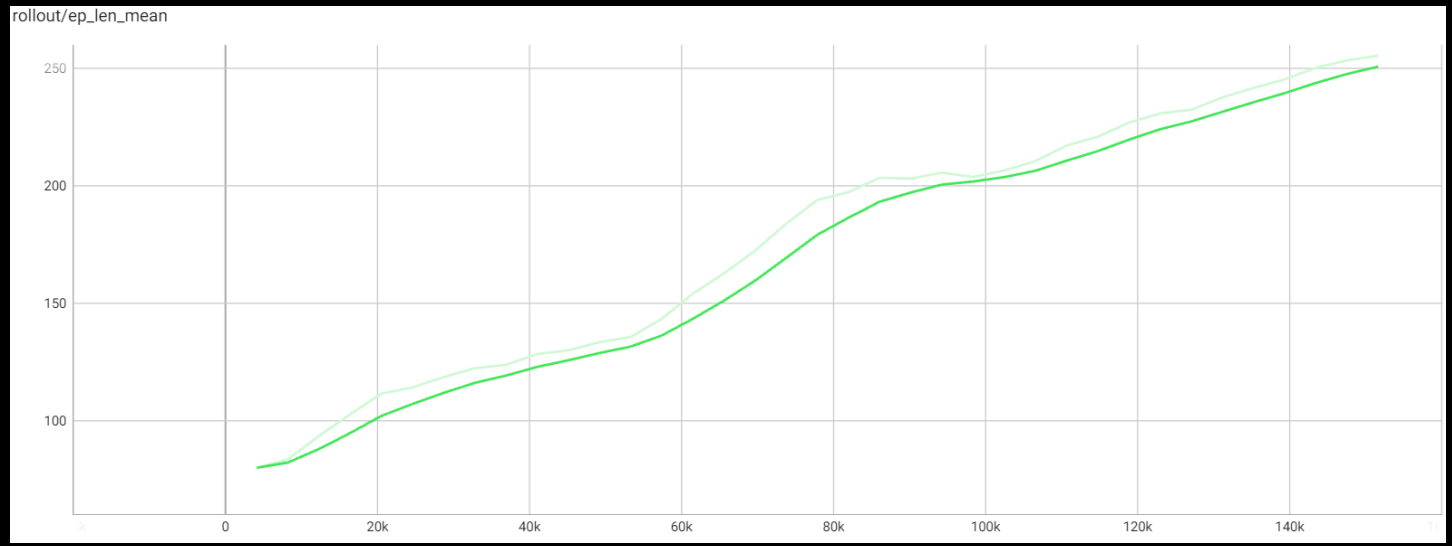


DEMONSTRACJA DZIAŁANIA AGENTA

DEFEND THE CENTER

DEFEND THE CENTER-TRENING





DEFEND THE CENTER-DEMONSTRACJA

DEADLY CORRIDOR

DEADLY CORRIDOR

OPIS SCENARIUSZA

Celem jest dojście do końca korytarza, przy czym na drodze pojawiają się demony.

Potwory pojawiają się zawsze tym samym miejscu.

Agent ma do dyspozycji **7 akcji**: move_left, move_right, attack, move_forward, move_backward, turn_left, turn_right

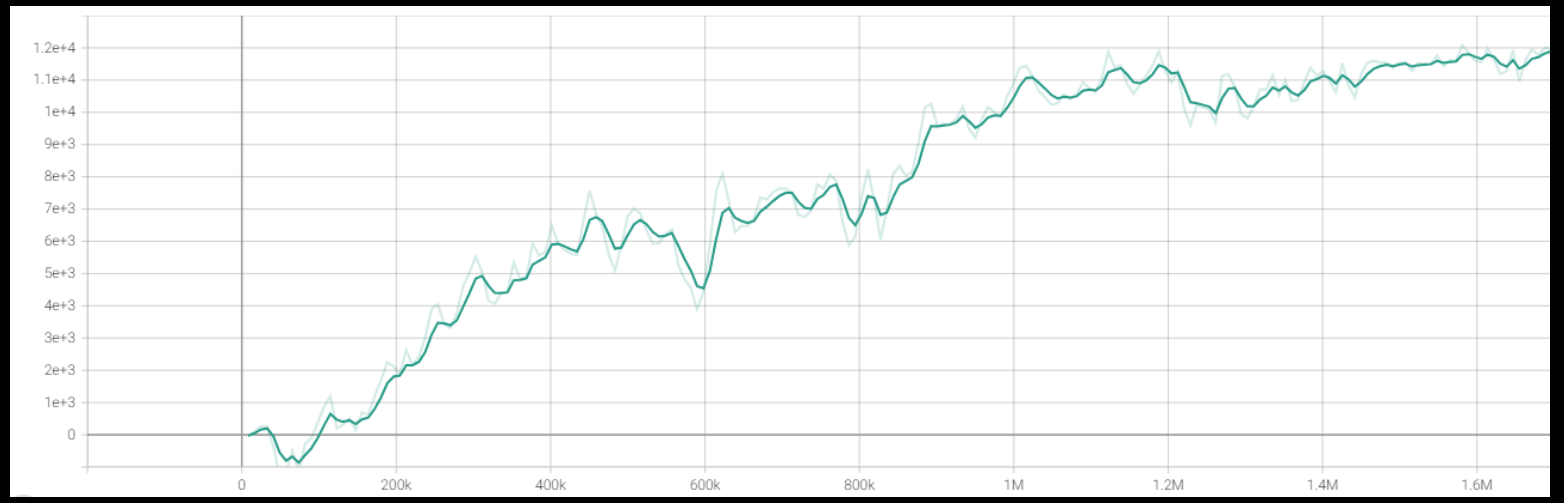
Gra kończy się gdy **gracz dojdzie do końca korytarza**

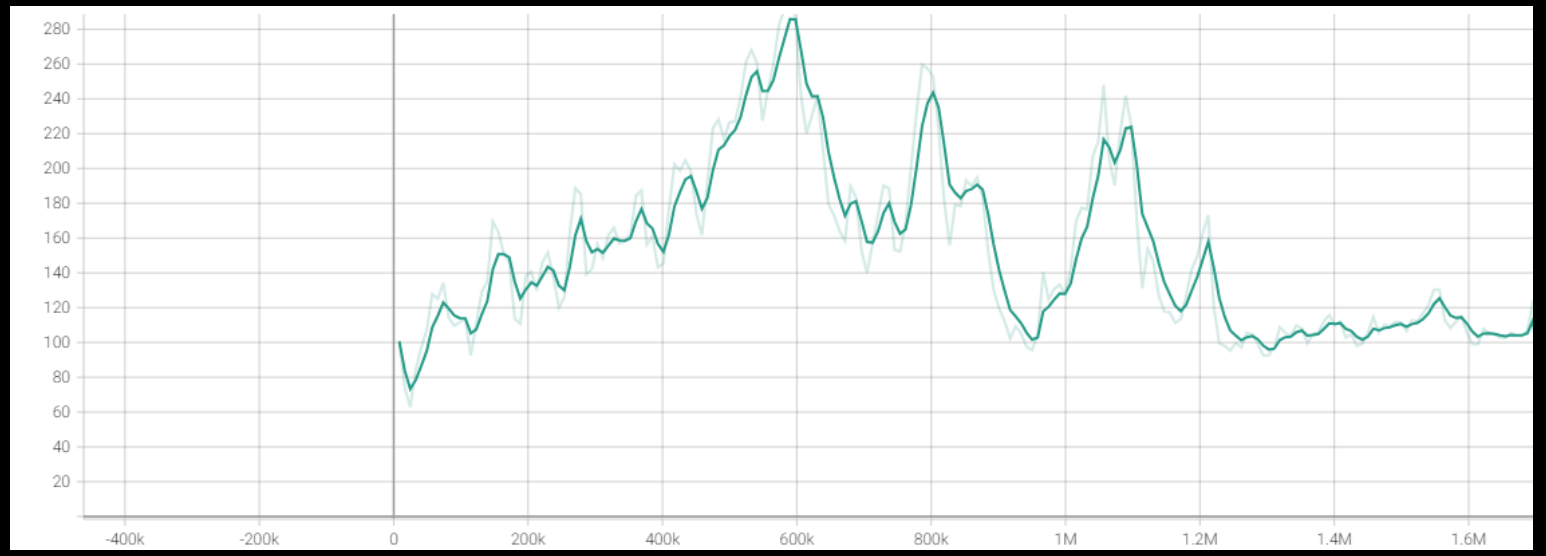
FAZA TRENINGU

Reward Function

$$\begin{aligned}\textbf{Reward} = \text{living\_reward} \\ &+ \text{movement\_reward} \\ &+ \text{damage\_taken\_delta} \times 10 \\ &+ \text{hitcount\_delta} \times 210 \\ &+ \text{ammo\_delta} \times 5 \\ &+ \text{camera\_reward}\end{aligned}$$

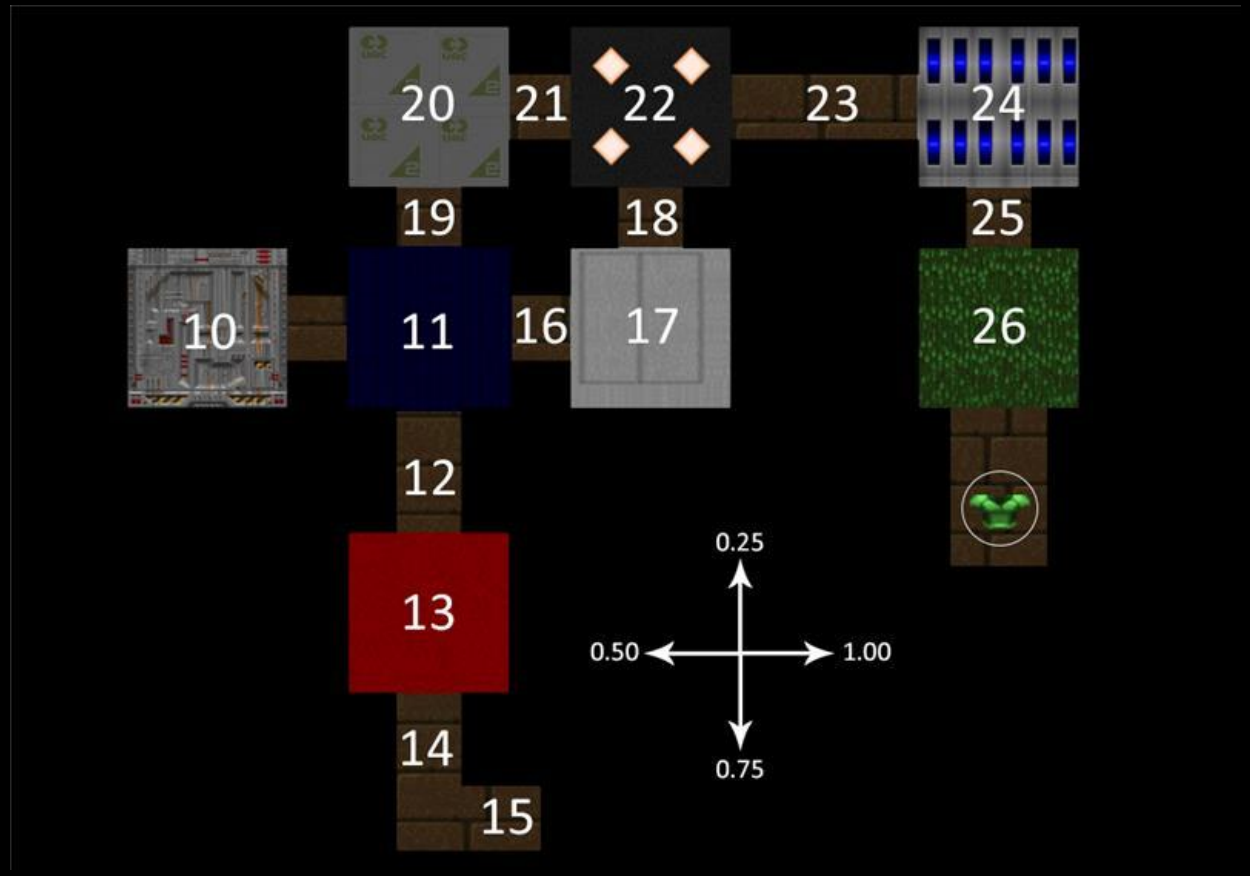
FAZA TRENINGU





DEMONSTRACJA AGENTA

MY WAY HOME

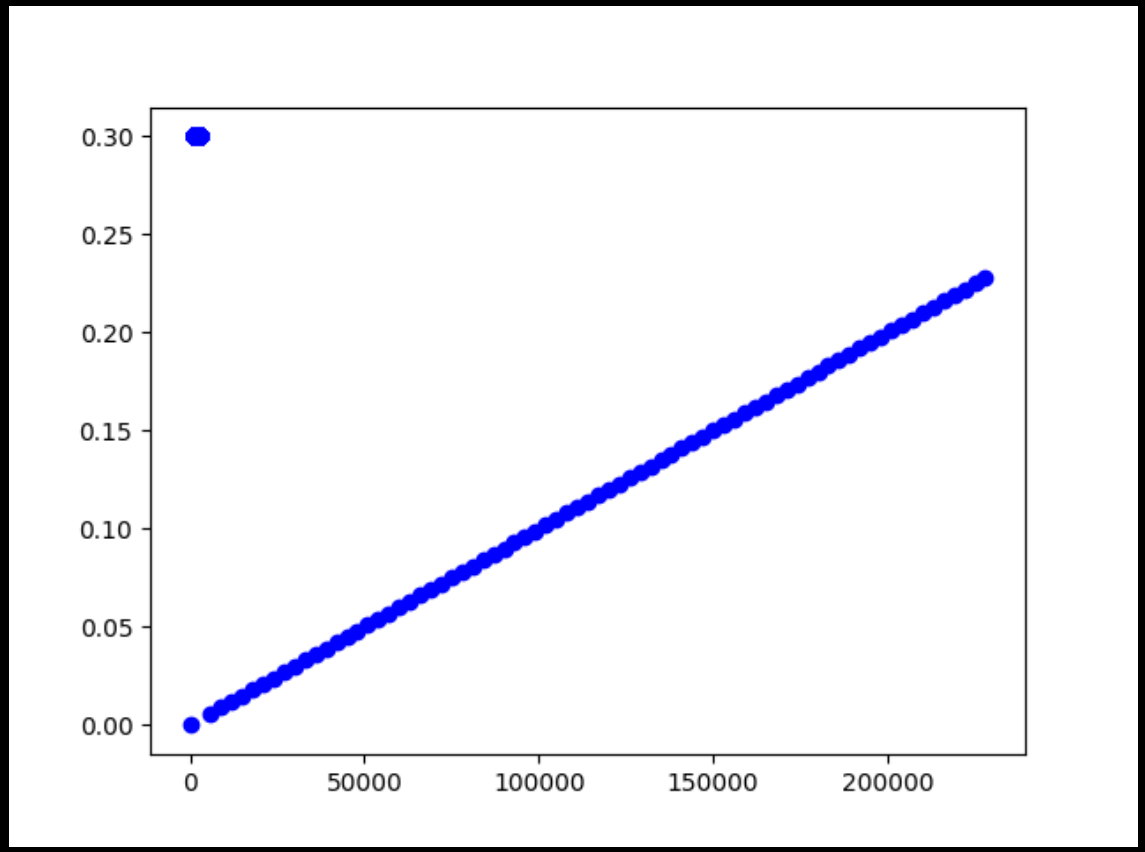


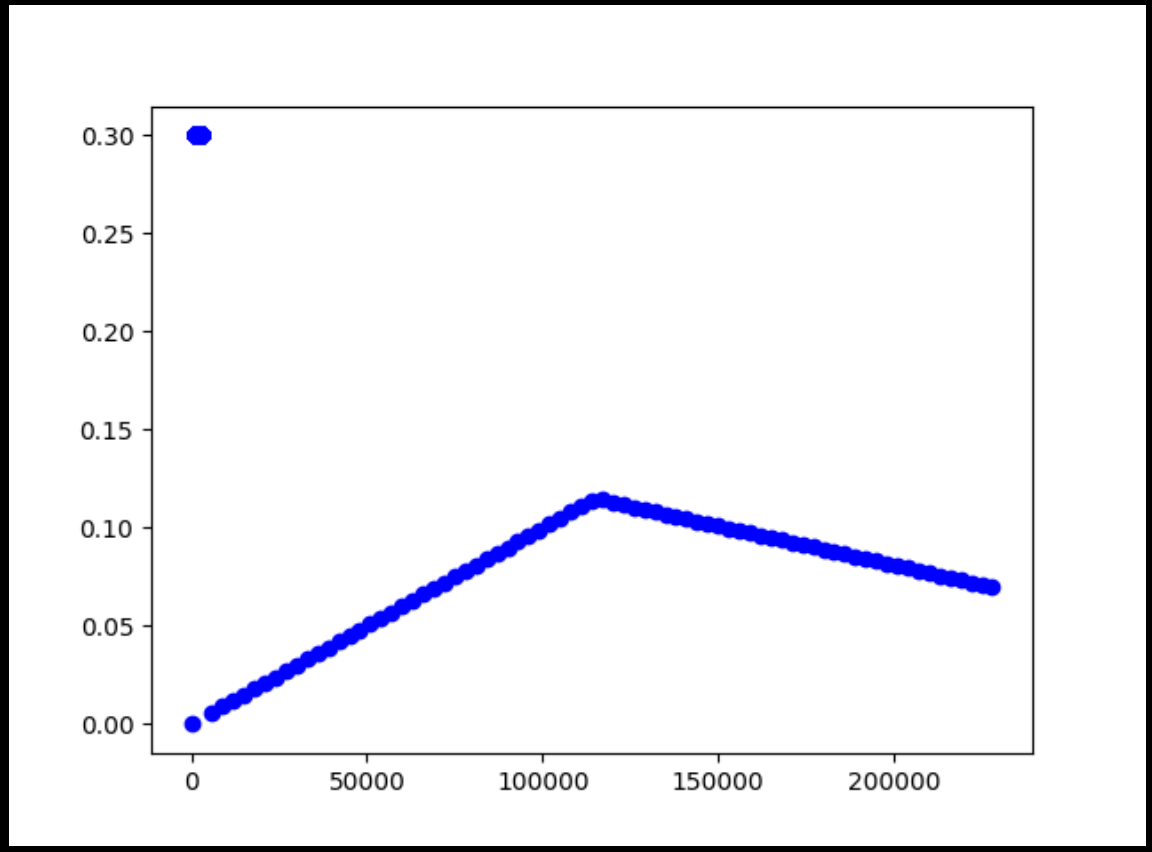
https://arxiv.org/html/2404.06529v1

HEURYSTYKI

- Trzeba się poruszać - bez ruchu nie dojdziemy do wyjścia
- Najlepiej poruszać się w kierunku koloru zielonego
- Zielonego nie może być zbyt dużo, bo to znaczy, że patrzymy / idziemy na ścianę
- Dobrze byłoby nie chodzić w kółko po tych samych miejscach

NAGRODA ZA "ZIELONE"





FUNKCJA NAGRODY

$$reward = 10^{-4} + 0.005 * \Delta_{dist} + greenReward + I$$

$$I = \begin{cases} 1 & znalezionio\ wyjscie \\ 0 & nie\ znaleziono\ wyjscia \end{cases}$$

TRENING