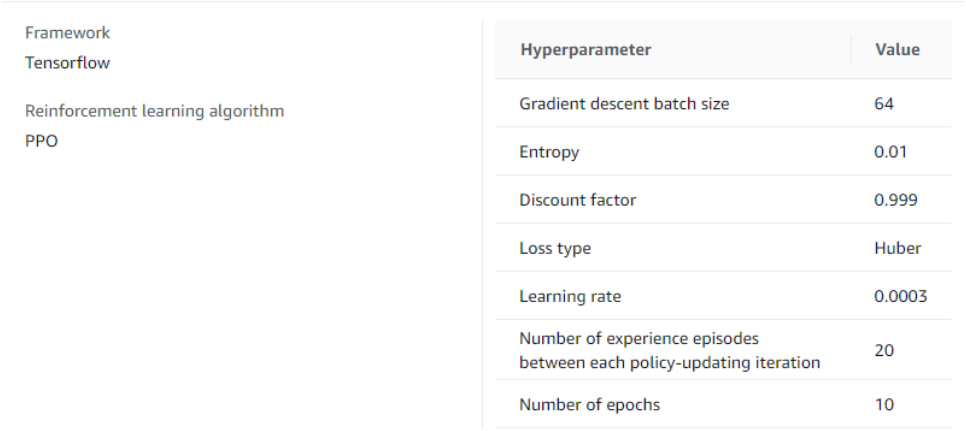
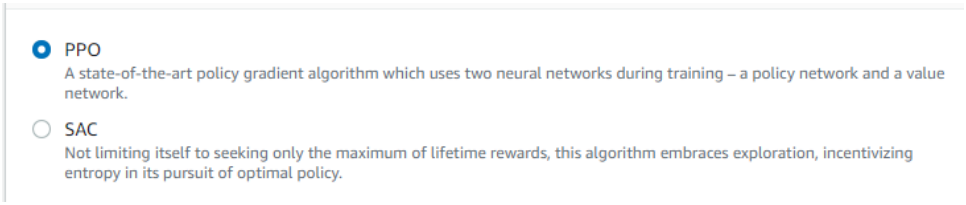
Jestę Kubicą! Zastosowanie AI w wyścigach wirtualnych pojazdów

Dawid Dzhafarov, Justyna Gręda, Mikołaj Sztaba

Opis projektu:
Celem projektu było zbadanie rozwiązania AWS DeepRacer, w którym uczyliśmy wirtualny model jazdy po różnych torach z jak największą prędkością. Każdy uczestnik projektu miał do wykorzystania 10h pracy z modelami na platformie Amazon.
Początkowe godziny spędziliśmy nad badaniem jak działa owe środowisko oraz dopasowywaniem kluczowych parametrów w naszych modelach/funkcjach.



Rys.1 Parametry modelu



Rys.2 Algorytmy uczenia wraz z opisami

Do wyboru były dwa algorytmy uczenia modeli: PPO i SAC. Zdecydowaliśmy się wybrać PPO ze względu na jego prostszą implementację.

Ze względu na ograniczoną ilość czasu trenowania oraz brak możliwości trenowania lokalnego, wybór funkcji kosztu (*reward function*) był kluczowym punktem projektu. Zdecydowaliśmy się na wybór poniższej funkcji, kierując się tym, że zbyt skomplikowana funkcja nie pozwoliłaby naszym modelom na uczenie się przy tak ograniczonych zasobach.

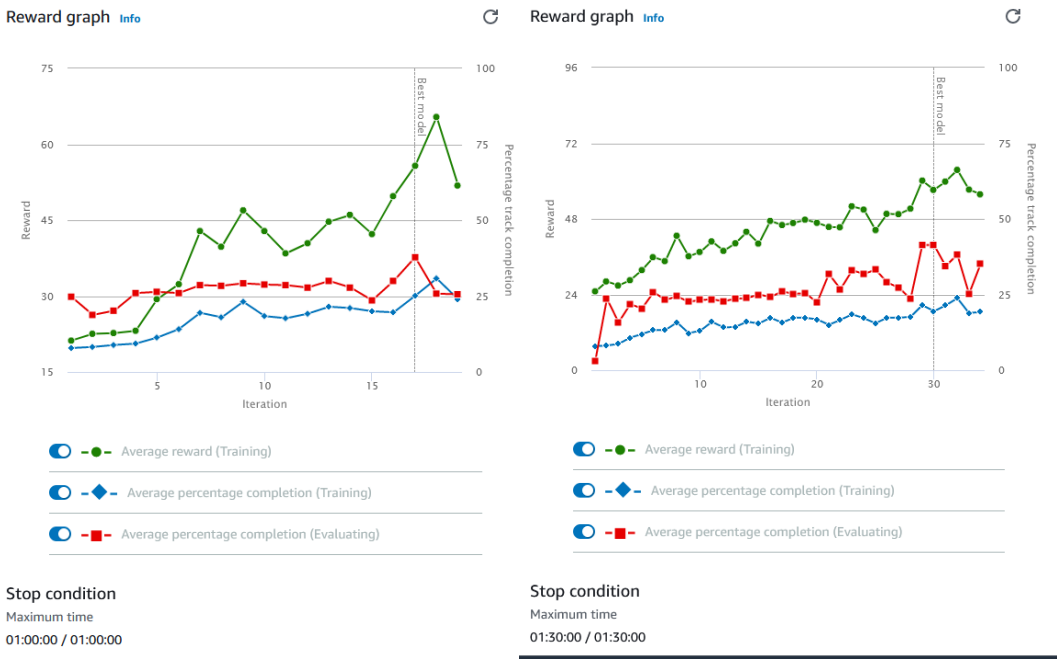
Reward function

```python
def reward_function(params):

    track_width = params['track_width']
    distance_from_center = params['distance_from_center']
    all_wheels_on_track = params['all_wheels_on_track']
    speed = params['speed']
    SPEED_THRESHOLD = 4.0


    # Calculate 3 markers that are at varying distances away from the center line
    marker_1 = 0.1 * track_width
    marker_2 = 0.25 * track_width
    marker_3 = 0.5 * track_width

    # Give higher reward if the car is closer to center line and vice versa
    if distance_from_center <= marker_1:
        reward = 1.0
    elif distance_from_center <= marker_2:
        reward = 0.5
    elif distance_from_center <= marker_3:
        reward = 0.1
    else:
        reward = 1e-3  # likely crashed/ close to off track

    if not all_wheels_on_track:
        # Penalize if the car goes off track
        reward = 1e-3
    elif speed < SPEED_THRESHOLD:
        # Penalize if the car goes too slow
        reward = reward + 0.5
    else:
        # High reward if the car stays on track and goes fast
        reward = reward + 1.0

    return float(reward)
```

Rys.3 Definicja *reward function*

Trening:

Pierwsze dwa modele pozwoliły na zbadanie możliwości dwóch środowisk - ciągłego i dyskretnego. Poniższe zdjęcia przedstawiają proces uczenia się algorytmu z wykorzystaniem tych dwóch metod. Bazując na obserwacji krzywych uczenia, szczególnie tej oznaczonej *Average percentage completion* zauważyliśmy, że oba modele się nie uczą, nie wykonują żadnego progresu. Spowodowane jest to niepoprawnie dobranymi parametrami związanymi z kątami skrętu oraz prędkością, przez co samochód często wypadał z toru.



Rys. 4, 5 Proces uczenia się modeli dyskretnego i ciągłego

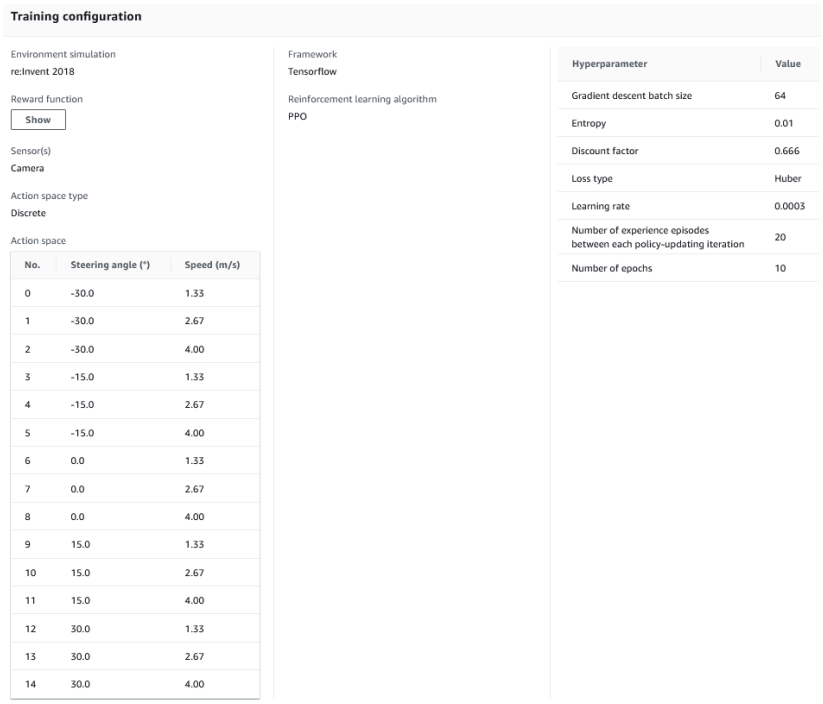
Poniższe zdjęcie przedstawia przykładowy wynik ewaluacji jednego z modeli (ciągłego), widać na nim brak ukończonego żadnego okrążenia, poprzez wypadnięcie z toru.

Evaluation results

Trial	Time (MM:SS.mmm)	Trial results (% track completed)	Status
1	00:04.334	33%	Off track
2	00:04.128	31%	Off track
3	00:04.136	33%	Off track

Rys. 6 Wynik ewaluacji modelu ciągłego

Poniższe zdjęcia przedstawiają wybrane przez nas parametry odnośnie powyższych modeli.



Training configuration

Environment simulation
re:Invent 2018

Reward function
Show

Sensor(s)
Camera

Action space type
Discrete

Action space

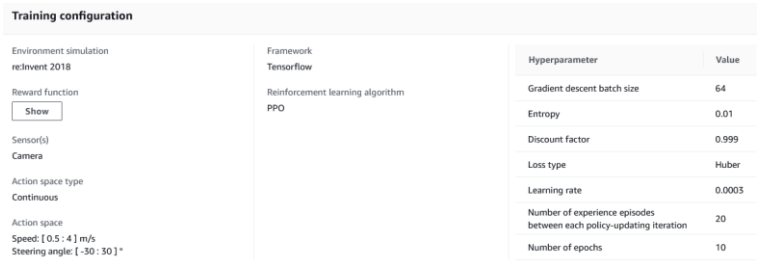
No.	Steering angle (°)	Speed (m/s)
0	-30.0	1.33
1	-30.0	2.67
2	-30.0	4.00
3	-15.0	1.33
4	-15.0	2.67
5	-15.0	4.00
6	0.0	1.33
7	0.0	2.67
8	0.0	4.00
9	15.0	1.33
10	15.0	2.67
11	15.0	4.00
12	30.0	1.33
13	30.0	2.67
14	30.0	4.00

Framework
Tensorflow

Reinforcement learning algorithm
PPO

Hyperparameter	Value
Gradient descent batch size	64
Entropy	0.01
Discount factor	0.666
Loss type	Huber
Learning rate	0.0003
Number of experience episodes between each policy-updating iteration	20
Number of epochs	10

Model dyskretny



Training configuration

Environment simulation
re:Invent 2018

Reward function
Show

Sensor(s)
Camera

Action space type
Continuous

Action space
Speed: [0.5 : 4] m/s
Steering angle: [-30 : 30] °

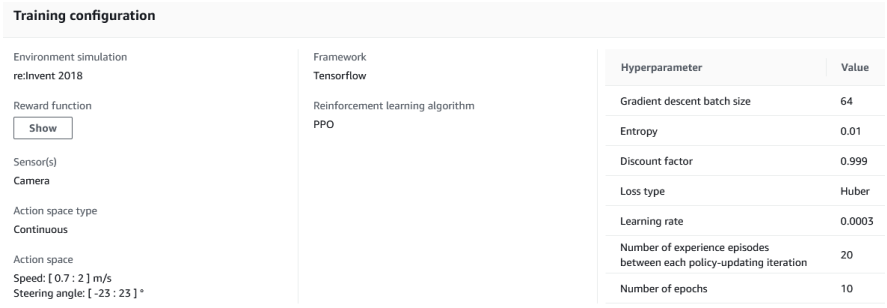
Framework
Tensorflow

Reinforcement learning algorithm
PPO

Hyperparameter	Value
Gradient descent batch size	64
Entropy	0.01
Discount factor	0.999
Loss type	Huber
Learning rate	0.0003
Number of experience episodes between each policy-updating iteration	20
Number of epochs	10

Model ciągły

Kolejny z modeli wykazał się lepszym trendem, jeśli chodzi o procent ukończenia okrążenia. Uzyskaliśmy to dzięki zmniejszeniu kątów skrętu oraz ograniczeniu prędkości. Pozostałe parametry pozostawiliśmy w pierwotnym stanie. Poniższe zdjęcie prezentuje parametry tego modelu.



Training configuration

Environment simulation
re:Invent 2018

Reward function
Show

Sensor(s)
Camera

Action space type
Continuous

Action space
Speed: [0.7 : 2] m/s
Steering angle: [-23 : 23] °

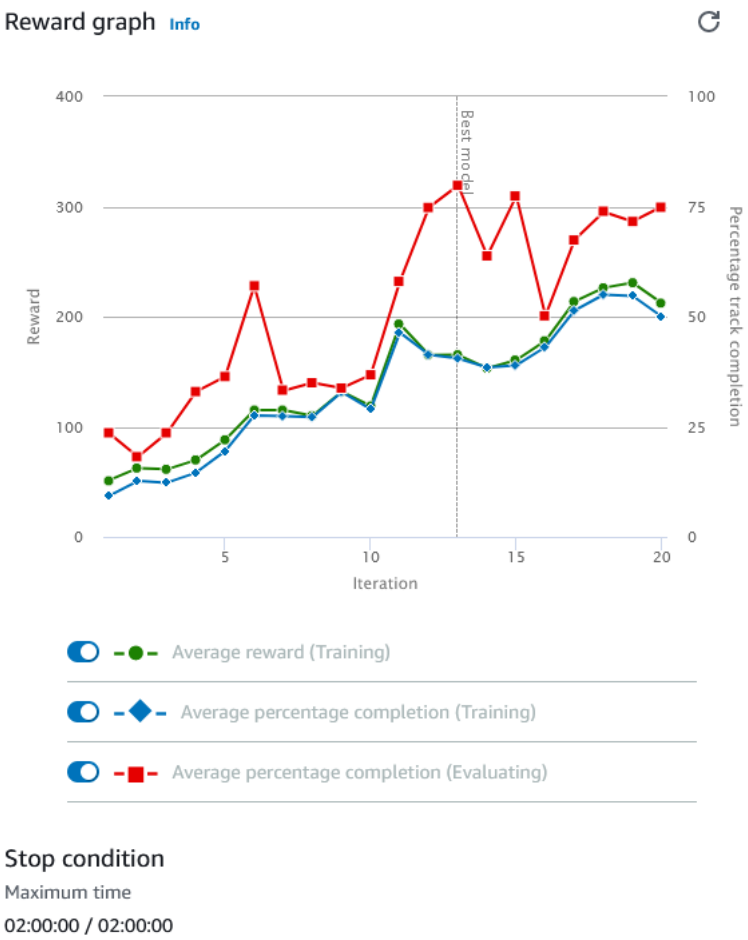
Framework
Tensorflow

Reinforcement learning algorithm
PPO

Hyperparameter	Value
Gradient descent batch size	64
Entropy	0.01
Discount factor	0.999
Loss type	Huber
Learning rate	0.0003
Number of experience episodes between each policy-updating iteration	20
Number of epochs	10

Rys. 7 Parametry trzeciego modelu

Kolejny rysunek przedstawia proces uczenia się tego modelu, po którym stwierdziliśmy, że warto jest nad nim pracować dalej, gdyż wykazuje tendencję do uczenia się. Z powodu ograniczenia czasowego model ten wyeksportowaliśmy tak, by kolejna osoba mogła kontynuować proces nauki.



Rys. 8 Proces uczenia się trzeciego modelu

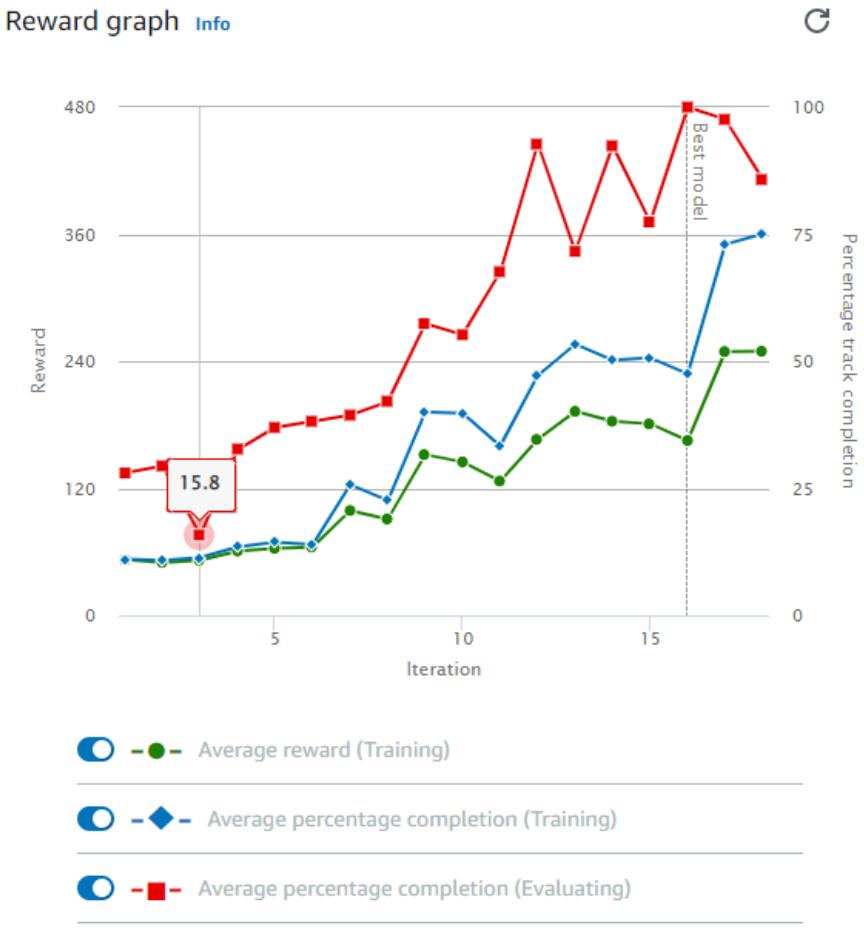
Potwierdzeniem naszej decyzji jest wynik ewaluacji, podczas której pierwszy raz udało się wykonać całe okrążenie:

Evaluation results

Trial	Time (MM:SS.mmm)	Trial results (% track completed)	Status
1	00:06.590	49%	Off track
2	00:04.206	32%	Off track
3	00:12.265	100%	Lap complete

Rys. 9 Wynik ewaluacji trzeciego modelu

Po pierwszych 10h trenowania zdecydowaliśmy się nie zmieniać już *hyperparametrów* oraz algorytmu uczenia, czyli PPO, które ustaliliśmy już wcześniej.
Zaczęliśmy za to eksperymentować z parametrami jazdy, czyli przedziałami prędkości osiąganej przez model na torze oraz możliwym promieniem skrętu kół pojazdu.
Dzięki zaimportowaniu poprzedniego modelu, mogliśmy kontynuować naukę bez straty czasu.



Rys. 10 Proces uczenia się zaimportowanego modelu

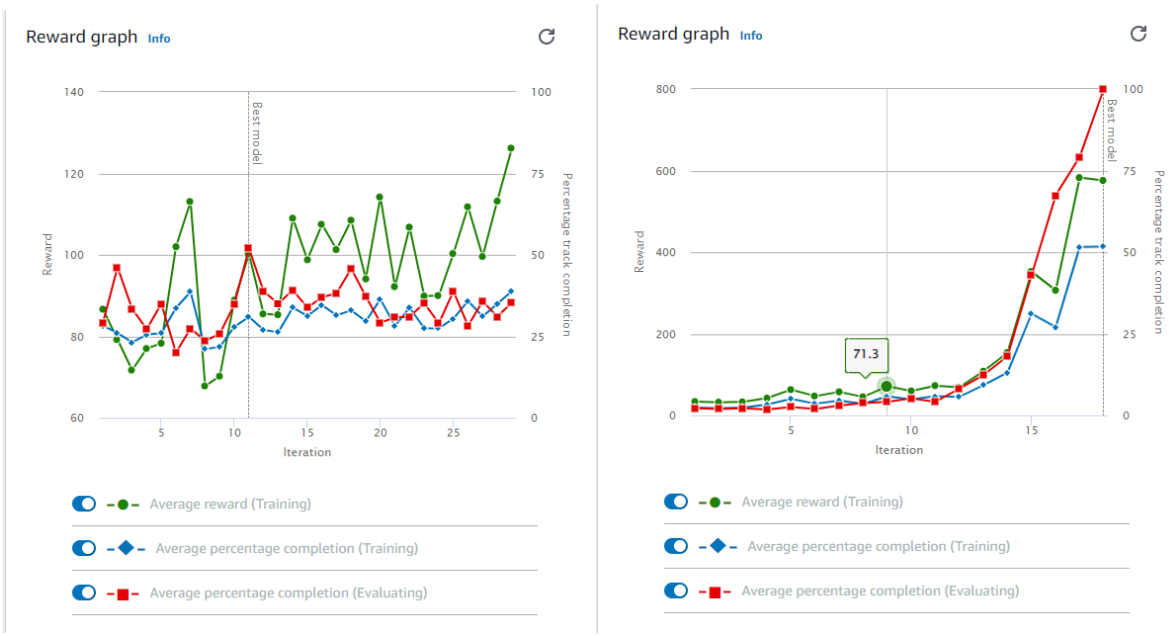
Możemy zauważyć, że nasz model osiągnął 100% w *Percentage track completion,* co oznacza, że udało mu się dojechać do mety we wszystkich przeprowadzonych próbach.

Evaluation results

Trial	Time (MM:SS.mmm)	Trial results (% track completed)	Status
1	00:10.067	100%	Lap complete
2	00:10.799	100%	Lap complete
3	00:09.930	100%	Lap complete

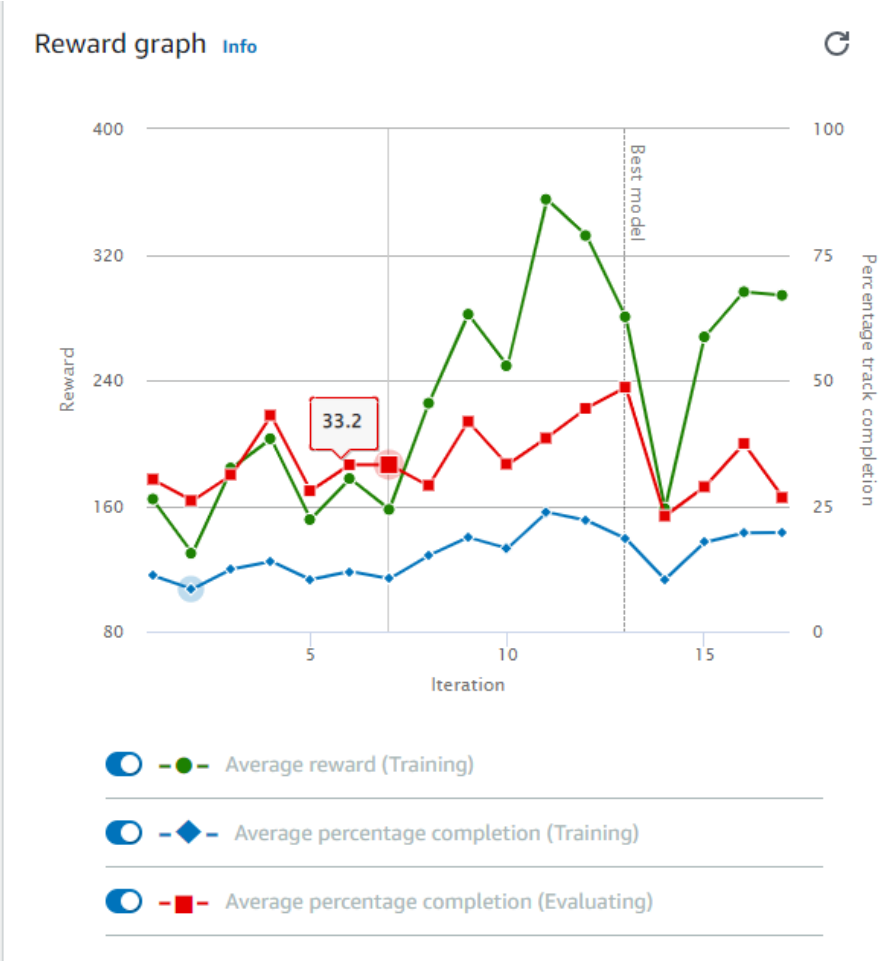
Rys. 11 Wynik ewaluacji zaimportowanego modelu

Później próby kończyły się z różnym efektem, co było czasem powiązane z źle dobraną szybkością czy kątem skrętu kół. Im większa prędkość minimalna czy maksymalna oraz większy możliwy promień skrętu, tym modelowi ciężej się uczyć, ponieważ ma więcej obliczeń do wykonania i więcej decyzji do podjęcia.



Rys. 12 Procesy uczenia się modeli z różnymi jezdnymi parametrami

Można zauważyć, że model po lewej stronie nie uczy się, natomiast ten po prawej po początkowej stagnacji nauczył się w sposób wzorowy jazdy po danym torze.
Warto nadmienić, że wybieraliśmy różne tory, aby modele nie przeuczyły się i nie umiały jeździć tylko po jednej trasie np. po torze Catalunya, który jest rzeczywistym torem w kalendarzu F1.
Końcowy model, który udało się uzyskać po 20h nauki wyglądał w taki sposób.



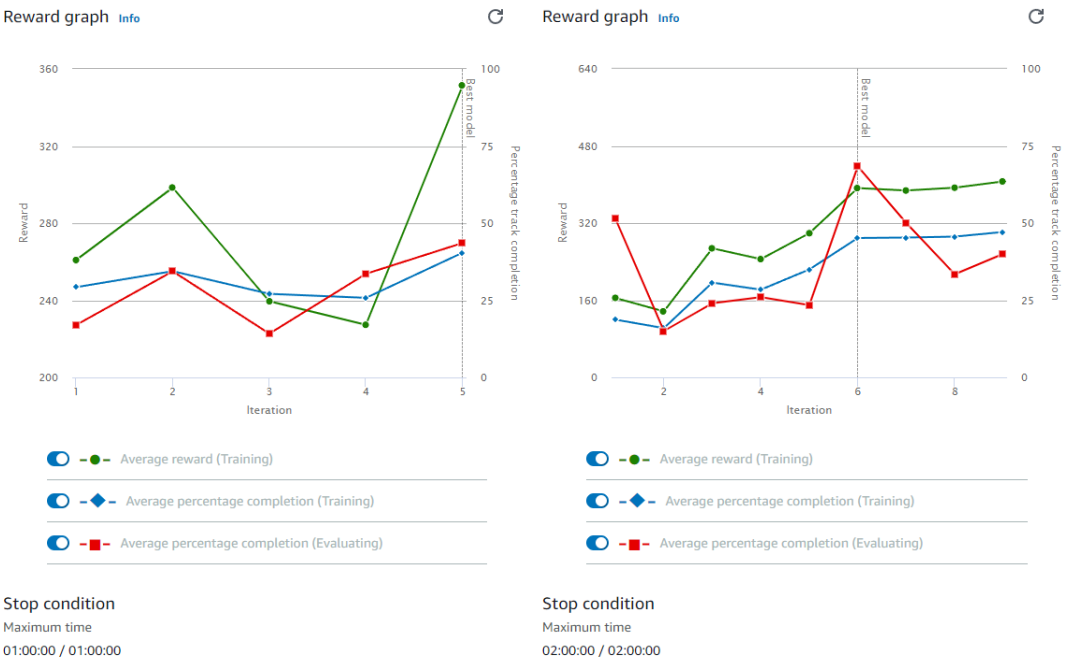
Rys. 13 Wykres uczenia się modelu po 20h nauki

Evaluation results

Trial	Time (MM:SS.mmm)	Trial results (% track completed)	Status
1	00:32.321	65%	Off track
2	00:25.276	49%	Off track
3	00:16.405	30%	Off track

Rys. 14 Ewaluacja modelu po 20h nauki

Warto dodać, że nie zawsze trzeba osiągnąć 100% w ewaluacji na danym torze, po prostu chcemy nauczyć nasz model jazdy po różnych torach po kompletnie innych nitkach toru.
Po kolejnych 10h uczenia model działał już wystarczająco dobrze. Zdecydowaliśmy się nadal eksperymentować już tylko z jego prędkością i kątem skrętu kół.



Rys.15 Proces uczenia się pierwszego modelu

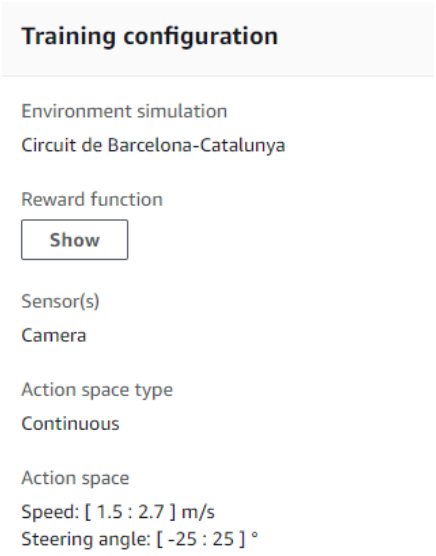
Przy niewielkiej zmianie maksymalnej prędkości pojazdu (z 2.2 na 2.4 m/s) w stosunku do otrzymanego od poprzednika modelu i dość krótkim czasie (łącznie 3 godziny) uczenia można zauważyć, że model nauczył się tylko trochę, znacznie mniej niż wskazywały na to wykresy opisane wyżej. Nie pozwoliło to nawet na ukończenie toru w trakcie ewaluacji:

Evaluation results

Trial	Time (MM:SS.mmm)	Trial results (% track completed)	Status
1	00:13.993	41%	Off track
2	00:03.398	8%	Off track
3	00:21.203	60%	Off track

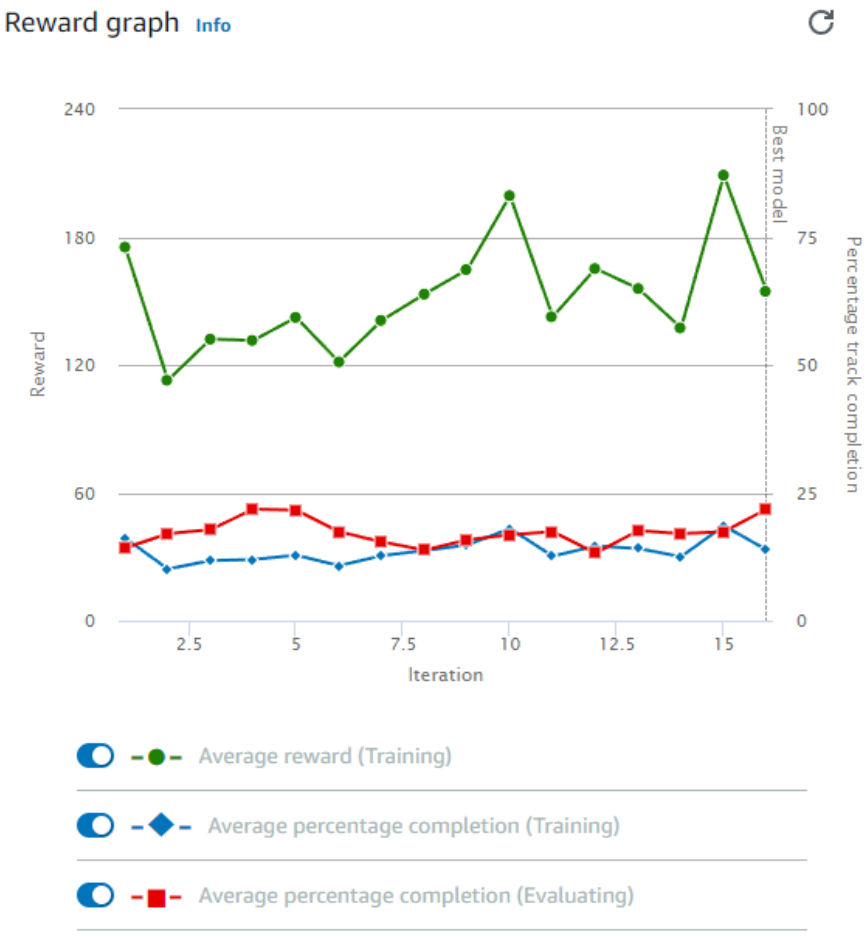
Rys.16 Ewaluacja pierwszego modelu

Mimo wszystko, zdecydowaliśmy się dalej zwiększać prędkość, gdyż naszym celem było już tylko polepszanie miejsca w rankingu AWS DeepRacer League. Kolejny model otrzymał już następujące parametry:



Rys.17 Parametry kolejnego modelu

W przypadku tego modelu zdecydowano się na nieco inny krok - ewaluację przeprowadzono na prostym torze, podczas gdy sam model trenowany był na stosukowo trudnym. Skutki takiego działania można zauważyć na poniższych zrzutach ekranu: chociaż model praktycznie się nie uczył, prosty tor pokonał w 100% nie wypadając z trasy.



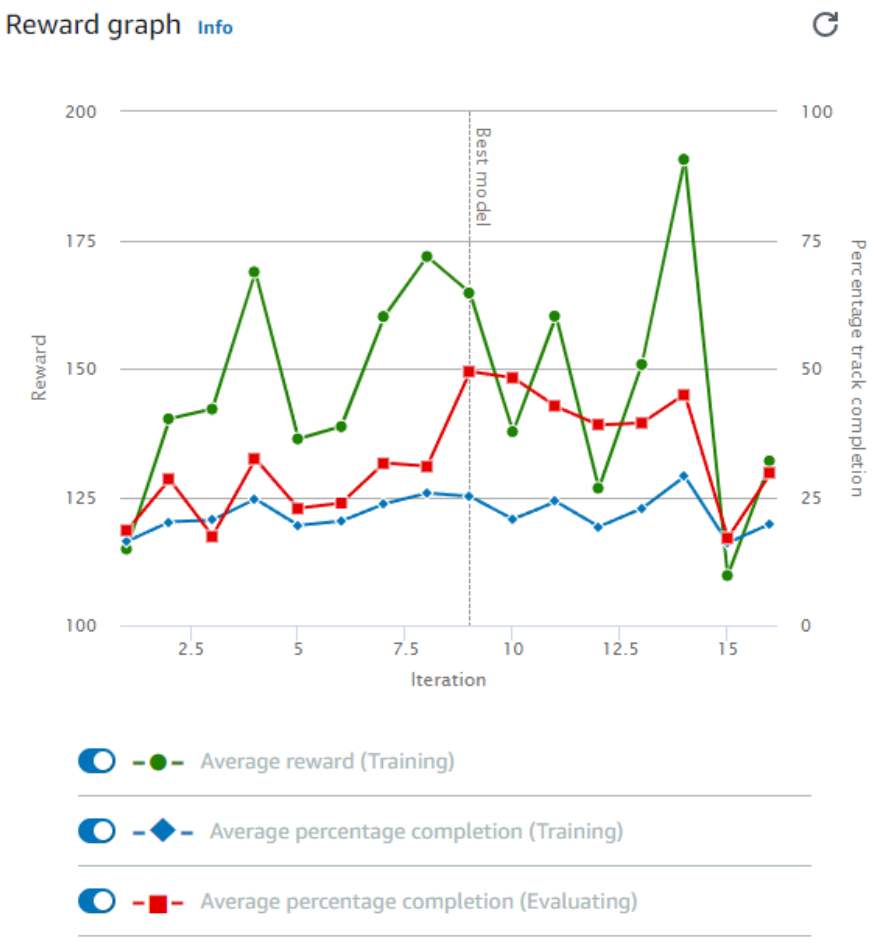
Rys.18 Proces uczenia się kolejnego modelu

Evaluation results

Trial	Time (MM:SS.mmm)	Trial results (% track completed)	Status
1	00:27.472	100%	Lap complete
2	00:27.401	100%	Lap complete
3	00:26.141	100%	Lap complete

Rys.19 Ewaluacja kolejnego modelu

Ostatni model, a jednocześnie ten, który zajął najlepsze miejsce w wirtualnym wyścigu, po 2 godzinach nauki również nauczył się niewiele:



Rys.20 Proces uczenia się ostatniego modelu

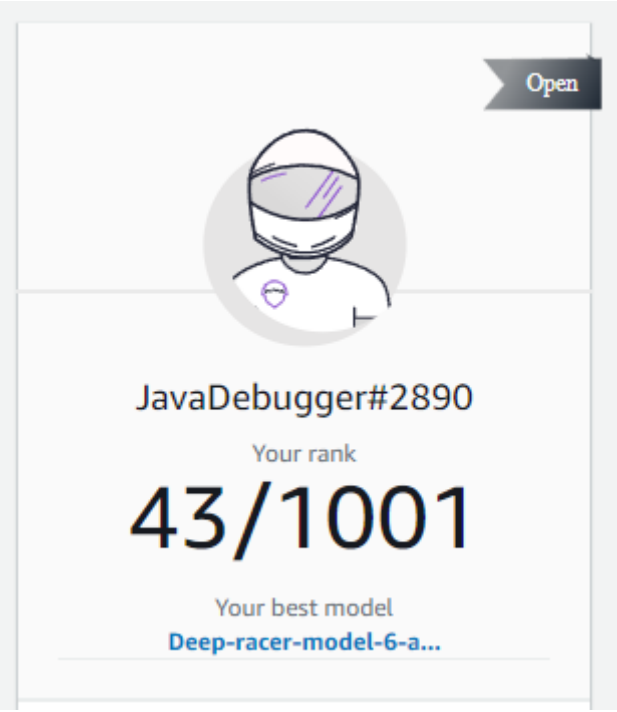
Evaluation results

Trial	Time (MM:SS.mmm)	Trial results (% track completed)	Status
1	00:23.654	100%	Lap complete
2	00:12.139	50%	Off track
3	00:12.401	49%	Off track

Rys.21 Ewaluacja uczenia się ostatniego modelu

Ustawiona w nim jednak była największa prędkość (od 1.8 do 3 m/s) i największy kąt skrętu kół (od -26 do 26 stopni). Mimo niewielkiego stopnia nauki w ciągu ostatnich 10 godzin, osiągnięto duży sukces w wyścigach względem poprzednich modeli, co zostanie przedstawione dokładniej w następnym punkcie.

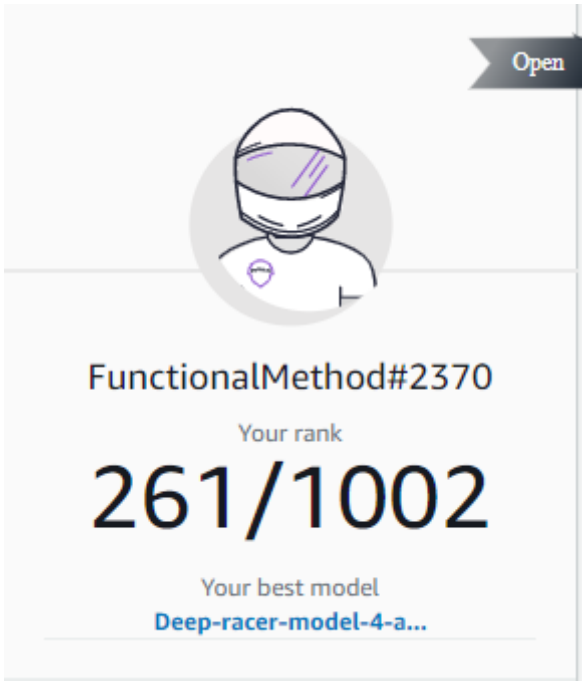
AWS DeepRacer League:
Po ukończeniu ewaluacji modelu możliwe jest zapisanie go do comiesięcznej ligi AWS DeepRacer. Utworzone przez nas modele osiągały w majowych kwalifikacjach dosyć wysokie wyniki, wytrenowany w ostatnich darmowych godzinach zajmuje aktualnie miejsce w top 5% wszystkich modeli, co pozwoli na jego awans na poziom prefesjonalny:



Rys. 22 Pozycja w rankingu najlepszego modelu

Gdy model zostanie zapisany do wirtualnego wyścigu, wykonuje on 3 okrążenia po ustalonym torze. Mierzony jest jego czas i dodawane są kary za wypadnięcie z toru. Nasz najlepszy model, chociaż wypadł z toru 4 razy, był wystarczająco szybki, żeby uplasować się wysoko, pomimo kar. Poniżej można obejrzeć filmik z jego przejazdu po majowym torze The Ross Raceway:

Dla porównania dorzucamy wyniki w AWS DeepRacer League naszego modelu po 20h nauki:



Rys.23 Pozycja w rankingu po 20h nauki

Rank	Racer	Time	Gap to 1st	Video	Off-track
261	**FunctionalMethod#2370**	01:58.587	+00:59.731	Watch	-

Rys.24 Wyniki w AWS DeepRacer League po 20h nauki

Jak można zauważyć, nasz model ani razu nie wyjechał poza trasę, co było powodem, dla którego zdecydowaliśmy się zwiększyć minimalną oraz maksymalną prędkość pojazdu. Koszt wyjazdu poza trasę wynosi jedynie 3 sekundy, co zostanie zniwelowane z nawiązką podczas szybszej jazdy w innych fragmentach toru.

Podsumowanie:
Praca nad projektem była dla całej grupy satysfakcjonująca. Nie udało nam się uruchomić treningu lokalnie, co ograniczyło nas do darmowych 10h przysługujących każdej osobie w ramach okresu próbnego. Napotkaną przez nas trudnością było także przesyłanie wytrenowanych modeli pomiędzy kontami (zapewne zostało to utrudnione właśnie po to, żeby nie uzyskać kolejnych darmowych 10h). Kiedy już udało nam się rozpoznać środowisko, trenowanie modeli, obserwowanie ich sukcesów na torze ewaluacyjnym i progresu w wirtualnych wyścigach było bardzo ciekawym zajęciem.