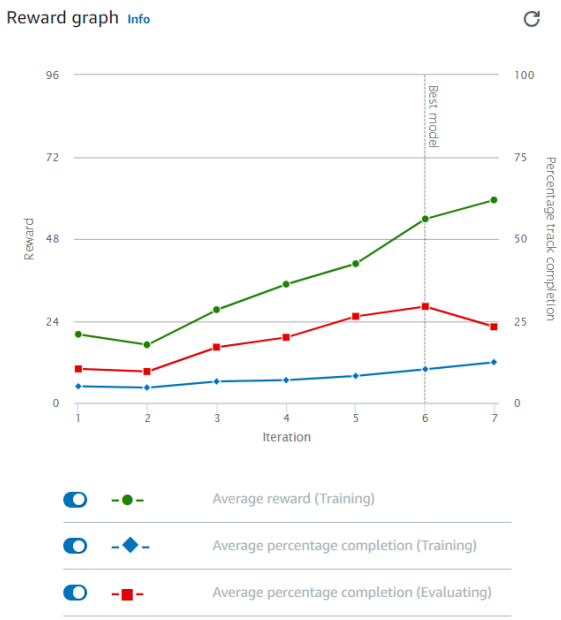
地圖: 2022 re:Invent Championship(Clockwise)

Model0:



Evaluation results

Trial	Time (MM:SS.mmm)	Trial results (% tra
1	00:49.957	100%
2	00:49.942	100%
3	00:49.928	100%

Evaluation results

Off-track	Off-track penalty	Crashes	Cr
3	6 seconds	0	--
3	6 seconds	0	--
3	6 seconds	0	--

Training configuration

Race type
Time trial

Environment simulation
2022 re:Invent Championship - Clockwise

Reward function
Show

Sensor(s)
Camera

Action space type
Continuous

Action space
Speed: [0.5 : 1] m/s
Steering angle: [-30 : 30] °

Framework
Tensorflow

Reinforcement learning algorithm
PPO

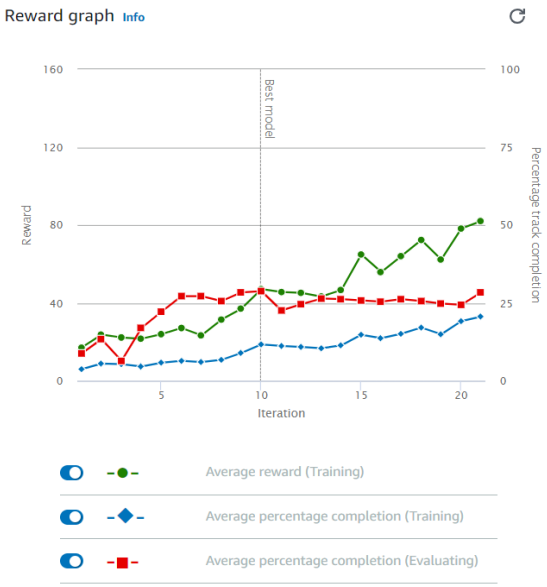
Hyperparameter	Value
Gradient descent batch size	64
Entropy	0.01
Discount factor	0.99
Loss type	Huber
Learning rate	0.0003
Number of experience episodes between each policy-updating iteration	20
Number of epochs	10

Reward function

```python
def reward_function(params):
    '''
    Example of rewarding the agent to follow center line
    '''

    # Read input parameters
    track_width = params['track_width']
    distance_from_center = params['distance_from_center']

    # 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

    return float(reward)
```

Model0 基本沒有做參數的修改，只將 Maximum time 改成 30。Evaluate 的結果單圈大概在 50 秒左右，並且出現 3 次的出軌情況。

Model1:



Trial	Time (MM:SS.mmm)	Trial results (% track completed)
1	00:43.021	100%
2	00:42.600	100%
3	00:42.736	100%

Off-track	Off-track penalty	Crashes	Crash penalty
8	16 seconds	0	--
8	16 seconds	0	--
8	16 seconds	0	--

Training configuration

Race type
Time trial

Environment simulation
2022 re:Invent Championship - Clockwise

Reward function
Show

Sensor(s)
Camera

Action space type
Discrete

Action space

No.	Steering angle (°)	Speed (m/s)
0	-30.0	0.83
1	-30.0	1.67
2	-29.8	2.20
3	-20.0	0.83
4	-20.0	1.67
5	-19.4	2.30
6	-10.0	0.83
7	-9.5	2.30
8	-9.8	3.10
9	-0.4	2.10
10	-0.3	2.80

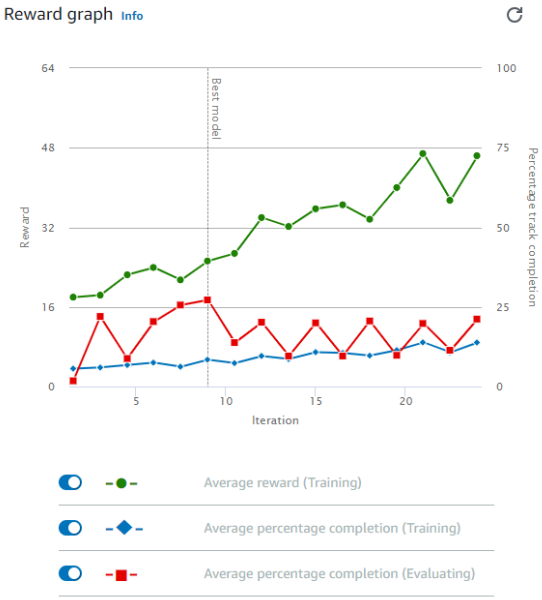
Hyperparameter	Value
Gradient descent batch size	128
Entropy	0.01
Discount factor	0.99
Loss type	Huber
Learning rate	0.0001
Number of experience episodes between each policy-updating iteration	20
Number of epochs	10

Reward function

```python
def reward_function(params):
    '''
    Example of rewarding the agent to follow center line
    '''

    # Read input parameters
    track_width = params['track_width']
    distance_from_center = params['distance_from_center']

    # 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 = 1.5
    elif distance_from_center <= marker_2:
        reward = 0.7
    elif distance_from_center <= marker_3:
        reward = 0.05
    else:
        reward = 1e-3  # likely crashed/ close to off track

    return float(reward)
```

Model1 開始進行 parameters 和 reward function 的調整。首先我將 batch size 稍微調高而 Learning rate 稍微調低。並且改以使用 Discrete 的方式自行加入了多個 Action space。reward function 的部分將 marker_1 的數值調高。雖然相較 Model1 的單圈結果較好，但很明顯由於速度設定太快導致出軌的次數高達 7 次，應對其進行改善。

Model2:



Trial	Time (MM:SS.mmm)	Trial results (% track completed)	Status
1	00:35.945	100%	Lap cor
2	00:32.879	100%	Lap cor
3	00:30.479	100%	Lap cor

Off-track	Off-track penalty	Crashes	Crash penalty
5	10 seconds	0	--
4	8 seconds	0	--
3	6 seconds	0	--

Training configuration

Race type
Time trial

Environment simulation
2022 re:Invent Championship - Clockwise

Reward function
Show

Sensor(s)
Camera

Action space type
Discrete

Action space

No.	Steering angle (°)	Speed (m/s)
0	-30.0	1.00
1	-30.0	2.00
2	-14.8	1.40
3	-14.9	2.50
4	-0.4	2.00
5	-0.3	3.10
6	15.1	1.30
7	14.7	2.40
8	30.0	1.00
9	30.0	2.00

Hyperparameter	Value
Gradient descent batch size	128
Entropy	0.01
Discount factor	0.99
Loss type	Mean squared error
Learning rate	0.0001
Number of experience episodes between each policy-updating iteration	30
Number of epochs	10

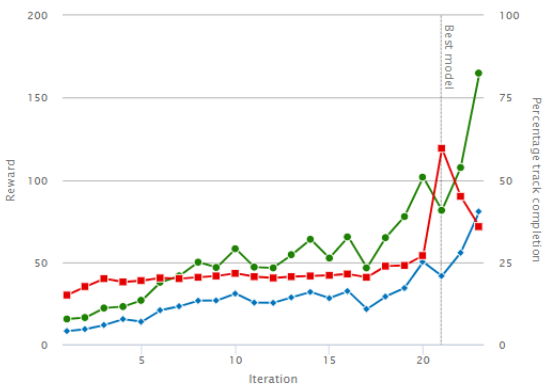
Reward function

```python
def reward_function(params):
    '''
    Example of rewarding the agent to follow center line
    '''

    # Read input parameters
    track_width = params['track_width']
    distance_from_center = params['distance_from_center']

    # 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 = 1.25
    elif distance_from_center <= marker_2:
        reward = 0.7
    elif distance_from_center <= marker_3:
        reward = 0.05
    else:
        reward = 1e-3  # likely crashed/ close to off track

    return float(reward)
```

Model2 大致的參數和 reward funtion 和 Model1 差不多，主要的相異處是我提高了訓練的 Maximum time 以及將整體的轉彎速度調低避免一直出軌的問題。結果的話進步到了單圈 30 秒左右，Off-track 的次數雖然有下降，但感覺還是因為出軌而浪費太多時間。

Model3:

Reward graph Info



Average reward (Training)

Average percentage completion (Training)

Average percentage completion (Evaluating)

Trial	Time (MM:SS.mmm)	Trial results (% track completed)	St
1	00:31.983	100%	La
2	00:34.737	100%	La
3	00:26.799	100%	La

Off-track	Off-track penalty	Crashes	Crash penalty
2	4 seconds	0	--
3	6 seconds	0	--
0	--	0	--

Training configuration

Race type
Time trial

Environment simulation
2022 re:Invent Championship - Clockwise

Reward function
Show

Sensor(s)
Camera

Action space type
Continuous

Action space
Speed: [1 : 1.5] m/s
Steering angle: [-30 : 30] °

Framework
Tensorflow

Reinforcement learning algorithm
PPO

Hyperparameter	Value
Gradient descent batch size	64
Entropy	0.01
Discount factor	0.99
Loss type	Huber
Learning rate	0.0003
Number of experience episodes between each policy-updating iteration	20
Number of epochs	10

Reward function

```python
def reward_function(params):
    '''
    Example of rewarding the agent to follow center line
    '''

    # Read input parameters
    track_width = params['track_width']
    distance_from_center = params['distance_from_center']

    # 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 = 1.25
    elif distance_from_center <= marker_2:
        reward = 0.5
    elif distance_from_center <= marker_3:
        reward = 0.05
    else:
        reward = 1e-3  # likely crashed/ close to off track

    return float(reward)
```

Model3 我將 batch_size 和 learning rate 調整回預設的值，並且改使用 Continuous 的 Action space type 設定 min_speed=1、max_speed=1.5、訓練時間增長到 100。Reward Function 的部分將 marker_1 的 reward 改為 1.25、marker_3 的 reward 改為 0.05。結果明顯能看出出軌的次數有大幅的改善，甚至最好的一次來到 26 秒並且沒有發生 off track 的情況。

最佳結果:

單圈 26 秒

off_track:0 次

結論:

隨著不斷的改變 parameters 以及 reward function 的值，並根據測試的結果進行相對應的調整，能發現整體做出的結果越來越好。要在車速、轉彎、出軌等條件下做出平衡才有辦法得到最好的一個結果。而我最終 Model 的結果雖然出軌的次數減少了，但明顯由於車速相對較慢而無法跑出一個更快的結果。並且適當的調整參數也非常的重要，太複雜或是訓練時間太長的模型結果不一定會是好的(如我將 Maximum time 改為 180，單圈時間反增至 45 秒左右)。