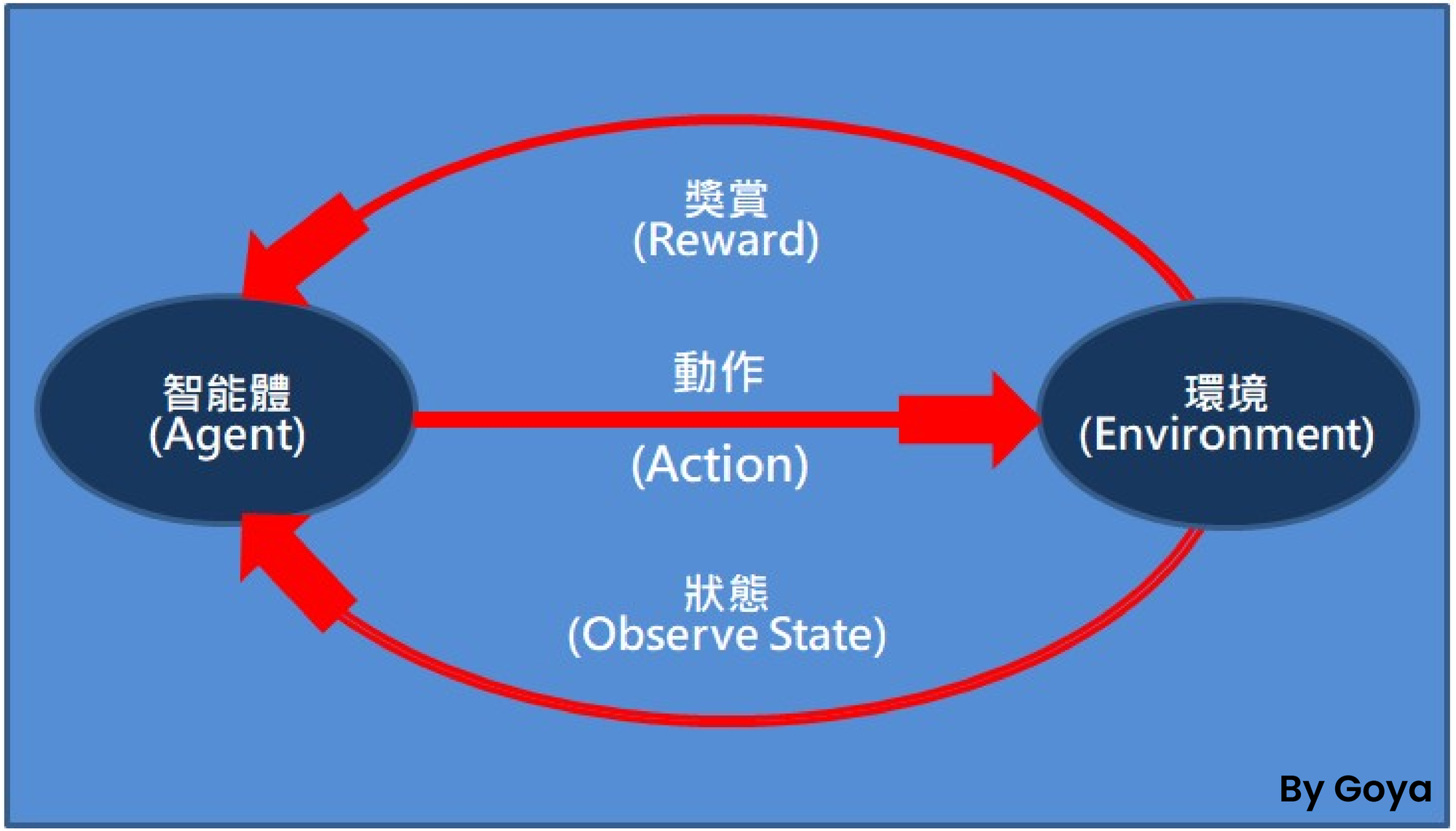
深度學習期末報告

Use Unity ML-Agent
to make Ai parking

組員:周翰文、李沐風

強化學習簡介


獎賞
(Reward)
智能體
(Agent)
動作
(Action)
環境
(Environment)
狀態
(Observe State)
By Goya

算法:PPO(近便政策最佳化)

策略梯度方法

數據效率低

策略更新穩定

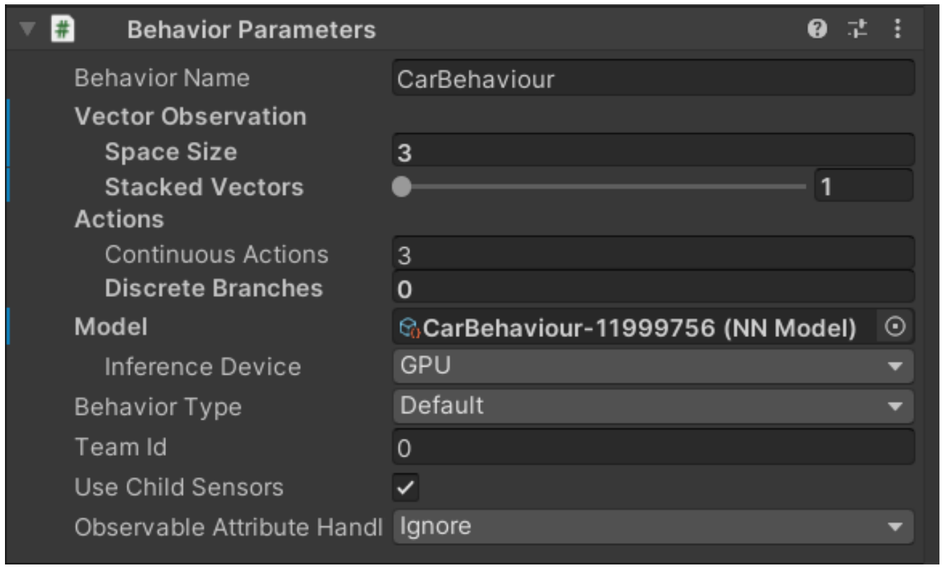
使用工具

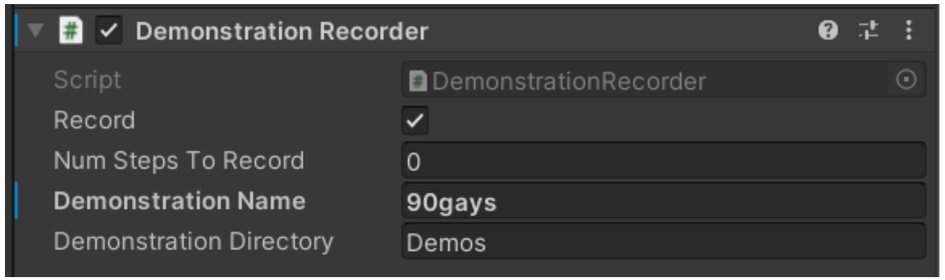
工具列表

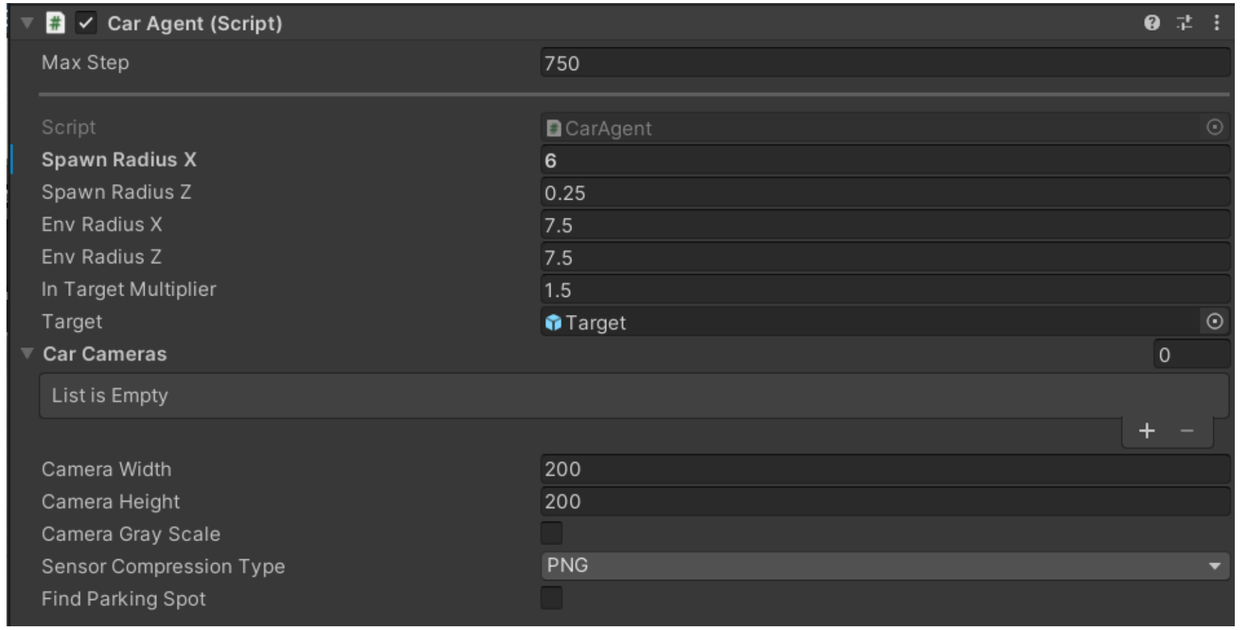
前置環境

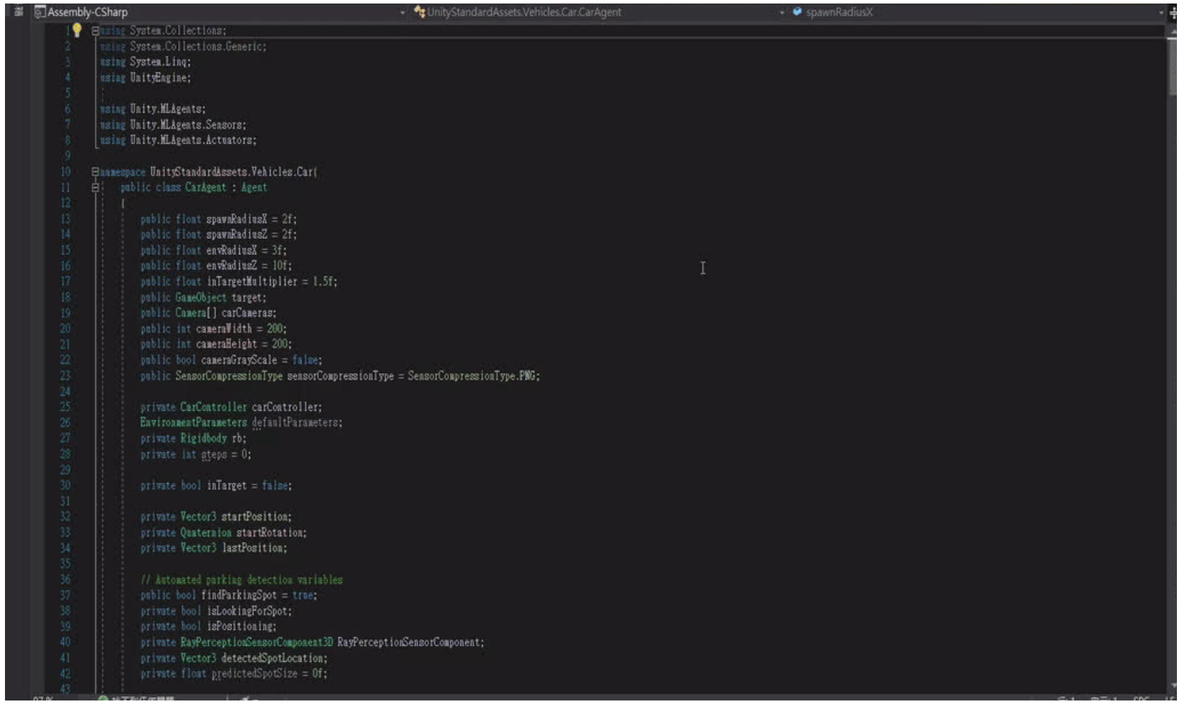
(設定程式、設定場景、設定訓練陣列、設定傳感器)

程式

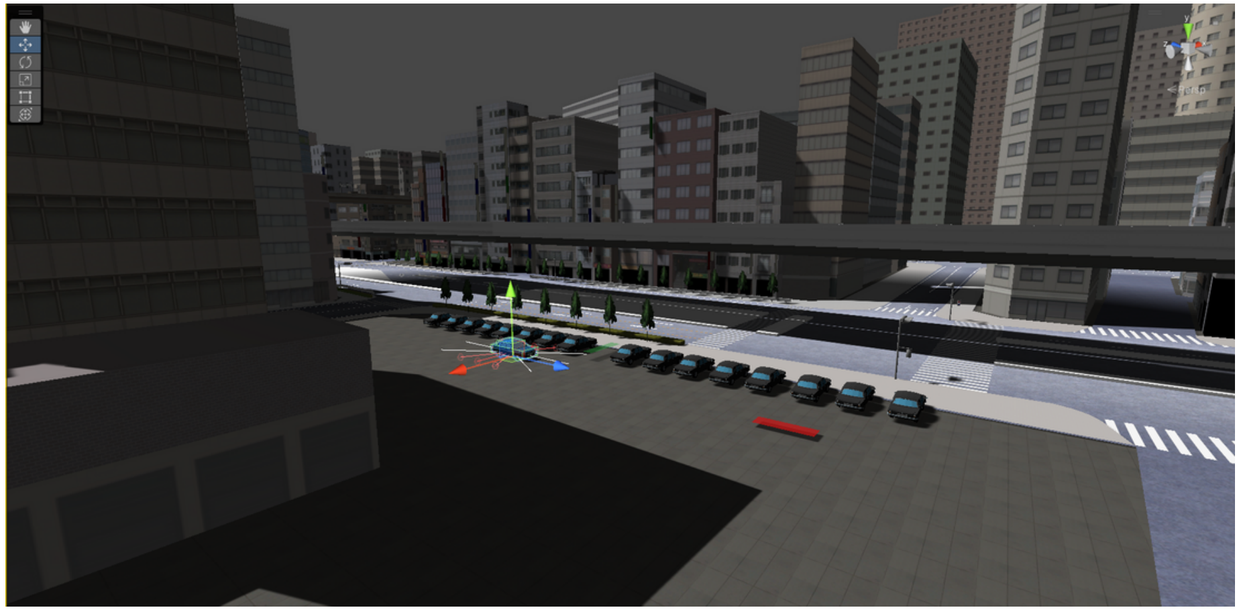

Behavior Parameters
Behavior Name
CarBehaviour
Vector Observation
Space Size
3
Stacked Vectors
1
Actions
Continuous Actions
3
Discrete Branches
0
Model
CarBehaviour-11999756 (NN Model)
Inference Device
GPU
Behavior Type
Default
Team Id
0
Use Child Sensors
Observable Attribute Handl
Ignore

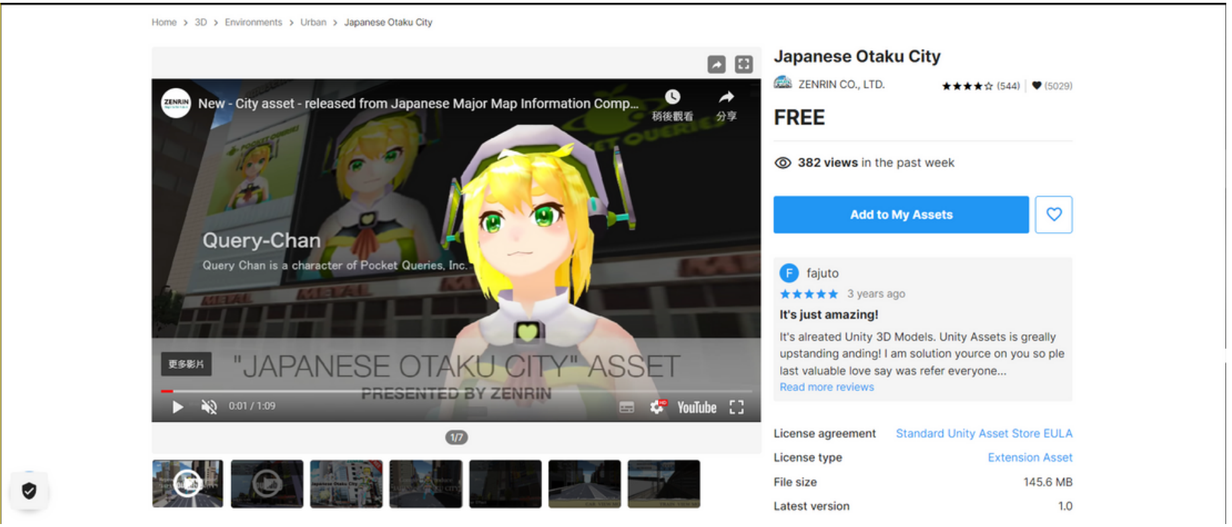

Demonstration Recorder
Script
DemonstrationRecorder
Record
Num Steps To Record
0
Demonstration Name
90gays
Demonstration Directory
Demos

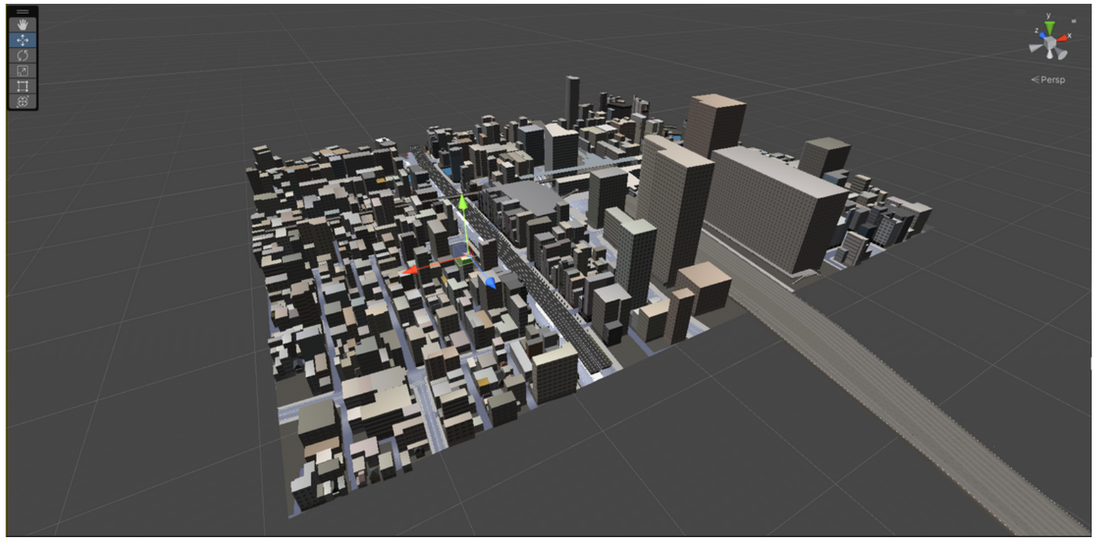

Car Agent (Script)
Max Step
750
Script
CarAgent
Spawn Radius X
6
Spawn Radius Z
0.25
Env Radius X
7.5
Env Radius Z
7.5
In Target Multiplier
1.5
Target
Target
Car Cameras
0
List is Empty
Camera Width
200
Camera Height
200
Camera Gray Scale
Sensor Compression Type
PNG
Find Parking Spot


Assembly-CSharp
UnityStandardAssets.Vehicles.Car.CarAgent
spawnRadiusX
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using UnityEngine;
using Unity.MLAgents;
using Unity.MLAgents.Sensors;
using Unity.MLAgents.Actuators;
namespace UnityStandardAssets.Vehicles.Car{
public class CarAgent : Agent
{
public float spawnRadiusX = 2f;
public float spawnRadiusZ = 2f;
public float envRadiusX = 3f;
public float envRadiusZ = 10f;
public float inTargetMultiplier = 1.5f;
public GameObject target;
public Camera[] carCameras;
public int cameraWidth = 200;
public int cameraHeight = 200;
public bool cameraGrayScale = false;
public SensorCompressionType sensorCompressionType = SensorCompressionType.PNG;
private CarController carController;
EnvironmentParameters defaultParameters;
private Rigidbody rb;
private int steps = 0;
private bool inTarget = false;
private Vector3 startPosition;
private Quaternion startRotation;
private Vector3 lastPosition;
// Automated parking detection variables
public bool findParkingSpot = true;
private bool isLookingForSpot;
private bool isPositioning;
private RayPerceptionSensorComponent3D RayPerceptionSensorComponent;
private Vector3 detectedSpotLocation;
private float predictedSpotSize = 0f;

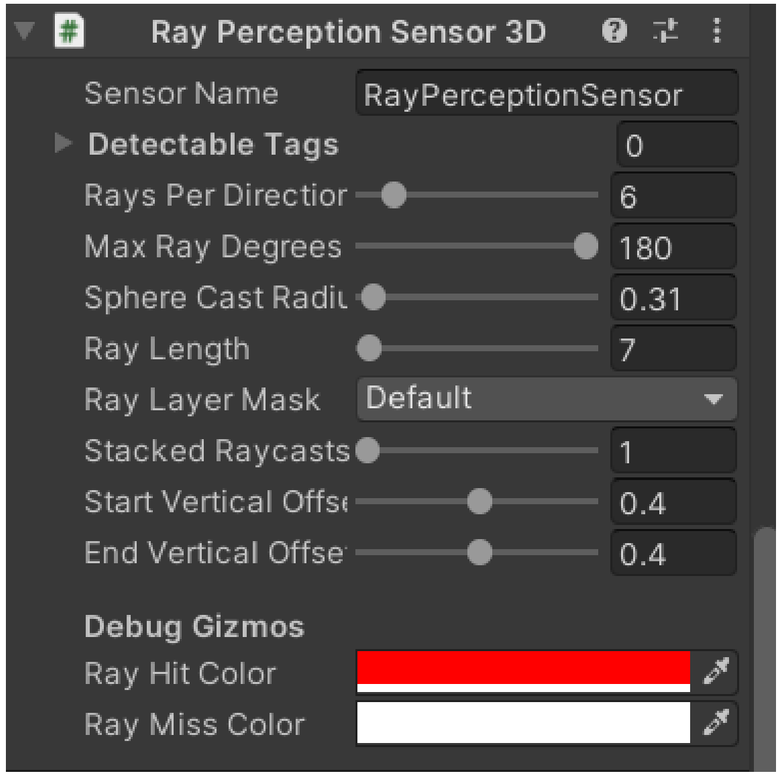
場景

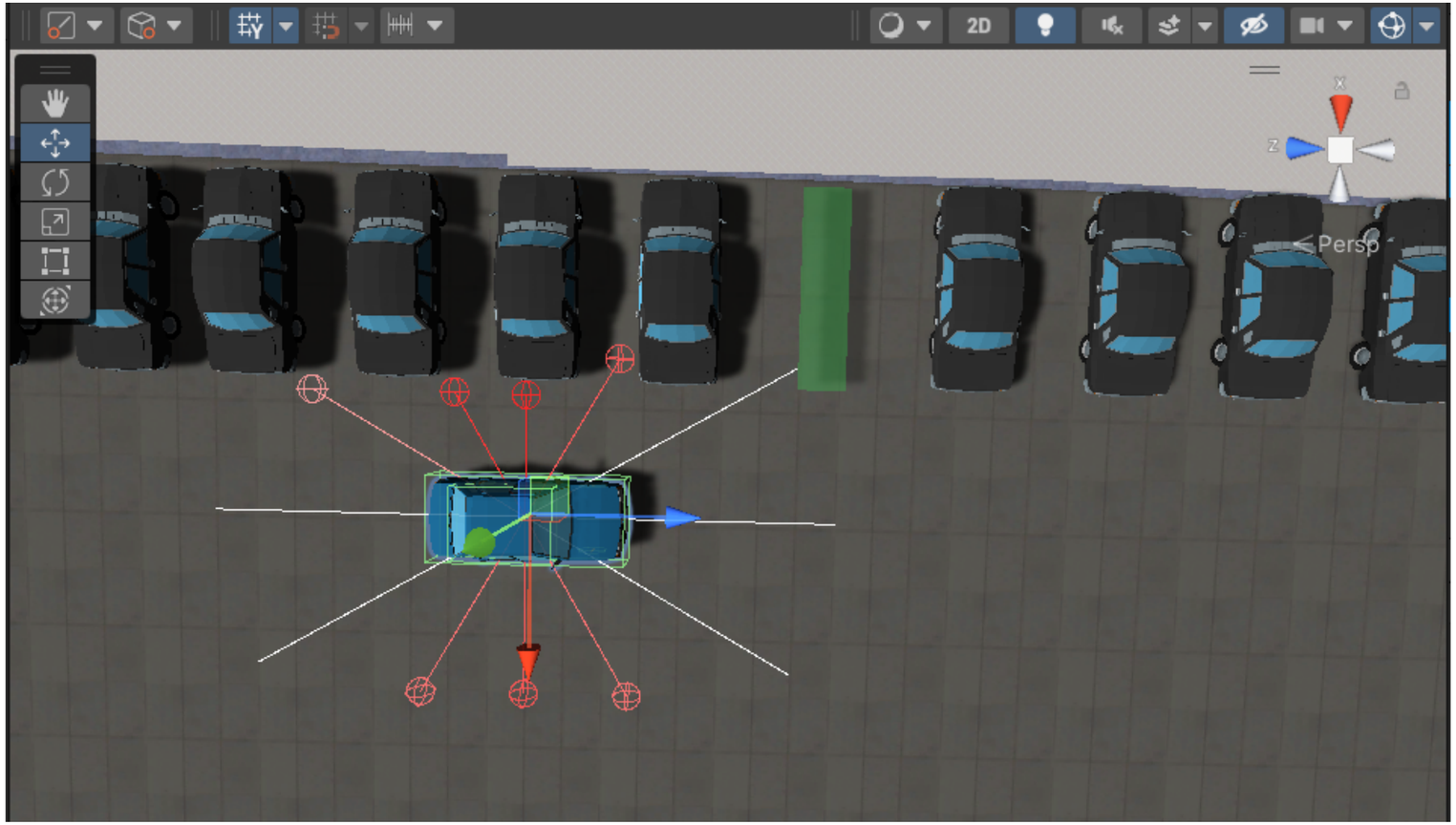



Home > 3D > Environments > Urban > Japanese Otaku City
New - City asset - released from Japanese Major Map Information Comp...
稍後觀看
分享
Query-Chan
Query Chan is a character of Pocket Queries, Inc.
更多影片
"JAPANESE OTAKU CITY" ASSET
PRESENTED BY ZENRIN
0:01 / 1:09
YouTube
1/7
Japanese Otaku City
ZENRIN CO., LTD.
★★★★☆ (544) | ♥ (5029)
FREE
382 views in the past week
Add to My Assets
fajuto
★★★★★ 3 years ago
It's just amazing!
It's alreated Unity 3D Models. Unity Assets is greally upstanding anding! I am solution yource on you so ple last valuable love say was refer everyone...
Read more reviews
License agreement Standard Unity Asset Store EULA
License type Extension Asset
File size 145.6 MB
Latest version 1.0

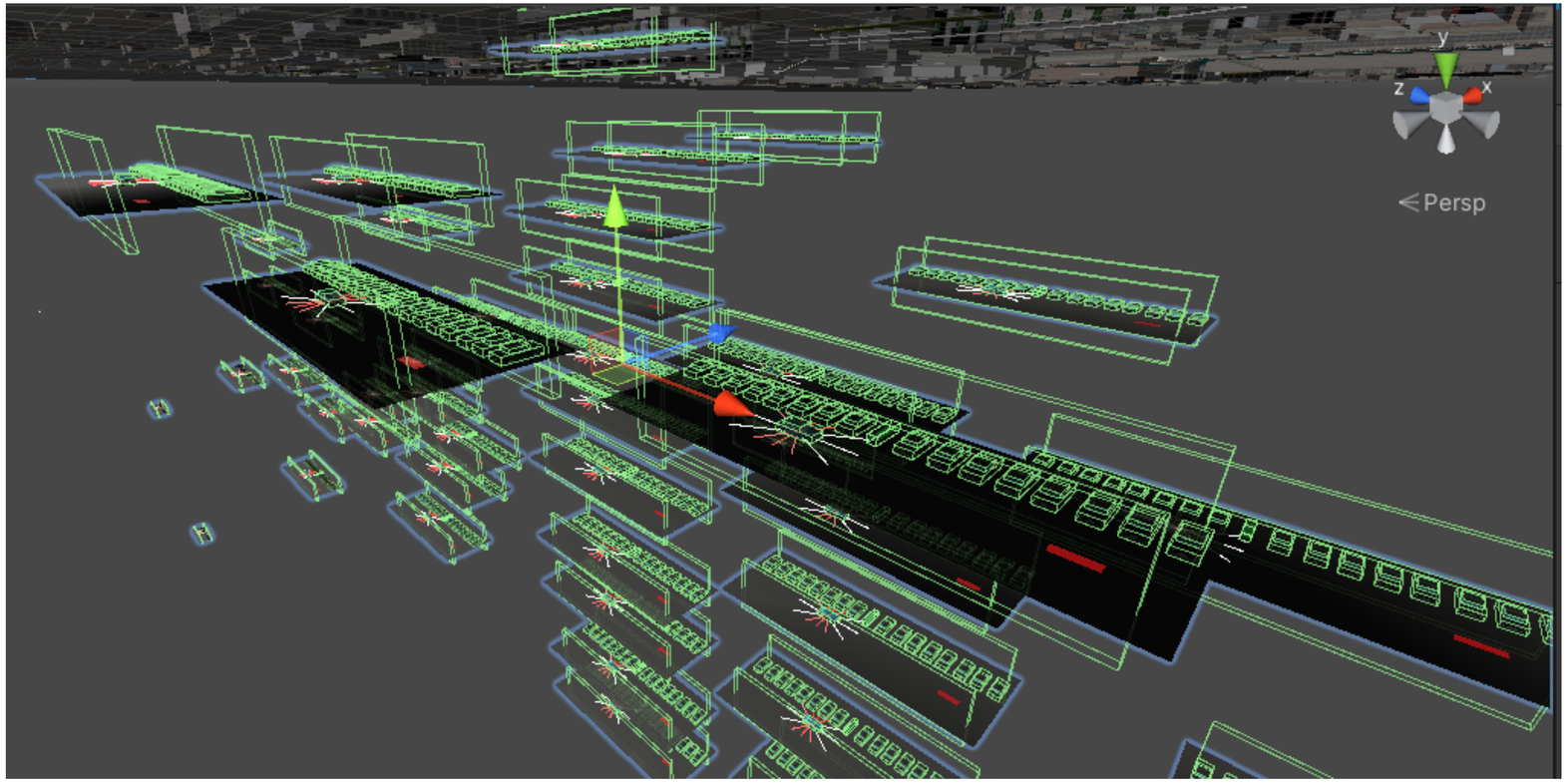

Persp

傳感器


Ray Perception Sensor 3D
Sensor Name
RayPerceptionSensor
Detectable Tags
0
Rays Per Direction
6
Max Ray Degrees
180
Sphere Cast Radius
0.31
Ray Length
7
Ray Layer Mask
Default
Stacked Raycasts
1
Start Vertical Offset
0.4
End Vertical Offset
0.4
Debug Gizmos
Ray Hit Color
Ray Miss Color


2D
Persp

訓練陣列 60個環境



獎勵算法

獎勵機制

增加獎勵:

- 成功停車(角度、速度)

扣除獎勵:

- 撞牆、撞車
- 超出訓練範圍

訓練程式

```yaml
CarBehaviour:
  trainer_type: ppo
  hyperparameters: #超參數設定
    batch_size: 1024 #每次更新參數時，會有多少步的（state，action， reward...）來進行學習
    buffer_size: 5120 #程式要先集滿5120步數之後才會開啟一輪訓練，再將其分成(5120/1024)個batch，然後每個 batch 進行一次訓練並更新參數，整個過程重複epoch次
    learning_rate: 0.00035 #梯度下降的學習率
    beta: 0.0025 #增加熵值來讓動作策略更加隨機，beta越大，越鼓勵探索
    epsilon: 0.3 #策略更新的速度，越大的話可以使訓練更加穩定，但收斂速度會變更慢
    lambd: 0.95 #決定了我們的算法多大程度上依賴預估與實際的獎勵值
    num_epoch: 5 #每次進行梯度下降更新的次數
    learning_rate_schedule: linear #對學習率使用線性遞減，以達到更快的收斂速度

  network_settings:
    normalize: true #使用特徵縮放，對於複雜的連續動作才會有幫助
    hidden_units: 264 #隱藏層的神經元數，數值的大小決定了神經網路對遊戲狀態的表達能力。越多的隱藏層對更大的觀察空間有著較好的表達能力
    num_layers: 3 #訓練網路的層數。數字越大，模型越深層。但可能會帶來梯度消失與訓練速度慢的問題

  reward_signals:
    extrinsic: #外在獎勵(環境)
      gamma: 0.95 #獎勵衰減度
      strength: 0.99 #獎勵強度
    gail: #生成對抗模仿學習
      strength: 0.3 #獎勵強度
      demo_path: D:/Unity Project/teat1230/Demos/parkteat2.demo #之前訓練出結果的策略文件
      use_actions: false #不必包含動作，只需策略

  behavioral_cloning: #監督學習，用於限制探索時間，並幫助其探索時找出正確方向。可節省時間與性能
    demo_path: D:/Unity Project/teat1230/Demos/parkteat2.demo
    steps: 750000
    strength: 0.4

  keep_checkpoints: 15 #保留多少個訓練時產生的checkpoint
  checkpoint_interval: 1000000 #每1000000步後，模型會存一個節點。
  time_horizon: 264 #在264步數之後，開始把收集到的經驗數據放入到經驗池(類似過擬合、欠擬合)
  max_steps: 50000000 #最多只會訓練50000000以下步數
  summary_freq: 100000 #每100000步數之後，開始紀錄我們的訓練統計數據
  threaded: true #開啟python內建的多線程功能來實現一邊訓練一邊存儲模型
```


Scene
Game
Asset Store
2D
Persp
Inspector
Team Id 0
Use Child Sensors
Observable Attribu Ignore
There is no model for this Brain; cannot run inference. (But can still train)
Decision Requester
Script DecisionRequester
Decision Period 1
Take Actions Betwe
Ray Perception Sensor 3D
Sensor Name RayPerceptionSensor
Detectable Tags 0
Rays Per Directior 6
Max Ray Degrees 180
Sphere Cast Radiu 0.31
Ray Length 7
Ray Layer Mask Default
Stacked Raycasts 1
Start Vertical Offs 0.4
End Vertical Offse 0.4
Debug Gizmos
Ray Hit Color
Ray Miss Color
Demonstration Recorder
Script DemonstrationReco
Record
Num Steps To Recor 0
Demonstration Nam 90gays
Demonstration Direc Demos
Add Component
Packages > ML Agents > Runtime > Demonstrations
Demonstra...
Demonstra...
Demonstra...
Demonstra...

https://www.youtube.com/watch?v=upyeNufLjs8

```yaml
CarBehaviour:
  trainer_type: ppo
  hyperparameters: #超參數設定
    batch_size: 1024 #每次更新參數時，會有多少步的（state，action， reward...）來進行學習
    buffer_size: 5120 #程式要先集滿5120步數之後才會開啟一輪訓練，再將其分成(5120/1024)個batch，然後每個 batch 進行一次訓練並更新參數，整個過程重複epoch次
    learning_rate: 0.00035 #梯度下降的學習率
    beta: 0.0025 #增加熵值來讓動作策略更加隨機，beta越大，越鼓勵探索
    epsilon: 0.3 #策略更新的速度，越大的話可以使訓練更加穩定，但收斂速度會變更慢
    lambd: 0.95 #決定了我們的算法多大程度上依賴預估與實際的獎勵值
    num_epoch: 5 #每次進行梯度下降更新的次數
    learning_rate_schedule: linear #對學習率使用線性遞減，以達到更快的收斂速度

  network_settings:
    normalize: true #使用特徵縮放，對於複雜的連續動作才會有幫助
    hidden_units: 264 #隱藏層的神經元數，數值的大小決定了神經網路對遊戲狀態的表達能力。越多的隱藏層對更大的觀察空間有著較好的表達能力
    num_layers: 3 #訓練網路的層數。數字越大，模型越深層。但可能會帶來梯度消失與訓練速度慢的問題

  reward_signals:
    extrinsic: #外在獎勵(環境)
      gamma: 0.95 #獎勵衰減度
      strength: 0.99 #獎勵強度
    gail: #生成對抗模仿學習
      strength: 0.3 #獎勵強度
      demo_path: D:/Unity Project/teat1230/Demos/parkteat2.demo #之前訓練出結果的策略文件
      use_actions: false #不必包含動作，只需策略

  behavioral_cloning: #監督學習，用於限制探索時間，並幫助其探索時找出正確方向。可節省時間與性能
    demo_path: D:/Unity Project/teat1230/Demos/parkteat2.demo
    steps: 750000
    strength: 0.4

  keep_checkpoints: 15 #保留多少個訓練時產生的checkpoint
  checkpoint_interval: 1000000 #每1000000步後，模型會存一個節點。
  time_horizon: 264 #在264步數之後，開始把收集到的經驗數據放入到經驗池(類似過擬合、欠擬合)
  max_steps: 50000000 #最多只會訓練50000000以下步數
  summary_freq: 100000 #每100000步數之後，開始紀錄我們的訓練統計數據
  threaded: true #開啟python內建的多線程功能來實現一邊訓練一邊存儲模型
```

Anaconda Prompt (Anaconda

```
[INFO] CarBehaviour. Step: 2800000. Time Elapsed: 3486.212 s. Mean Reward: -109.374. Std of Reward: 170.545. Training.
[INFO] CarBehaviour. Step: 2900000. Time Elapsed: 3572.751 s. Mean Reward: -93.423. Std of Reward: 175.991. Training.
[INFO] CarBehaviour. Step: 3000000. Time Elapsed: 3658.638 s. Mean Reward: -108.460. Std of Reward: 179.354. Training.
[INFO] Exported results\mlagents1\CarBehaviour\CarBehaviour-2999951.onnx
[INFO] CarBehaviour. Step: 3100000. Time Elapsed: 3759.530 s. Mean Reward: -103.407. Std of Reward: 168.830. Training.
[INFO] CarBehaviour. Step: 3200000. Time Elapsed: 3855.764 s. Mean Reward: -103.247. Std of Reward: 174.972. Training.
[INFO] CarBehaviour. Step: 3300000. Time Elapsed: 3950.040 s. Mean Reward: -93.265. Std of Reward: 157.743. Training.
[INFO] CarBehaviour. Step: 3400000. Time Elapsed: 4040.181 s. Mean Reward: -88.253. Std of Reward: 150.180. Training.
[INFO] CarBehaviour. Step: 3500000. Time Elapsed: 4137.314 s. Mean Reward: -100.797. Std of Reward: 172.829. Training.
[INFO] CarBehaviour. Step: 3600000. Time Elapsed: 4232.873 s. Mean Reward: -97.138. Std of Reward: 169.860. Training.
[INFO] CarBehaviour. Step: 3700000. Time Elapsed: 4321.822 s. Mean Reward: -106.370. Std of Reward: 185.393. Training.
[INFO] CarBehaviour. Step: 3800000. Time Elapsed: 4414.687 s. Mean Reward: -109.490. Std of Reward: 169.717. Training.
[INFO] CarBehaviour. Step: 3900000. Time Elapsed: 4504.885 s. Mean Reward: -105.258. Std of Reward: 157.714. Training.
[INFO] CarBehaviour. Step: 4000000. Time Elapsed: 4598.343 s. Mean Reward: -101.874. Std of Reward: 165.826. Training.
[INFO] Exported results\mlagents1\CarBehaviour\CarBehaviour-3999938.onnx
[INFO] CarBehaviour. Step: 4100000. Time Elapsed: 4691.743 s. Mean Reward: -94.384. Std of Reward: 173.884. Training.
[INFO] CarBehaviour. Step: 4200000. Time Elapsed: 4785.094 s. Mean Reward: -100.277. Std of Reward: 182.768. Training.
[INFO] CarBehaviour. Step: 4300000. Time Elapsed: 4878.896 s. Mean Reward: -94.410. Std of Reward: 147.766. Training.
[INFO] CarBehaviour. Step: 4400000. Time Elapsed: 4971.541 s. Mean Reward: -97.918. Std of Reward: 167.456. Training.
[INFO] CarBehaviour. Step: 4500000. Time Elapsed: 5068.152 s. Mean Reward: -83.488. Std of Reward: 151.603. Training.
[INFO] CarBehaviour. Step: 4600000. Time Elapsed: 5160.794 s. Mean Reward: -100.727. Std of Reward: 193.059. Training.
[INFO] CarBehaviour. Step: 4700000. Time Elapsed: 5255.509 s. Mean Reward: -80.636. Std of Reward: 200.708. Training.
[INFO] CarBehaviour. Step: 4800000. Time Elapsed: 5346.742 s. Mean Reward: -105.449. Std of Reward: 223.087. Training.
[INFO] CarBehaviour. Step: 4900000. Time Elapsed: 5438.903 s. Mean Reward: -71.209. Std of Reward: 197.954. Training.
[INFO] CarBehaviour. Step: 5000000. Time Elapsed: 5536.375 s. Mean Reward: -83.424. Std of Reward: 188.374. Training.
[INFO] Exported results\mlagents1\CarBehaviour\CarBehaviour-4999830.onnx
[INFO] CarBehaviour. Step: 5100000. Time Elapsed: 5633.252 s. Mean Reward: -99.596. Std of Reward: 192.804. Training.
[INFO] CarBehaviour. Step: 5200000. Time Elapsed: 5728.199 s. Mean Reward: -84.441. Std of Reward: 205.504. Training.
[INFO] CarBehaviour. Step: 5300000. Time Elapsed: 5822.570 s. Mean Reward: -87.887. Std of Reward: 178.291. Training.
[INFO] CarBehaviour. Step: 5400000. Time Elapsed: 5916.263 s. Mean Reward: -67.072. Std of Reward: 171.992. Training.
[INFO] CarBehaviour. Step: 5500000. Time Elapsed: 6011.229 s. Mean Reward: -75.982. Std of Reward: 159.282. Training.
[INFO] CarBehaviour. Step: 5600000. Time Elapsed: 6105.291 s. Mean Reward: -94.287. Std of Reward: 178.745. Training.
[INFO] CarBehaviour. Step: 5700000. Time Elapsed: 6197.936 s. Mean Reward: -107.858. Std of Reward: 185.582. Training.
[INFO] CarBehaviour. Step: 5800000. Time Elapsed: 6293.844 s. Mean Reward: -96.976. Std of Reward: 191.394. Training.
[INFO] CarBehaviour. Step: 5900000. Time Elapsed: 6388.978 s. Mean Reward: -87.319. Std of Reward: 179.422. Training.
[INFO] CarBehaviour. Step: 6000000. Time Elapsed: 6482.283 s. Mean Reward: -80.371. Std of Reward: 176.388. Training.
[INFO] Exported results\mlagents1\CarBehaviour\CarBehaviour-5999909.onnx
[INFO] CarBehaviour. Step: 6100000. Time Elapsed: 6572.174 s. Mean Reward: -92.133. Std of Reward: 162.141. Training.
[INFO] CarBehaviour. Step: 6200000. Time Elapsed: 6671.574 s. Mean Reward: -82.007. Std of Reward: 153.439. Training.
[INFO] CarBehaviour. Step: 6300000. Time Elapsed: 6766.448 s. Mean Reward: -74.190. Std of Reward: 204.158. Training.
```

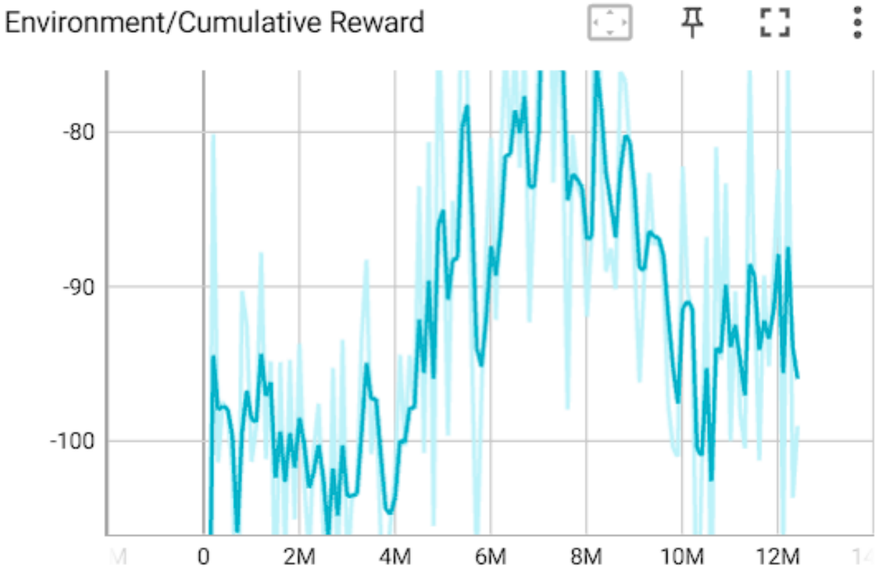
Mean reward

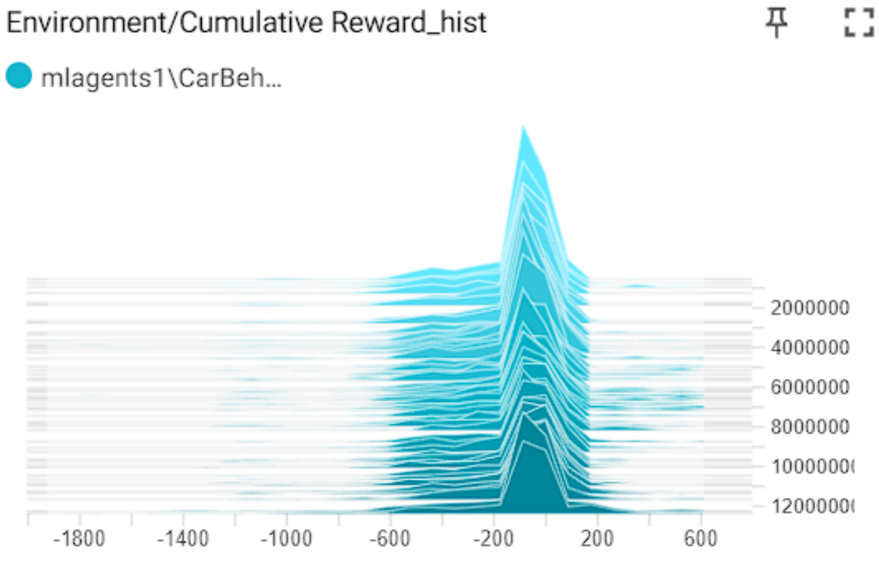
在每10萬步裡, Agents 拿到的Reward

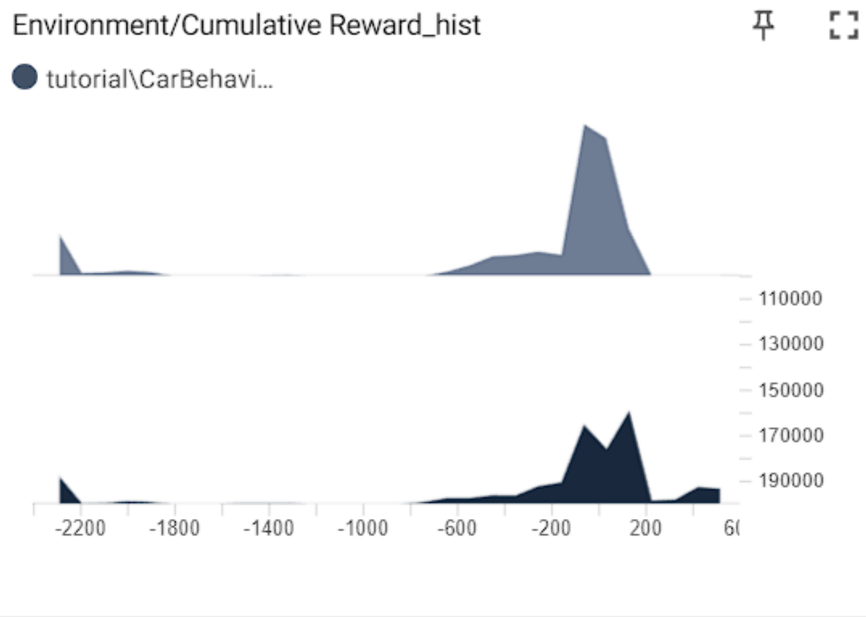
Std reward

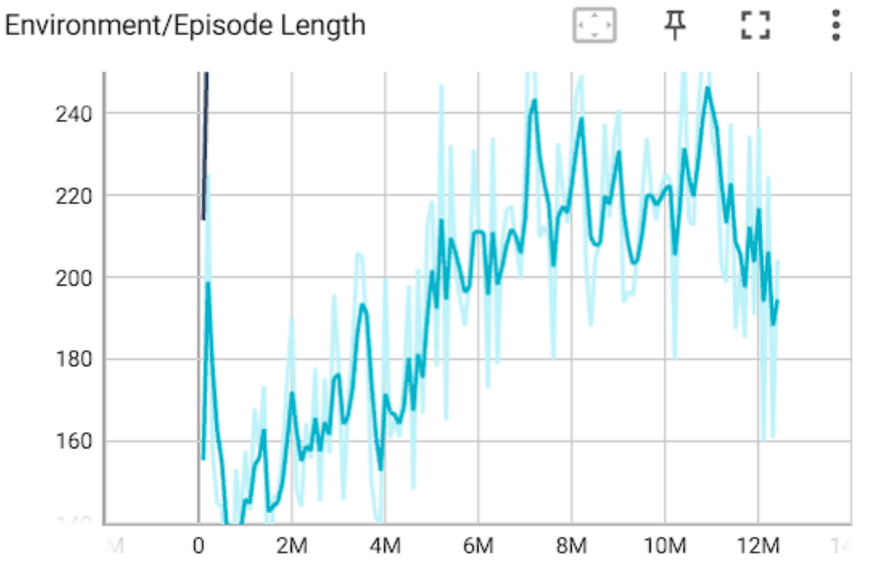
衡量平均獎勵分佈的量度。較大的值表示收到的獎勵變化很大，而較小的值則相反。

Tensorboard

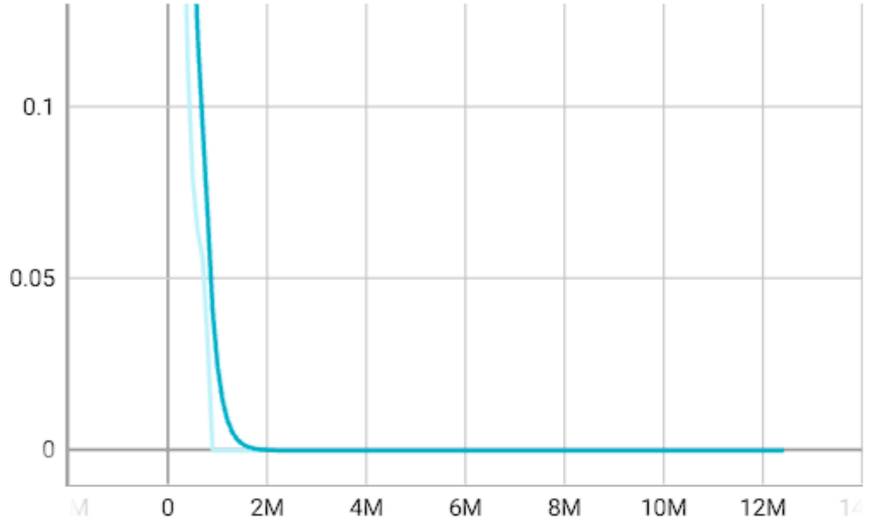

Environment/Cumulative Reward
-80
-90
-100
0
2M
4M
6M
8M
10M
12M


Environment/Cumulative Reward_hist
mlagents1\CarBeh...
2000000
4000000
6000000
8000000
-1800
-1400
-1000
-600
-200
200
600

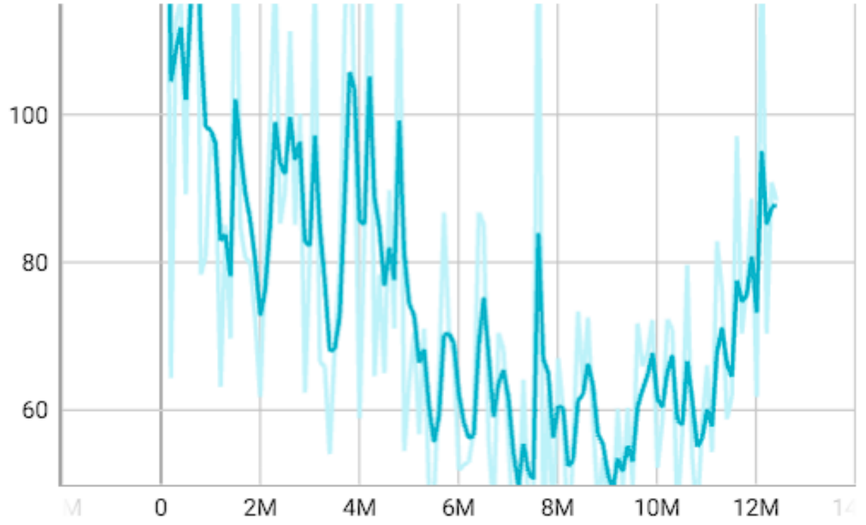

Environment/Cumulative Reward_hist
tutorial\CarBehavi...
110000
130000
150000
170000
190000
-2200
-1800
-1400
-1000
-600
-200
200


Environment/Episode Length
240
220
200
180
160
0
2M
4M
6M
8M
10M
12M

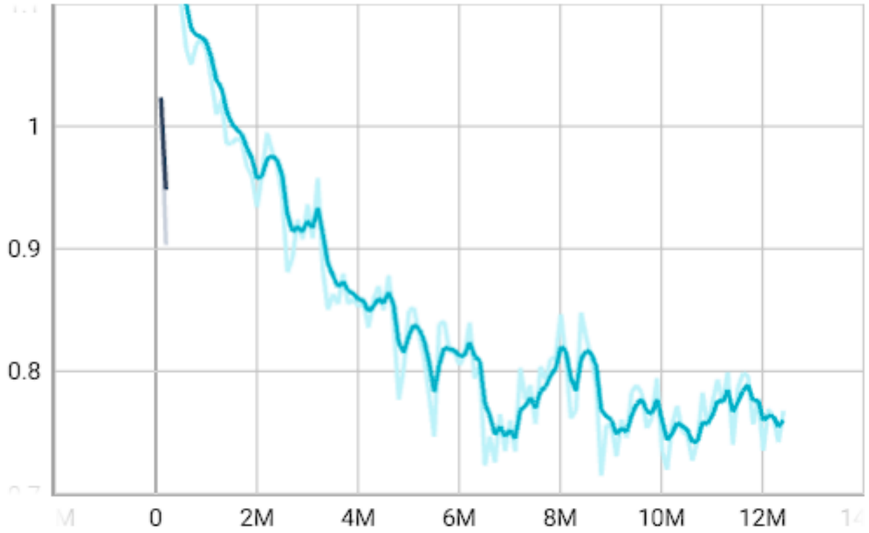
Losses/Pretraining Loss



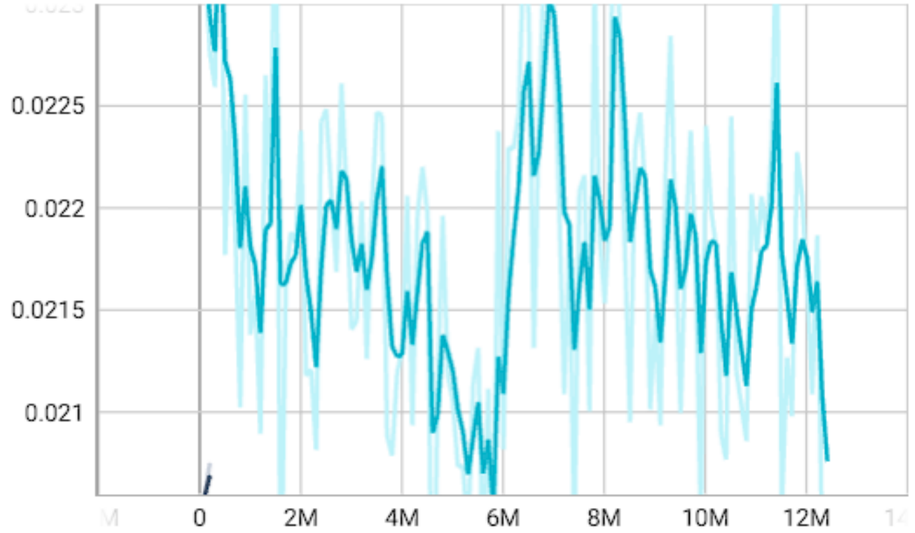
Losses/Value Loss



Losses/GAIL Loss



Losses/Policy Loss



成果展示

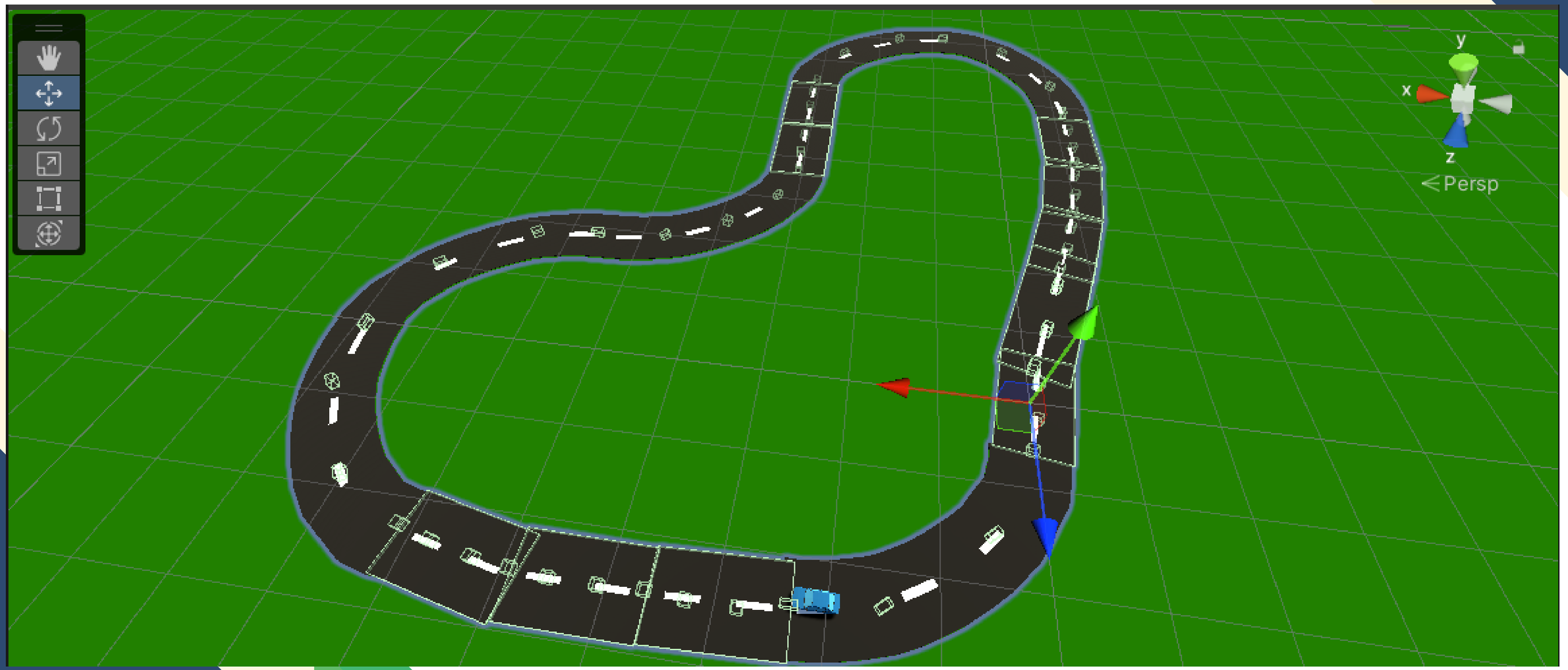
https://www.youtube.com/watch?v=KmknP3KVYIs

貢獻與後續

貢獻

1. 更改環境，更改與優化參數
2. Tensorboard 分析
3. 優化car spawner
4. 自己的模型
5. 自走車半成品(無法轉彎)

自走車原理



如何改善

訓練跑滿5千萬步

更好的獎勵機制

謝謝各位