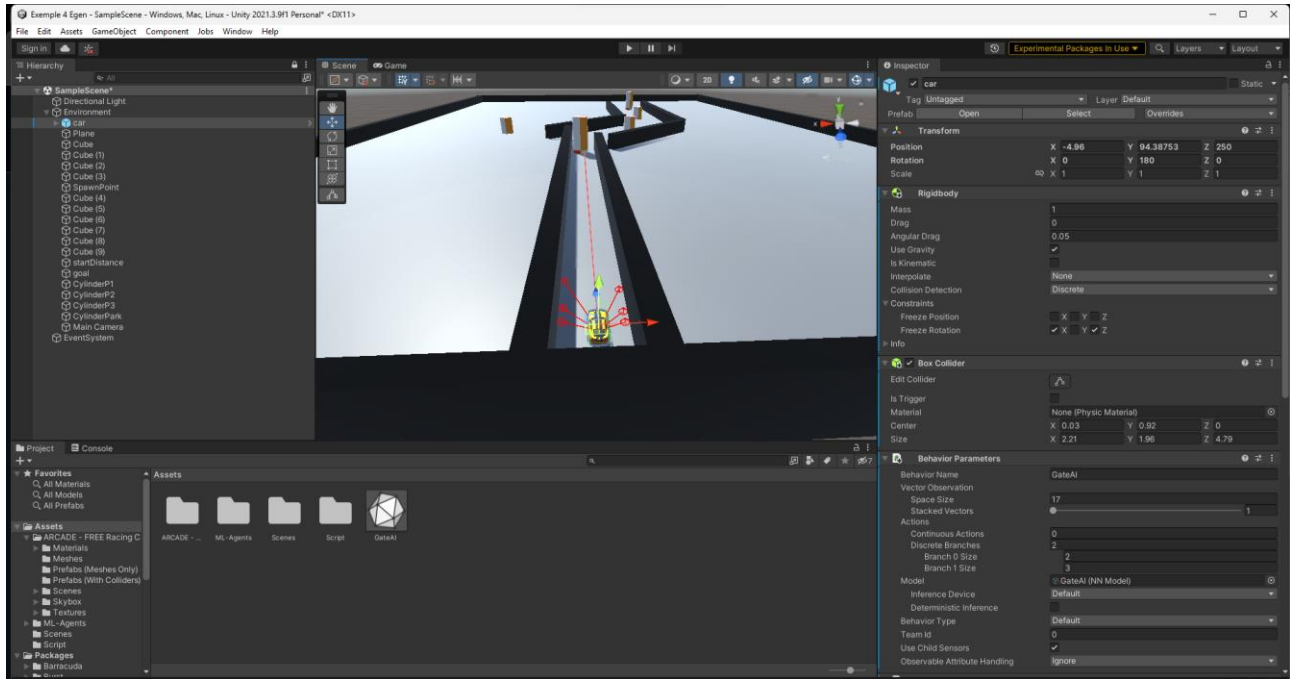


Preskriptiv analytik Slutprojekt med Unity av Ammar Mandawi

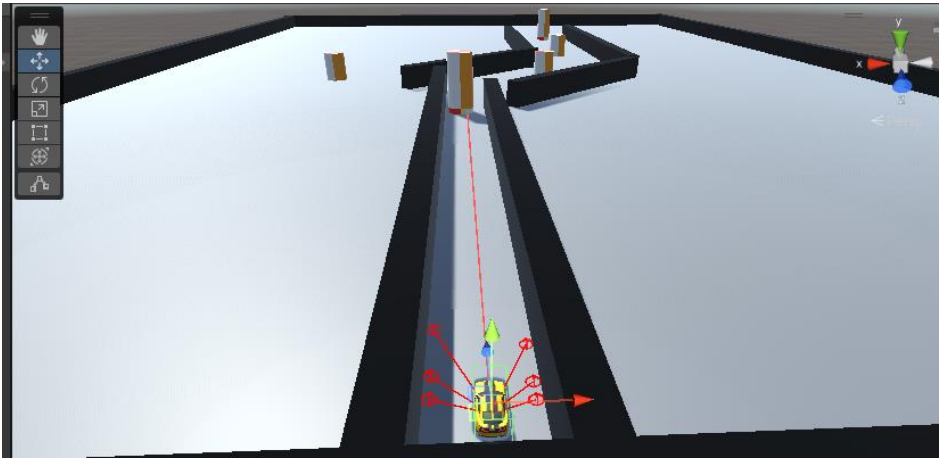
Bil som navigerar genom en 3D-miljö med Unity ML-agent implementation



Den 2023/12

Contents

Bil som navigerar genom en 3D-miljö med Unity ML-agent implementation.....	1
Konfigurationsanalys:	3
Kodanalys:.....	3
Sammanfattningsvis:.....	3
Two different methods Basic.yml and Walker.yml	4
Kod/Script beskrivning:.....	5
Terminalkommandon för att använda Anaconda med ML-Agents för träning:.....	7
Länk för projekt:	8
Github länk för kod alla material för kursen:	8
ml-agents /config /ppo/ länk:	8
Bilder om hela projekten:	8



Projektbeskrivning:

Projektet involverar en bil som navigerar genom en 3D-miljö med målet att undvika kollision med vägen och nå slutet av den. Unity ML-agent implementeras för att lära bilen undvika kollisioner och belönas när den når målet eller träffar en cylinder. Målet är att träningen leder till att bilen navigerar säkert och effektivt.

Konfigurationsanalys:

Jämförelse mellan walker.yml och Basic.yml visar skillnader i parametrar som påverkar träningsresultaten. walker.yml presterar snabbare och stabilare med högre batch-storlek, mindre buffer, normalisering, fler dolda enheter, högre gamma, längre tidshorisont och längre träningssteg.

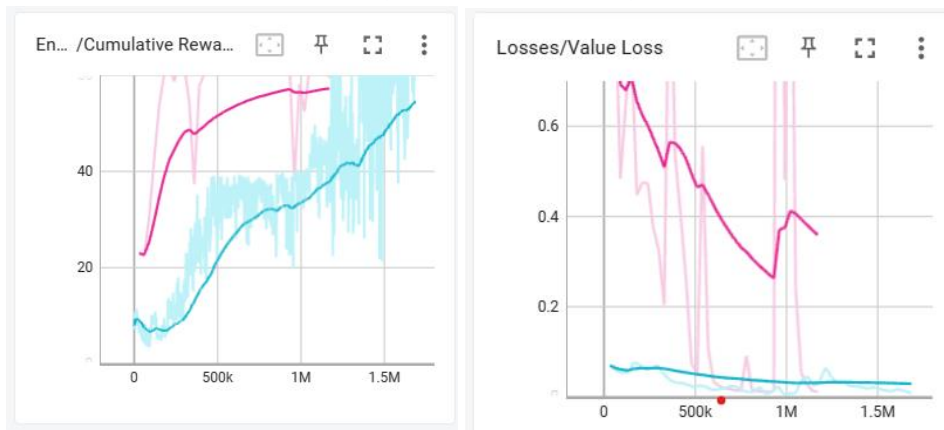
Kodanalys:

Unity ML-Agents Agent Script används för att styra bilens beteende. Initialisering och parametrar sätts upp, handlingar och observationer hanteras, och olika metoder styr rörelse och logik. Kodstrukturen möjliggör träning av en intelligent agent genom maskininlärning.

Sammanfattningsvis:

Projektet går ut på spelmekanik och maskininlärning för att skapa en autonom bil, och konfigurationsanalysen belyser hur olika inställningar påverkar träningsresultaten. Kodstrukturen ger en översikt över agentens beteende och styrning. Problem löst med två olika metoder:

Två olika metoder Basic.yml och Walker.yml



```
! Walker.yml
1 behaviors:
2   GateAI:
3     trainer_type: ppo
4     hyperparameters:
5       batch_size: 2048
6       buffer_size: 20480
7       learning_rate: 0.0003
8       beta: 0.005
9       epsilon: 0.2
10      lambda: 0.95
11      num_epoch: 3
12      learning_rate_schedule: linear
13    network_settings:
14      normalize: true
15      hidden_units: 256
16      num_layers: 3
17      vis_encode_type: simple
18    reward_signals:
19      extrinsic:
20        gamma: 0.995
21        strength: 1.0
22      keep_checkpoints: 5
23      max_steps: 3000000
24      time_horizon: 1000
25      summary_freq: 30000
26

! Basic.yml
1 behaviors:
2   GateAI:
3     trainer_type: ppo
4     hyperparameters:
5       batch_size: 512
6       buffer_size: 32768
7       learning_rate: 0.0003
8       beta: 0.005
9       epsilon: 0.2
10      lambda: 0.95
11      num_epoch: 3
12      learning_rate_schedule: linear
13    network_settings:
14      normalize: false
15      hidden_units: 20
16      num_layers: 3
17      vis_encode_type: simple
18    reward_signals:
19      extrinsic:
20        gamma: 0.9
21        strength: 1.0
22      keep_checkpoints: 5
23      max_steps: 5000000
24      time_horizon: 3
25      summary_freq: 2000
26
```

Batch-storlek:

walker.yml: Använder en större batch-storlek på 2048. Detta betyder att fler exempel används i varje träningsuppdatering.

Basic.yml: Använder en mindre batch-storlek på 512.

Buffer-storlek:

walker.yml: Använder en mindre upplevelsebuffer på 20480. Bufferstorleken påverkar minnet för träningsupplevelser.

Basic.yml: Använder en större upplevelsebuffer på 32768.

Normalisering:

walker.yml: Normaliserar observationerna (sätter dem på en gemensam skala).

Basic.yml: Använder inte normalisering av observationerna.

Dolda enheter:

walker.yml: Har fler dolda enheter (256) i varje lager av det neurala nätverket.

Basic.yml: Har färre dolda enheter (20).

Gamma (rabattfaktor):

walker.yml: Använder en högre rabattfaktor (0.995) för den extrinsiska belöningssignalen. Detta ger högre viktning åt framtida belöningar.

Basic.yml: Använder en lägre rabattfaktor (0.9).

Maximala steg:

walker.yml: Anger ett högre maximalt antal träningssteg (30 miljoner).

Basic.yml: Använder ett lägre maximalt antal träningssteg (5 miljoner).

Tidshorisont:

walker.yml: Använder en längre tidshorisont (1000). Detta påverkar hur långt in i framtiden agenten ser när den tar beslut.

Basic.yml: Använder en kortare tidshorisont (3).

Dessa konfigurationsskillnader har påverkat resultaten så att *walker.yml* har varit mycket snabbare och uppnått målet på ett mer stabilt sätt och kortare tid jämfört med *Basic.yml*.

Kod/Script beskrivning:

```
using UnityEngine.UI;

public class DriveScript : Agent
{
    // Initial positions and parameters
    Vector3 StartPosition;
    Quaternion StartingRotation;
    float startDistance = 0;

    // Movement parameters
    float speed = 0.2f;
    float turnSpeed = 60.0f;
    float maxSpeed = 10.0f;

    // Components and game objects
    Rigidbody m_Rigidbody;
    public GameObject GoalDistance;
    public GameObject Waypoint;
    public GameObject C_point;
    public GameObject treasure;
    public GameObject StartDistance;

    // Variables for observations
    Vector3 carPosition;
    float carSpeed;
    float updatedDistance;

    // Timer for training episodes
    float trainingTimer = 0;

    // UI element for text information
    private Text TextInfo;

    // Unity callback for initialization
    void Start()
    {
        // Initialize starting positions
        StartPosition = transform.localPosition;
        StartingRotation = transform.localRotation;

        // Initialize rigidbody and start distance
        m_Rigidbody = GetComponent<Rigidbody>();
        startDistance = GetGoalDistance();

        // Initialize cylinders (points)
        StartAllCylinders();

        // Initialize all cylinders (points)
        void StartAllCylinders()
        {
            // Find all point game objects with tag "Point"
            GameObject[] points = GameObject.FindGameObjectsWithTag("Point");

            // Define preset coordinates for points
            float[] xCoordinates = new float[points.Length];
            float[] yCoordinates = new float[points.Length];
            float[] zCoordinates = new float[points.Length];

            // Set positions for each point
            for (int i = 0; i < points.Length; i++)
            {
                Vector3 newPosition = new Vector3(
                    points[i].transform.localPosition.x,
                    points[i].transform.localPosition.y,
                    points[i].transform.localPosition.z
                );
                xCoordinates[i] = newPosition.x;
                yCoordinates[i] = newPosition.y;
                zCoordinates[i] = newPosition.z;
            }
        }

        private void FixedUpdate()
        {
            // Update distance to goal
            updatedDistance = GetGoalDistance();

            // Check if agent is getting closer to goal
            if (updatedDistance < startDistance)
            {
                AddReward(0.1f); // Add reward for getting closer
                trainingTimer = 0;
                startDistance = updatedDistance;
            }
            else
            {
                AddReward(-0.1f); // Penalize for not getting closer
                trainingTimer++;
            }

            // End episode if training time is up
            if (trainingTimer > 500)
            {
                trainingTimer = 0;
                EndEpisode();
            }

            // Update car position and speed
            var vel = m_Rigidbody.velocity;
            carPosition = transform.position;
            carSpeed = vel.magnitude;

            // Request decision for the next step
            RequestDecision();
        }
    }
}
```

Unity ML-Agents Agent Script

Initialization och Parametrar

- **StartPosition**: Lagrar agentens initiala position.
- **StartingRotation**: Lagrar agentens initiala rotation.
- **startDistance**: Representerar det initiala avståndet till målet.
- **speed**: Agentens hastighet.
- **turnSpeed**: Hastighet för att svänga agenten.
- **maxSpeed**: Maximal hastighet för agenten.

- **m_Rigidbody:** Rigidbody-komponenten för agenten för fysiska interaktioner.
- Olika spelobjekt (**GoalDistance**, **Waypoint**, **C_point**, **treasure**, **StartDistance**) är offentliga fält för referenser i Unity-editorn.

Handlingar och Observationer

- **OnActionReceived:** Denna metod kallas när agenten får en handling. Den tolkar diskreta handlingar (t.ex. röra sig framåt, röra sig bakåt, svänga vänster, svänga höger) och utför motsvarande handlingar.
- **CollectObservations:** Denna metod används för att samla observationer från omgivningen för agenten. Den inkluderar information som agentens position, waypoint-position, checkpoint-position, hastighet, avstånd till waypoint och rotation.

Start-metod

- **Start:** Initialiserar startposition, rotation och andra parametrar. Den ställer också in de initiala positionerna för cylindrarna (punkterna) med hjälp av **StartAllCylinders**.

StartAllCylinders-metod

- **StartAllCylinders:** Positionerar cylindrarna (punkterna) på fördefinierade koordinater. Koordinaterna anges i tre arrayer (**xCoordinates**, **yCoordinates**, **zCoordinates**).

FixedUpdate-metod

- **FixedUpdate:** Denna metod kallas vid fasta intervall och hanterar agentens huvudlogik. Den beräknar avståndet till målet, belönar eller straffar agenten baserat på dess rörelse och kontrollerar träningsperioden. Den uppdaterar också bilens position och hastighet.

OnEpisodeBegin-metod

- **OnEpisodeBegin:** Kallas i början av en ny episod. Den återställer agentens position och rotation.

Reset-metod

- **Reset:** Återställer agentens position och rotation till det initiala tillståndet och stoppar dess hastighet.

GetGoalDistance-metod

- **GetGoalDistance:** Beräknar avståndet mellan agenten och ett angivet målobjekt.

ScoreCloserToGoal-metod

- **ScoreCloserToGoal:** Belönar eller bestraffar agenten baserat på dess närhet till målet.

TextInfoOutput-metod

- **TextInfoOutput:** Skickar textinformation till ett UI-element.

Heuristic-metod

- **Heuristic:** Ger ett manuellt (heuristiskt) sätt att styra agenten med hjälp av tangentbordsinmatning.

Rörelsemetoder

- **LeftDrive**, **RightDrive**, **DriveForward**, **DriveBackwards:** Metoder som styr agentens rörelse i olika riktningar.

RandomPosition-metod

- RandomPosition: Positionerar cylindrarna (waypoints) slumpmässigt.

otherRandomPosition-metod

- otherRandomPosition: Positionerar slumpmässigt en specifierad kolliderare.

Kollisions- och Trigger-metoder

- OnTriggerEnter: Hanterar händelser när agenten kommer in i en triggerkolliderare. Belönar eller bestraffar agenten baserat på kolliderarens tag.
- OnCollisionExit: Hanterar händelser när agenten lämnar en kollision. Den stoppar agenten och avslutar episoden om den kolliderar med en tagg "plane".

Terminalkommandon för att använda Anaconda med ML-Agents för träning:

```
# Installera ML Agents toolkit 2.3.0 i Unity
#       Window->Packe Manager
#       Tryck på + och "Install from Git URL"
#       Kopiera in följande länk i URL-rutan:
# git+https://github.com/Unity-Technologies/ml-agents.git?path=com.unity.ml-
agents#release_20

# ML Agents trainer API (installera i Anaconda)
# conda create -n mlagentsUnity python=3.8.13
# conda activate mlagentsUnity
# pip install torch==1.8.0
# pip install mlagents==0.30.0

# # Finale projekt
# cd /d D:\2024\Period 1\Autonoma Agenter\lektion 3\Exemple 4 Egen\Training_Walker
# # RobotTestBuild18C5 sista test
# # to start conda
# conda activate mlagents2

# # visa visualisering resultat
# tensorboard --logdir results --port 5004

# # Start game in unity
# mlagents-learn config.yml --run-id="RobotTestBuild18C3" --time-scale=5 --resume
```

```
# # 10 med skärm
# mlagents-learn config.yml --run-id="MyAi-5" --env="D:\2024\Period 1\Autonoma
Agenter\lektion 3\Exemple 4 Egen\Training\newBuild\Exemple 3 auto drive car.exe" --
width=800 --height=512 --time-scale=10 --num-envs=1 --resume

# # 10 utan skärm
# mlagents-learn config.yml --run-id="RobotTestBuild18C3" --env="D:\2024\Period
1\Autonoma Agenter\Projekt1\ArmRobot\Training\build4X\ArmRobot.exe" --no-graphics -
-time-scale=5 --num-envs=5 --resume

# # build
# # 10 X 10 gånger snabbare utan screen
# mlagents-learn config.yml --run-id="MyAi-walker" --env="D:\2024\Period 1\Autonoma
Agenter\lektion 3\Exemple 4 Egen\Training_Walker\newBuild\Exemple 3 auto drive
car.exe" --no-graphics --time-scale=10 --num-envs=5
```

Länk för projekt:

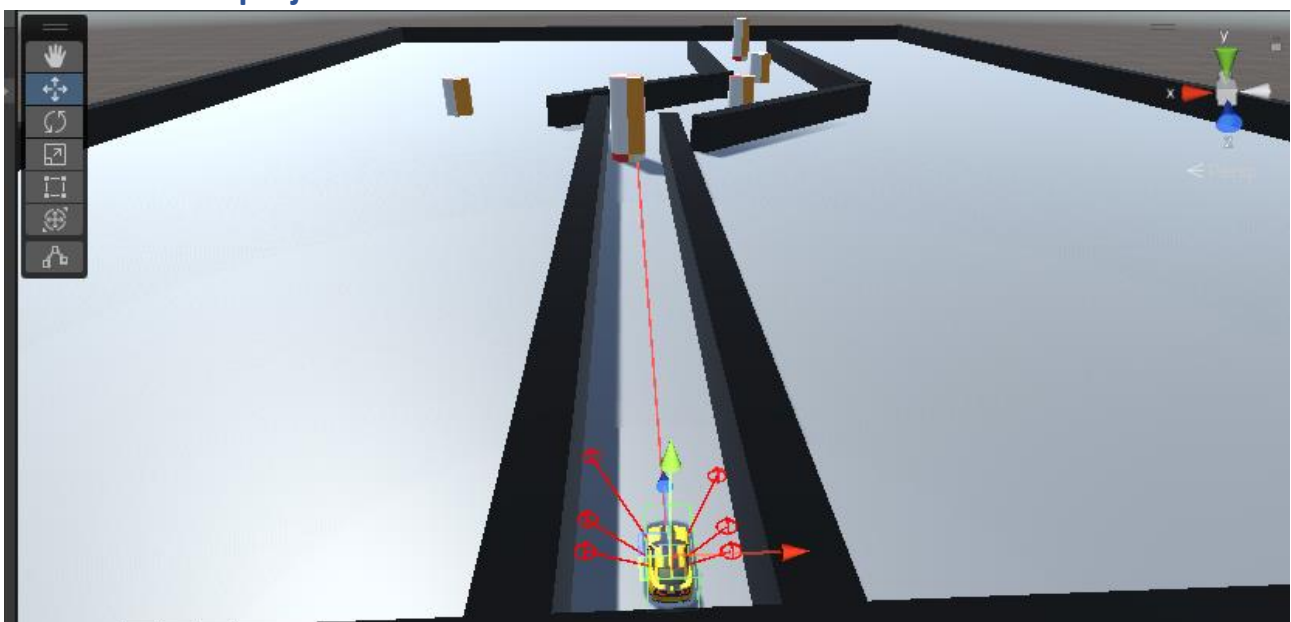
Github länk för kode alla material för kursen:

https://github.com/Ammar-Man/Preskriptiv_analytik_2023_p2/tree/main/Final_Project

ml-agents /config /ppo/ länk:

<https://github.com/Unity-Technologies/ml-agents/tree/develop/config/ppo>

Bilder om hele projekten:



Behavior Parameters

?

Behavior Name

GateAI

Vector Observation

Space Size

17

Stacked Vectors

1

Actions

Continuous Actions

0

Discrete Branches

2

Branch 0 Size

2

Branch 1 Size

3

Model

GateAI (NN Model)

Inference Device

Default

Deterministic Inference

Behavior Type

Default

Team Id

0

Use Child Sensors

Observable Attribute Handling

Ignore

Ray Perception Sensor 3D

?

Sensor Name

RayPerceptionSensor

▼ Detectable Tags

3

Element 0

wall

Element 1

waypoint

Element 2

point

+

-

Rays Per Direction

3

Max Ray Degrees

70

Sphere Cast Radius

0.5

Ray Length

138

Ray Layer Mask

Default, TransparentFX, Water, UI

Stacked Raycasts

1

Start Vertical Offset

0

End Vertical Offset

0

Alternating Ray Order

Debug Gizmos

Ray Hit Color

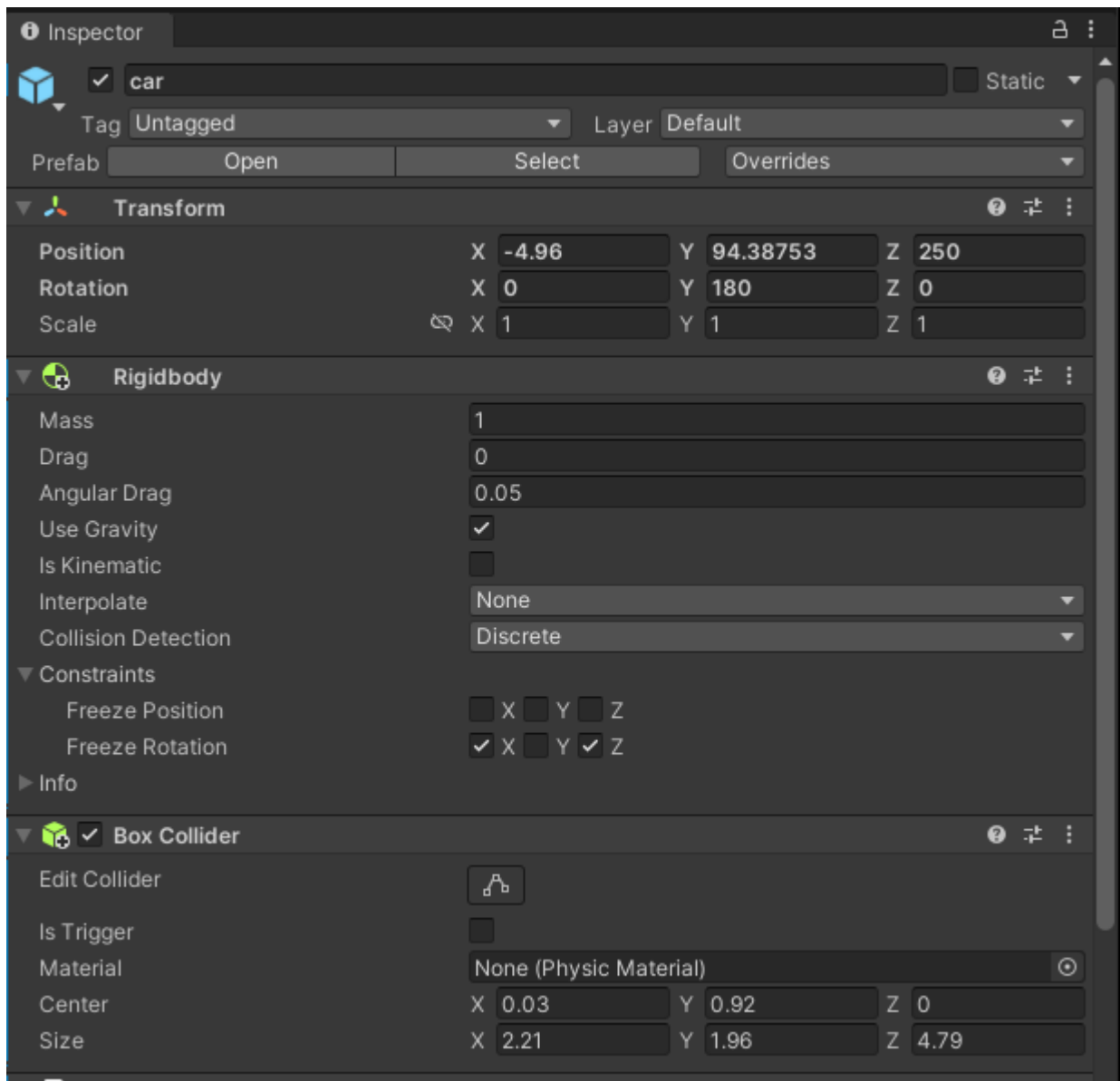
Ray Miss Color

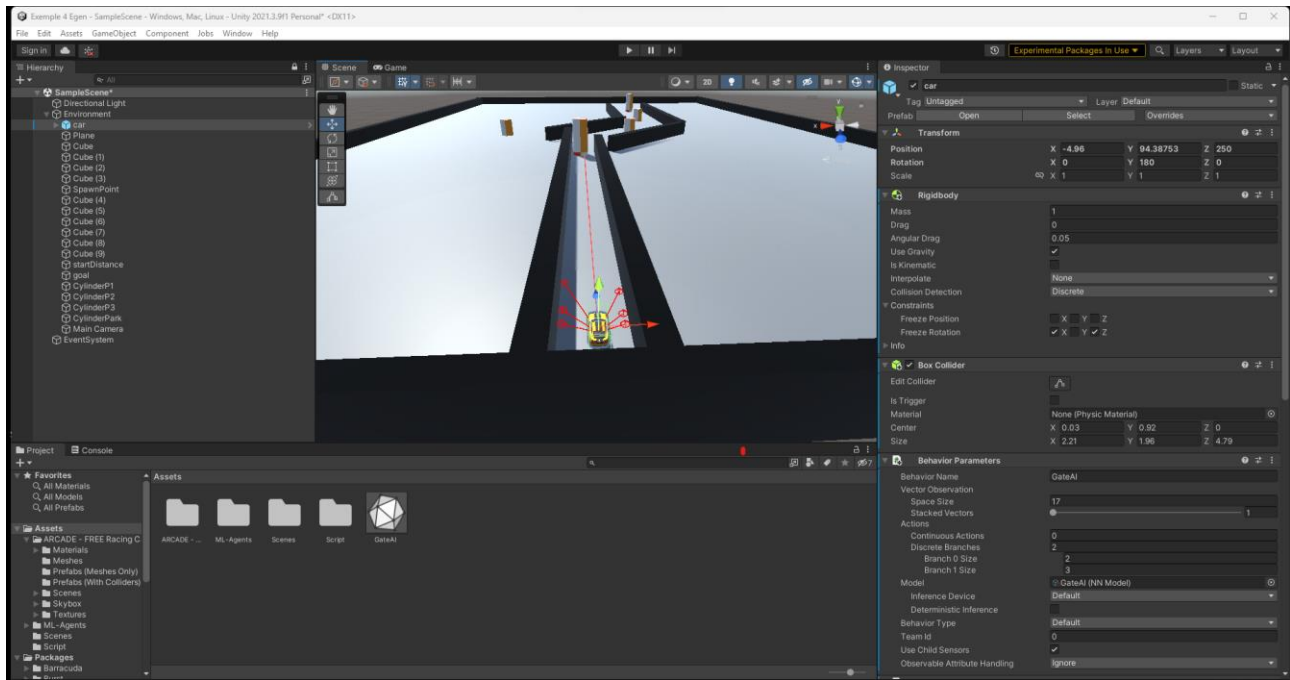
Behavior Parameters

?

Behavior Name	GateAI
Vector Observation	
Space Size	17
Stacked Vectors	<div><div></div><div>1</div></div>
Actions	
Continuous Actions	0
Discrete Branches	2
Branch 0 Size	2
Branch 1 Size	3
Model	<div><div></div>GateAI (NN Model)<div></div></div>
Inference Device	Default
Deterministic Inference	<input type="checkbox"/>
Behavior Type	Default
Team Id	0
Use Child Sensors	<input checked="" type="checkbox"/>
Observable Attribute Handling	Ignore







```

[INFO] GateAI. Step: 540000. Time Elapsed: 1415.707 s. Mean Reward: 62.939. Std of Reward: 0.209. Training.
[INFO] GateAI. Step: 570000. Time Elapsed: 1499.312 s. Mean Reward: 60.942. Std of Reward: 14.030. Training.
[INFO] GateAI. Step: 600000. Time Elapsed: 1575.638 s. Mean Reward: 62.909. Std of Reward: 0.150. Training.
[INFO] GateAI. Step: 630000. Time Elapsed: 1660.751 s. Mean Reward: 62.825. Std of Reward: 0.092. Training.
[INFO] GateAI. Step: 660000. Time Elapsed: 1739.094 s. Mean Reward: 62.835. Std of Reward: 0.158. Training.
[INFO] GateAI. Step: 690000. Time Elapsed: 1825.738 s. Mean Reward: 62.823. Std of Reward: 0.090. Training.
[INFO] GateAI. Step: 720000. Time Elapsed: 1905.703 s. Mean Reward: 62.784. Std of Reward: 0.135. Training.
[INFO] GateAI. Step: 750000. Time Elapsed: 1985.603 s. Mean Reward: 62.828. Std of Reward: 0.153. Training.
[INFO] GateAI. Step: 780000. Time Elapsed: 2069.353 s. Mean Reward: 60.804. Std of Reward: 14.676. Training.
[INFO] GateAI. Step: 810000. Time Elapsed: 2147.070 s. Mean Reward: 62.847. Std of Reward: 0.127. Training.
[INFO] GateAI. Step: 840000. Time Elapsed: 2231.762 s. Mean Reward: 63.000. Std of Reward: 0.225. Training.
[INFO] GateAI. Step: 870000. Time Elapsed: 2308.900 s. Mean Reward: 63.051. Std of Reward: 0.172. Training.
[INFO] GateAI. Step: 900000. Time Elapsed: 2394.800 s. Mean Reward: 63.176. Std of Reward: 0.289. Training.
[INFO] GateAI. Step: 930000. Time Elapsed: 2470.612 s. Mean Reward: 63.235. Std of Reward: 0.224. Training.
[INFO] GateAI. Step: 960000. Time Elapsed: 2555.448 s. Mean Reward: 37.871. Std of Reward: 27.911. Training.
[INFO] GateAI. Step: 990000. Time Elapsed: 2631.676 s. Mean Reward: 58.154. Std of Reward: 16.038. Training.
===== Diagnostic Run torch.onnx.export version 2.0.1+cpu =====
verbose: False, log level: Level.ERROR
===== 0 NONE 0 NOTE 0 WARNING 0 ERROR =====

[INFO] Exported results\MyAi-walker\GateAI\GateAI-999807.onnx
[INFO] GateAI. Step: 1020000. Time Elapsed: 2716.938 s. Mean Reward: 52.421. Std of Reward: 22.371. Training.
[INFO] GateAI. Step: 1050000. Time Elapsed: 2794.409 s. Mean Reward: 61.507. Std of Reward: 8.433. Training.
[INFO] GateAI. Step: 1080000. Time Elapsed: 2871.780 s. Mean Reward: 63.146. Std of Reward: 0.171. Training.
[INFO] GateAI. Step: 1110000. Time Elapsed: 2957.151 s. Mean Reward: 63.141. Std of Reward: 0.190. Training.
[INFO] GateAI. Step: 1140000. Time Elapsed: 3031.932 s. Mean Reward: 63.137. Std of Reward: 0.188. Training.
[INFO] GateAI. Step: 1170000. Time Elapsed: 3120.554 s. Mean Reward: 59.427. Std of Reward: 18.892. Training.
[INFO] Learning was interrupted. Please wait while the graph is generated.
===== Diagnostic Run torch.onnx.export version 2.0.1+cpu =====
verbose: False, log level: Level.ERROR
===== 0 NONE 0 NOTE 0 WARNING 0 ERROR =====

[INFO] Exported results\MyAi-walker\GateAI\GateAI-1183259.onnx
[INFO] Copied results\MyAi-walker\GateAI\GateAI-1183259.onnx to results\MyAi-walker\GateAI.onnx.

(mlagents2) D:\2024\Period 1\Autonoma Agenter\lektion 3\Exemple 4 Egen\Training_Walker>

```

```
(base) C:\Users\Admin>cd /d D:\2024\Period 1\Autonoma Agenter\lektion 3\Exemple 4 Egen\Training_Walker
```

```
(base) D:\2024\Period 1\Autonoma Agenter\lektion 3\Exemple 4 Egen\Training_Walker>tensorboard --logdir results --port 5004
```

TensorFlow installation not found - running with reduced feature set.

Serving TensorBoard on localhost; to expose to the network, use a proxy or pass --bind_all

TensorBoard 2.11.0 at http://localhost:5004/ (Press CTRL+C to quit)

E1207 14:33:07.993248 9888 _internal.py:225] 127.0.0.1 - - [07/Dec/2023 14:33:07] code 505, message Invalid HTTP version (2.0)

W1207 14:49:02.231091 16232 plugin_event_multiplexer.py:265] Deleting accumulator 'MyAi-5\\GateAI'

