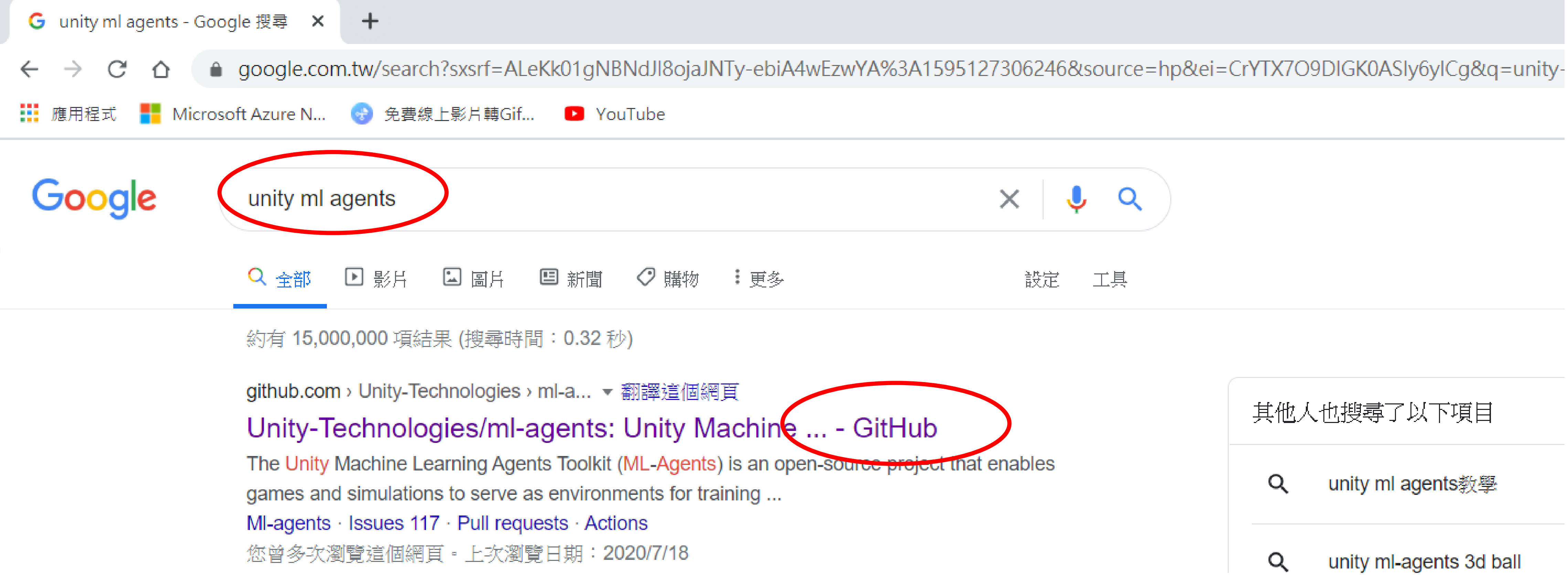
1. Download ML Agent

Go to Unity ML Agent Git hub.



unity ml agents - Google 搜尋

google.com.tw/search?sxsrf=ALeKk01gNBNdJl8ojaJNTy-ebiA4wEzwYA%3A1595127306246&source=hp&ei=CrYTX7O9DIGK0ASly6ylCg&q=unity-

unity ml agents

約有 15,000,000 項結果 (搜尋時間：0.32 秒)

github.com › Unity-Technologies › ml-a... ▾ 翻譯這個網頁

Unity-Technologies/ml-agents: Unity Machine ... - GitHub

The Unity Machine Learning Agents Toolkit (ML-Agents) is an open-source project that enables games and simulations to serve as environments for training ...

Ml-agents · Issues 117 · Pull requests · Actions

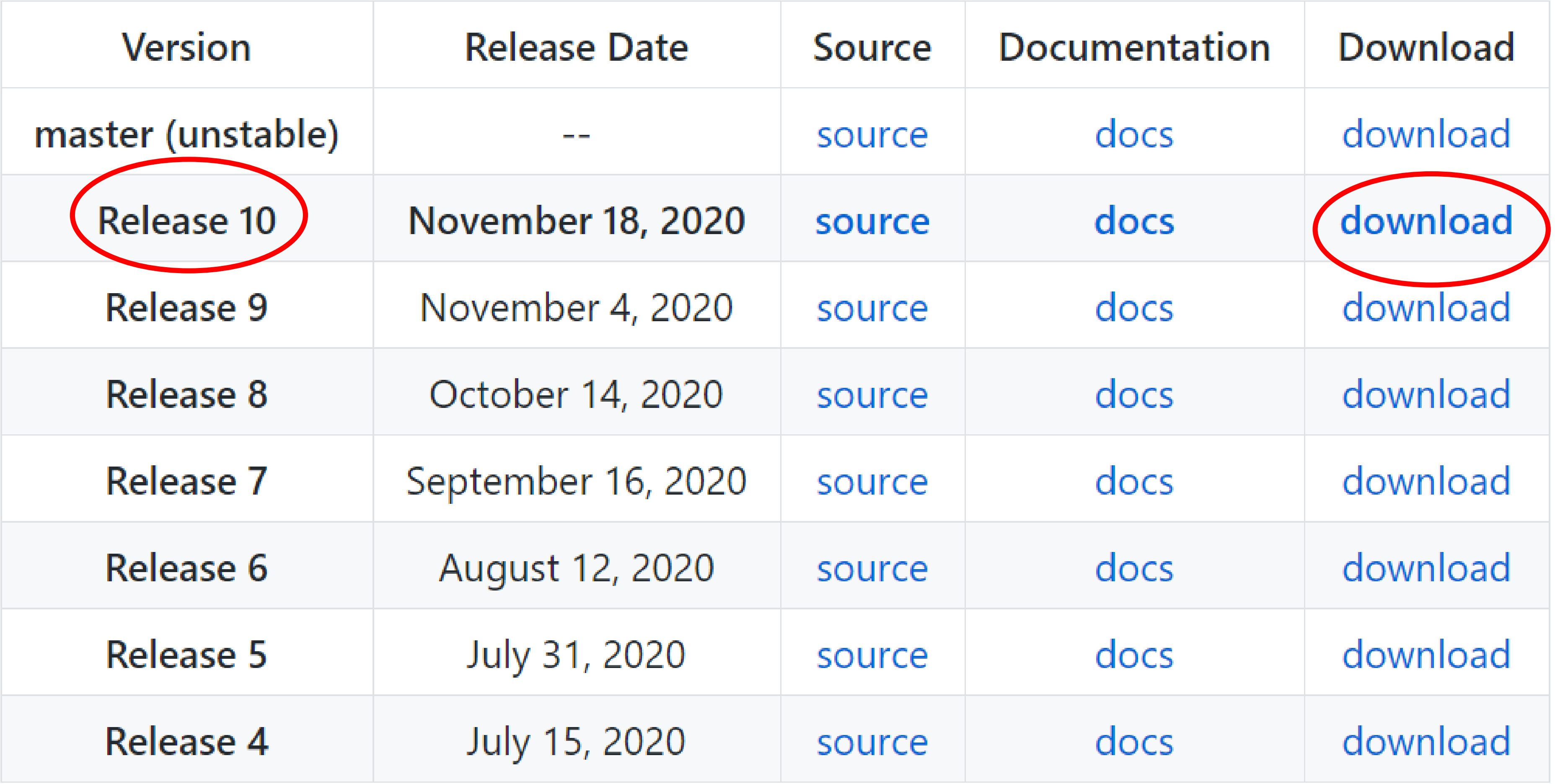
您曾多次瀏覽這個網頁。上次瀏覽日期：2020/7/18

其他人也搜尋了以下項目

unity ml agents教學

unity ml-agents 3d ball

1. Download ML Agent

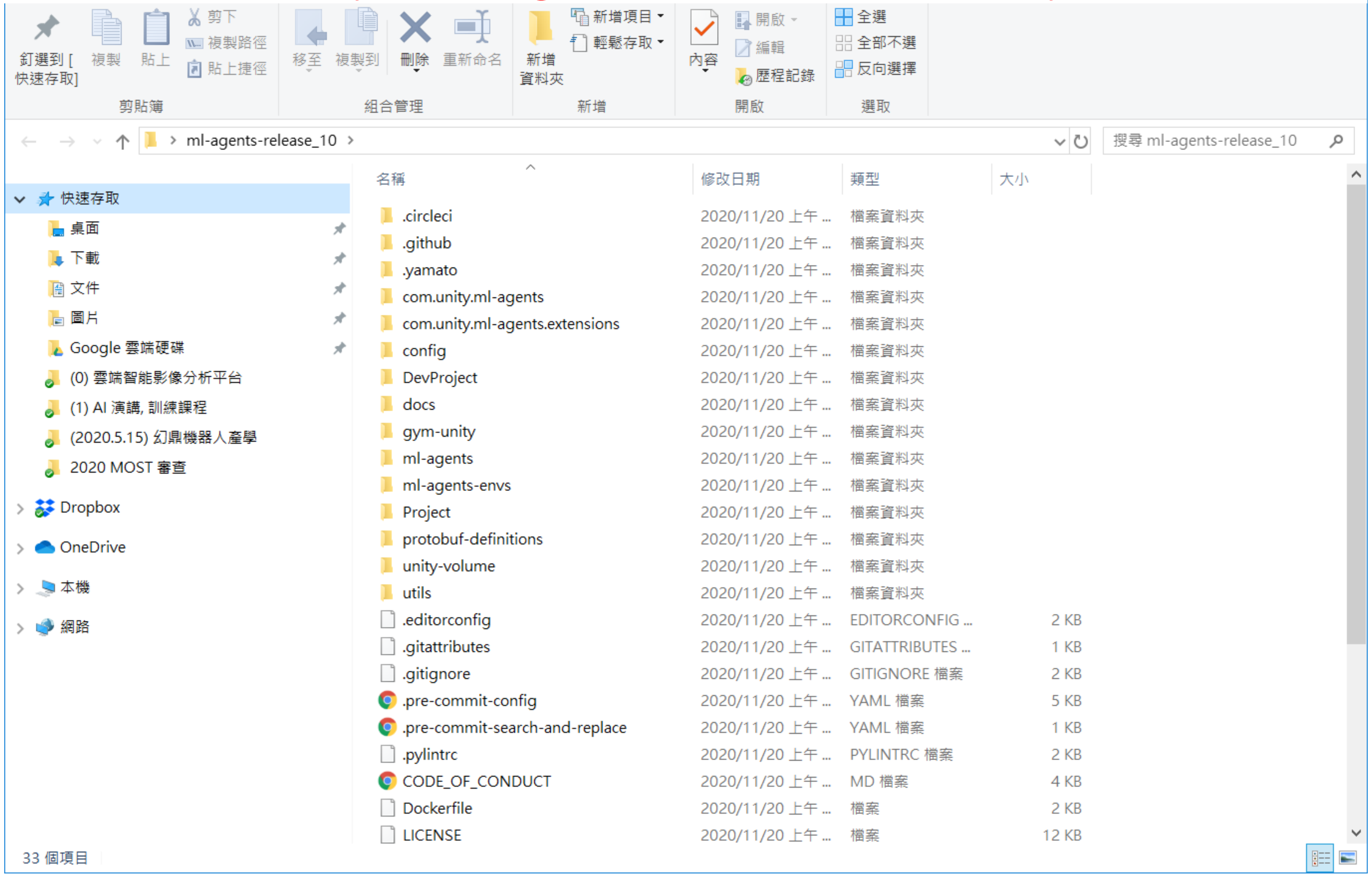


Version	Release Date	Source	Documentation	Download
master (unstable)	--	source	docs	download
Release 10	November 18, 2020	source	docs	download
Release 9	November 4, 2020	source	docs	download
Release 8	October 14, 2020	source	docs	download
Release 7	September 16, 2020	source	docs	download
Release 6	August 12, 2020	source	docs	download
Release 5	July 31, 2020	source	docs	download
Release 4	July 15, 2020	source	docs	download

https://github.com/Unity-Technologies/ml-agents

1. Download ML Agent

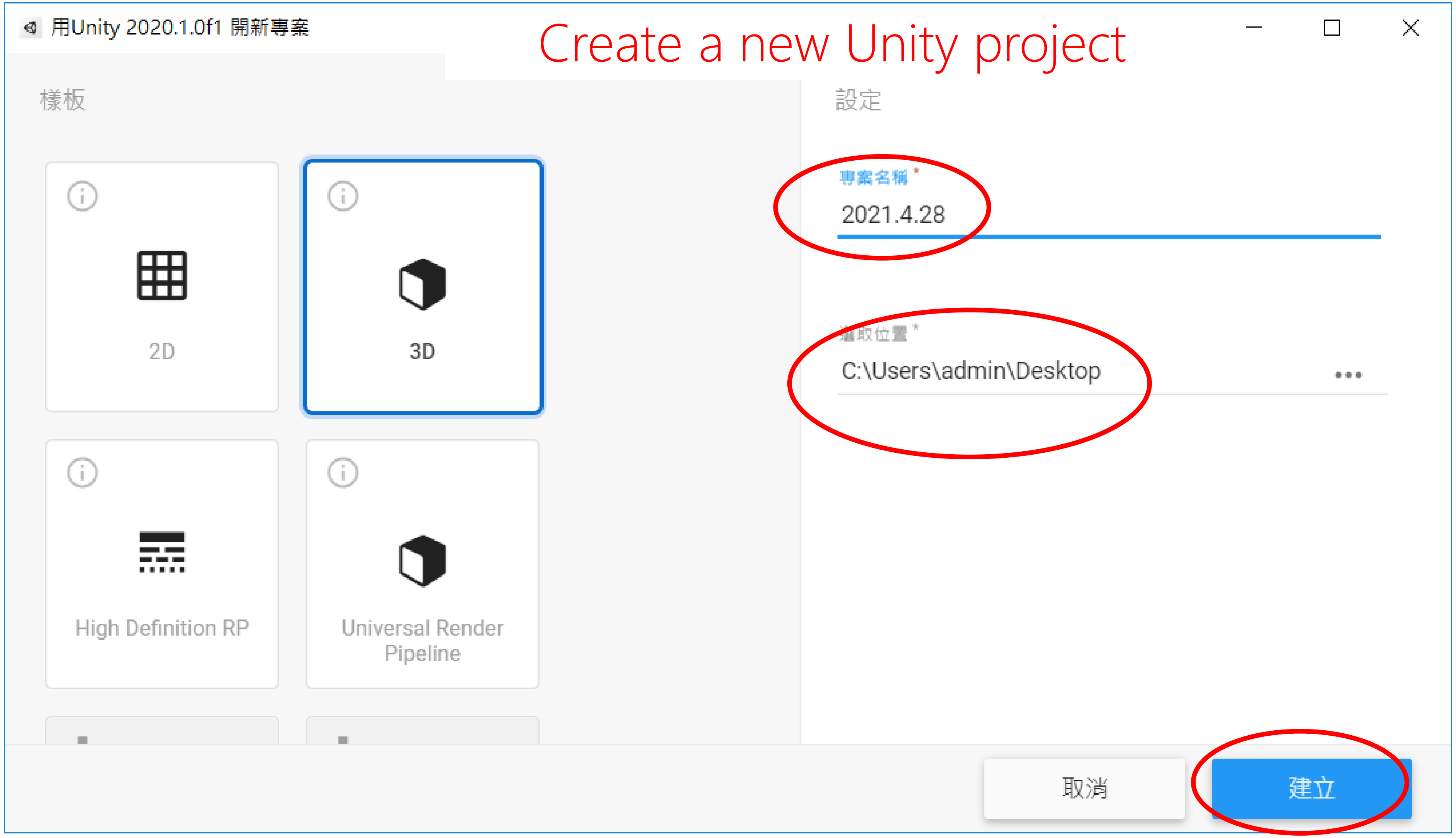
Unzip ML Agent folder to Desktop



名稱	修改日期	類型	大小
.circleci	2020/11/20 上午 ...	檔案資料夾	
.github	2020/11/20 上午 ...	檔案資料夾	
.yamato	2020/11/20 上午 ...	檔案資料夾	
com.unity.ml-agents	2020/11/20 上午 ...	檔案資料夾	
com.unity.ml-agents.extensions	2020/11/20 上午 ...	檔案資料夾	
config	2020/11/20 上午 ...	檔案資料夾	
DevProject	2020/11/20 上午 ...	檔案資料夾	
docs	2020/11/20 上午 ...	檔案資料夾	
gym-unity	2020/11/20 上午 ...	檔案資料夾	
ml-agents	2020/11/20 上午 ...	檔案資料夾	
ml-agents-envs	2020/11/20 上午 ...	檔案資料夾	
Project	2020/11/20 上午 ...	檔案資料夾	
protobuf-definitions	2020/11/20 上午 ...	檔案資料夾	
unity-volume	2020/11/20 上午 ...	檔案資料夾	
utils	2020/11/20 上午 ...	檔案資料夾	
.editorconfig	2020/11/20 上午 ...	EDITORCONFIG ...	2 KB
.gitattributes	2020/11/20 上午 ...	GITATTRIBUTES ...	1 KB
.gitignore	2020/11/20 上午 ...	GITIGNORE 檔案	2 KB
.pre-commit-config	2020/11/20 上午 ...	YAML 檔案	5 KB
.pre-commit-search-and-replace	2020/11/20 上午 ...	YAML 檔案	1 KB
.pylintrc	2020/11/20 上午 ...	PYLINTRC 檔案	2 KB
CODE_OF_CONDUCT	2020/11/20 上午 ...	MD 檔案	4 KB
Dockerfile	2020/11/20 上午 ...	檔案	2 KB
LICENSE	2020/11/20 上午 ...	檔案	12 KB

33 個項目

2. Import ML Agent to Unity project



Create a new Unity project

用Unity 2020.1.0f1 開新專案

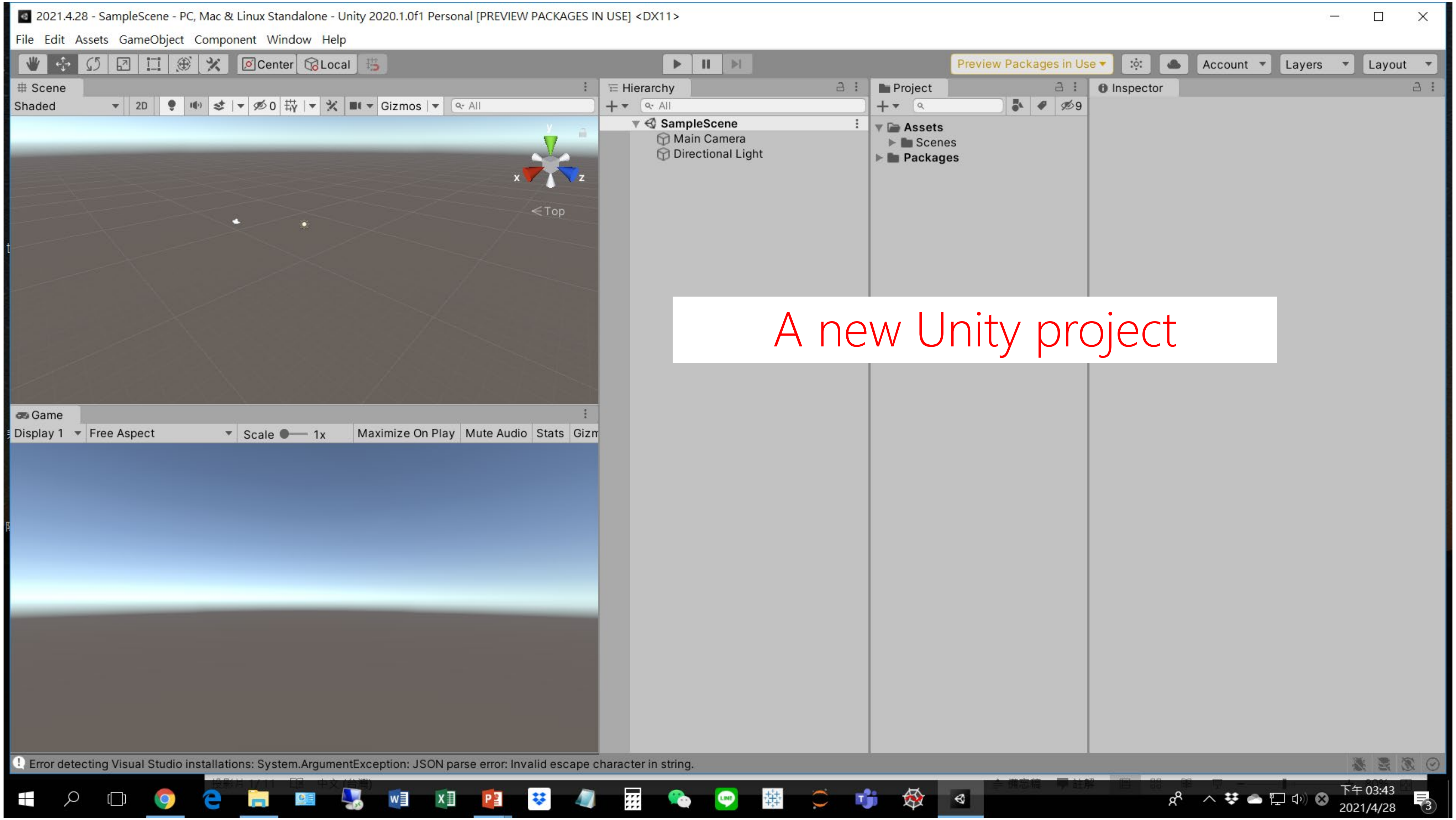
樣板

設定

專案名稱 *: 2021.4.28

讀取位置 *: C:\Users\admin\Desktop

2. Import ML Agent package to Unity project

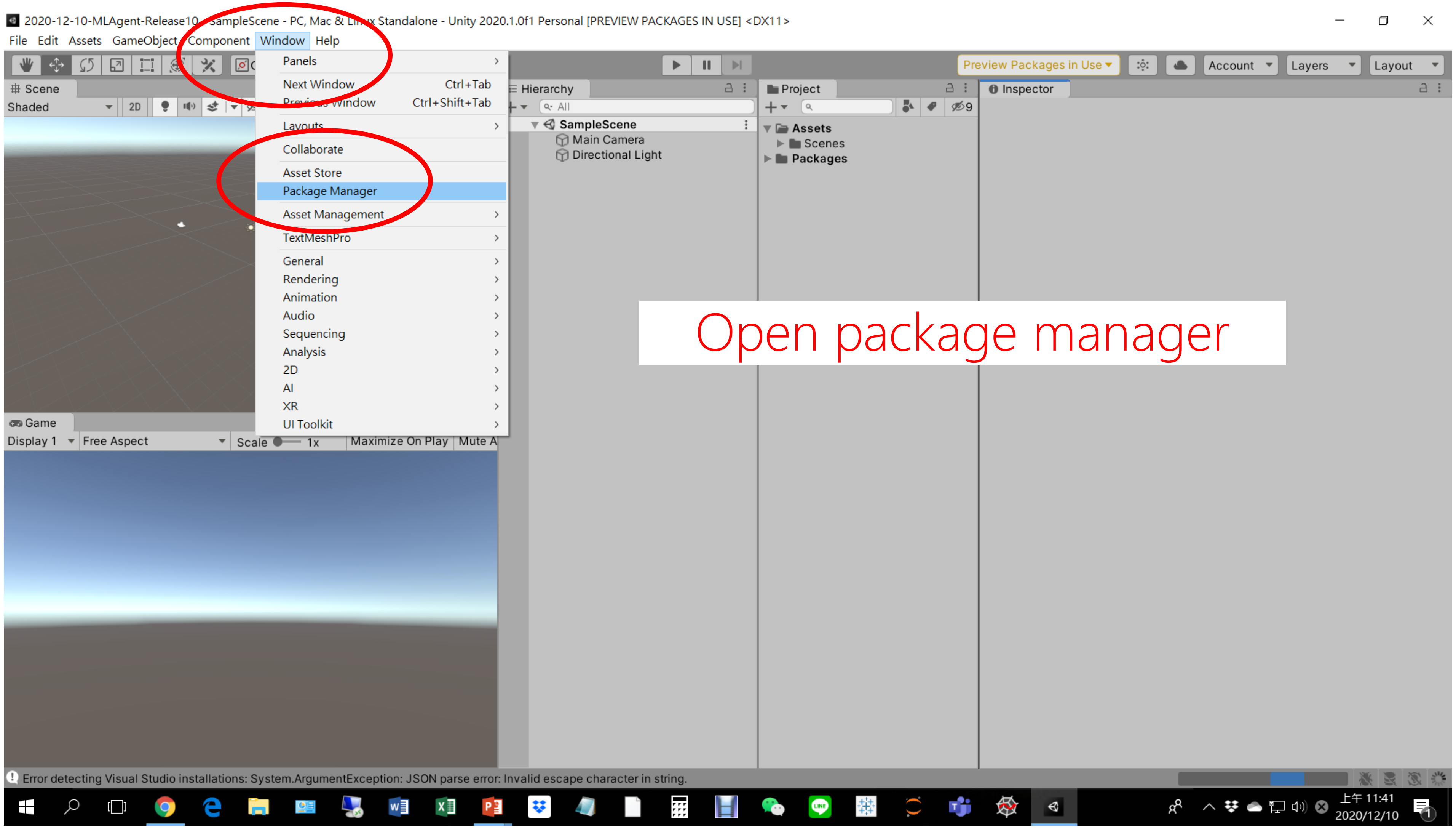


2021.4.28 - SampleScene - PC, Mac & Linux Standalone - Unity 2020.1.0f1 Personal [PREVIEW PACKAGES IN USE] <DX11>

Error detecting Visual Studio installations: System.ArgumentException: JSON parse error: Invalid escape character in string.

A new Unity project

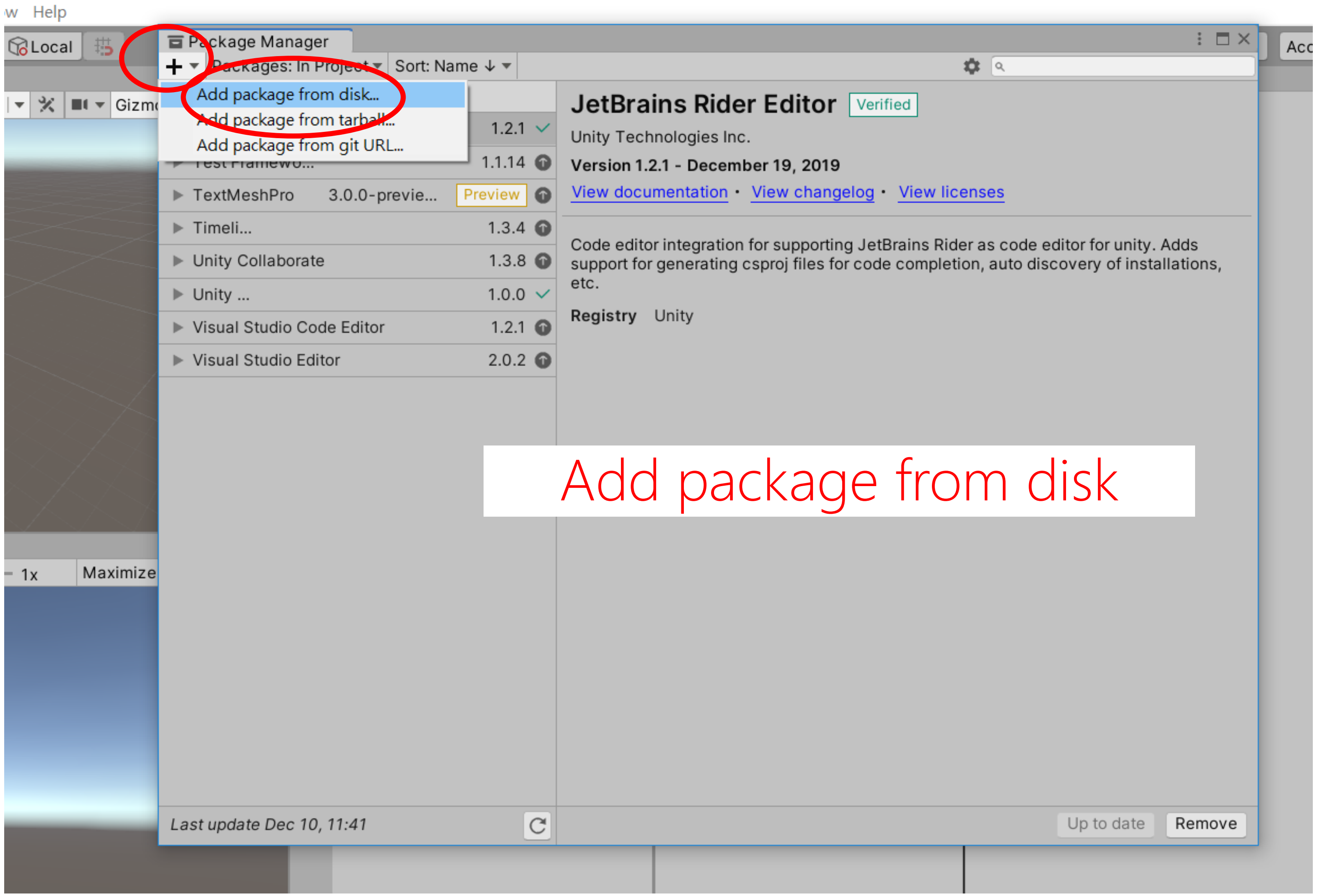
2. Import ML Agent package to Unity project


2020-12-10-MLAgent-Release10 - SampleScene - PC, Mac & Linux Standalone - Unity 2020.1.0f1 Personal [PREVIEW PACKAGES IN USE] <DX11>

Error detecting Visual Studio installations: System.ArgumentException: JSON parse error: Invalid escape character in string.

Open package manager

2. Import ML Agent package to Unity project



JetBrains Rider Editor Verified

Unity Technologies Inc.

Version 1.2.1 - December 19, 2019

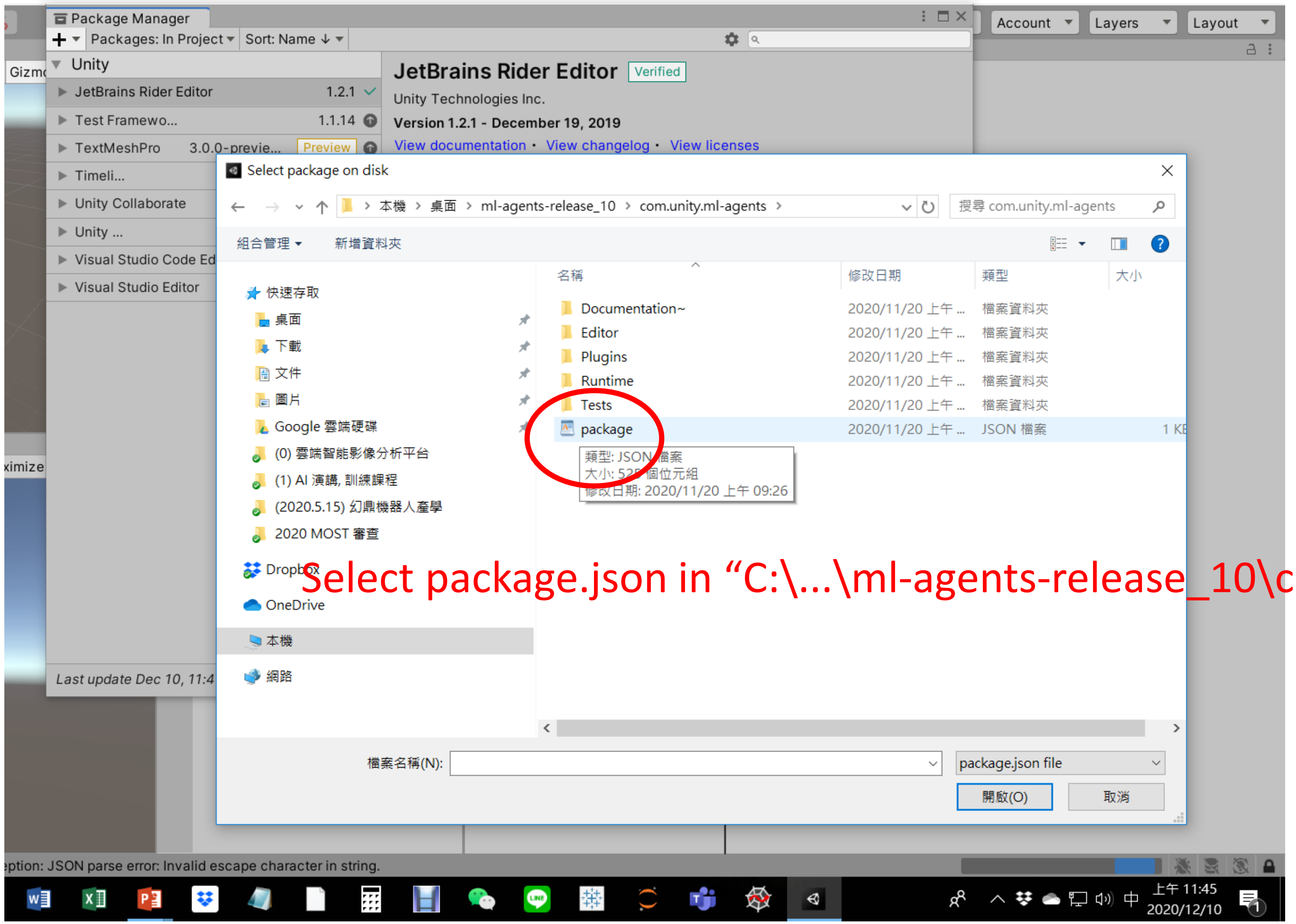
Code editor integration for supporting JetBrains Rider as code editor for unity. Adds support for generating csproj files for code completion, auto discovery of installations, etc.

Registry Unity

Last update Dec 10, 11:41

Add package from disk

2. Import ML Agent package to Unity project



JetBrains Rider Editor Verified

Unity Technologies Inc.

Version 1.2.1 - December 19, 2019

View documentation • View changelog • View licenses

Select package on disk

本機 › 桌面 › ml-agents-release_10 › com.unity.ml-agents ›

搜尋 com.unity.ml-agents

名稱	修改日期	類型	大小
Documentation~	2020/11/20 上午 ...	檔案資料夾	
Editor	2020/11/20 上午 ...	檔案資料夾	
Plugins	2020/11/20 上午 ...	檔案資料夾	
Runtime	2020/11/20 上午 ...	檔案資料夾	
Tests	2020/11/20 上午 ...	檔案資料夾	
package	2020/11/20 上午 ...	JSON 檔案	1 KE

類型: JSON 檔案
大小: 525 個位元組
修改日期: 2020/11/20 上午 09:26

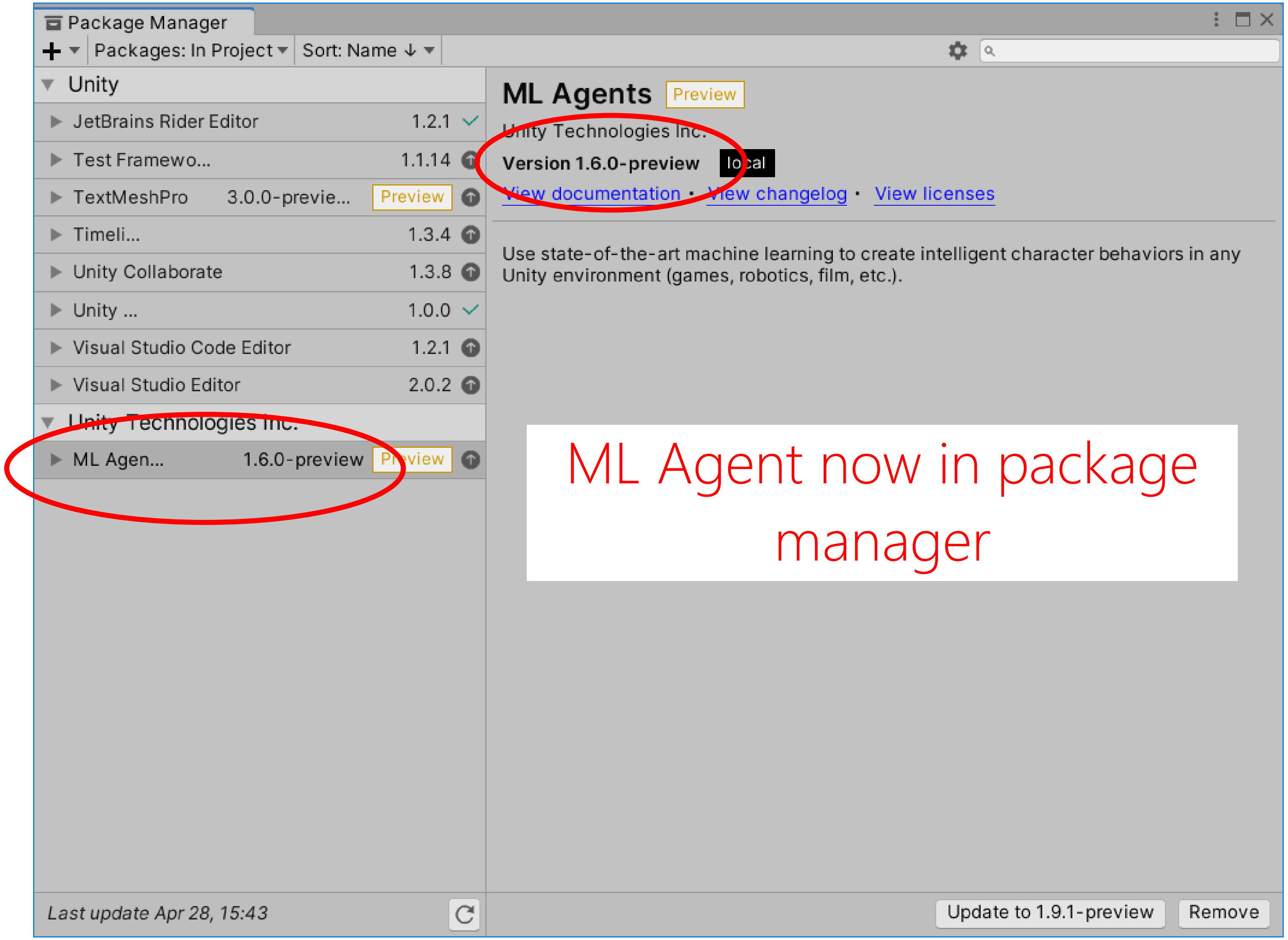
檔案名稱(N):

package.json file

eption: JSON parse error: Invalid escape character in string.

Select package.json in “C:\...\ml-agents-release_10\com.unity.ml-agents”

2. Import ML Agent package to Unity project



Package Manager

Unity

JetBrains Rider Editor	1.2.1	✓
Test Framewo...	1.1.14	
TextMeshPro	3.0.0-previe...	Preview
Timeli...	1.3.4	
Unity Collaborate	1.3.8	
Unity ...	1.0.0	✓
Visual Studio Code Editor	1.2.1	
Visual Studio Editor	2.0.2	

Unity Technologies Inc.

ML Agen...	1.6.0-preview	Preview

Last update Apr 28, 15:43

ML Agents Preview

Unity Technologies Inc.

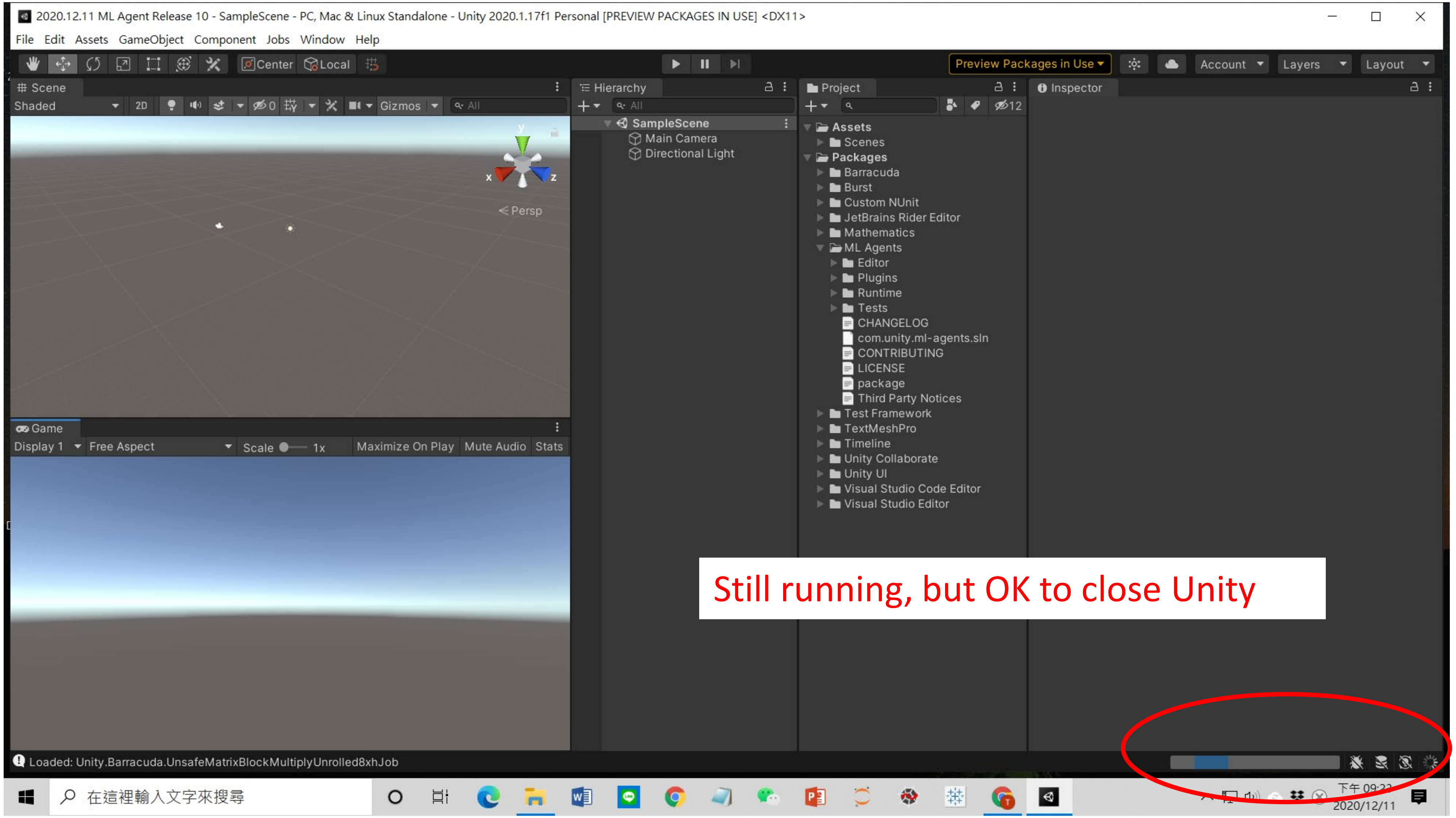
Version 1.6.0-preview local

View documentation • View changelog • View licenses

Use state-of-the-art machine learning to create intelligent character behaviors in any Unity environment (games, robotics, film, etc.).

ML Agent now in package manager

2. Import ML Agent package to Unity project

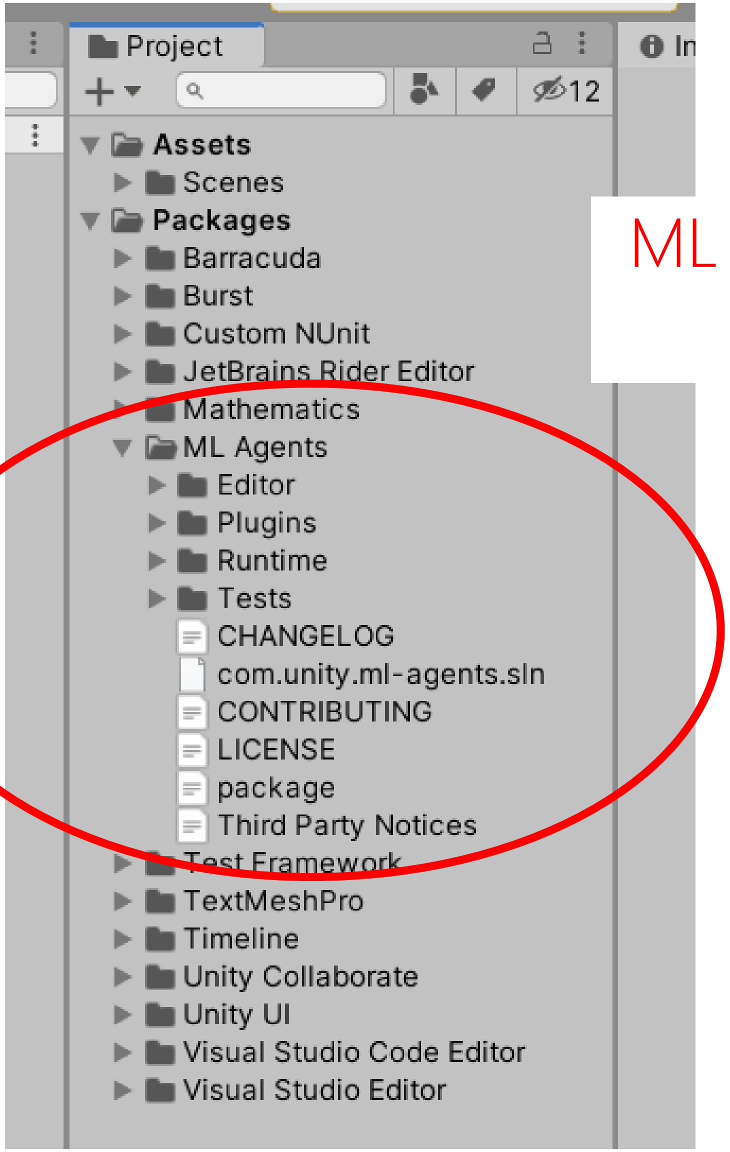


2020.12.11 ML Agent Release 10 - SampleScene - PC, Mac & Linux Standalone - Unity 2020.1.17f1 Personal [PREVIEW PACKAGES IN USE] <DX11>

Loaded: Unity.Barracuda.UnsafeMatrixBlockMultiplyUnrolled8xhJob

Still running, but OK to close Unity

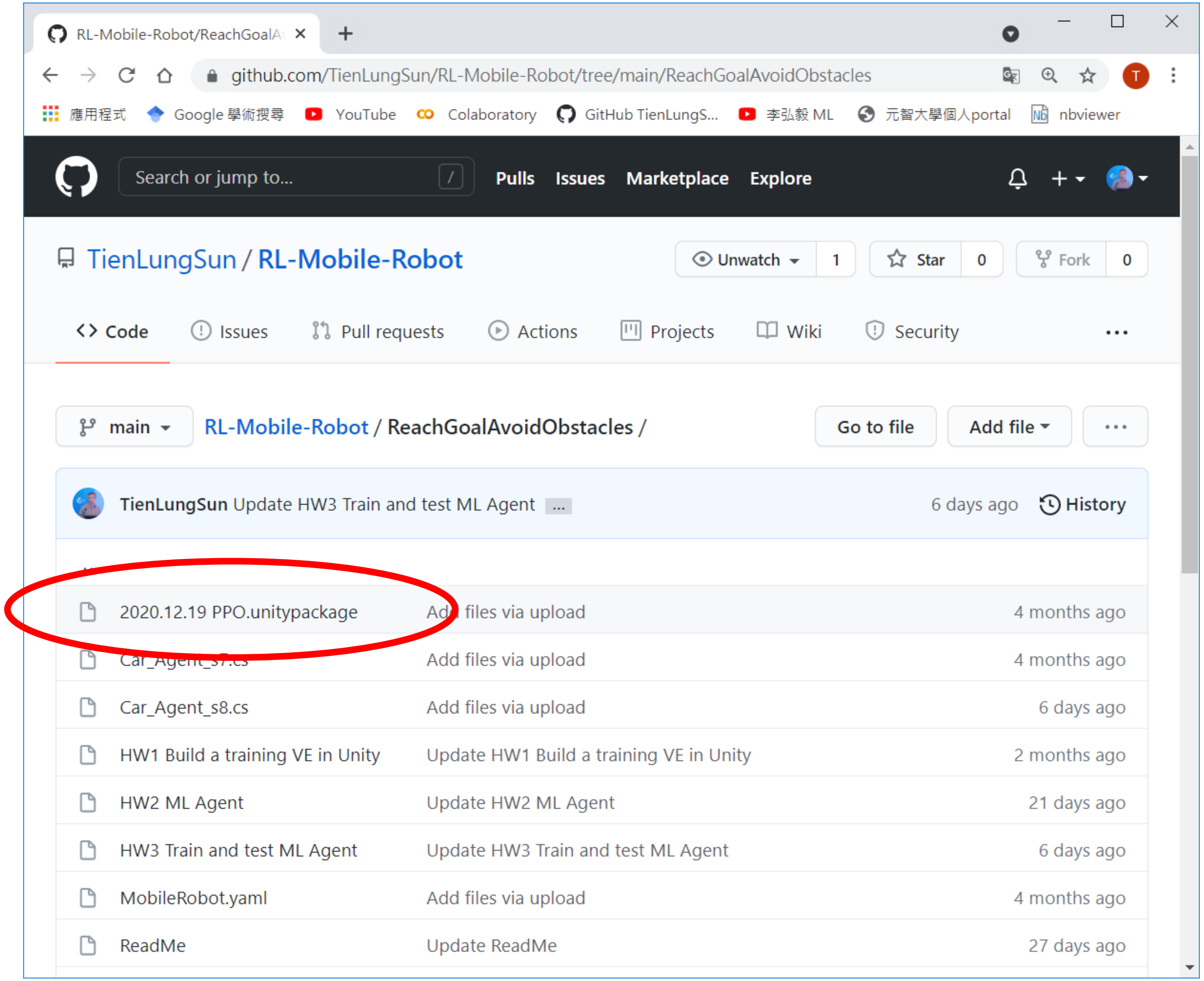
2. Import ML Agent package to Unity project



- Assets
  - Scenes
- Packages
  - Barracuda
  - Burst
  - Custom NUnit
  - JetBrains Rider Editor
  - Mathematics
  - ML Agents
    - Editor
    - Plugins
    - Runtime
    - Tests
    - CHANGELOG
    - com.unity.ml-agents.sln
    - CONTRIBUTING
    - LICENSE
    - package
    - Third Party Notices
  - Test Framework
  - TextMeshPro
  - Timeline
  - Unity Collaborate
  - Unity UI
  - Visual Studio Code Editor
  - Visual Studio Editor

ML Agent is now in Project window

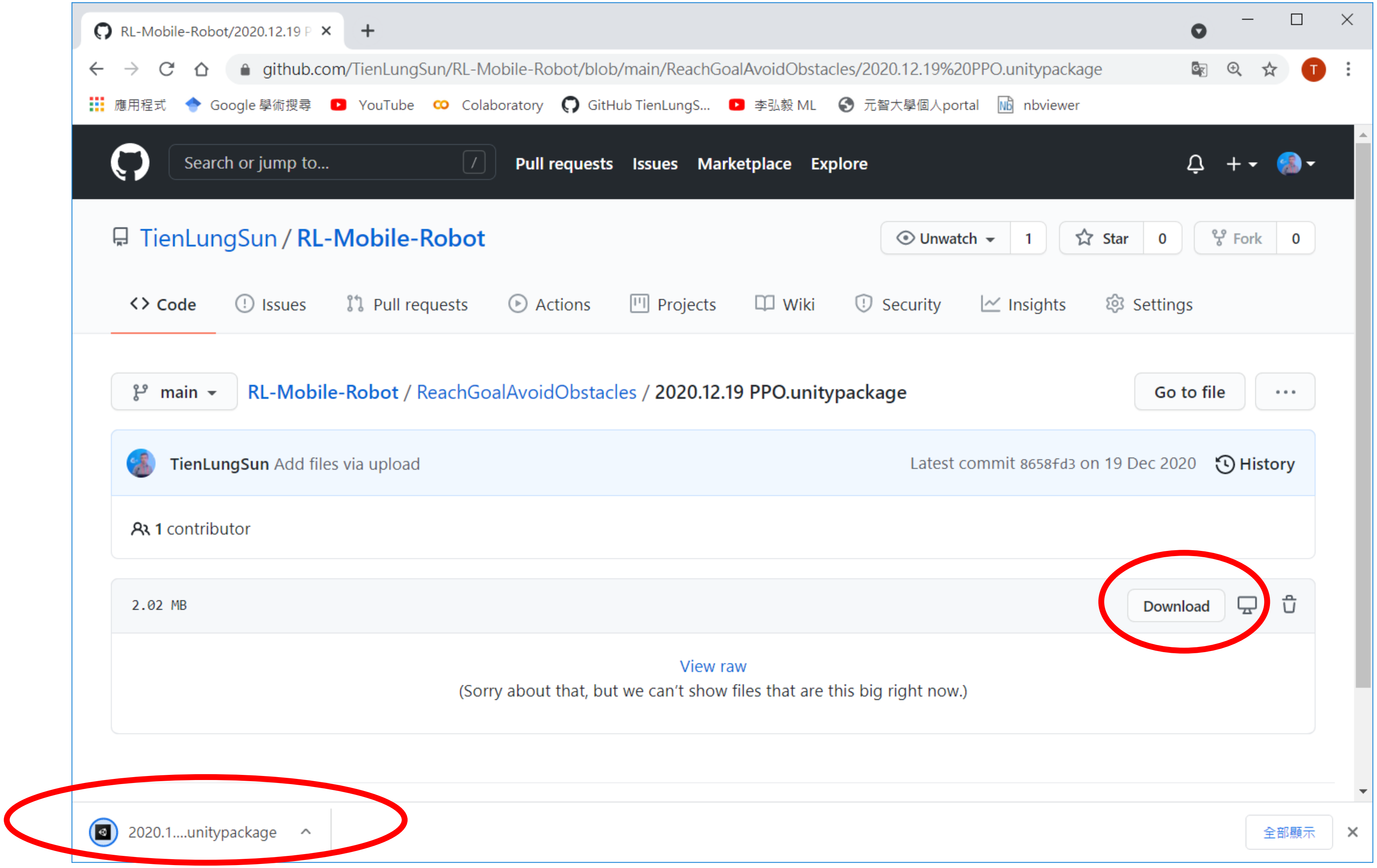
3. Download Unity package from my GitHub



TienLungSun Update HW3 Train and test ML Agent ... 6 days ago

2020.12.19 PPO.unitypackage	Add files via upload	4 months ago
Car_Agent_s7.cs	Add files via upload	4 months ago
Car_Agent_s8.cs	Add files via upload	6 days ago
HW1 Build a training VE in Unity	Update HW1 Build a training VE in Unity	2 months ago
HW2 ML Agent	Update HW2 ML Agent	21 days ago
HW3 Train and test ML Agent	Update HW3 Train and test ML Agent	6 days ago
MobileRobot.yaml	Add files via upload	4 months ago
ReadMe	Update ReadMe	27 days ago

3. Download Unity package from my GitHub



github.com/TienLungSun/RL-Mobile-Robot/blob/main/ReachGoalAvoidObstacles/2020.12.19%20PPO.unitypackage

TienLungSun / RL-Mobile-Robot

RL-Mobile-Robot / ReachGoalAvoidObstacles / 2020.12.19 PPO.unitypackage

TienLungSun Add files via upload

Latest commit 8658fd3 on 19 Dec 2020

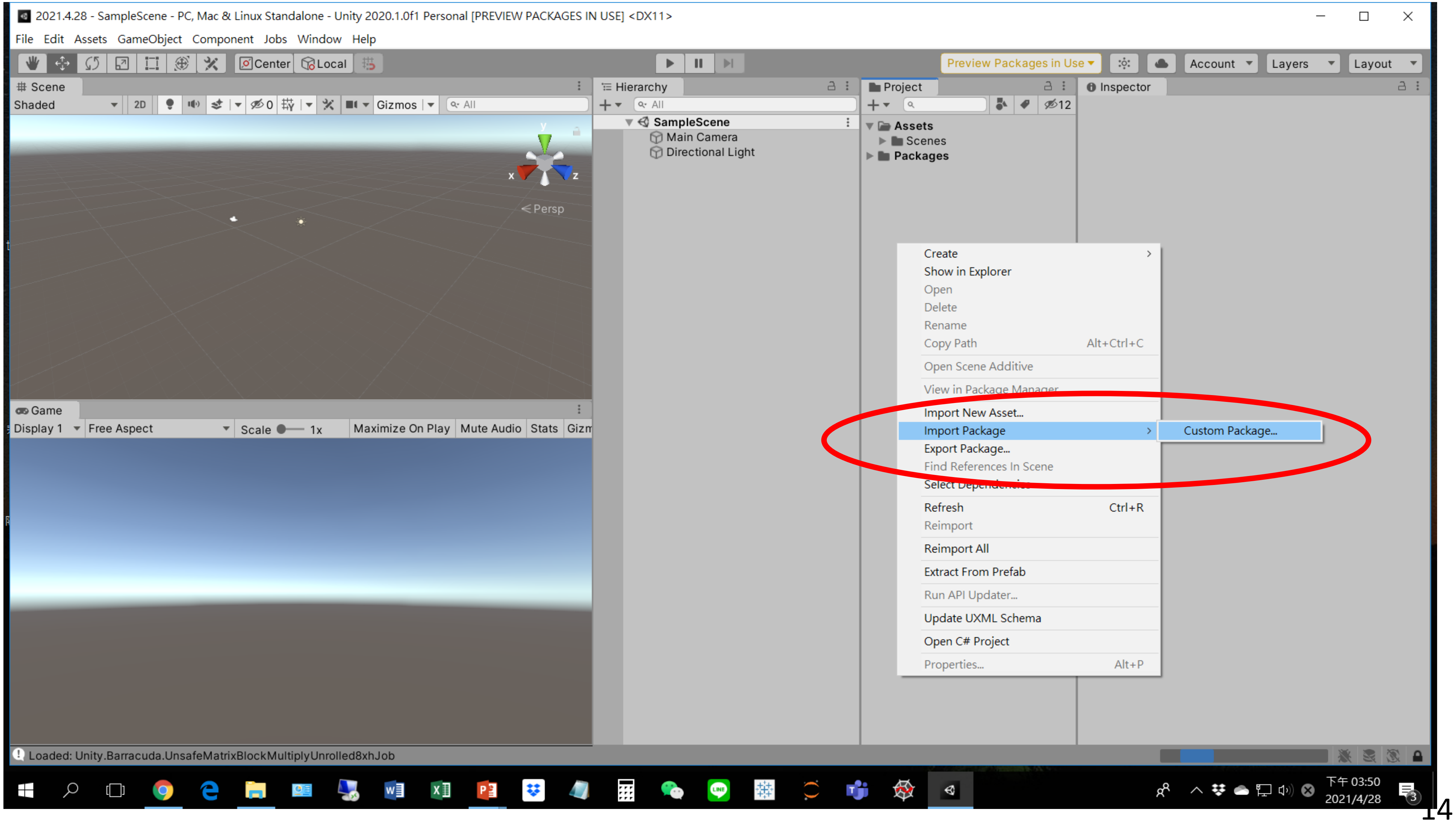
1 contributor

2.02 MB

View raw

(Sorry about that, but we can't show files that are this big right now.)

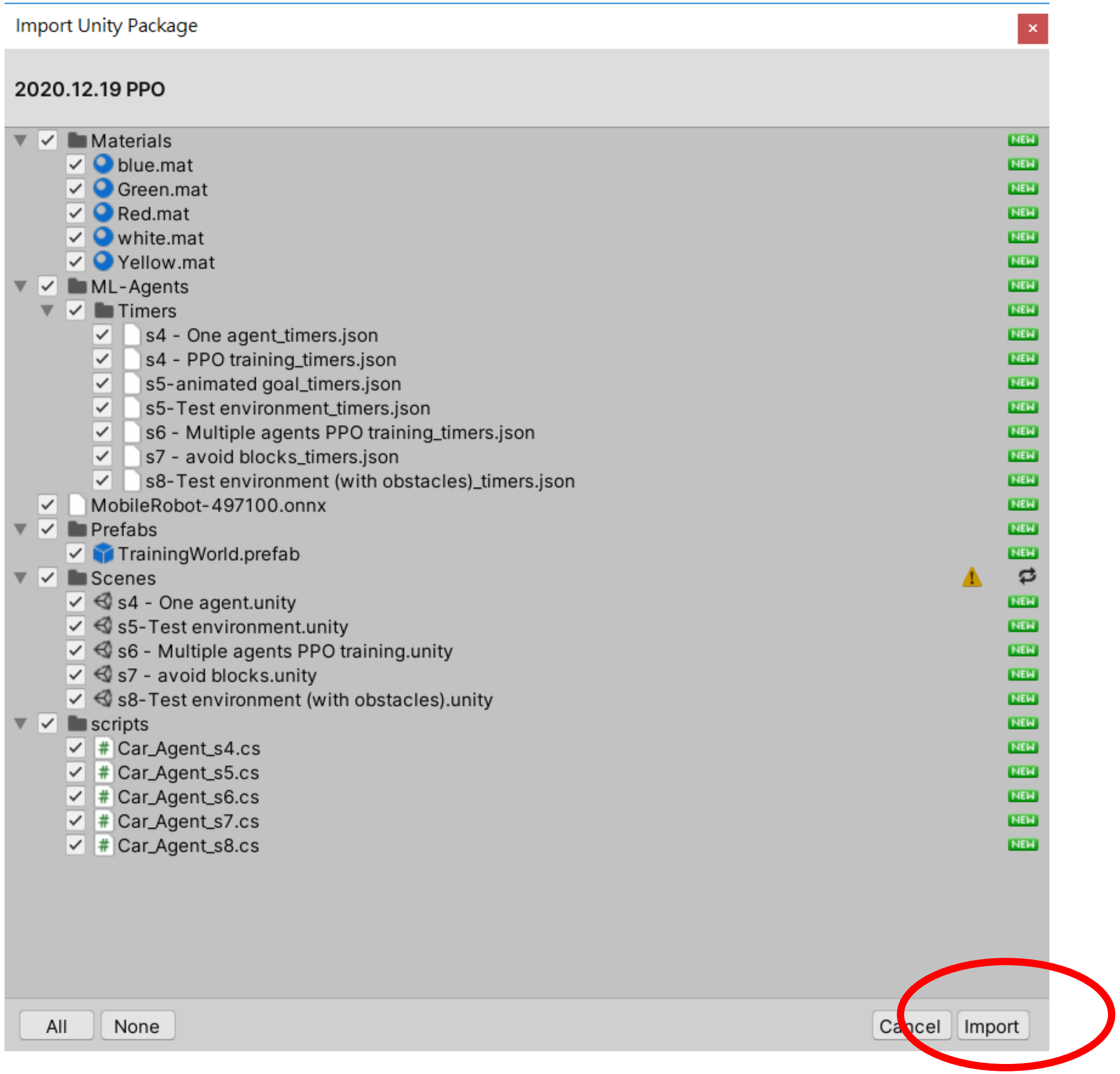
4. Import Unity package



2021.4.28 - SampleScene - PC, Mac & Linux Standalone - Unity 2020.1.0f1 Personal [PREVIEW PACKAGES IN USE] <DX11>

Loaded: Unity.Barracuda.UnsafeMatrixBlockMultiplyUnrolled8xhJob

4. Import Unity package

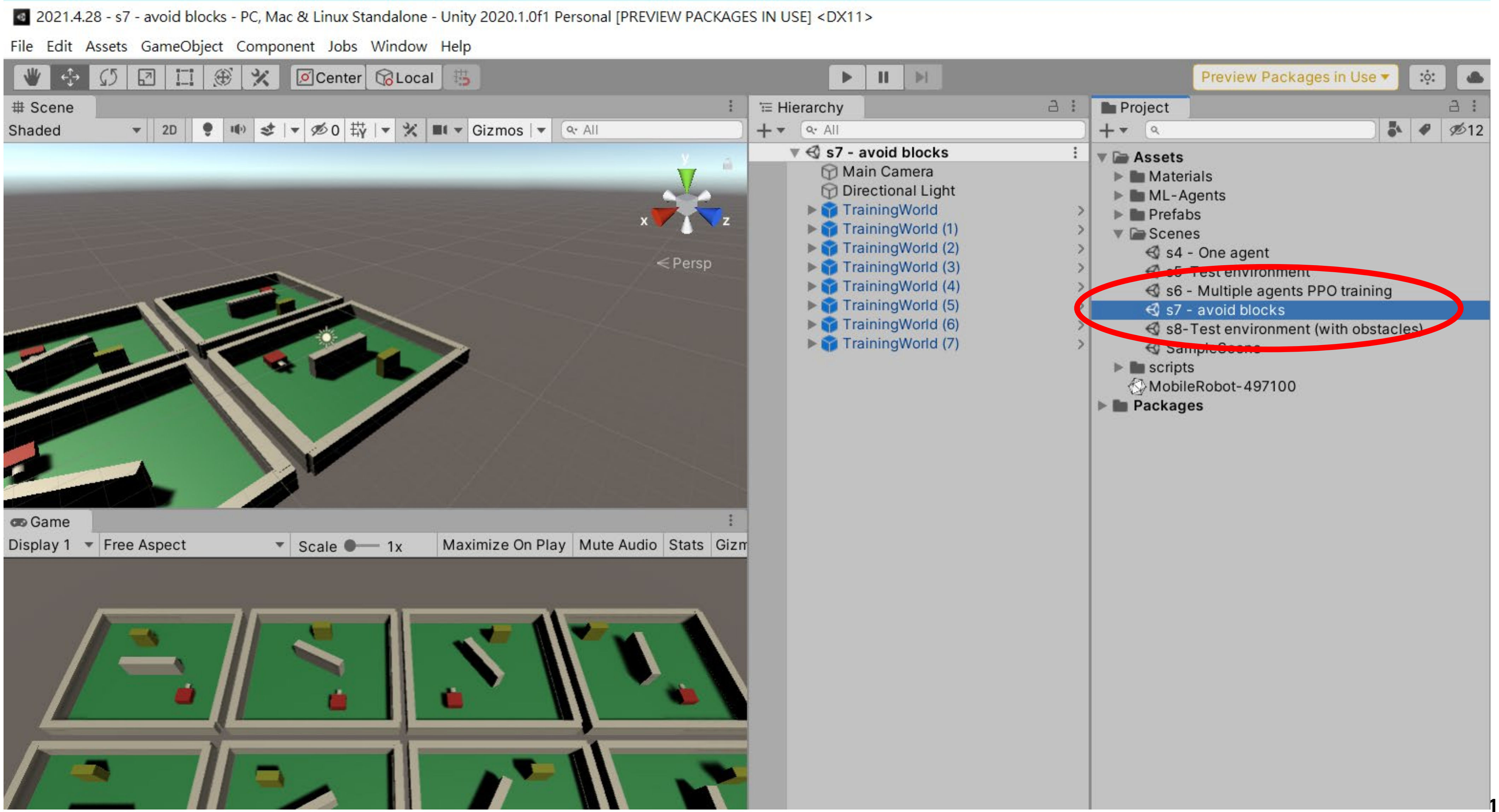


Import Unity Package

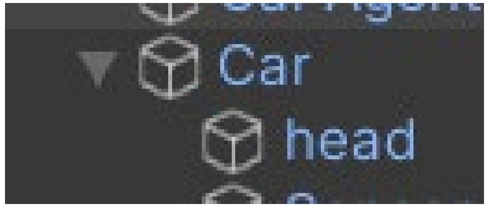
2020.12.19 PPO

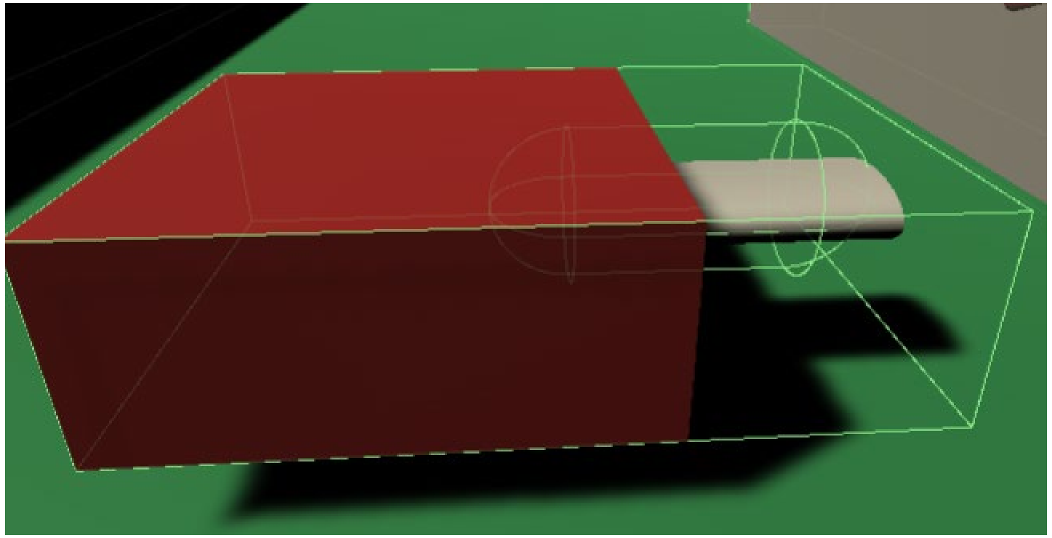
- ☑ Materials NEW
  - ☑ blue.mat NEW
  - ☑ Green.mat NEW
  - ☑ Red.mat NEW
  - ☑ white.mat NEW
  - ☑ Yellow.mat NEW
- ☑ ML-Agents NEW
  - ☑ Timers NEW
    - ☑ s4 - One agent_timers.json NEW
    - ☑ s4 - PPO training_timers.json NEW
    - ☑ s5-animated goal_timers.json NEW
    - ☑ s5-Test environment_timers.json NEW
    - ☑ s6 - Multiple agents PPO training_timers.json NEW
    - ☑ s7 - avoid blocks_timers.json NEW
    - ☑ s8-Test environment (with obstacles)_timers.json NEW
  - ☑ MobileRobot-497100.onnx NEW
- ☑ Prefabs NEW
  - ☑ TrainingWorld.prefab NEW
- ☑ Scenes ⚠
  - ☑ s4 - One agent.unity NEW
  - ☑ s5-Test environment.unity NEW
  - ☑ s6 - Multiple agents PPO training.unity NEW
  - ☑ s7 - avoid blocks.unity NEW
  - ☑ s8-Test environment (with obstacles).unity NEW
- ☑ scripts NEW
  - ☑ Car_Agent_s4.cs NEW
  - ☑ Car_Agent_s5.cs NEW
  - ☑ Car_Agent_s6.cs NEW
  - ☑ Car_Agent_s7.cs NEW
  - ☑ Car_Agent_s8.cs NEW

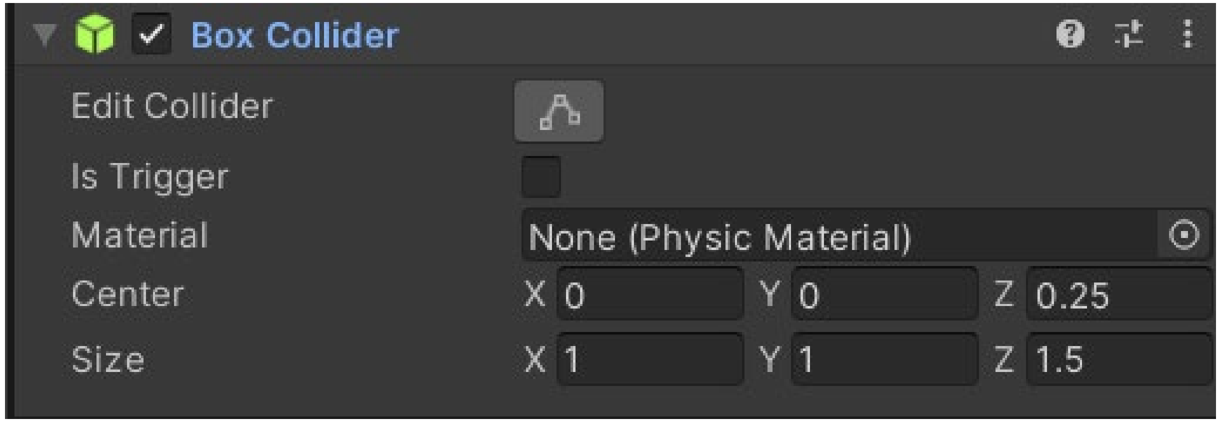
4. Import Unity package



Add a cylinder to represent the front direction







Box Collider

Edit Collider

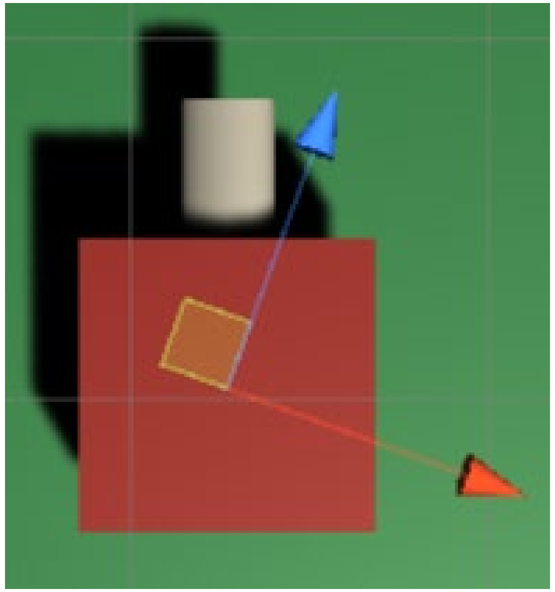
Is Trigger □

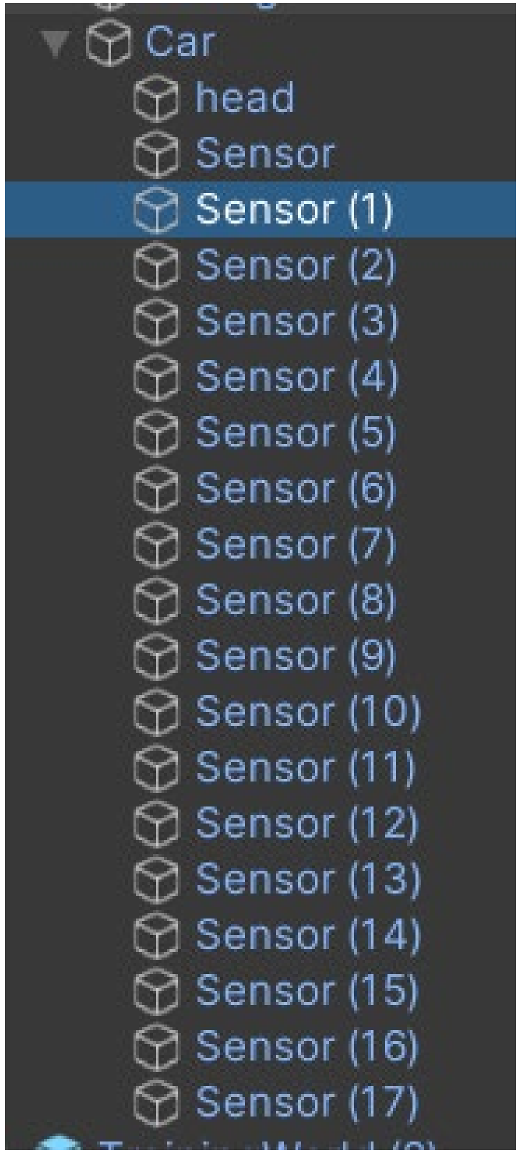
Material None (Physic Material)

Center X 0 Y 0 Z 0.25

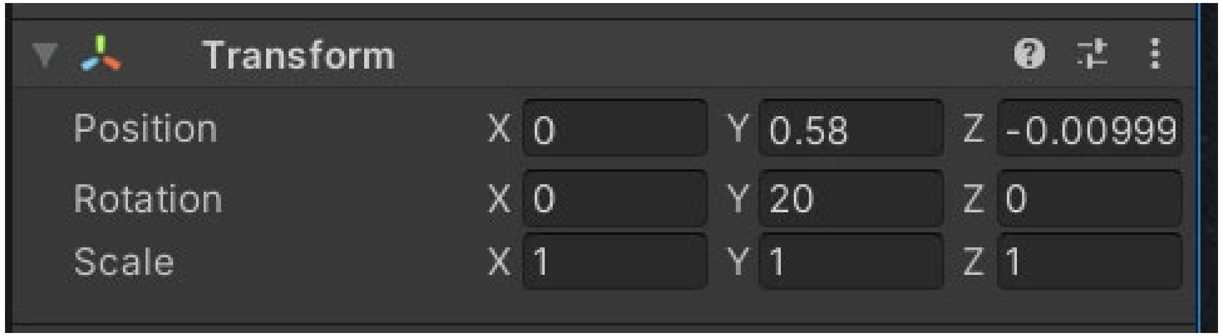
Size X 1 Y 1 Z 1.5

Add distance sensors





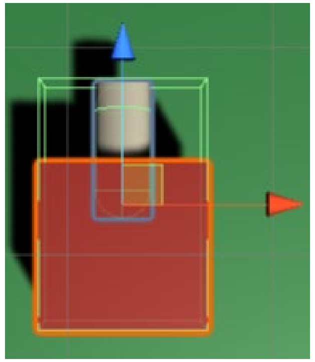
- Car
  - head
  - Sensor
  - Sensor (1)
  - Sensor (2)
  - Sensor (3)
  - Sensor (4)
  - Sensor (5)
  - Sensor (6)
  - Sensor (7)
  - Sensor (8)
  - Sensor (9)
  - Sensor (10)
  - Sensor (11)
  - Sensor (12)
  - Sensor (13)
  - Sensor (14)
  - Sensor (15)
  - Sensor (16)
  - Sensor (17)

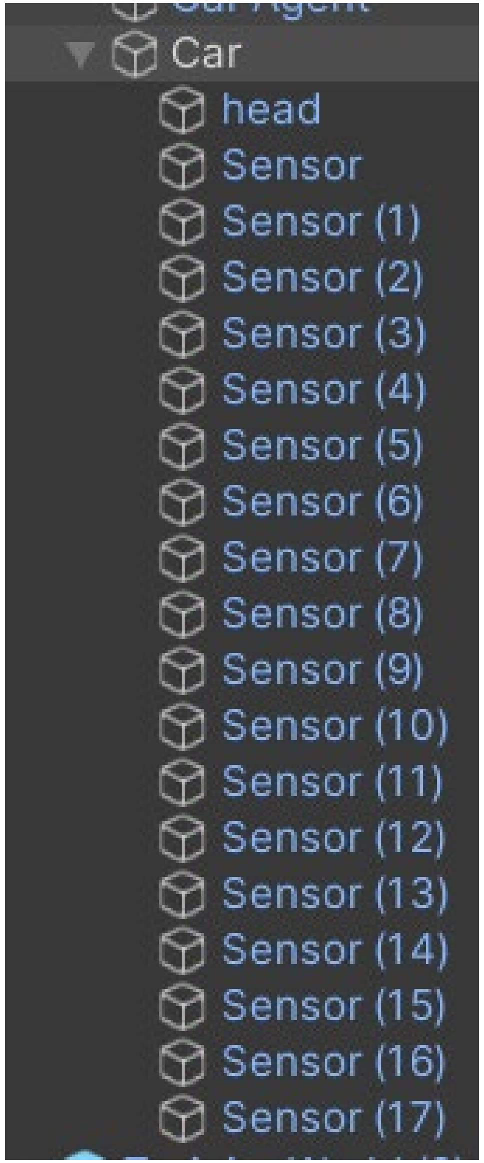


Transform

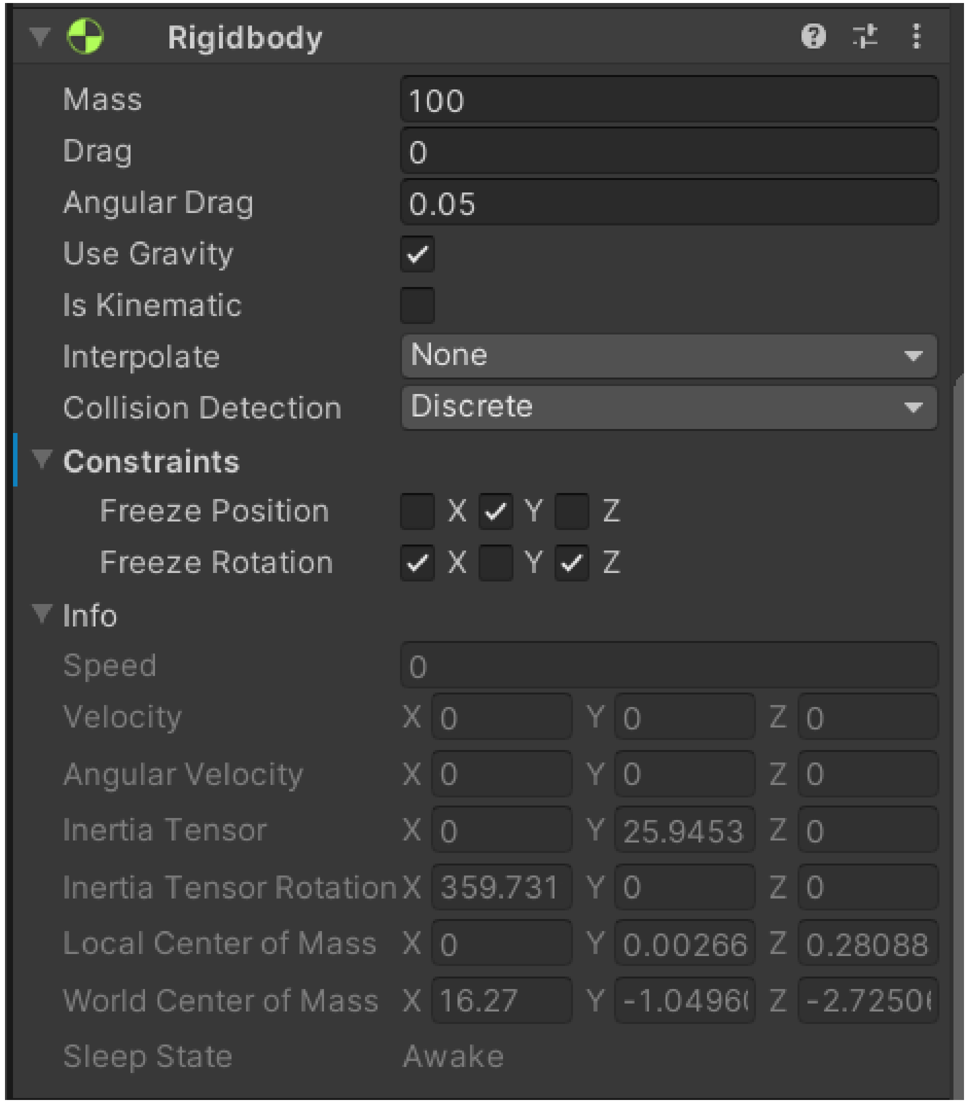
Position	X 0	Y 0.58	Z -0.00999
Rotation	X 0	Y 20	Z 0
Scale	X 1	Y 1	Z 1

Add rigid body component





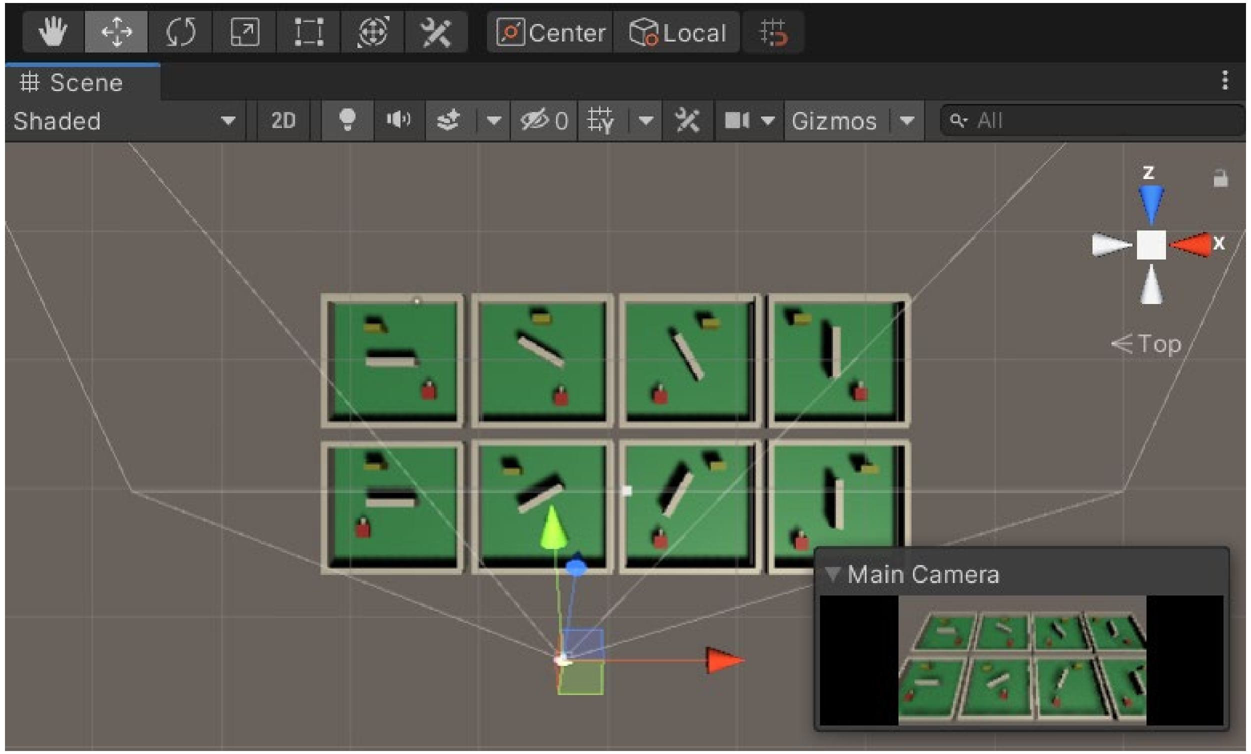
- Car
  - head
  - Sensor
  - Sensor (1)
  - Sensor (2)
  - Sensor (3)
  - Sensor (4)
  - Sensor (5)
  - Sensor (6)
  - Sensor (7)
  - Sensor (8)
  - Sensor (9)
  - Sensor (10)
  - Sensor (11)
  - Sensor (12)
  - Sensor (13)
  - Sensor (14)
  - Sensor (15)
  - Sensor (16)
  - Sensor (17)

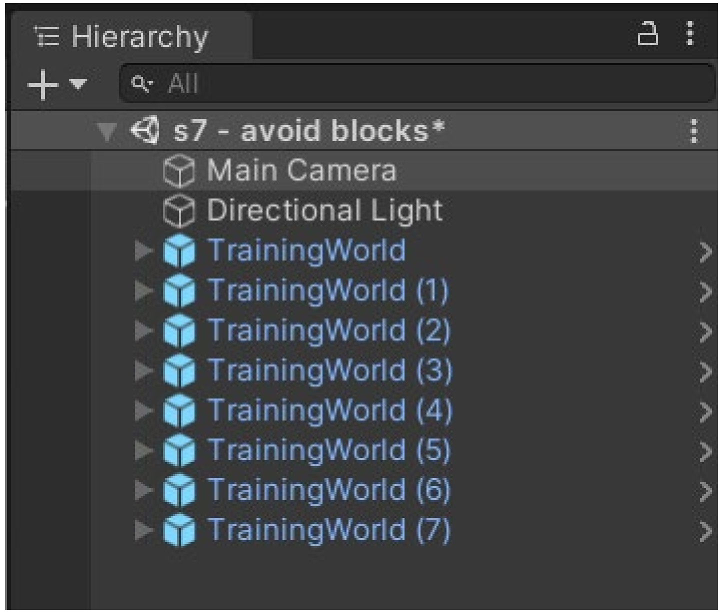


Rigidbody

Mass	100				
Drag	0				
Angular Drag	0.05				
Use Gravity	☑				
Is Kinematic	☐				
Interpolate	None				
Collision Detection	Discrete				
Constraints					
Freeze Position	☐ X	☑ Y	☐ Z		
Freeze Rotation	☑ X	☐ Y	☑ Z		
Info					
Speed	0				
Velocity	X 0	Y 0	Z 0		
Angular Velocity	X 0	Y 0	Z 0		
Inertia Tensor	X 0	Y 25.9453	Z 0		
Inertia Tensor Rotation	X 359.731	Y 0	Z 0		
Local Center of Mass	X 0	Y 0.00266	Z 0.28088		
World Center of Mass	X 16.27	Y -1.0496(	Z -2.7250(		
Sleep State	Awake				

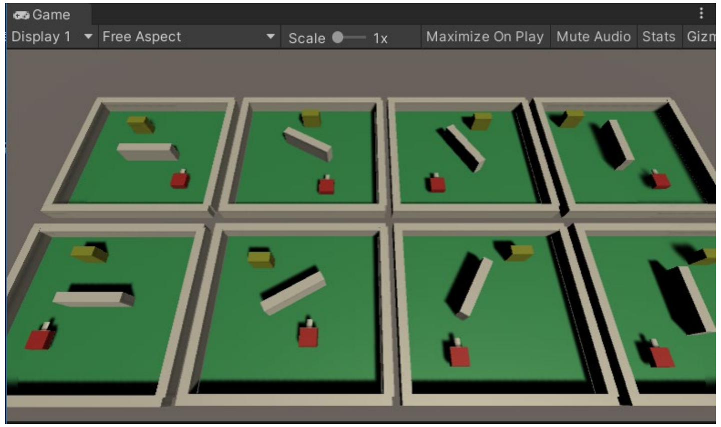
Adjust main camera positon and rotation



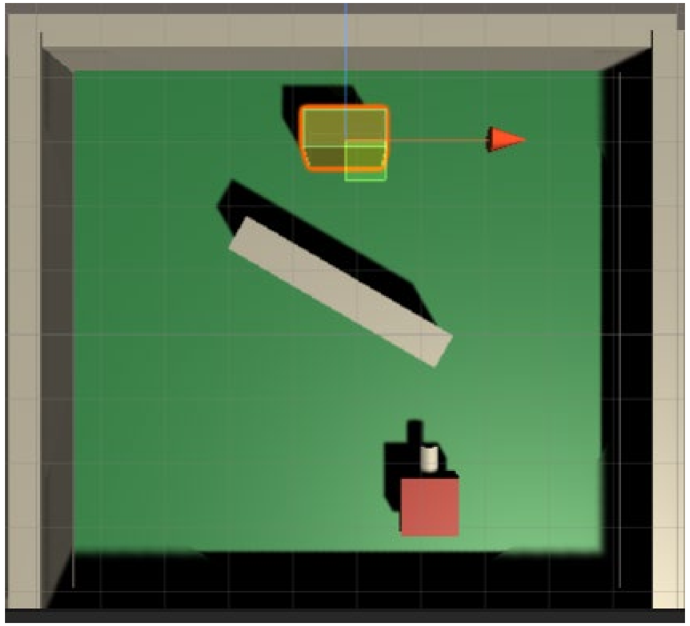

Hierarchy

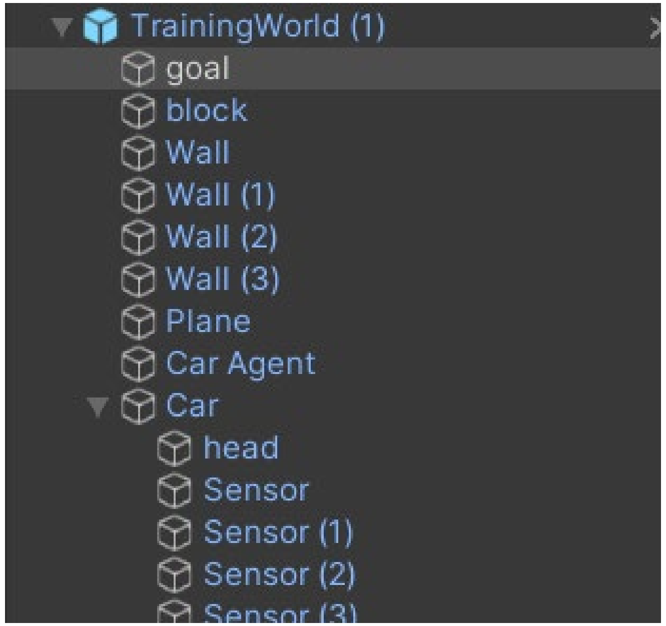
All

s7 - avoid blocks*
- Main Camera
- Directional Light
- TrainingWorld
- TrainingWorld (1)
- TrainingWorld (2)
- TrainingWorld (3)
- TrainingWorld (4)
- TrainingWorld (5)
- TrainingWorld (6)
- TrainingWorld (7)

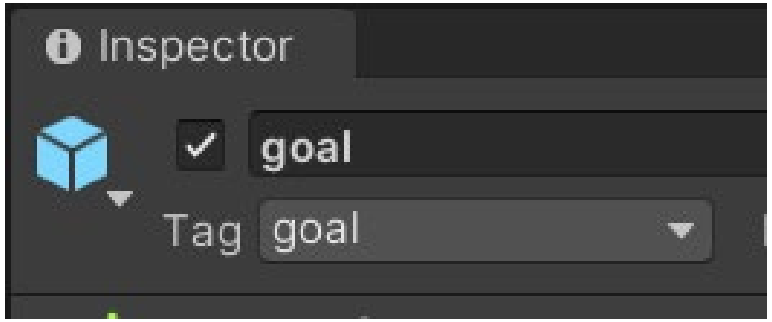


Add goal tag





- TrainingWorld (1)
  - goal
  - block
  - Wall
  - Wall (1)
  - Wall (2)
  - Wall (3)
  - Plane
  - Car Agent
  - Car
    - head
    - Sensor
    - Sensor (1)
    - Sensor (2)
    - Sensor (3)

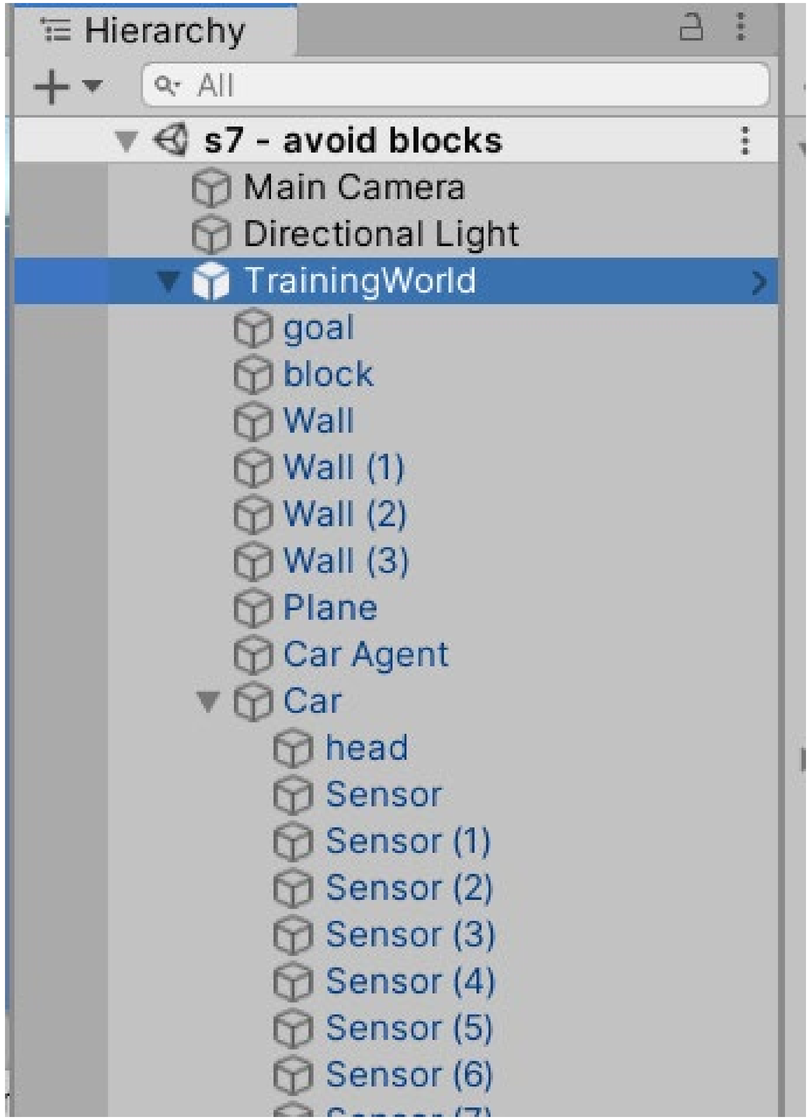


Inspector

☑ goal

Tag goal

Group together to form a training VE

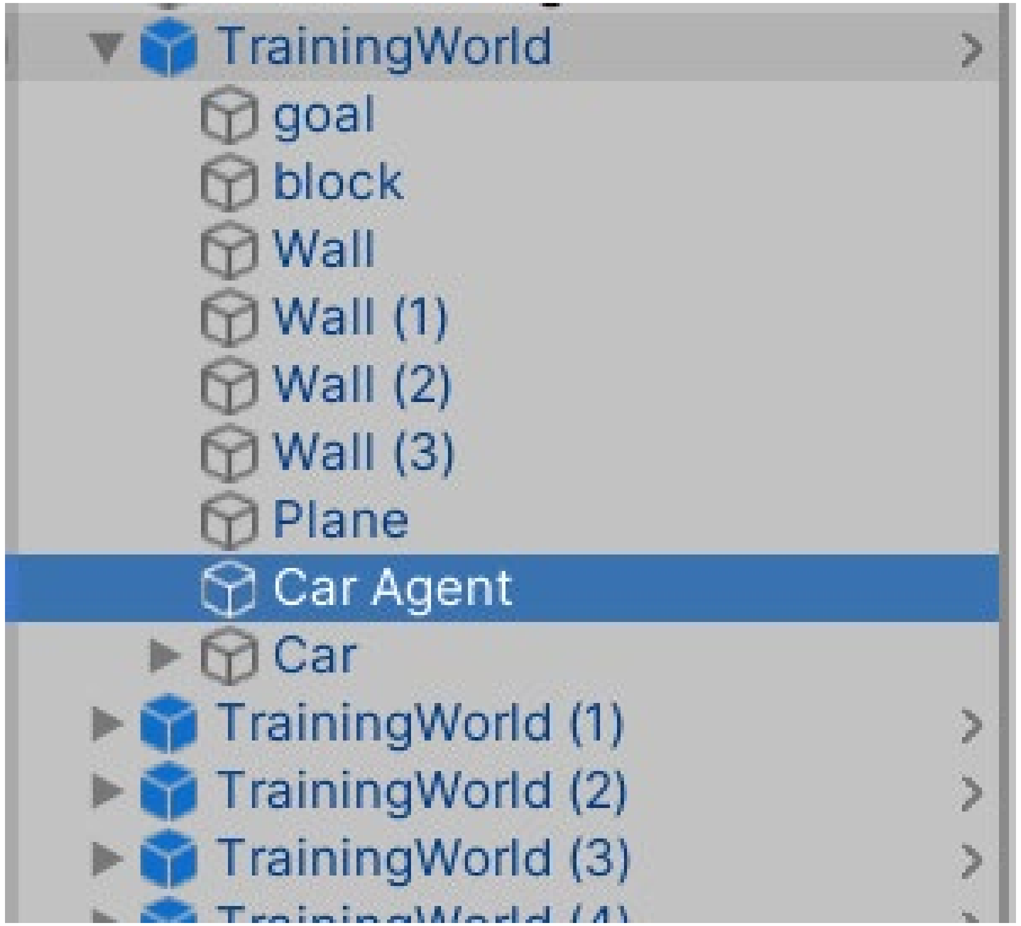


Hierarchy

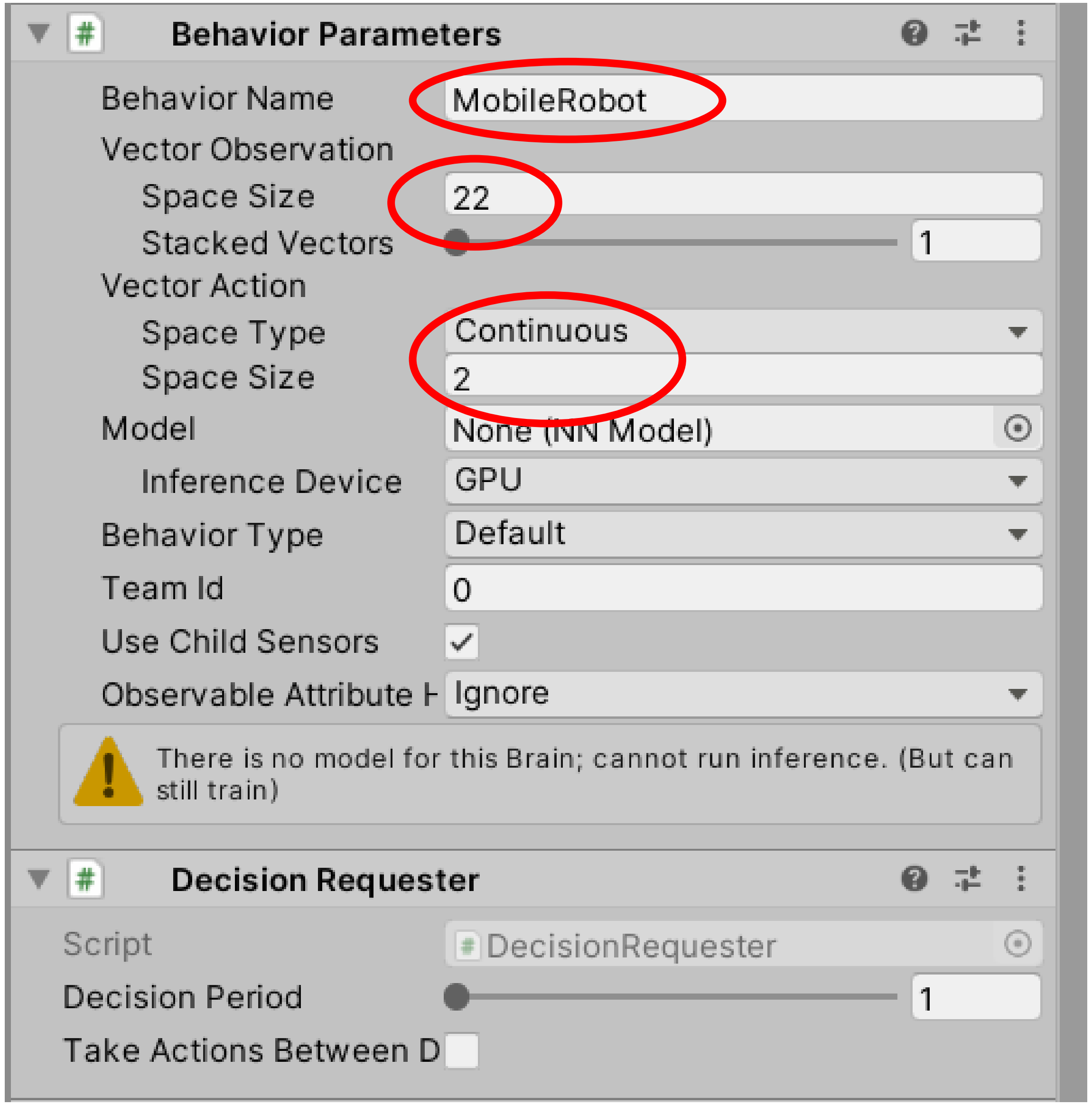
All

- s7 - avoid blocks
  - Main Camera
  - Directional Light
  - TrainingWorld
    - goal
    - block
    - Wall
    - Wall (1)
    - Wall (2)
    - Wall (3)
    - Plane
    - Car Agent
    - Car
      - head
      - Sensor
      - Sensor (1)
      - Sensor (2)
      - Sensor (3)
      - Sensor (4)
      - Sensor (5)
      - Sensor (6)
      - Sensor (7)

Add Car Agent



- TrainingWorld
  - goal
  - block
  - Wall
  - Wall (1)
  - Wall (2)
  - Wall (3)
  - Plane
  - Car Agent
  - Car
- TrainingWorld (1)
- TrainingWorld (2)
- TrainingWorld (3)
- TrainingWorld (4)



Behavior Parameters

Behavior Name: MobileRobot

Vector Observation

Space Size: 22

Stacked Vectors: 1

Vector Action

Space Type: Continuous

Space Size: 2

Model: None (NN Model)

Inference Device: GPU

Behavior Type: Default

Team Id: 0

Use Child Sensors: ☑

Observable Attribute H: Ignore

There is no model for this Brain; cannot run inference. (But can still train)

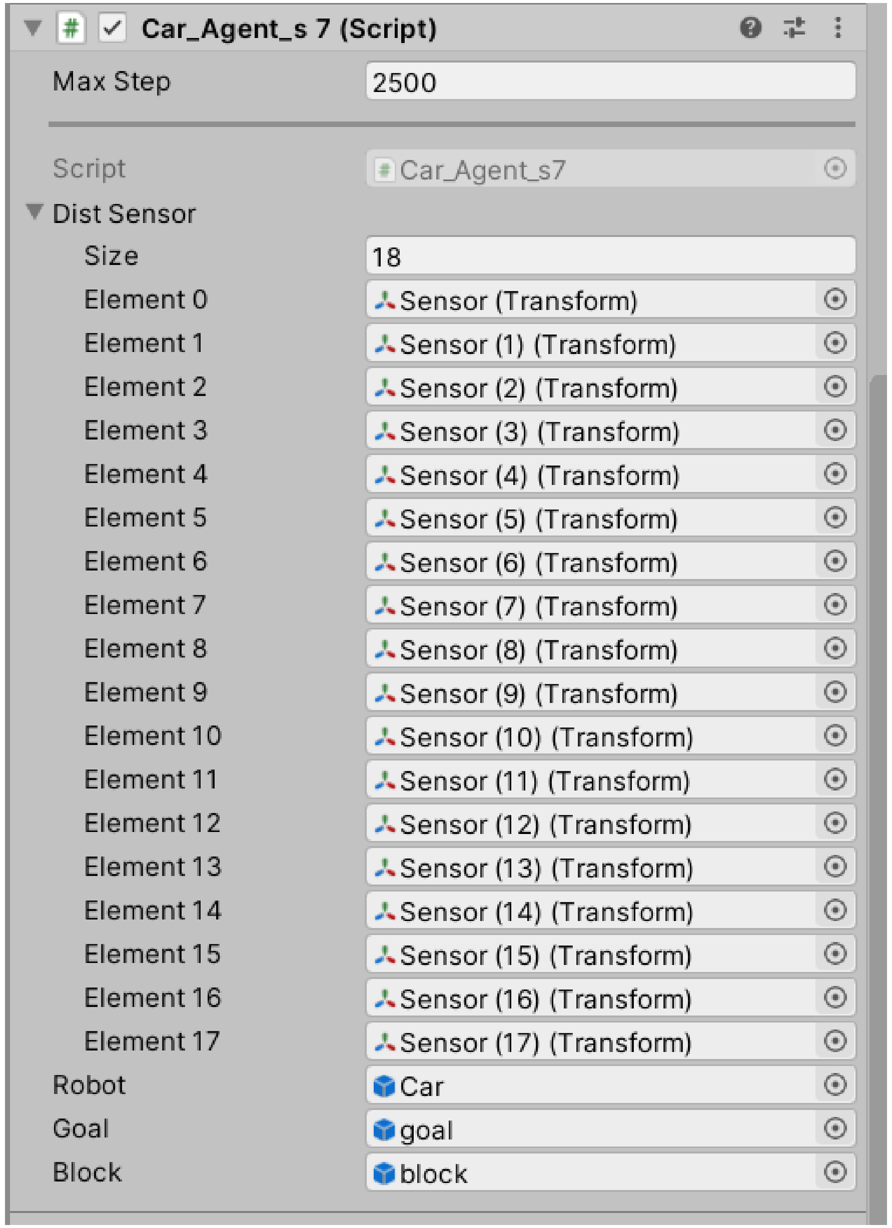
Decision Requester

Script: DecisionRequester

Decision Period: 1

Take Actions Between D: □

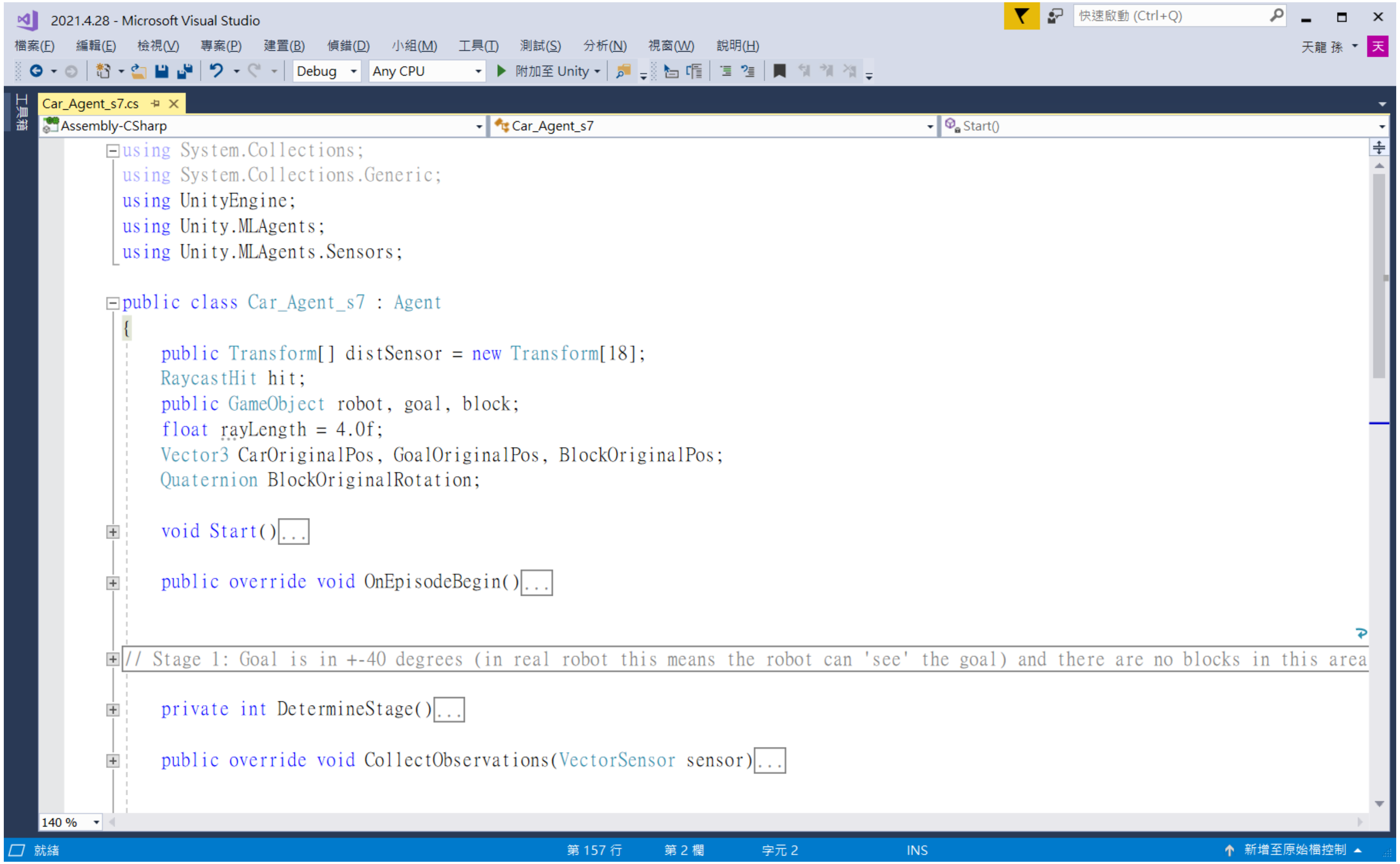
Car Agent script



Car_Agent_s 7 (Script)

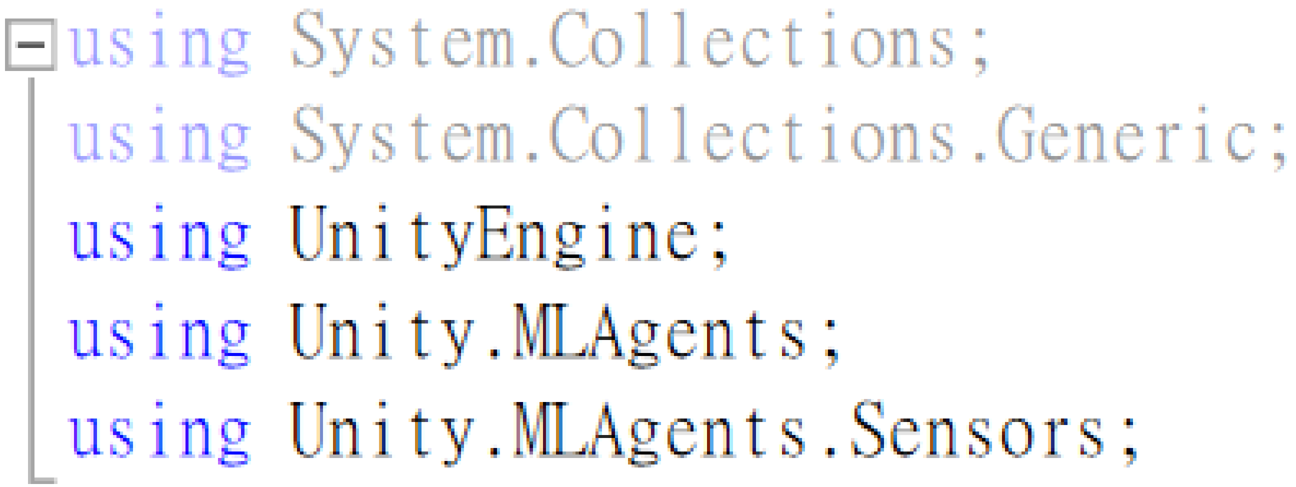
Max Step	2500
Script	Car_Agent_s7
Dist Sensor	
Size	18
Element 0	Sensor (Transform)
Element 1	Sensor (1) (Transform)
Element 2	Sensor (2) (Transform)
Element 3	Sensor (3) (Transform)
Element 4	Sensor (4) (Transform)
Element 5	Sensor (5) (Transform)
Element 6	Sensor (6) (Transform)
Element 7	Sensor (7) (Transform)
Element 8	Sensor (8) (Transform)
Element 9	Sensor (9) (Transform)
Element 10	Sensor (10) (Transform)
Element 11	Sensor (11) (Transform)
Element 12	Sensor (12) (Transform)
Element 13	Sensor (13) (Transform)
Element 14	Sensor (14) (Transform)
Element 15	Sensor (15) (Transform)
Element 16	Sensor (16) (Transform)
Element 17	Sensor (17) (Transform)
Robot	Car
Goal	goal
Block	block

Car Agent script



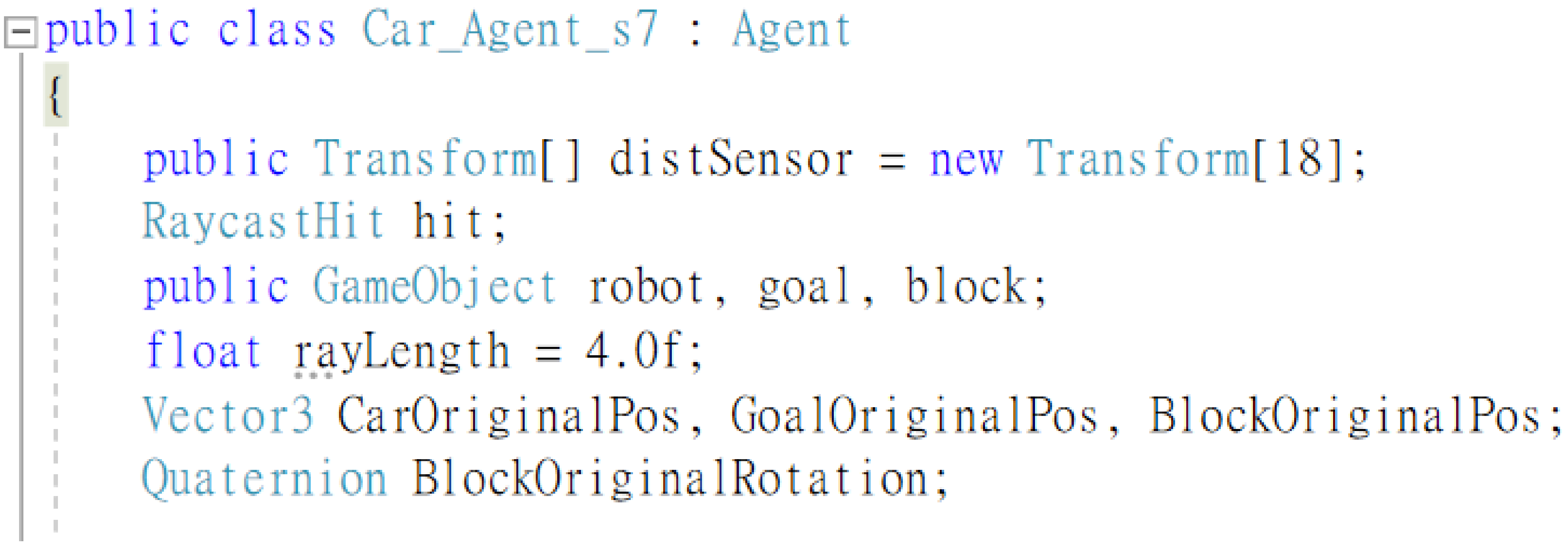
```csharp
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Unity.MLAgents;
using Unity.MLAgents.Sensors;

public class Car_Agent_s7 : Agent
{
    public Transform[] distSensor = new Transform[18];
    RaycastHit hit;
    public GameObject robot, goal, block;
    float rayLength = 4.0f;
    Vector3 CarOriginalPos, GoalOriginalPos, BlockOriginalPos;
    Quaternion BlockOriginalRotation;

    void Start()...

    public override void OnEpisodeBegin()...

// Stage 1: Goal is in +-40 degrees (in real robot this means the robot can 'see' the goal) and there are no blocks in this area

    private int DetermineStage()...

    public override void CollectObservations(VectorSensor sensor)...
```

Using Unity ML Agents



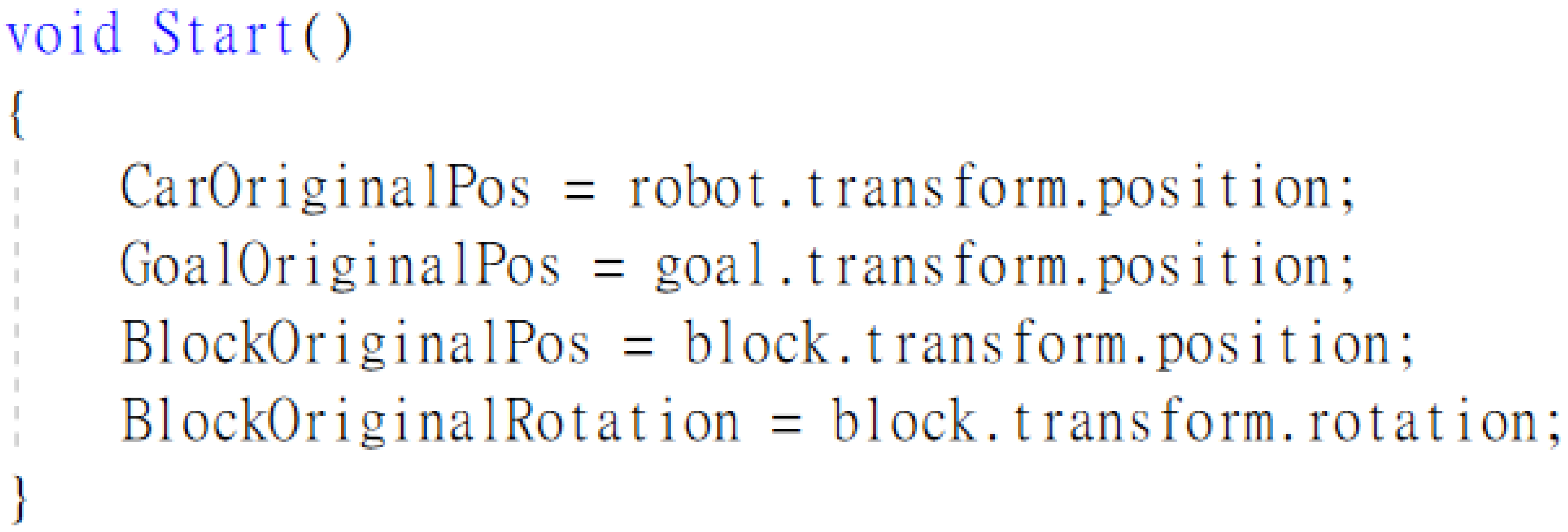
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Unity.MLAgents;
using Unity.MLAgents.Sensors;
```

Class and variables



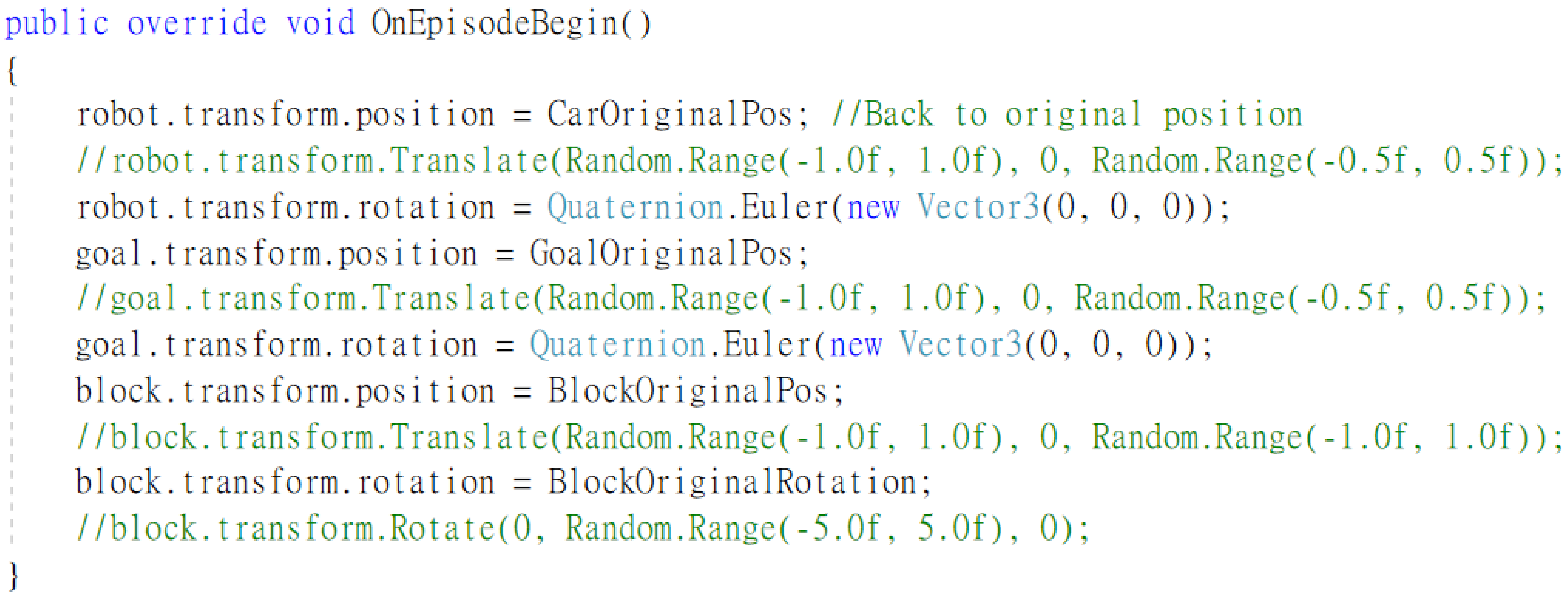
```
public class Car_Agent_s7 : Agent
{
    public Transform[] distSensor = new Transform[18];
    RaycastHit hit;
    public GameObject robot, goal, block;
    float rayLength = 4.0f;
    Vector3 CarOriginalPos, GoalOriginalPos, BlockOriginalPos;
    Quaternion BlockOriginalRotation;
```

Start



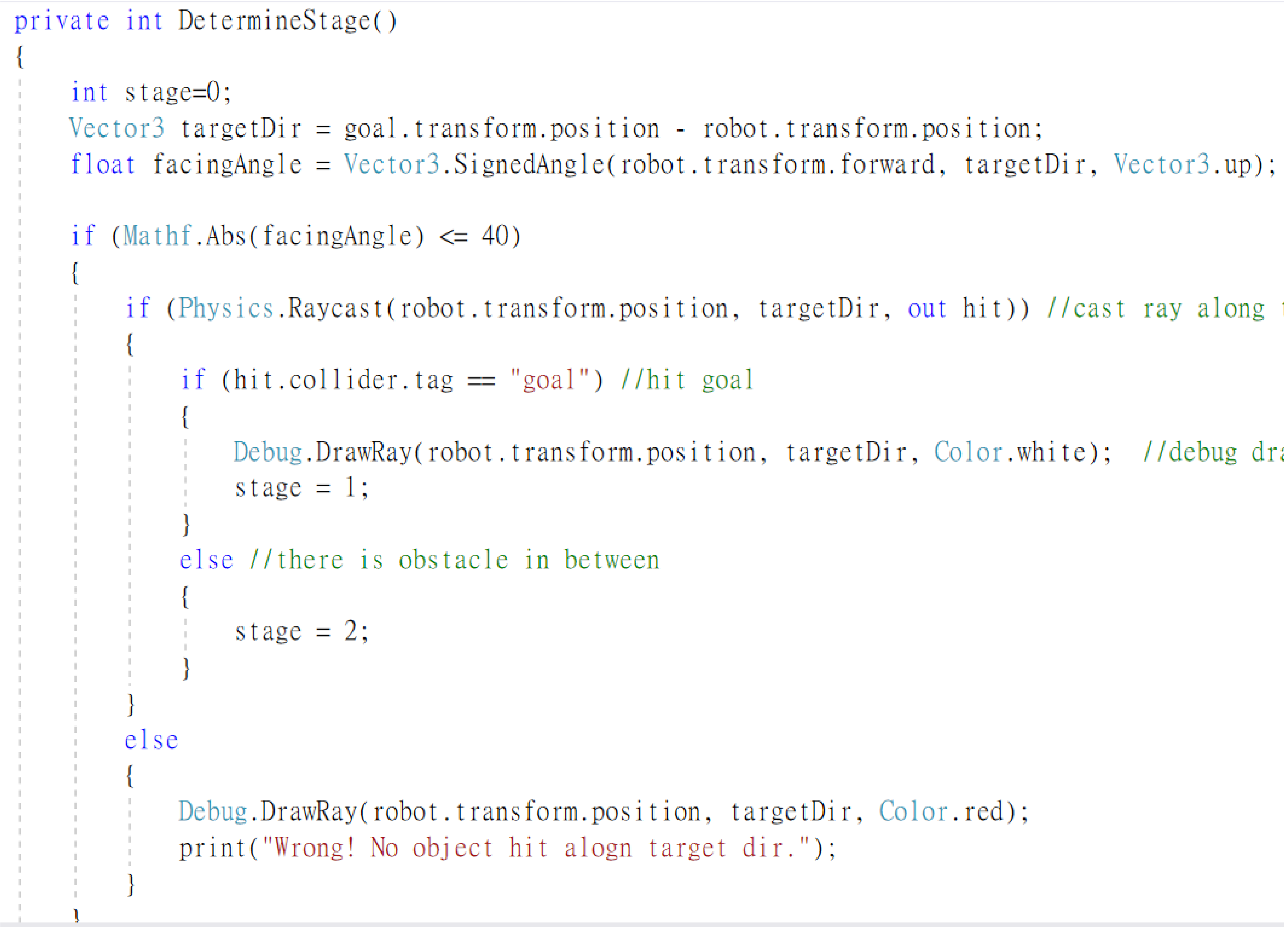
```
void Start()
{
    CarOriginalPos = robot.transform.position;
    GoalOriginalPos = goal.transform.position;
    BlockOriginalPos = block.transform.position;
    BlockOriginalRotation = block.transform.rotation;
}
```

On Episode Begin



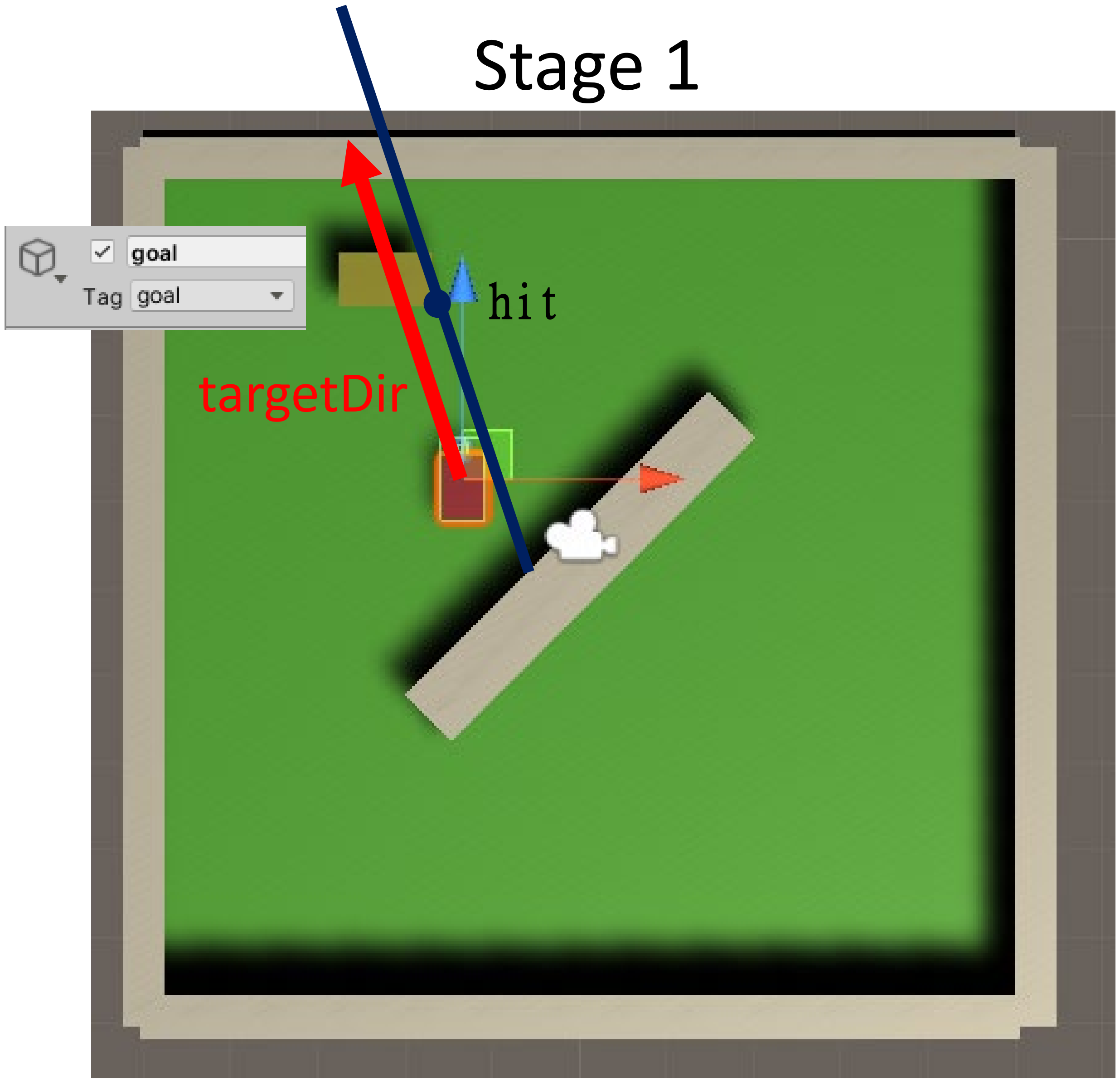
```
public override void OnEpisodeBegin()
{
    robot.transform.position = CarOriginalPos; //Back to original position
    //robot.transform.Translate(Random.Range(-1.0f, 1.0f), 0, Random.Range(-0.5f, 0.5f));
    robot.transform.rotation = Quaternion.Euler(new Vector3(0, 0, 0));
    goal.transform.position = GoalOriginalPos;
    //goal.transform.Translate(Random.Range(-1.0f, 1.0f), 0, Random.Range(-0.5f, 0.5f));
    goal.transform.rotation = Quaternion.Euler(new Vector3(0, 0, 0));
    block.transform.position = BlockOriginalPos;
    //block.transform.Translate(Random.Range(-1.0f, 1.0f), 0, Random.Range(-1.0f, 1.0f));
    block.transform.rotation = BlockOriginalRotation;
    //block.transform.Rotate(0, Random.Range(-5.0f, 5.0f), 0);
}
```

Determine stages

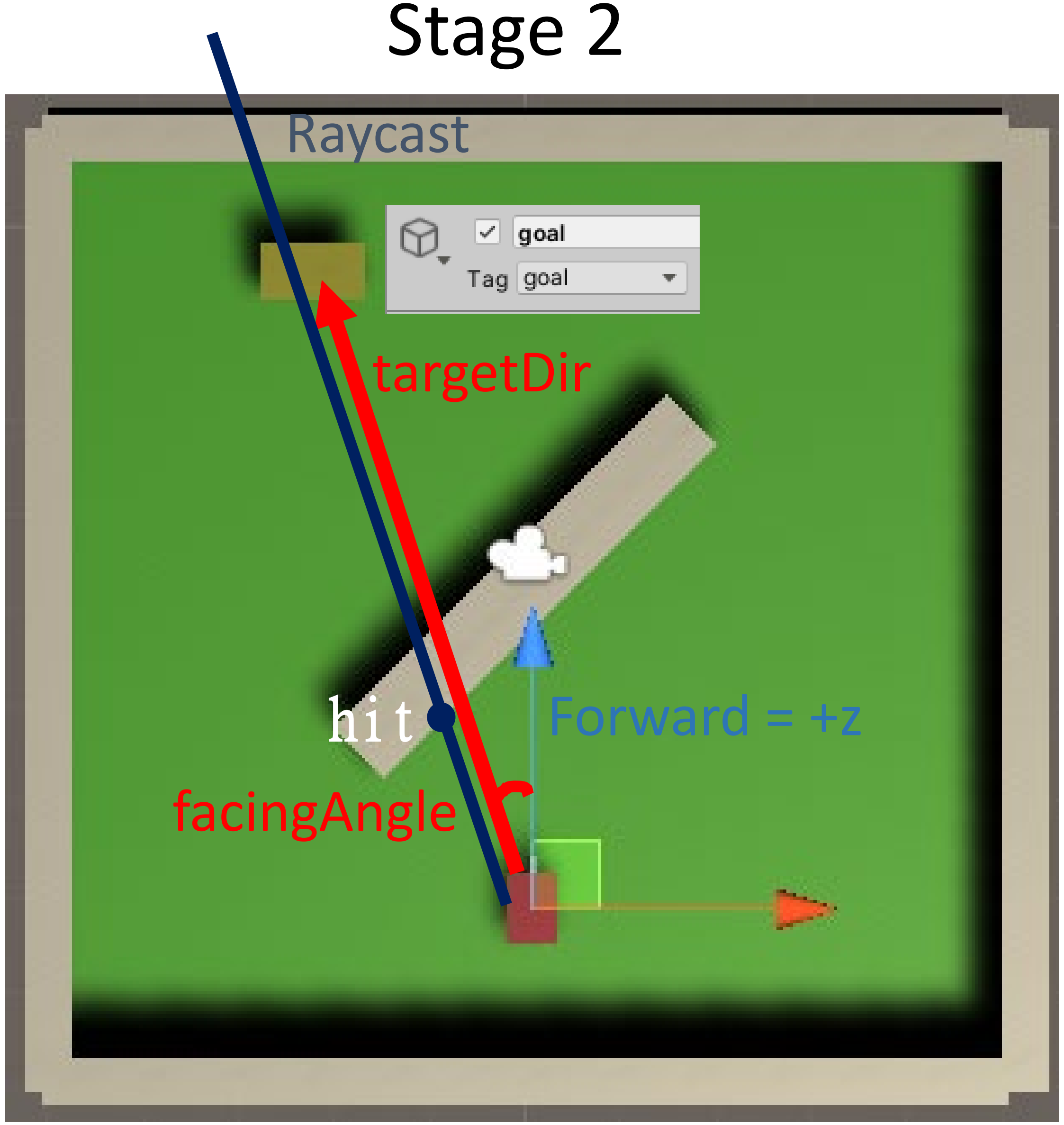


```
private int DetermineStage()
{
    int stage=0;
    Vector3 targetDir = goal.transform.position - robot.transform.position;
    float facingAngle = Vector3.SignedAngle(robot.transform.forward, targetDir, Vector3.up);

    if (Mathf.Abs(facingAngle) <= 40)
    {
        if (Physics.Raycast(robot.transform.position, targetDir, out hit)) //cast ray along
        {
            if (hit.collider.tag == "goal") //hit goal
            {
                Debug.DrawRay(robot.transform.position, targetDir, Color.white);  //debug dra
                stage = 1;
            }
            else //there is obstacle in between
            {
                stage = 2;
            }
        }
        else
        {
            Debug.DrawRay(robot.transform.position, targetDir, Color.red);
            print("Wrong! No object hit alogn target dir.");
        }
    }
```

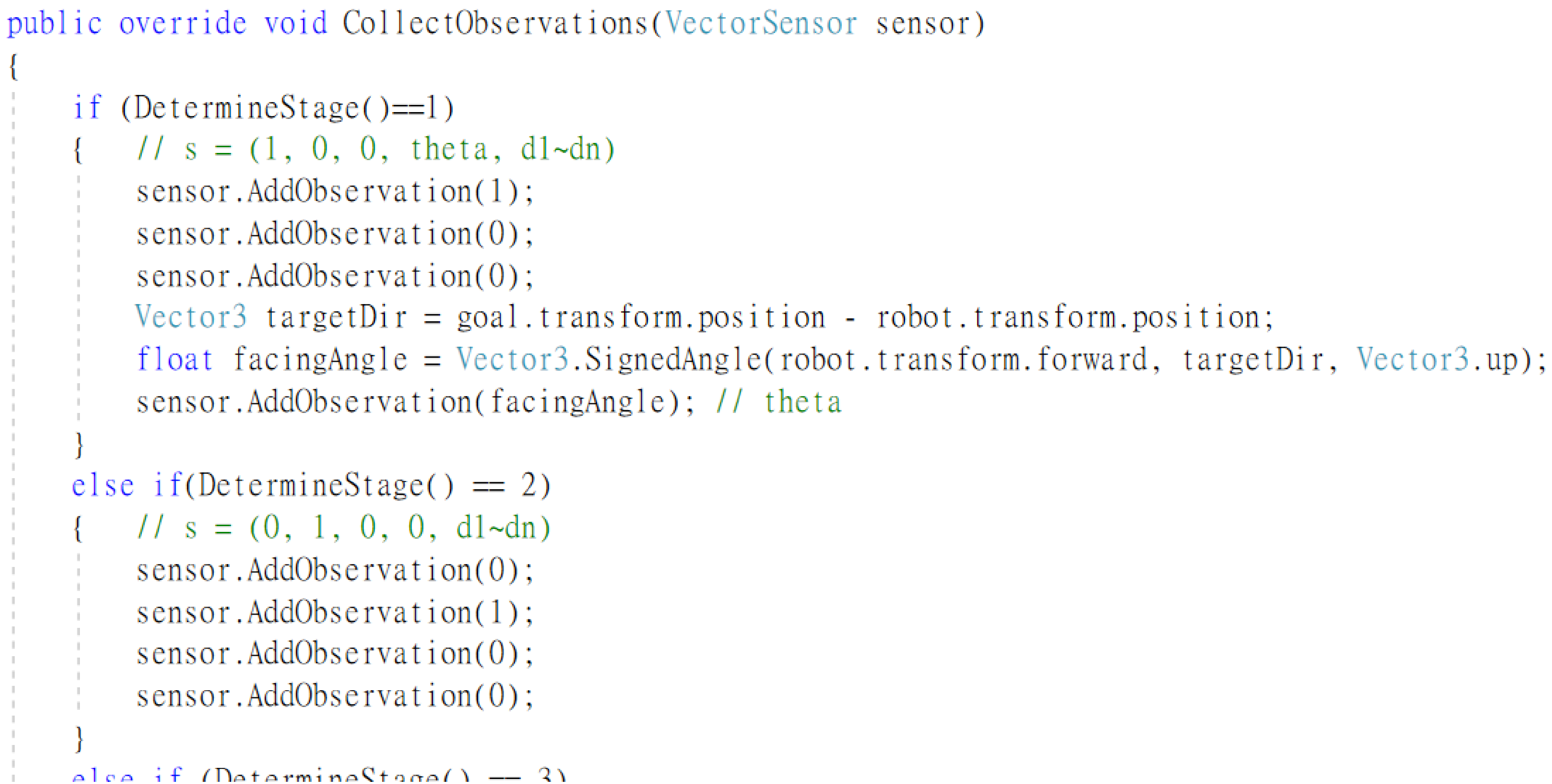
Stage 1



Stage 2

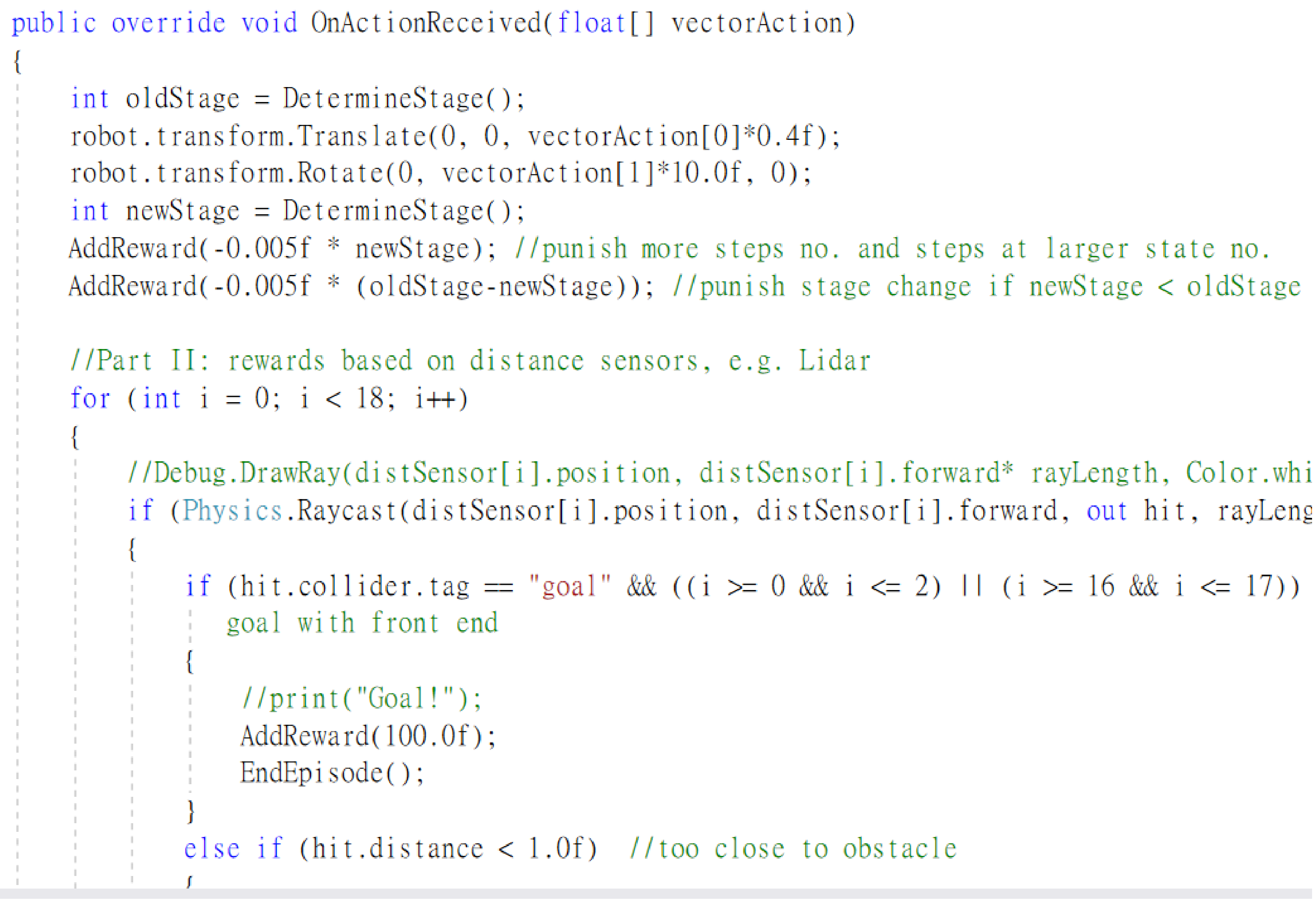


Collect observations



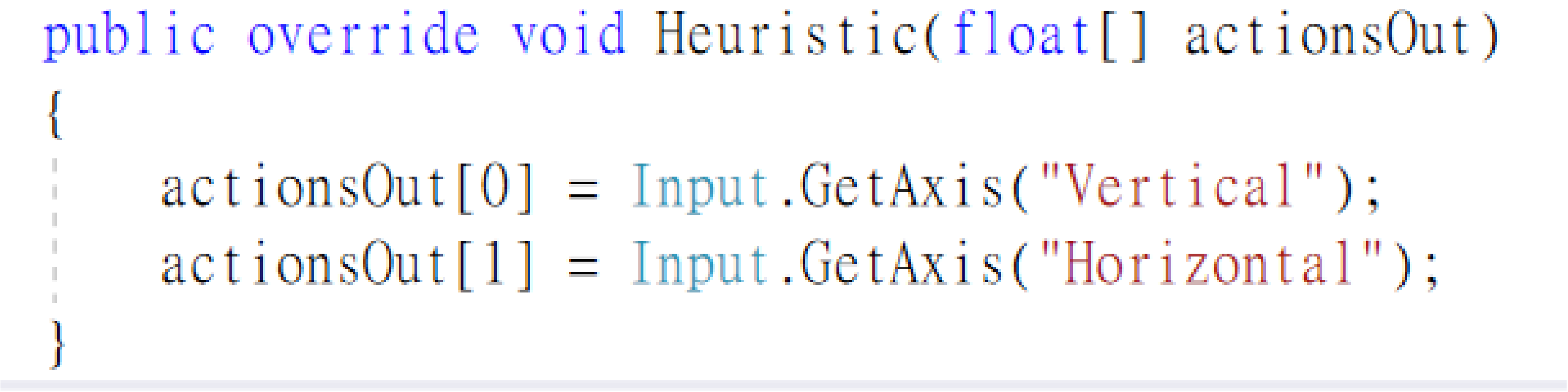
```
public override void CollectObservations(VectorSensor sensor)
{
    if (DetermineStage()==1)
    {   // s = (1, 0, 0, theta, d1~dn)
        sensor.AddObservation(1);
        sensor.AddObservation(0);
        sensor.AddObservation(0);
        Vector3 targetDir = goal.transform.position - robot.transform.position;
        float facingAngle = Vector3.SignedAngle(robot.transform.forward, targetDir, Vector3.up);
        sensor.AddObservation(facingAngle); // theta
    }
    else if(DetermineStage() == 2)
    {   // s = (0, 1, 0, 0, d1~dn)
        sensor.AddObservation(0);
        sensor.AddObservation(1);
        sensor.AddObservation(0);
        sensor.AddObservation(0);
    }
    else if (DetermineStage() == 3)
```

On Action Received



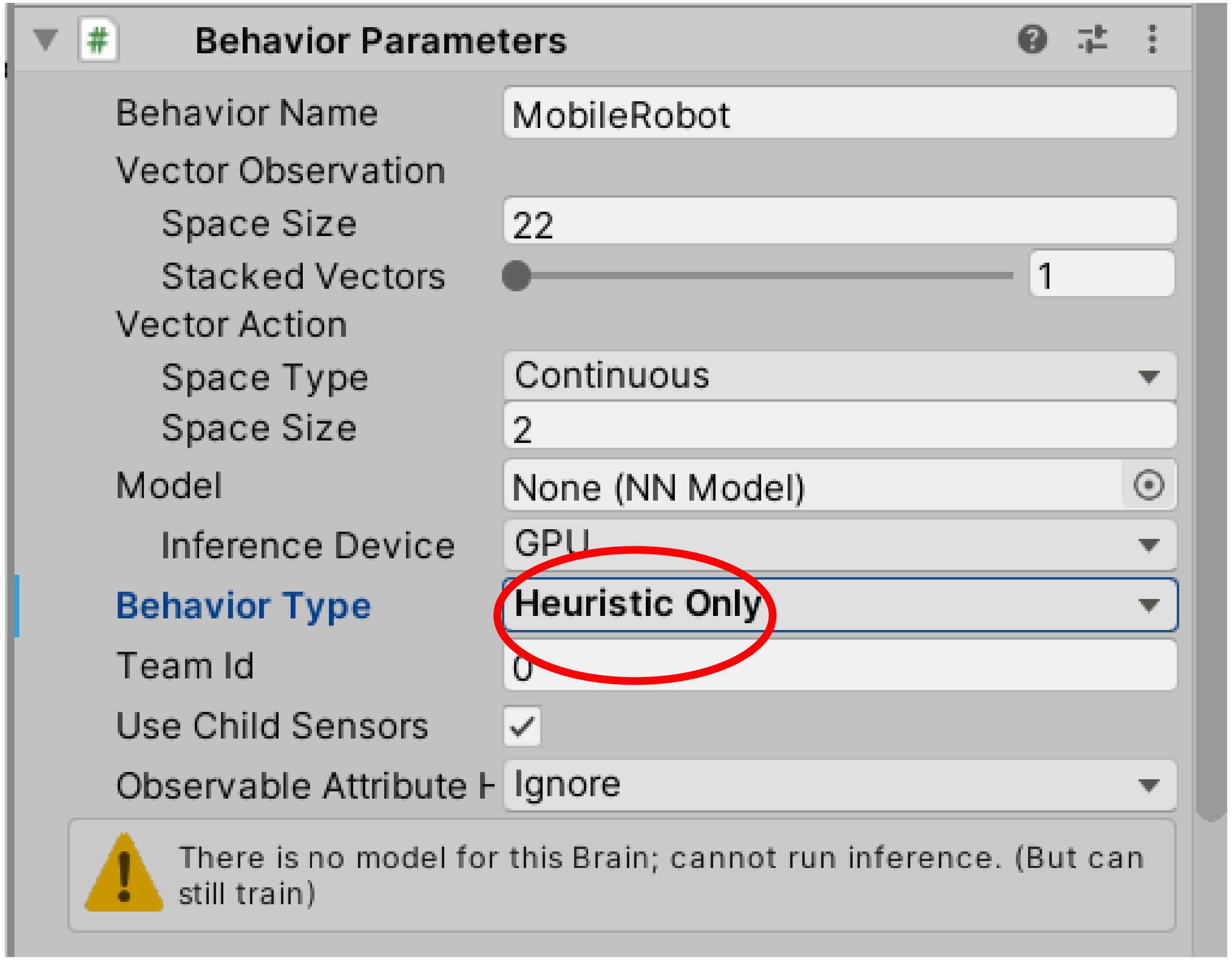
```
public override void OnActionReceived(float[] vectorAction)
{
    int oldStage = DetermineStage();
    robot.transform.Translate(0, 0, vectorAction[0]*0.4f);
    robot.transform.Rotate(0, vectorAction[1]*10.0f, 0);
    int newStage = DetermineStage();
    AddReward(-0.005f * newStage); //punish more steps no. and steps at larger state no.
    AddReward(-0.005f * (oldStage-newStage)); //punish stage change if newStage < oldStage

    //Part II: rewards based on distance sensors, e.g. Lidar
    for (int i = 0; i < 18; i++)
    {
        //Debug.DrawRay(distSensor[i].position, distSensor[i].forward* rayLength, Color.whi
        if (Physics.Raycast(distSensor[i].position, distSensor[i].forward, out hit, rayLeng
        {
            if (hit.collider.tag == "goal" && ((i >= 0 && i <= 2) || (i >= 16 && i <= 17))
               goal with front end
            {
                //print("Goal!");
                AddReward(100.0f);
                EndEpisode();
            }
            else if (hit.distance < 1.0f)  //too close to obstacle
            {
```

Heuristic



```
public override void Heuristic(float[] actionsOut)
{
    actionsOut[0] = Input.GetAxis("Vertical");
    actionsOut[1] = Input.GetAxis("Horizontal");
}
```

5. Test in heuristic mode



Behavior Parameters

Behavior Name: MobileRobot

Vector Observation
- Space Size: 22
- Stacked Vectors: 1

Vector Action
- Space Type: Continuous
- Space Size: 2

Model: None (NN Model)
- Inference Device: GPU

Behavior Type: Heuristic Only

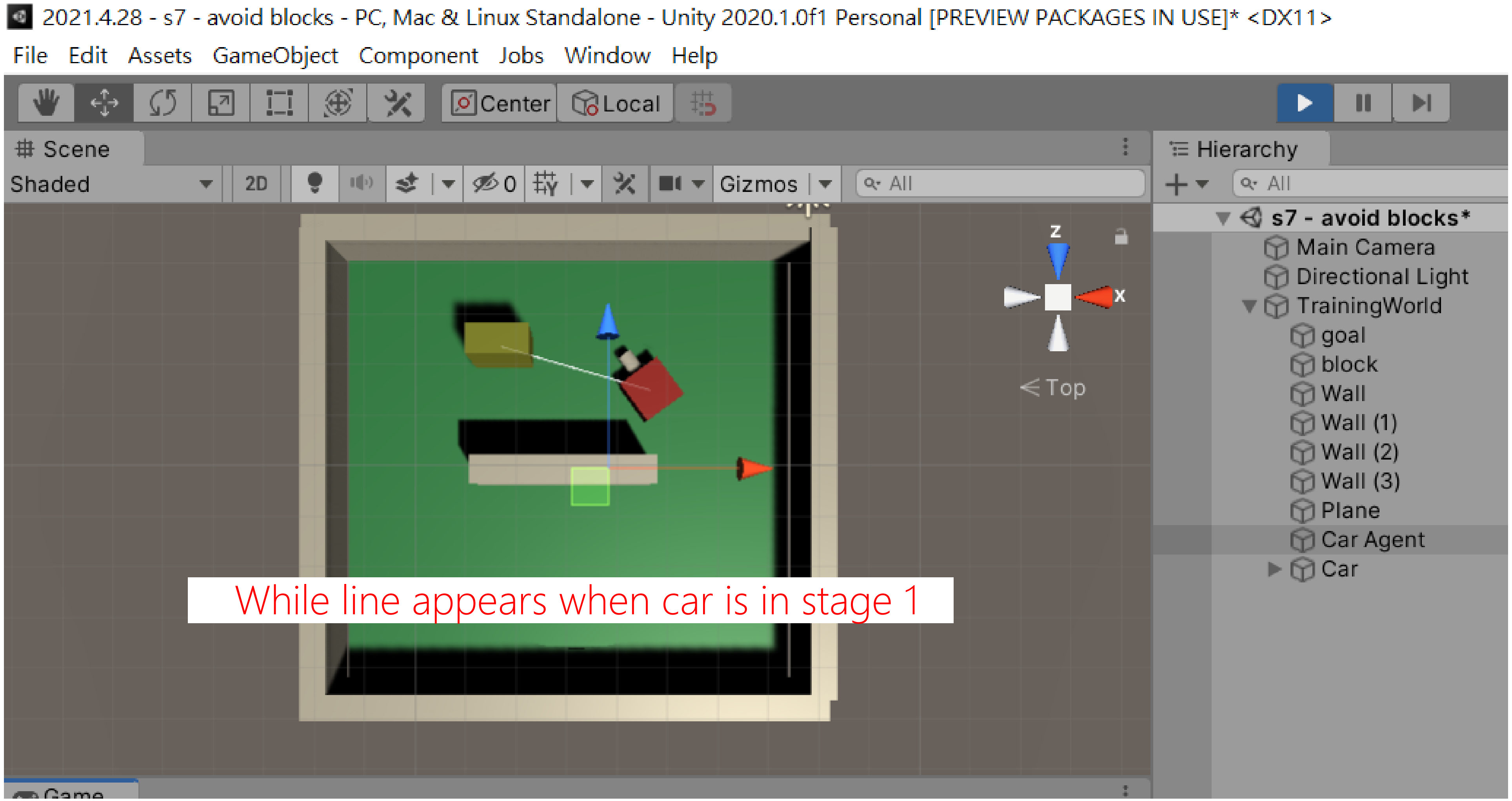
Team Id: 0

Use Child Sensors: ☑

Observable Attribute H: Ignore

There is no model for this Brain; cannot run inference. (But can still train)

5. Test in heuristic mode



2021.4.28 - s7 - avoid blocks - PC, Mac & Linux Standalone - Unity 2020.1.0f1 Personal [PREVIEW PACKAGES IN USE]* <DX11>
File Edit Assets GameObject Component Jobs Window Help
Center
Local
Scene
Shaded
2D
0
Gizmos
All
Z
X
< Top
Hierarchy
All
s7 - avoid blocks*
Main Camera
Directional Light
TrainingWorld
goal
block
Wall
Wall (1)
Wall (2)
Wall (3)
Plane
Car Agent
Car
Game

While line appears when car is in stage 1