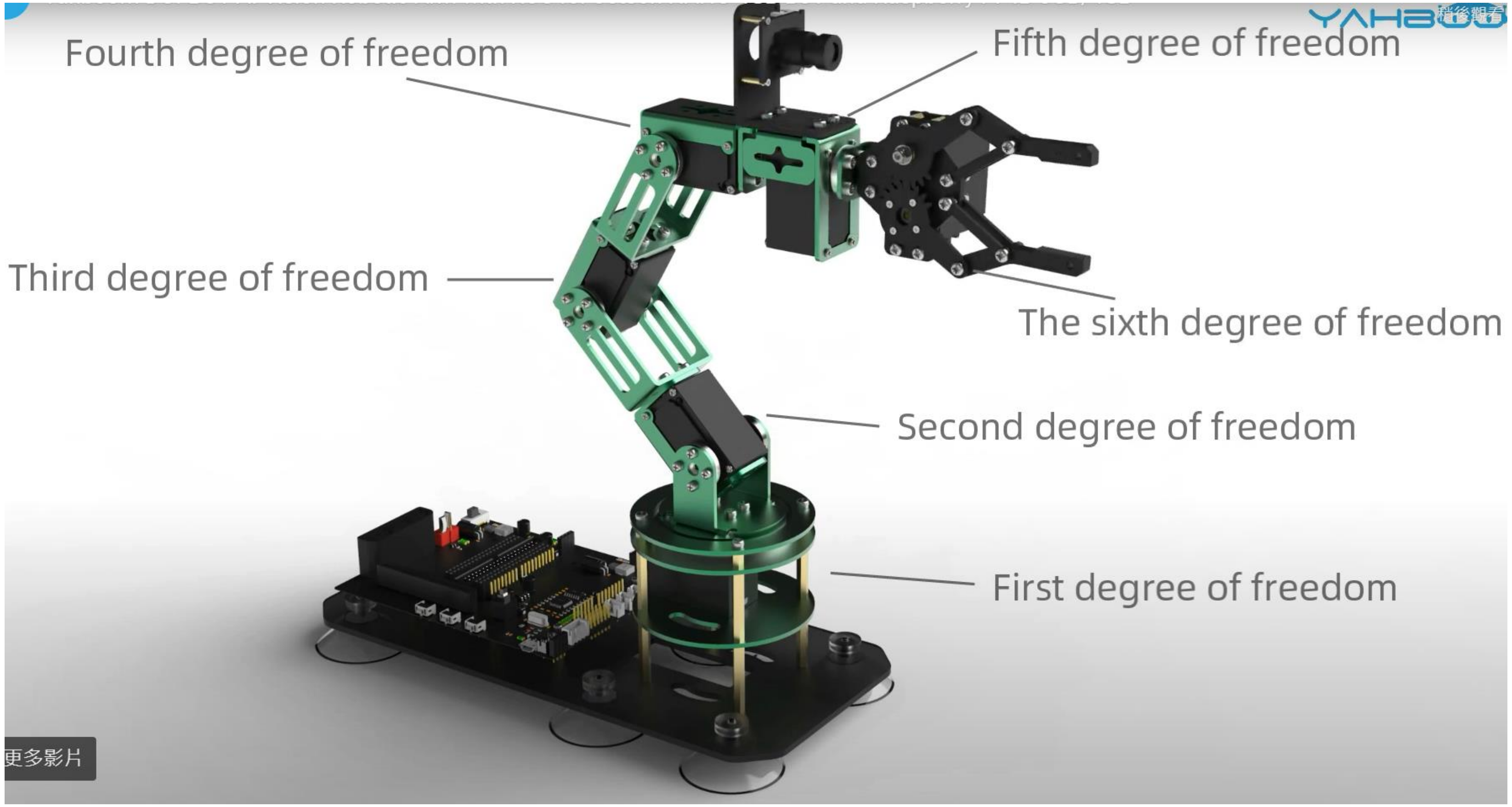
Ultimate goal: build intelligent robot that can interact with human



Yahboom DOFBOT AI Vision Robotic Arm with ROS for Jetson NANO
https://category.yahboom.net/collections/jatson-nano/products/dofbot-jetson_nano

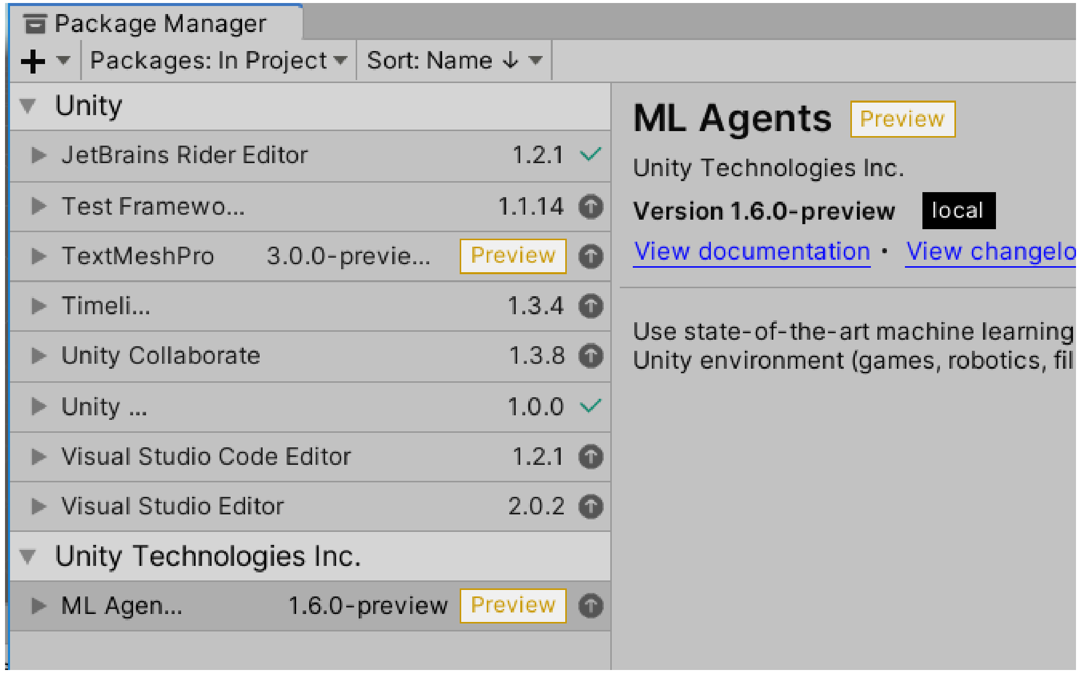
1. Download and save ML Agent to C:\

This will make it convenient to type commands to train and monitor performance

```
cd C:\ml-agents-release_10\config\ppo
mlagents-learn TouchCube.yaml --run-id=1 --force
```

```
cd C:\ml-agents-release_10\config\ppo\results
tensorboard --logdir=1
```

2. Create a new Unity project and import ML Agent package to this new project



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

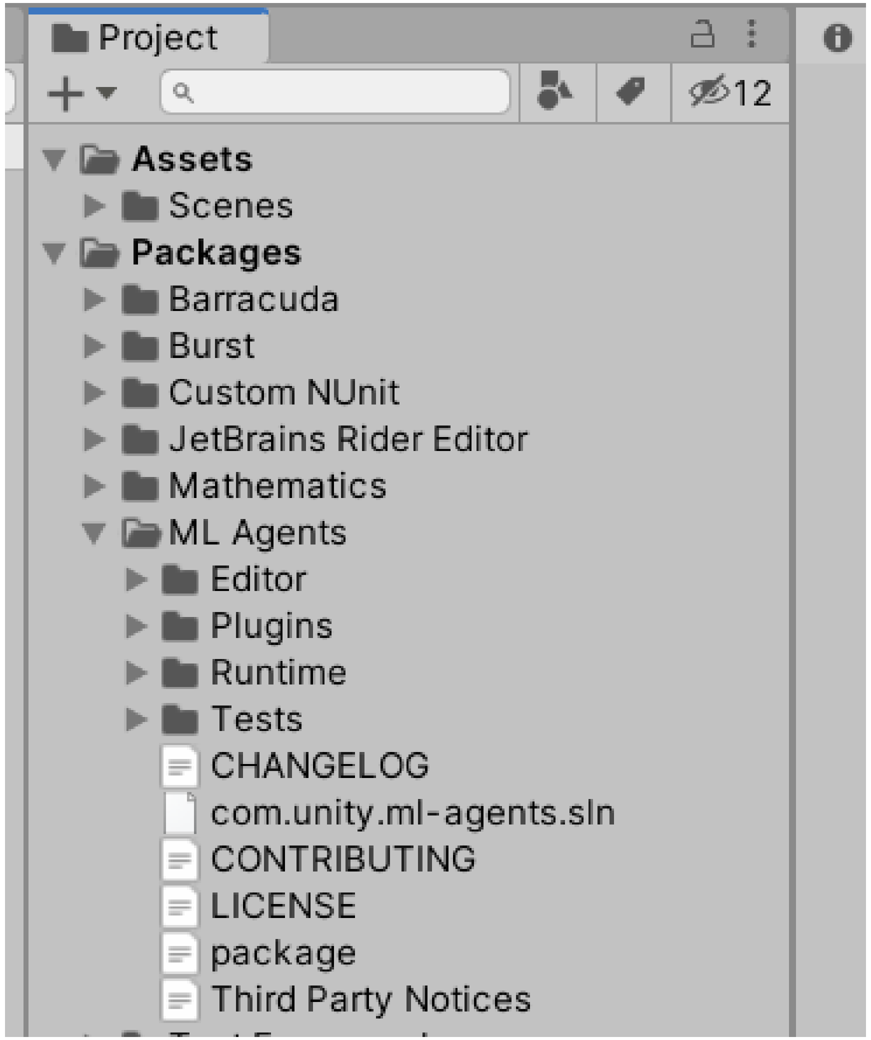
ML Agents Preview

Unity Technologies Inc.

Version 1.6.0-preview local

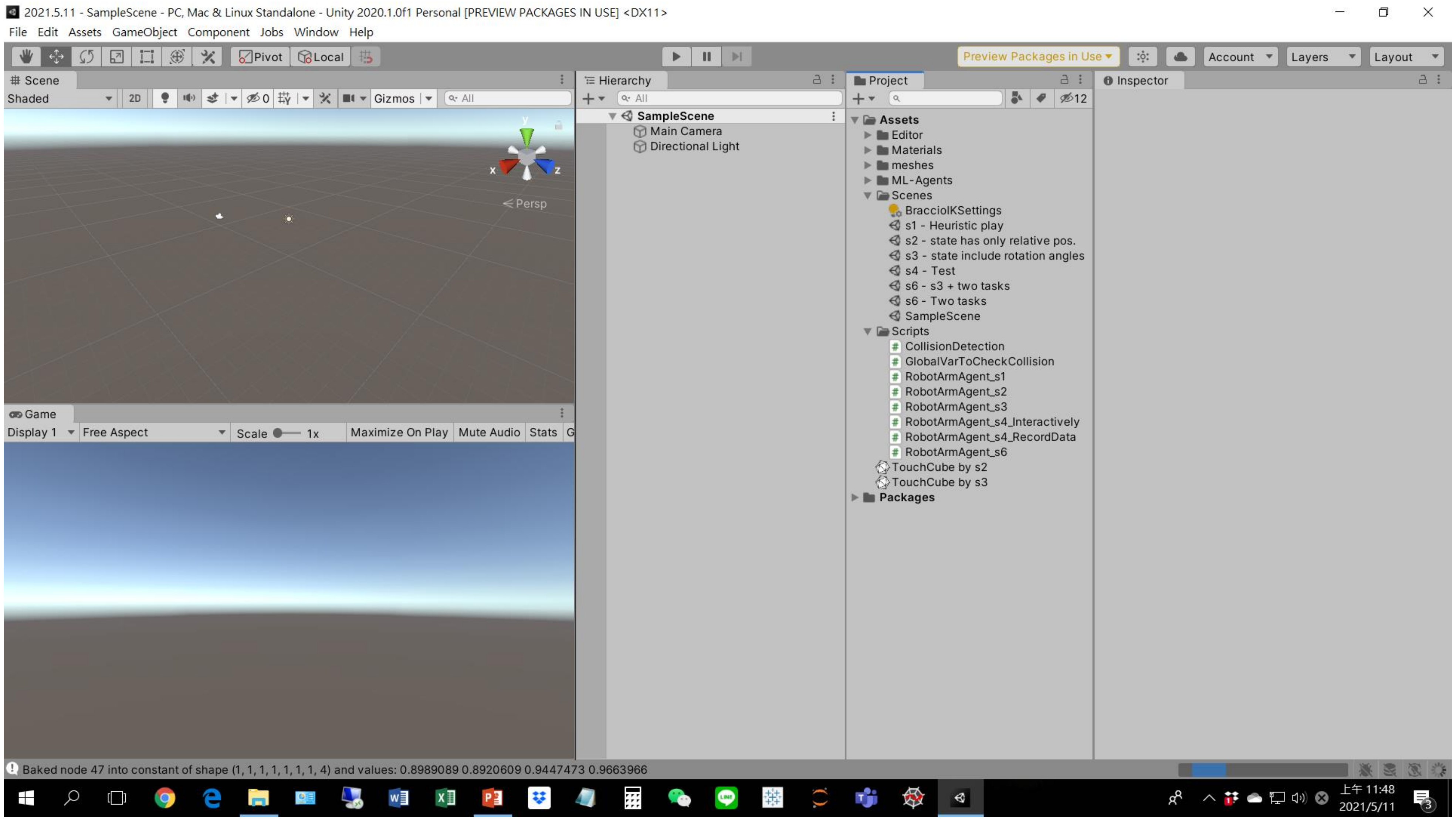
View documentation • View changelo

Use state-of-the-art machine learning
Unity environment (games, robotics, fil



- **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

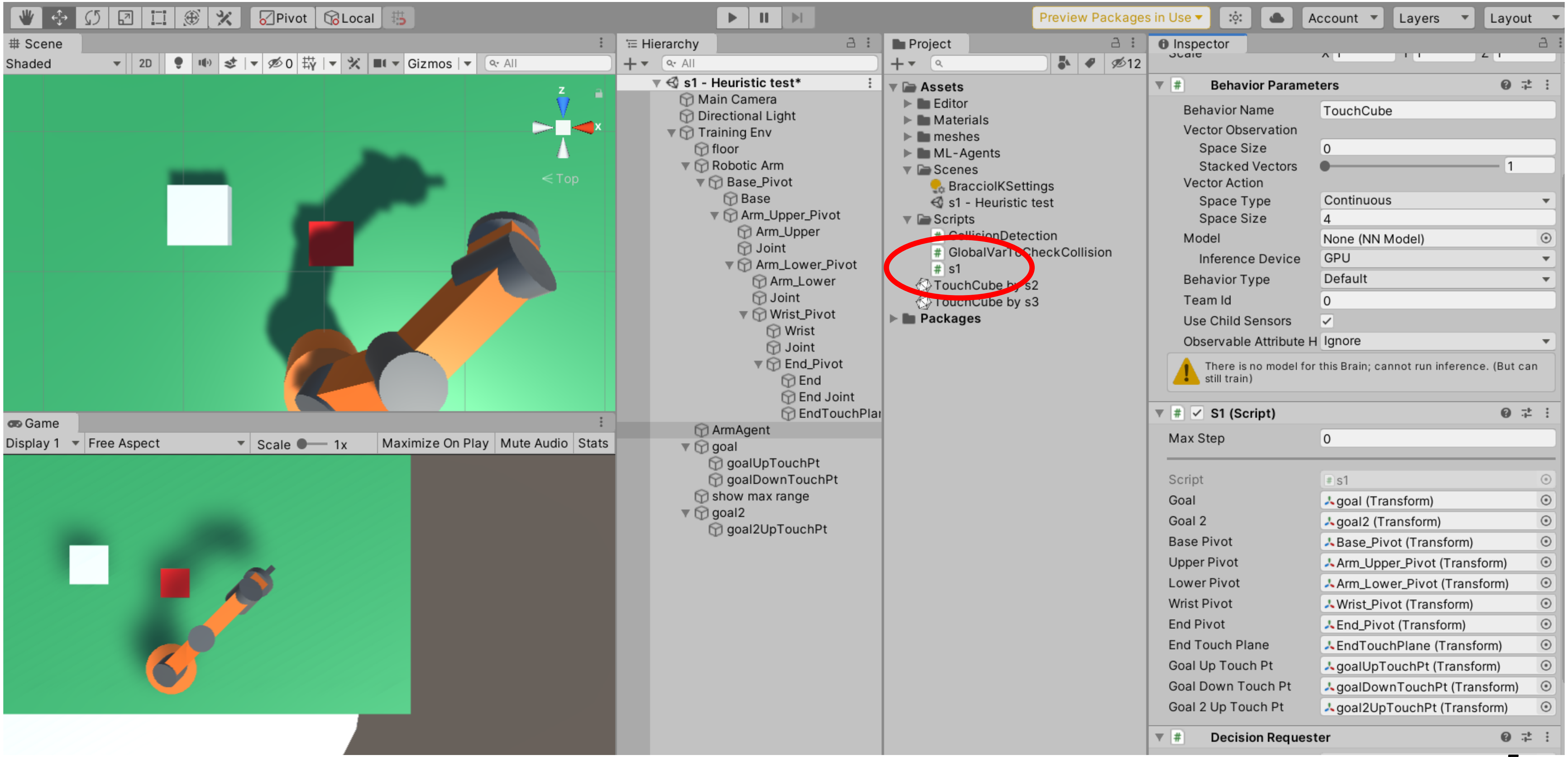
3. Import Robot arm package to Unity project



2021.5.11 - SampleScene - PC, Mac & Linux Standalone - Unity 2020.1.0f1 Personal [PREVIEW PACKAGES IN USE] <DX11>

Baked node 47 into constant of shape (1, 1, 1, 1, 1, 1, 1, 4) and values: 0.8989089 0.8920609 0.9447473 0.9663966

Open scene “s1 - Heuristic play”



Hierarchy:

- s1 - Heuristic test*
  - Main Camera
  - Directional Light
  - Training Env
    - floor
    - Robotic Arm
      - Base_Pivot
        - Base
        - Arm_Upper_Pivot
          - Arm_Upper
          - Joint
          - Arm_Lower_Pivot
            - Arm_Lower
            - Joint
            - Wrist_Pivot
              - Wrist
              - Joint
              - End_Pivot
                - End
                - End Joint
                - EndTouchPlai
    - ArmAgent
    - goal
      - goalUpTouchPt
      - goalDownTouchPt
    - show max range
    - goal2
      - goal2UpTouchPt

Project:

- Assets
  - Editor
  - Materials
  - meshes
  - ML-Agents
  - Scenes
    - BraccioIKSettings
    - s1 - Heuristic test
  - Scripts
    - CollisionDetection
    - GlobalVarToCheckCollision
    - s1
  - TouchCube by s2
  - TouchCube by s3
- Packages

Inspector:

Behavior Parameters

Behavior Name	TouchCube
Vector Observation	
Space Size	0
Stacked Vectors	1
Vector Action	
Space Type	Continuous
Space Size	4
Model	None (NN Model)
Inference Device	GPU
Behavior Type	Default
Team Id	0
Use Child Sensors	✓
Observable Attribute H	Ignore

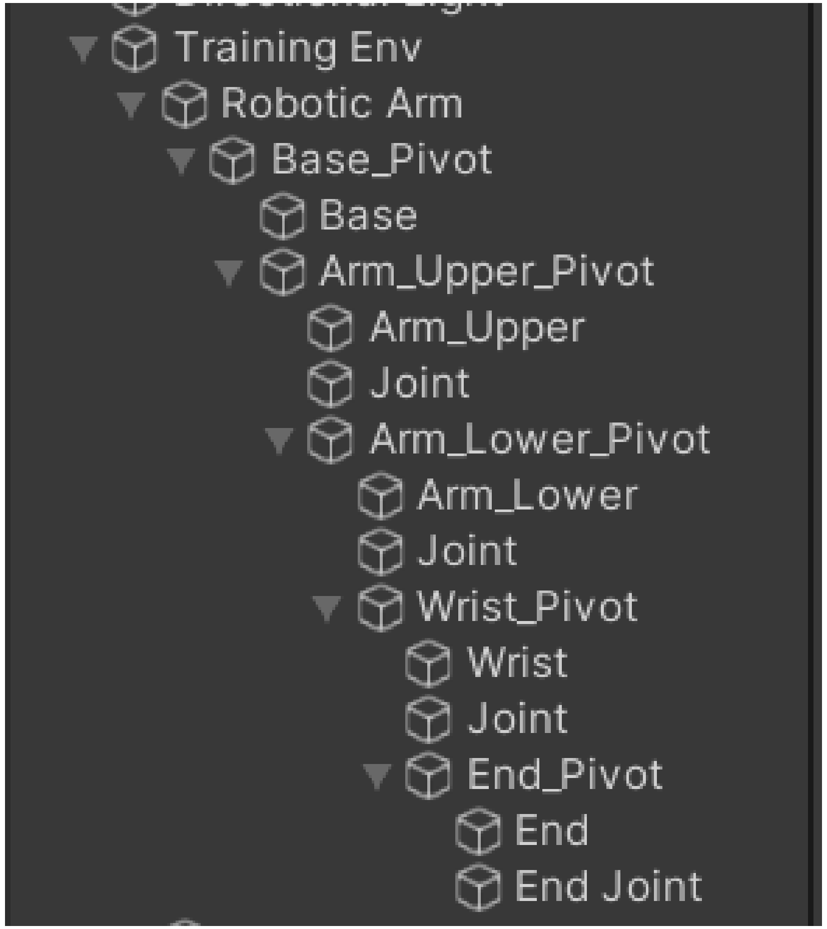
There is no model for this Brain; cannot run inference. (But can still train)

S1 (Script)

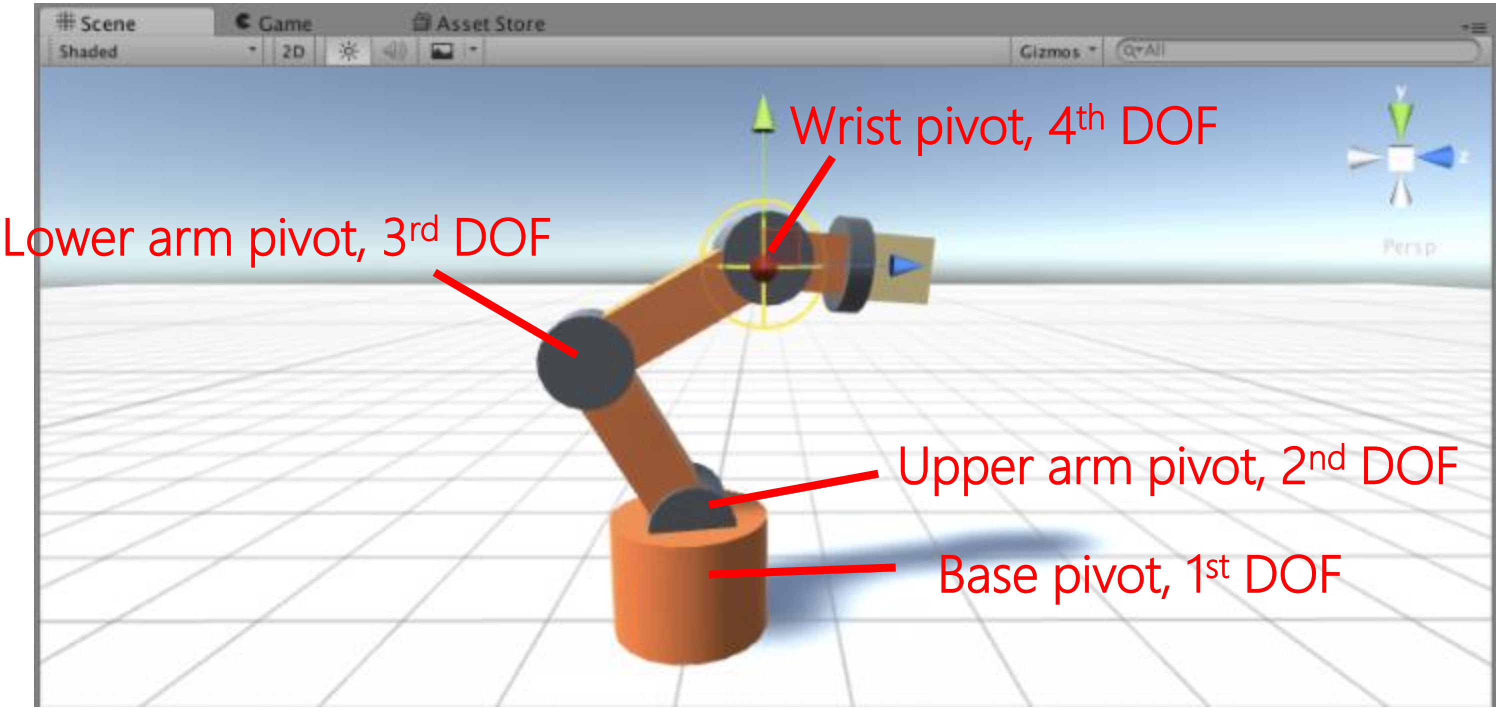
Max Step	0
Script	s1
Goal	goal (Transform)
Goal 2	goal2 (Transform)
Base Pivot	Base_Pivot (Transform)
Upper Pivot	Arm_Upper_Pivot (Transform)
Lower Pivot	Arm_Lower_Pivot (Transform)
Wrist Pivot	Wrist_Pivot (Transform)
End Pivot	End_Pivot (Transform)
End Touch Plane	EndTouchPlane (Transform)
Goal Up Touch Pt	goalUpTouchPt (Transform)
Goal Down Touch Pt	goalDownTouchPt (Transform)
Goal 2 Up Touch Pt	goal2UpTouchPt (Transform)

Decision Requester

This Unity project contains a Braccio robot arm



- Training Env
  - Robotic Arm
    - Base_Pivot
      - Base
      - Arm_Upper_Pivot
        - Arm_Upper
        - Joint
        - Arm_Lower_Pivot
          - Arm_Lower
          - Joint
          - Wrist_Pivot
            - Wrist
            - Joint
            - End_Pivot
              - End
              - End Joint

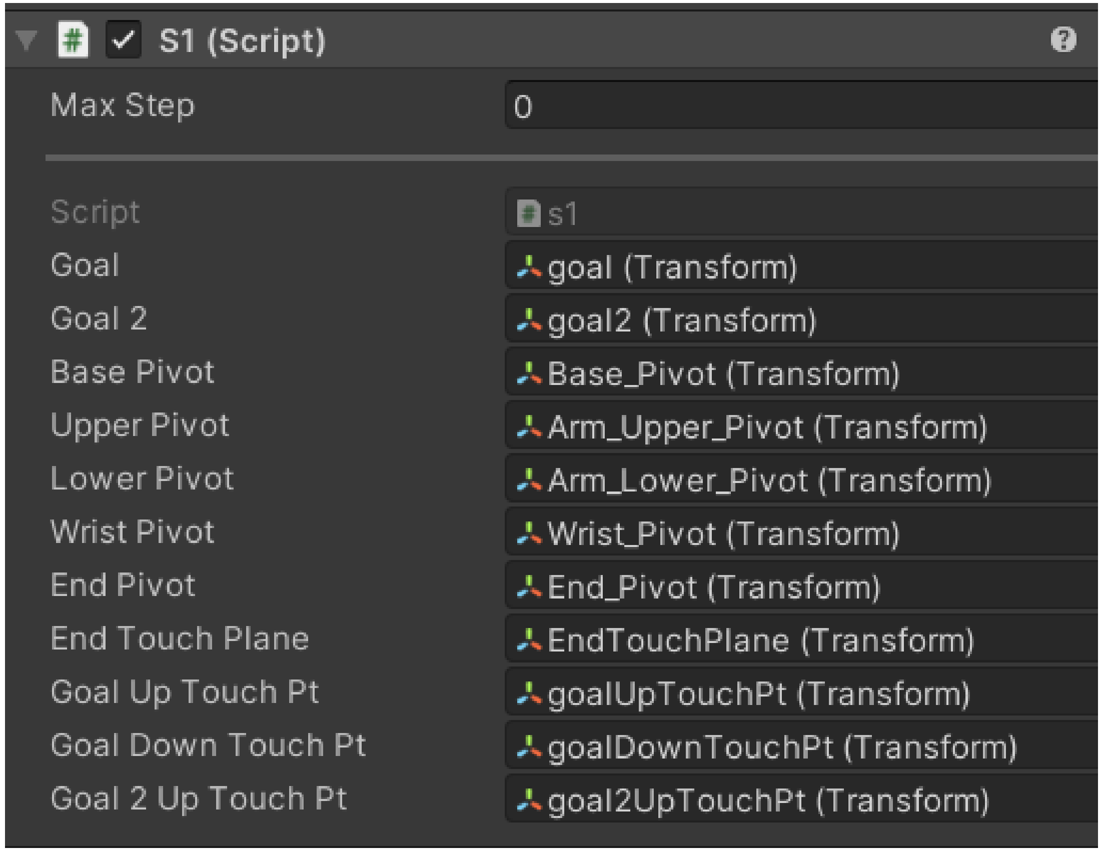


https://github.com/tanyuan/braccio-ik-unity

Manually manipulate the 4DOF robot arm

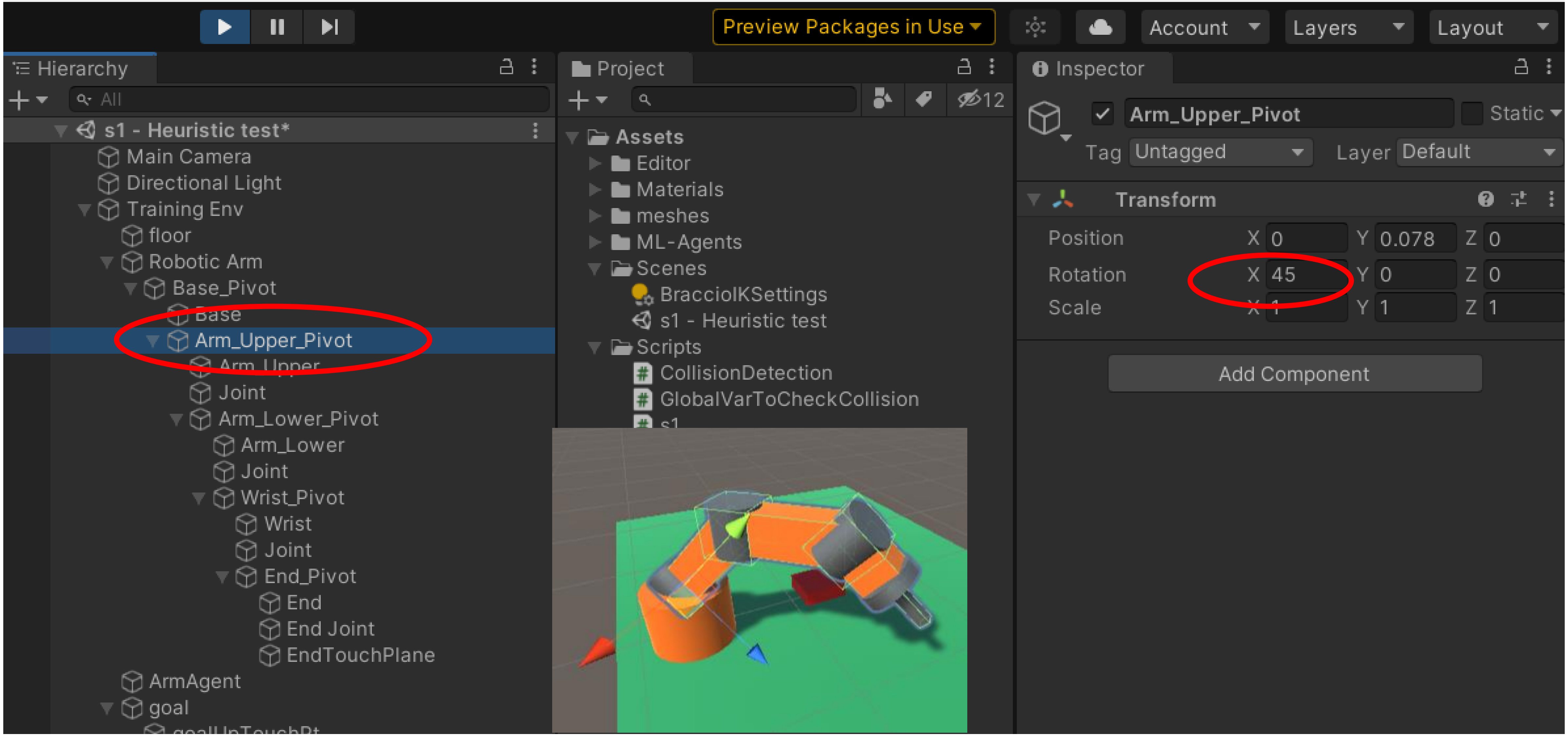
Public variables to link agent script with scene objects

```
public Transform goal, goal2;
public Transform BasePivot, UpperPivot, LowerPivot, WristPivot, EndPivot;
public Transform EndTouchPlane, goalUpTouchPt, goalDownTouchPt, goal2UpTouchPt;
int stage = 1;
```



```
▼ # ☑ S1 (Script)
Max Step                0
Script                  s1
Goal                    goal (Transform)
Goal 2                  goal2 (Transform)
Base Pivot              Base_Pivot (Transform)
Upper Pivot             Arm_Upper_Pivot (Transform)
Lower Pivot             Arm_Lower_Pivot (Transform)
Wrist Pivot             Wrist_Pivot (Transform)
End Pivot               End_Pivot (Transform)
End Touch Plane         EndTouchPlane (Transform)
Goal Up Touch Pt        goalUpTouchPt (Transform)
Goal Down Touch Pt      goalDownTouchPt (Transform)
Goal 2 Up Touch Pt      goal2UpTouchPt (Transform)
```

Play, rotate arm by changing rotation angle in Inspector window



```
s1 - Heuristic test*
    Main Camera
    Directional Light
    Training Env
        floor
        Robotic Arm
            Base_Pivot
                Base
                Arm_Upper_Pivot
                    Arm_Upper
                    Joint
                    Arm_Lower_Pivot
                        Arm_Lower
                        Joint
                        Wrist_Pivot
                            Wrist
                            Joint
                            End_Pivot
                                End
                                End Joint
                                EndTouchPlane
        ArmAgent
        goal
            goalUpTouchPt
```

```
Assets
    Editor
    Materials
    meshes
    ML-Agents
    Scenes
        BraccioIKSettings
        s1 - Heuristic test
    Scripts
        CollisionDetection
        GlobalVarToCheckCollision
        s1
```

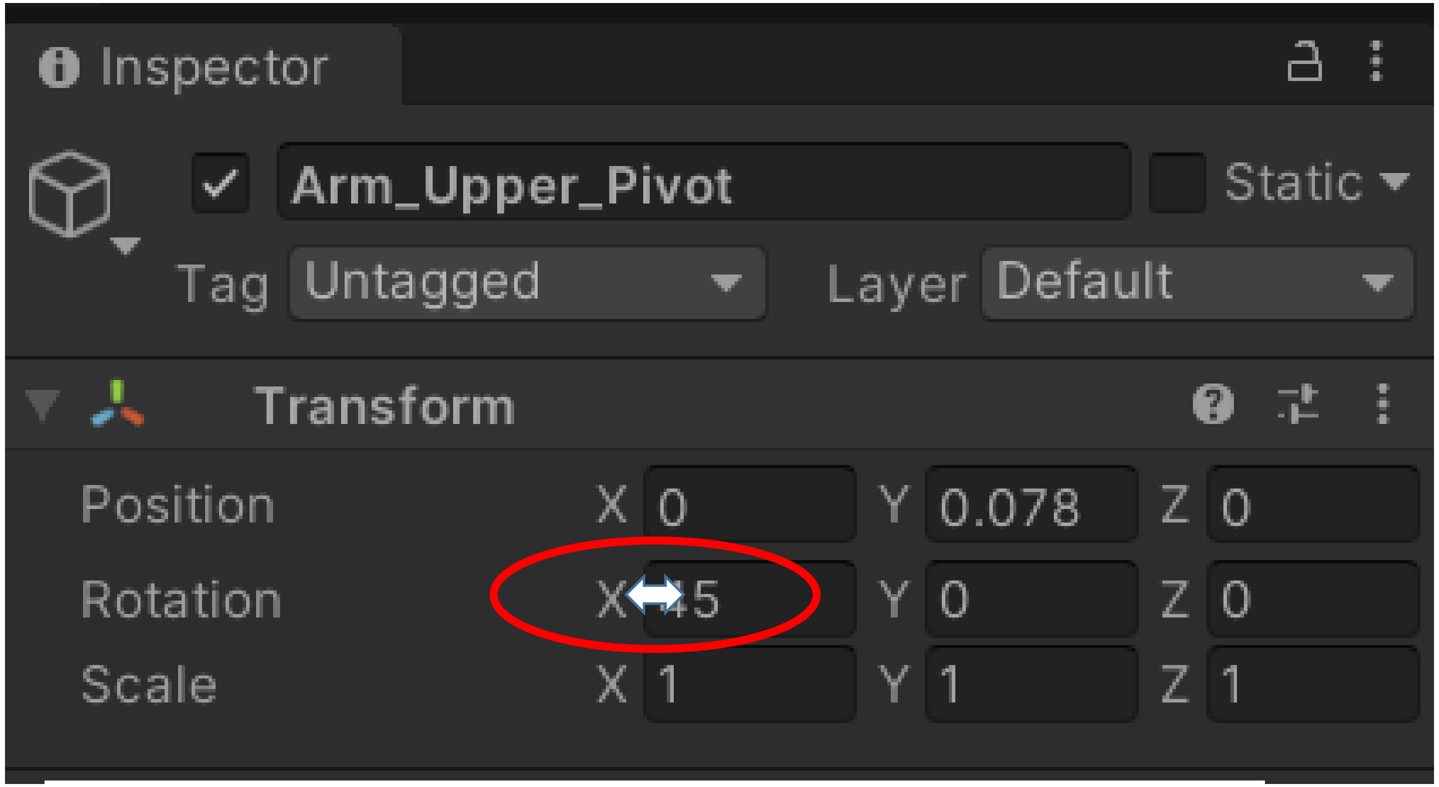
Arm_Upper_Pivot ☐ Static

Tag Untagged Layer Default

Transform

Position	X	0	Y	0.078	Z	0
Rotation	X	45	Y	0	Z	0
Scale	X	1	Y	1	Z	1

Play, rotate arm by changing rotation angle in Inspector window



Inspector

☑ Arm_Upper_Pivot □ Static

Tag Untagged Layer Default

Transform

Position	X	0	Y	0.078	Z	0
Rotation	X	45	Y	0	Z	0
Scale	X	1	Y	1	Z	1

Place your mouse here and you can use the arrow to easily adjust the values

Use Up/Down, L/R keys to rotate arm

← and → key

↑ and ↓ key

```
actionsOut[0] = Input.GetAxis("Horizontal");
actionsOut[1] = Input.GetAxis("Vertical");
```

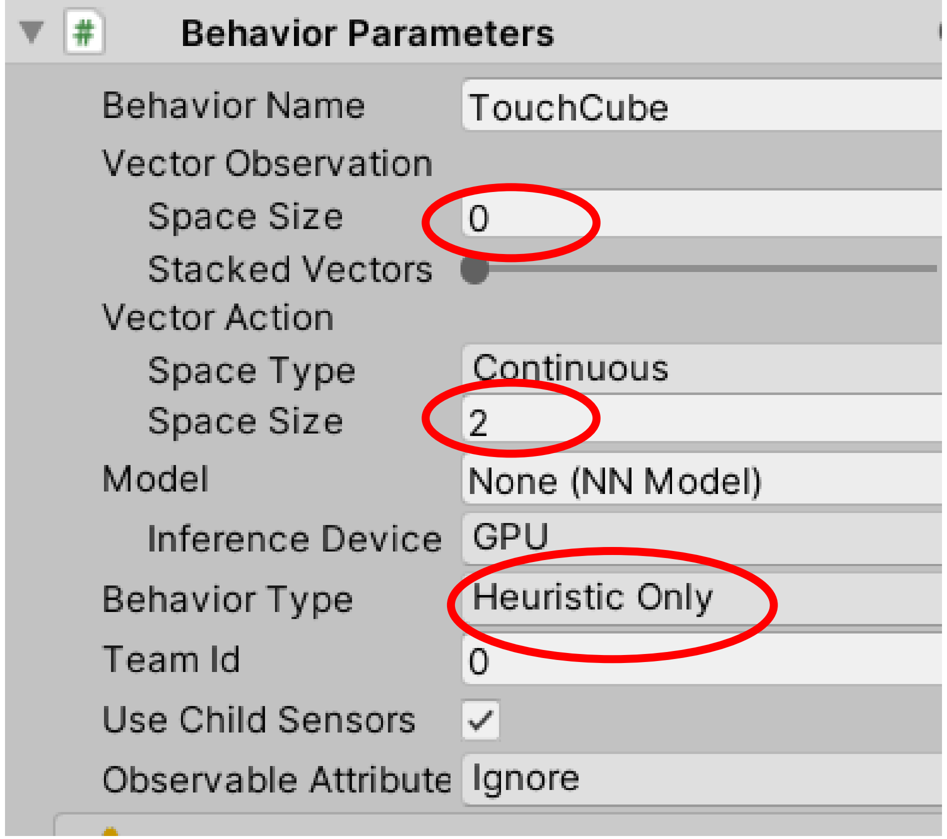
Base -90 ~ 90

U arm range: 0 ~ 90

L arm, Wrist: -90~90

```
BasePivot.Rotate(0, vectorAction[0] * speed, 0);
float RotationAngle = UnityEditor.TransformUtils.GetInspectorRotation(BasePivot).y;

UpperPivot.Rotate(vectorAction[1] * speed, 0, 0);
//float RotationAngle = UnityEditor.TransformUtils.GetInspectorRotation(UpperPivot).x;
```



Behavior Parameters

Behavior Name: TouchCube

Vector Observation

Space Size: 0

Stacked Vectors

Vector Action

Space Type: Continuous

Space Size: 2

Model: None (NN Model)

Inference Device: GPU

Behavior Type: Heuristic Only

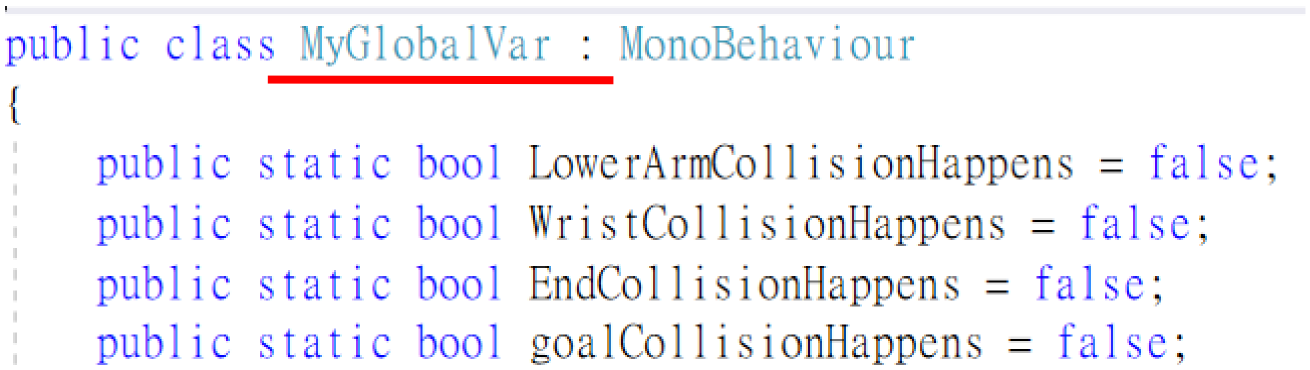
Team Id: 0

Use Child Sensors: ✓

Observable Attribute: Ignore

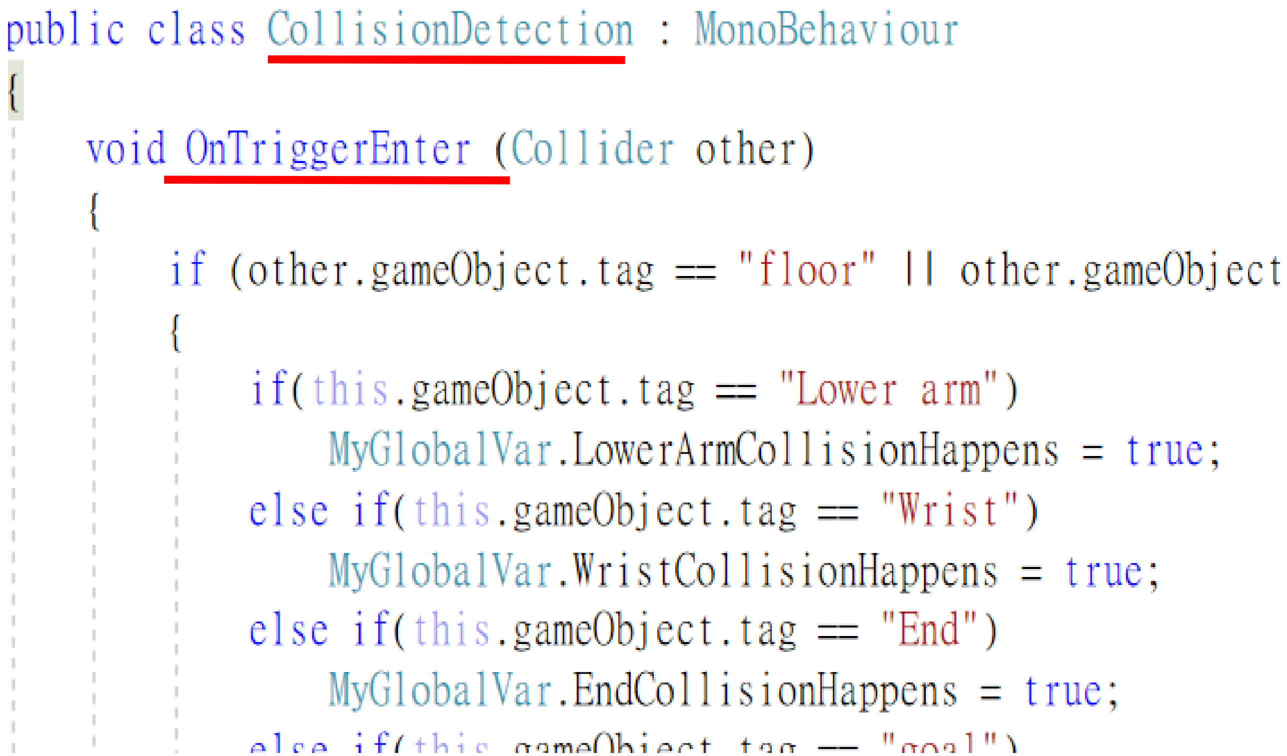
Collision detection

Static global variables to record collision of lower arm, wrist, end, and goal

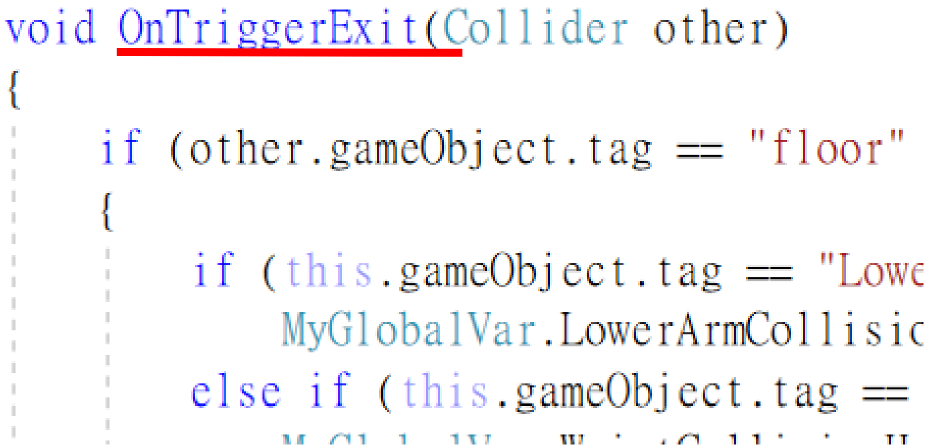


```
public class MyGlobalVar : MonoBehaviour
{
    public static bool LowerArmCollisionHappens = false;
    public static bool WristCollisionHappens = false;
    public static bool EndCollisionHappens = false;
    public static bool goalCollisionHappens = false;
```

On trigger enter/exist to record collisions of lower arm, wrist, and end with floor and goal objects

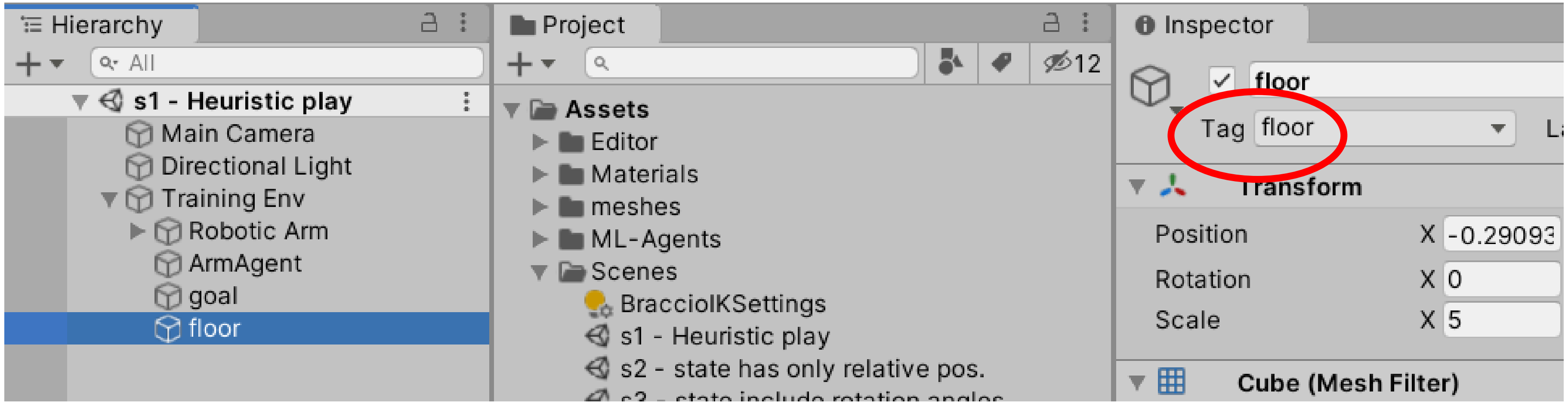


```
public class CollisionDetection : MonoBehaviour
{
    void OnTriggerEnter (Collider other)
    {
        if (other.gameObject.tag == "floor" || other.gameObject
        {
            if(this.gameObject.tag == "Lower arm")
                MyGlobalVar.LowerArmCollisionHappens = true;
            else if(this.gameObject.tag == "Wrist")
                MyGlobalVar.WristCollisionHappens = true;
            else if(this.gameObject.tag == "End")
                MyGlobalVar.EndCollisionHappens = true;
            else if(this.gameObject.tag == "goal")
```



```
void OnTriggerExit(Collider other)
{
    if (other.gameObject.tag == "floor"
    {
        if (this.gameObject.tag == "Lowe
            MyGlobalVar.LowerArmCollisic
        else if (this.gameObject.tag ==
```

Add tags to floor and goal object



Hierarchy

All

- s1 - Heuristic play
  - Main Camera
  - Directional Light
  - Training Env
    - Robotic Arm
    - ArmAgent
    - goal
    - floor

Project

- Assets
  - Editor
  - Materials
  - meshes
  - ML-Agents
  - Scenes
    - BraccioIKSettings
    - s1 - Heuristic play
    - s2 - state has only relative pos.
    - s3 - state include rotation angles

Inspector

☑ floor

Tag floor

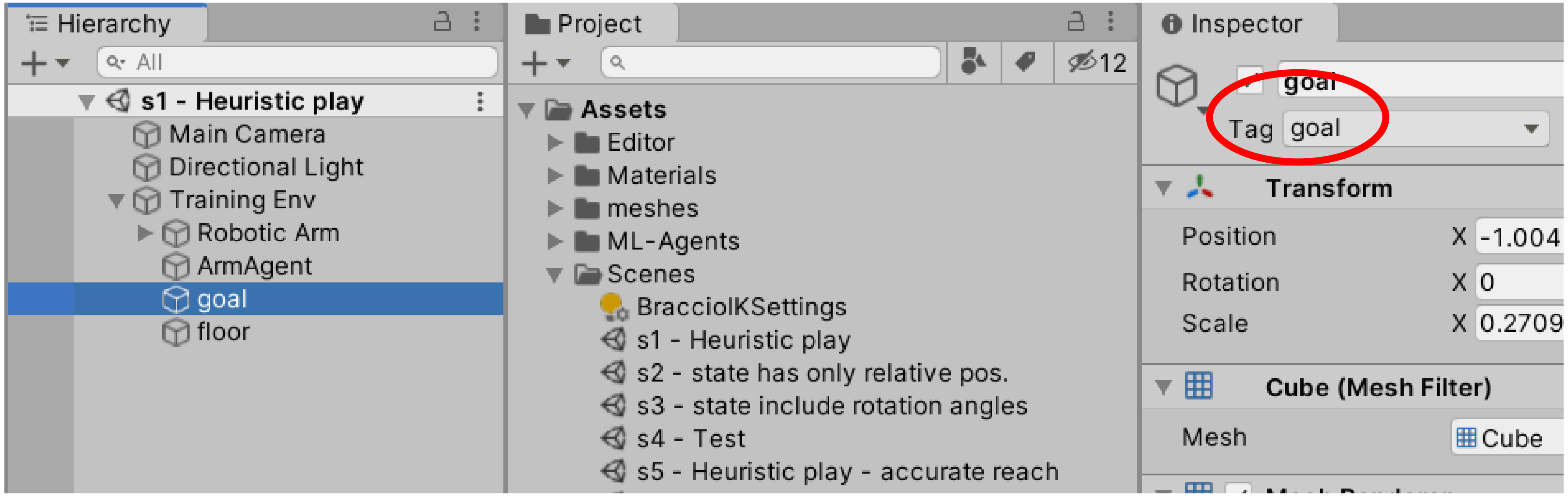
Transform

Position X -0.29093

Rotation X 0

Scale X 5

Cube (Mesh Filter)



Hierarchy

All

- s1 - Heuristic play
  - Main Camera
  - Directional Light
  - Training Env
    - Robotic Arm
    - ArmAgent
    - goal
    - floor

Project

- Assets
  - Editor
  - Materials
  - meshes
  - ML-Agents
  - Scenes
    - BraccioIKSettings
    - s1 - Heuristic play
    - s2 - state has only relative pos.
    - s3 - state include rotation angles
    - s4 - Test
    - s5 - Heuristic play - accurate reach

Inspector

goal

Tag goal

Transform

Position X -1.004

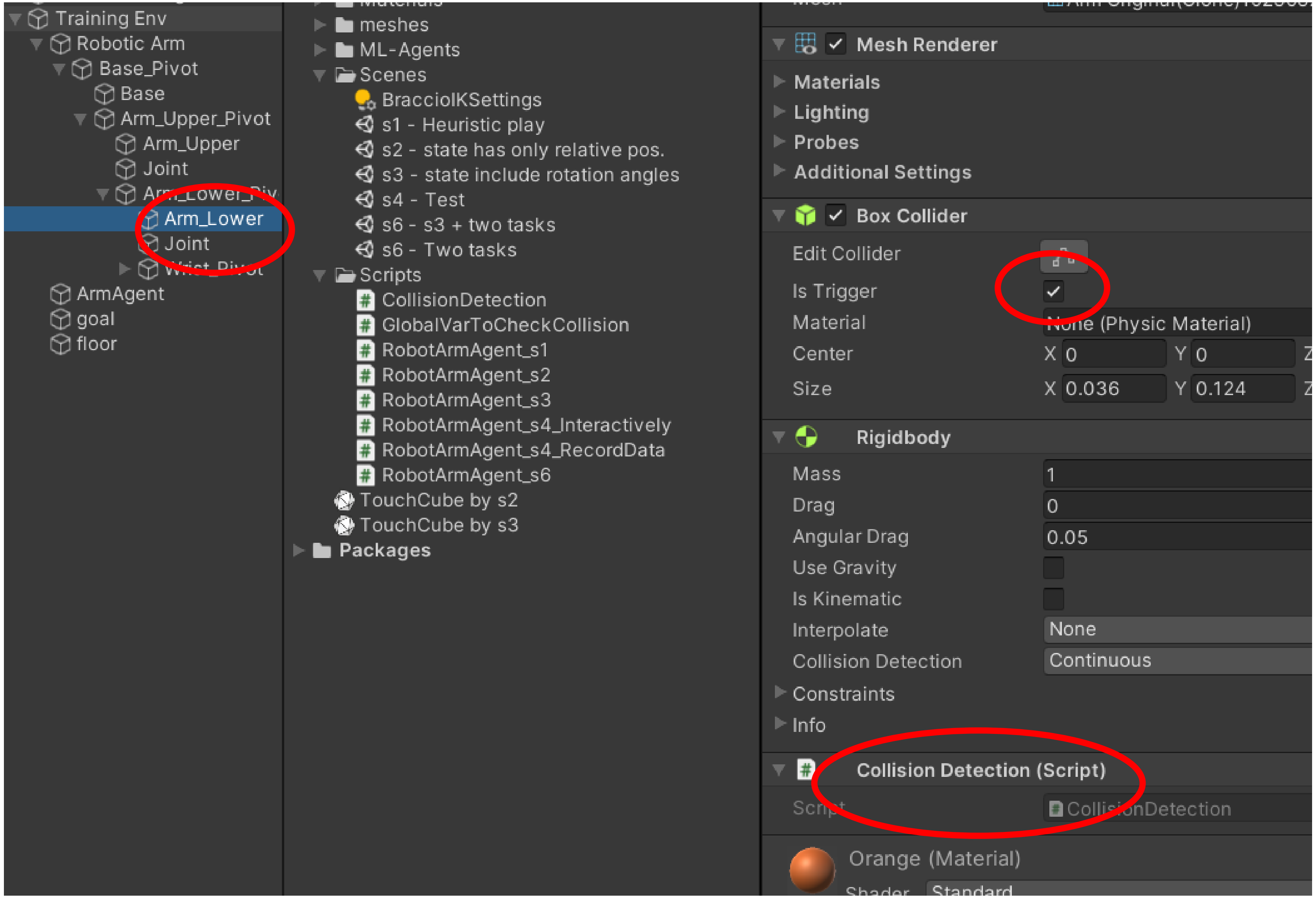
Rotation X 0

Scale X 0.2709

Cube (Mesh Filter)

Mesh Cube

Add trigger collider, Rigid body, and collision detection script to Lower Arm



☑ Mesh Renderer

- Materials
- Lighting
- Probes
- Additional Settings

☑ Box Collider

Edit Collider

Is Trigger ☑

Material None (Physic Material)

Center X 0 Y 0

Size X 0.036 Y 0.124

Rigidbody

Mass 1

Drag 0

Angular Drag 0.05

Use Gravity □

Is Kinematic □

Interpolate None

Collision Detection Continuous

- Constraints
- Info

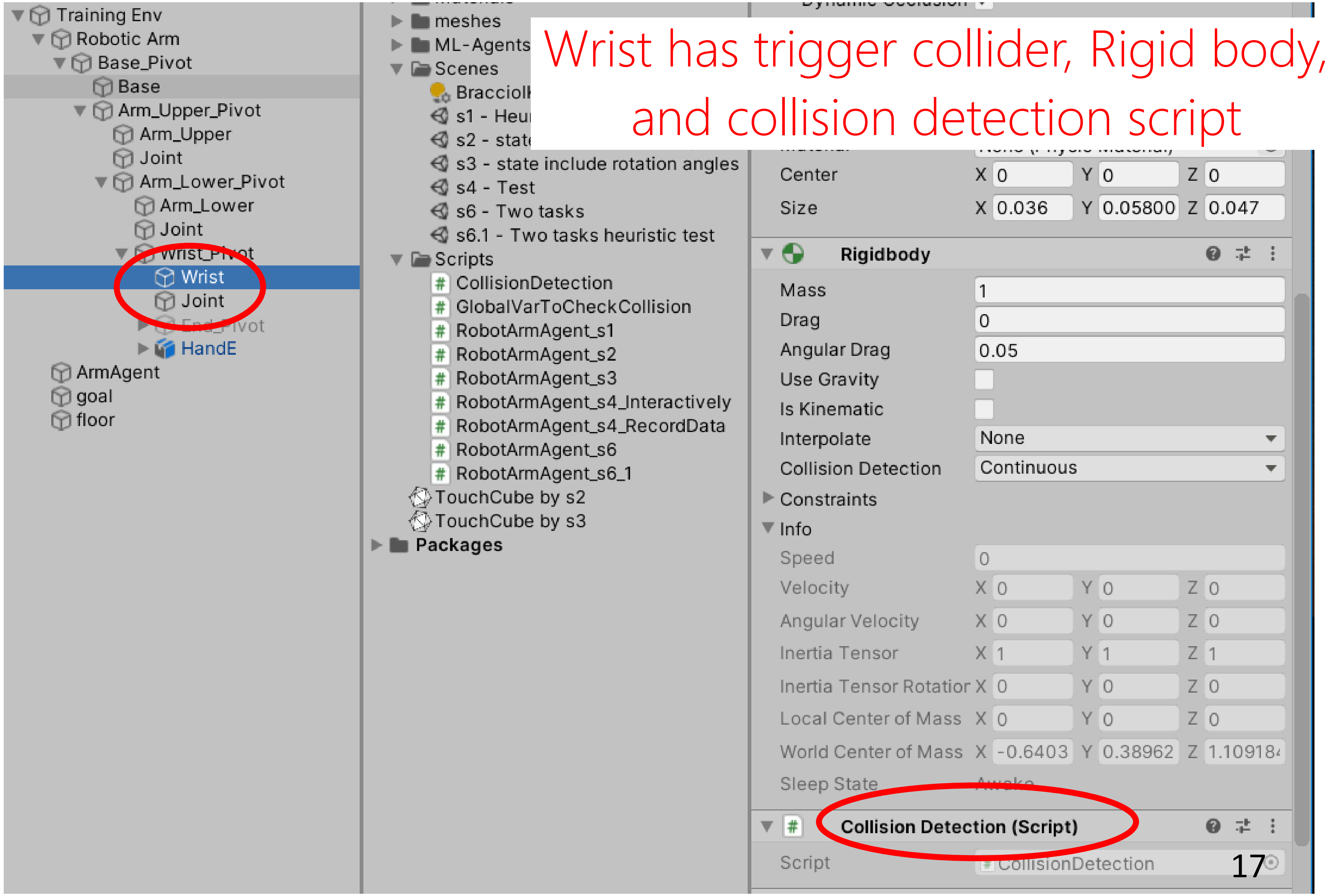
Collision Detection (Script)

Script CollisionDetection

Orange (Material)

Shader Standard

Add trigger collider, Rigid body, and collision detection script to Wrist



Wrist has trigger collider, Rigid body, and collision detection script

- Training Env
  - Robotic Arm
    - Base_Pivot
      - Base
      - Arm_Upper_Pivot
        - Arm_Upper
        - Joint
        - Arm_Lower_Pivot
          - Arm_Lower
          - Joint
          - Wrist_Pivot
            - Wrist
            - Joint
          - End_Pivot
          - HandE
  - ArmAgent
  - goal
  - floor

- meshes
- ML-Agents
- Scenes
  - Bracciol
  - s1 - Heu
  - s2 - stat
  - s3 - state include rotation angles
  - s4 - Test
  - s6 - Two tasks
  - s6.1 - Two tasks heuristic test
- Scripts
  - CollisionDetection
  - GlobalVarToCheckCollision
  - RobotArmAgent_s1
  - RobotArmAgent_s2
  - RobotArmAgent_s3
  - RobotArmAgent_s4_Interactively
  - RobotArmAgent_s4_RecordData
  - RobotArmAgent_s6
  - RobotArmAgent_s6_1
  - TouchCube by s2
  - TouchCube by s3
- Packages

Center X 0 Y 0 Z 0

Size X 0.036 Y 0.05800 Z 0.047

Rigidbody

Mass 1

Drag 0

Angular Drag 0.05

Use Gravity □

Is Kinematic □

Interpolate None

Collision Detection Continuous

Constraints

Info

Speed 0

Velocity X 0 Y 0 Z 0

Angular Velocity X 0 Y 0 Z 0

Inertia Tensor X 1 Y 1 Z 1

Inertia Tensor Rotation X 0 Y 0 Z 0

Local Center of Mass X 0 Y 0 Z 0

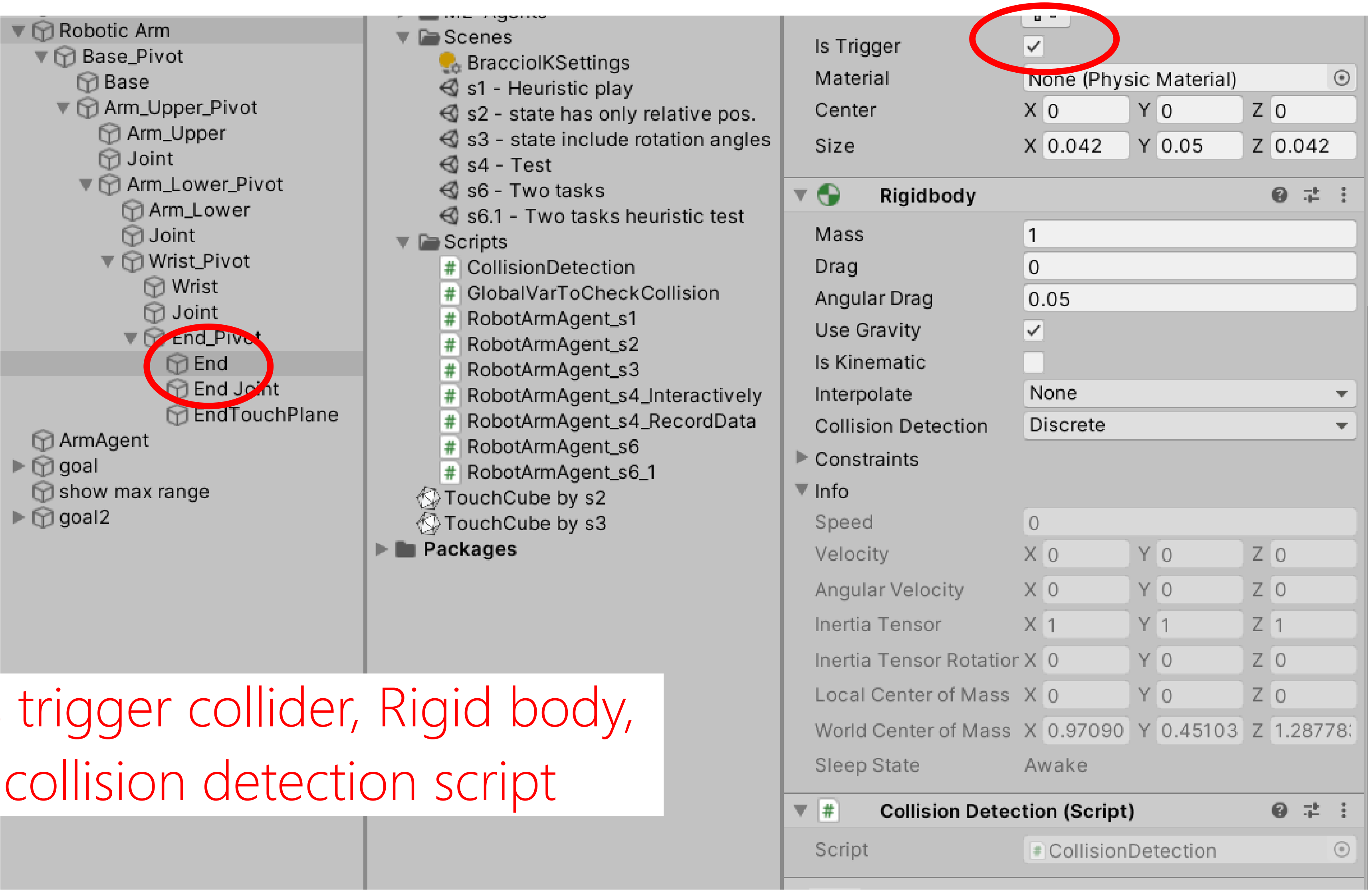
World Center of Mass X -0.6403 Y 0.38962 Z 1.10918

Sleep State Awake

Collision Detection (Script)

Script CollisionDetection

Add trigger collider, Rigid body, and collision detection script to Robot End



- Robotic Arm
  - Base_Pivot
    - Base
    - Arm_Upper_Pivot
      - Arm_Upper
      - Joint
      - Arm_Lower_Pivot
        - Arm_Lower
        - Joint
        - Wrist_Pivot
          - Wrist
          - Joint
          - End_Pivot
            - End
            - End Joint
            - EndTouchPlane
- ArmAgent
- goal
- show max range
- goal2

- Scenes
  - BraccioIKSettings
  - s1 - Heuristic play
  - s2 - state has only relative pos.
  - s3 - state include rotation angles
  - s4 - Test
  - s6 - Two tasks
  - s6.1 - Two tasks heuristic test
- Scripts
  - CollisionDetection
  - GlobalVarToCheckCollision
  - RobotArmAgent_s1
  - RobotArmAgent_s2
  - RobotArmAgent_s3
  - RobotArmAgent_s4_Interactively
  - RobotArmAgent_s4_RecordData
  - RobotArmAgent_s6
  - RobotArmAgent_s6_1
- TouchCube by s2
- TouchCube by s3
- **Packages**

Is Trigger ✓
Material None (Physic Material)
Center X 0 Y 0 Z 0
Size X 0.042 Y 0.05 Z 0.042

Rigidbody

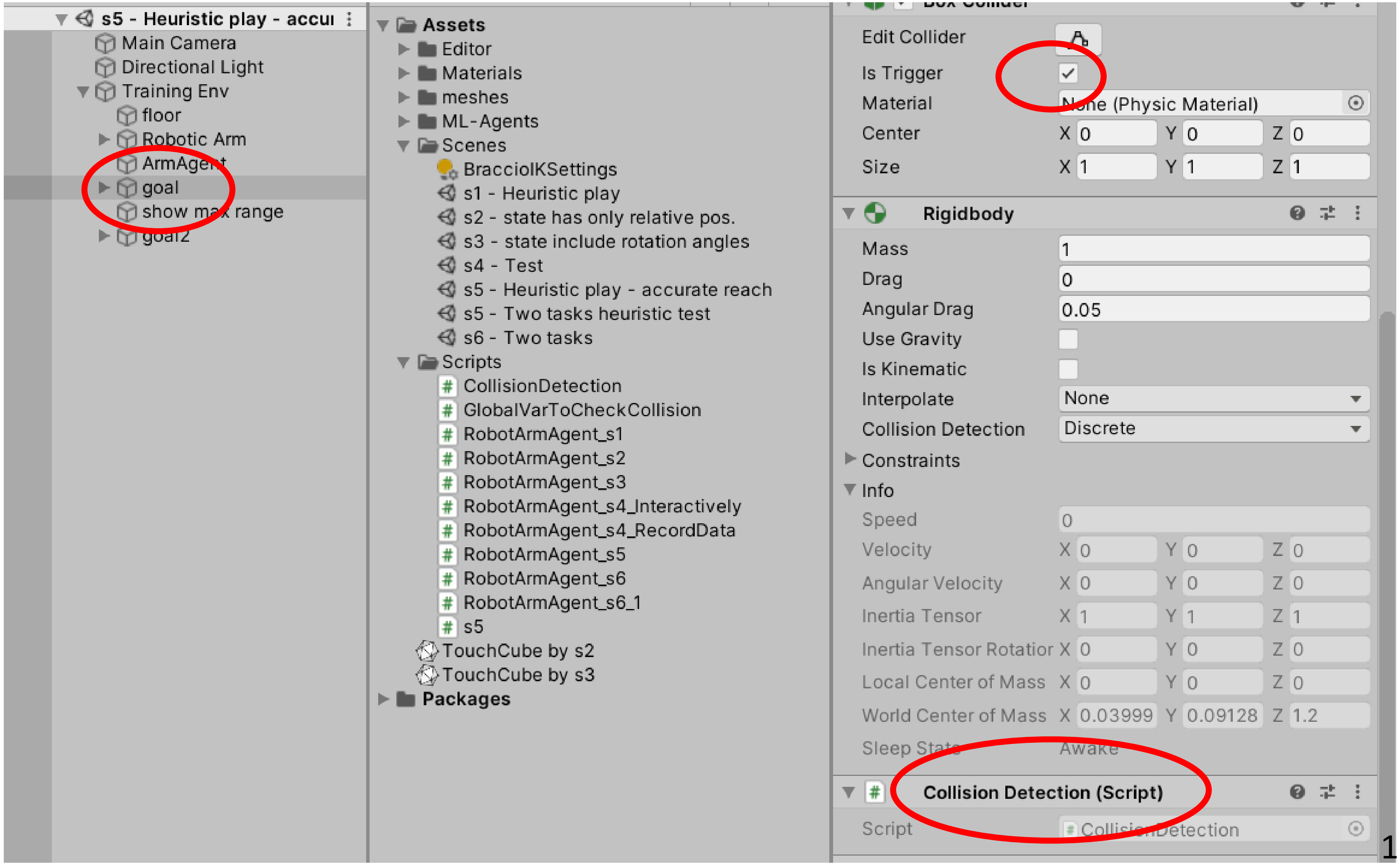
Mass 1
Drag 0
Angular Drag 0.05
Use Gravity ✓
Is Kinematic □
Interpolate None
Collision Detection Discrete
Constraints
Info
Speed 0
Velocity X 0 Y 0 Z 0
Angular Velocity X 0 Y 0 Z 0
Inertia Tensor X 1 Y 1 Z 1
Inertia Tensor Rotatior X 0 Y 0 Z 0
Local Center of Mass X 0 Y 0 Z 0
World Center of Mass X 0.97090 Y 0.45103 Z 1.28778:
Sleep State Awake

Collision Detection (Script)

Script CollisionDetection

End has trigger collider, Rigid body, and collision detection script

Add trigger collider, Rigid body, and collision detection script to goal object



- s5 - Heuristic play - accu
  - Main Camera
  - Directional Light
  - Training Env
    - floor
    - Robotic Arm
    - ArmAgent
    - goal
    - show max range
    - goal2

- Assets
  - Editor
  - Materials
  - meshes
  - ML-Agents
  - Scenes
    - BracciolKSettings
    - s1 - Heuristic play
    - s2 - state has only relative pos.
    - s3 - state include rotation angles
    - s4 - Test
    - s5 - Heuristic play - accurate reach
    - s5 - Two tasks heuristic test
    - s6 - Two tasks
  - Scripts
    - CollisionDetection
    - GlobalVarToCheckCollision
    - RobotArmAgent_s1
    - RobotArmAgent_s2
    - RobotArmAgent_s3
    - RobotArmAgent_s4_Interactively
    - RobotArmAgent_s4_RecordData
    - RobotArmAgent_s5
    - RobotArmAgent_s6
    - RobotArmAgent_s6_1
    - s5
  - TouchCube by s2
  - TouchCube by s3
- Packages

Box Collider

Edit Collider	
Is Trigger	☑
Material	None (Physic Material)
Center	X 0 Y 0 Z 0
Size	X 1 Y 1 Z 1

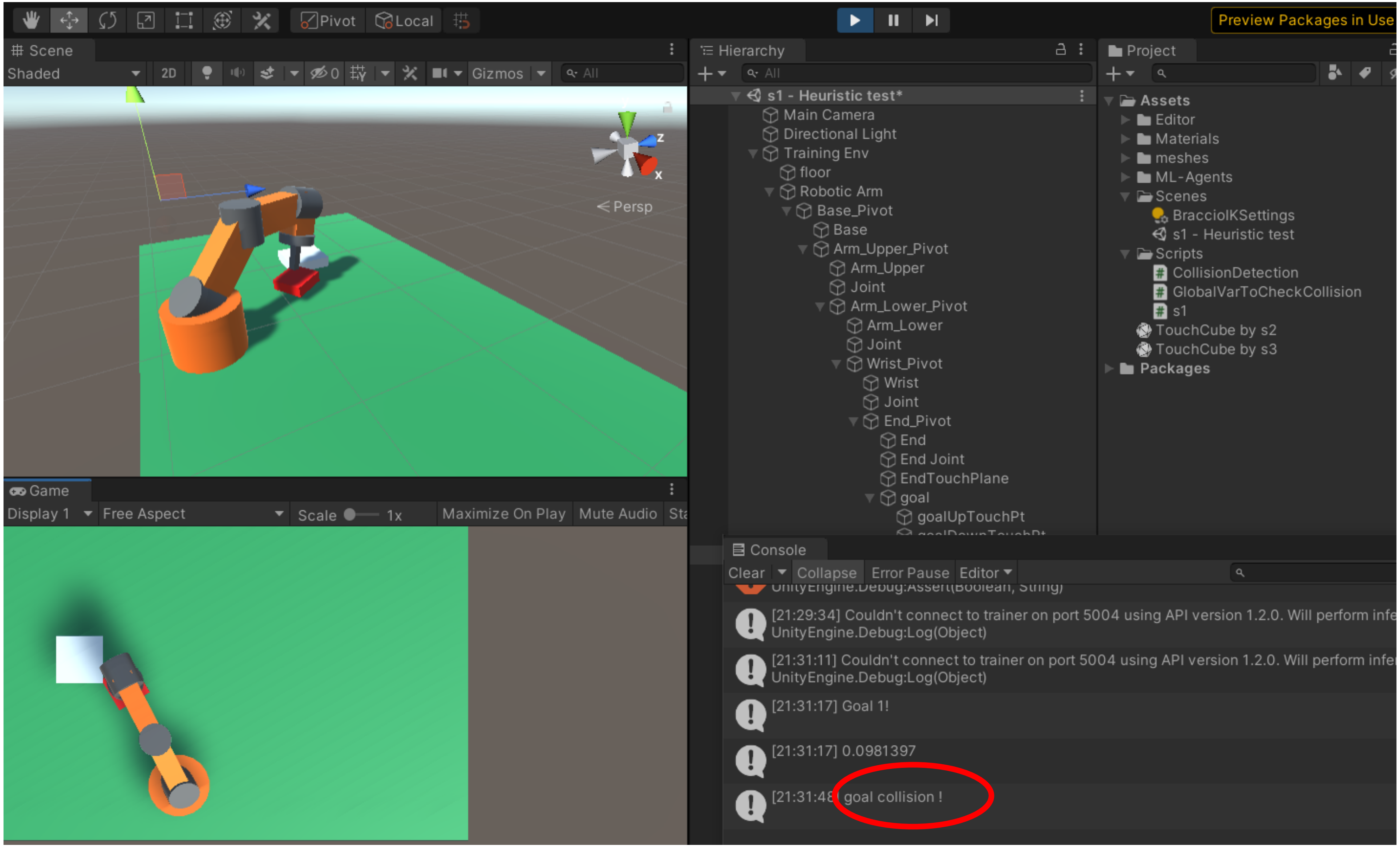
Rigidbody

Mass	1
Drag	0
Angular Drag	0.05
Use Gravity	□
Is Kinematic	□
Interpolate	None
Collision Detection	Discrete
Constraints	
Info	
Speed	0
Velocity	X 0 Y 0 Z 0
Angular Velocity	X 0 Y 0 Z 0
Inertia Tensor	X 1 Y 1 Z 1
Inertia Tensor Rotatior	X 0 Y 0 Z 0
Local Center of Mass	X 0 Y 0 Z 0
World Center of Mass	X 0.03999 Y 0.09128 Z 1.2
Sleep State	Awake

Collision Detection (Script)

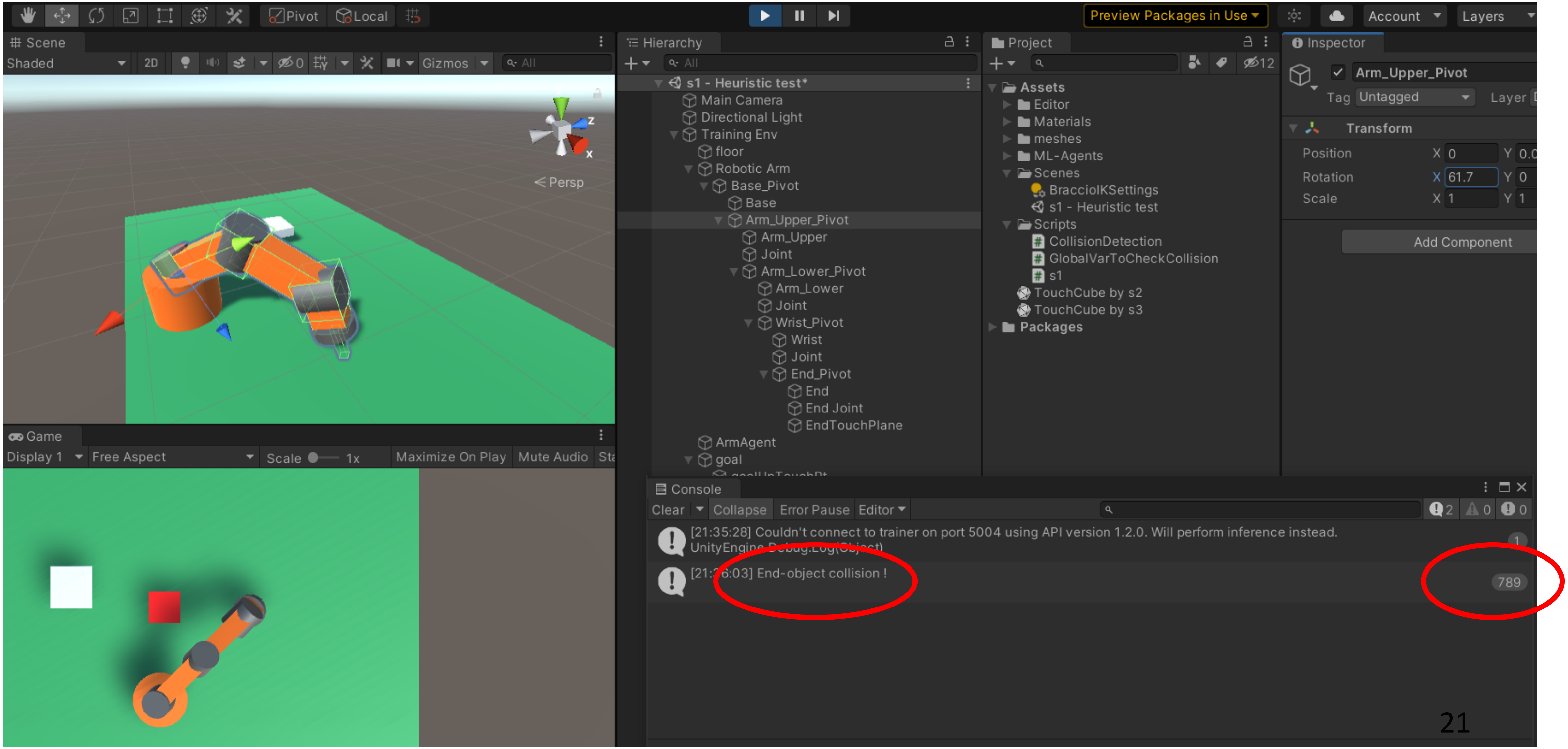
Script	CollisionDetection

Control the robot arm to collide with the floor or goal and check the error message



```
UnityEngine.Debug:Assert(Boolean, String)
[21:29:34] Couldn't connect to trainer on port 5004 using API version 1.2.0. Will perform infe
UnityEngine.Debug:Log(Object)
[21:31:11] Couldn't connect to trainer on port 5004 using API version 1.2.0. Will perform infe
UnityEngine.Debug:Log(Object)
[21:31:17] Goal 1!
[21:31:17] 0.0981397
[21:31:48] goal collision !
```

Control the robot arm to collide with the floor or goal and check the error message



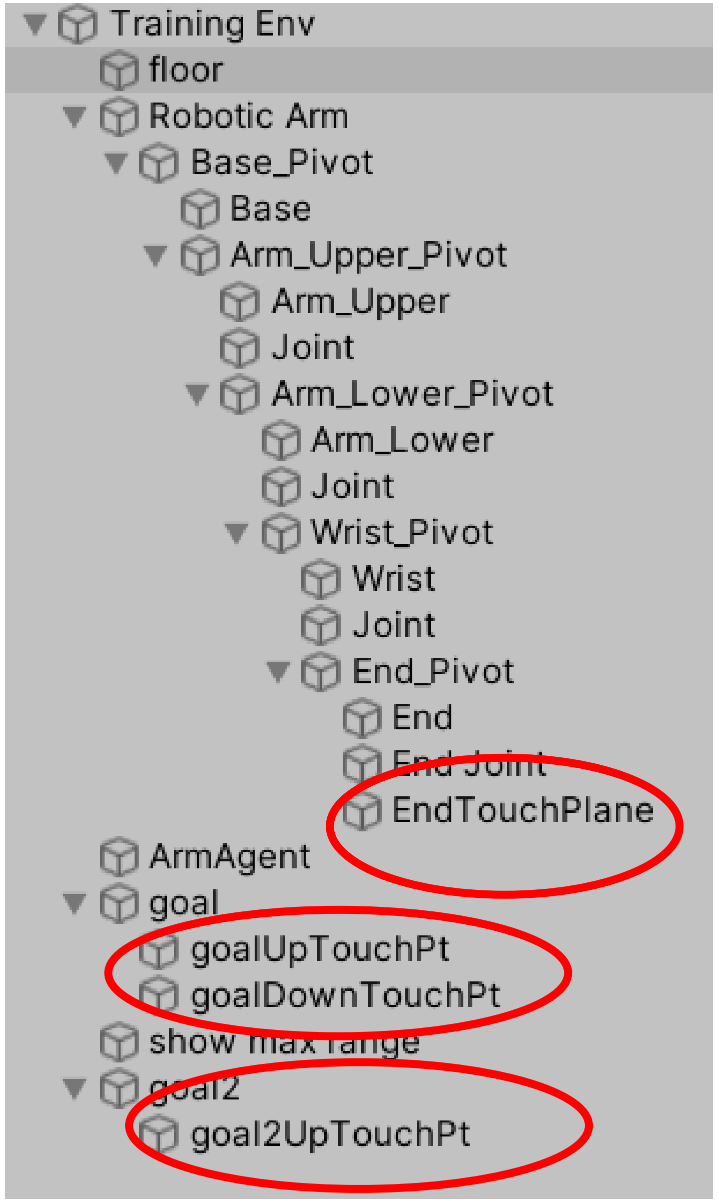
Arm_Upper_Pivot
Tag Untagged

Transform
Position X 0 Y 0.0
Rotation X 61.7 Y 0
Scale X 1 Y 1

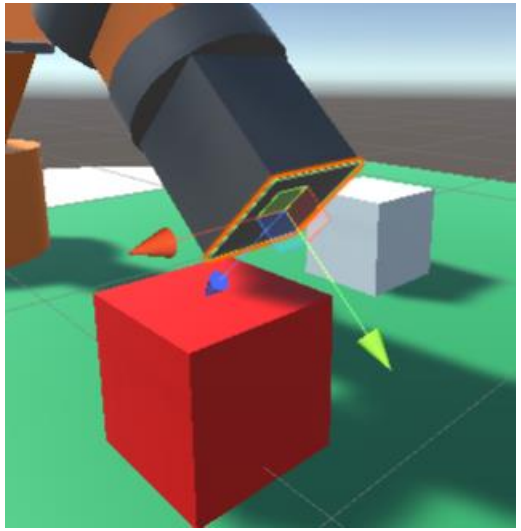
```
[21:35:28] Couldn't connect to trainer on port 5004 using API version 1.2.0. Will perform inference instead.
UnityEngine.Debug:Log(Object)                                                                                  1
[21:36:03] End-object collision !                                                                            789
```

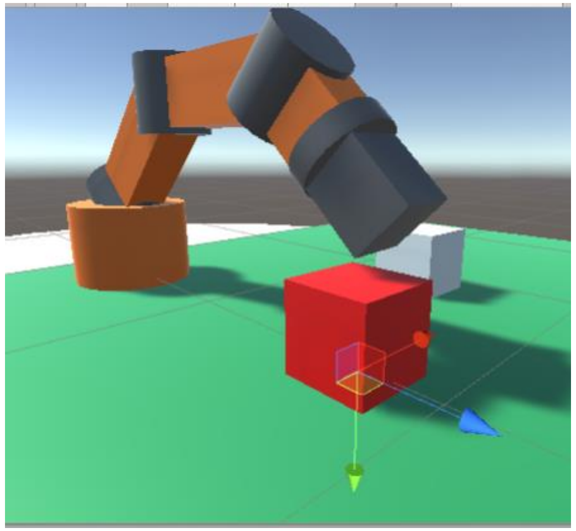
Determine whether the robot end reaches the goal or not

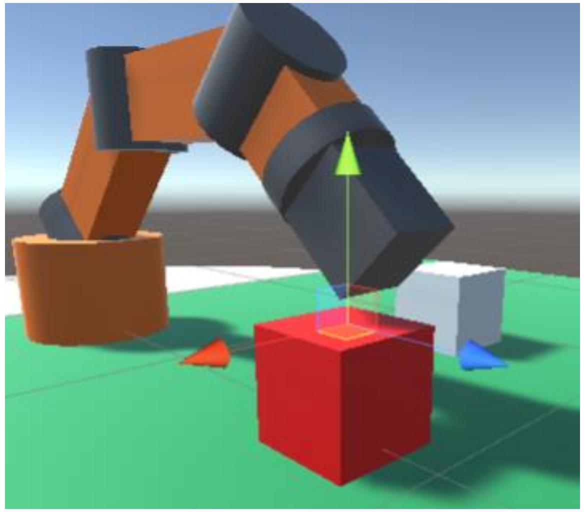
Points for goal-reaching detection

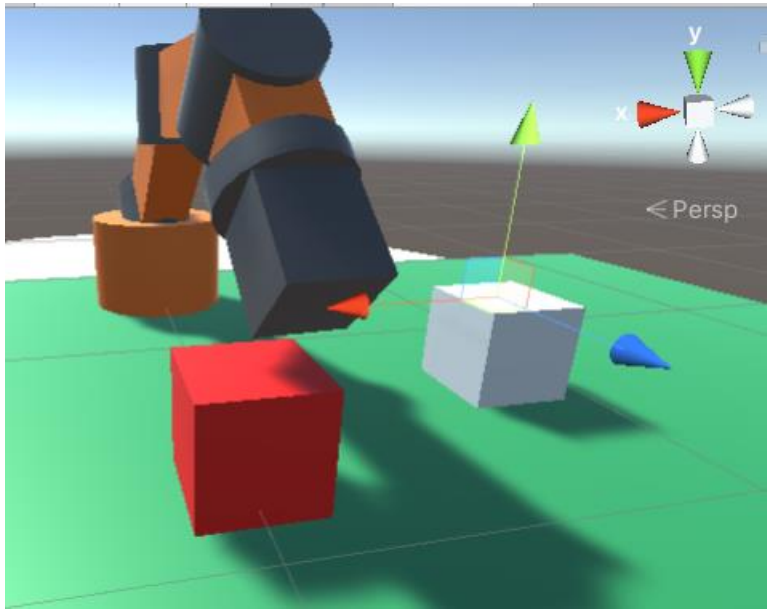


- Training Env
  - floor
  - Robotic Arm
    - Base_Pivot
      - Base
      - Arm_Upper_Pivot
        - Arm_Upper
        - Joint
        - Arm_Lower_Pivot
          - Arm_Lower
          - Joint
          - Wrist_Pivot
            - Wrist
            - Joint
            - End_Pivot
              - End
              - End Joint
              - EndTouchPlane
  - ArmAgent
  - goal
    - goalUpTouchPt
    - goalDownTouchPt
  - show max range
  - goal2
    - goal2UpTouchPt

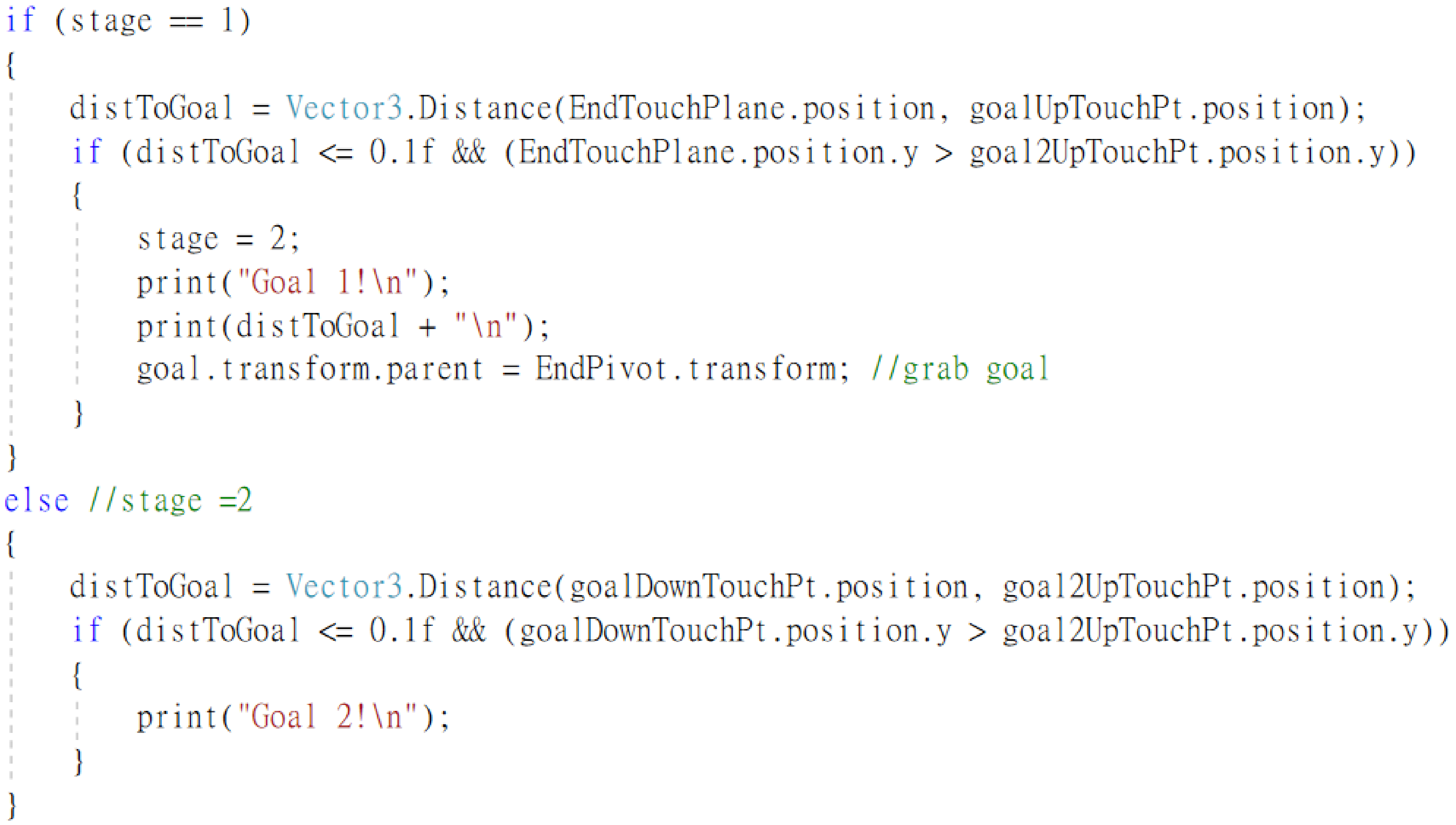






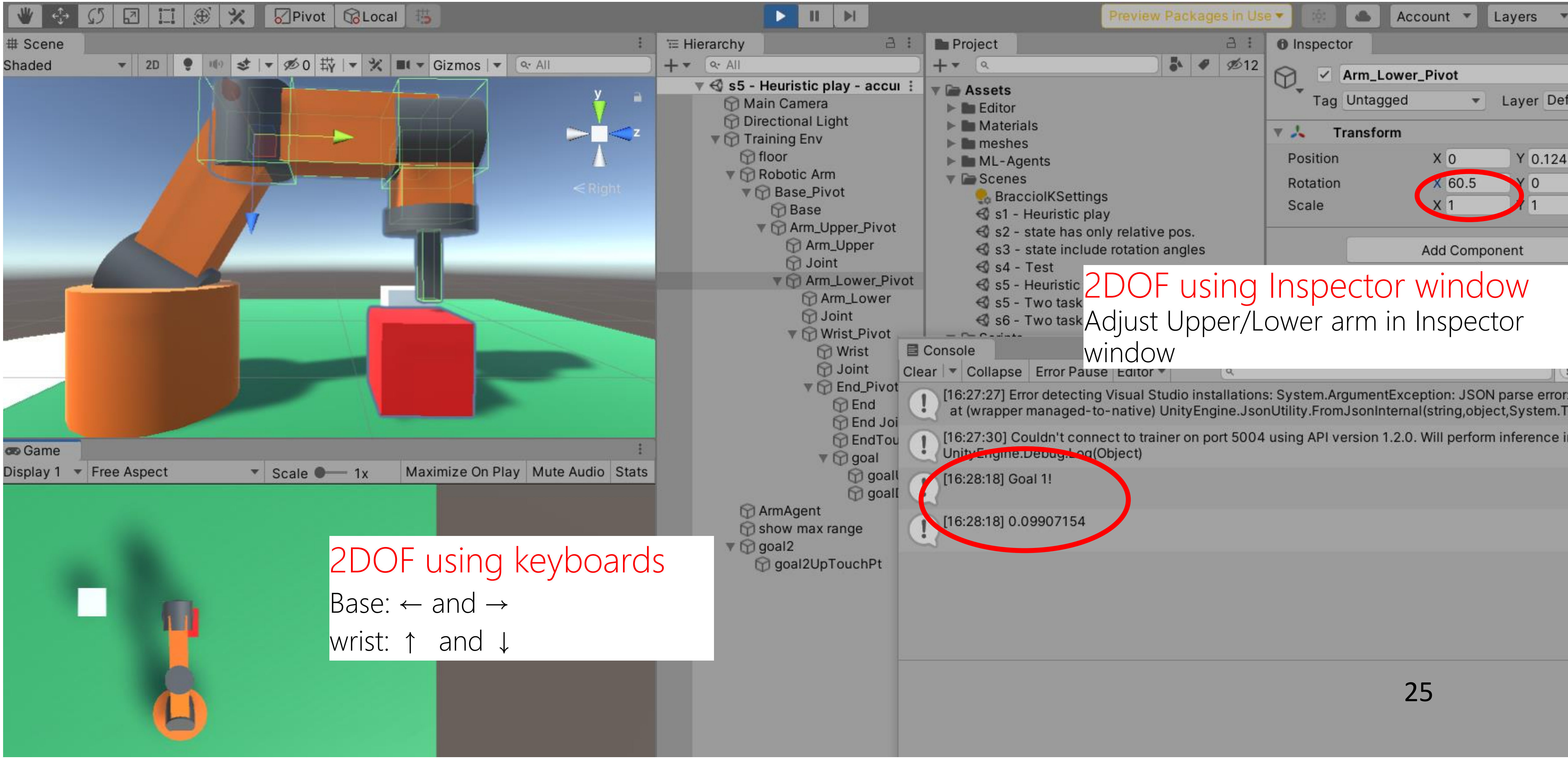


Use distance and y-value to determine goal reach



```
if (stage == 1)
{
    distToGoal = Vector3.Distance(EndTouchPlane.position, goalUpTouchPt.position);
    if (distToGoal <= 0.1f && (EndTouchPlane.position.y > goal2UpTouchPt.position.y))
    {
        stage = 2;
        print("Goal 1!\n");
        print(distToGoal + "\n");
        goal.transform.parent = EndPivot.transform; //grab goal
    }
}
else //stage =2
{
    distToGoal = Vector3.Distance(goalDownTouchPt.position, goal2UpTouchPt.position);
    if (distToGoal <= 0.1f && (goalDownTouchPt.position.y > goal2UpTouchPt.position.y))
    {
        print("Goal 2!\n");
    }
}
```

Manually control robot arm to reach goal (avoid collision!)



Position X 0 Y 0.124

Rotation X 60.5 Y 0

Scale X 1 Y 1

```
[16:27:27] Error detecting Visual Studio installations: System.ArgumentException: JSON parse error
 at (wrapper managed-to-native) UnityEngine.JsonUtility.FromJsonInternal(string,object,System.T
[16:27:30] Couldn't connect to trainer on port 5004 using API version 1.2.0. Will perform inference i
UnityEngine.Debug:Log(Object)
[16:28:18] Goal 1!
[16:28:18] 0.09907154
```

2DOF using Inspector window

Adjust Upper/Lower arm in Inspector window

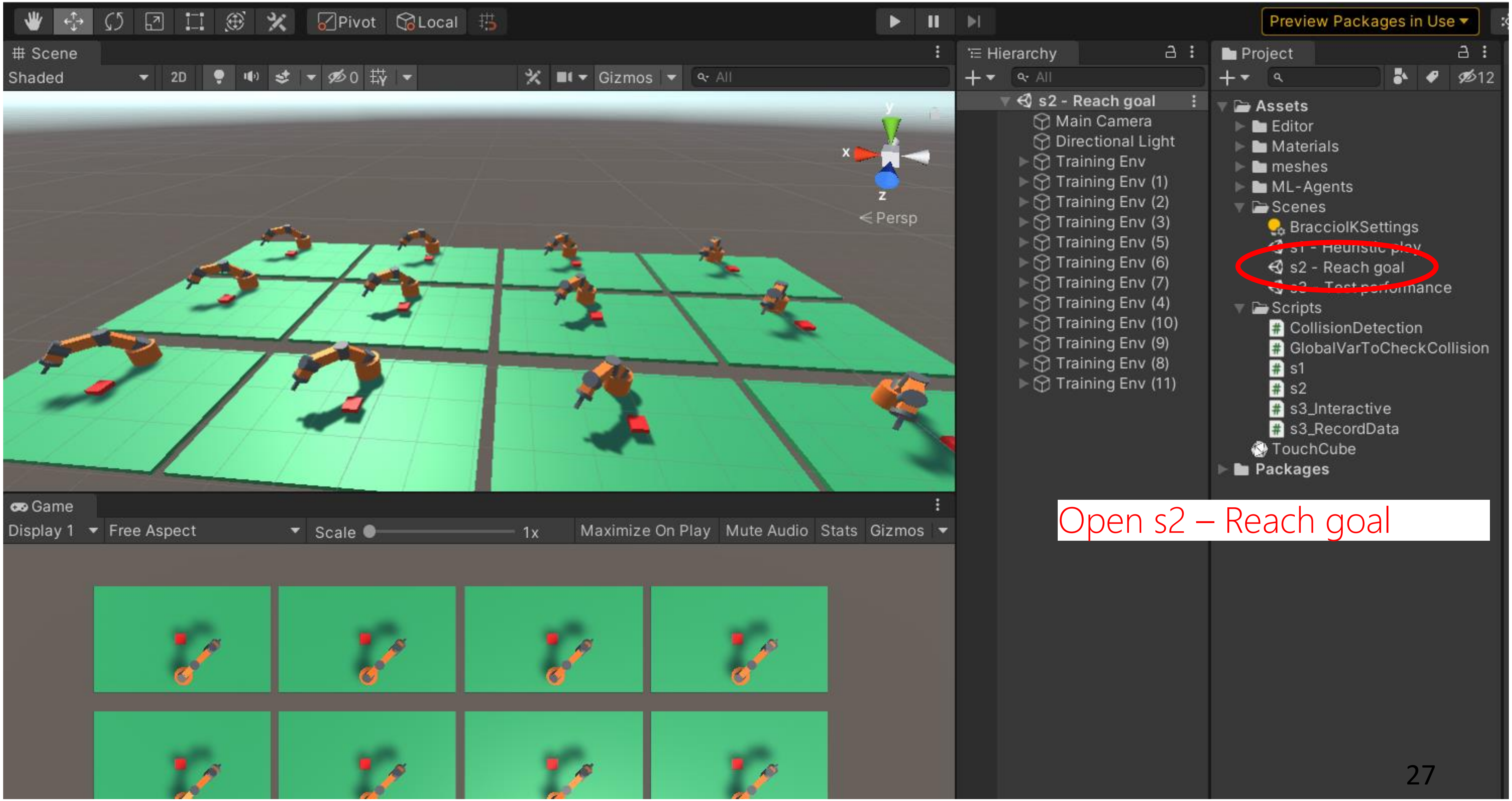
2DOF using keyboards

Base: ← and →

wrist: ↑ and ↓

Train agent to control the robot arm to reach goal

4. Open the training environment



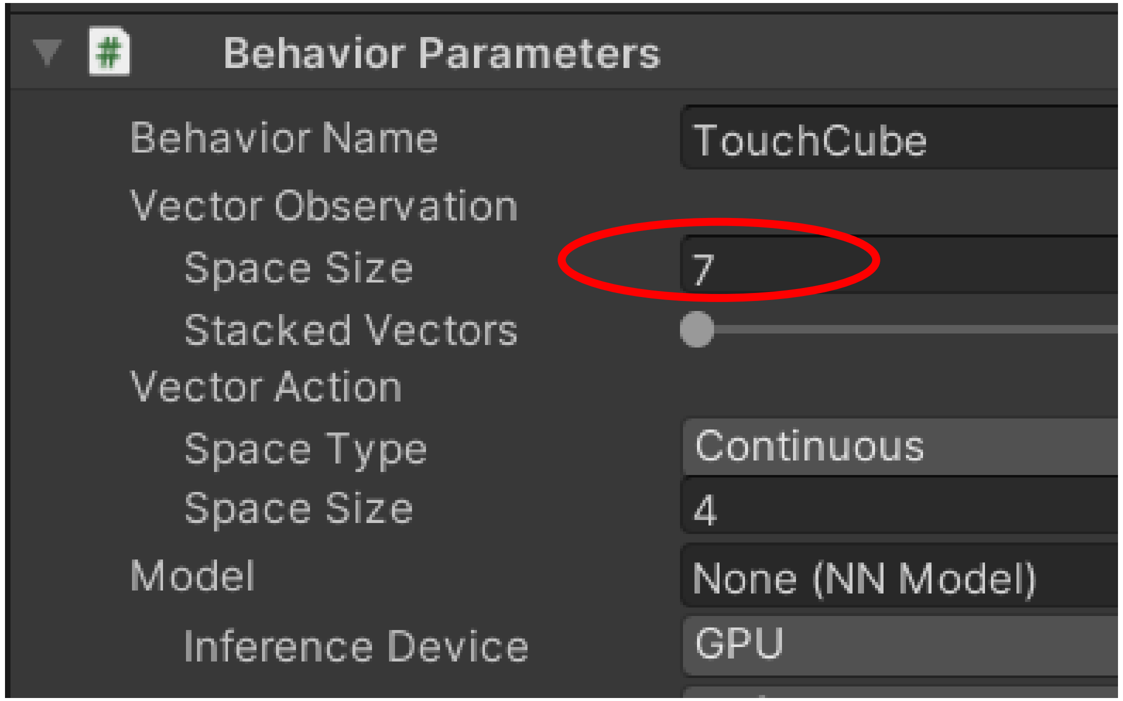
Open s2 – Reach goal

State has 7 variables

```
sensor.AddObservation(EndTouchPlane.position - goalUpTouchPt.transform.position);

float BaseRotationAngle = UnityEditor.TransformUtils.GetInspectorRotation(BasePivot).y;
float UArmRotationAngle = UnityEditor.TransformUtils.GetInspectorRotation(UpperPivot).x;
float LArmRotationAngle = UnityEditor.TransformUtils.GetInspectorRotation(LowerPivot).x;
float WRotationAngle = UnityEditor.TransformUtils.GetInspectorRotation(WristPivot).x;

sensor.AddObservation(BaseRotationAngle);
sensor.AddObservation(UArmRotationAngle);
sensor.AddObservation(LArmRotationAngle);
sensor.AddObservation(WRotationAngle);
```



Behavior Parameters

Behavior Name: TouchCube

Vector Observation

Space Size: 7

Stacked Vectors: ●

Vector Action

Space Type: Continuous

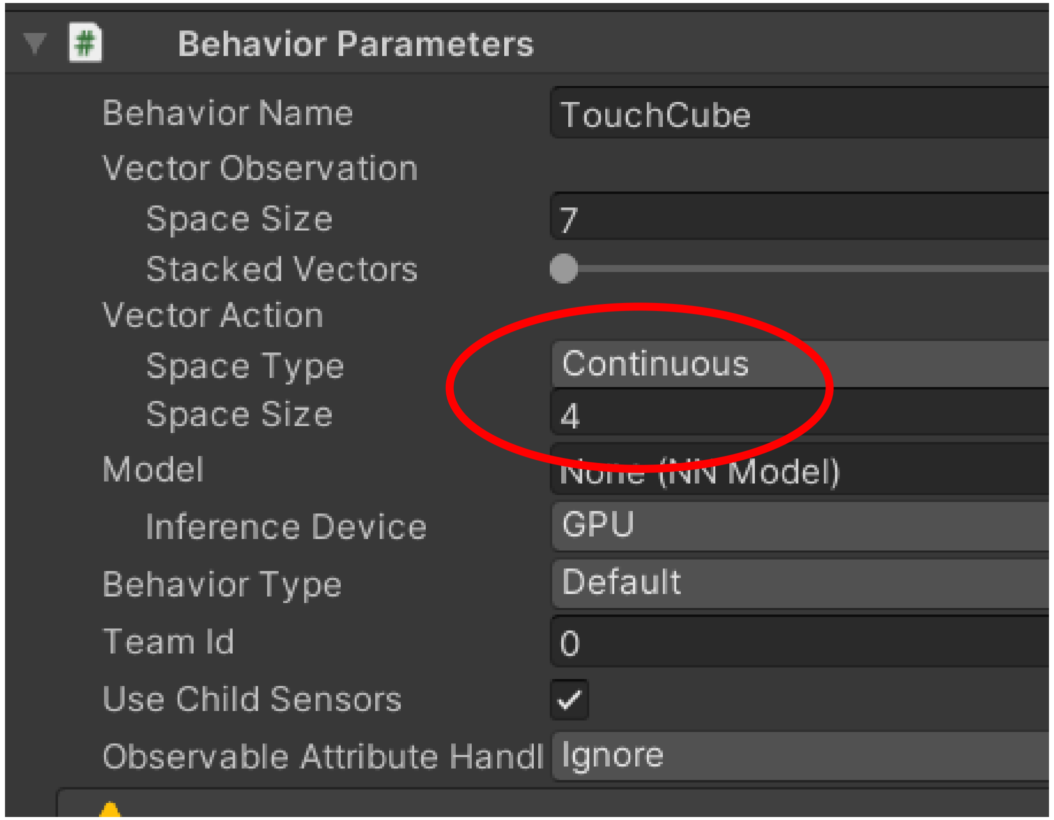
Space Size: 4

Model: None (NN Model)

Inference Device: GPU

Action has 4 values

```
BasePivot.Rotate(0, vectorAction[0] * speed, 0);
UpperPivot.Rotate(vectorAction[1] * speed, 0, 0);
LowerPivot.Rotate(vectorAction[2] * speed, 0, 0);
WristPivot.Rotate(vectorAction[3] * speed, 0, 0);
```



Behavior Parameters

Behavior Name: TouchCube

Vector Observation

Space Size: 7

Stacked Vectors

Vector Action

Space Type: Continuous

Space Size: 4

Model: None (NN Model)

Inference Device: GPU

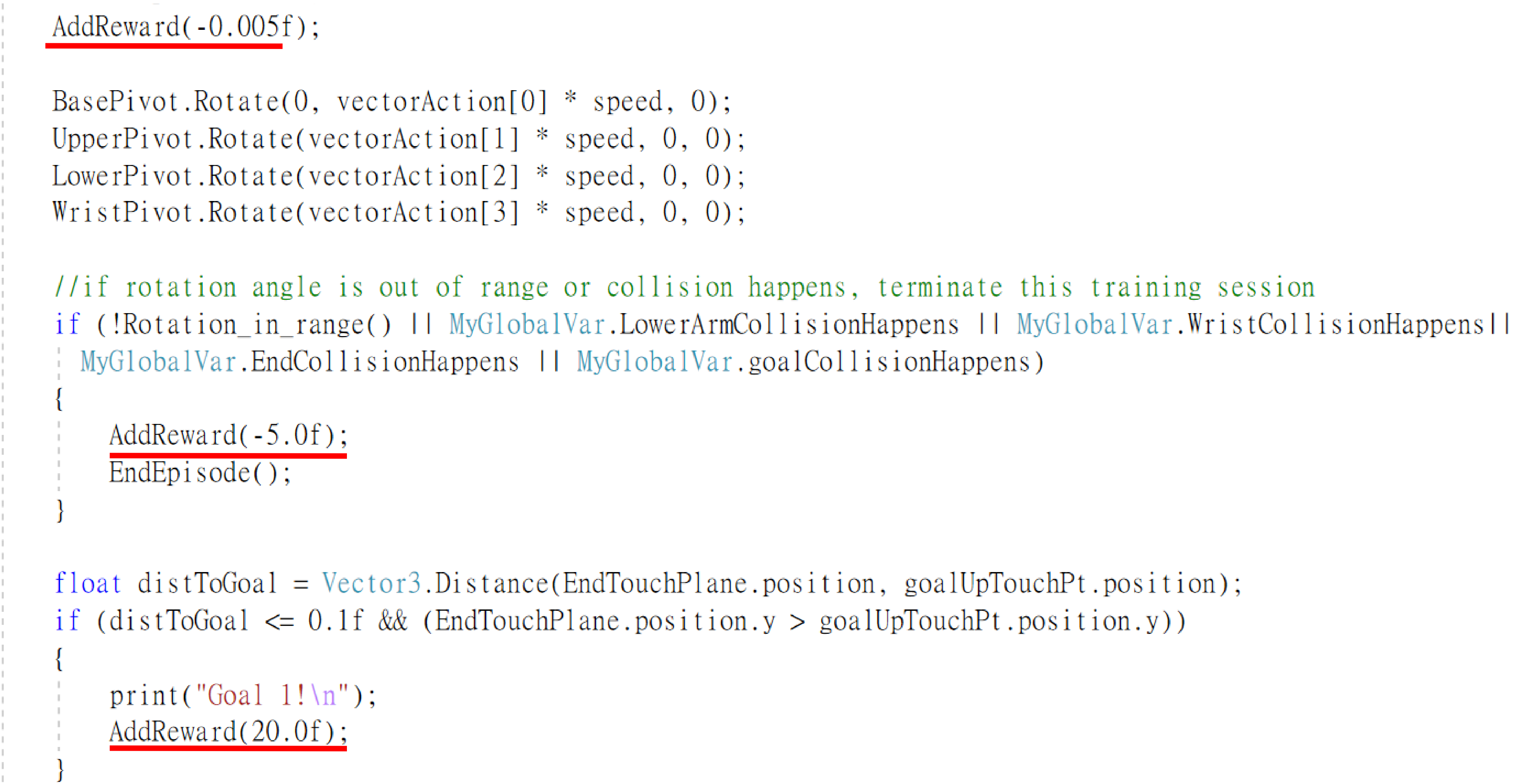
Behavior Type: Default

Team Id: 0

Use Child Sensors: ✓

Observable Attribute Handl: Ignore

3 types of rewards



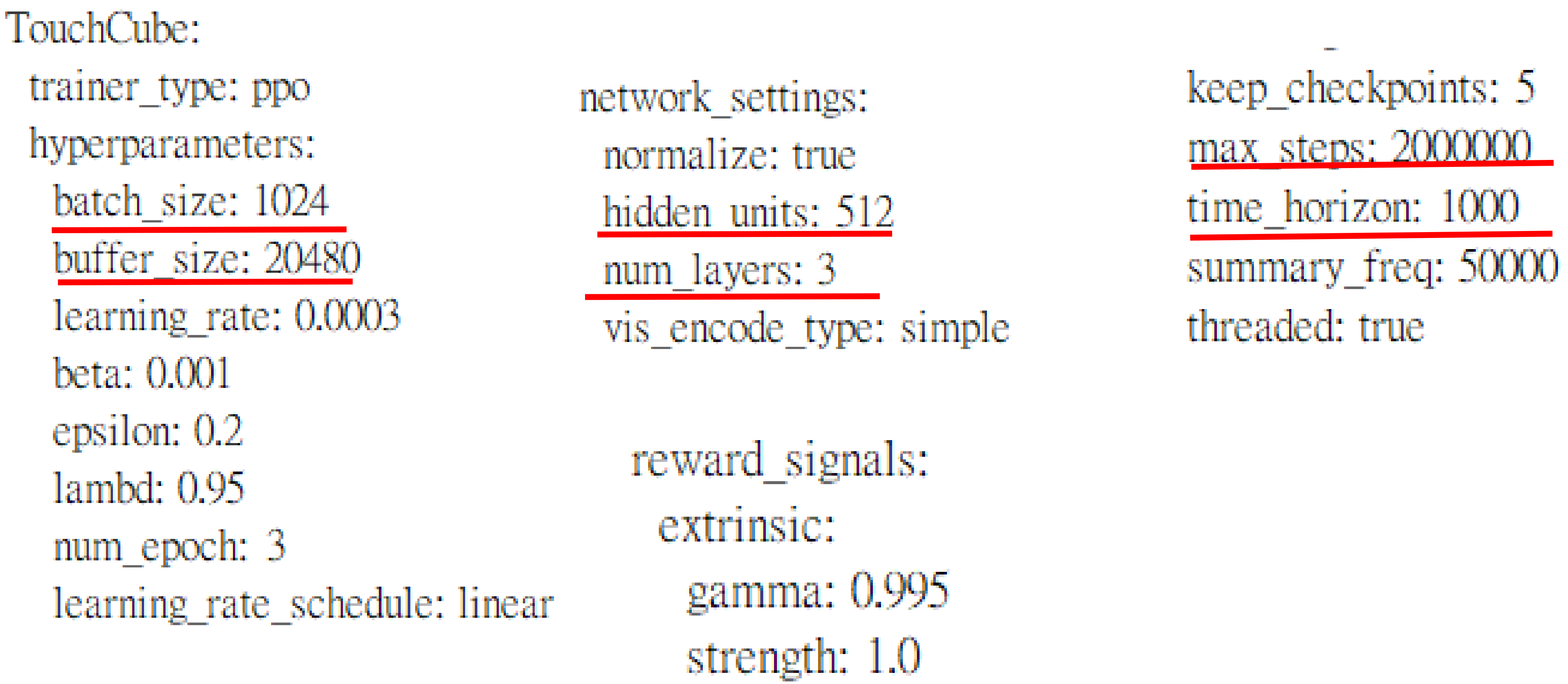
```
AddReward(-0.005f);

BasePivot.Rotate(0, vectorAction[0] * speed, 0);
UpperPivot.Rotate(vectorAction[1] * speed, 0, 0);
LowerPivot.Rotate(vectorAction[2] * speed, 0, 0);
WristPivot.Rotate(vectorAction[3] * speed, 0, 0);

//if rotation angle is out of range or collision happens, terminate this training session
if (!Rotation_in_range() || MyGlobalVar.LowerArmCollisionHappens || MyGlobalVar.WristCollisionHappens||
 MyGlobalVar.EndCollisionHappens || MyGlobalVar.goalCollisionHappens)
{
    AddReward(-5.0f);
    EndEpisode();
}

float distToGoal = Vector3.Distance(EndTouchPlane.position, goalUpTouchPt.position);
if (distToGoal <= 0.1f && (EndTouchPlane.position.y > goalUpTouchPt.position.y))
{
    print("Goal 1!\n");
    AddReward(20.0f);
}
```

Training configuration file



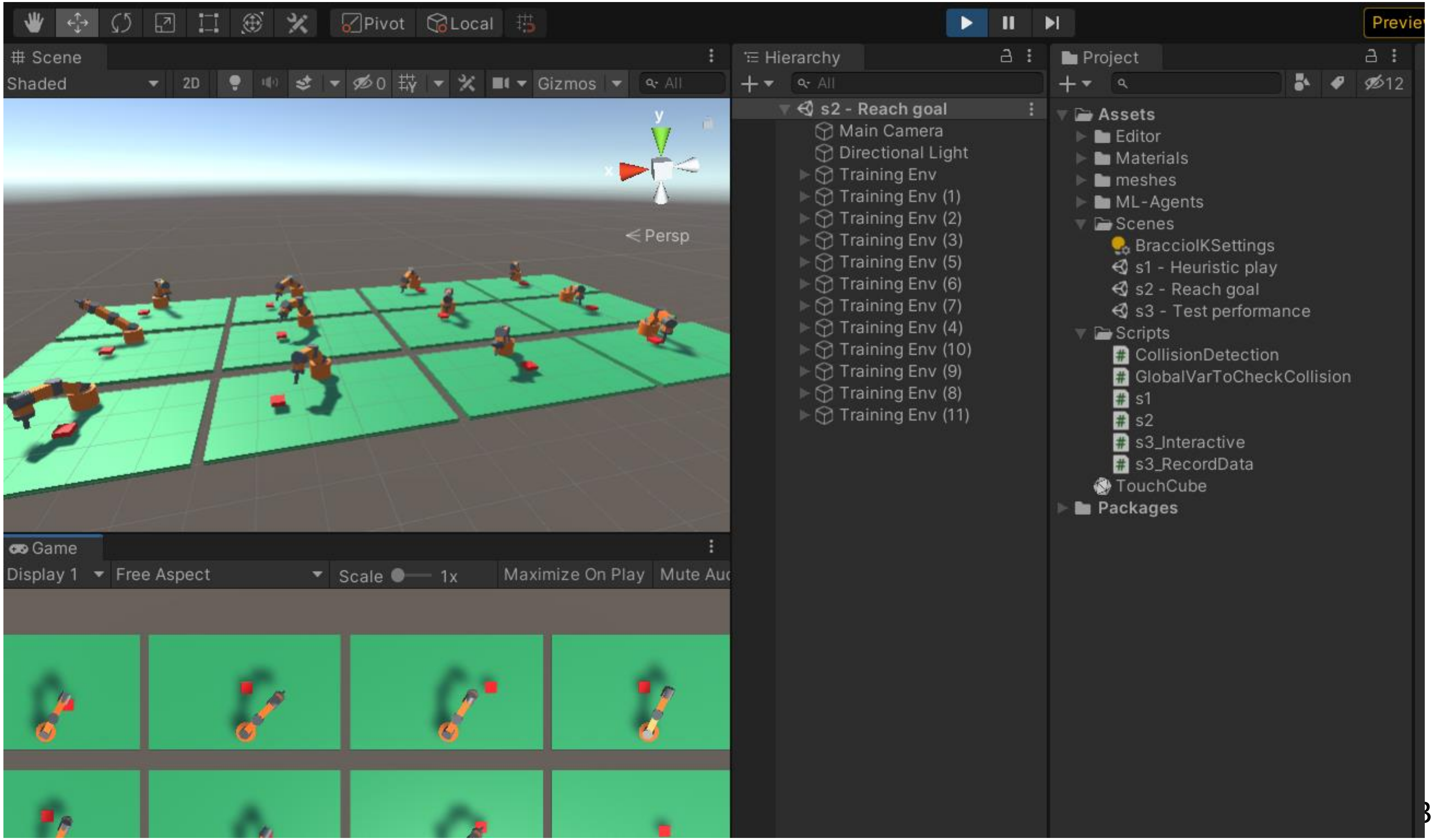
```
TouchCube:
  trainer_type: ppo
  hyperparameters:
    batch_size: 1024
    buffer_size: 20480
    learning_rate: 0.0003
    beta: 0.001
    epsilon: 0.2
    lambd: 0.95
    num_epoch: 3
    learning_rate_schedule: linear
```

```
network_settings:
  normalize: true
  hidden_units: 512
  num_layers: 3
  vis_encode_type: simple

  reward_signals:
    extrinsic:
      gamma: 0.995
      strength: 1.0
```

```
keep_checkpoints: 5
max_steps: 2000000
time_horizon: 1000
summary_freq: 50000
threaded: true
```

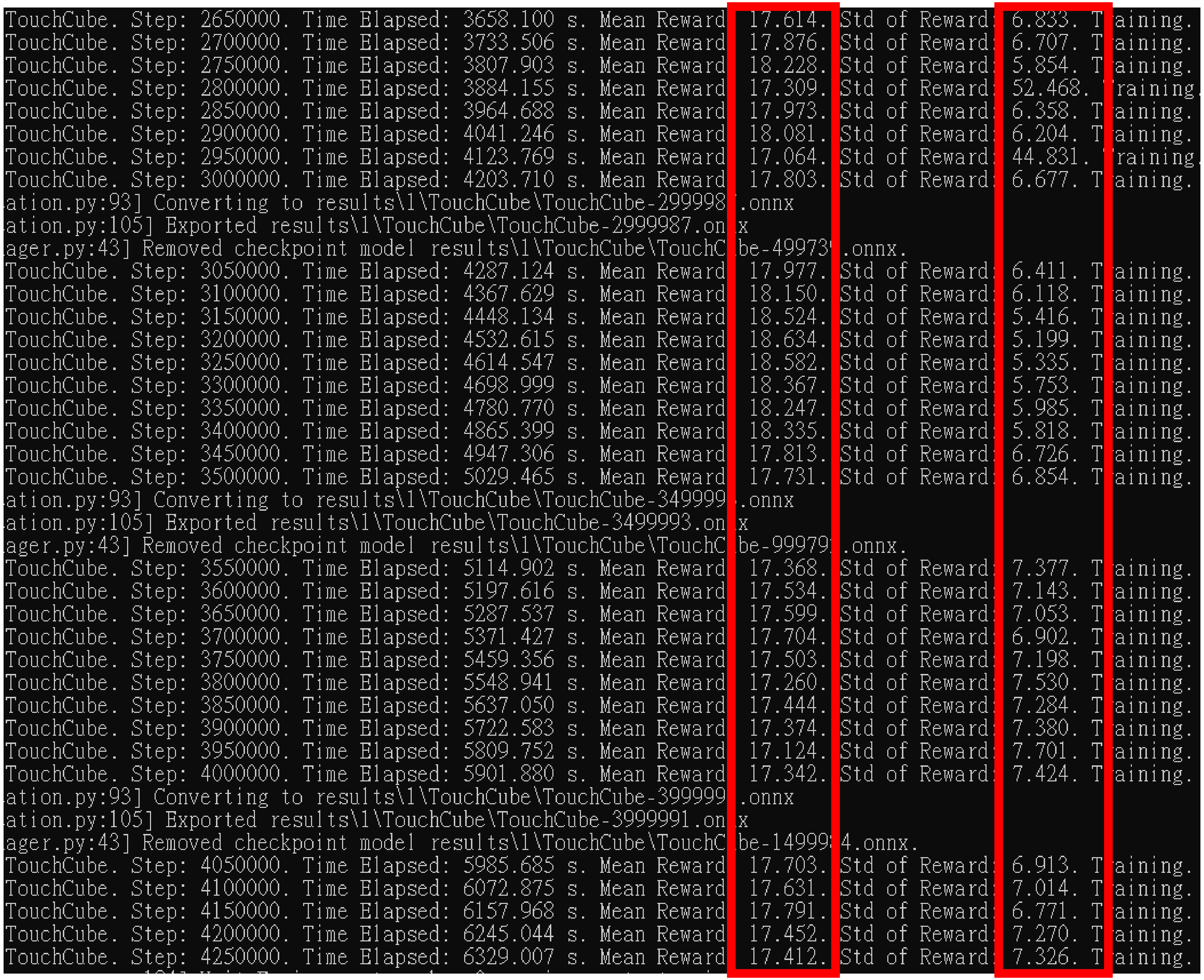
Train and watch performance



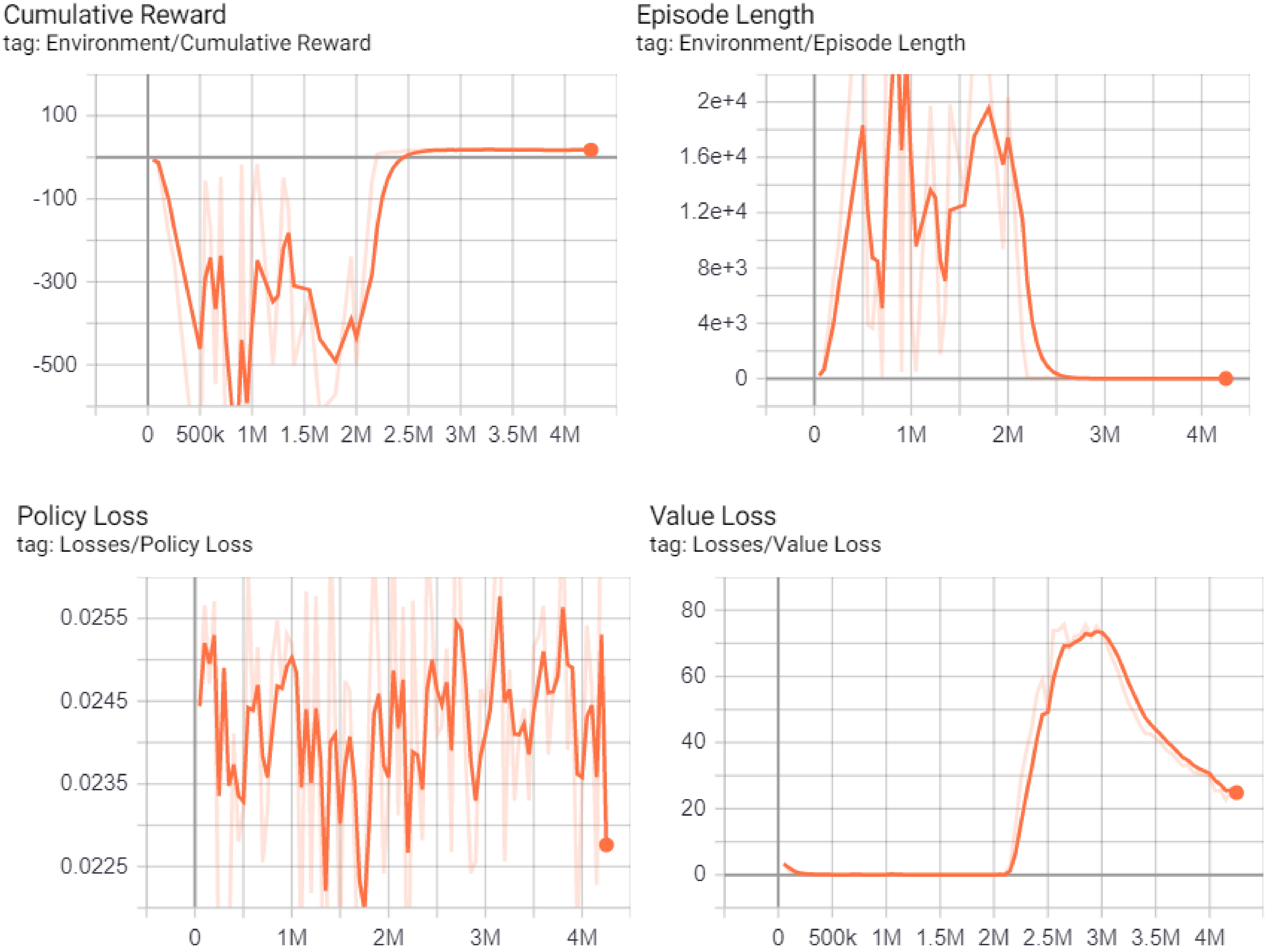
- s2 - Reach goal
  - Main Camera
  - Directional Light
  - Training Env
  - Training Env (1)
  - Training Env (2)
  - Training Env (3)
  - Training Env (5)
  - Training Env (6)
  - Training Env (7)
  - Training Env (4)
  - Training Env (10)
  - Training Env (9)
  - Training Env (8)
  - Training Env (11)

Project

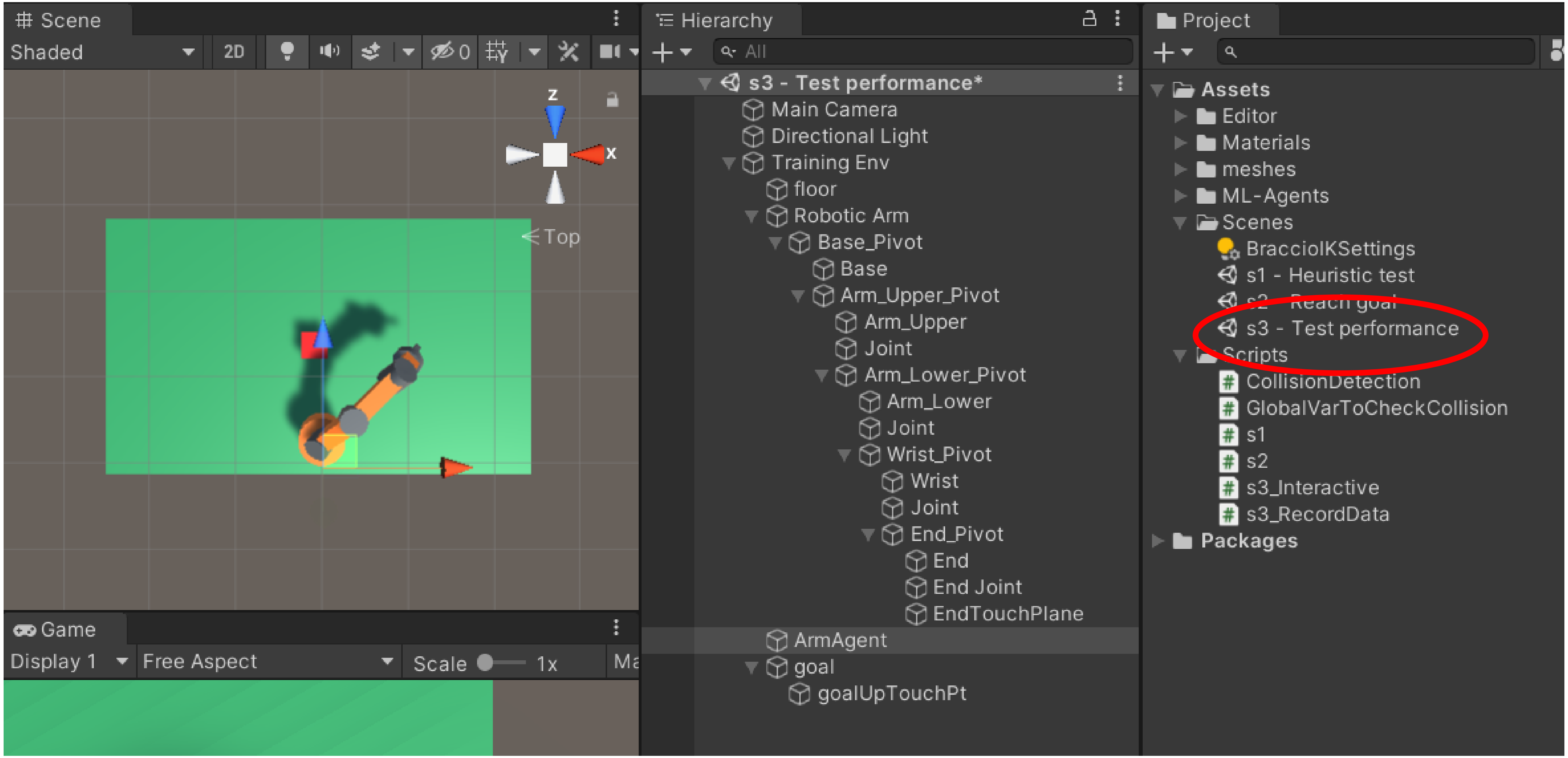
- Assets
  - Editor
  - Materials
  - meshes
  - ML-Agents
  - Scenes
    - BraccioIKSettings
    - s1 - Heuristic play
    - s2 - Reach goal
    - s3 - Test performance
  - Scripts
    - CollisionDetection
    - GlobalVarToCheckCollision
    - s1
    - s2
    - s3_Interactive
    - s3_RecordData
  - TouchCube
- Packages



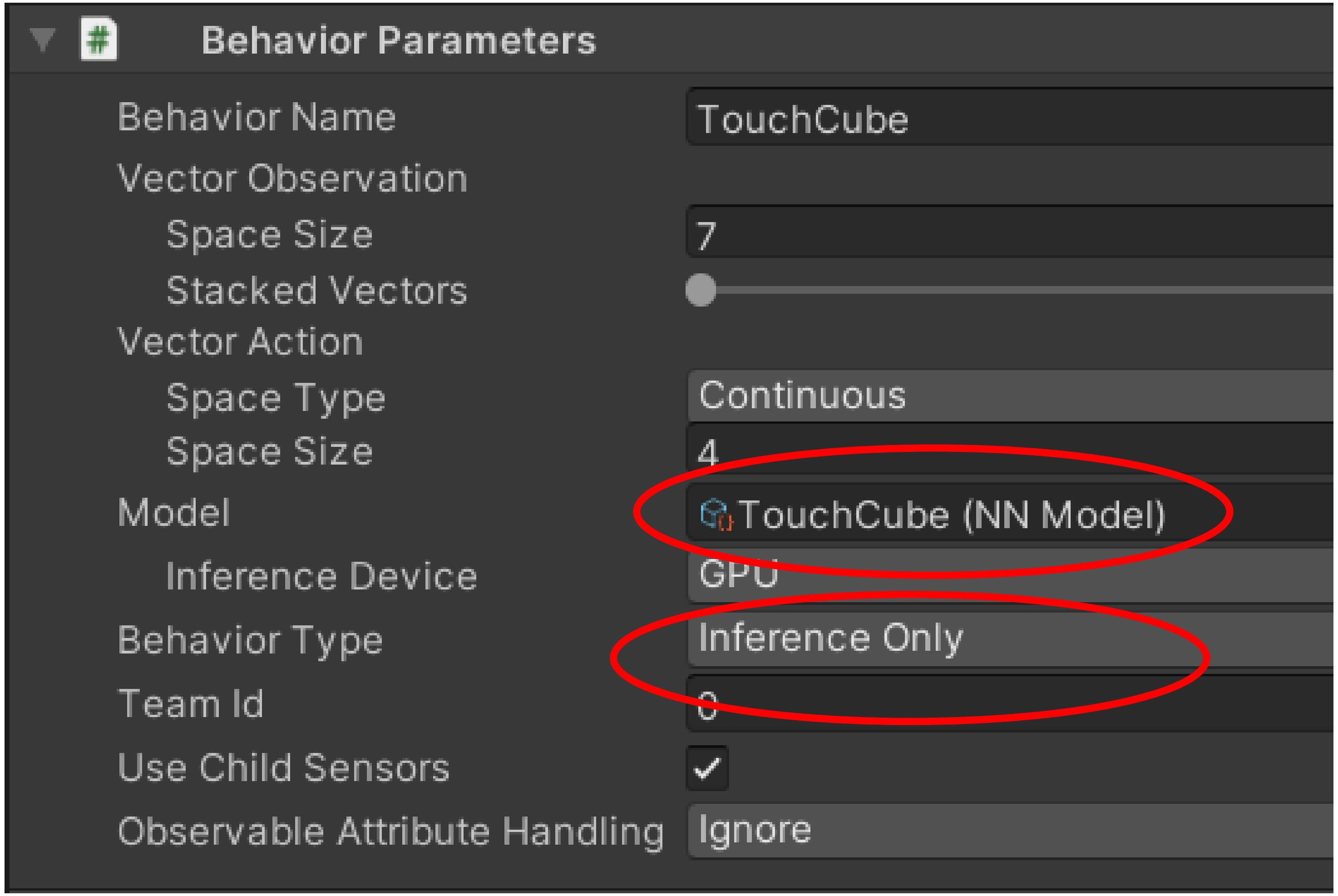
```
TouchCube. Step: 2650000. Time Elapsed: 3658.100 s. Mean Reward  17.614. Std of Reward  6.833. T aining.
TouchCube. Step: 2700000. Time Elapsed: 3733.506 s. Mean Reward  17.876. Std of Reward  6.707. T aining.
TouchCube. Step: 2750000. Time Elapsed: 3807.903 s. Mean Reward  18.228. Std of Reward  5.854. T aining.
TouchCube. Step: 2800000. Time Elapsed: 3884.155 s. Mean Reward  17.309. Std of Reward  52.468.  raining.
TouchCube. Step: 2850000. Time Elapsed: 3964.688 s. Mean Reward  17.973. Std of Reward  6.358. T aining.
TouchCube. Step: 2900000. Time Elapsed: 4041.246 s. Mean Reward  18.081. Std of Reward  6.204. T aining.
TouchCube. Step: 2950000. Time Elapsed: 4123.769 s. Mean Reward  17.064. Std of Reward  44.831.  raining.
TouchCube. Step: 3000000. Time Elapsed: 4203.710 s. Mean Reward  17.803. Std of Reward  6.677. T aining.
ation.py:93] Converting to results\1\TouchCube\TouchCube-299998 .onnx
ation.py:105] Exported results\1\TouchCube\TouchCube-2999987.on x
ager.py:43] Removed checkpoint model results\1\TouchCube\TouchC be-49973 .onnx.
TouchCube. Step: 3050000. Time Elapsed: 4287.124 s. Mean Reward  17.977. Std of Reward  6.411. T aining.
TouchCube. Step: 3100000. Time Elapsed: 4367.629 s. Mean Reward  18.150. Std of Reward  6.118. T aining.
TouchCube. Step: 3150000. Time Elapsed: 4448.134 s. Mean Reward  18.524. Std of Reward  5.416. T aining.
TouchCube. Step: 3200000. Time Elapsed: 4532.615 s. Mean Reward  18.634. Std of Reward  5.199. T aining.
TouchCube. Step: 3250000. Time Elapsed: 4614.547 s. Mean Reward  18.582. Std of Reward  5.335. T aining.
TouchCube. Step: 3300000. Time Elapsed: 4698.999 s. Mean Reward  18.367. Std of Reward  5.753. T aining.
TouchCube. Step: 3350000. Time Elapsed: 4780.770 s. Mean Reward  18.247. Std of Reward  5.985. T aining.
TouchCube. Step: 3400000. Time Elapsed: 4865.399 s. Mean Reward  18.335. Std of Reward  5.818. T aining.
TouchCube. Step: 3450000. Time Elapsed: 4947.306 s. Mean Reward  17.813. Std of Reward  6.726. T aining.
TouchCube. Step: 3500000. Time Elapsed: 5029.465 s. Mean Reward  17.731. Std of Reward  6.854. T aining.
ation.py:93] Converting to results\1\TouchCube\TouchCube-349999 .onnx
ation.py:105] Exported results\1\TouchCube\TouchCube-3499993.on x
ager.py:43] Removed checkpoint model results\1\TouchCube\TouchC be-99979 .onnx.
TouchCube. Step: 3550000. Time Elapsed: 5114.902 s. Mean Reward  17.368. Std of Reward  7.377. T aining.
TouchCube. Step: 3600000. Time Elapsed: 5197.616 s. Mean Reward  17.534. Std of Reward  7.143. T aining.
TouchCube. Step: 3650000. Time Elapsed: 5287.537 s. Mean Reward  17.599. Std of Reward  7.053. T aining.
TouchCube. Step: 3700000. Time Elapsed: 5371.427 s. Mean Reward  17.704. Std of Reward  6.902. T aining.
TouchCube. Step: 3750000. Time Elapsed: 5459.356 s. Mean Reward  17.503. Std of Reward  7.198. T aining.
TouchCube. Step: 3800000. Time Elapsed: 5548.941 s. Mean Reward  17.260. Std of Reward  7.530. T aining.
TouchCube. Step: 3850000. Time Elapsed: 5637.050 s. Mean Reward  17.444. Std of Reward  7.284. T aining.
TouchCube. Step: 3900000. Time Elapsed: 5722.583 s. Mean Reward  17.374. Std of Reward  7.380. T aining.
TouchCube. Step: 3950000. Time Elapsed: 5809.752 s. Mean Reward  17.124. Std of Reward  7.701. T aining.
TouchCube. Step: 4000000. Time Elapsed: 5901.880 s. Mean Reward  17.342. Std of Reward  7.424. T aining.
ation.py:93] Converting to results\1\TouchCube\TouchCube-399999 .onnx
ation.py:105] Exported results\1\TouchCube\TouchCube-3999991.on x
ager.py:43] Removed checkpoint model results\1\TouchCube\TouchC be-14999 4.onnx.
TouchCube. Step: 4050000. Time Elapsed: 5985.685 s. Mean Reward  17.703. Std of Reward  6.913. T aining.
TouchCube. Step: 4100000. Time Elapsed: 6072.875 s. Mean Reward  17.631. Std of Reward  7.014. T aining.
TouchCube. Step: 4150000. Time Elapsed: 6157.968 s. Mean Reward  17.791. Std of Reward  6.771. T aining.
TouchCube. Step: 4200000. Time Elapsed: 6245.044 s. Mean Reward  17.452. Std of Reward  7.270. T aining.
TouchCube. Step: 4250000. Time Elapsed: 6329.007 s. Mean Reward  17.412. Std of Reward  7.326. T aining.
```


Cumulative Reward
tag: Environment/Cumulative Reward
100
-100
-300
-500
0
500k
1M
1.5M
2M
2.5M
3M
3.5M
4M
Episode Length
tag: Environment/Episode Length
2e+4
1.6e+4
1.2e+4
8e+3
4e+3
0
0
1M
2M
3M
4M
Policy Loss
tag: Losses/Policy Loss
0.0255
0.0245
0.0235
0.0225
0
1M
2M
3M
4M
Value Loss
tag: Losses/Value Loss
80
60
40
20
0
0
500k
1M
1.5M
2M
2.5M
3M
3.5M
4M

5. Test performance



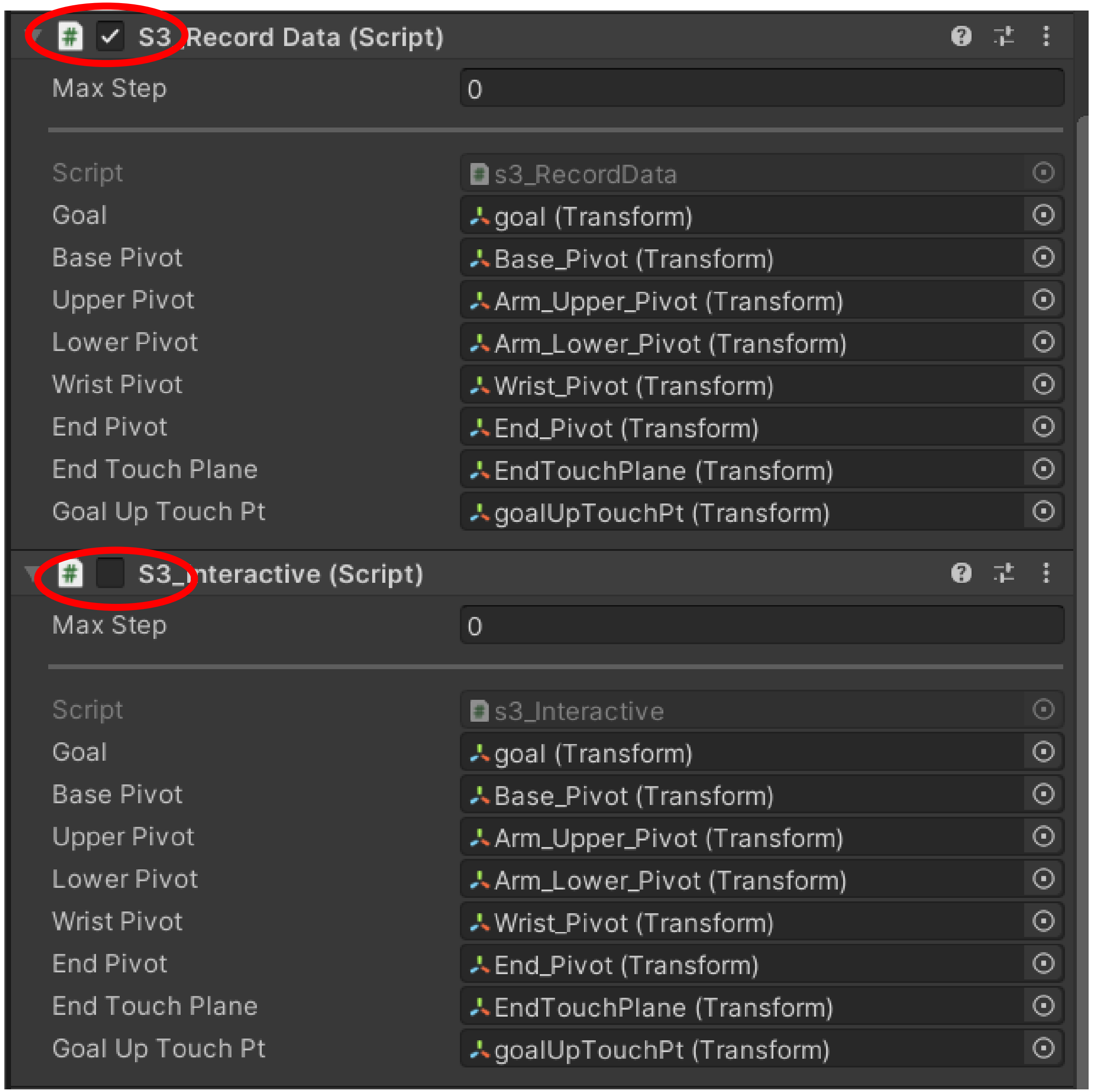
Assign trained NN



Behavior Parameters

Behavior Name	TouchCube
Vector Observation	
Space Size	7
Stacked Vectors	●
Vector Action	
Space Type	Continuous
Space Size	4
Model	TouchCube (NN Model)
Inference Device	GPU
Behavior Type	Inference Only
Team Id	0
Use Child Sensors	✓
Observable Attribute Handling	Ignore

2 types of tests – (1) data recording



☑ S3_Record Data (Script)

Max Step	0
Script	s3_RecordData
Goal	goal (Transform)
Base Pivot	Base_Pivot (Transform)
Upper Pivot	Arm_Upper_Pivot (Transform)
Lower Pivot	Arm_Lower_Pivot (Transform)
Wrist Pivot	Wrist_Pivot (Transform)
End Pivot	End_Pivot (Transform)
End Touch Plane	EndTouchPlane (Transform)
Goal Up Touch Pt	goalUpTouchPt (Transform)

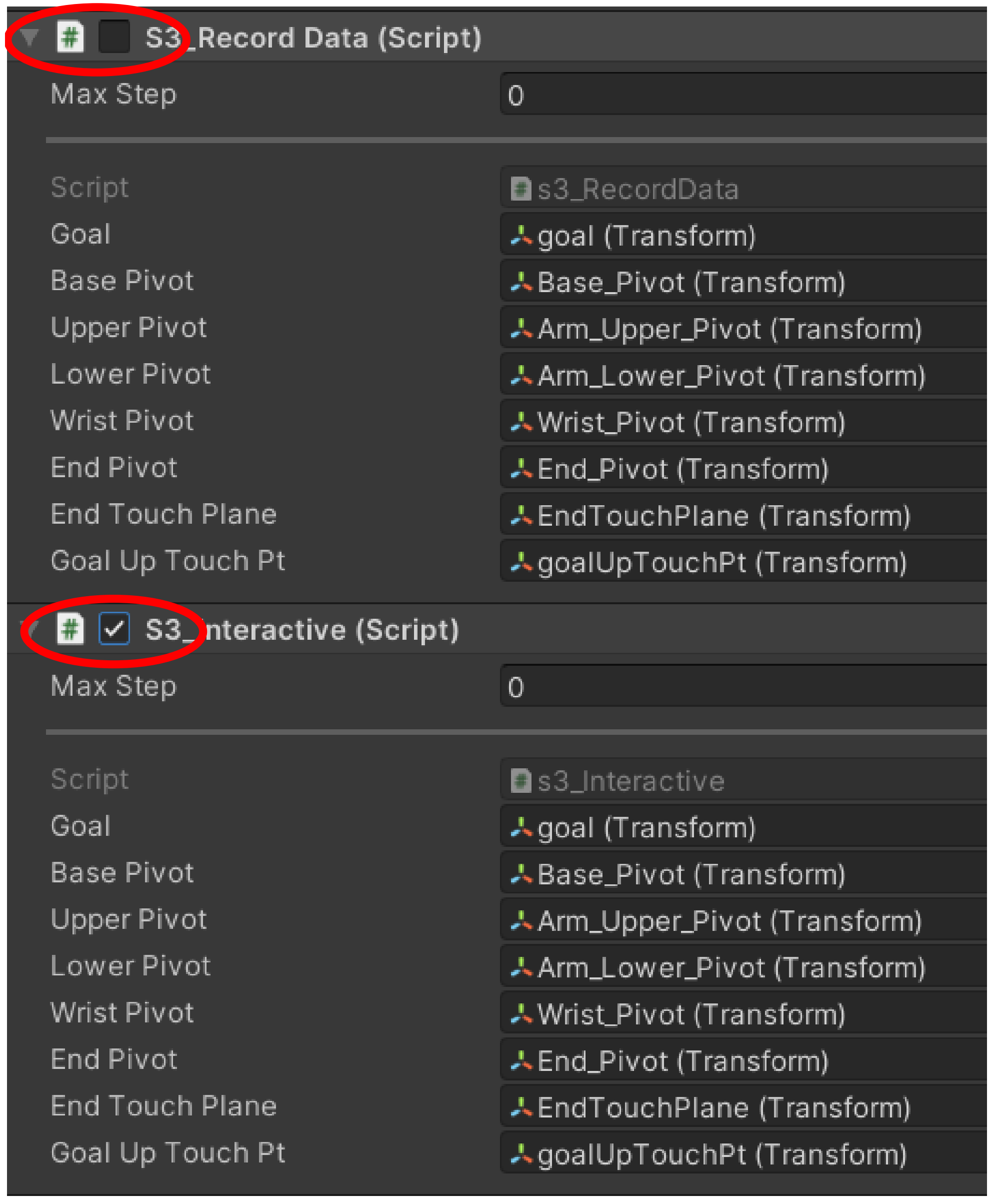
□ S3_Interactive (Script)

Max Step	0
Script	s3_Interactive
Goal	goal (Transform)
Base Pivot	Base_Pivot (Transform)
Upper Pivot	Arm_Upper_Pivot (Transform)
Lower Pivot	Arm_Lower_Pivot (Transform)
Wrist Pivot	Wrist_Pivot (Transform)
End Pivot	End_Pivot (Transform)
End Touch Plane	EndTouchPlane (Transform)
Goal Up Touch Pt	goalUpTouchPt (Transform)

Uncheck interactive test

2 types of tests – (2) interactive test

Uncheck data-recording test



□ S3_Record Data (Script)

Max Step	0
Script	s3_RecordData
Goal	goal (Transform)
Base Pivot	Base_Pivot (Transform)
Upper Pivot	Arm_Upper_Pivot (Transform)
Lower Pivot	Arm_Lower_Pivot (Transform)
Wrist Pivot	Wrist_Pivot (Transform)
End Pivot	End_Pivot (Transform)
End Touch Plane	EndTouchPlane (Transform)
Goal Up Touch Pt	goalUpTouchPt (Transform)

☑ S3_Interactive (Script)

Max Step	0
Script	s3_Interactive
Goal	goal (Transform)
Base Pivot	Base_Pivot (Transform)
Upper Pivot	Arm_Upper_Pivot (Transform)
Lower Pivot	Arm_Lower_Pivot (Transform)
Wrist Pivot	Wrist_Pivot (Transform)
End Pivot	End_Pivot (Transform)
End Touch Plane	EndTouchPlane (Transform)
Goal Up Touch Pt	goalUpTouchPt (Transform)

Practice

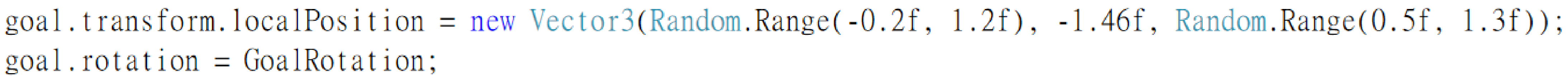
1. Initial position, all rotation angles =0
2. Different robot has different initial positions.

Base pivot (0, 45, 0)
Upper pivot (45, 0, 0)
Lower pivot (45, 0, 0)
Wrist pivot (45, 0, 0)

Base pivot (0, 0, 0)
Upper pivot (0, 0, 0)
Lower pivot (0, 0, 0)
Wrist pivot (0, 0, 0)

Base pivot (0, y, 0)
Upper pivot (x1, 0, 0)
Lower pivot (x2, 0, 0)
Wrist pivot (x3, 0, 0)

3. Wider goal initial position



```
goal.transform.localPosition = new Vector3(Random.Range(-0.2f, 1.2f), -1.46f, Random.Range(0.5f, 1.3f));
goal.rotation = GoalRotation;
```

Training to reach goal 1 → goal 2