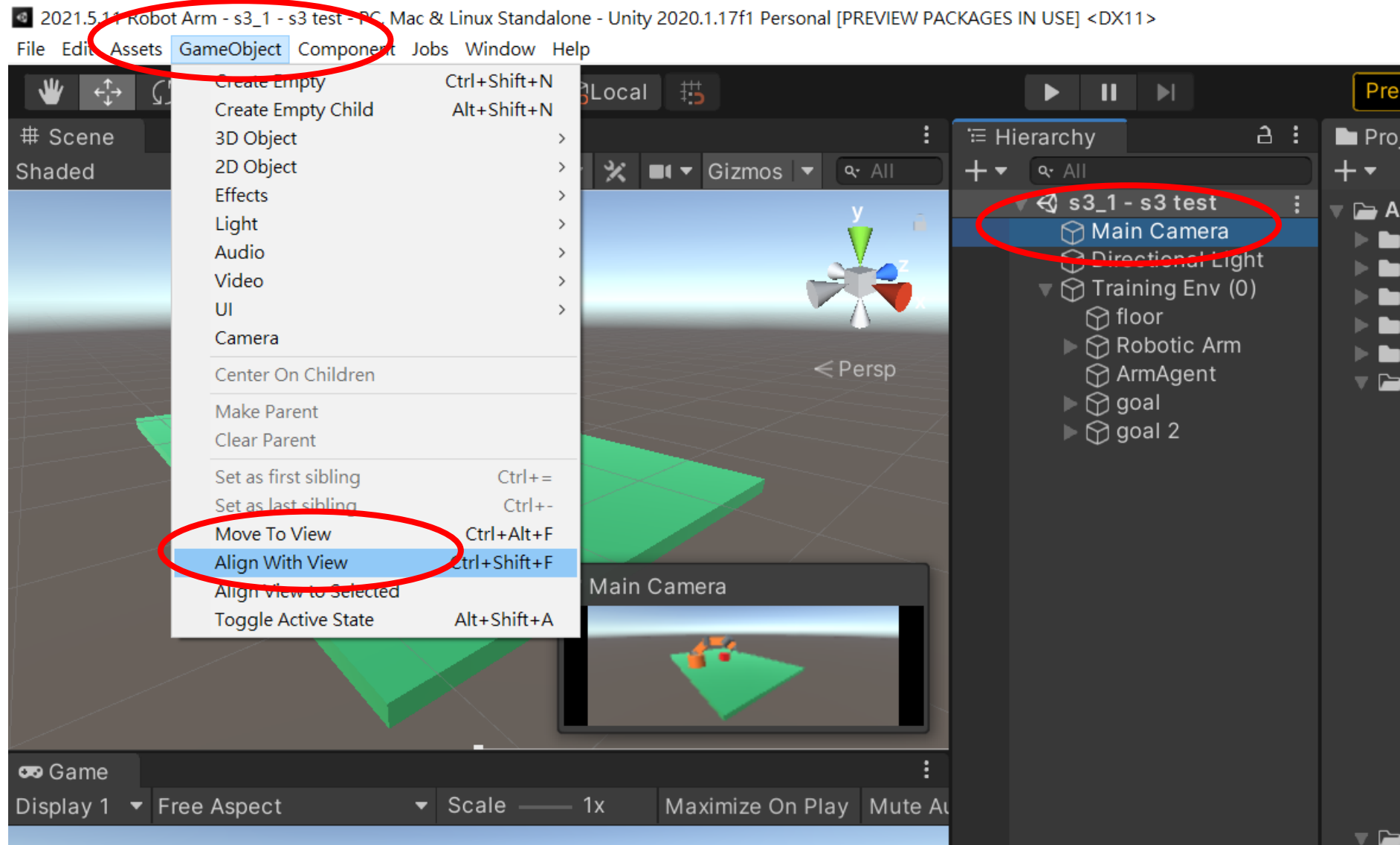
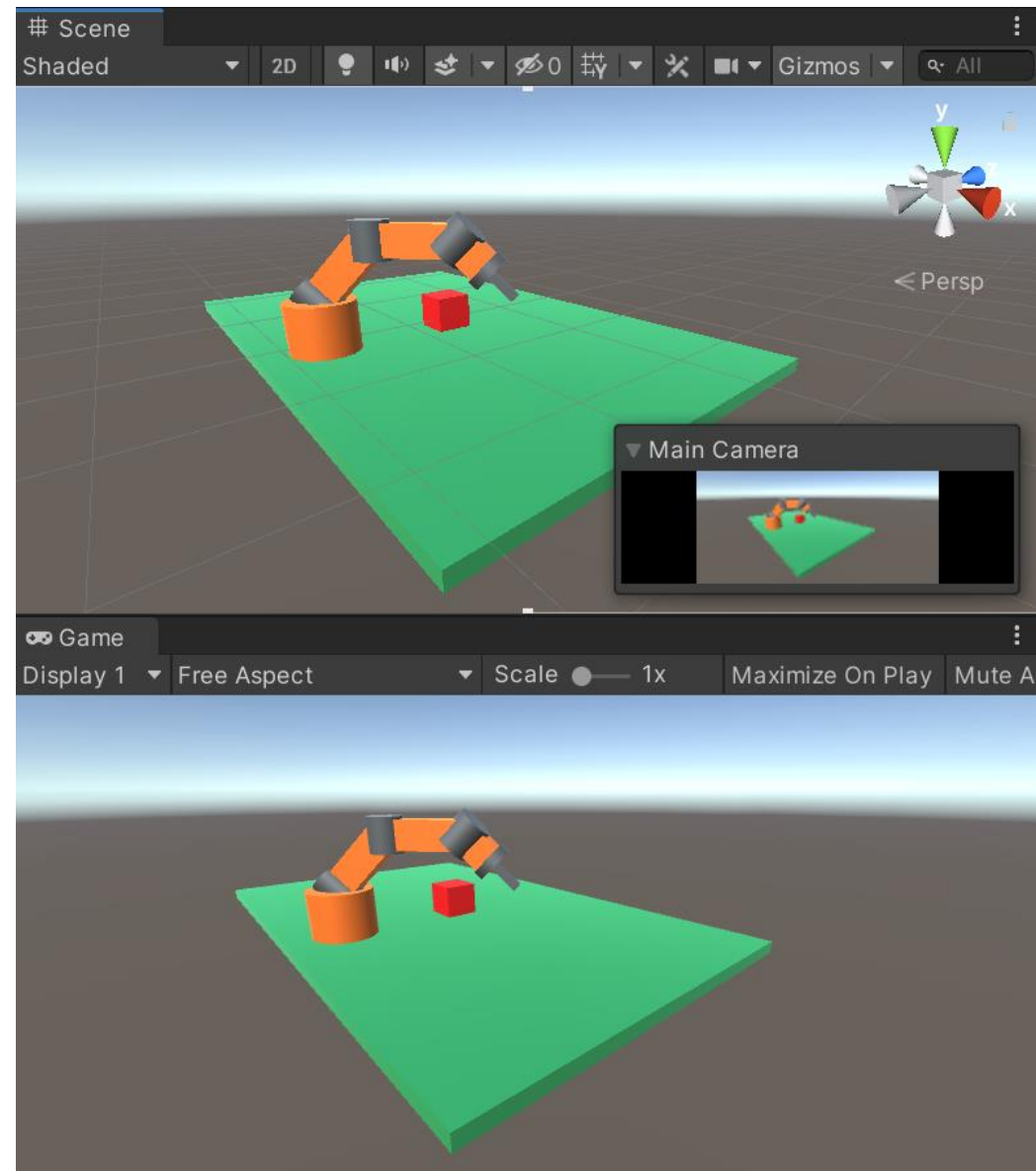


Unity – Camera manipulation

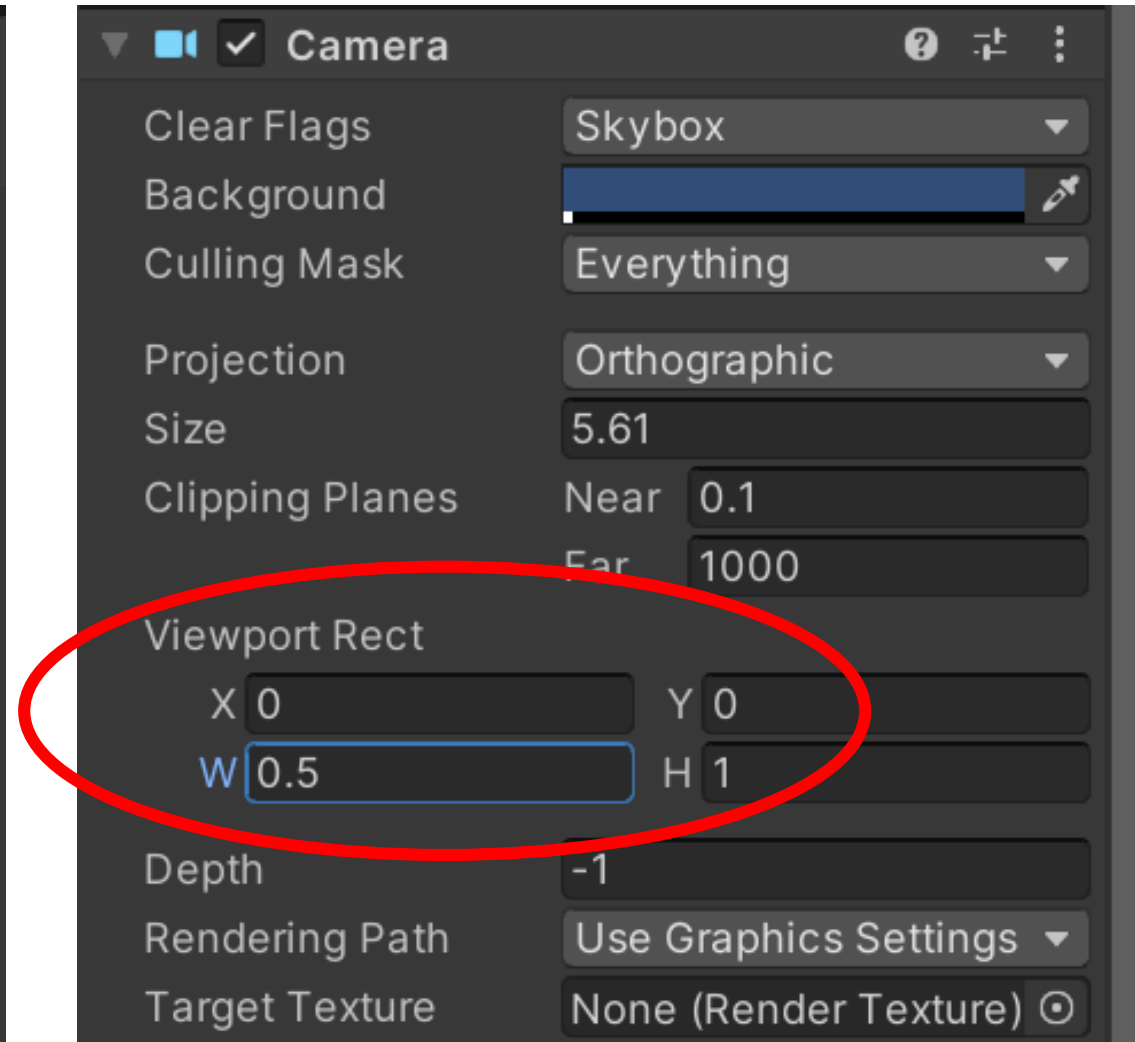
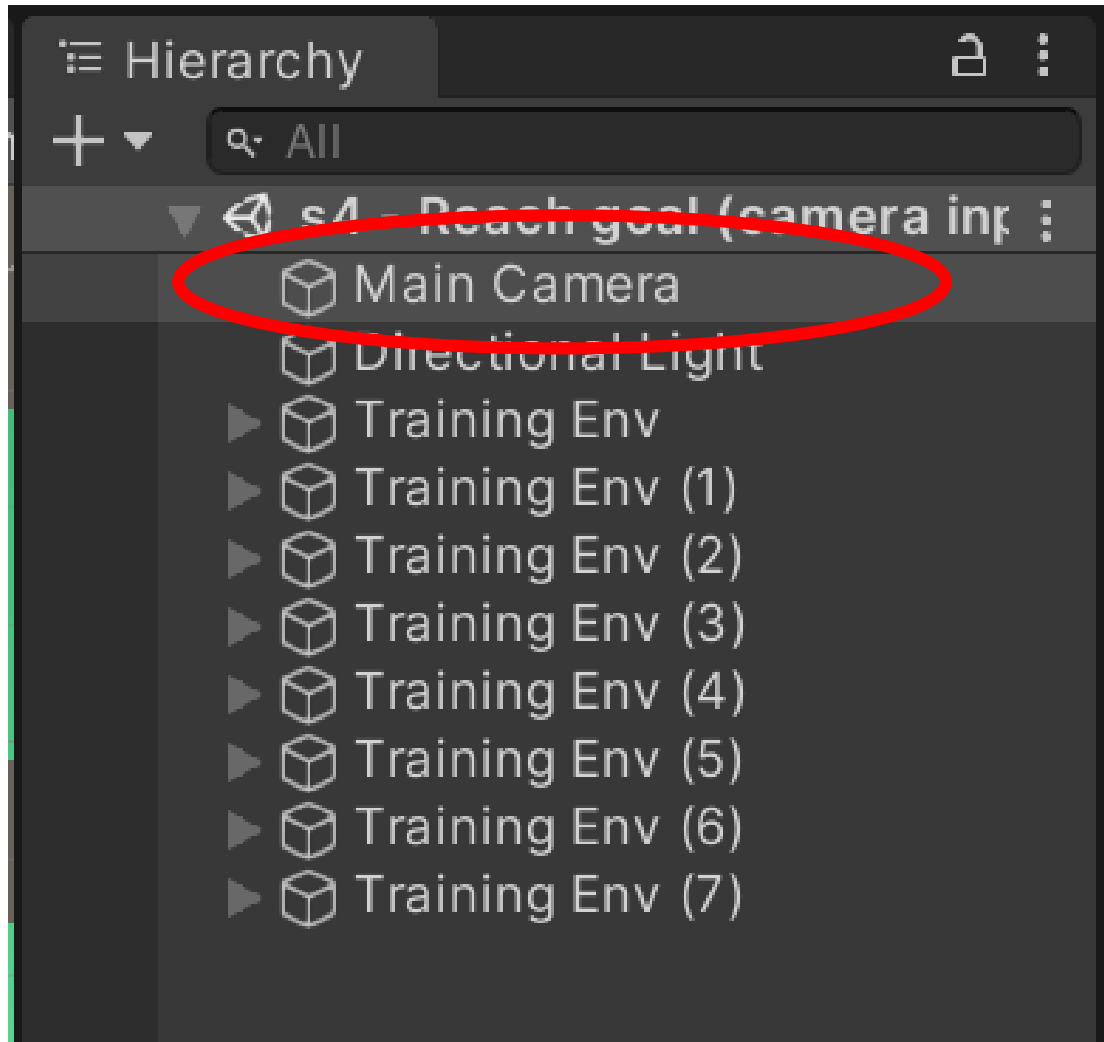
Align a camera with scene view



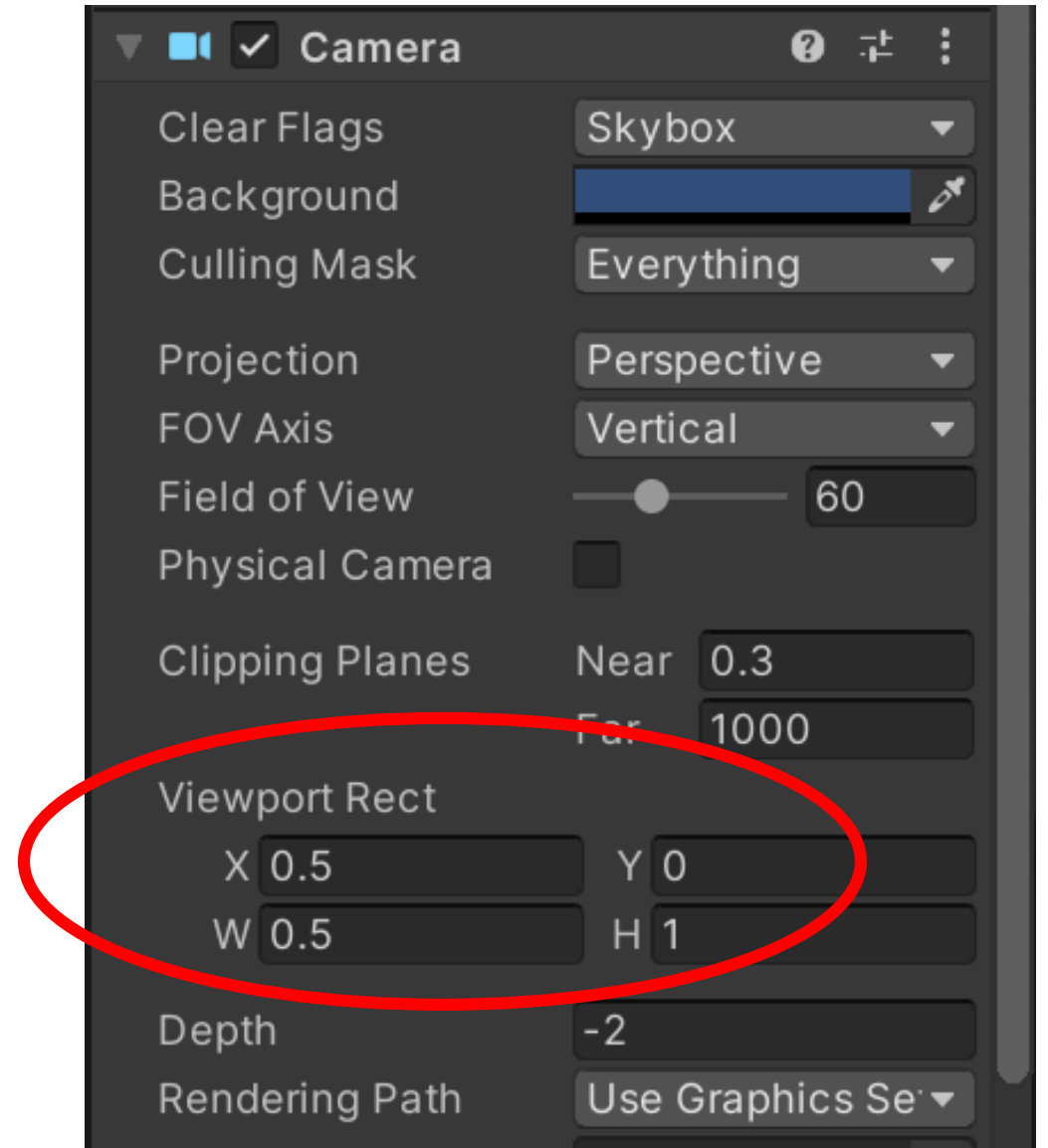
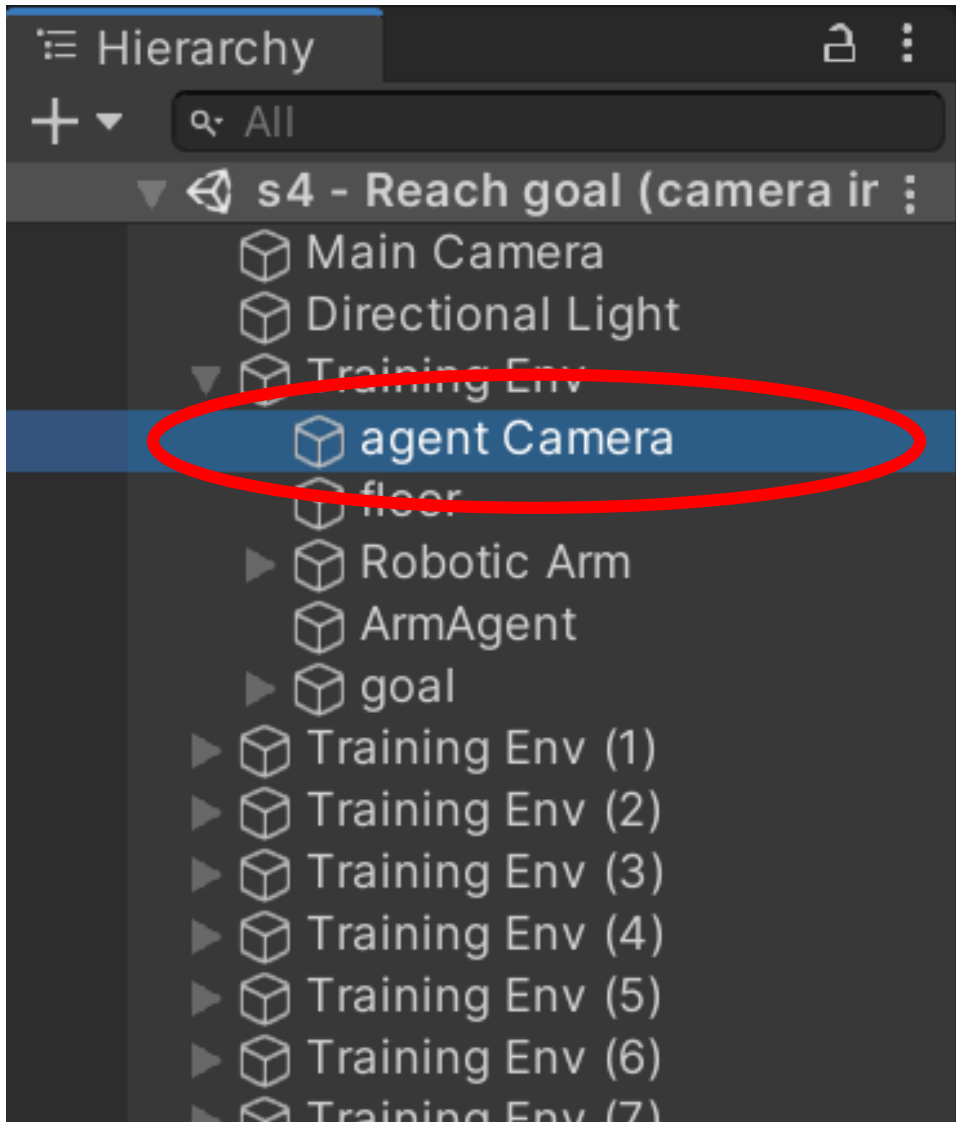
Game window view now is the same as scene view



Camera view port adjustment



Camera view port adjustment



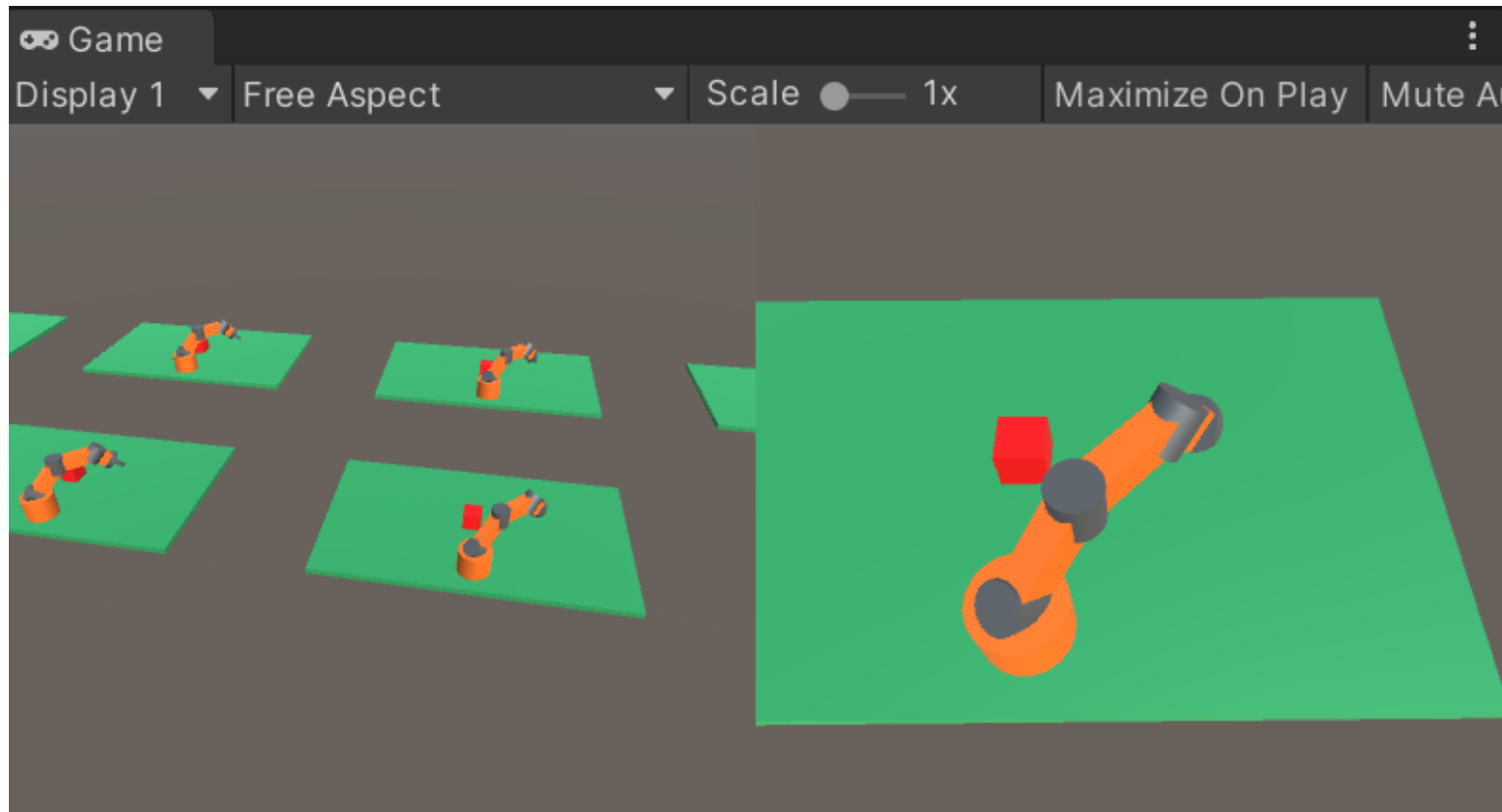
Camera view port adjustment

Main camera

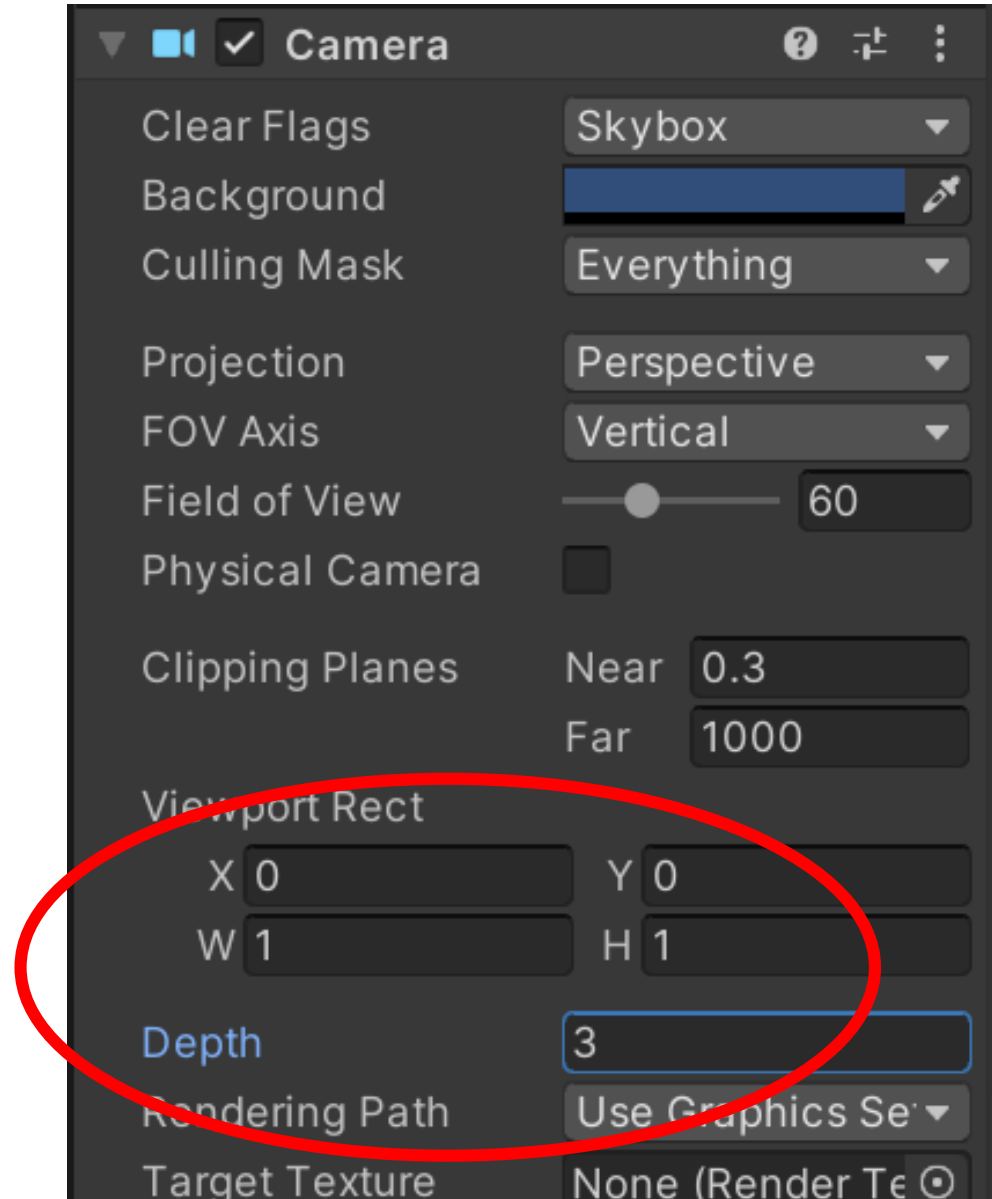
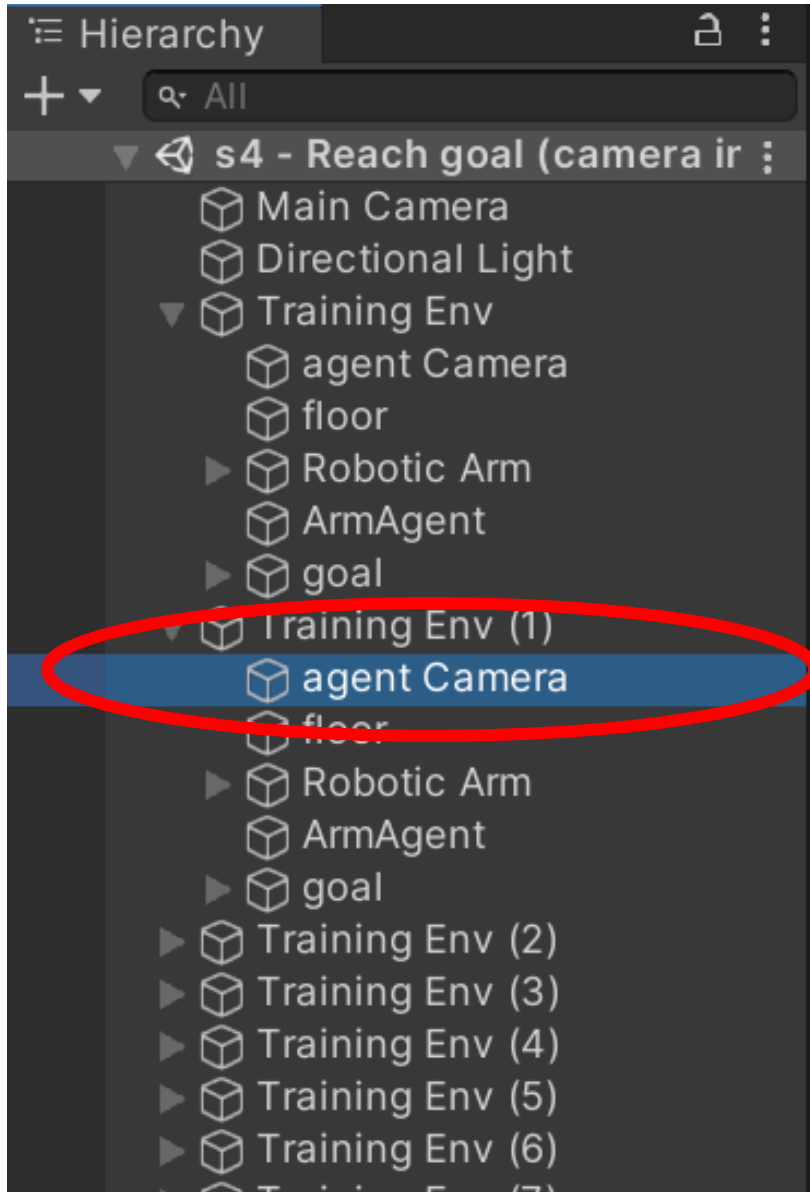
$(x=0, y=0), (w=0.5, h=1)$

Agent camera

$(x=0.5, y=0), (w=0.5, h=1)$



Camera depth



Game view display the camera with largest depth

Main camera

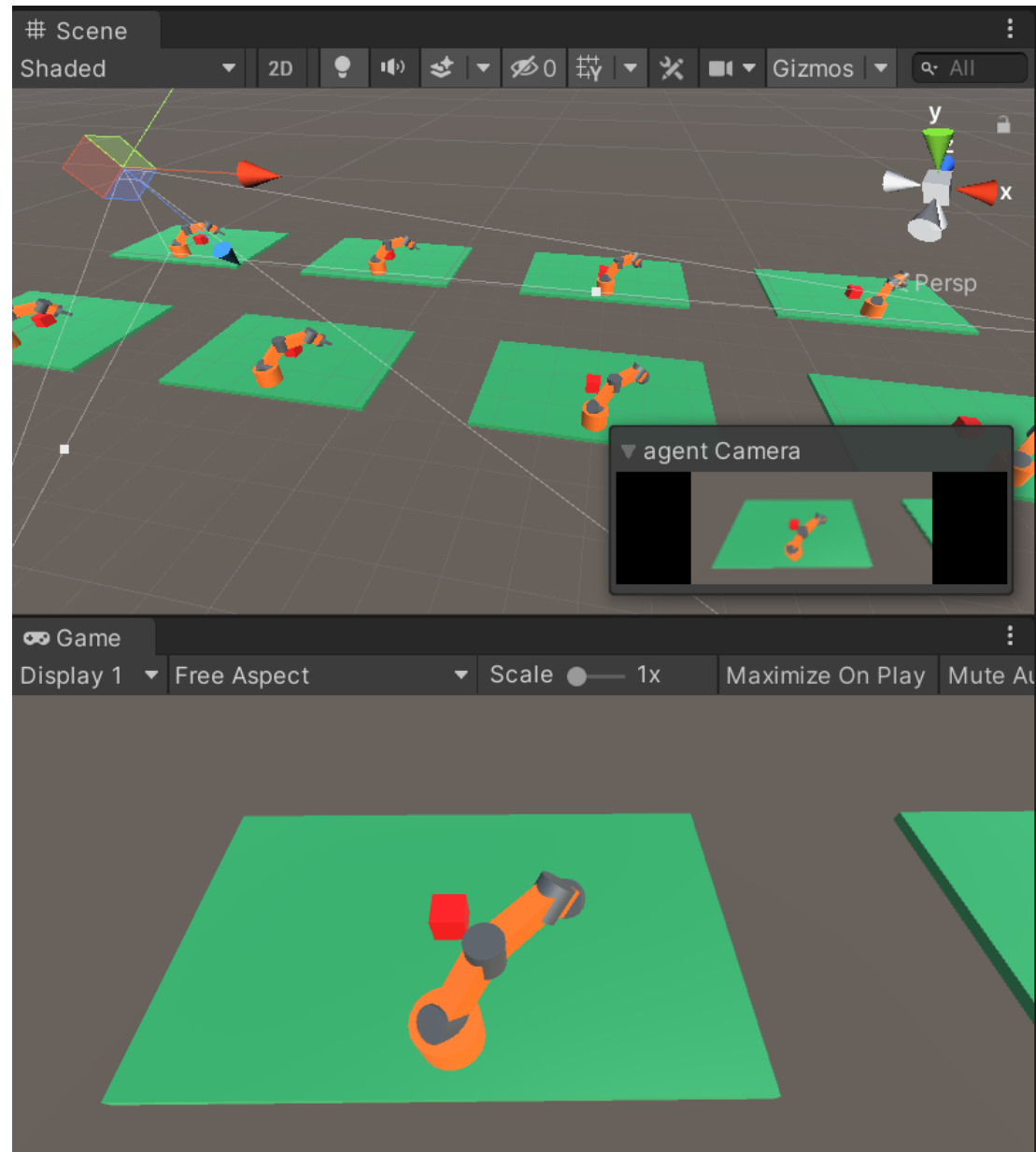
($x=0$, $y=0$), ($w=0.5$, $h=1$)

depth = -1

Agent camera

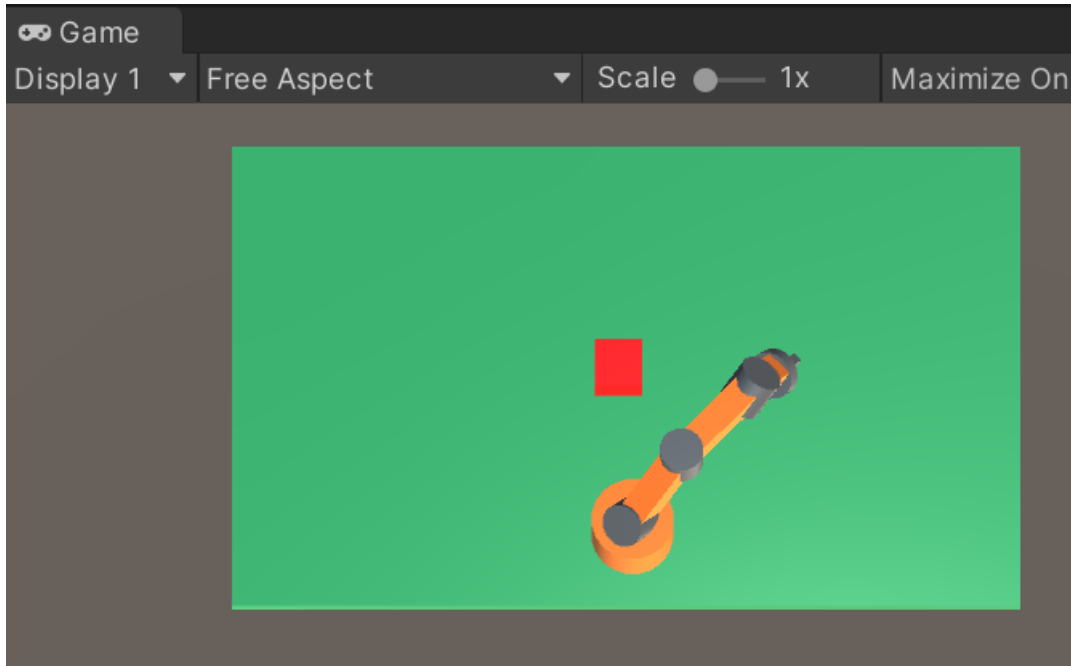
($x=0$, $y=0$), ($w=1$, $h=1$)

depth = 0

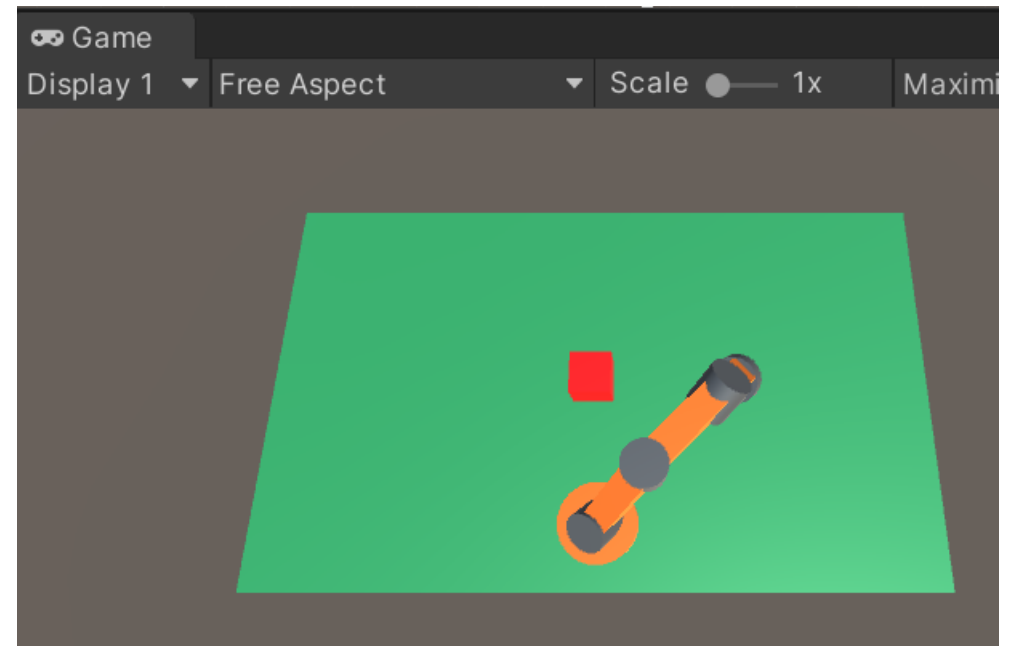


Orthographic or perspective camera

Orthographic, size=5

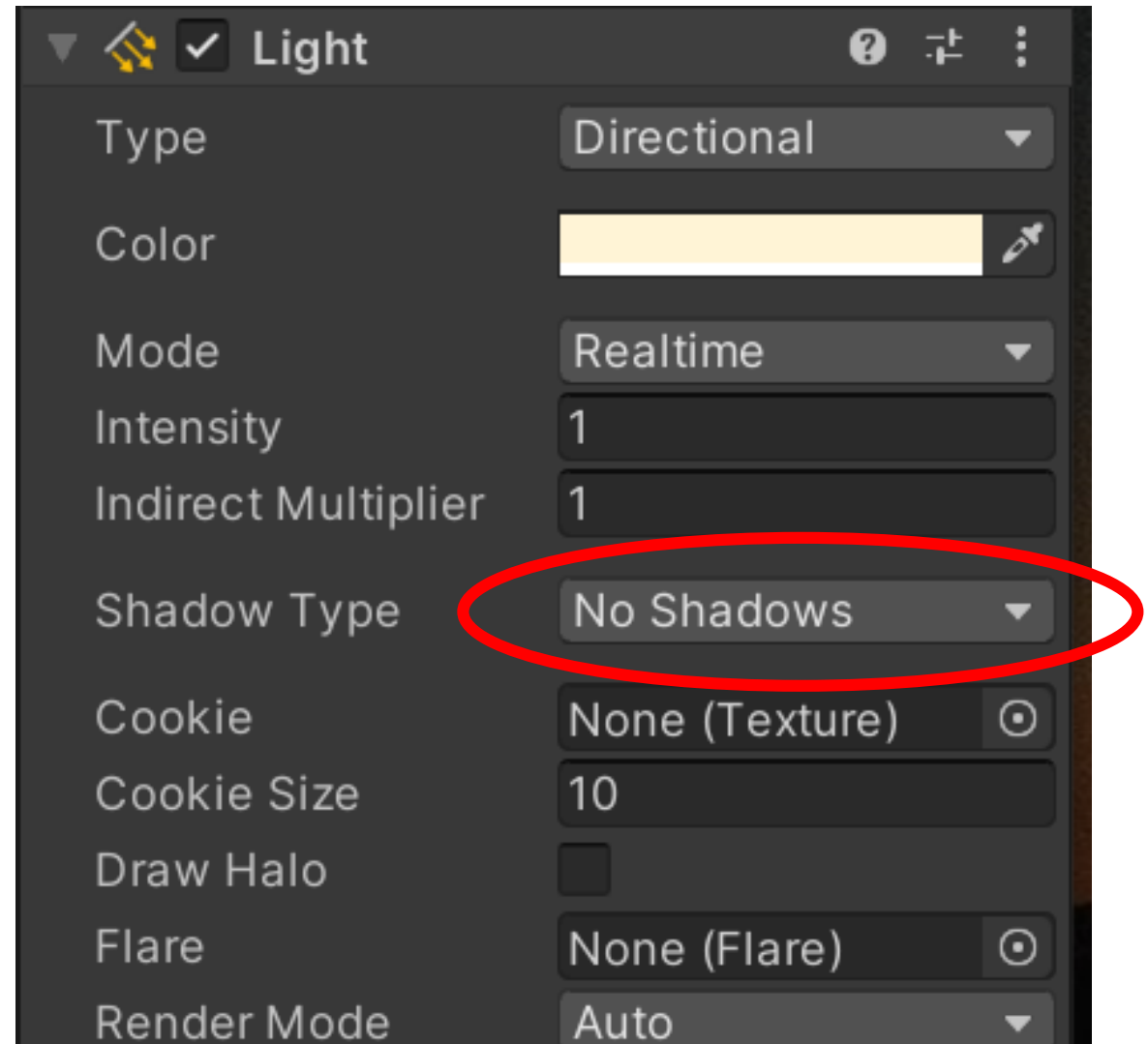
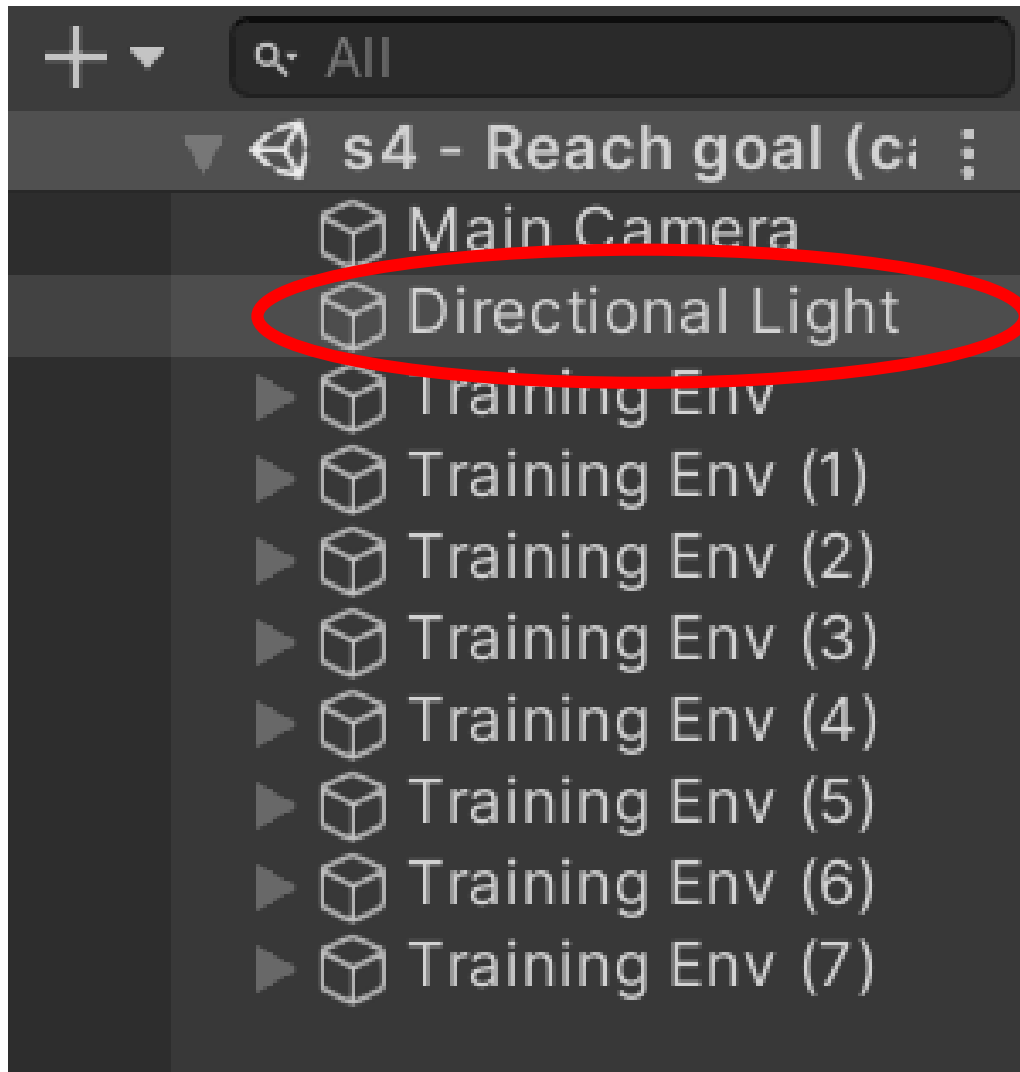


Perspective, DOF=60



Unity – Shadow

Turn on/off shadow



Unity – Interaction scripting

Start() function

```
public class s3 : Agent
{
    public Transform goal, goal2;
    public Transform BasePivot, UpperPivot, LowerPivot, WristPivot
    public Transform EndTouchPlane, goalUpTouchPt, goalDownTouchPt
    public GameObject LowerArmObj, WristObj, EndObj, GoalObj;
    public GameObject trainingVE;
    Quaternion BasePivotRoation, UpperPivotRotation, LowerPivotRot:
        GoalRotation, Goal2Rotation;
    Transform goalParent;

    void Start()
    {
        BasePivotRoation = BasePivot.rotation;
        UpperPivotRotation = UpperPivot.rotation;
        LowerPivotRotation = LowerPivot.rotation;
        WristPivotRotation = WristPivot.rotation;
        GoalRotation = goal.rotation;
        Goal2Rotation = goal2.rotation;
        goalParent = goal.transform.parent; //goal
        goal_y = goal.localPosition.y;
        goal2_y = goal2.localPosition.y;
    }
}
```

Update() function

```
void Update()
{
    if (stage == 1)
    {
        if (!PointTouch(EndTouchPlane, goalUpTouchPt, 0.6f))
        {
            RequestDecision();
        }
        else
        {
            accumulatedTime += Time.deltaTime; //accumulate time
            if (accumulatedTime > WaitTime)
            {
                //calculate parameters to perform animation to 1
                d1 = (45 - UnityEditor.TransformUtils.GetInspector
                    frames;
                d2 = (45 - UnityEditor.TransformUtils.GetInspector
                    frames;
```

Wait 0.8 sec in Update() loop

```
int frames = 500, frameNo = 1; // used in Upda
float d1 = 0, d2 = 0, d3 = 0, d4 = 0; // used
int stage=1;
float accumulatedTime=0, WaitTime = 0.8f;

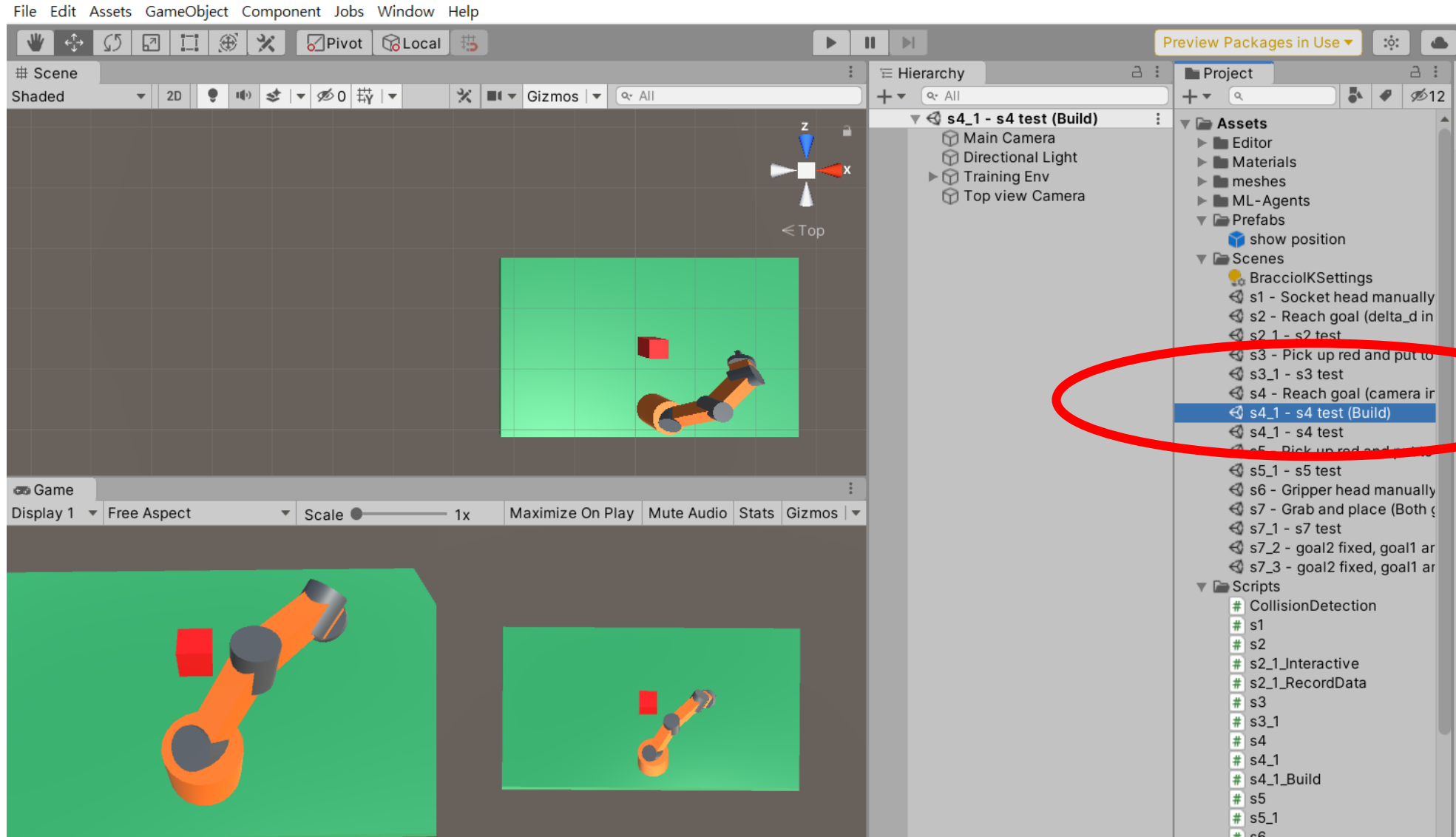
void Update()
{
    if (stage == 1)
    {
        if (!PointTouch(EndTouchPlane, goalUpTouchPt, 0.6f))
        {
            RequestDecision();
        }
        else
        {
            accumulatedTime += Time.deltaTime; //accumulate time
            if (accumulatedTime > WaitTime)
            {
```

Keyboard interaction

```
float x = Input.GetAxis("Horizontal");  
float z = Input.GetAxis("Vertical");  
goal.Translate(x*0.01f, 0, z*0.01f);  
if (Input.GetKeyDown(KeyCode.Space)) //  
{  
    stage = 1;  
    accumulatedTime = 0;  
}
```


Unity – Build executable AI game

Open the scene you want to build



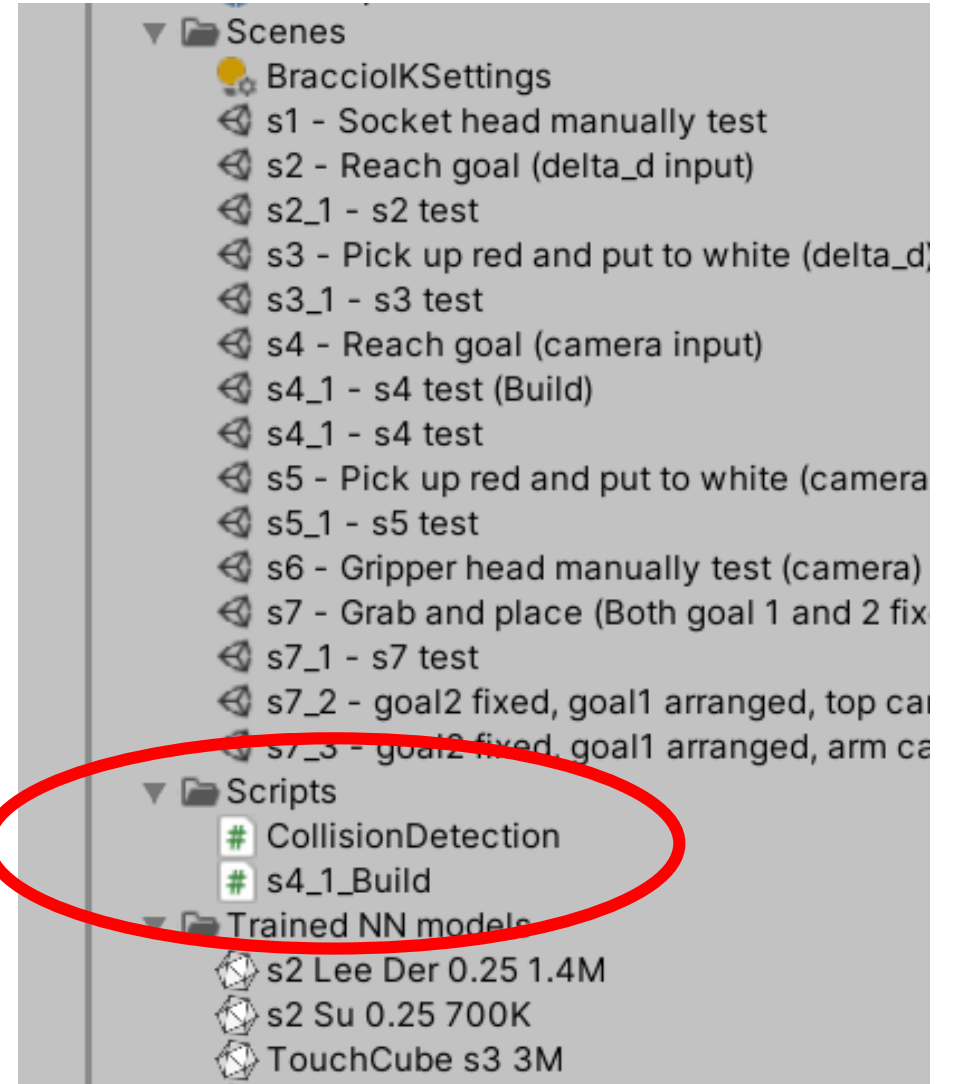
Delete scripts containing Unity Editor functions

- Unity can not build Unity Editor functions
- Export your project to Unity package
- Delete all scripts that contains UnityEditor functions

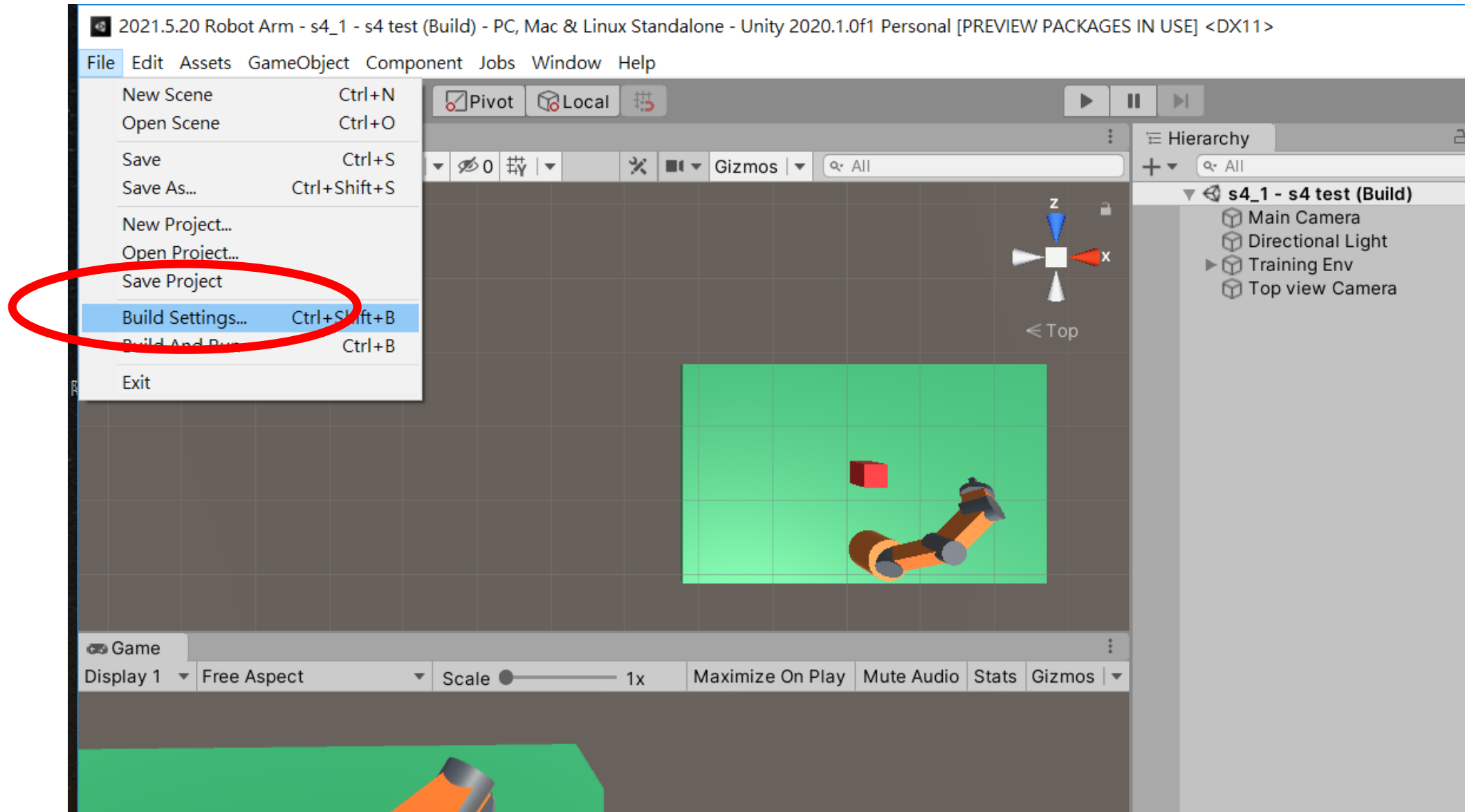
```

: UnityEditor.TransformUtils.GetInspectorRotation(BasePivot).y;
: UnityEditor.TransformUtils.GetInspectorRotation(UpperPivot).x;
: UnityEditor.TransformUtils.GetInspectorRotation(LowerPivot).x;
: UnityEditor.TransformUtils.GetInspectorRotation(WristPivot).x;
    00.00 Base Rotation Angle = 00.00

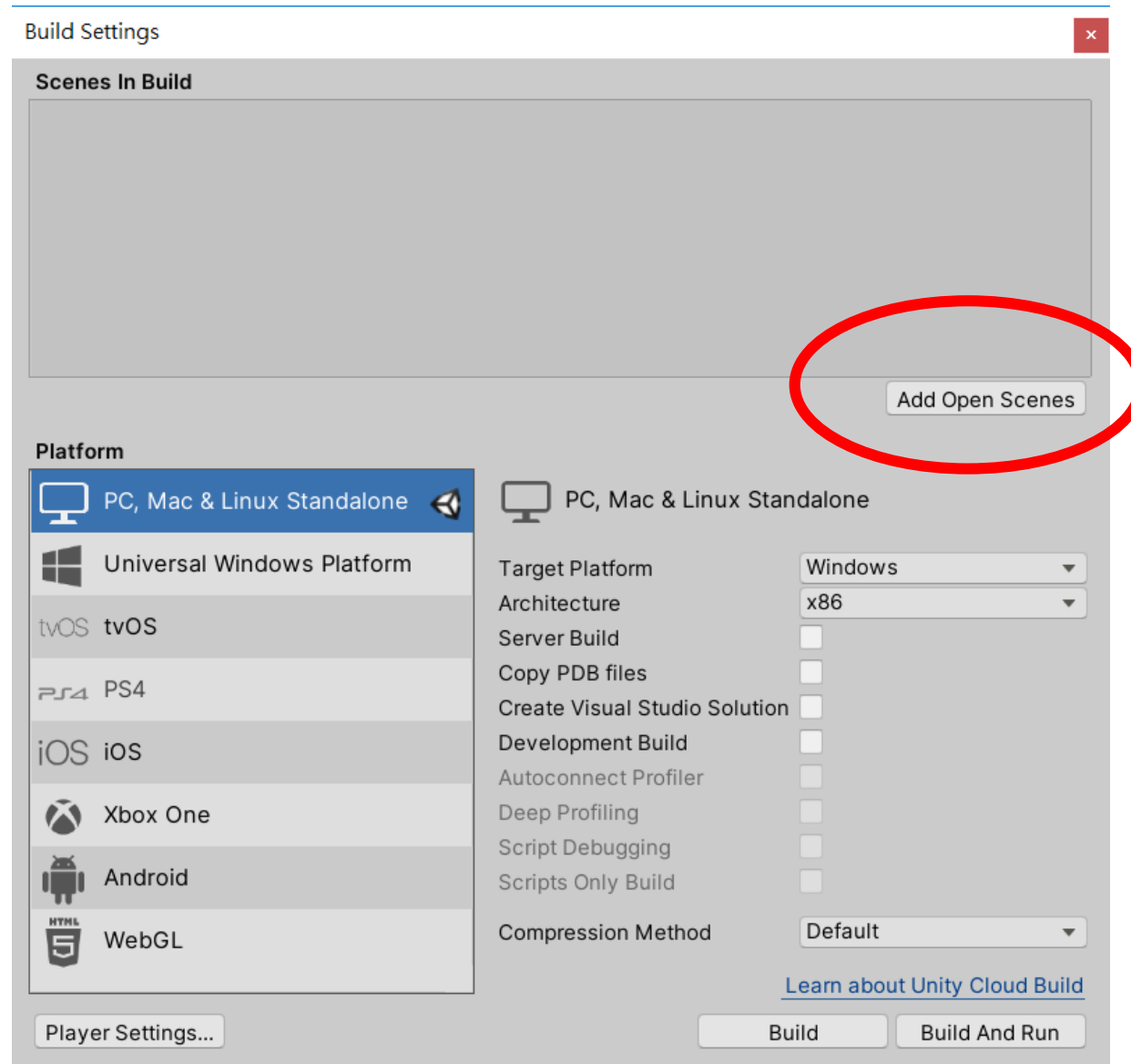
```



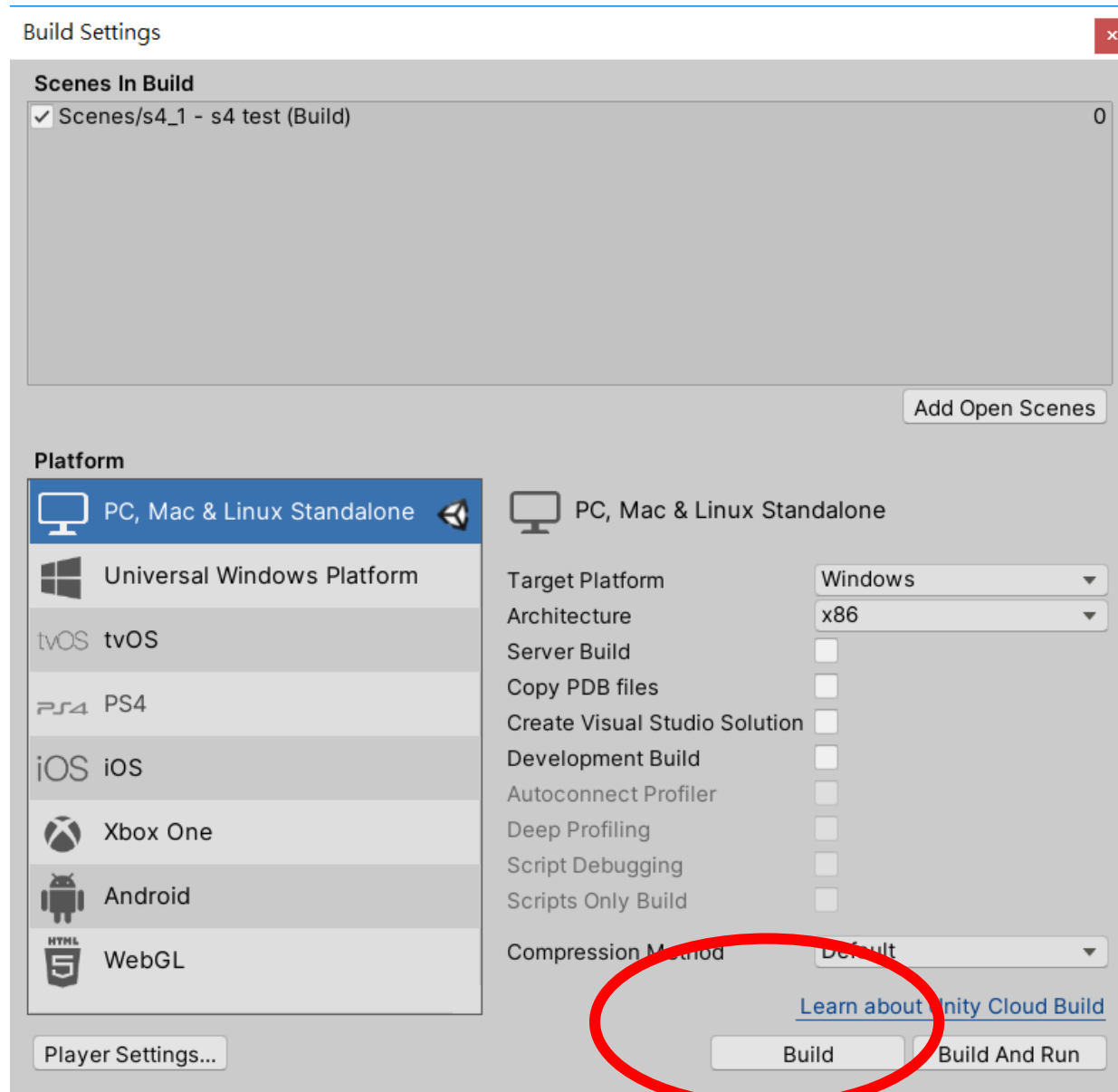
Build settings



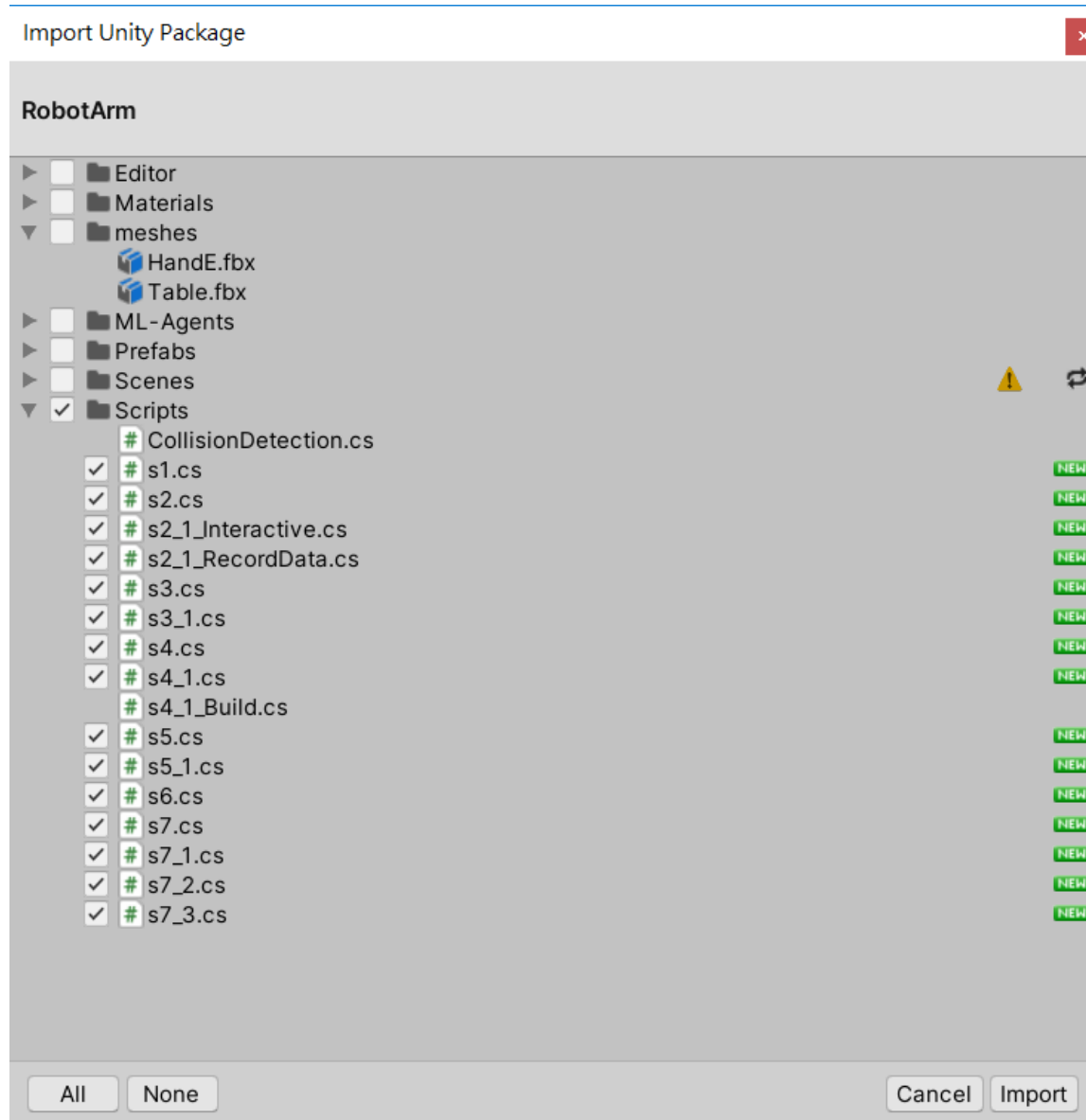
Build settings



Build executable file



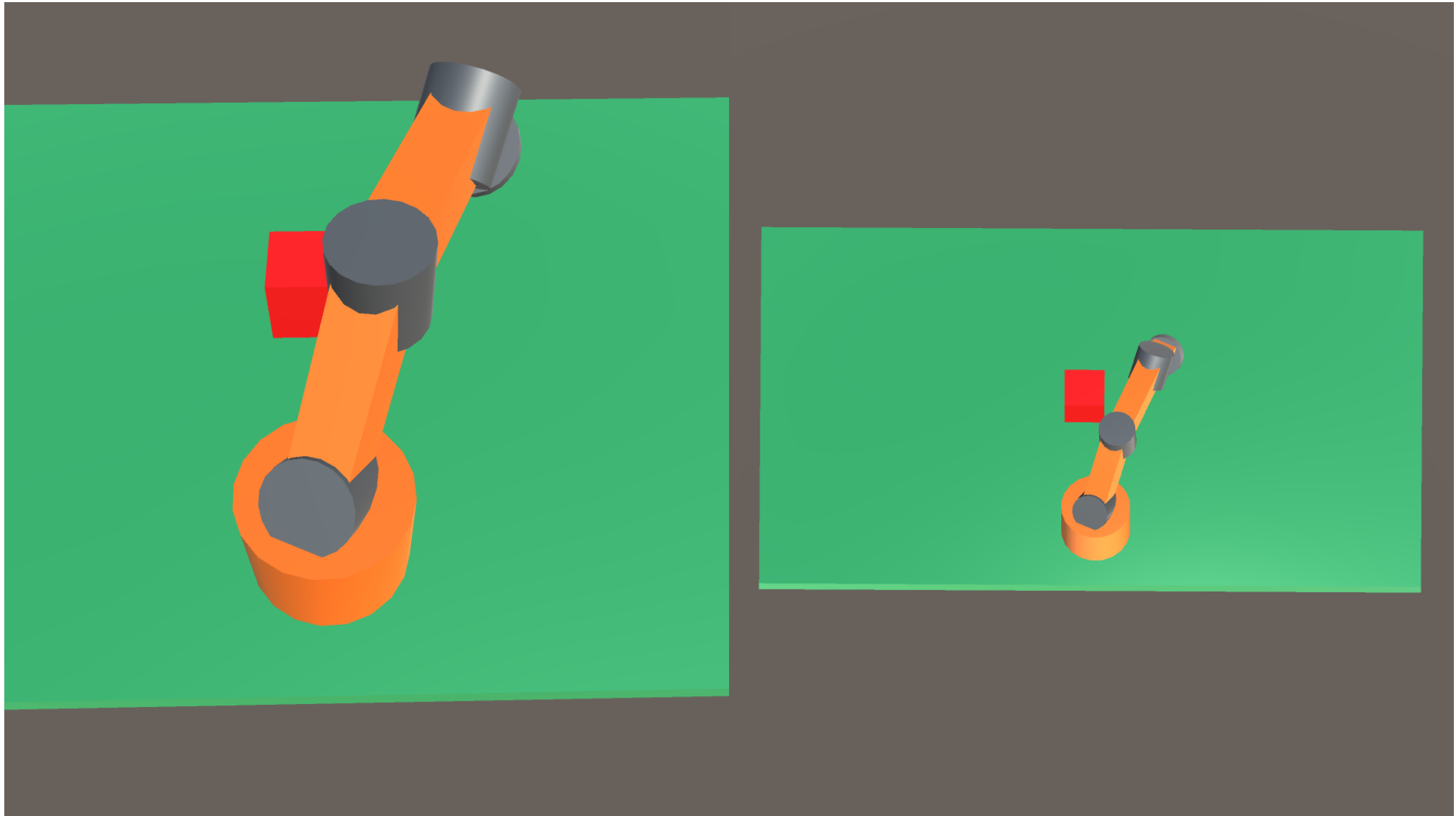
Remember to import scripts back to your project



Run your AI game

名稱	修改日期	類型	大小
 2021.5.20 Robot Arm_Data	2021/6/17 上午 0...	檔案資料夾	
 MonoBleedingEdge	2021/6/17 上午 0...	檔案資料夾	
 <u>2021.5.20 Robot Arm</u>	2020/7/16 上午 0...	應用程式	625 KB
 UnityCrashHandler32	2020/7/16 上午 0...	應用程式	1,064 KB
 UnityPlayer.dll	2020/7/16 上午 0...	應用程式擴充	20,782 KB

Run your AI game



Unity – ML agent scripting

On Episode Begin

```
public override void OnEpisodeBegin()  
{  
    float dist;  
    stage = 1; //1: reach goal, 2 - grabbin  
    goal.transform.parent = goalParent;  
  
    //use polar coordinate to calculate x,  
    goal.transform.localPosition = ReturnRa  
    goal.rotation = GoalRotation;  
  
    goal2.transform.localPosition = ReturnR
```

Collect Observations

```
public override void CollectObservations(VectorSensor sensor)
{
    sensor.AddObservation(stage);

    if (stage == 1)
        sensor.AddObservation(EndTouchPlane.position - goalUpTou
    else //stage =2
        sensor.AddObservation(goalDownTouchPt.position - goal2Up'

    float BaseRotationAngle = UnityEditor.TransformUtils.GetInsp
    float UArmRotationAngle = UnityEditor.TransformUtils.GetInsp
    float LArmRotationAngle = UnityEditor.TransformUtils.GetInsp
    float WRotationAngle = UnityEditor.TransformUtils.GetInspect
```

On Action Received

```
public override void OnActionReceived(float[] vectorAction)
{
    float speed = 1.0f;
    string msg;

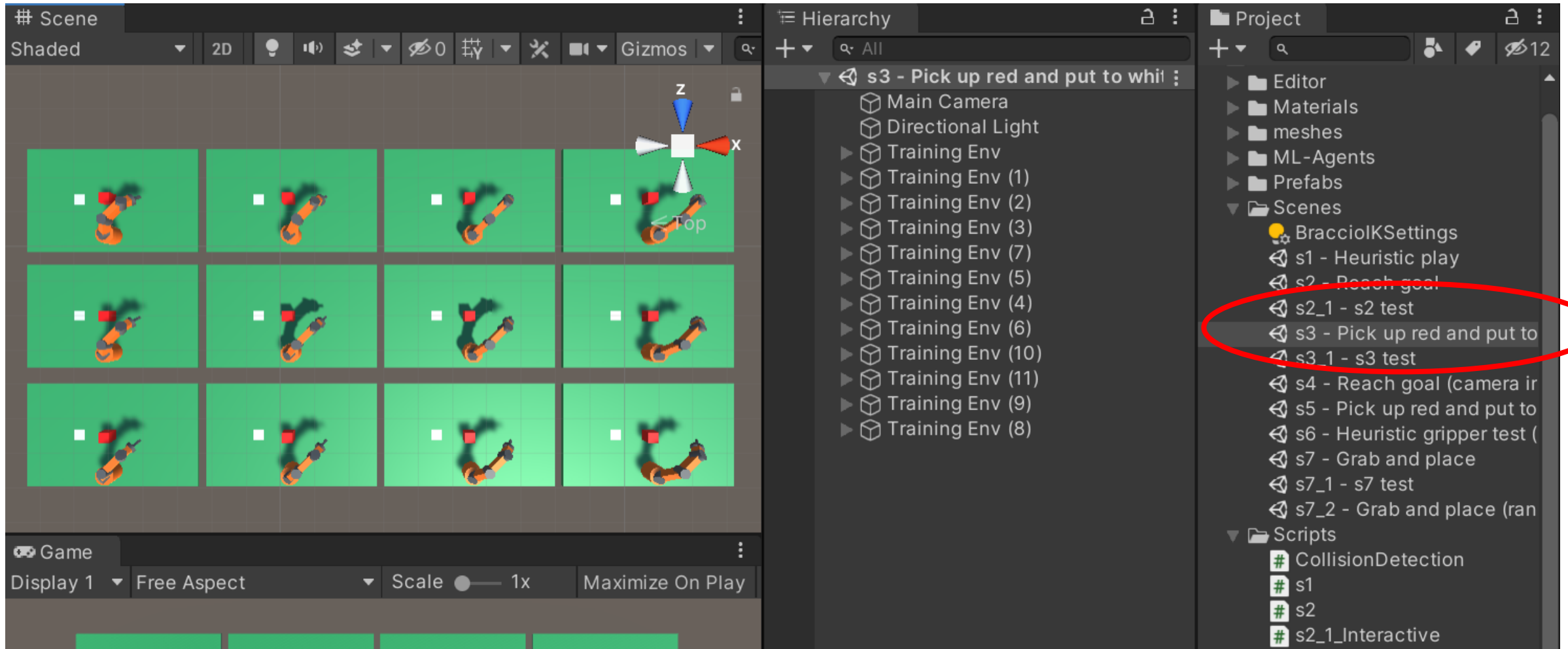
    AddReward(-0.005f); //avoid 要廢

    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);

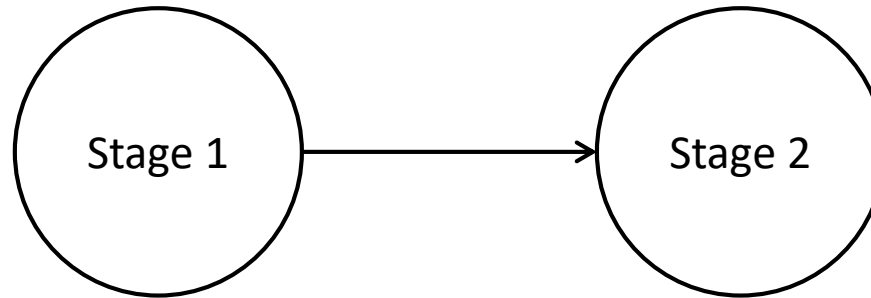
    //if rotation angle is out of range or collision happens
    if (!Rotation_in_range())
    {
```

S3 – Pick up red object and place it on top of white area using relative distance information

Open "s3 – Pick up red and put to white"



Training setting



$$s = (\Delta x, \Delta y, \Delta z, \theta_B, \theta_U, \theta_L, \theta_W)$$

In my script, I use +15 for goal 1 and +100 for goal 2. But +20 for both goal 1 and goal 2 should be OK.

$$r = \begin{cases} -0.005 & \text{per step} \\ -5 & \text{collision, out of range} \\ +20 & d_{stage1} \leq 0.1, d_{stage2} \leq 0.3 \end{cases}$$

$$a = (\Delta\theta_B, \Delta\theta_U, \Delta\theta_L, \Delta\theta_W)$$

NN: 8-512-512-512-4

Time horizon = 2000

Buffer size = 20480

Batch size = 2048

No. of training environment = 8

Goal initialize = randomly positioned in polar system $\theta = -80 \sim 80$, $r = 0.8 \sim 1.5$

Goal2 initialize = same as goal 1

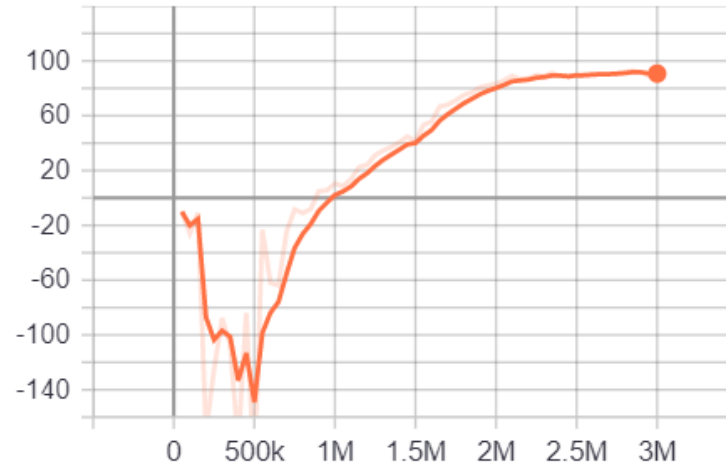
Arm initialize: $(\theta_B = 0, \theta_U = 45, \theta_L = 45, \theta_W = 45)$

Train 3M steps, looks promising

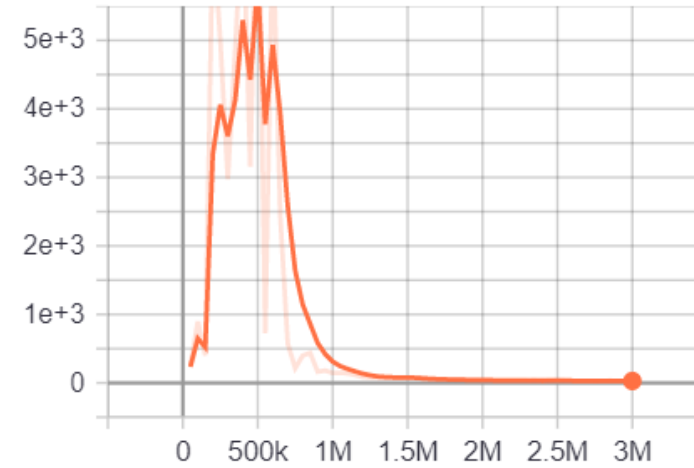
```
TouchCube. Step: 1400000. Time Elapsed: 1521.994 s. Mean Reward: 40.746. Std of Reward: 56.700. Training.
TouchCube. Step: 1450000. Time Elapsed: 1596.218 s. Mean Reward: 44.816. Std of Reward: 57.113. Training.
TouchCube. Step: 1500000. Time Elapsed: 1654.829 s. Mean Reward: 41.636. Std of Reward: 56.026. Training.
zation.py:93] Converting to results\1\TouchCube\TouchCube-1499998.onnx
zation.py:105] Exported results\1\TouchCube\TouchCube-1499998.onnx
TouchCube. Step: 1550000. Time Elapsed: 1710.662 s. Mean Reward: 53.289. Std of Reward: 58.197. Training.
TouchCube. Step: 1600000. Time Elapsed: 1772.814 s. Mean Reward: 55.664. Std of Reward: 57.972. Training.
TouchCube. Step: 1650000. Time Elapsed: 1839.616 s. Mean Reward: 66.710. Std of Reward: 57.145. Training.
TouchCube. Step: 1700000. Time Elapsed: 1905.361 s. Mean Reward: 68.004. Std of Reward: 56.702. Training.
TouchCube. Step: 1750000. Time Elapsed: 1974.348 s. Mean Reward: 71.199. Std of Reward: 55.970. Training.
TouchCube. Step: 1800000. Time Elapsed: 2034.998 s. Mean Reward: 75.024. Std of Reward: 54.928. Training.
TouchCube. Step: 1850000. Time Elapsed: 2093.756 s. Mean Reward: 77.129. Std of Reward: 54.333. Training.
TouchCube. Step: 1900000. Time Elapsed: 2153.092 s. Mean Reward: 80.611. Std of Reward: 52.868. Training.
TouchCube. Step: 1950000. Time Elapsed: 2214.371 s. Mean Reward: 82.048. Std of Reward: 52.347. Training.
TouchCube. Step: 2000000. Time Elapsed: 2273.202 s. Mean Reward: 83.202. Std of Reward: 51.604. Training.
zation.py:93] Converting to results\1\TouchCube\TouchCube-1999932.onnx
zation.py:105] Exported results\1\TouchCube\TouchCube-1999932.onnx
TouchCube. Step: 2050000. Time Elapsed: 2335.471 s. Mean Reward: 85.839. Std of Reward: 50.198. Training.
TouchCube. Step: 2100000. Time Elapsed: 2396.409 s. Mean Reward: 89.061. Std of Reward: 48.055. Training.
TouchCube. Step: 2150000. Time Elapsed: 2460.368 s. Mean Reward: 86.739. Std of Reward: 49.768. Training.
TouchCube. Step: 2200000. Time Elapsed: 2521.083 s. Mean Reward: 87.105. Std of Reward: 49.479. Training.
TouchCube. Step: 2250000. Time Elapsed: 2583.198 s. Mean Reward: 89.747. Std of Reward: 47.747. Training.
TouchCube. Step: 2300000. Time Elapsed: 2643.755 s. Mean Reward: 89.104. Std of Reward: 48.263. Training.
TouchCube. Step: 2350000. Time Elapsed: 2705.644 s. Mean Reward: 91.258. Std of Reward: 46.869. Training.
TouchCube. Step: 2400000. Time Elapsed: 2770.443 s. Mean Reward: 88.986. Std of Reward: 48.436. Training.
TouchCube. Step: 2450000. Time Elapsed: 2833.605 s. Mean Reward: 88.044. Std of Reward: 49.433. Training.
TouchCube. Step: 2500000. Time Elapsed: 2895.328 s. Mean Reward: 90.368. Std of Reward: 48.504. Training.
zation.py:93] Converting to results\1\TouchCube\TouchCube-2499995.onnx
zation.py:105] Exported results\1\TouchCube\TouchCube-2499995.onnx
TouchCube. Step: 2550000. Time Elapsed: 2956.370 s. Mean Reward: 89.639. Std of Reward: 48.371. Training.
TouchCube. Step: 2600000. Time Elapsed: 3028.253 s. Mean Reward: 90.657. Std of Reward: 47.175. Training.
TouchCube. Step: 2650000. Time Elapsed: 3095.190 s. Mean Reward: 90.670. Std of Reward: 47.247. Training.
TouchCube. Step: 2700000. Time Elapsed: 3157.692 s. Mean Reward: 90.483. Std of Reward: 47.232. Training.
TouchCube. Step: 2750000. Time Elapsed: 3226.099 s. Mean Reward: 91.113. Std of Reward: 46.861. Training.
TouchCube. Step: 2800000. Time Elapsed: 3285.665 s. Mean Reward: 91.785. Std of Reward: 46.426. Training.
TouchCube. Step: 2850000. Time Elapsed: 3349.785 s. Mean Reward: 92.851. Std of Reward: 46.171. Training.
TouchCube. Step: 2900000. Time Elapsed: 3413.532 s. Mean Reward: 91.497. Std of Reward: 46.714. Training.
TouchCube. Step: 2950000. Time Elapsed: 3480.496 s. Mean Reward: 89.342. Std of Reward: 48.568. Training.
TouchCube. Step: 3000000. Time Elapsed: 3545.361 s. Mean Reward: 90.762. Std of Reward: 47.138. Training.
```

Train 3M steps, looks promising

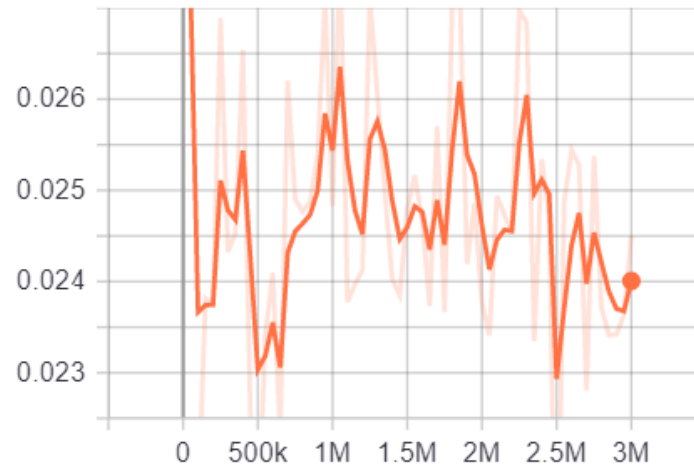
Cumulative Reward
tag: Environment/Cumulative Reward



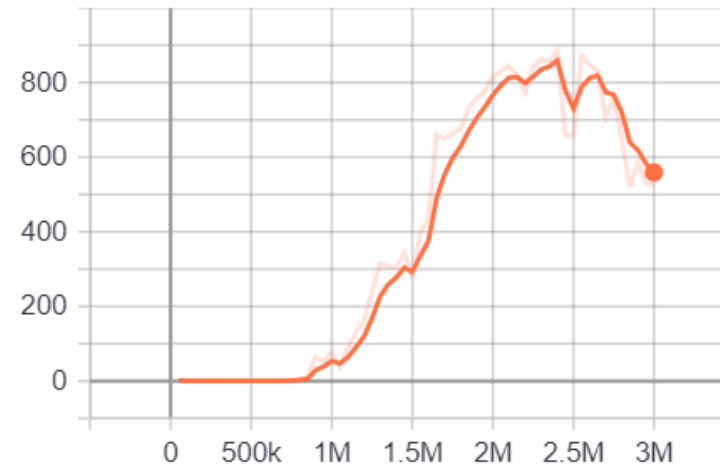
Episode Length
tag: Environment/Episode Length



Policy Loss
tag: Losses/Policy Loss

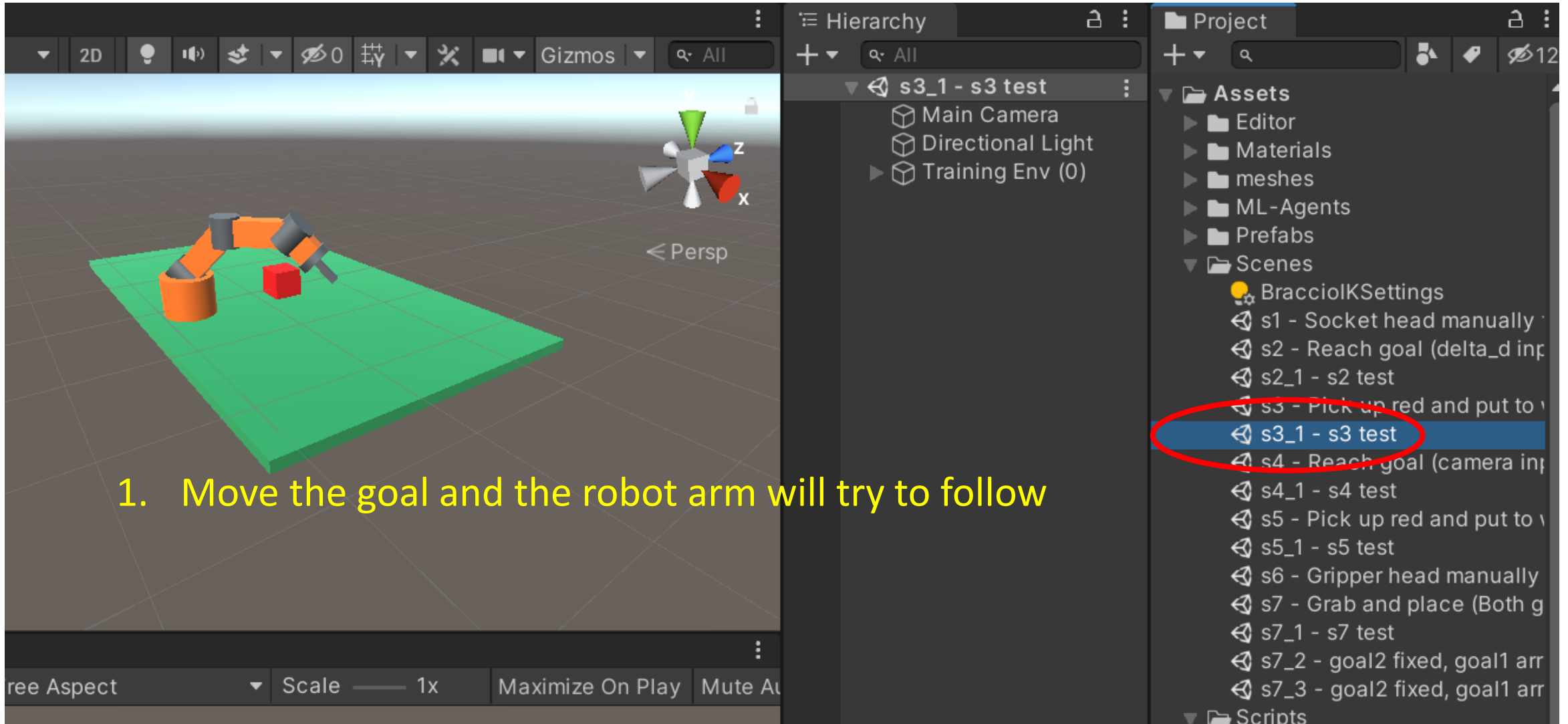


Value Loss
tag: Losses/Value Loss



Test – Play with robot arm

Open "s3_1"



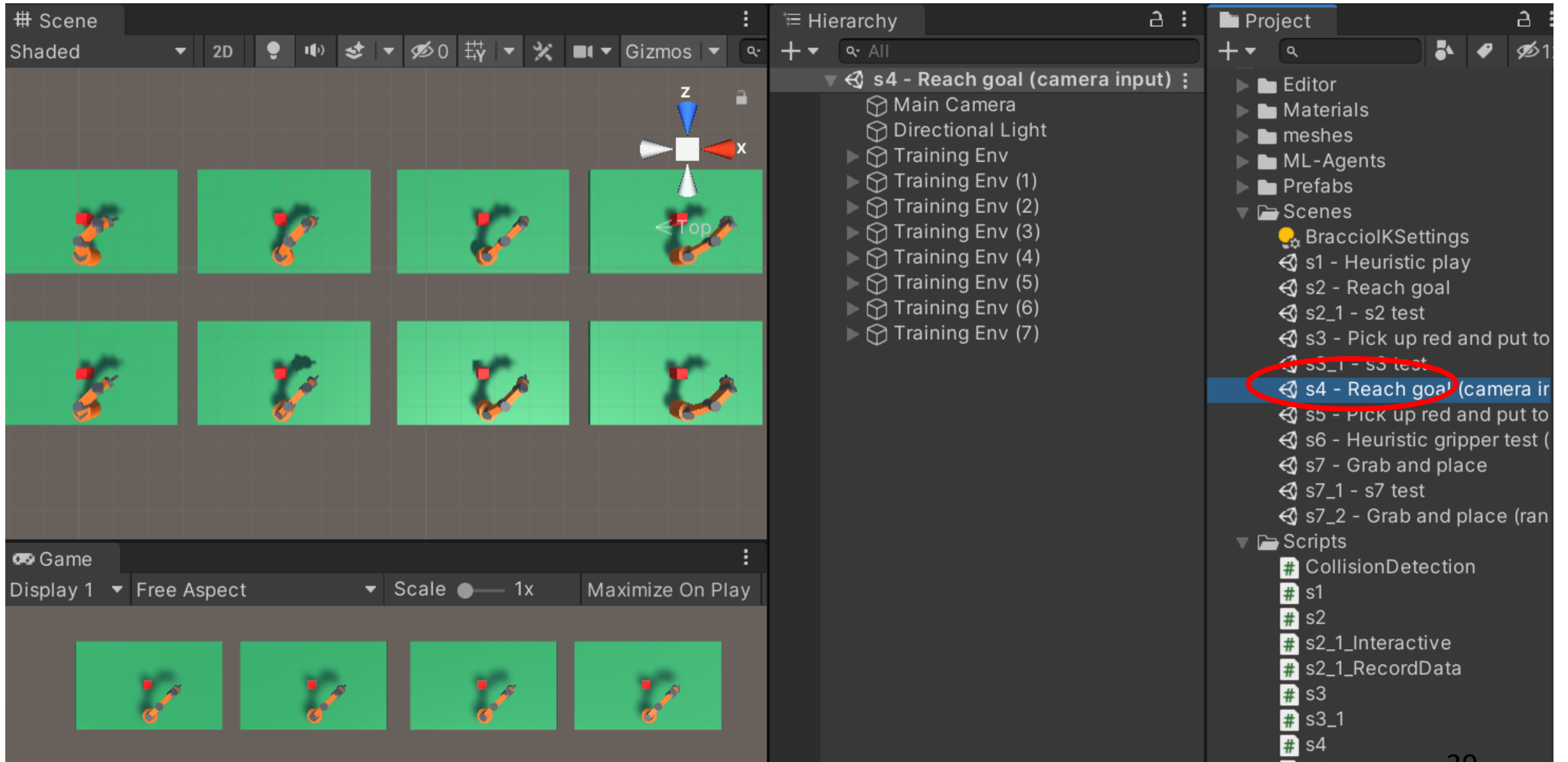
HW4(2)

- Describe training setting
- Show tensor board plots and discuss your training performance
- Describe test performance (recorded data, interactive test)



Scene s4 – Reach goal using camera image

Open s4



Training setting

s = feature map vector from a CNN, size = 2592

Input image to the CNN is captured by a camera from top, size = 84x84x3

$$a = (\Delta\theta_B, \Delta\theta_U, \Delta\theta_L, \Delta\theta_W)$$

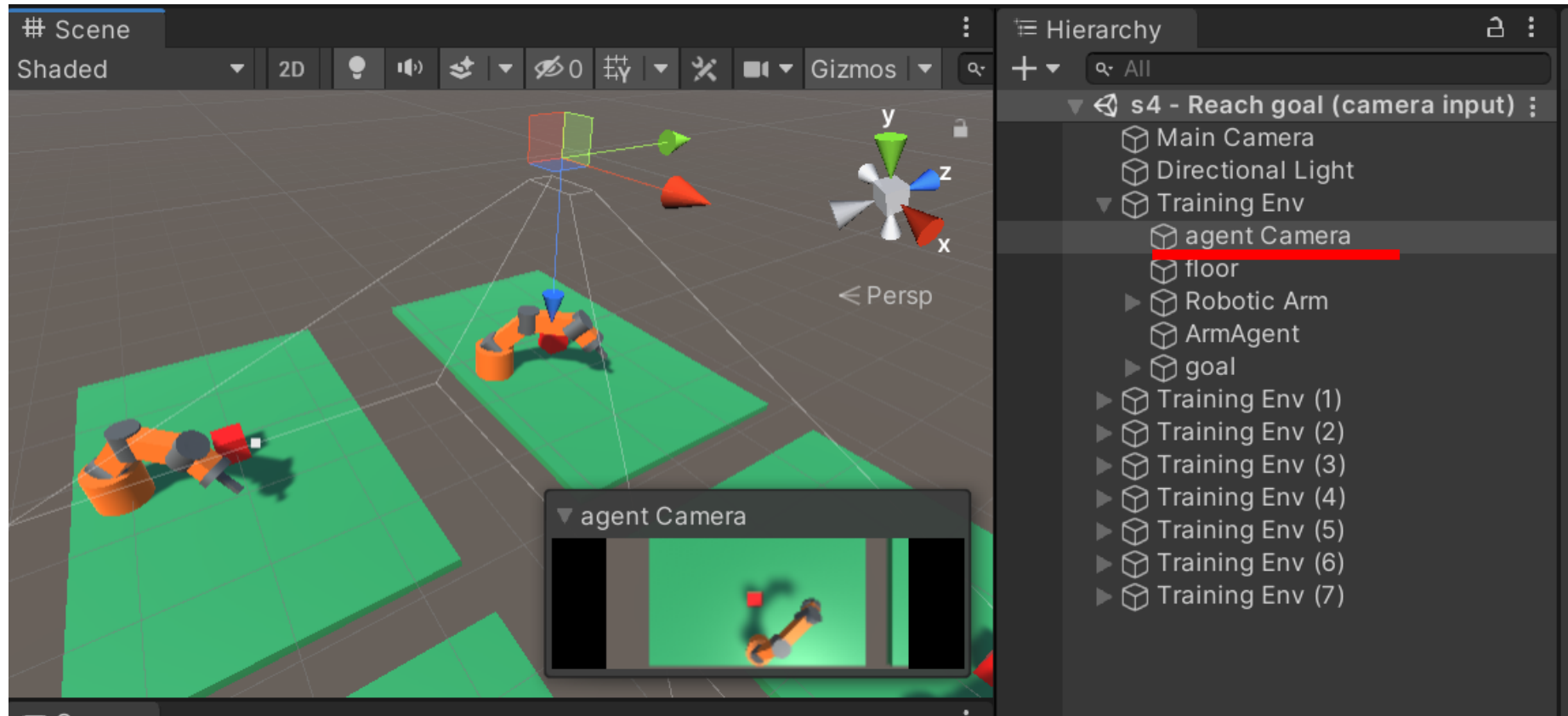
$$r = \begin{cases} -0.005 & \text{per step} \\ -5 & \text{collision, out of range} \\ +20 & \text{goal, } d \leq 0.5 \end{cases} \quad \text{Try 0.3? 0.25}$$

No. of training environment = 8

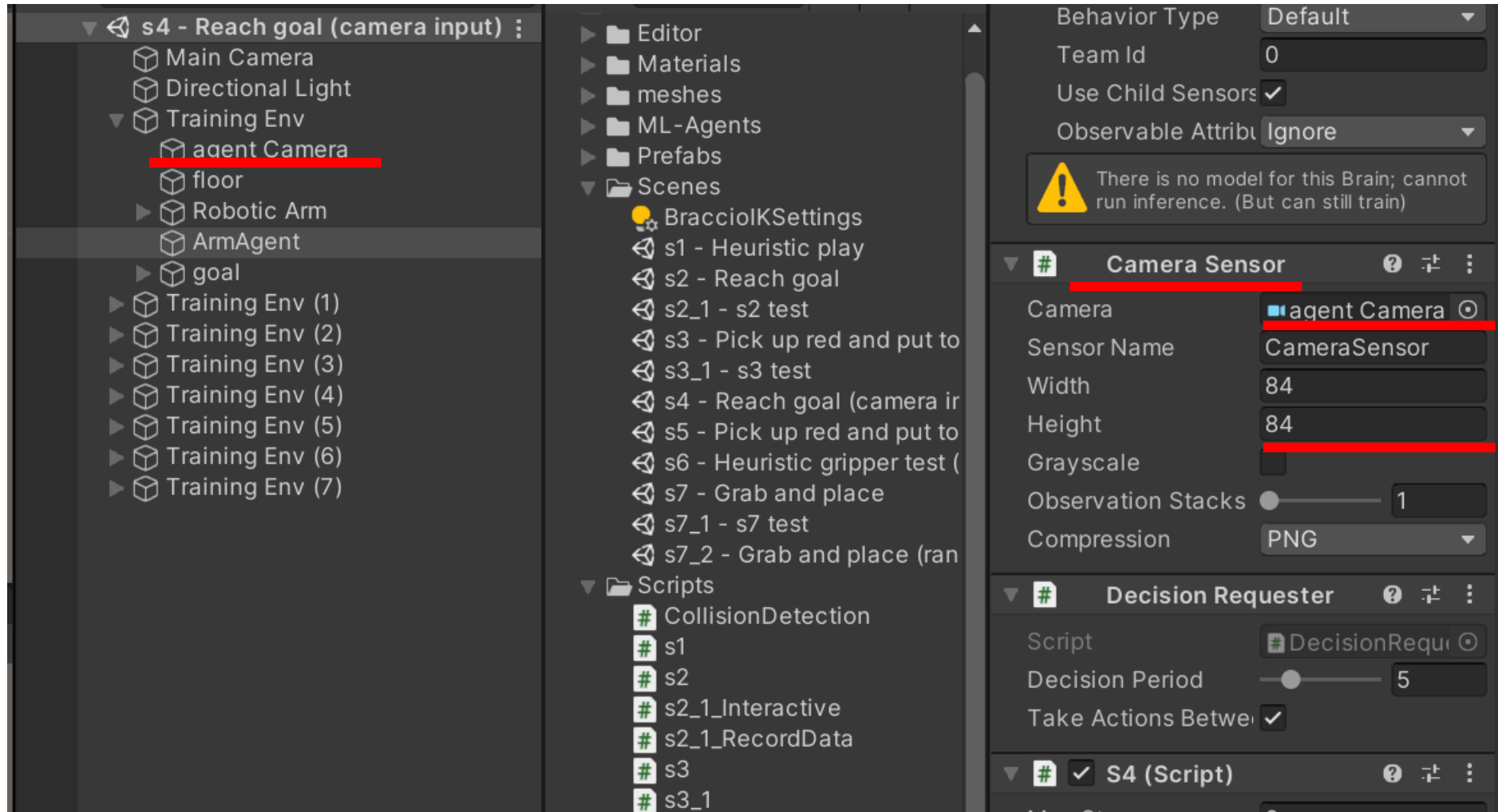
Goal initialize = randomly positioned in polar system $\theta = -80 \sim 80$, $r = 0.8 \sim 1.5$

Arm initialize: $(\theta_b = 0, \theta_u = 45, \theta_l = 45, \theta_w = 45)$

Add a top view camera to each training environment

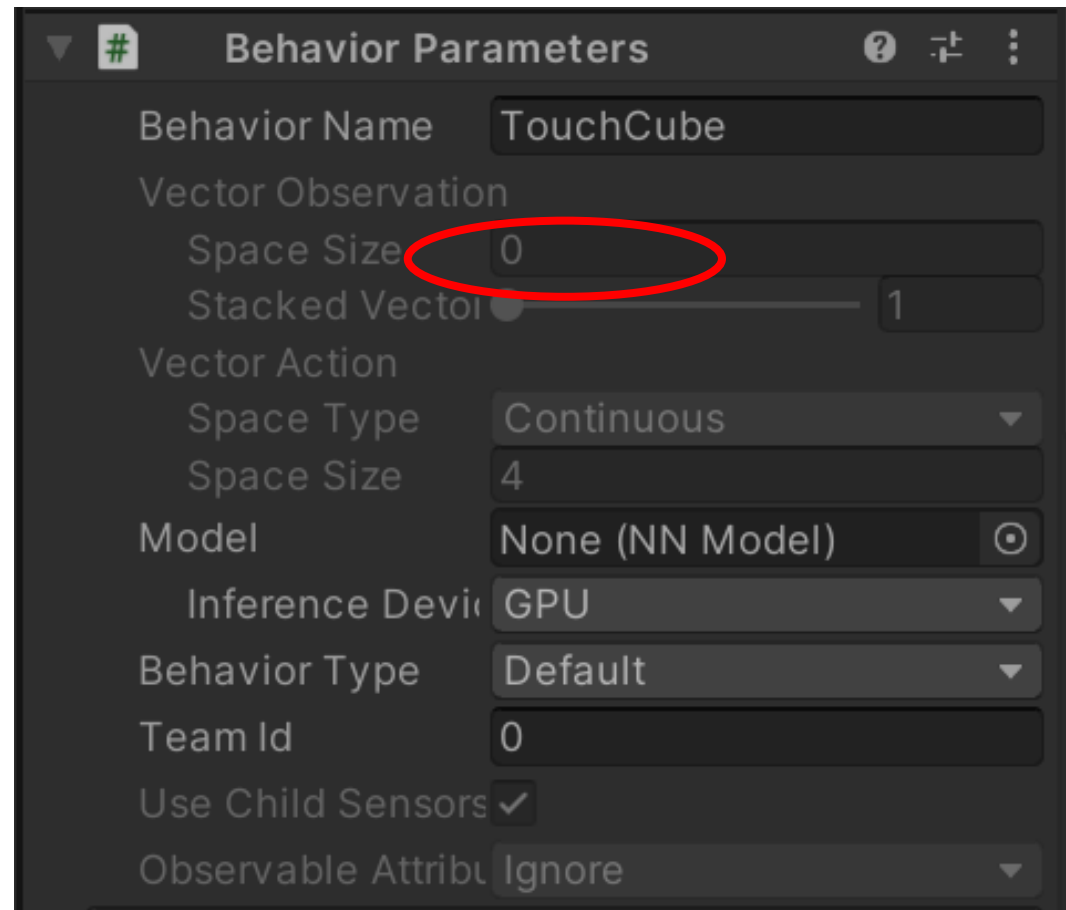


Add camera sensor to the robot agent



Vector observation = 0

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



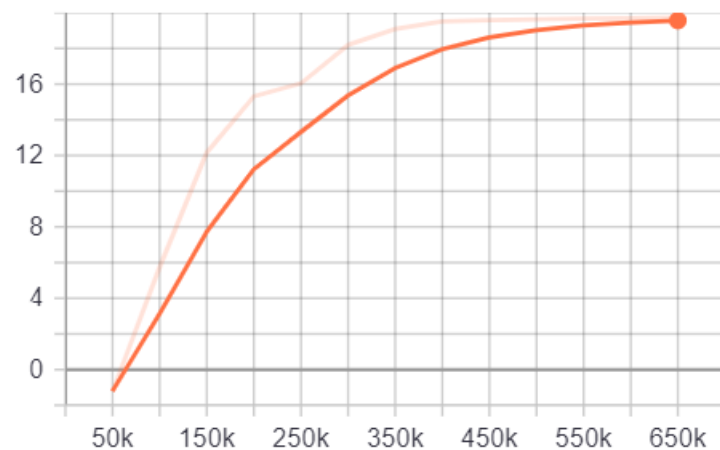
Train for 600K

```
TouchCube. Step: 50000. Time Elapsed: 222.493 s. Mean Reward: -1.205. Std of Reward: 12.602. Training.
TouchCube. Step: 100000. Time Elapsed: 417.921 s. Mean Reward: 5.743. Std of Reward: 13.921. Training.
TouchCube. Step: 150000. Time Elapsed: 618.958 s. Mean Reward: 12.166. Std of Reward: 11.773. Training.
TouchCube. Step: 200000. Time Elapsed: 819.051 s. Mean Reward: 15.312. Std of Reward: 9.549. Training.
TouchCube. Step: 250000. Time Elapsed: 1027.214 s. Mean Reward: 16.052. Std of Reward: 9.106. Training.
TouchCube. Step: 300000. Time Elapsed: 1230.887 s. Mean Reward: 18.197. Std of Reward: 5.736. Training.
TouchCube. Step: 350000. Time Elapsed: 1446.545 s. Mean Reward: 19.086. Std of Reward: 3.321. Training.
TouchCube. Step: 400000. Time Elapsed: 1662.651 s. Mean Reward: 19.512. Std of Reward: 1.555. Training.
TouchCube. Step: 450000. Time Elapsed: 1899.574 s. Mean Reward: 19.595. Std of Reward: 1.210. Training.
TouchCube. Step: 500000. Time Elapsed: 2154.839 s. Mean Reward: 19.635. Std of Reward: 0.976. Training.
zation.py:93] Converting to results\1\TouchCube\TouchCube-499992.onnx
ges\mlagents\trainers\torch\distributions.py:163: TracerWarning: Converting a tensor to a Python index might
n't record the data flow of Python values, so this value will be treated as a constant in the future. This
e to other inputs!
] * inputs.shape[0], axis=0)
ges\mlagents\trainers\torch\networks.py:352: TracerWarning: torch.Tensor results are registered as constant
is warning if you use this function to create tensors out of constant variables that would be the same eve
her case, this might cause the trace to be incorrect.
y_size]),
zation.py:105] Exported results\1\TouchCube\TouchCube-499992.onnx
TouchCube. Step: 550000. Time Elapsed: 2391.675 s. Mean Reward: 19.671. Std of Reward: 0.946. Training.
TouchCube. Step: 600000. Time Elapsed: 2647.336 s. Mean Reward: 19.707. Std of Reward: 0.785. Training.
```

Train for 600K

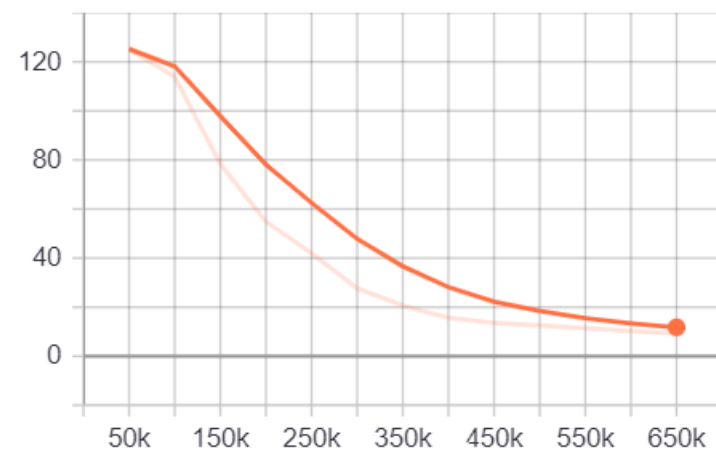
Cumulative Reward

tag: Environment/Cumulative Reward



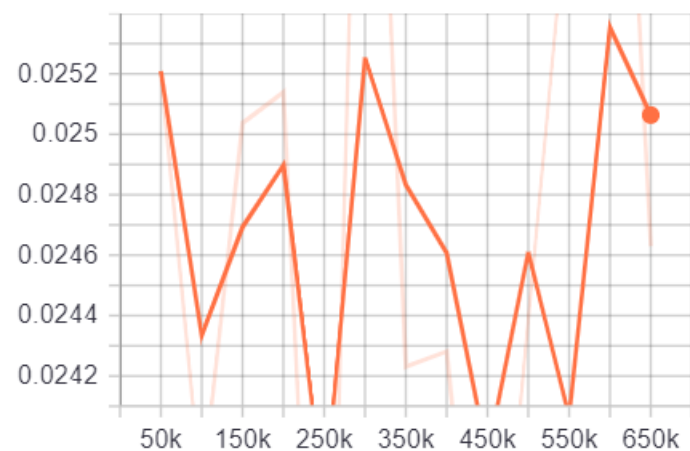
Episode Length

tag: Environment/Episode Length



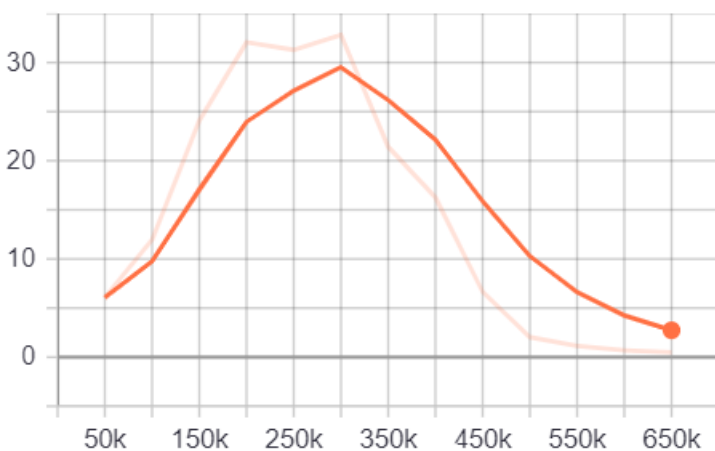
Policy Loss

tag: Losses/Policy Loss



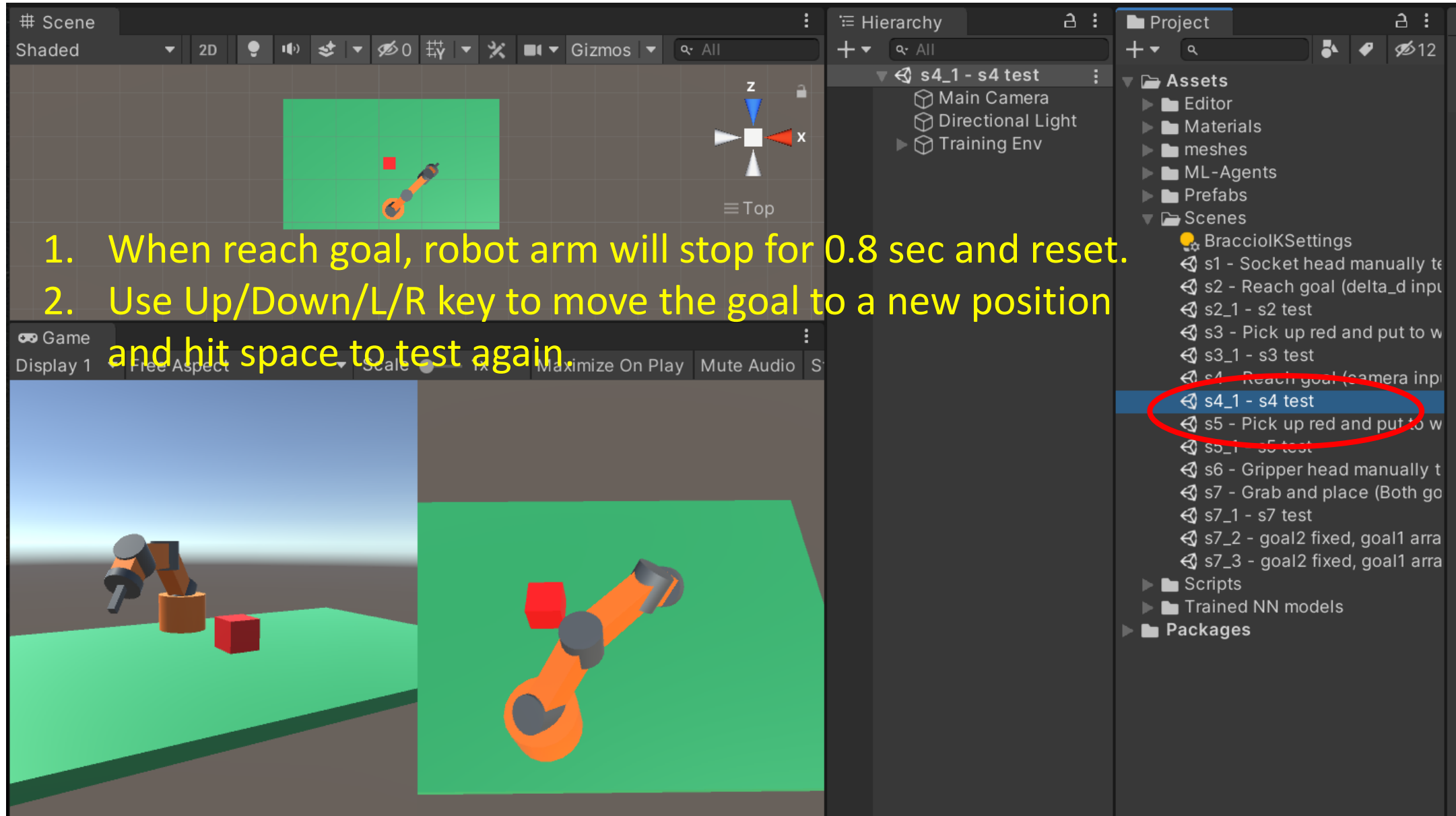
Value Loss

tag: Losses/Value Loss



Test – Play with robot arm

Open "s4_1"



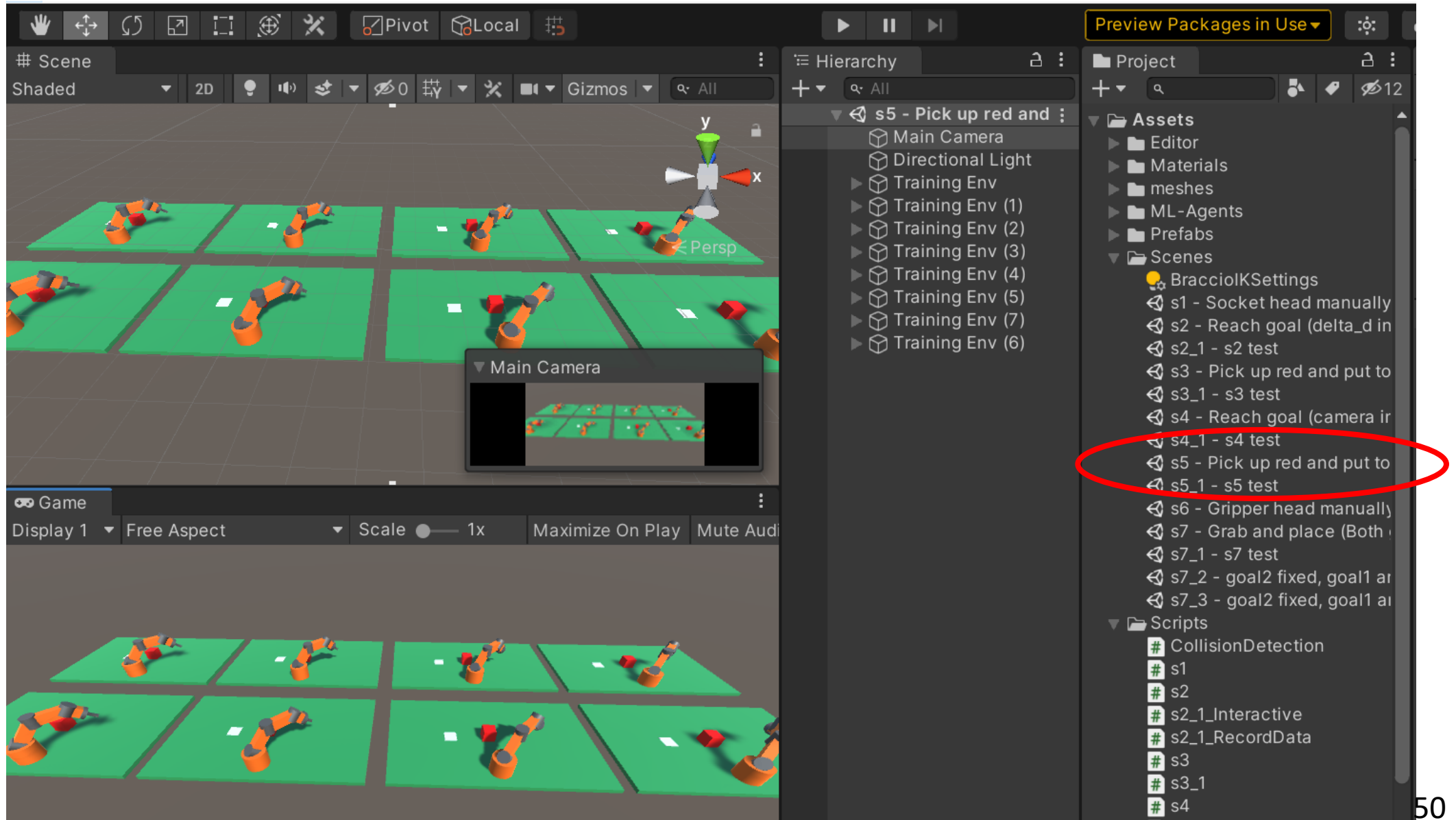
HW5(1)

- Describe training setting
- Show tensor board plots and discuss your training performance
- Describe test performance (recorded data, interactive test)

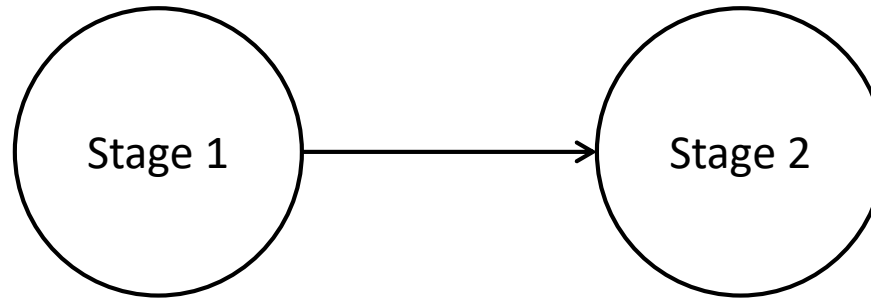


Scene s5 – Pick red cube and place it on top of the white cube using camera image

Open s5



Training setting



s = feature map vector from a CNN, size = 2592

Input image to the CNN is captured by a camera from top, size = 84x84x3

In my script, I use +15 for goal 1 and +100 for goal 2. But +20 for both goal 1 and goal 2 should be OK.

$$r = \begin{cases} -0.005 & \text{per step} \\ -5 & \text{collision, out of range} \\ +20 & d_{stage1} \leq 0.5, d_{stage2} \leq 0.5 \end{cases}$$

0.5 is easier to succeed. But the resulted behavior is unreal, like long-distance socket. Try 0.3 or 0.25 for your HW.

$$a = (\Delta\theta_B, \Delta\theta_U, \Delta\theta_L, \Delta\theta_W)$$

No. of training environment = 8

Goal initialize = randomly positioned in polar system $\theta = -80 \sim 80$, $r = 0.8 \sim 1.5$

Goal2 initialize = same as goal 1

Arm initialize: $(\theta_B = 0, \theta_U = 45, \theta_L = 45, \theta_W = 45)$

NN: ?-512-512-512-4

Time horizon = 2000

Buffer size = 20480

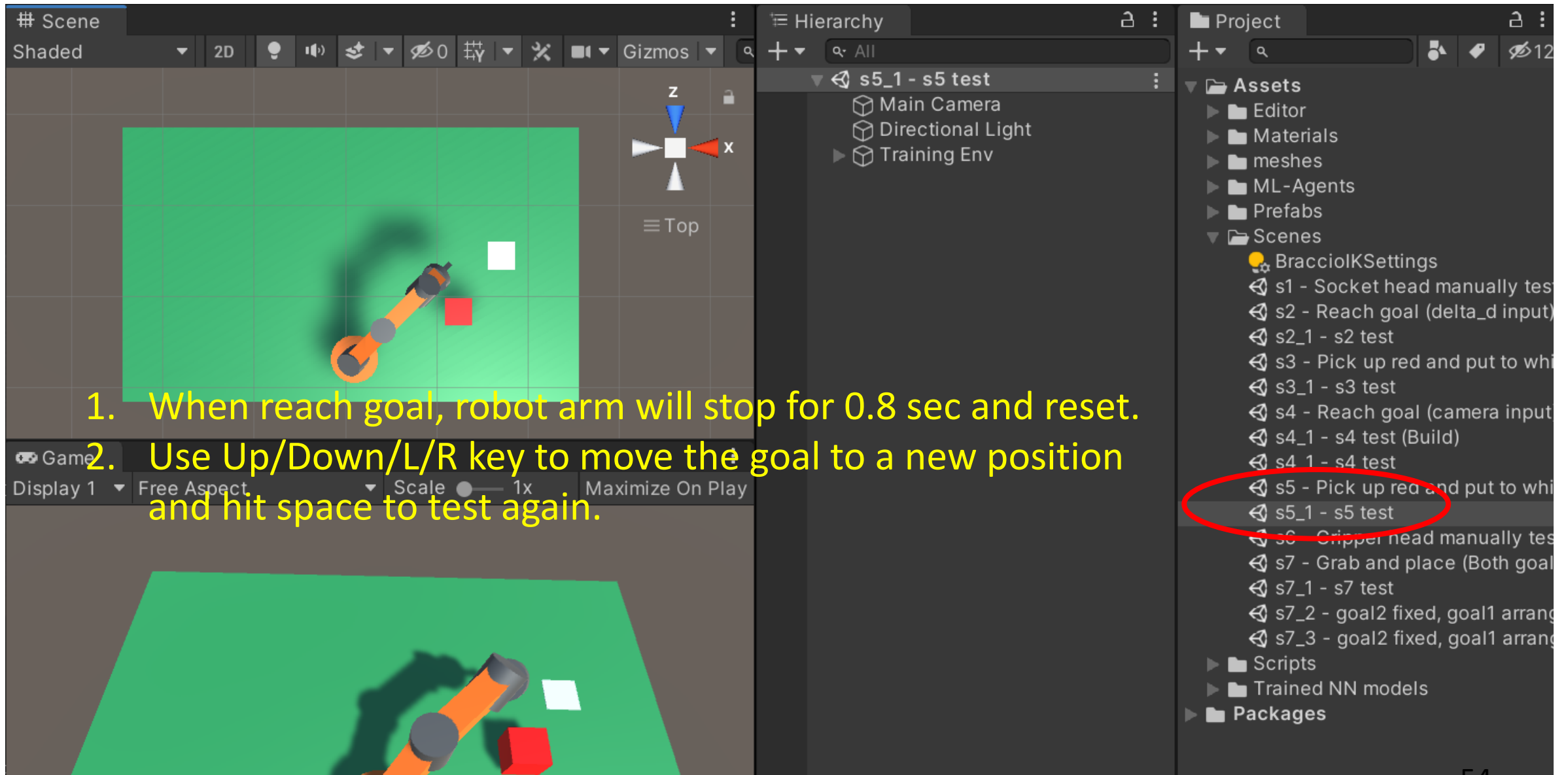
Batch size = 2048

Intermediate results after 2.4M

```
TouchCube. Step: 900000. Time Elapsed: 3622.903 s. Mean Reward: 42.232. Std of Reward: 51.025.
TouchCube. Step: 950000. Time Elapsed: 3817.672 s. Mean Reward: 43.436. Std of Reward: 52.013.
TouchCube. Step: 1000000. Time Elapsed: 4012.238 s. Mean Reward: 40.262. Std of Reward: 51.225.
ation.py:93] Converting to results\1\TouchCube\TouchCube-999986.onnx
ation.py:105] Exported results\1\TouchCube\TouchCube-999986.onnx
TouchCube. Step: 1050000. Time Elapsed: 4198.270 s. Mean Reward: 41.678. Std of Reward: 51.639.
TouchCube. Step: 1100000. Time Elapsed: 4396.988 s. Mean Reward: 43.416. Std of Reward: 51.913.
TouchCube. Step: 1150000. Time Elapsed: 4593.338 s. Mean Reward: 43.914. Std of Reward: 51.657.
TouchCube. Step: 1200000. Time Elapsed: 4792.435 s. Mean Reward: 40.471. Std of Reward: 50.521.
TouchCube. Step: 1250000. Time Elapsed: 4984.331 s. Mean Reward: 42.493. Std of Reward: 51.315.
TouchCube. Step: 1300000. Time Elapsed: 5197.395 s. Mean Reward: 43.941. Std of Reward: 51.623.
TouchCube. Step: 1350000. Time Elapsed: 5403.109 s. Mean Reward: 47.774. Std of Reward: 51.834.
TouchCube. Step: 1400000. Time Elapsed: 5608.913 s. Mean Reward: 47.696. Std of Reward: 52.174.
TouchCube. Step: 1450000. Time Elapsed: 5836.131 s. Mean Reward: 45.999. Std of Reward: 51.536.
TouchCube. Step: 1500000. Time Elapsed: 6037.497 s. Mean Reward: 45.637. Std of Reward: 51.625.
ation.py:93] Converting to results\1\TouchCube\TouchCube-1499845.onnx
ation.py:105] Exported results\1\TouchCube\TouchCube-1499845.onnx
TouchCube. Step: 1550000. Time Elapsed: 6243.114 s. Mean Reward: 46.052. Std of Reward: 52.346.
TouchCube. Step: 1600000. Time Elapsed: 6435.717 s. Mean Reward: 48.292. Std of Reward: 52.147.
TouchCube. Step: 1650000. Time Elapsed: 6634.507 s. Mean Reward: 45.468. Std of Reward: 51.541.
TouchCube. Step: 1700000. Time Elapsed: 6835.629 s. Mean Reward: 49.736. Std of Reward: 51.866.
TouchCube. Step: 1750000. Time Elapsed: 7030.518 s. Mean Reward: 50.424. Std of Reward: 51.975.
TouchCube. Step: 1800000. Time Elapsed: 7232.622 s. Mean Reward: 52.385. Std of Reward: 52.520.
TouchCube. Step: 1850000. Time Elapsed: 7427.402 s. Mean Reward: 53.477. Std of Reward: 52.988.
TouchCube. Step: 1900000. Time Elapsed: 7624.994 s. Mean Reward: 49.200. Std of Reward: 52.777.
TouchCube. Step: 1950000. Time Elapsed: 7820.621 s. Mean Reward: 51.300. Std of Reward: 52.514.
TouchCube. Step: 2000000. Time Elapsed: 8031.034 s. Mean Reward: 51.976. Std of Reward: 52.098.
ation.py:93] Converting to results\1\TouchCube\TouchCube-1999899.onnx
ation.py:105] Exported results\1\TouchCube\TouchCube-1999899.onnx
TouchCube. Step: 2050000. Time Elapsed: 8223.305 s. Mean Reward: 51.876. Std of Reward: 52.860.
TouchCube. Step: 2100000. Time Elapsed: 8423.297 s. Mean Reward: 49.998. Std of Reward: 52.224.
TouchCube. Step: 2150000. Time Elapsed: 8609.655 s. Mean Reward: 51.952. Std of Reward: 52.659.
TouchCube. Step: 2200000. Time Elapsed: 8819.758 s. Mean Reward: 54.323. Std of Reward: 53.211.
TouchCube. Step: 2250000. Time Elapsed: 9014.827 s. Mean Reward: 48.848. Std of Reward: 52.707.
TouchCube. Step: 2300000. Time Elapsed: 9210.316 s. Mean Reward: 52.215. Std of Reward: 52.870.
TouchCube. Step: 2350000. Time Elapsed: 9401.985 s. Mean Reward: 49.891. Std of Reward: 52.837.
TouchCube. Step: 2400000. Time Elapsed: 9609.943 s. Mean Reward: 49.739. Std of Reward: 52.804.
```

Test – Play with robot arm

Open "s5_1"



HW5(2)

- Describe training setting
- Show tensor board plots and discuss your training performance
- Describe test performance (recorded data, interactive test)

