(1) Reach goal

Training setting

$$s = (\Delta x, \Delta y, \Delta z, \theta_{\mathrm{B}}, \theta_{\mathrm{U}}, \theta_{\mathrm{L}}, \theta_{\mathrm{W}})$$

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

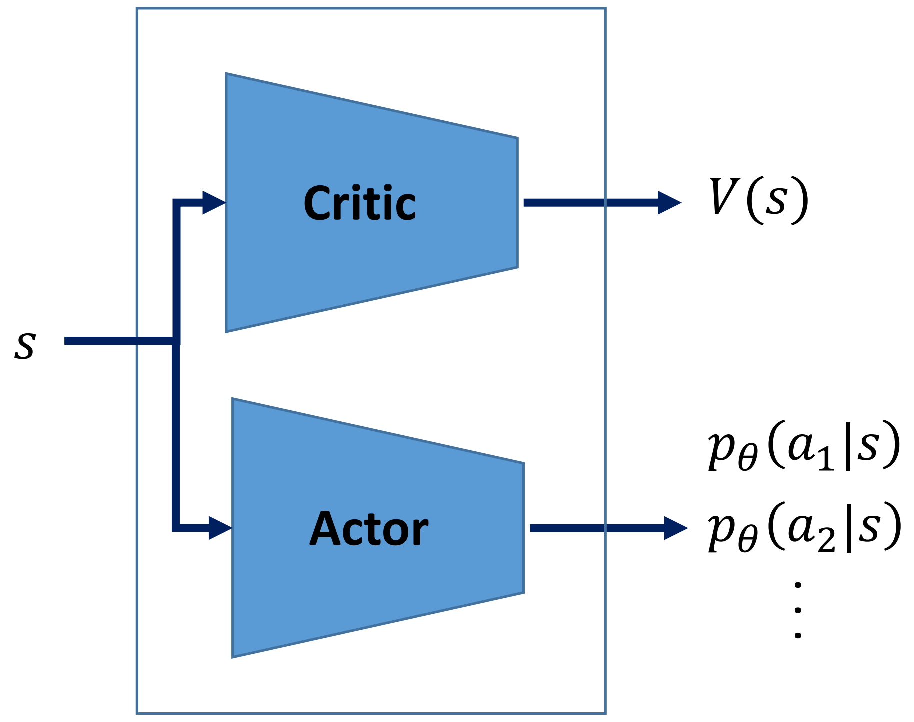
$$r = \begin{cases} -0.005 & \text{per step} \\ -5 & \text{collision, out of range} \\ +20 & \text{goal, } d \leq 0.5 \end{cases}$$

No. of training environment = 9

Goal initialize = randomly positioned in polar system θ=-80~80, r=0.8~1.5

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

Summary of training



NN: 7-512-512-512-4

Time horizon = 2000

Buffer size = 20480

Batch size = 2048

$$\tau = \left(s_1, a_1, r_1, s_2, a_2, r_{2,} \cdots s_T, a_{T=2000}\right)$$

$$\nabla \bar{R}_\theta \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n}\left(\boxed{\sum_{t'}^{T_n} \gamma^{t'-t} r_{t'}^{n}} - \boxed{b}\right)\nabla \log p_\theta(a_t^n|s_t^n)$$

$\boxed{V^{\pi_\theta}(s_t^i)}$ Expected value of b

$\boxed{E[G_t^i] = Q^{\pi_\theta}\left(s_t^i, a_t^i\right)}$ Expected value of G_t^i

$$A^\theta(s_t, a_t) = \left(r_t^n + V^{\pi_\theta}(s_{t+1}^n) - V^{\pi_\theta}(s_t^n)\right)$$

$$L_v = (G_t^n - V^{\pi_\theta}(s_t^n))^2 = \left(r_t^n + \gamma V^{\pi_\theta}(s_{t+1}^n) - V^{\pi_\theta}(s_t^n)\right)^2$$

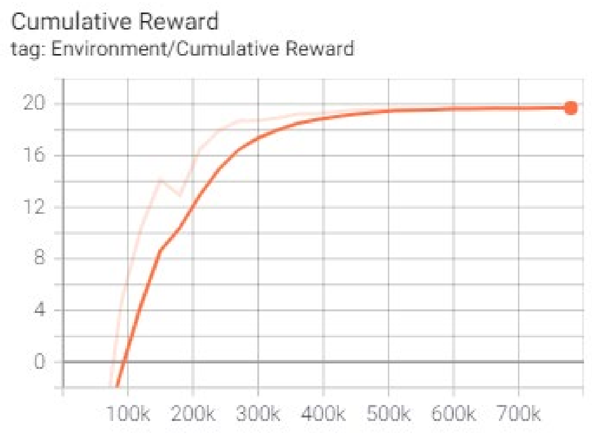
$$L_\pi = \sum_{(s_t,a_t)} min\left(\frac{p_\theta(a_t|s_t)}{p_{\theta'}(a_t|s_t)} A^{\theta'}(s_t, a_t), clip\left(\frac{p_\theta(a_t|s_t)}{p_{\theta'}(a_t|s_t)}, 1-\varepsilon, 1+\varepsilon\right) A^{\theta'}(s_t, a_t)\right)$$

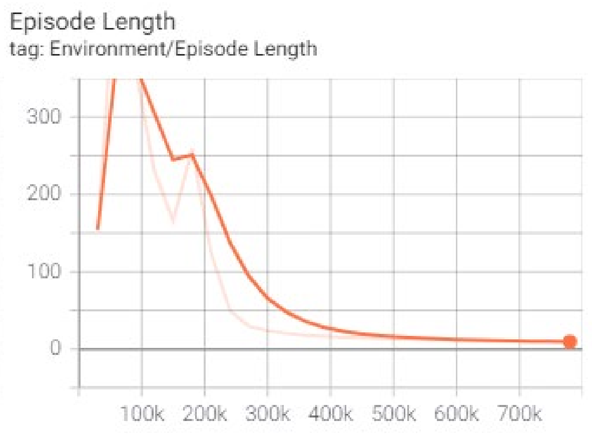
$$L = L_\pi + c_v L_v + c_{reg} L_{reg}$$

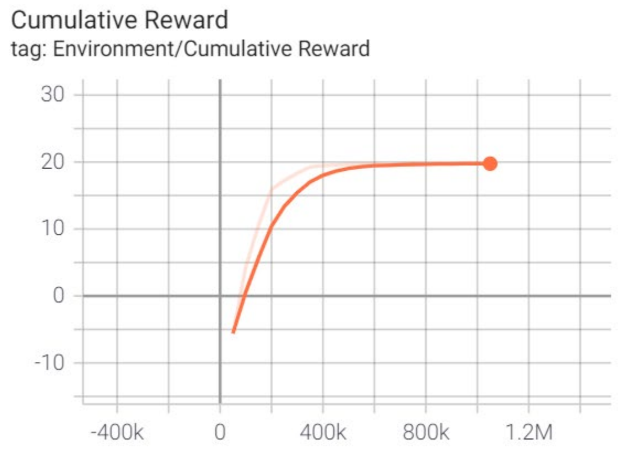
HW4(1)

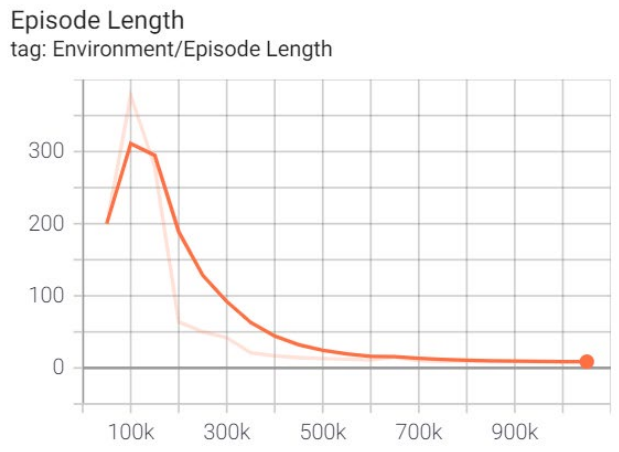
- Describe the training setting
- Show tensor board plots and discuss your training performance
- Describe your test performance

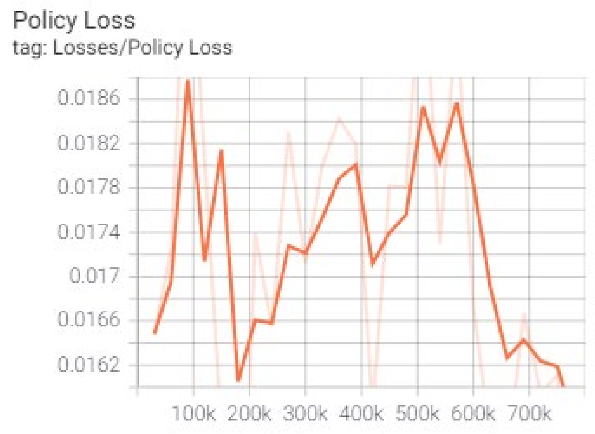
Typical successful training experiences, goal reach $d \leq 0.25$

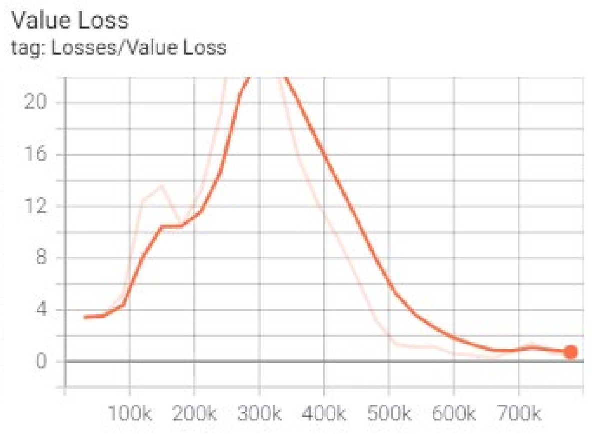




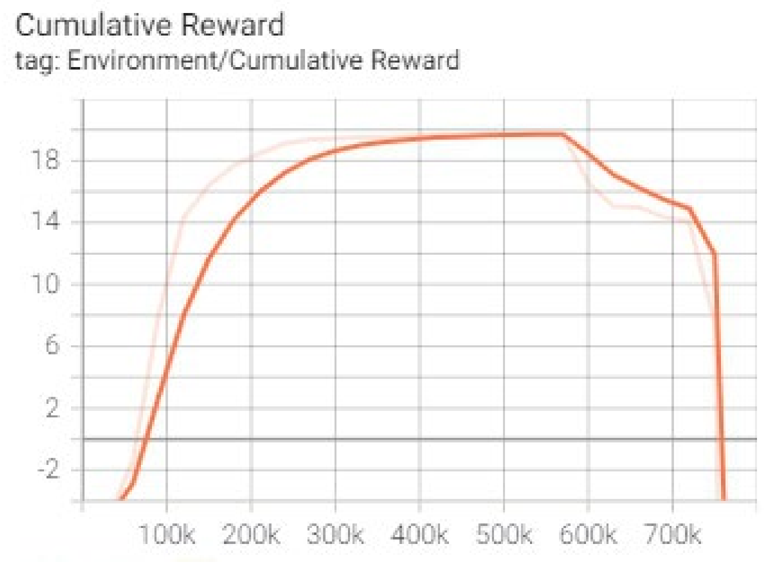


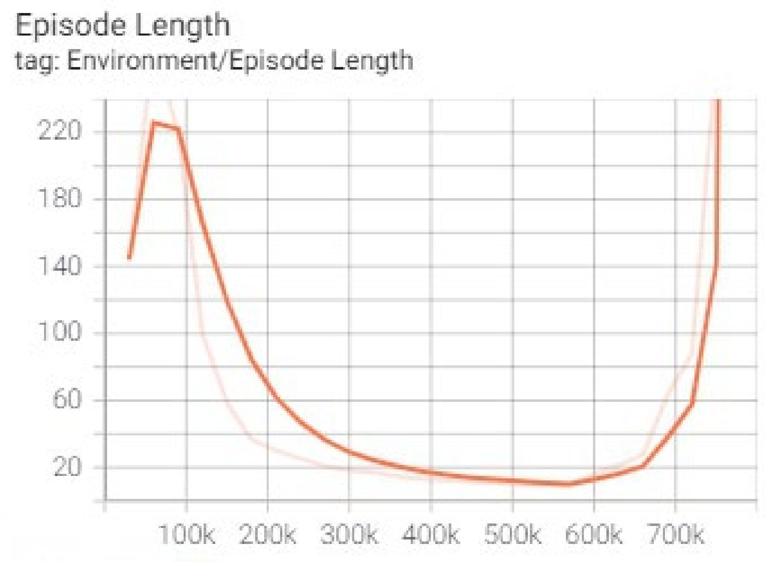


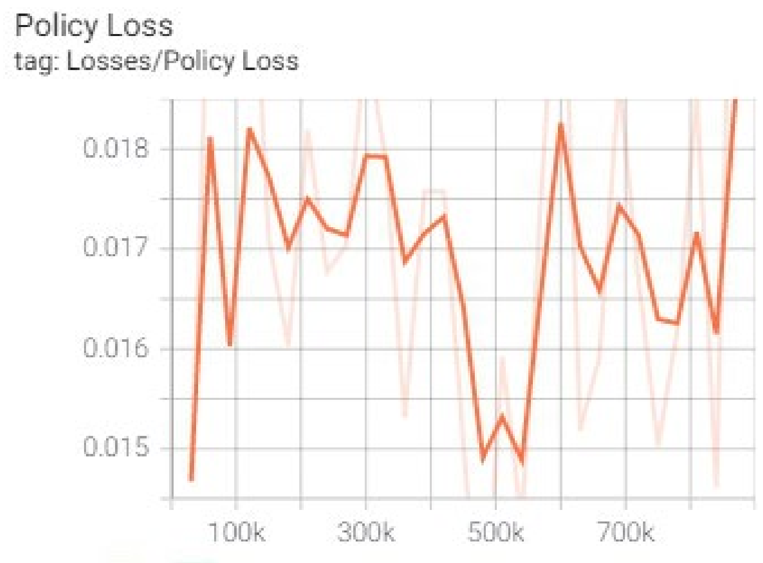


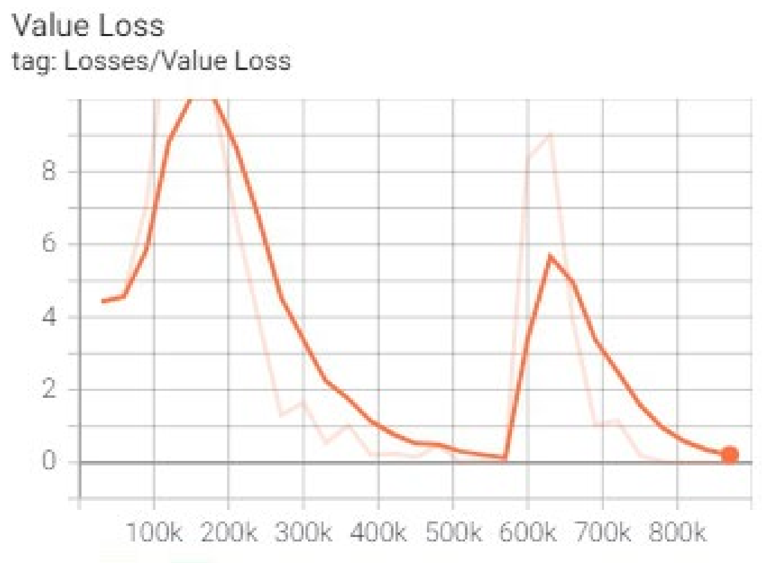


Failed training, goal reach $d \le 0.25$



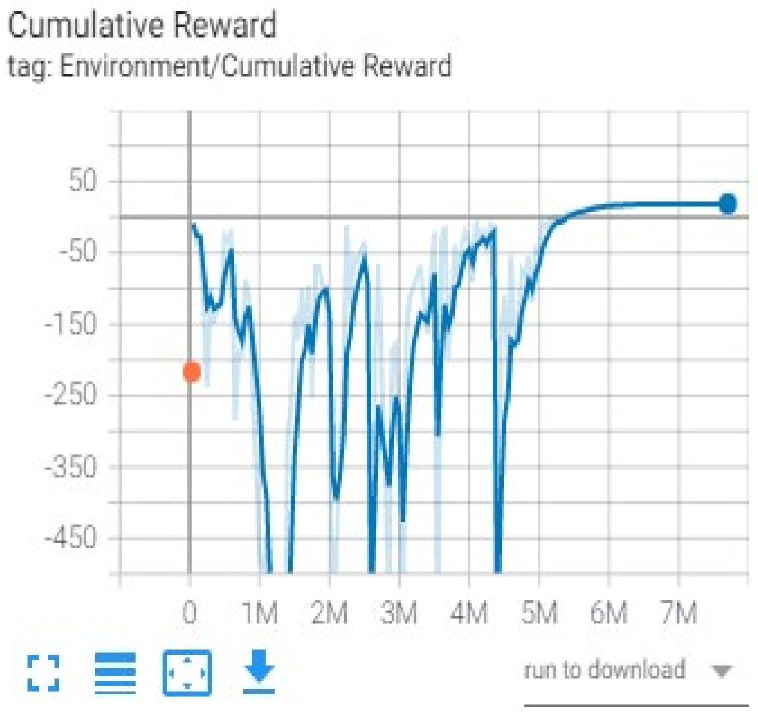


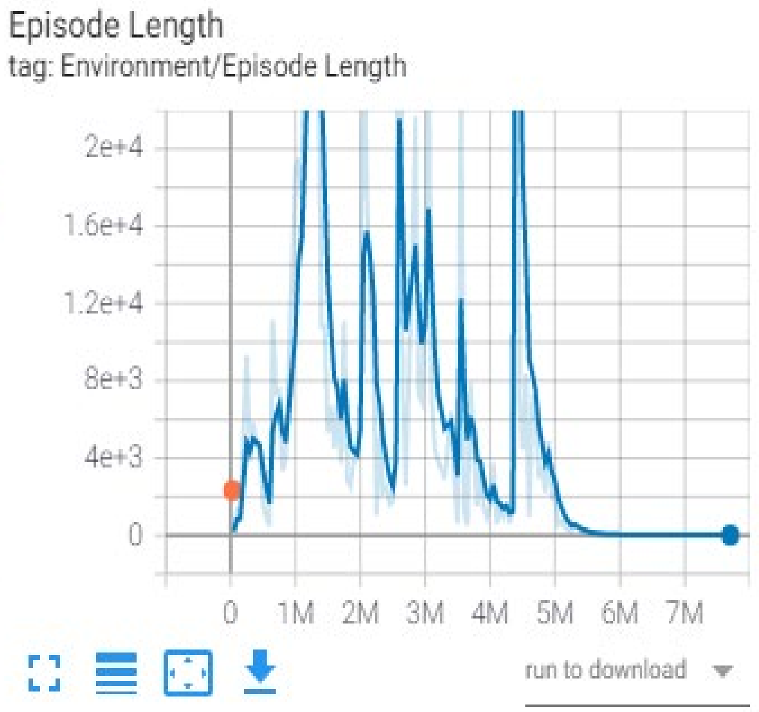


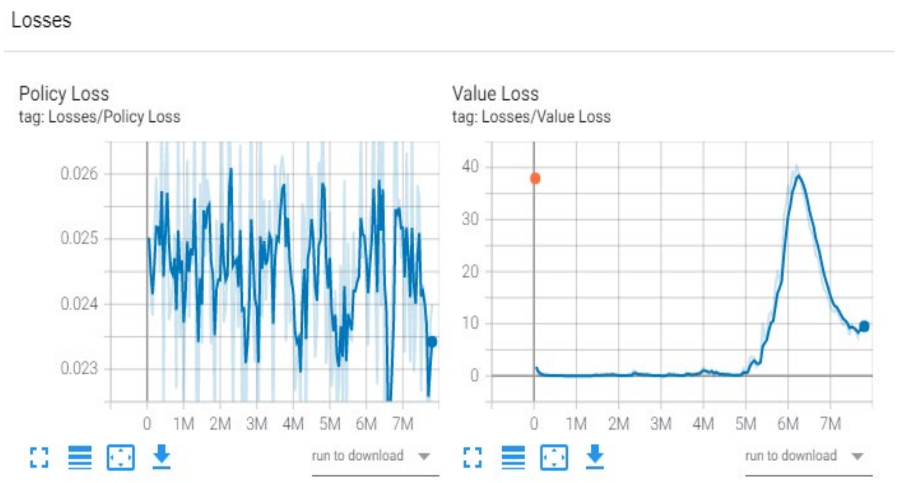


Successful training, goal reach $d \leq 0.1$

In this training, the point reach threshold is set to be 0.1, which makes it very difficult for AI to learn to reach goal. For the first 5M, the reward performance looks un-hope and most people will give up. But luckily the reward goes to maximum and become steady after 6M.



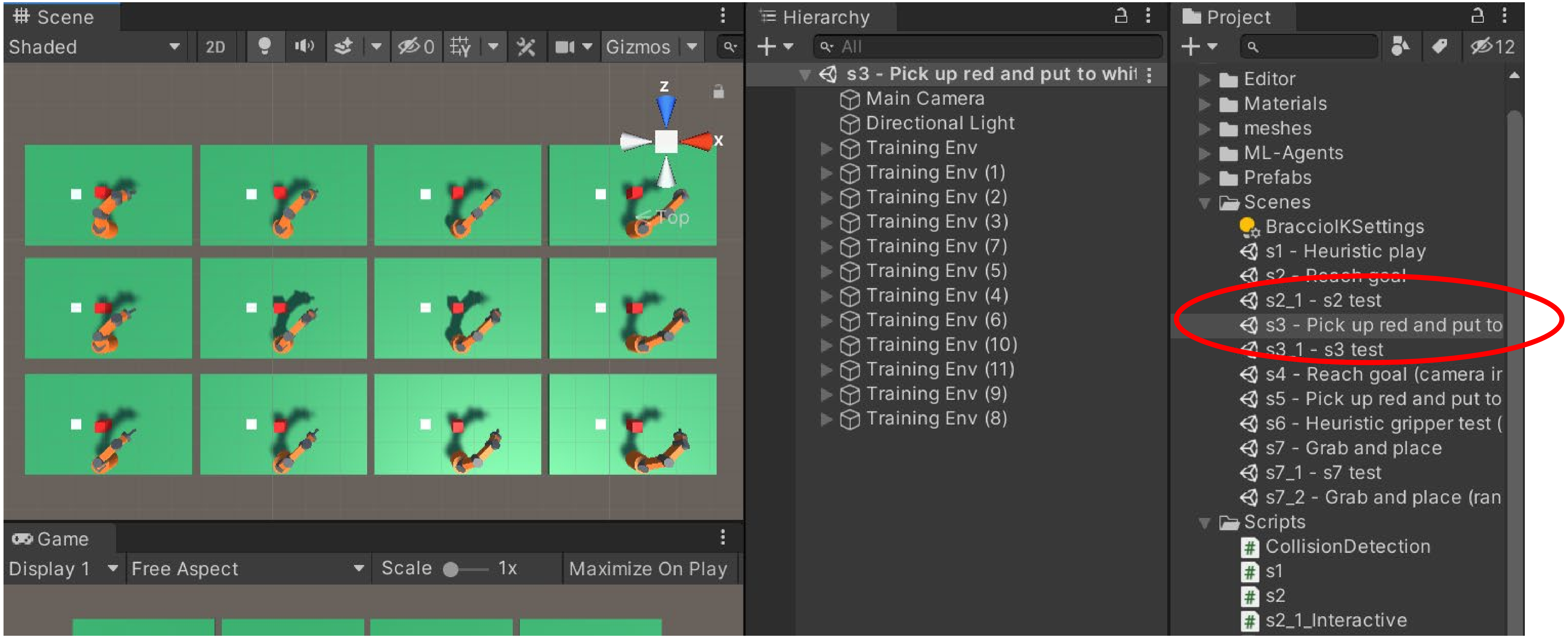




1061259 Keren

(2) Pick up goal and place it to white area

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



Hierarchy:

- s3 - Pick up red and put to whit
  - Main Camera
  - Directional Light
  - Training Env
  - Training Env (1)
  - Training Env (2)
  - Training Env (3)
  - Training Env (7)
  - Training Env (5)
  - Training Env (4)
  - Training Env (6)
  - Training Env (10)
  - Training Env (11)
  - Training Env (9)
  - Training Env (8)

Project:

- Editor
- Materials
- meshes
- ML-Agents
- Prefabs
- Scenes
  - BraccioIKSettings
  - s1 - Heuristic play
  - s2 - Reach goal
  - s2_1 - s2 test
  - s3 - Pick up red and put to
  - s3_1 - s3 test
  - s4 - Reach goal (camera ir
  - s5 - Pick up red and put to
  - s6 - Heuristic gripper test (
  - s7 - Grab and place
  - s7_1 - s7 test
  - s7_2 - Grab and place (ran
- Scripts
  - CollisionDetection
  - s1
  - s2
  - s2_1_Interactive

Training setting

$$\begin{tikzcd}\boxed{\text{Stage 1}} \arrow[r] & \boxed{\text{Stage 2}}\end{tikzcd}$$

$s = (sNo, \Delta_1 x, \Delta_1 y, \Delta_1 z, \theta_{\mathrm{B}}, \theta_{\mathrm{U}}, \theta_{\mathrm{L}}, \theta_{\mathrm{W}})$

$s = (sNo, \Delta_2 x, \Delta_2 y, \Delta_2 z, \theta_{\mathrm{B}}, \theta_{\mathrm{U}}, \theta_{\mathrm{L}}, \theta_{\mathrm{W}})$

$$r = \begin{cases} -0.005 & \\ -5 & \\ +20 & d_1 \leq 0.5 \end{cases}$$

$$r = \begin{cases} -0.005 & \text{per step} \\ -5 & \text{collision, out of range} \\ +20 & \text{goal}, d_2 \leq 0.5 \end{cases}$$

NN: 8-512-512-512-4

Time horizon = 2000

Buffer size = 20480

Batch size = 2048

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

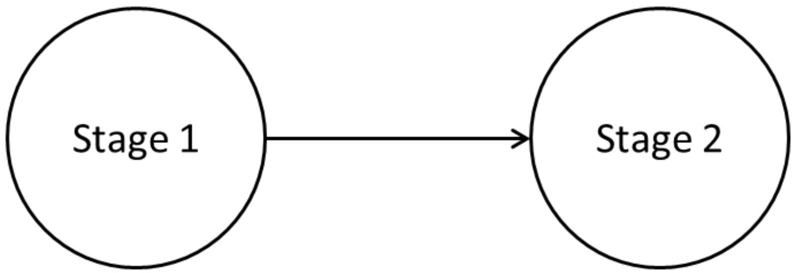
No. of training environment = 9

Goal initialize = randomly positioned in polar system θ=-80~80, r=0.8~1.5

Goal2 initialize = same as goal 1

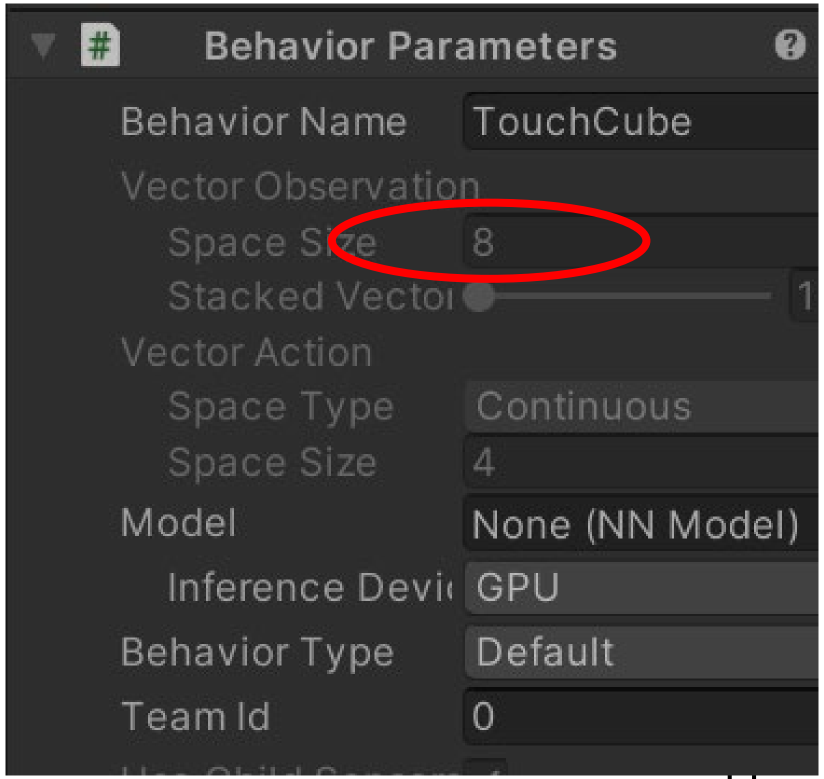
Arm initialize: $(\theta_B = 0, \theta_U = 45, \theta_L = 45, \theta_W = 45)$

State



$s = (sNo, \Delta_1 x, \Delta_1 y, \Delta_1 z, \theta_B, \theta_U, \theta_L, \theta_W)$ $s = (sNo, \Delta_2 x, \Delta_2 y, \Delta_2 z, \theta_B, \theta_U, \theta_L, \theta_W)$

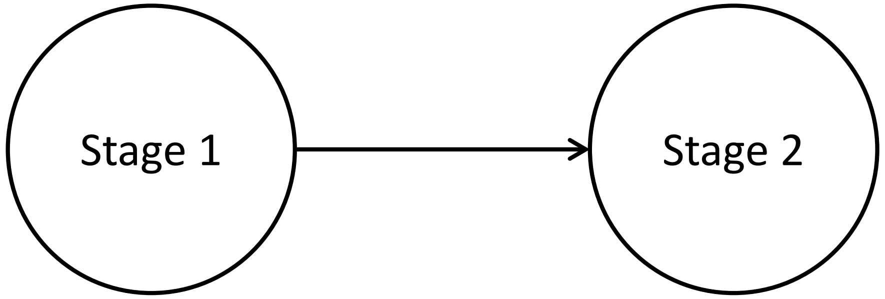
```
sensor.AddObservation(stage);

if (stage == 1)
    sensor.AddObservation(EndTouchPlane.position - goalUpTouchPt.posi
else //stage =2
    sensor.AddObservation(goalDownTouchPt.position - goal2UpTouchPt.p

float BaseRotationAngle = UnityEditor.TransformUtils.GetInspectorRota
float UArmRotationAngle = UnityEditor.TransformUtils.GetInspectorRota
float LArmRotationAngle = UnityEditor.TransformUtils.GetInspectorRota
float WRotationAngle = UnityEditor.TransformUtils.GetInspectorRotatio

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



Behavior Parameters

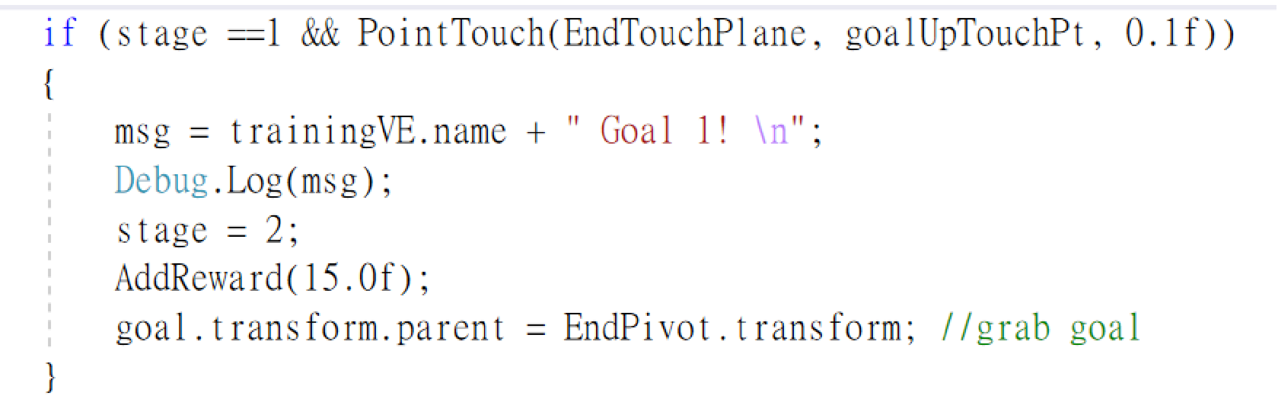
Behavior Name	TouchCube
Vector Observation	
Space Size	8
Stacked Vector	1
Vector Action	
Space Type	Continuous
Space Size	4
Model	None (NN Model)
Inference Devi	GPU
Behavior Type	Default
Team Id	0

Rewards

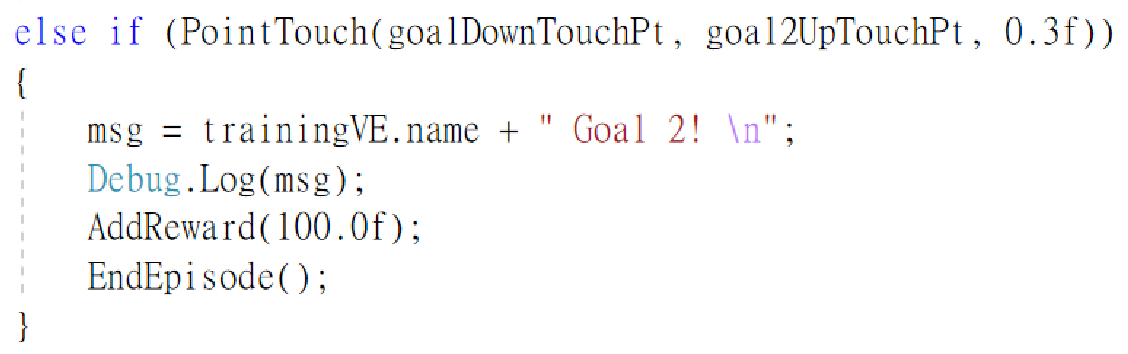


$$r = \begin{cases} -0.005 & \\ -5 & \\ +20 & d_1 \leq 0.5 \end{cases}$$

$$r = \begin{cases} -0.005 & \text{per step} \\ -5 & \text{collision, out of range} \\ +20 & \text{goal,} d_2 \leq 0.5 \end{cases}$$

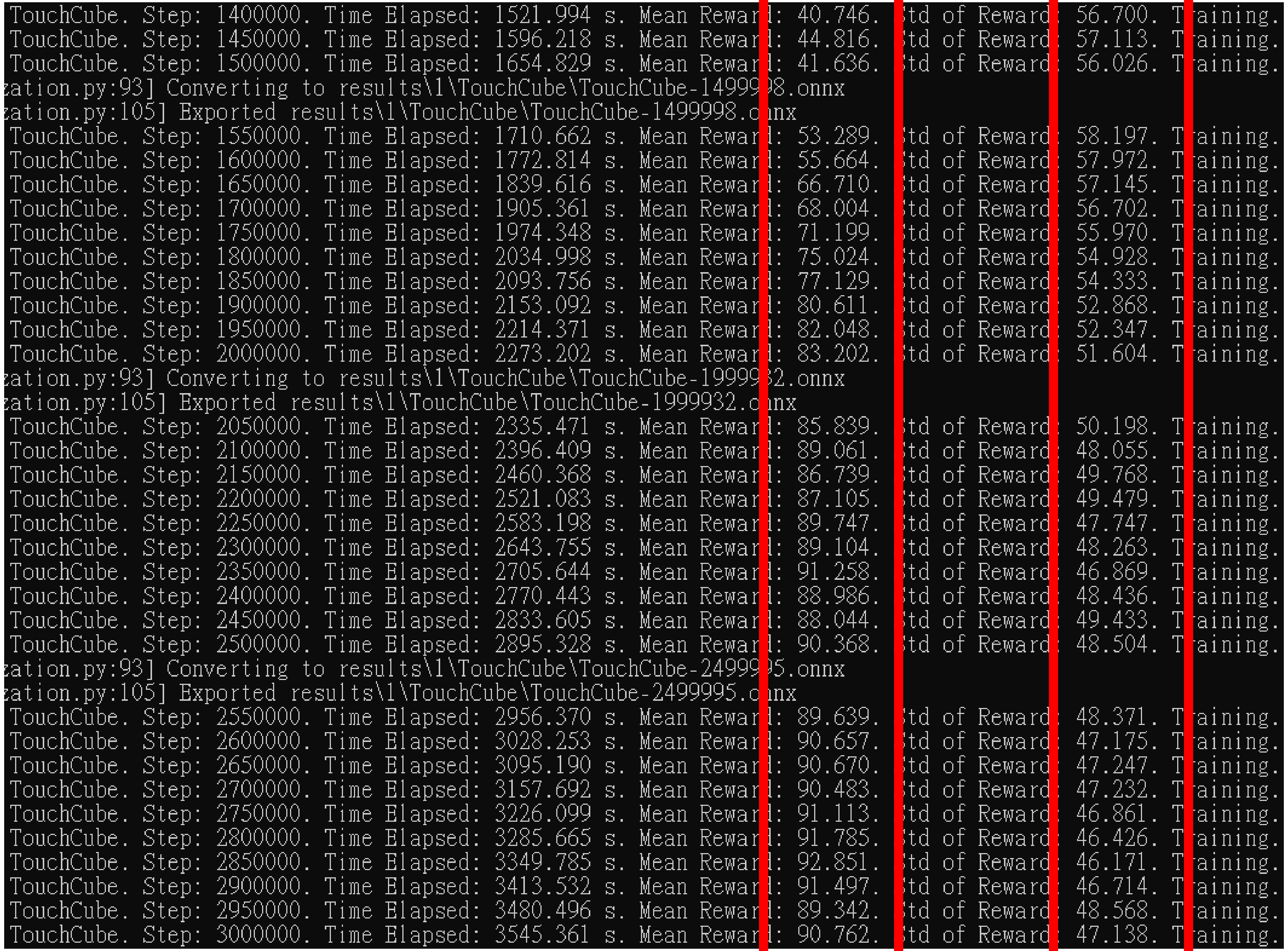


```
if (stage ==1 && PointTouch(EndTouchPlane, goalUpTouchPt, 0.1f))
{
    msg = trainingVE.name + " Goal 1! \n";
    Debug.Log(msg);
    stage = 2;
    AddReward(15.0f);
    goal.transform.parent = EndPivot.transform; //grab goal
}
```



```
else if (PointTouch(goalDownTouchPt, goal2UpTouchPt, 0.3f))
{
    msg = trainingVE.name + " Goal 2! \n";
    Debug.Log(msg);
    AddReward(100.0f);
    EndEpisode();
}
```

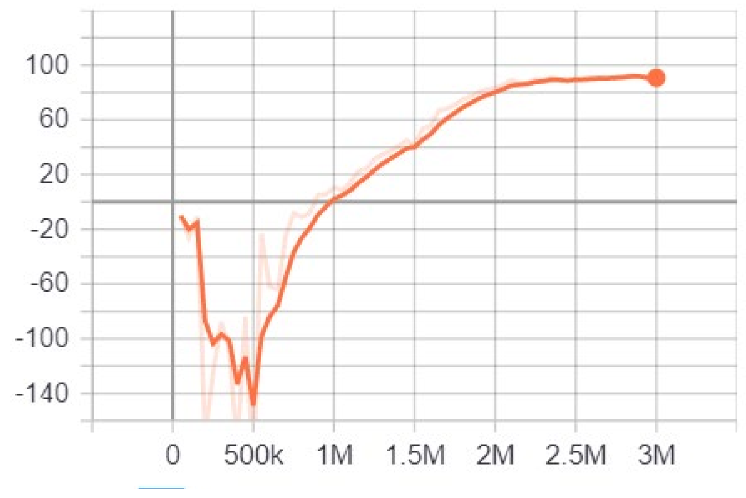
I quit at 3M, looks promising



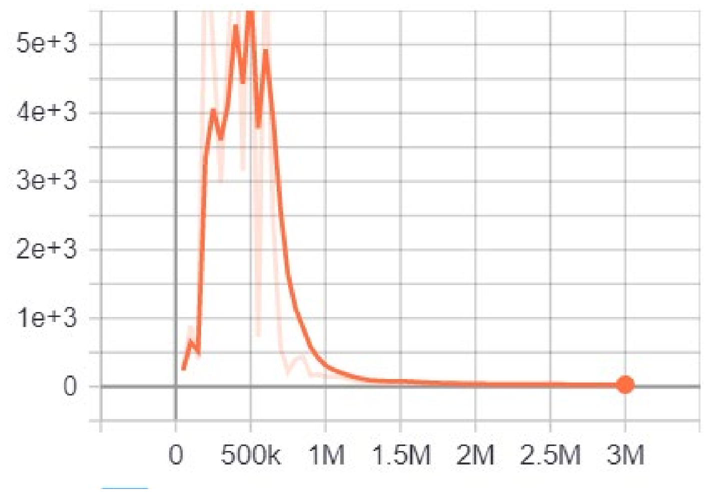
```
 TouchCube. Step: 1400000. Time Elapsed: 1521.994 s. Mean Rewar : 40.746.  td of Reward  56.700. T aining.
 TouchCube. Step: 1450000. Time Elapsed: 1596.218 s. Mean Rewar : 44.816.  td of Reward  57.113. T aining.
 TouchCube. Step: 1500000. Time Elapsed: 1654.829 s. Mean Rewar : 41.636.  td of Reward  56.026. T aining.
zation.py:93] Converting to results\1\TouchCube\TouchCube-1499 8.onnx
zation.py:105] Exported results\1\TouchCube\TouchCube-1499998.o nx
 TouchCube. Step: 1550000. Time Elapsed: 1710.662 s. Mean Rewar : 53.289.  td of Reward  58.197. T aining.
 TouchCube. Step: 1600000. Time Elapsed: 1772.814 s. Mean Rewar : 55.664.  td of Reward  57.972. T aining.
 TouchCube. Step: 1650000. Time Elapsed: 1839.616 s. Mean Rewar : 66.710.  td of Reward  57.145. T aining.
 TouchCube. Step: 1700000. Time Elapsed: 1905.361 s. Mean Rewar : 68.004.  td of Reward  56.702. T aining.
 TouchCube. Step: 1750000. Time Elapsed: 1974.348 s. Mean Rewar : 71.199.  td of Reward  55.970. T aining.
 TouchCube. Step: 1800000. Time Elapsed: 2034.998 s. Mean Rewar : 75.024.  td of Reward  54.928. T aining.
 TouchCube. Step: 1850000. Time Elapsed: 2093.756 s. Mean Rewar : 77.129.  td of Reward  54.333. T aining.
 TouchCube. Step: 1900000. Time Elapsed: 2153.092 s. Mean Rewar : 80.611.  td of Reward  52.868. T aining.
 TouchCube. Step: 1950000. Time Elapsed: 2214.371 s. Mean Rewar : 82.048.  td of Reward  52.347. T aining.
 TouchCube. Step: 2000000. Time Elapsed: 2273.202 s. Mean Rewar : 83.202.  td of Reward  51.604. T aining.
zation.py:93] Converting to results\1\TouchCube\TouchCube-1999 2.onnx
zation.py:105] Exported results\1\TouchCube\TouchCube-1999932.o nx
 TouchCube. Step: 2050000. Time Elapsed: 2335.471 s. Mean Rewar : 85.839.  td of Reward  50.198. T aining.
 TouchCube. Step: 2100000. Time Elapsed: 2396.409 s. Mean Rewar : 89.061.  td of Reward  48.055. T aining.
 TouchCube. Step: 2150000. Time Elapsed: 2460.368 s. Mean Rewar : 86.739.  td of Reward  49.768. T aining.
 TouchCube. Step: 2200000. Time Elapsed: 2521.083 s. Mean Rewar : 87.105.  td of Reward  49.479. T aining.
 TouchCube. Step: 2250000. Time Elapsed: 2583.198 s. Mean Rewar : 89.747.  td of Reward  47.747. T aining.
 TouchCube. Step: 2300000. Time Elapsed: 2643.755 s. Mean Rewar : 89.104.  td of Reward  48.263. T aining.
 TouchCube. Step: 2350000. Time Elapsed: 2705.644 s. Mean Rewar : 91.258.  td of Reward  46.869. T aining.
 TouchCube. Step: 2400000. Time Elapsed: 2770.443 s. Mean Rewar : 88.986.  td of Reward  48.436. T aining.
 TouchCube. Step: 2450000. Time Elapsed: 2833.605 s. Mean Rewar : 88.044.  td of Reward  49.433. T aining.
 TouchCube. Step: 2500000. Time Elapsed: 2895.328 s. Mean Rewar : 90.368.  td of Reward  48.504. T aining.
zation.py:93] Converting to results\1\TouchCube\TouchCube-2499 5.onnx
zation.py:105] Exported results\1\TouchCube\TouchCube-2499995.o nx
 TouchCube. Step: 2550000. Time Elapsed: 2956.370 s. Mean Rewar : 89.639.  td of Reward  48.371. T aining.
 TouchCube. Step: 2600000. Time Elapsed: 3028.253 s. Mean Rewar : 90.657.  td of Reward  47.175. T aining.
 TouchCube. Step: 2650000. Time Elapsed: 3095.190 s. Mean Rewar : 90.670.  td of Reward  47.247. T aining.
 TouchCube. Step: 2700000. Time Elapsed: 3157.692 s. Mean Rewar : 90.483.  td of Reward  47.232. T aining.
 TouchCube. Step: 2750000. Time Elapsed: 3226.099 s. Mean Rewar : 91.113.  td of Reward  46.861. T aining.
 TouchCube. Step: 2800000. Time Elapsed: 3285.665 s. Mean Rewar : 91.785.  td of Reward  46.426. T aining.
 TouchCube. Step: 2850000. Time Elapsed: 3349.785 s. Mean Rewar : 92.851.  td of Reward  46.171. T aining.
 TouchCube. Step: 2900000. Time Elapsed: 3413.532 s. Mean Rewar : 91.497.  td of Reward  46.714. T aining.
 TouchCube. Step: 2950000. Time Elapsed: 3480.496 s. Mean Rewar : 89.342.  td of Reward  48.568. T aining.
 TouchCube. Step: 3000000. Time Elapsed: 3545.361 s. Mean Rewar : 90.762.  td of Reward  47.138. T aining.
```

I quit at 3M, looks promising

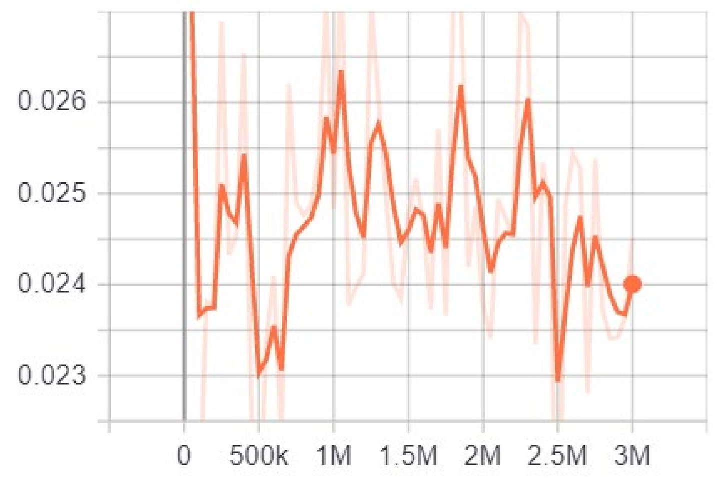
Cumulative Reward
tag: Environment/Cumulative Reward



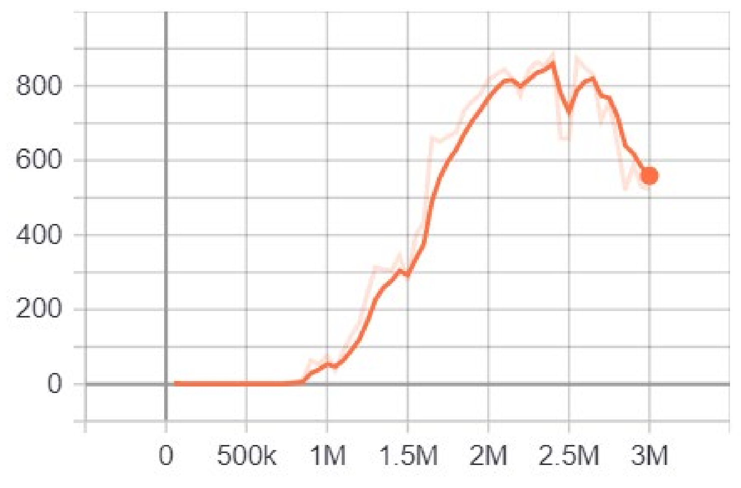
Episode Length
tag: Environment/Episode Length



Policy Loss
tag: Losses/Policy Loss



Value Loss
tag: Losses/Value Loss

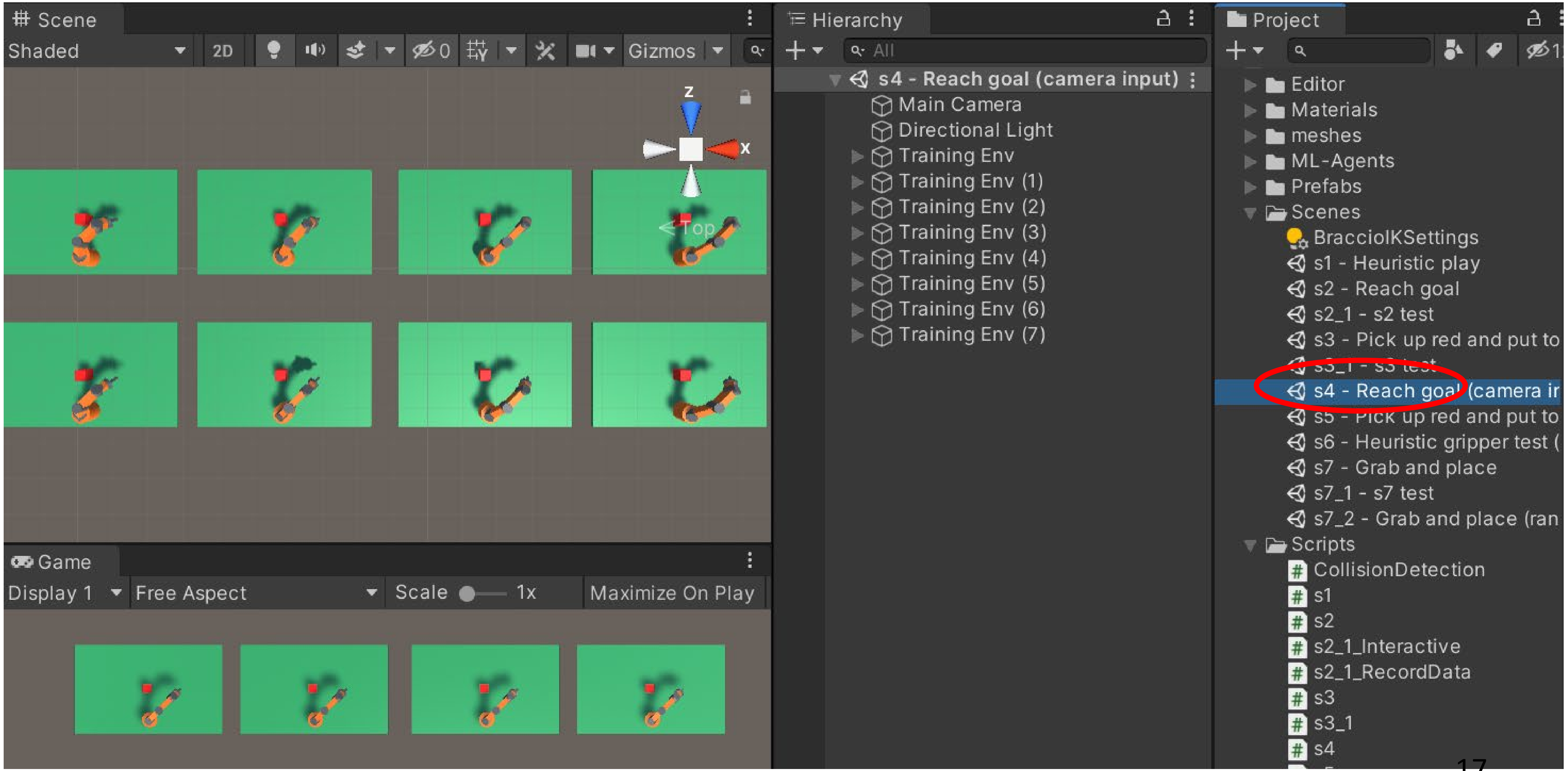


HW4(2)

- Describe the training setting
- Show tensor board plots and discuss your training performance
- Describe your test performance

(3) Reach goal using camera sensor

Open s4



s4 - Reach goal (camera input)
- Main Camera
- Directional Light
- Training Env
- Training Env (1)
- Training Env (2)
- Training Env (3)
- Training Env (4)
- Training Env (5)
- Training Env (6)
- Training Env (7)

- Editor
- Materials
- meshes
- ML-Agents
- Prefabs
- Scenes
  - BraccioIKSettings
  - s1 - Heuristic play
  - s2 - Reach goal
  - s2_1 - s2 test
  - s3 - Pick up red and put to
  - s3_1 - s3 test
  - s4 - Reach goal (camera ir
  - s5 - Pick up red and put to
  - s6 - Heuristic gripper test (
  - s7 - Grab and place
  - s7_1 - s7 test
  - s7_2 - Grab and place (ran
- Scripts
  - CollisionDetection
  - s1
  - s2
  - s2_1_Interactive
  - s2_1_RecordData
  - s3
  - s3_1
  - s4

Training setting

s = feature map vector from a CNN, size = ?
Input image to the CNN is captured by a camera from top, size = 84x84x3

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

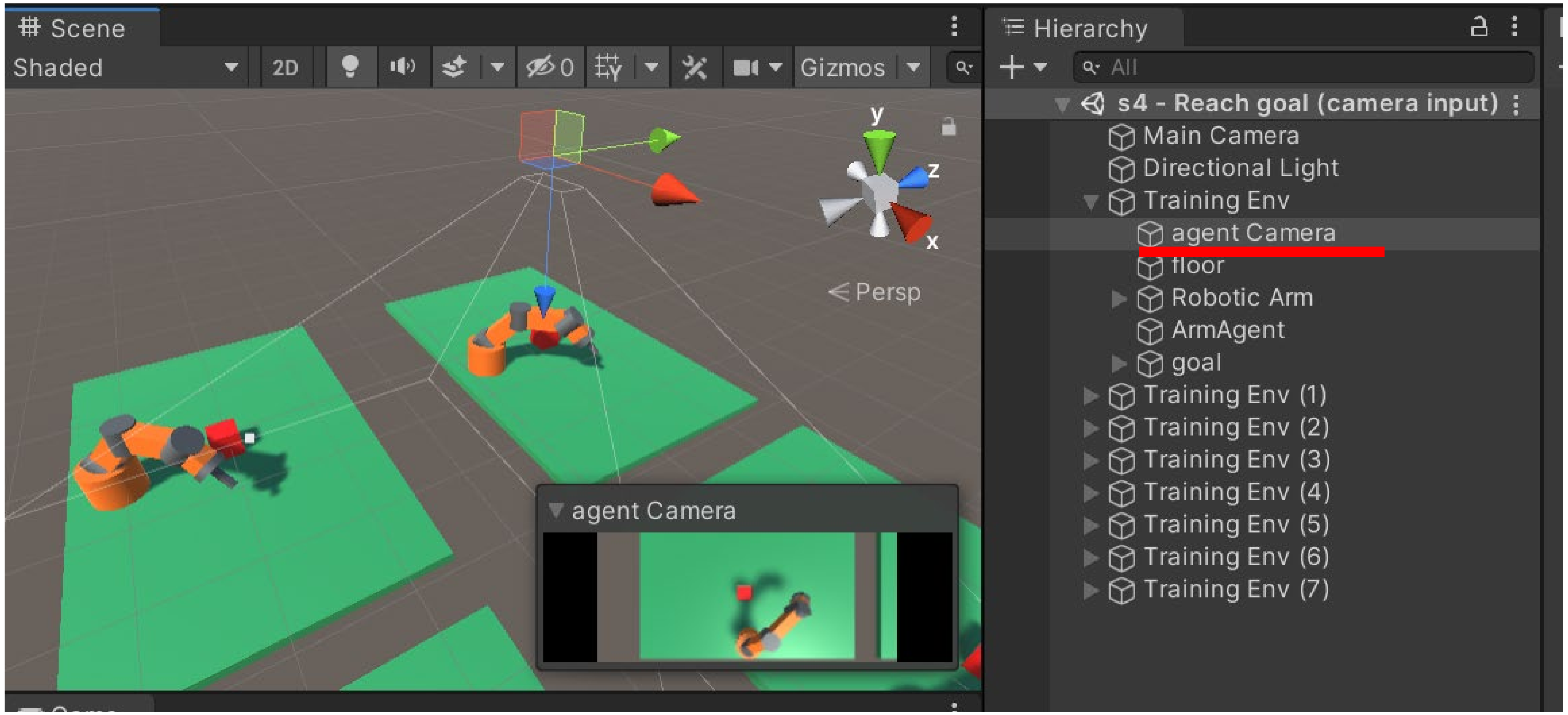
$$r = \begin{cases} -0.005 & \text{per step} \\ -5 & \text{collision, out of range} \\ +20 & \text{goal, } d \leq 0.5 \end{cases}$$

No. of training environment = 9

Goal initialize = randomly positioned in polar system θ=-80~80, r=0.8~1.5

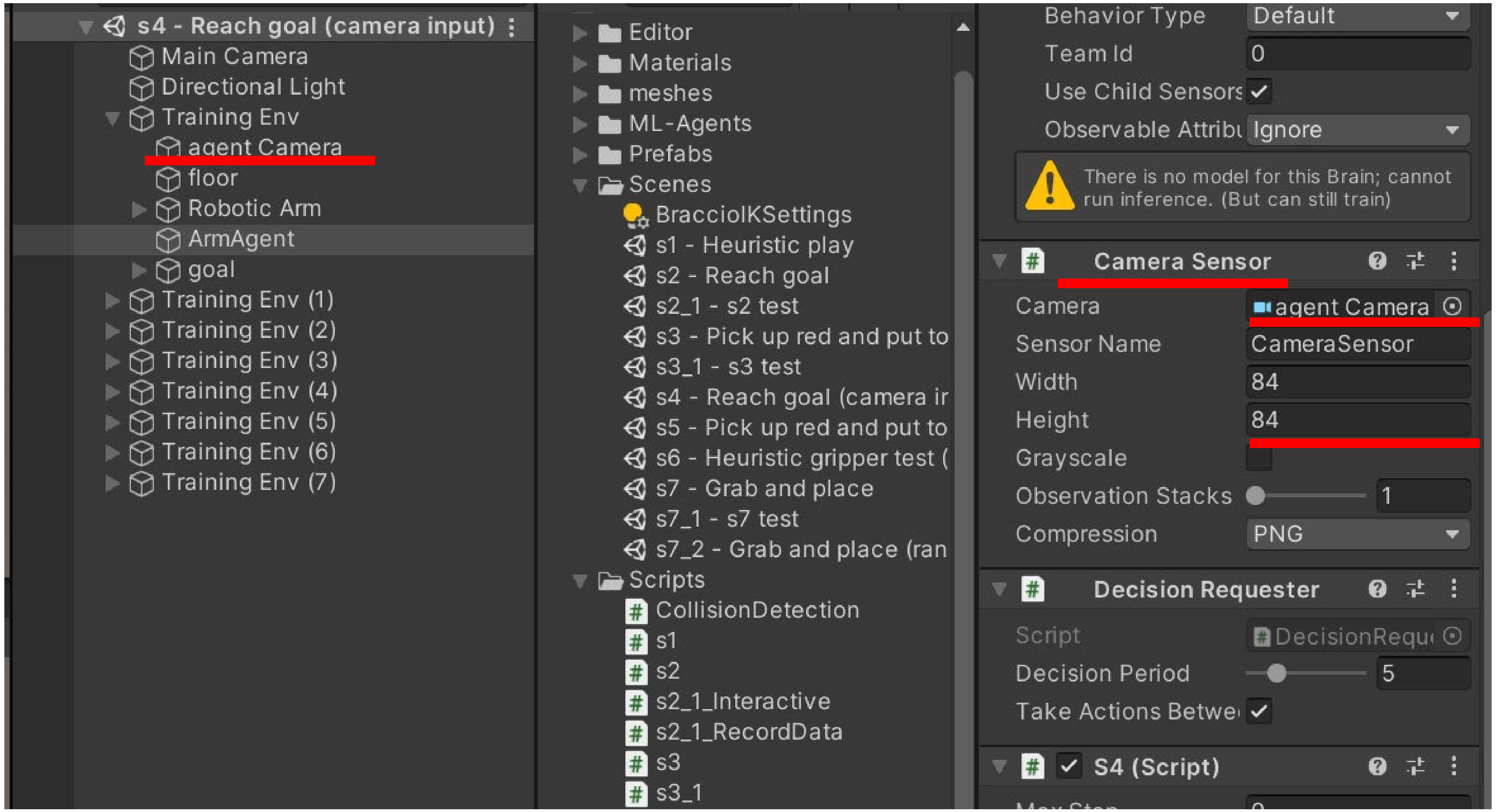
Arm initialize: $(\theta_b = 0, \theta_u = 45, \theta_l = 45, \theta_w = 45)$

Add camera sensor to the robot agent



s4 - Reach goal (camera input)
- Main Camera
- Directional Light
- Training Env
  - agent Camera
  - floor
  - Robotic Arm
  - ArmAgent
  - goal
- Training Env (1)
- Training Env (2)
- Training Env (3)
- Training Env (4)
- Training Env (5)
- Training Env (6)
- Training Env (7)

Add camera sensor to the robot agent



s4 - Reach goal (camera input)
Main Camera
Directional Light
Training Env
agent Camera
floor
Robotic Arm
ArmAgent
goal
Training Env (1)
Training Env (2)
Training Env (3)
Training Env (4)
Training Env (5)
Training Env (6)
Training Env (7)

Editor
Materials
meshes
ML-Agents
Prefabs
Scenes
BraccioIKSettings
s1 - Heuristic play
s2 - Reach goal
s2_1 - s2 test
s3 - Pick up red and put to
s3_1 - s3 test
s4 - Reach goal (camera ir
s5 - Pick up red and put to
s6 - Heuristic gripper test (
s7 - Grab and place
s7_1 - s7 test
s7_2 - Grab and place (ran
Scripts
CollisionDetection
s1
s2
s2_1_Interactive
s2_1_RecordData
s3
s3_1

Behavior Type	Default
Team Id	0
Use Child Sensors	✓
Observable Attribu	Ignore

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

Camera Sensor

Camera	agent Camera
Sensor Name	CameraSensor
Width	84
Height	84
Grayscale	
Observation Stacks	1
Compression	PNG

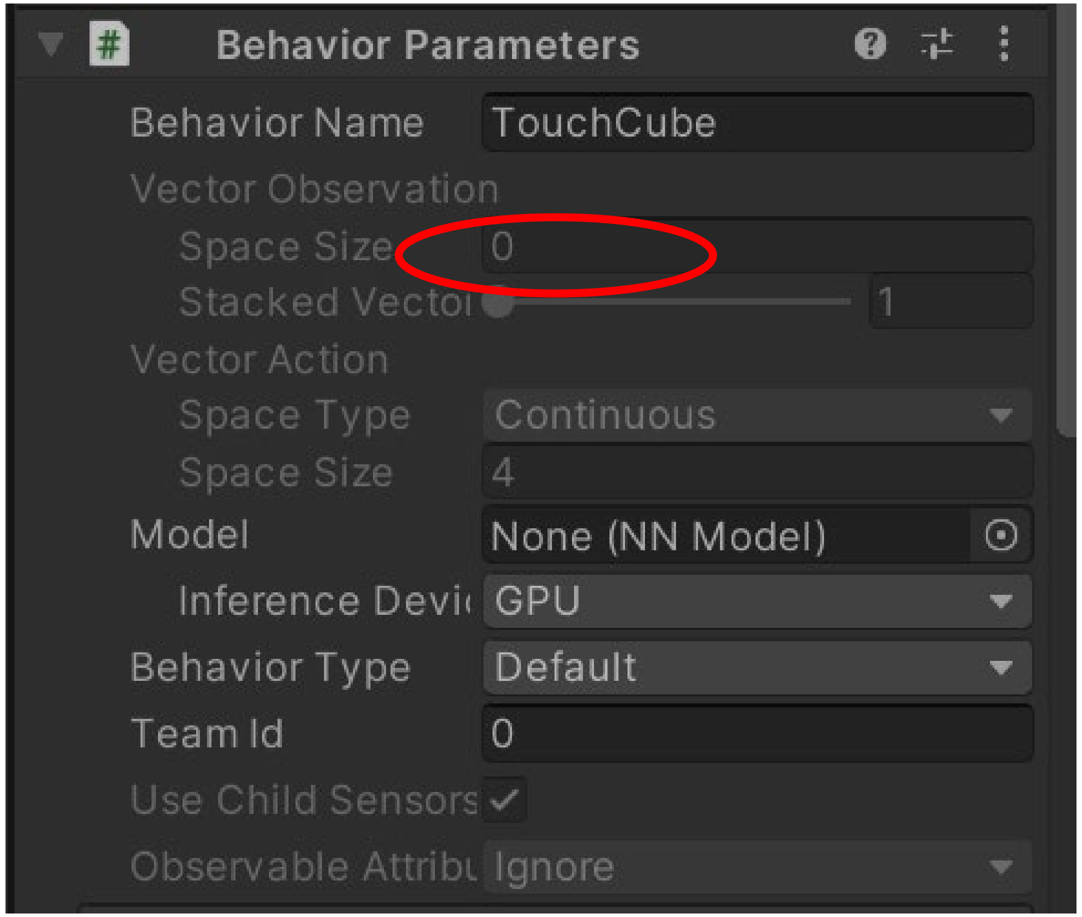
Decision Requester

Script	DecisionRequ
Decision Period	5
Take Actions Betwe	✓

S4 (Script)

Vector observation = 0

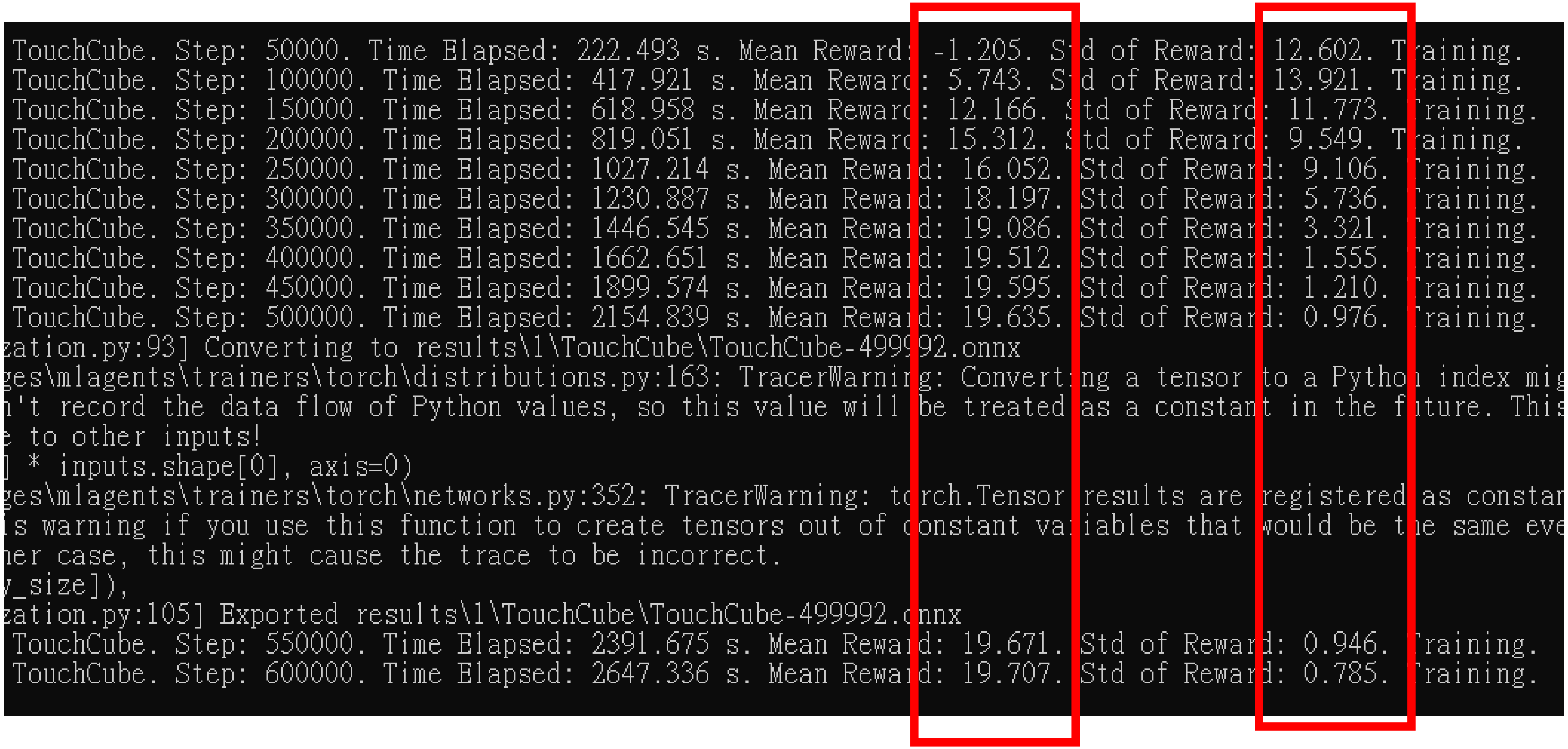
```
public override void CollectObservations(VectorSensor sensor)
{

}
```



Behavior Parameters

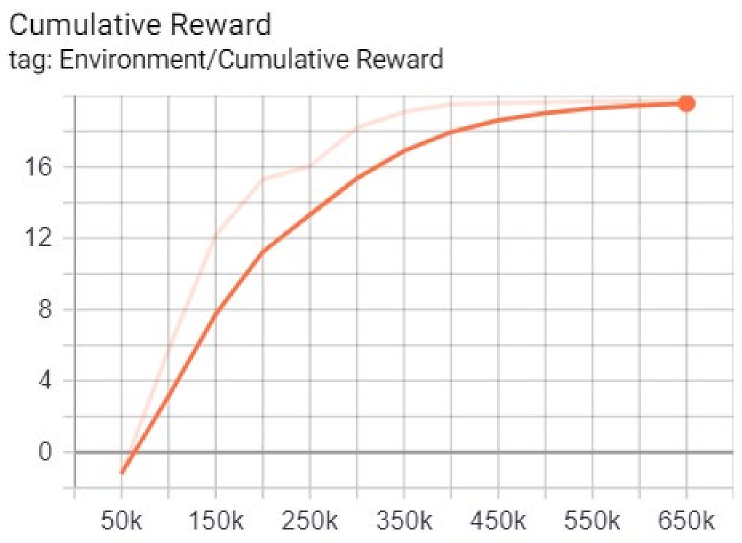
Behavior Name	TouchCube
Vector Observation	
Space Size	0
Stacked Vectoı	1
Vector Action	
Space Type	Continuous
Space Size	4
Model	None (NN Model)
Inference Devi	GPU
Behavior Type	Default
Team Id	0
Use Child Sensors	✓
Observable Attribu	Ignore

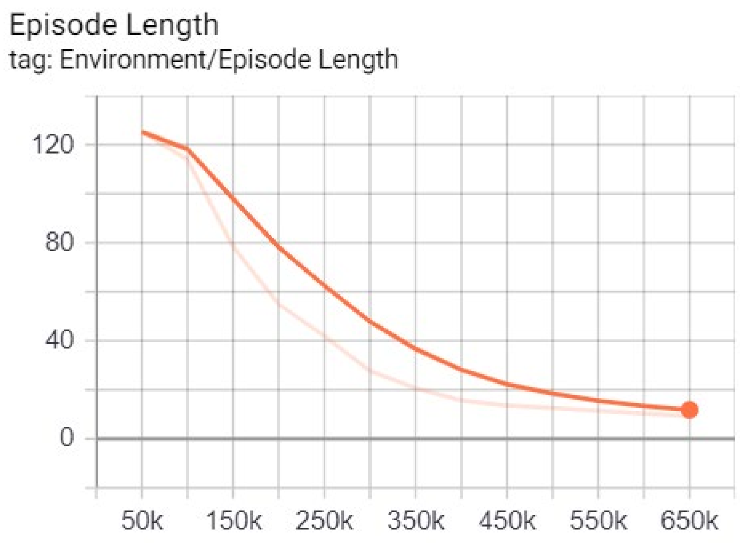
Looks good shortly (600K only)

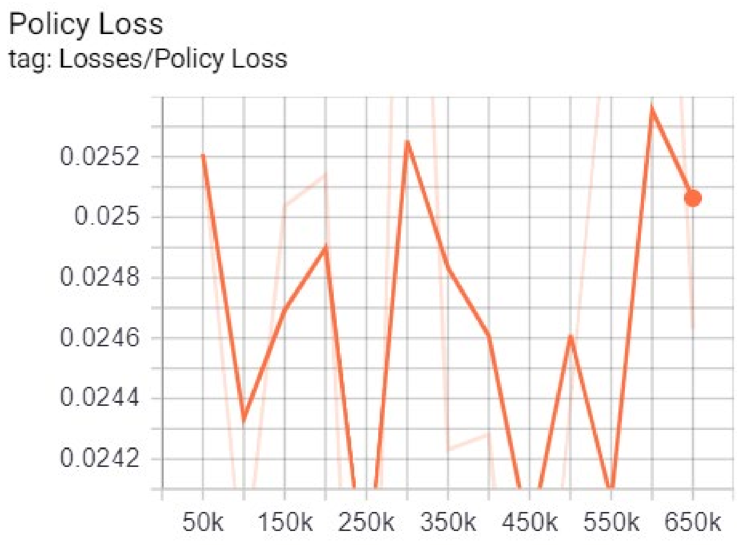


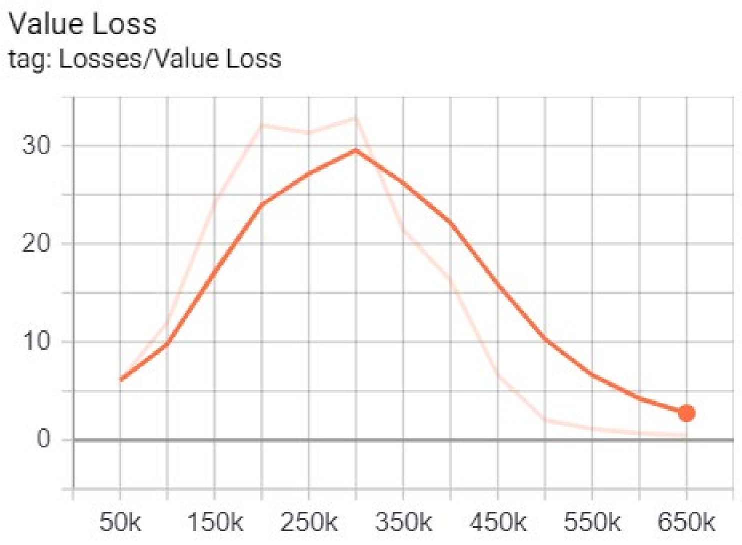
```
 TouchCube. Step: 50000. Time Elapsed: 222.493 s. Mean Reward: -1.205. Std of Reward: 12.602. Training.
 TouchCube. Step: 100000. Time Elapsed: 417.921 s. Mean Reward: 5.743. Std of Reward: 13.921. Training.
 TouchCube. Step: 150000. Time Elapsed: 618.958 s. Mean Reward: 12.166. Std of Reward: 11.773. 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 mig
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 constan
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.
```

Good results only after 650K

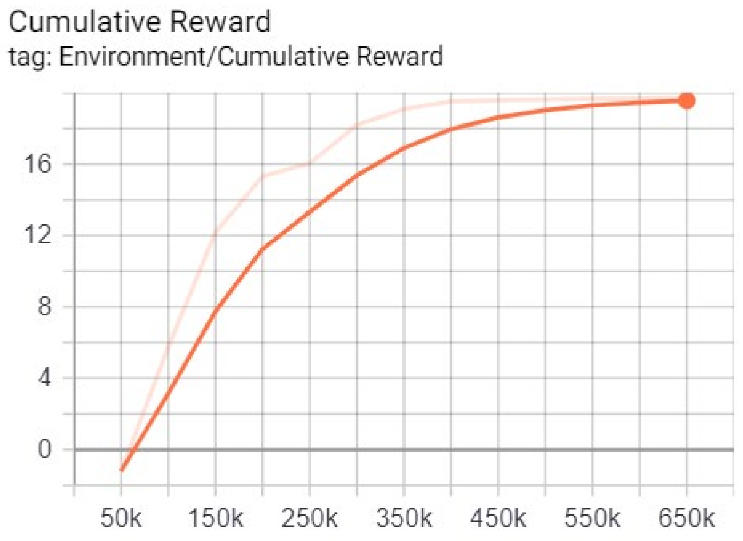


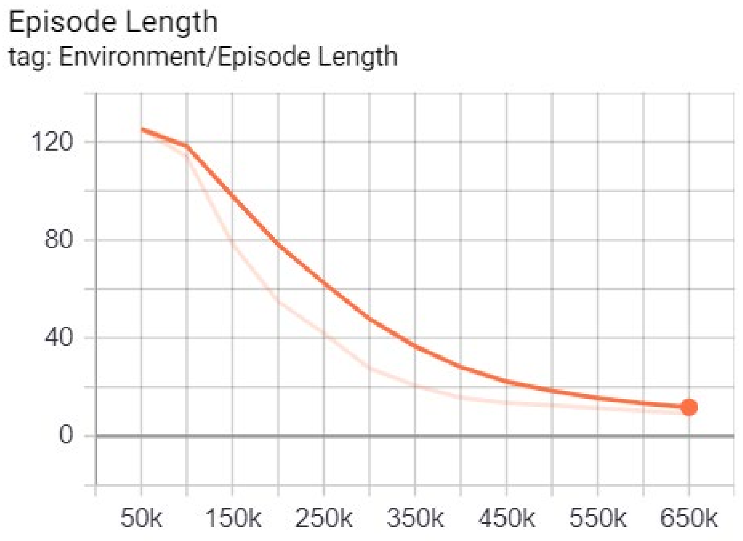






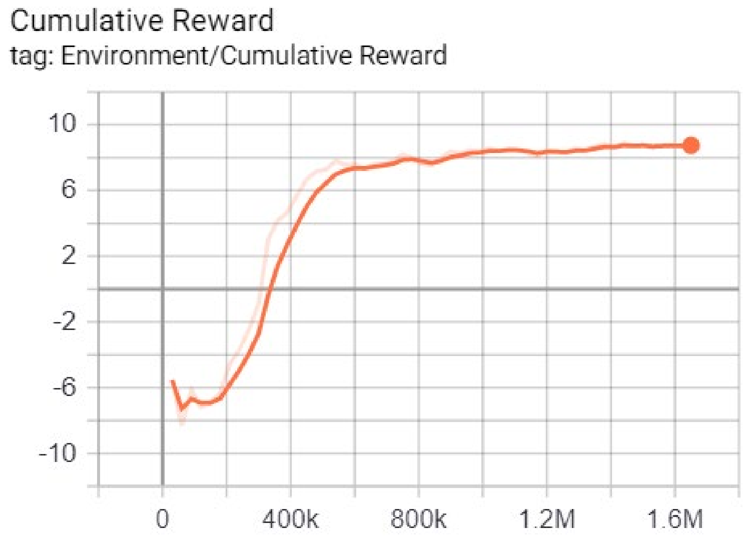
For simple environment, image input is easier than vector input

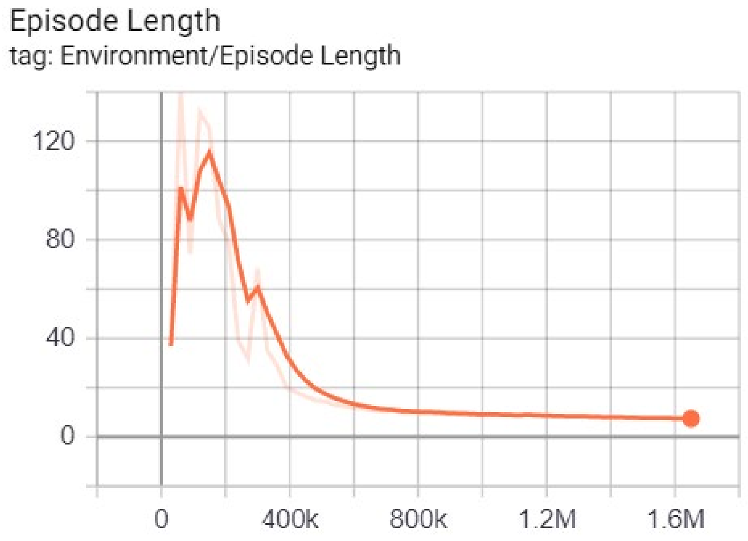




s = feature map vector from a CNN, size = ?

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





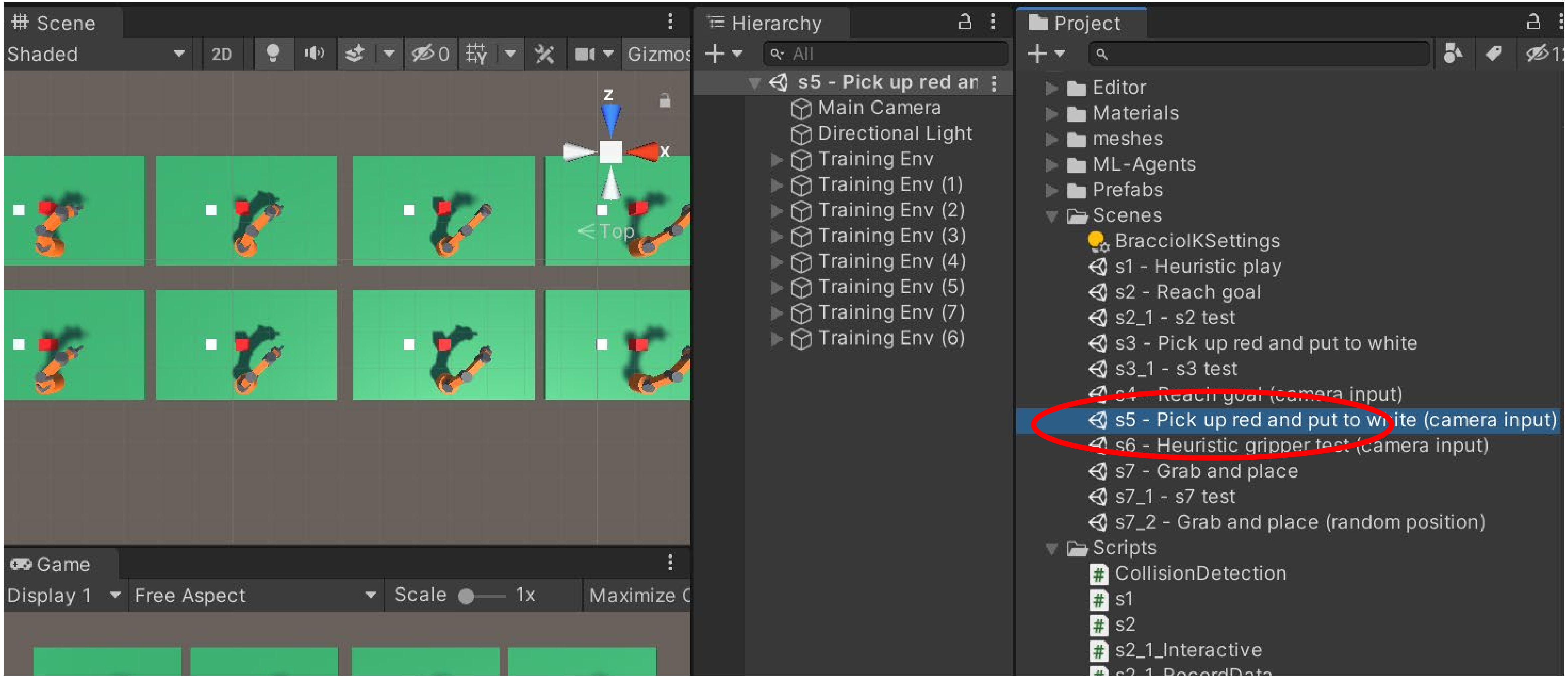
$$s = (\Delta x, \Delta y, \Delta z, \theta_{\mathrm{B}}, \theta_{\mathrm{U}}, \theta_{\mathrm{L}}, \theta_{\mathrm{W}})$$

HW4(3)

- Describe the training setting
- Show tensor board plots and discuss your training performance
- Describe your test performance

(4) Pick red cube and place it on top of the white cube using camera input

Open s5

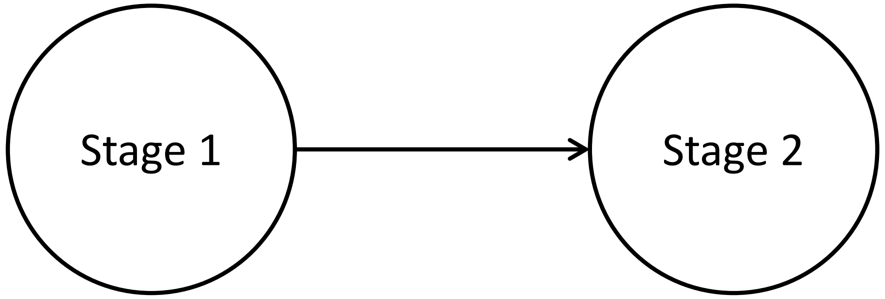


Scene
Shaded
2D
Gizmos
Hierarchy
All
s5 - Pick up red an
Main Camera
Directional Light
Training Env
Training Env (1)
Training Env (2)
Training Env (3)
Training Env (4)
Training Env (5)
Training Env (7)
Training Env (6)

Project
Editor
Materials
meshes
ML-Agents
Prefabs
Scenes
BraccioIKSettings
s1 - Heuristic play
s2 - Reach goal
s2_1 - s2 test
s3 - Pick up red and put to white
s3_1 - s3 test
s4 - Reach goal (camera input)
s5 - Pick up red and put to white (camera input)
s6 - Heuristic gripper test (camera input)
s7 - Grab and place
s7_1 - s7 test
s7_2 - Grab and place (random position)
Scripts
CollisionDetection
s1
s2
s2_1_Interactive

Game
Display 1
Free Aspect
Scale 1x
Maximize

Training setting



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

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

$$r = \begin{cases} -0.005 & \\ -5 & \\ +20 & d_1 \leq 0.5 \end{cases}$$

$$r = \begin{cases} -0.005 & \text{per step} \\ -5 & \text{collision, out of range} \\ +20 & \text{goal}, d_2 \leq 0.5 \end{cases}$$

NN: ?-512-512-512-4

Time horizon = 2000

Buffer size = 20480

Batch size = 2048

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

No. of training environment = 9

Goal initialize = randomly positioned in polar system θ=-80~80, r=0.8~1.5

Goal2 initialize = same as goal 1

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

HW4(4)

- Describe the training setting
- Show tensor board plots and discuss your training performance
- Describe your test performance