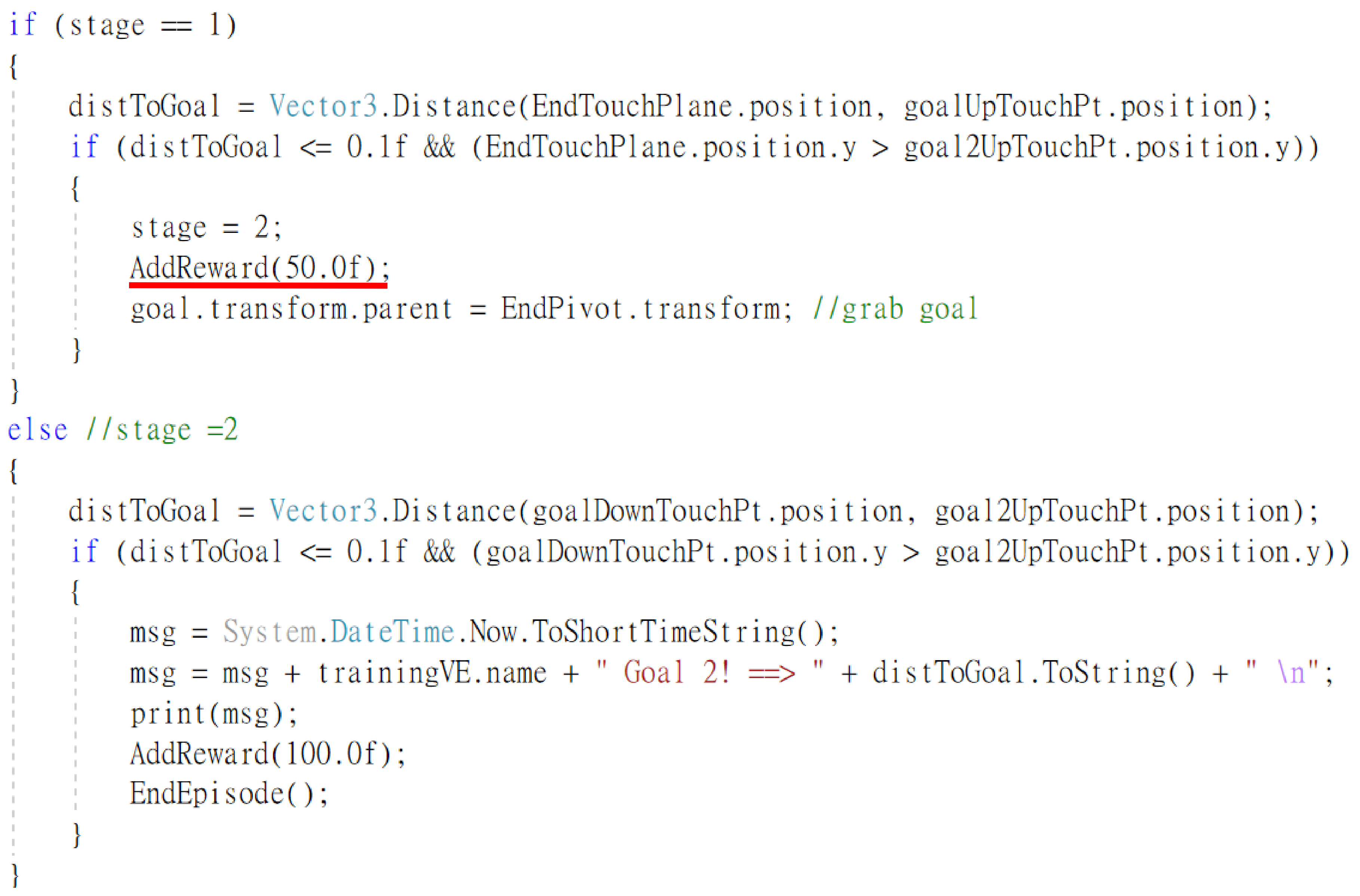
Discussion

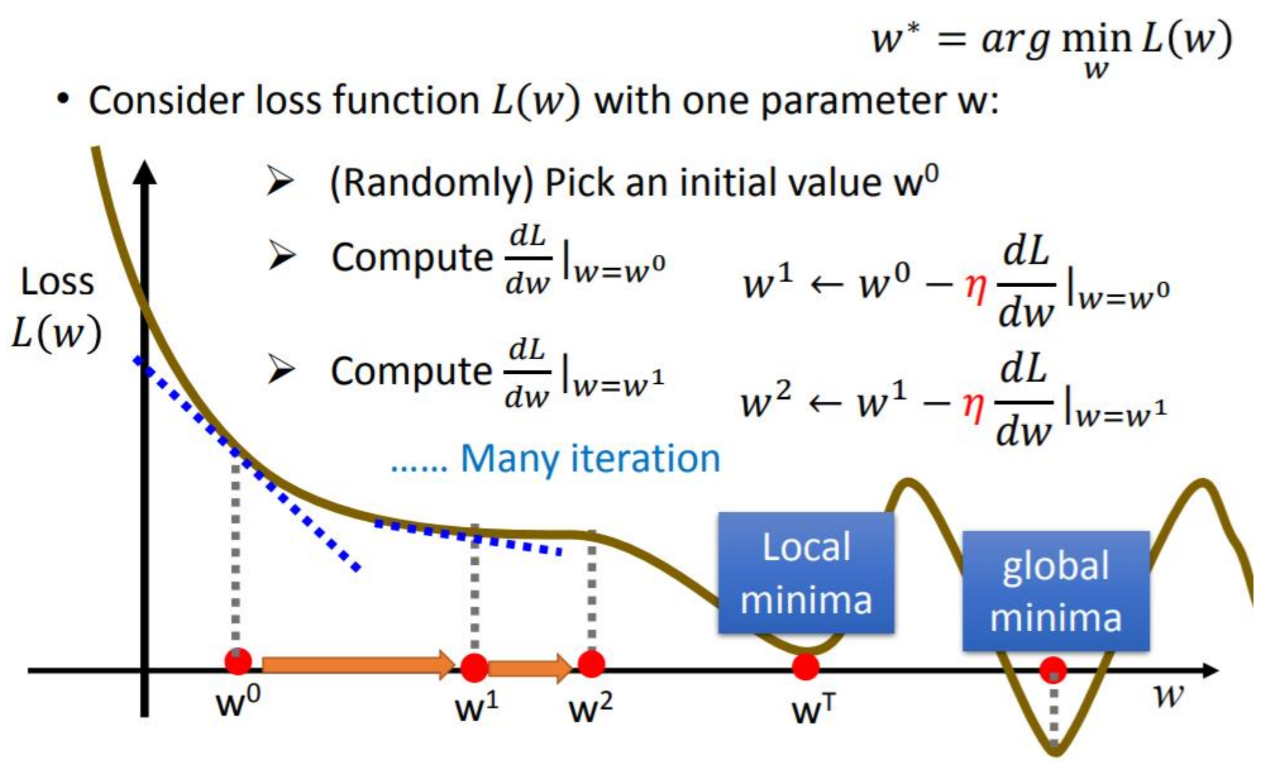
Reward engineering

Goal 1 rewards is too large, NN will think it is optimal and not try to reach goal 2



```
if (stage == 1)
{
    distToGoal = Vector3.Distance(EndTouchPlane.position, goalUpTouchPt.position);
    if (distToGoal <= 0.1f && (EndTouchPlane.position.y > goal2UpTouchPt.position.y))
    {
        stage = 2;
        AddReward(50.0f);
        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))
    {
        msg = System.DateTime.Now.ToShortTimeString();
        msg = msg + trainingVE.name + " Goal 2! ==> " + distToGoal.ToString() + " \n";
        print(msg);
        AddReward(100.0f);
        EndEpisode();
    }
}
```

Local optimization problem



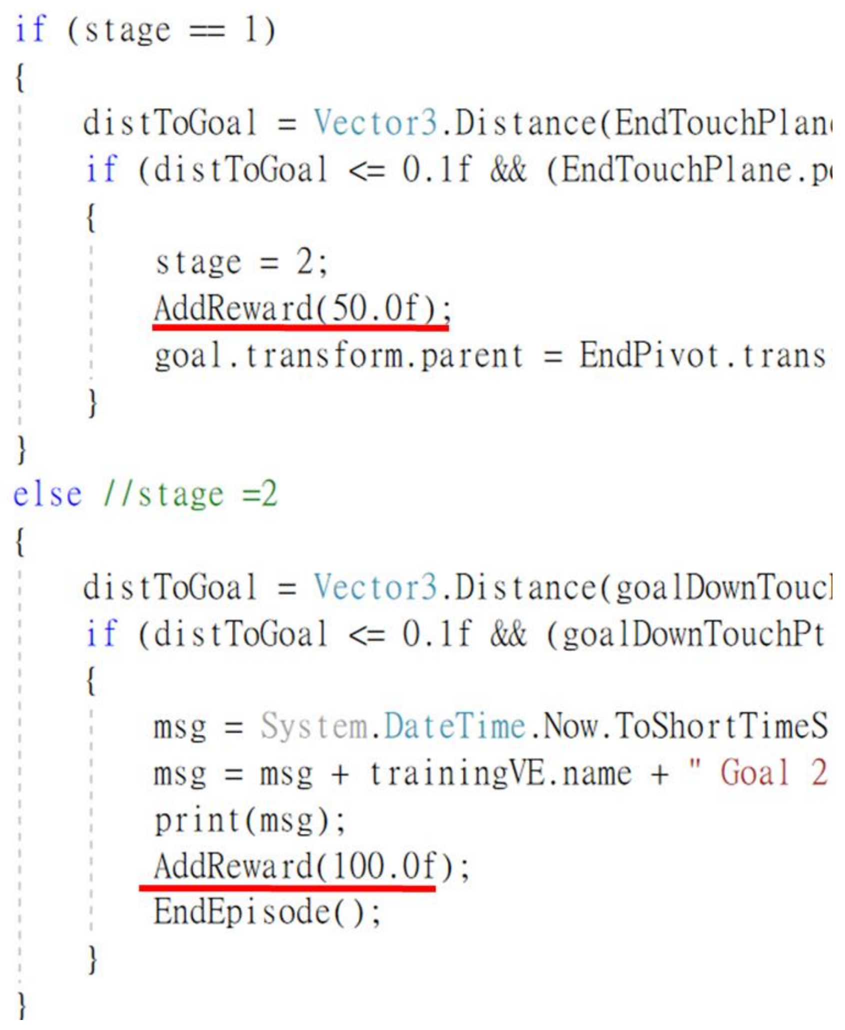
$$w^* = arg\min_w L(w)$$

- Consider loss function $L(w)$ with one parameter w:
  - ➢ (Randomly) Pick an initial value w^0
  - ➢ Compute $\frac{dL}{dw}|_{w=w^0}$ $\quad w^1 \leftarrow w^0 - \eta \frac{dL}{dw}|_{w=w^0}$
  - ➢ Compute $\frac{dL}{dw}|_{w=w^1}$ $\quad w^2 \leftarrow w^1 - \eta \frac{dL}{dw}|_{w=w^1}$

…… Many iteration

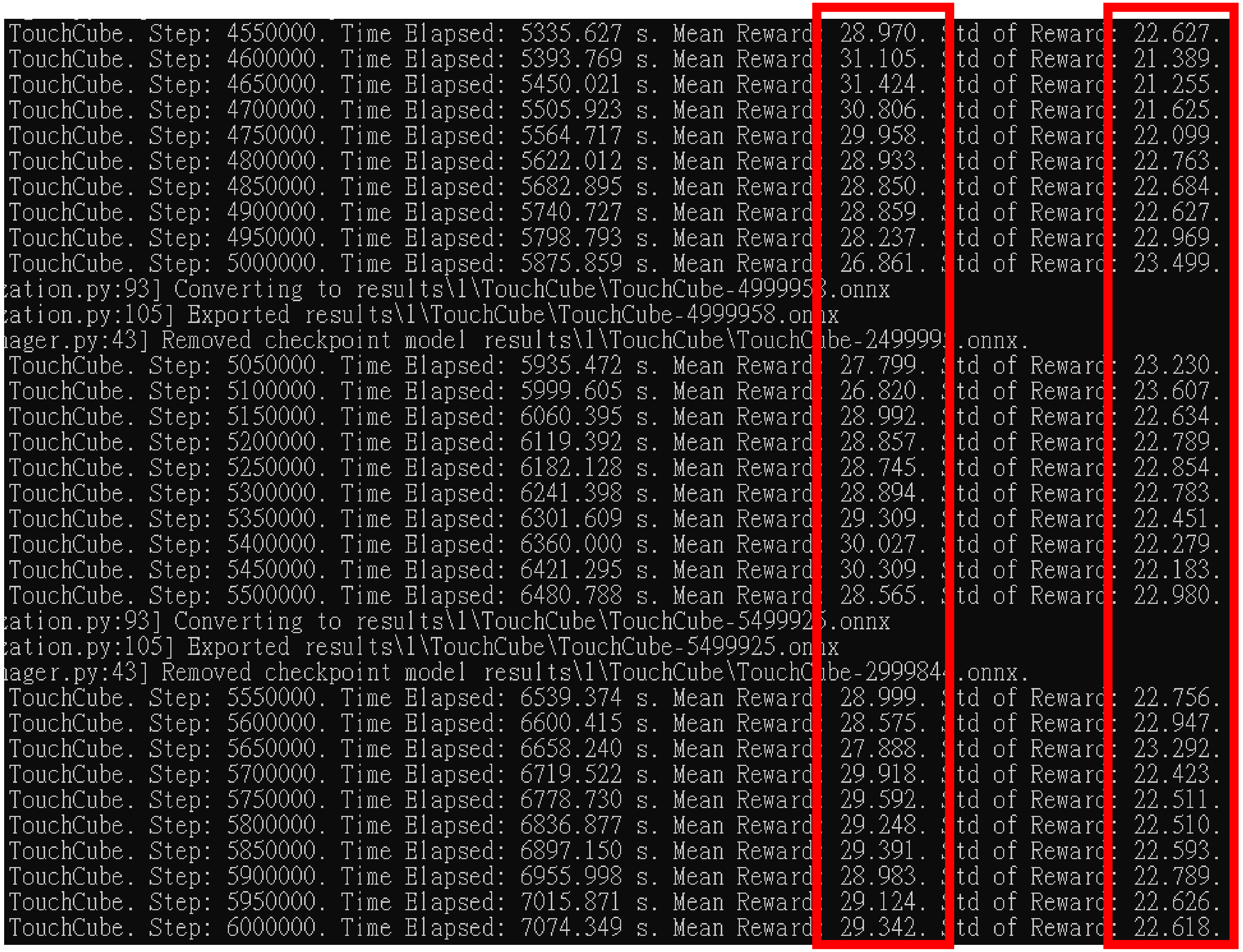
Reference: 李弘毅 ML Lecture 1 https://youtu.be/CXgbekl66jc

AI lecture 1. Introduction.pdf



```
if (stage == 1)
{
    distToGoal = Vector3.Distance(EndTouchPlan
    if (distToGoal <= 0.1f && (EndTouchPlane.p
    {
        stage = 2;
        AddReward(50.0f);
        goal.transform.parent = EndPivot.trans
    }
}
else //stage =2
{
    distToGoal = Vector3.Distance(goalDownTouc
    if (distToGoal <= 0.1f && (goalDownTouchPt
    {
        msg = System.DateTime.Now.ToShortTimeS
        msg = msg + trainingVE.name + " Goal 2
        print(msg);
        AddReward(100.0f);
        EndEpisode();
    }
}
```

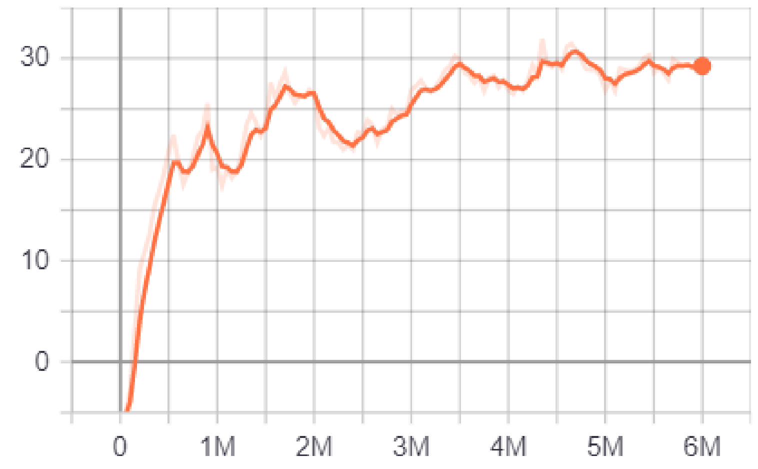
Results after 6M, not trying to reach goal 2



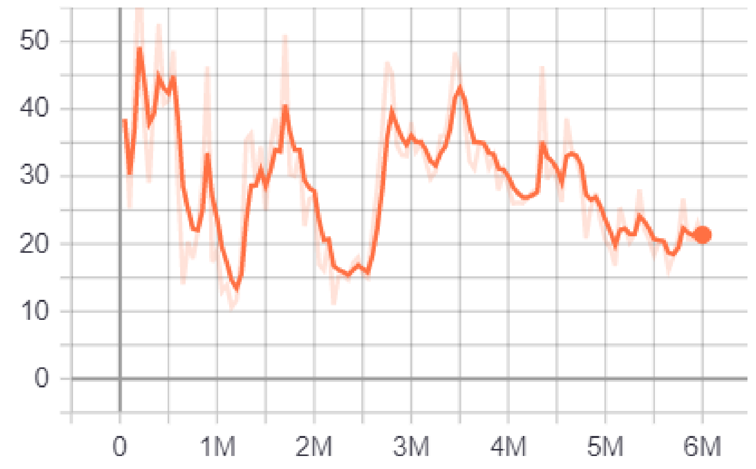
```
 TouchCube. Step: 4550000. Time Elapsed: 5335.627 s. Mean Reward  28.970.  td of Reward  22.627.
 TouchCube. Step: 4600000. Time Elapsed: 5393.769 s. Mean Reward  31.105.  td of Reward  21.389.
 TouchCube. Step: 4650000. Time Elapsed: 5450.021 s. Mean Reward  31.424.  td of Reward  21.255.
 TouchCube. Step: 4700000. Time Elapsed: 5505.923 s. Mean Reward  30.806.  td of Reward  21.625.
 TouchCube. Step: 4750000. Time Elapsed: 5564.717 s. Mean Reward  29.958.  td of Reward  22.099.
 TouchCube. Step: 4800000. Time Elapsed: 5622.012 s. Mean Reward  28.933.  td of Reward  22.763.
 TouchCube. Step: 4850000. Time Elapsed: 5682.895 s. Mean Reward  28.850.  td of Reward  22.684.
 TouchCube. Step: 4900000. Time Elapsed: 5740.727 s. Mean Reward  28.859.  td of Reward  22.627.
 TouchCube. Step: 4950000. Time Elapsed: 5798.793 s. Mean Reward  28.237.  td of Reward  22.969.
 TouchCube. Step: 5000000. Time Elapsed: 5875.859 s. Mean Reward  26.861.  td of Reward  23.499.
ation.py:93] Converting to results\1\TouchCube\TouchCube-49999 8.onnx
ation.py:105] Exported results\1\TouchCube\TouchCube-4999958.on x
ager.py:43] Removed checkpoint model results\1\TouchCube\TouchC be-249999 .onnx.
 TouchCube. Step: 5050000. Time Elapsed: 5935.472 s. Mean Reward  27.799.  td of Reward  23.230.
 TouchCube. Step: 5100000. Time Elapsed: 5999.605 s. Mean Reward  26.820.  td of Reward  23.607.
 TouchCube. Step: 5150000. Time Elapsed: 6060.395 s. Mean Reward  28.992.  td of Reward  22.634.
 TouchCube. Step: 5200000. Time Elapsed: 6119.392 s. Mean Reward  28.857.  td of Reward  22.789.
 TouchCube. Step: 5250000. Time Elapsed: 6182.128 s. Mean Reward  28.745.  td of Reward  22.854.
 TouchCube. Step: 5300000. Time Elapsed: 6241.398 s. Mean Reward  28.894.  td of Reward  22.783.
 TouchCube. Step: 5350000. Time Elapsed: 6301.609 s. Mean Reward  29.309.  td of Reward  22.451.
 TouchCube. Step: 5400000. Time Elapsed: 6360.000 s. Mean Reward  30.027.  td of Reward  22.279.
 TouchCube. Step: 5450000. Time Elapsed: 6421.295 s. Mean Reward  30.309.  td of Reward  22.183.
 TouchCube. Step: 5500000. Time Elapsed: 6480.788 s. Mean Reward  28.565.  td of Reward  22.980.
ation.py:93] Converting to results\1\TouchCube\TouchCube-549992 .onnx
ation.py:105] Exported results\1\TouchCube\TouchCube-5499925.on x
ager.py:43] Removed checkpoint model results\1\TouchCube\TouchC be-299984 .onnx.
 TouchCube. Step: 5550000. Time Elapsed: 6539.374 s. Mean Reward  28.999.  td of Reward  22.756.
 TouchCube. Step: 5600000. Time Elapsed: 6600.415 s. Mean Reward  28.575.  td of Reward  22.947.
 TouchCube. Step: 5650000. Time Elapsed: 6658.240 s. Mean Reward  27.888.  td of Reward  23.292.
 TouchCube. Step: 5700000. Time Elapsed: 6719.522 s. Mean Reward  29.918.  td of Reward  22.423.
 TouchCube. Step: 5750000. Time Elapsed: 6778.730 s. Mean Reward  29.592.  td of Reward  22.511.
 TouchCube. Step: 5800000. Time Elapsed: 6836.877 s. Mean Reward  29.248.  td of Reward  22.510.
 TouchCube. Step: 5850000. Time Elapsed: 6897.150 s. Mean Reward  29.391.  td of Reward  22.593.
 TouchCube. Step: 5900000. Time Elapsed: 6955.998 s. Mean Reward  28.983.  td of Reward  22.789.
 TouchCube. Step: 5950000. Time Elapsed: 7015.871 s. Mean Reward  29.124.  td of Reward  22.626.
 TouchCube. Step: 6000000. Time Elapsed: 7074.349 s. Mean Reward  29.342.  td of Reward  22.618.
```

Results after 6M, not trying to reach goal 2

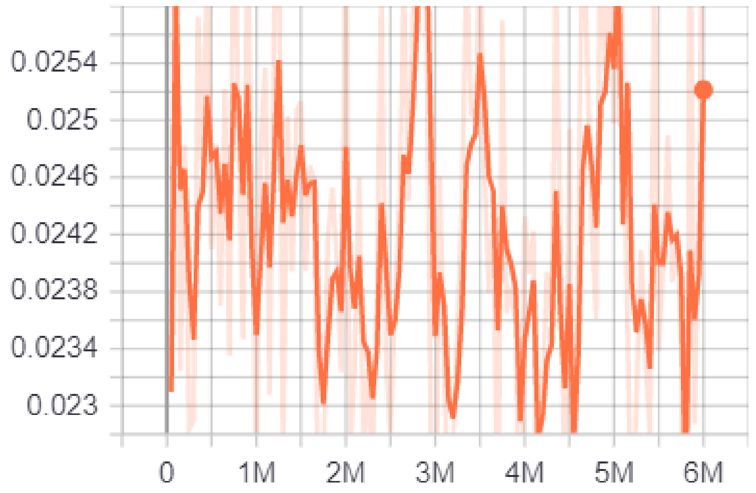
Cumulative Reward
tag: Environment/Cumulative Reward



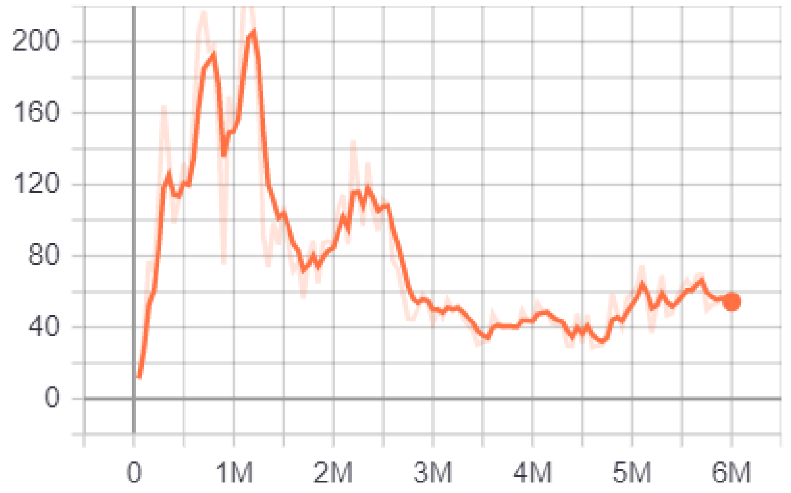
Episode Length
tag: Environment/Episode Length



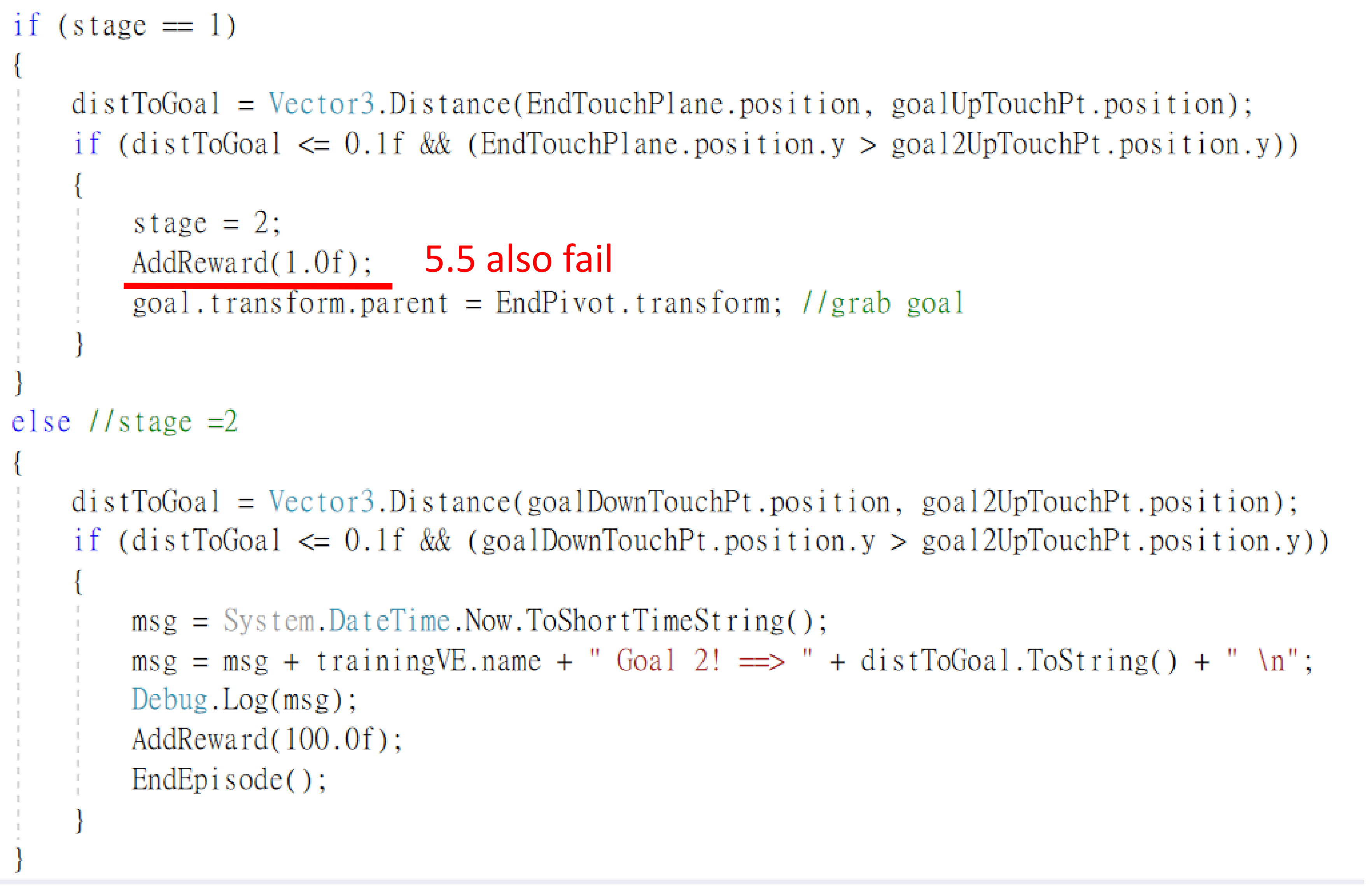
Policy Loss
tag: Losses/Policy Loss



Value Loss
tag: Losses/Value Loss



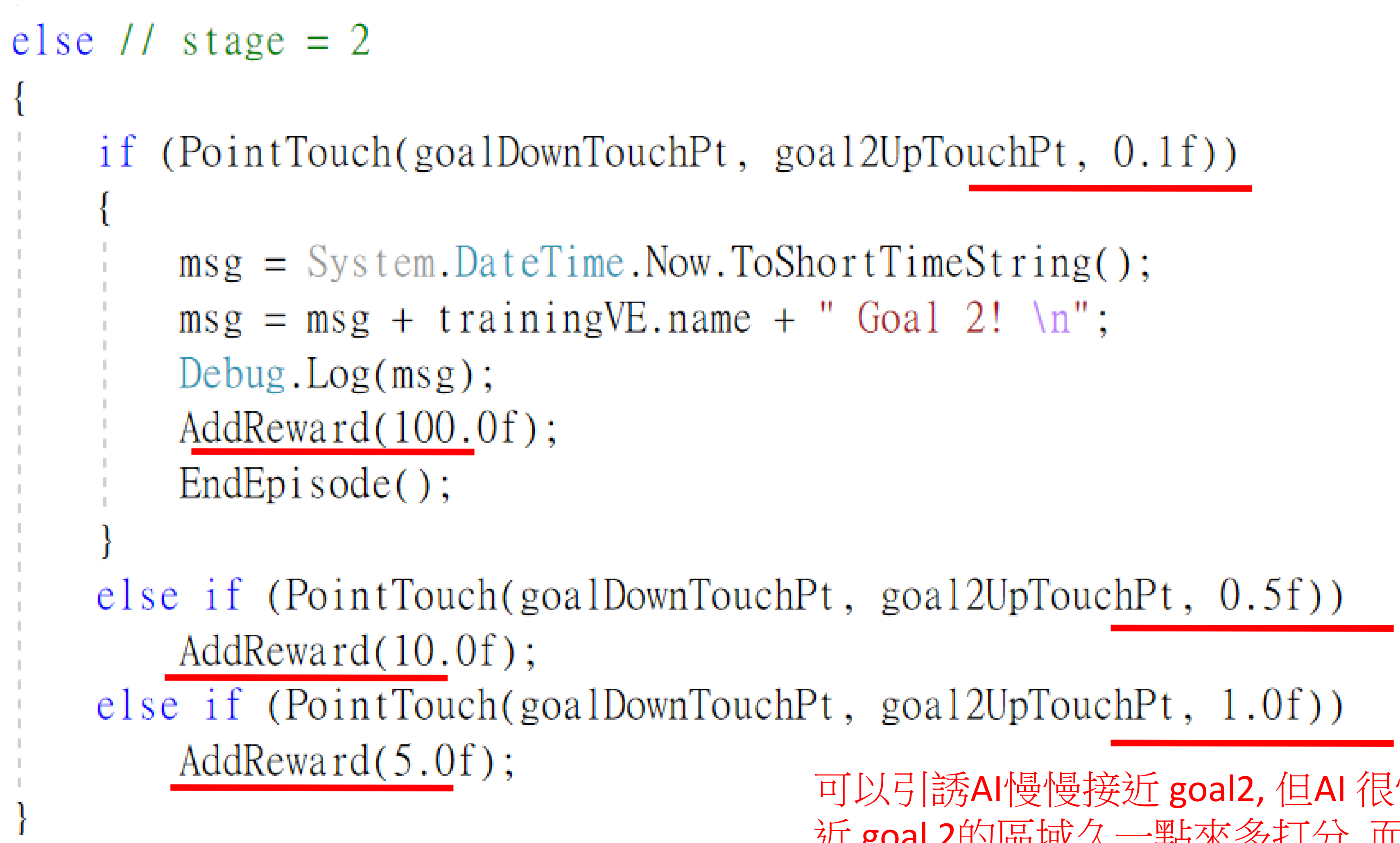
Reward is too small to encourage reach goal 1



```
if (stage == 1)
{
    distToGoal = Vector3.Distance(EndTouchPlane.position, goalUpTouchPt.position);
    if (distToGoal <= 0.1f && (EndTouchPlane.position.y > goal2UpTouchPt.position.y))
    {
        stage = 2;
        AddReward(1.0f);
        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))
    {
        msg = System.DateTime.Now.ToShortTimeString();
        msg = msg + trainingVE.name + " Goal 2! ==> " + distToGoal.ToString() + " \n";
        Debug.Log(msg);
        AddReward(100.0f);
        EndEpisode();
    }
}
```

5.5 also fail

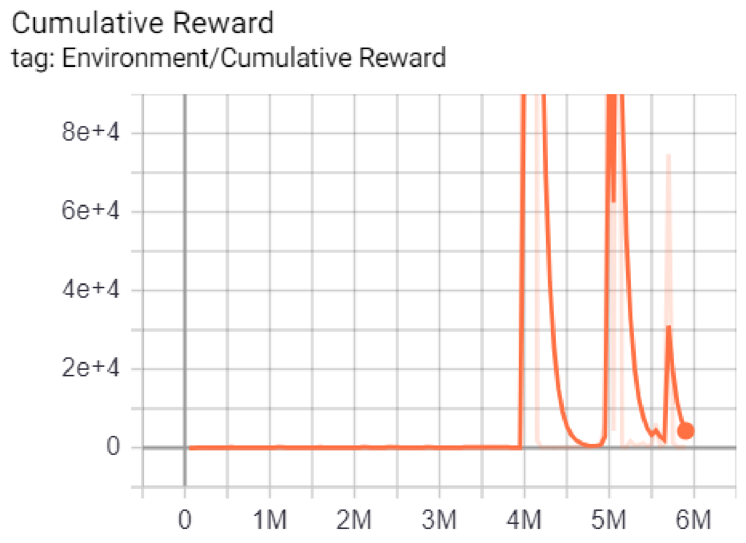
This reward design will guild AI to reach goal2, but AI learns to stay at sweet zone to get higher rewards rather than touch goal2

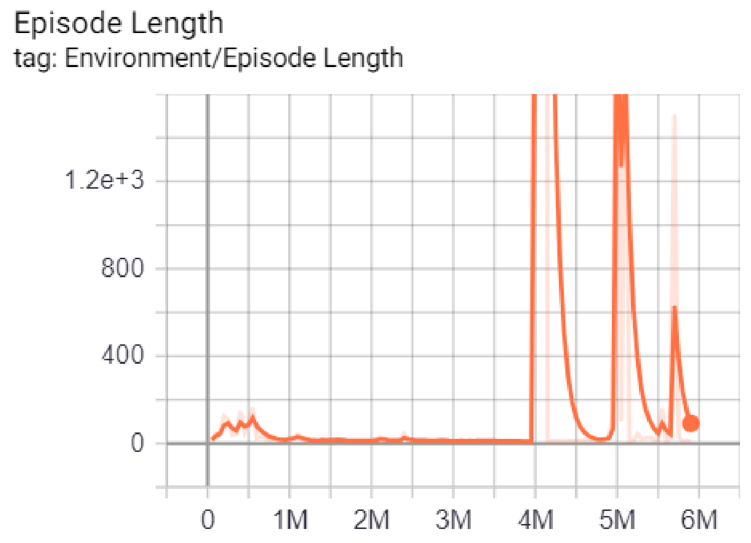


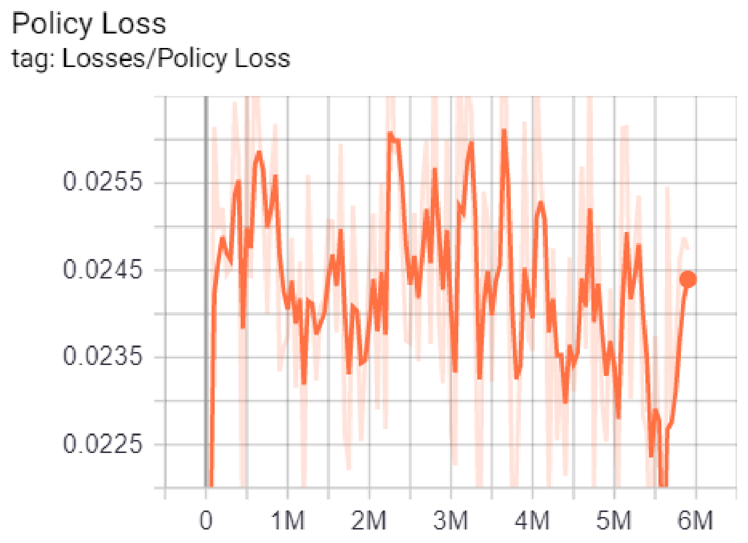
```
else // stage = 2
{
    if (PointTouch(goalDownTouchPt, goal2UpTouchPt, 0.1f))
    {
        msg = System.DateTime.Now.ToShortTimeString();
        msg = msg + trainingVE.name + " Goal 2! \n";
        Debug.Log(msg);
        AddReward(100.0f);
        EndEpisode();
    }
    else if (PointTouch(goalDownTouchPt, goal2UpTouchPt, 0.5f))
        AddReward(10.0f);
    else if (PointTouch(goalDownTouchPt, goal2UpTouchPt, 1.0f))
        AddReward(5.0f);
}
```

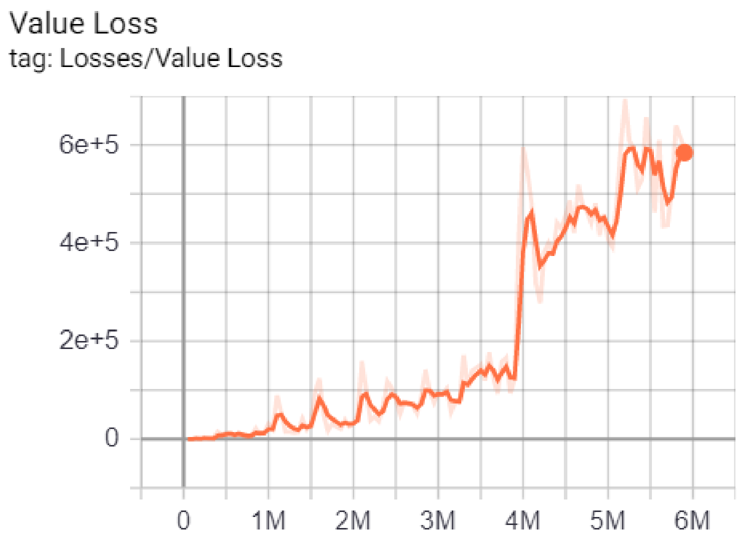
可以引誘AI慢慢接近 goal2, 但AI 很快就學會停在接近 goal 2的區域久一點來多打分, 而不去 touch goal2!

AI learns to stay at sweet zone to get higher rewards rather than touch goal2



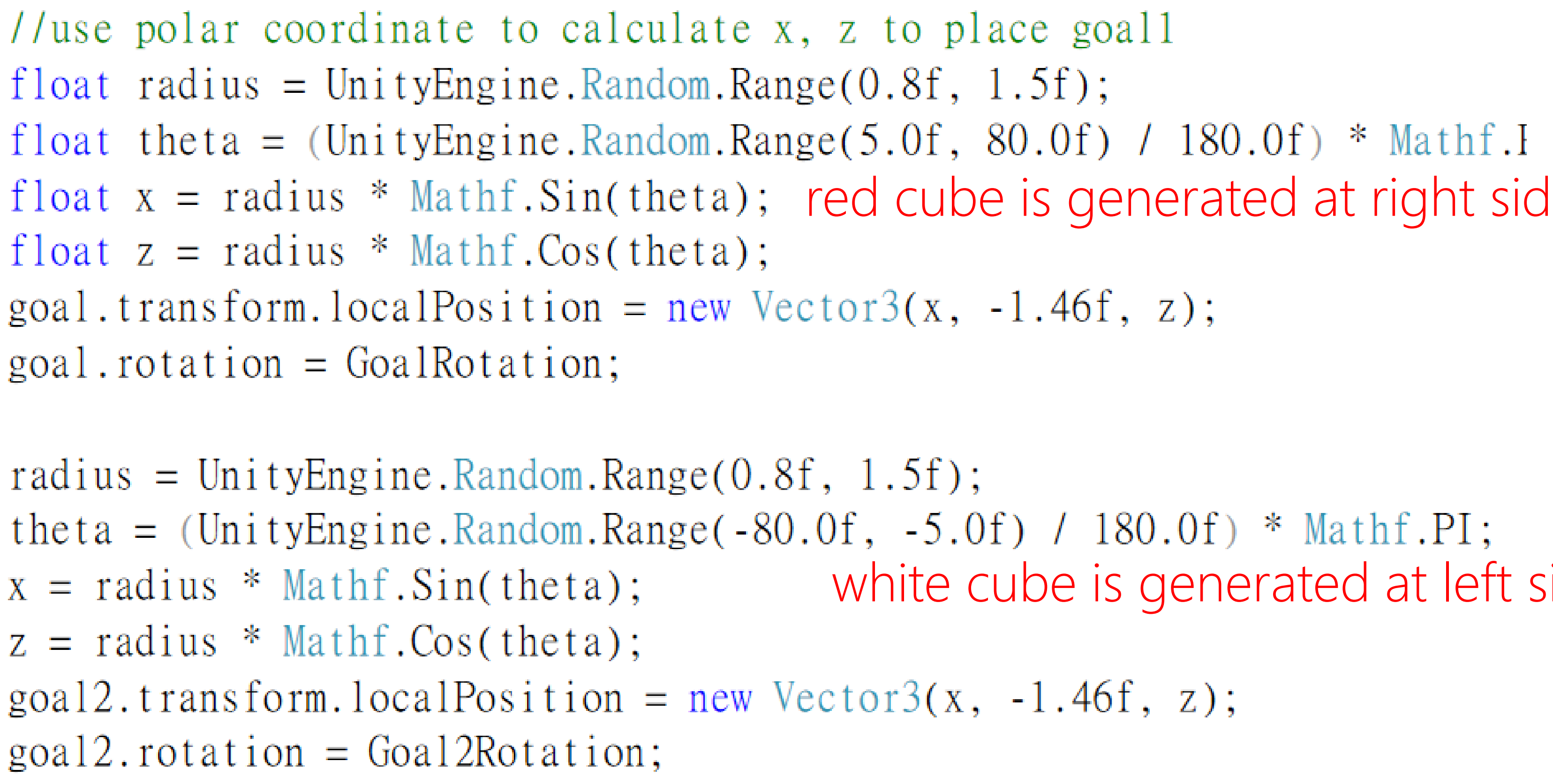






Improper initial position will result in biased behavior

Improper initial position design



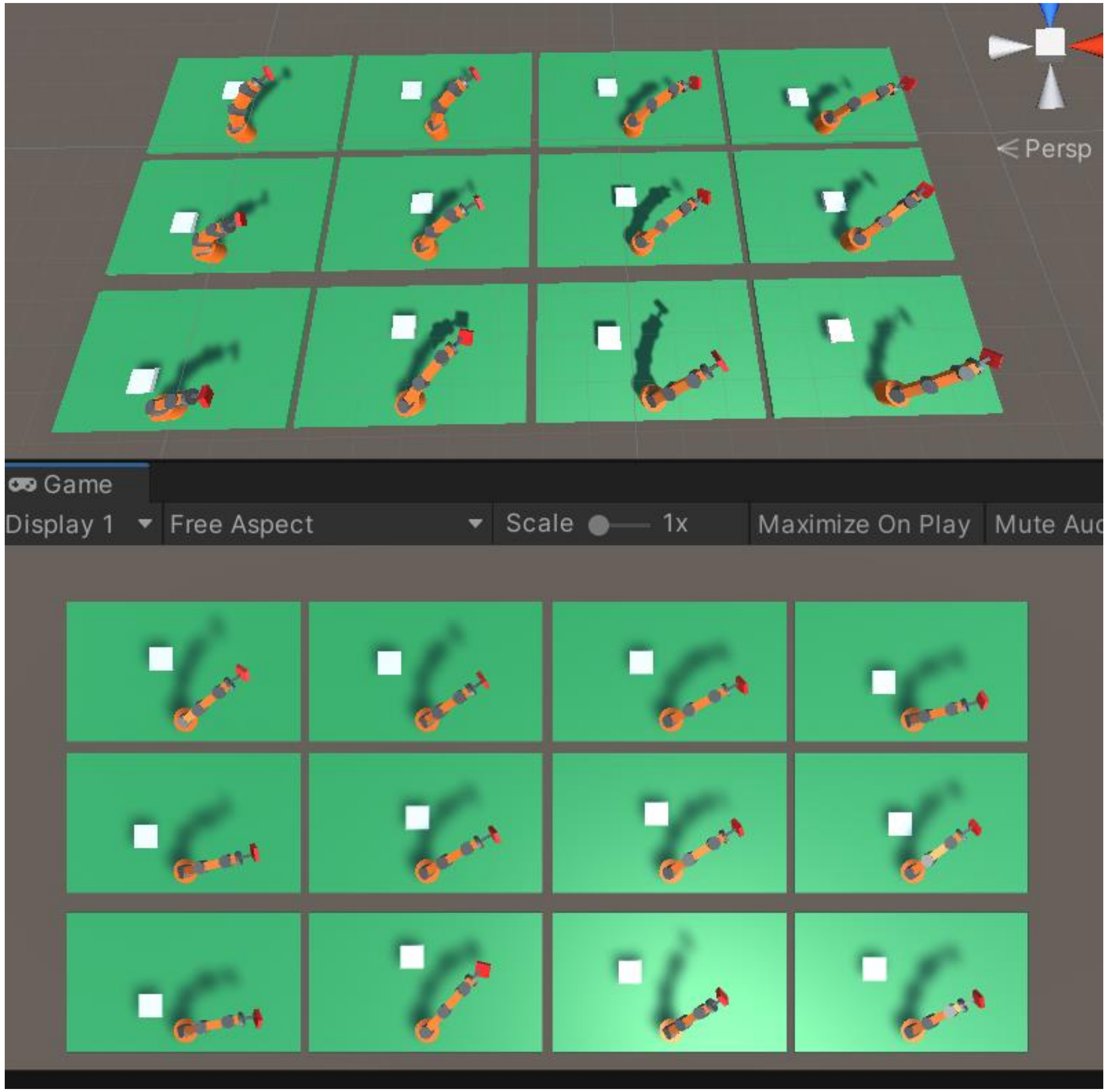
```
//use polar coordinate to calculate x, z to place goal1
float radius = UnityEngine.Random.Range(0.8f, 1.5f);
float theta = (UnityEngine.Random.Range(5.0f, 80.0f) / 180.0f) * Mathf.I
float x = radius * Mathf.Sin(theta);
float z = radius * Mathf.Cos(theta);
goal.transform.localPosition = new Vector3(x, -1.46f, z);
goal.rotation = GoalRotation;

radius = UnityEngine.Random.Range(0.8f, 1.5f);
theta = (UnityEngine.Random.Range(-80.0f, -5.0f) / 180.0f) * Mathf.PI;
x = radius * Mathf.Sin(theta);
z = radius * Mathf.Cos(theta);
goal2.transform.localPosition = new Vector3(x, -1.46f, z);
goal2.rotation = Goal2Rotation;
```

red cube is generated at right side

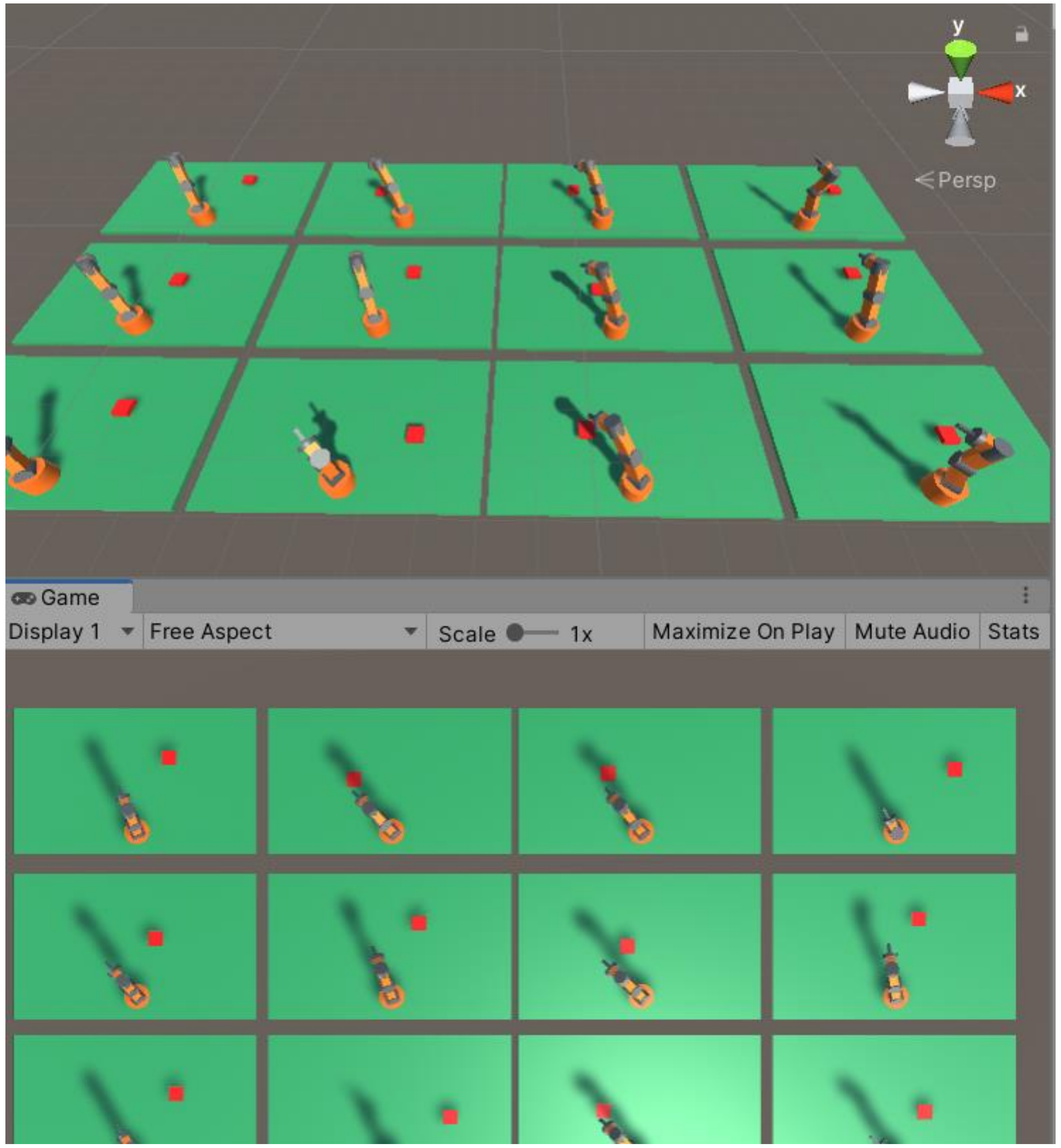
white cube is generated at left side

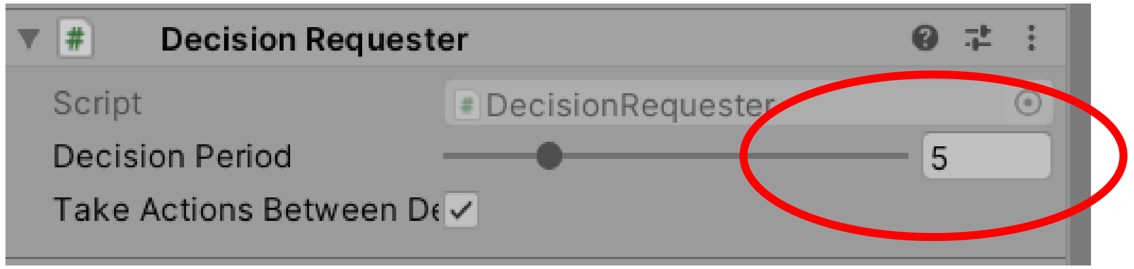
Biased behavior



Decision period

decision period = 1 will result strange arm behavior




Decision Requester

Script DecisionRequester

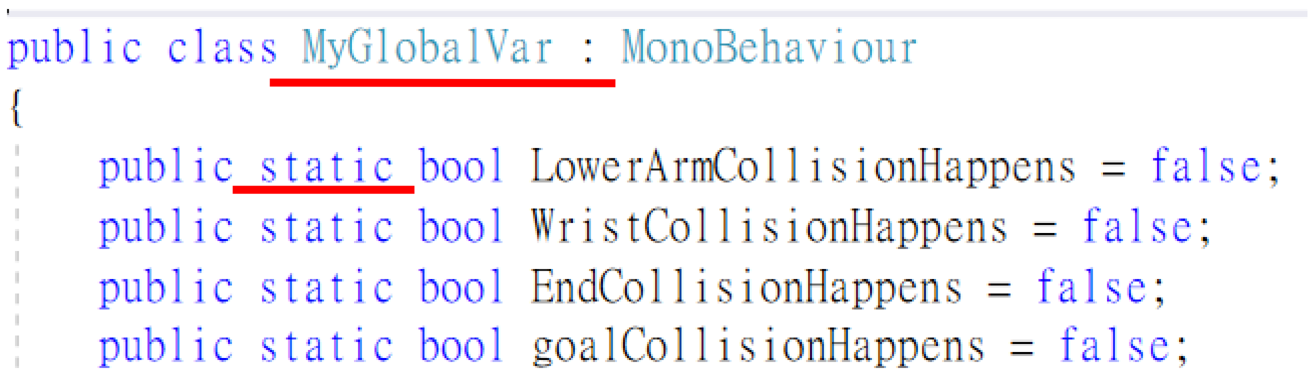
Decision Period 5

Take Actions Between De ✓

Default decision period =5

Static variables remember their values across scenes !

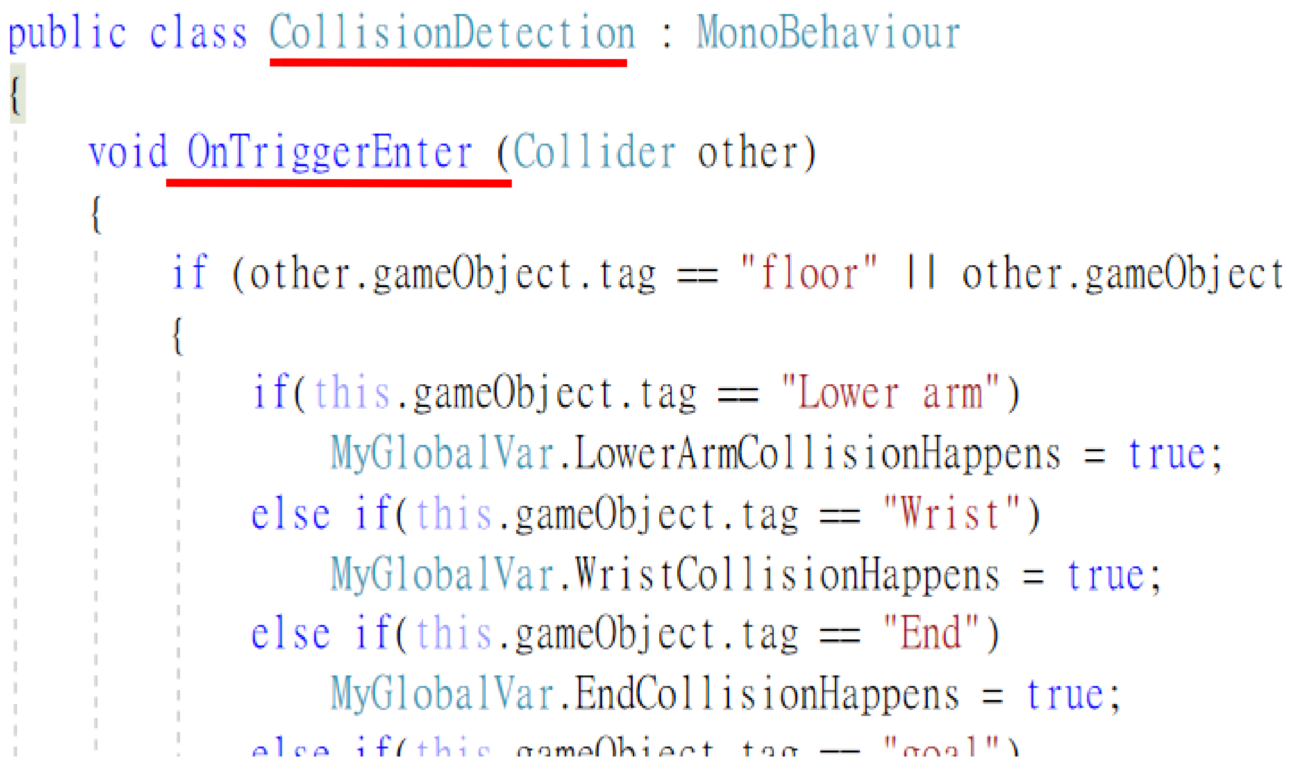
Static global variables to record collision of lower arm, wrist, end, and goal



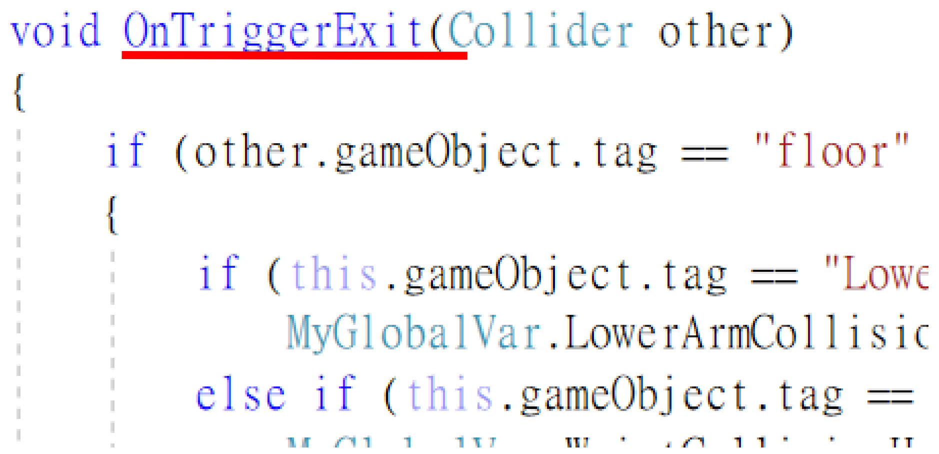
```
public class MyGlobalVar : MonoBehaviour
{
    public static bool LowerArmCollisionHappens = false;
    public static bool WristCollisionHappens = false;
    public static bool EndCollisionHappens = false;
    public static bool goalCollisionHappens = false;
```

Static variables remember their values across scenes. It may happen that one training environment sets it to True, but another scene set it back to False! That is why if we use one training environment, the collision detection works fine. But when we train with multiple training environments, the collision detection may have problems.

If you use multiple training scene, in one environment the red cube runs very fast.



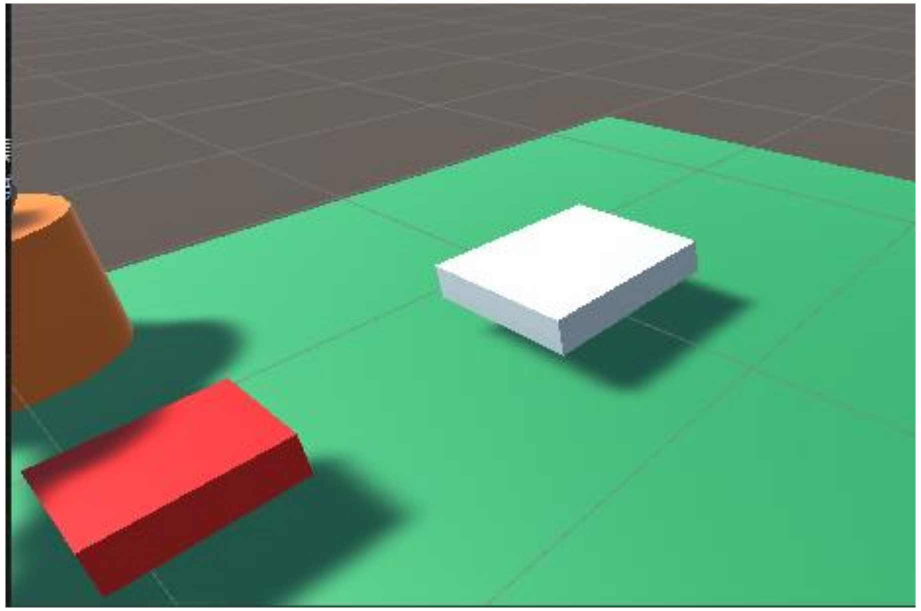
```
public class CollisionDetection : MonoBehaviour
{
    void OnTriggerEnter (Collider other)
    {
        if (other.gameObject.tag == "floor" || other.gameObject
        {
            if(this.gameObject.tag == "Lower arm")
                MyGlobalVar.LowerArmCollisionHappens = true;
            else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



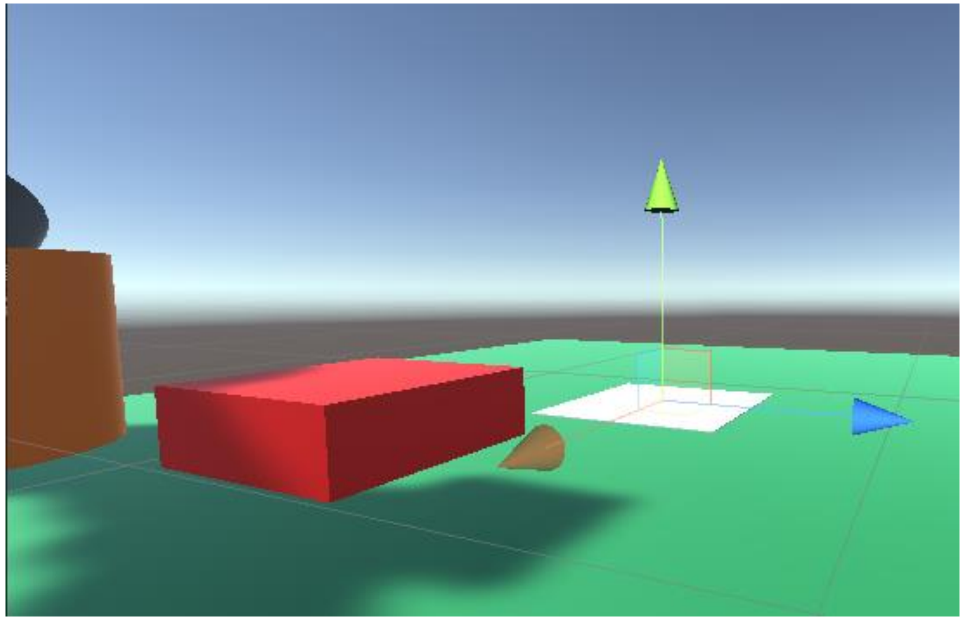
```
void OnTriggerExit(Collider other)
{
    if (other.gameObject.tag == "floor"
    {
        if (this.gameObject.tag == "Lowe
            MyGlobalVar.LowerArmCollisic
        else if (this.gameObject.tag ==
```

Point-based vs plane-based reach detection

Point-based touch detection



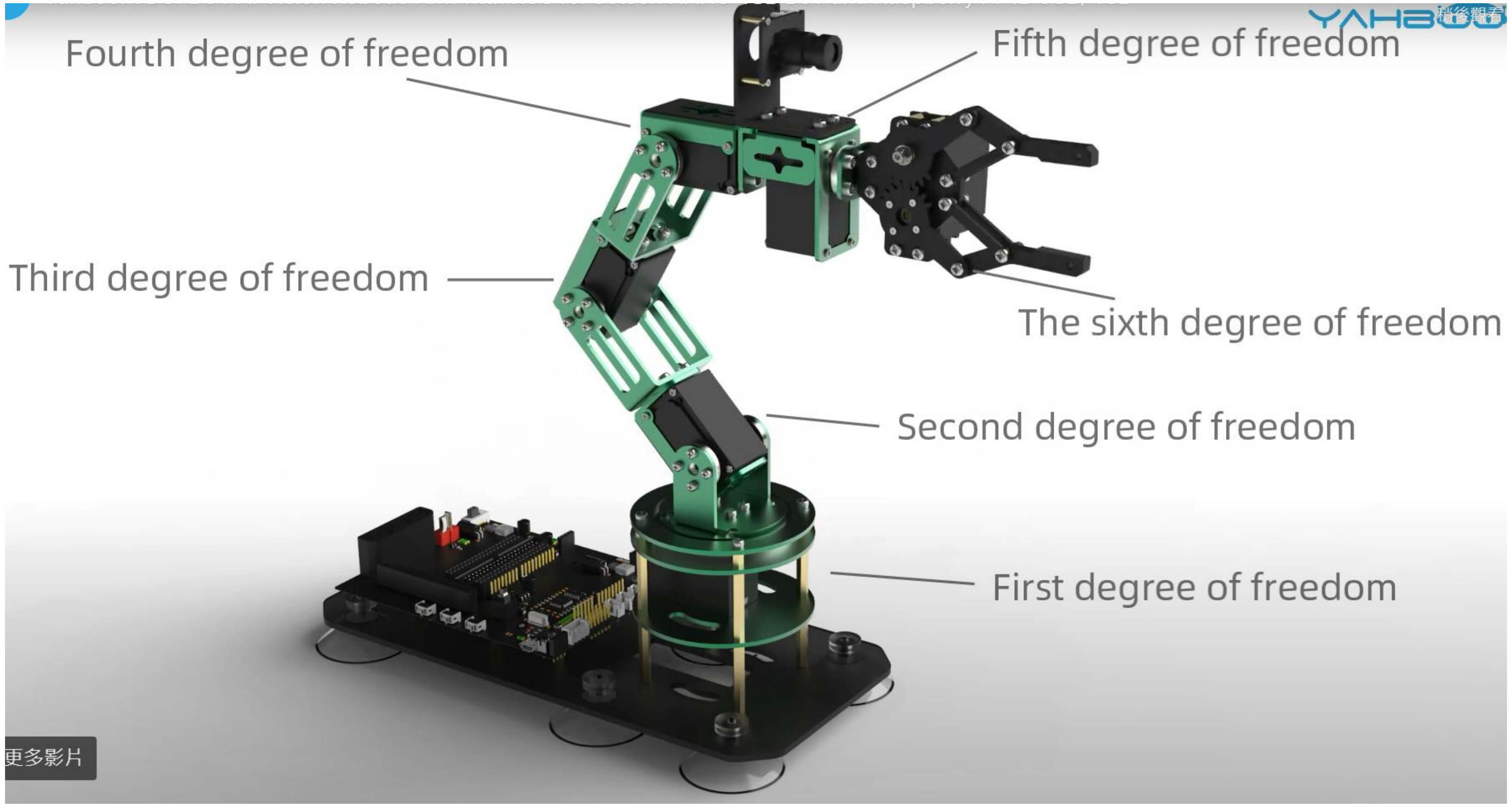
When both red and white cubes are randomly placed and only point-touch information is given (without contact face data) NN will fail to learn due to collisions.



White area does not have volume and NN can learn using point-touch information.

From VE to real robot

DOFBot



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