



[2-3주차] 3DBall Agent

인천정보과학고등학교 3학년 2학기 게임프로그래밍

© all right reserved 인천정보과학고 IT소프트웨어과 정수영

목차

- 프로젝트 생성 및 배치
- Agent 구성
- Ball3DAgent 스크립트
- Model 생성 및 테스트
- 더 해보기

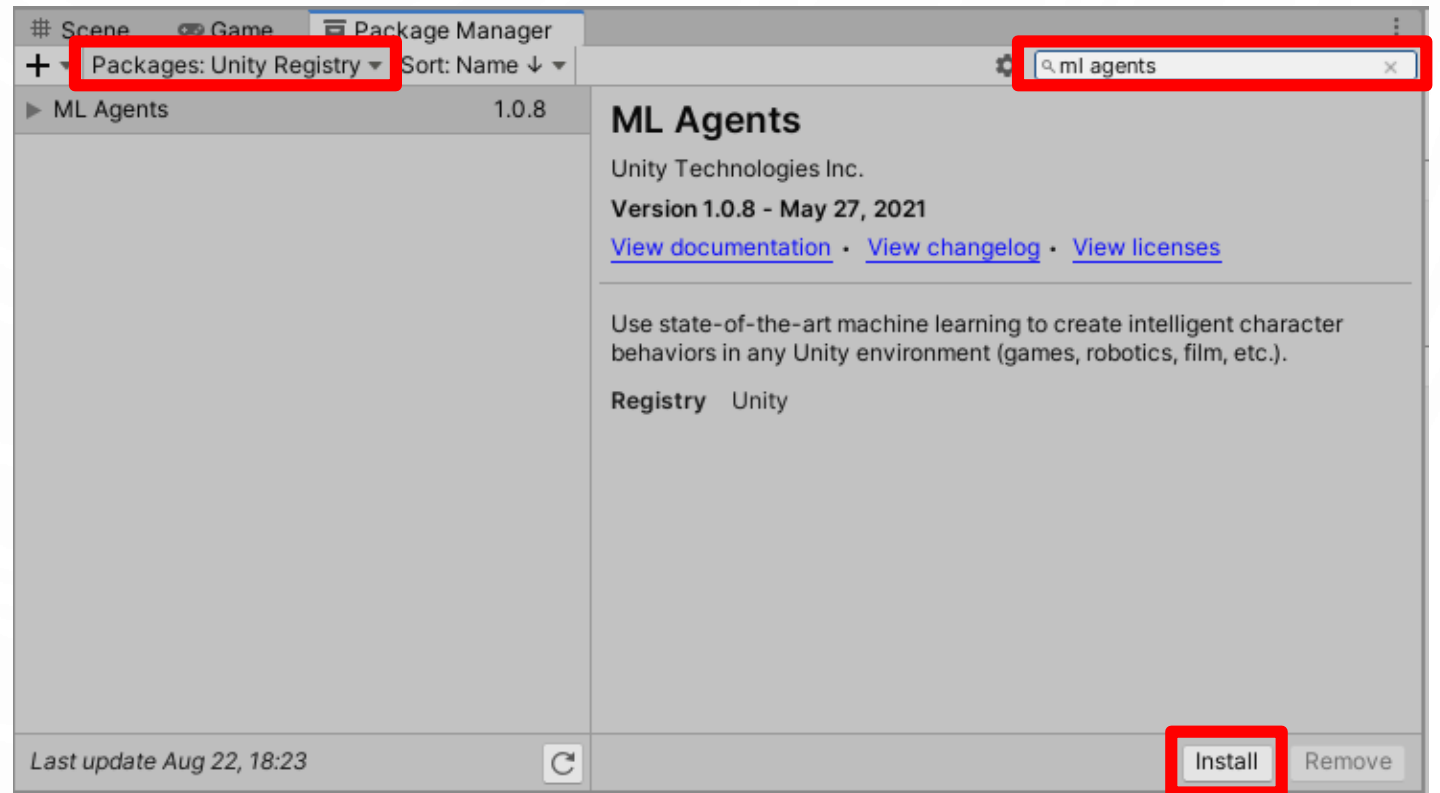
프로젝트 생성 및 배치

프로젝트 생성

- D/E 드라이브에 영문/숫자로만 된 폴더 만들기
 - 예) 30100hong
- Unity Editor를 실행하고 생성된 폴더 안에 프로젝트 생성
 - 프로젝트명 : 3DBall (영문/숫자로 자유롭게 가능)
 - 버전 : 2020.1.4f1
 - 종류 : 3D 프로젝트

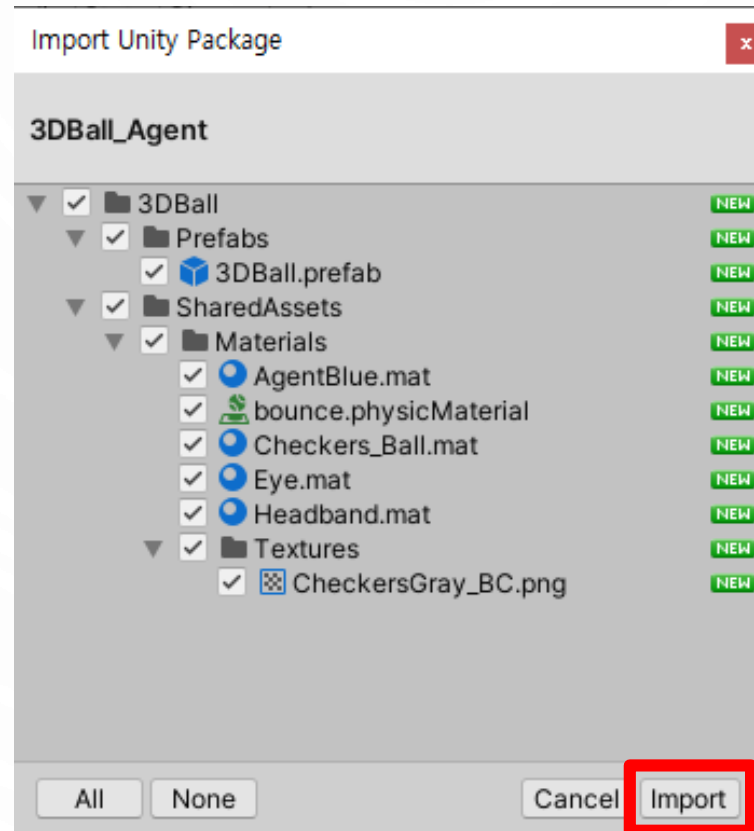
ML-Agents 패키지 설치

- Package Manager 창
- Packages: Unity Registry로 수정 후
- “ML-Agents” 검색
- 1.0.8 버전 확인 후
- 설치



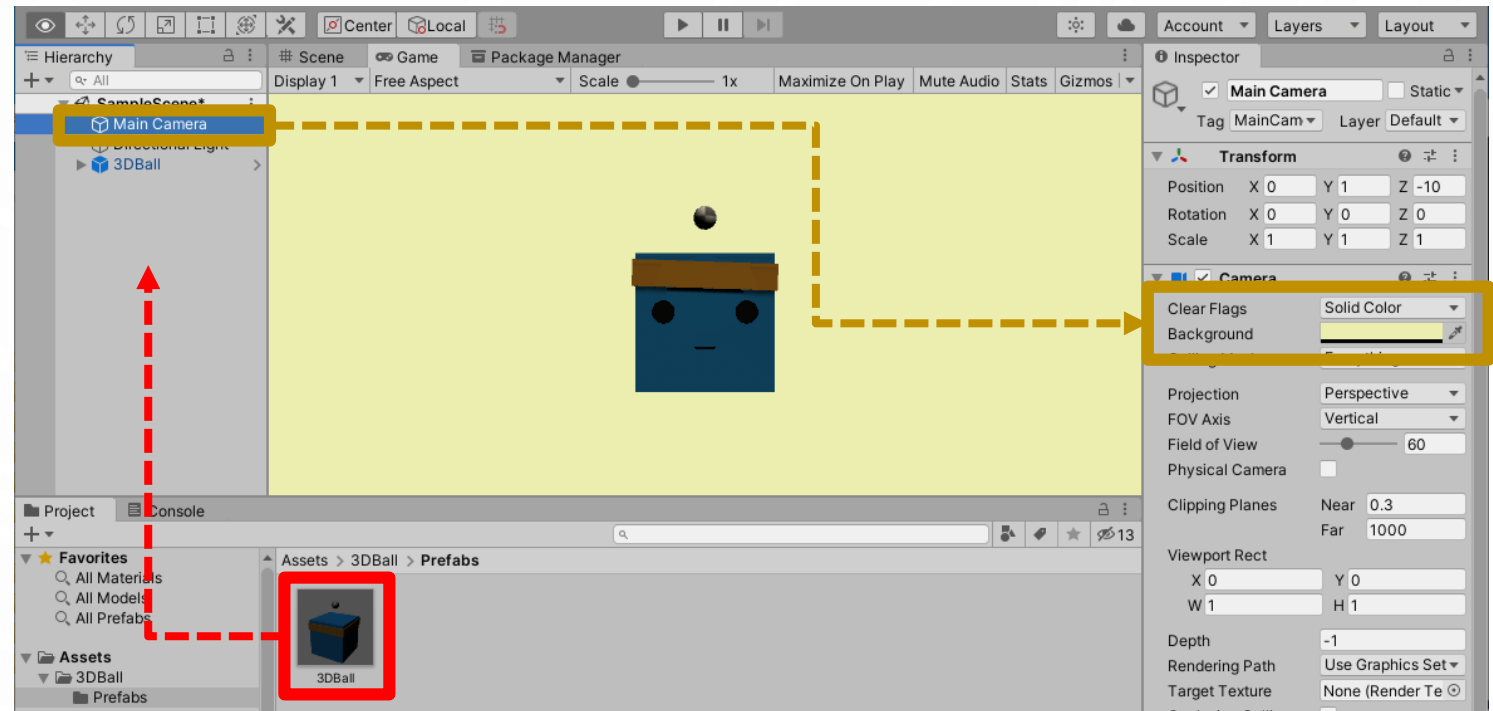
3DBall Agent 패키지 импорт

- 3DBall_Agent.unitypackage를 import
- Agent 모양만 들어 있으므로 그냥 자유롭게 본인이 만들어도 됨.



Agent 오브젝트 배치

- Assets/3DBall/Prefabs/3DBall 프리팹을 하이어라키뷰로 던지기
- Main Camera 오브젝트의 Camera 컴포넌트 수정
 - Clear Flags : Solid
 - Background : 맘대로



AGENT 구성

Agent 컴포넌트 설치

- 하이어라키뷰에서 3DBall 오브젝트 하위의 Agent 오브젝트 선택
- Add Component... 하여 다음을 컴포넌트 설치
 - Behavior Parameters
 - Decision Requester

Behavior Parameters 수정

- Behavior Parameter를 다음과 같이 수정(이유는 차차)
- 주의 문구는 Model이 없다는 이야기 인데 무시한다.

Behavior Parameters

Behavior Name: 3DBall

Vector Observation: Space Size: 8, Stacked Vectors: 1

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

Model: None (NN Model)

Inference Device: CPU

Behavior Type: Default

Team Id: 0

Use Child Sensors: ☒

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

대소문자 정확하게!!

BALL3DAGENT 스크립트

Ba113DAgent 스크립트 생성

- Assets 폴더 하위에 Scripts 폴더를 생성하고 그 안에 Ba113DAgent.cs를 만든다.
- 스크립트 Agent 오브젝트에 넣는다.

Ball3DAgent 스크립트 구조

```
class Ball3DAgent : Agent
{
    // 모두 Agent 클래스 내의 매서드를 override 함.
    Initialize() {} // 초기 세팅. Start()와 같음
    CollectObservations(VectorSensor) {} // 관찰하기. Input 벡터
    OnActionReceived(float[]) {} // 행동(Action). Output 벡터
    OnEpisodeBegin() {} // 각 에피소드 시작할 때 초기값 세팅
    Heuristic(float[]) {} // 인간이 조정해서 Output 값 바꿈
}
```

Ball3DAgent 스크립트 (1)

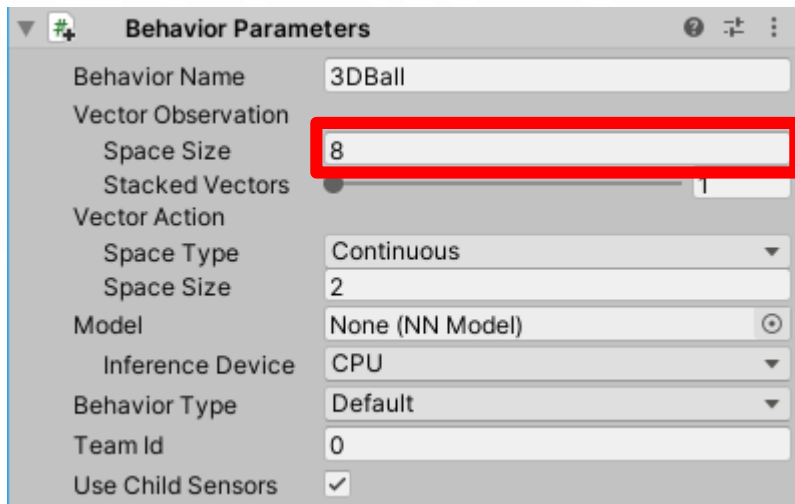
```
using UnityEngine;
using Unity.MLAgents;
using Unity.MLAgents.Sensors;

public class Ball3DAgent : Agent
{
    [Header("Specific to Ball3D")]
    public GameObject ball;
    Rigidbody m_BallRb;
    EnvironmentParameters m_ResetParams;

    public override void Initialize()
    {
        m_BallRb = ball.GetComponent<Rigidbody>();
        m_ResetParams = Academy.Instance.EnvironmentParameters;
        SetResetParameters();
    }
}
```

Ball3DAgent 스크립트 (2)

```
public override void CollectObservations(VectorSensor sensor)
{
    sensor.AddObservation(gameObject.transform.rotation.z);
    sensor.AddObservation(gameObject.transform.rotation.x);
    sensor.AddObservation(ball.transform.position - gameObject.transform.position);
    sensor.AddObservation(m_BallRb.velocity);
}
```



```
벡터1 = gameObject.transform.rotation.z  
벡터2 = gameObject.transform.rotation.x  
벡터3 = ball.transform.position - gameObject.transform.position x  
벡터4 = ball.transform.position - gameObject.transform.position y  
벡터5 = ball.transform.position - gameObject.transform.position z  
벡터6 = ball.transform.GetComponent().velocity x  
벡터7 = ball.transform.GetComponent().velocity y  
벡터8 = ball.transform.GetComponent().velocity z
```

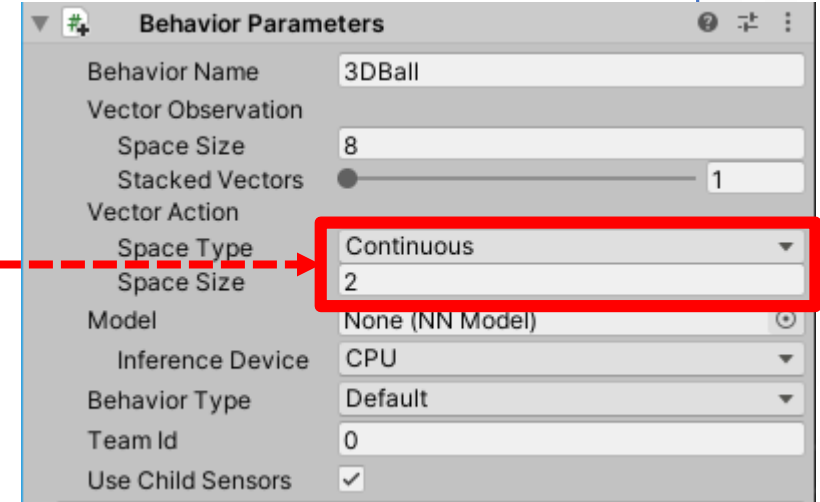
Ball3DAgent 스크립트 (3)

```
public override void OnActionReceived(float[] vectorAction)
{
    var actionZ = 2f * Mathf.Clamp(vectorAction[0], -1f, 1f);
    var actionX = 2f * Mathf.Clamp(vectorAction[1], -1f, 1f);

    if ((gameObject.transform.rotation.z < 0.25f && actionZ > 0f) ||
        (gameObject.transform.rotation.z > -0.25f && actionZ < 0f))
    {
        gameObject.transform.Rotate(new Vector3(0, 0, 1), actionZ);
    }

    if ((gameObject.transform.rotation.x < 0.25f && actionX > 0f) ||
        (gameObject.transform.rotation.x > -0.25f && actionX < 0f))
    {
        gameObject.transform.Rotate(new Vector3(1, 0, 0), actionX);
    }

    if ((ball.transform.position.y - gameObject.transform.position.y) < -2f ||
        Mathf.Abs(ball.transform.position.x - gameObject.transform.position.x) > 3f ||
        Mathf.Abs(ball.transform.position.z - gameObject.transform.position.z) > 3f)
    {
        SetReward(-1f);
        EndEpisode();
    }
    else
    {
        SetReward(0.1f);
    }
}
```



Ball3DAgent 스크립트 (4)

```
public override void OnEpisodeBegin()
{
    gameObject.transform.rotation = new Quaternion(0f, 0f, 0f, 0f);
    gameObject.transform.Rotate(new Vector3(1, 0, 0), Random.Range(-10f, 10f));
    gameObject.transform.Rotate(new Vector3(0, 0, 1), Random.Range(-10f, 10f));
    m_BallRb.velocity = new Vector3(0f, 0f, 0f);
    ball.transform.position =
        new Vector3(Random.Range(-1.5f, 1.5f), 4f, Random.Range(-1.5f, 1.5f))
        + gameObject.transform.position;
    SetResetParameters();
}

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

Ball3DAgent 스크립트 (5)

```
public void SetBall()
{
    m_BallRb.mass = m_ResetParams.GetWithDefault("mass", 1.0f);
    var scale = m_ResetParams.GetWithDefault("scale", 1.0f);
    ball.transform.localScale = new Vector3(scale, scale, scale);
}

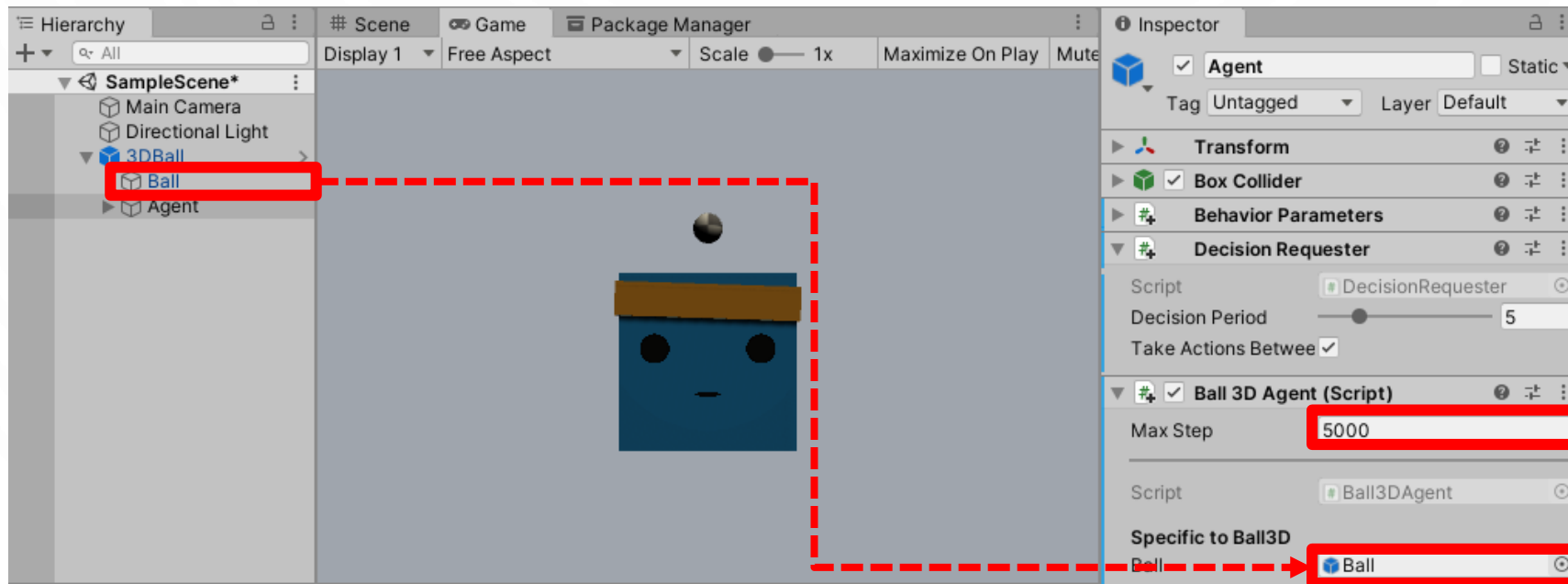
public void SetResetParameters()
{
    SetBall();
}
}
```

Ball3DAgent 스크립트 설정

- Agent (Script) 컴포넌트는 삭제 한다.



- Ball 3D Agent (Script) 컴포넌트를 수정한다.



MODEL 생성 및 테스트

3DBall.yaml 생성

- .yaml은 .xml과 .json과 같이 데이터를 전송할 때 사용되는 포맷

Xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
  <apiVersion>v1</apiVersion>
  <kind>Pod</kind>
  <metadata>
    <name>hello-pod</name>
    <labels>
      <app>hello</app>
    </labels>
  </metadata>
  <spec>
    <containers>
      <name>hello-container</name>
      <image>tmkub/hello</image>
      <ports>
        <containerPort>8000</containerPort>
      </ports>
    </containers>
  </spec>
</root>
```

Object (metadata), Array (containers), Key (containerPort), Value (8000)

Json

```
{
  "apiVersion": "v1",
  "kind": "Pod",
  "metadata": {
    "name": "hello-pod",
    "labels": {
      "app": "hello"
    }
  },
  "spec": {
    "containers": [
      {
        "name": "hello-container",
        "image": "tmkub/hello",
        "ports": [
          {
            "containerPort": 8000
          }
        ]
      }
    ]
  }
}
```

Object (metadata), Array (containers), Key (containerPort), Value (8000)

Yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: hello-pod
  labels:
    app: hello
spec:
  containers:
    - name: hello-container
      image: tmkub/hello
      ports:
        - containerPort: 8000
```

Object (metadata), Array (containers), Key (containerPort), Value (8000)

출처 : <https://www.infllearn.com/questions/16184>

3DBall.yaml

default:

```
trainer: ppo
batch_size: 1024
beta: 5.0e-3
buffer_size: 10240
epsilon: 0.2
hidden_units: 128
lambda: 0.95
learning_rate: 0.003
learning_rate_schedule: linear
max_steps: 10000
memory_size: 128
normalize: false
num_epoch: 3
num_layers: 2
time_horizon: 64
sequence_length: 64
summary_freq: 10000
use_recurrent: false
vis_encode_type: simple
reward_signals:
  extrinsic:
    strength: 1.0
    gamma: 0.99
```

3DBall:

```
normalize: true
batch_size: 16
buffer_size: 120
summary_freq: 120
time_horizon: 100
lambda: 0.99
beta: 0.01
```

완성된 3DBall.yaml은 바탕화면에 저장한다.

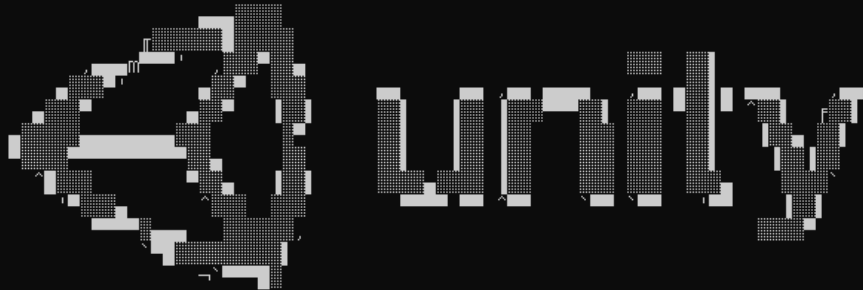
Tensorflow 모델 생성

- 윈도우 검색창에 Anaconda prompt 검색 후 열기
- 가상 환경 생성
 - `conda create -n my3dball`
- 가상 환경 리스트 확인
 - `conda env list`
- 가상 환경 활성화
 - `conda activate my3dball`

M1-Agents 학습하기

```
mlagents-learn "C:\Users\User\Desktop\3DBall.yaml" --run-id 3DBall-30100-v1
```

```
(my3dball) C:\Users\User>mlagents-learn "D:\project\3DBall_Sample\My3DBall.yaml" --run-id 3DBall-30100-v1
2021-08-22 19:47:33.373149: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library cudart64_110.dll
WARNING:tensorflow:From d:\program files\anaconda3\lib\site-packages\tensorflow\python\compat\v2_compat.py:96: disable_resource_variables (from tensorflow.python.ops.variable_scope) is deprecated and will be removed in a future version.
Instructions for updating:
non-resource variables are not supported in the long term

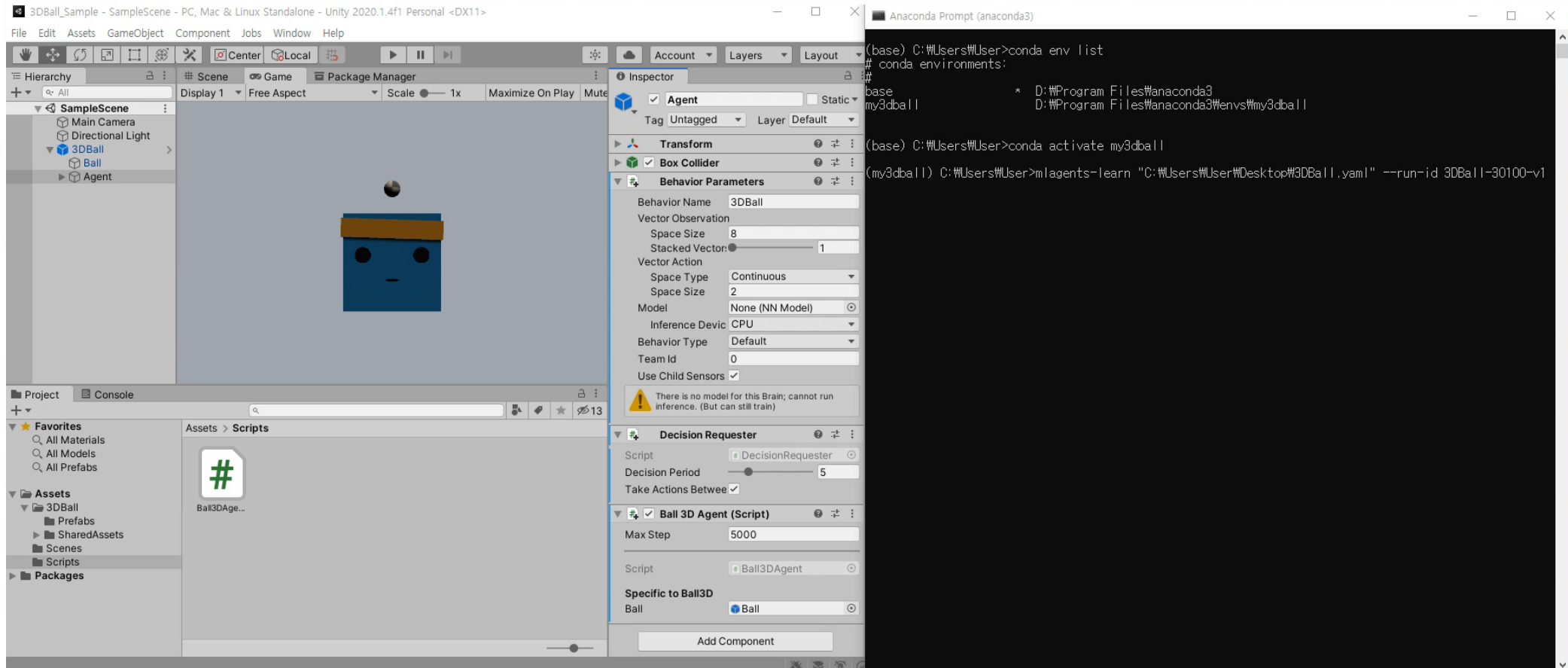


Version information:
ml-agents: 0.16.1,
ml-agents-envs: 0.16.1,
Communicator API: 1.0.0,
TensorFlow: 2.5.0
2021-08-22 19:47:36.998797: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library cudart64_110.dll
WARNING:tensorflow:From d:\program files\anaconda3\lib\site-packages\tensorflow\python\compat\v2_compat.py:96: disable_resource_variables (from tensorflow.python.ops.variable_scope) is deprecated and will be removed in a future version.
Instructions for updating:
non-resource variables are not supported in the long term
2021-08-22 19:47:39 INFO [environment.py:200] Listening on port 5004. Start training by pressing the Play button in the Unity Editor.
```

모델이 저장되는 폴더명
에러가 나면 v2, v3와 같이
이름을 수정해서 넣어준다.

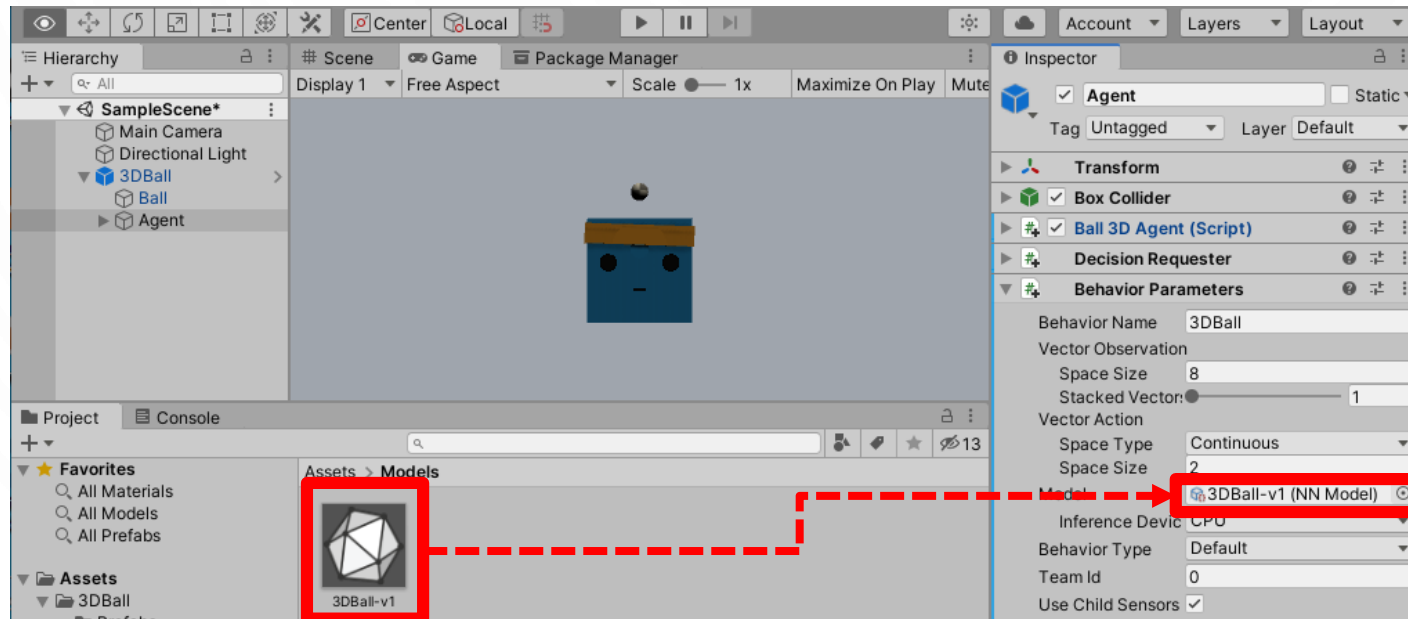
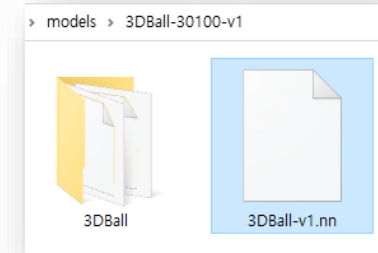
이런 문구가 뜨면 유니티로
돌아가 play 버튼을 누른다.

TFModel1 컴파일

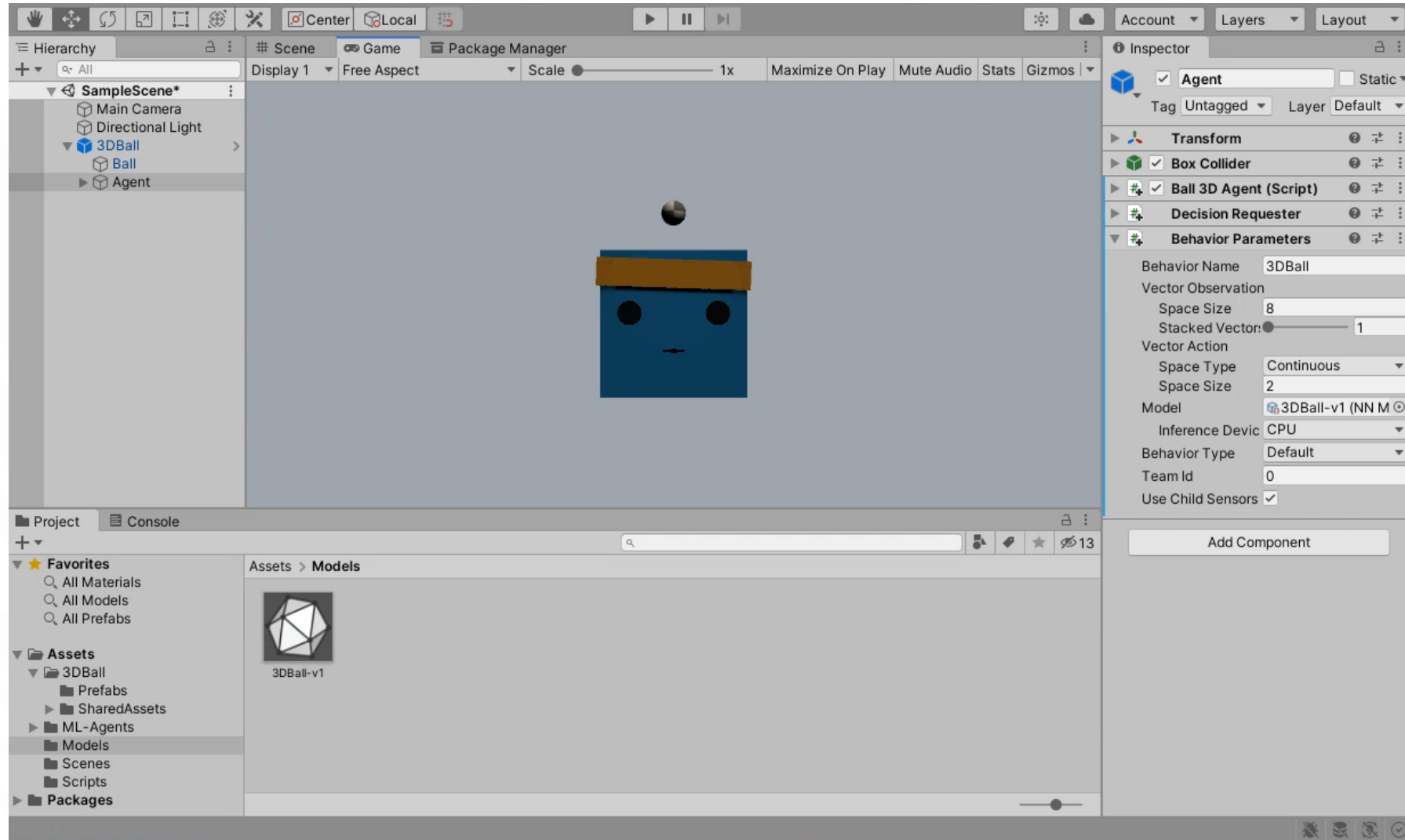


모델 적용하기

- C:\Users\User\models\3DBall-30100-v1 로 이동한다.
- 3DBall.nn을 3DBall-v1.nn으로 수정한다.
- Assets 내부에 Models 폴더 만들어 넣기
- Agent의 Behavior Parameters의 Model에 적용



모델 테스트



더 해보기

- 3DBa11.yaml에서 max_steps 값을 더 다르게 해서 새로운 모델(v2)을 만들어 적용해보자.
- 3DBa11을 여러 개 만들어서 한꺼번에 학습시켜 새로운 모델(v3)을 만들어 적용해보자.
- 만든 모델을 교사에게 제출하면 됩니다.
- 수고하셨습니다!

END

다음 시간부터는 직접 3DBALL을 구현해보겠습니다.