Handmade RL - Second Story

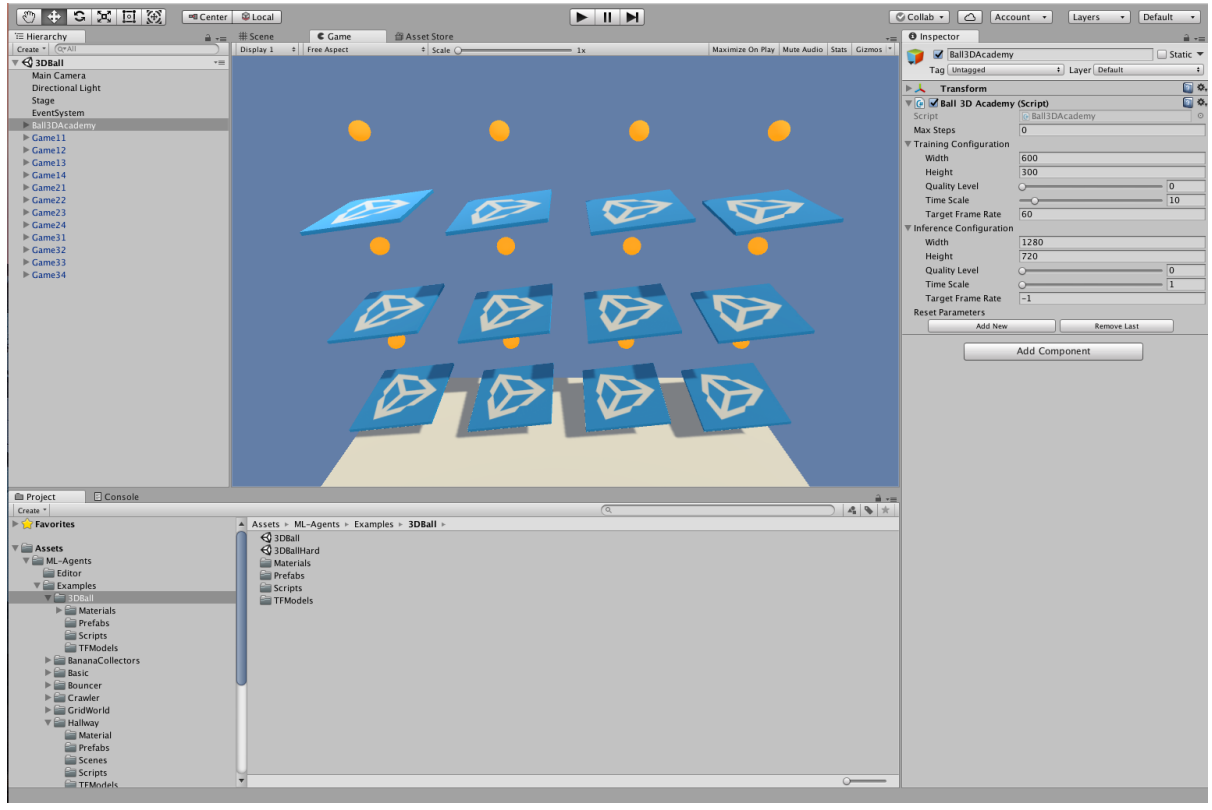
환경구성

Wonseok Jung

3D Balance Ball Environment 실행

1. 아래의 링크를 따라 ml-agents를 다운로드 받는다.
   https://github.com/Unity-Technologies/ml-agents

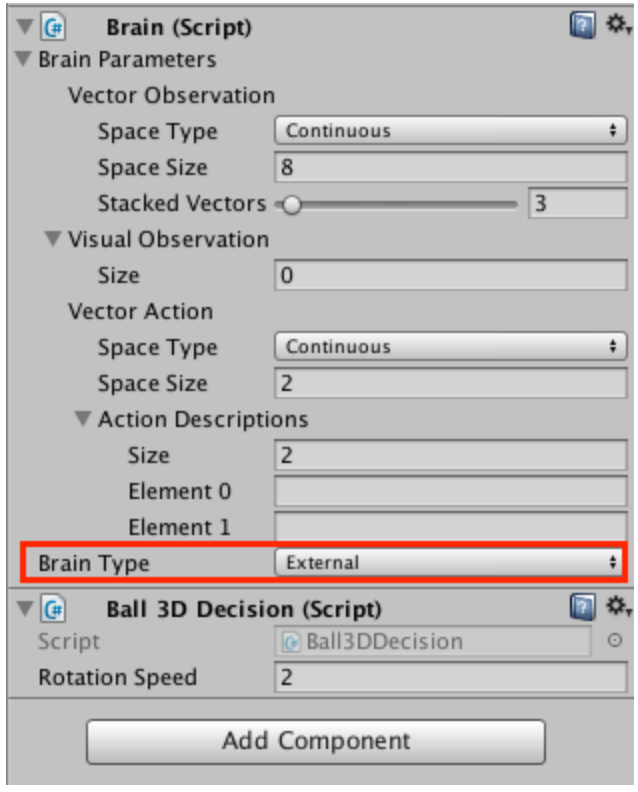
2. \ml-agents-master\unity-environment\Assets\ML-Agents\Examples\3DBall 의 경로에서 3DBall Scene파일을 실행시킨다.



3. Hierarchy 에서 Ball3DBrain 을 검색

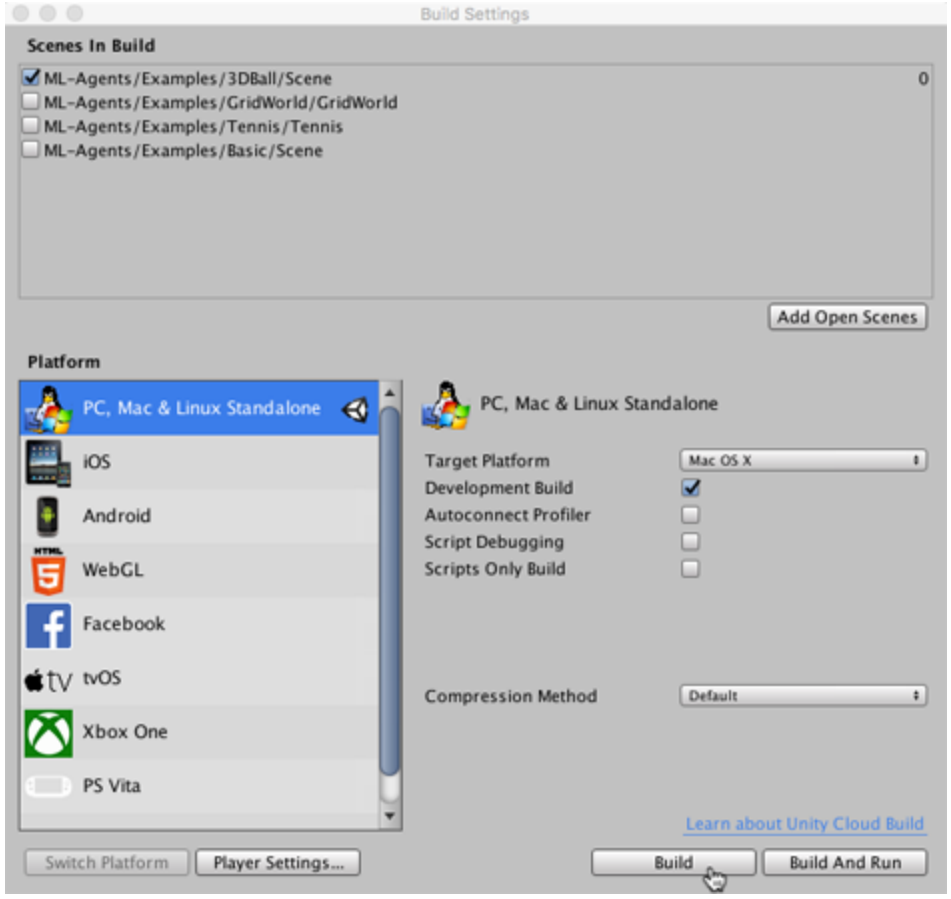
4. Ball3Dbrain double click

5. Brain(script)에서 BrainType을 External로 설정한다.



- Set up scene to play correctly when the training process launches our environment executable
    - Player setting을 연다. Edit-> Project stteings ->player
    - Resolution and Presentation :
    - Run in Background 체크
    - Display Resoution Dialog : Disabled

- File-> Build Setting 를 연다
- 3DBall의 scene만 체크
- Build Click
- ml-agent 디렉토리의 python 폴더를 선택한뒤 save



Training the Brain with Reinforcement Learning

- Use Proximal Policy Optimization (PPO) to train an agent
- learn.py : wrapper script 제공
- command line이동:

  ```
  python3 python/learn.py <env_file_path> --run-id=<run-identifier> --train
  ```

- Tensorflow 최신버전에서 에러가 났음 : pip install tensorflow==1.5.0으로 해결

Training with PPO

- ml-agent 디렉토리에서 커맨드창을 열고 다음의 명령어 입력

```
python3 python/learn.py <env_file_path> --run-id=<run-identifier> --train
```

또는

```
python python/learn.py <env_file_path> --run-id=<run-identifier> --train
```

- env_file_path : 위에서 build한 3Dball 파일의 경로
    - example :

      ```
      python python/learn.py C:\Users\wonseok\Desktop\ml-agents-master\3Dball/3dball --run-id=test --train
      ```

Observing Training Progress

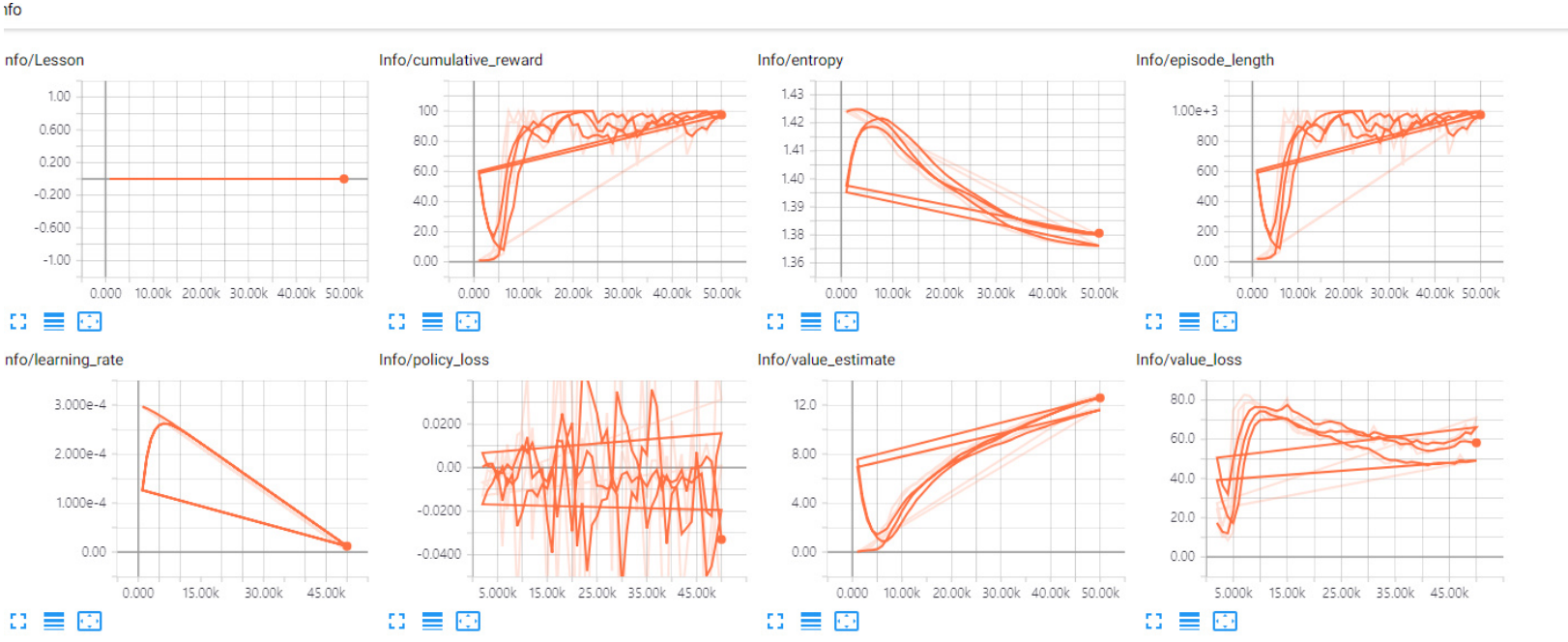
- After you start your training using learn.py, ml-agent folder will contain a summaries directory.
- You can use tensorboard in order to observe the training process.
- Run this code

```
tensorboard --logdir=summaries
```

Using Tensorflow

- 위의 ``tensorboard --logdir=summaries` 커맨드를 하면 local host의 주소가 나온다.
- 예 : `http://DESKTOP-5NM5TIB:6006`
- 그 주소를 웹브라우저 주소창에 입력하면 Tensorboard가 나오며, 훈련과정을 그래프로 볼수 있다.

- 현재 훈련과정


ıfo
nfo/Lesson
1.00
0.600
0.200
-0.200
-0.600
-1.00
0.000 10.00k 20.00k 30.00k 40.00k 50.00k
Info/cumulative_reward
100
80.0
60.0
40.0
20.0
0.00
0.000 10.00k 20.00k 30.00k 40.00k 50.00k
Info/entropy
1.43
1.42
1.41
1.40
1.39
1.38
1.36
0.000 10.00k 20.00k 30.00k 40.00k 50.00k
Info/episode_length
1.00e+3
800
600
400
200
0.00
0.000 10.00k 20.00k 30.00k 40.00k 50.00k
nfo/learning_rate
3.000e-4
2.000e-4
1.000e-4
0.00
0.000 15.00k 30.00k 45.00k
Info/policy_loss
0.0200
0.00
-0.0200
-0.0400
5.000k 15.00k 25.00k 35.00k 45.00k
Info/value_estimate
12.0
8.00
4.00
0.00
0.000 10.00k 20.00k 30.00k 40.00k 50.00k
Info/value_loss
80.0
60.0
40.0
20.0
0.00
5.000k 15.00k 25.00k 35.00k 45.00k

그래프 설명

- Lesson - only interesting when performing curriculum training. This is not used in the 3D Balance Ball environment.
- Cumulative Reward - The mean cumulative episode reward over all agents. Should increase during a successful training session.

- Entropy - How random the decisions of the model are. Should slowly decrease during a successful training process. If it decreases too quickly, the beta hyperparameter should be increased.
- Episode Length - The mean length of each episode in the environment for all agents.
- Learning Rate - How large a step the training algorithm takes as it searches for the optimal policy. Should decrease over time.

- Policy Loss - The mean loss of the policy function update. Correlates to how much the policy (process for deciding actions) is changing. The magnitude of this should decrease during a successful training session.
- Value Estimate - The mean value estimate for all states visited by the agent. Should increase during a successful training session.
- Value Loss - The mean loss of the value function update. Correlates to how well the model is able to predict the value of each state. This should decrease during a successful training session.

Embedding the Trained Brain into the Unity Environment (Experimental)

- 트레이닝이 완료되면 모델이 save된다.
- 그 save된 모델을 사용하여 agent의 Internal brain type으로 사용할수 있다.

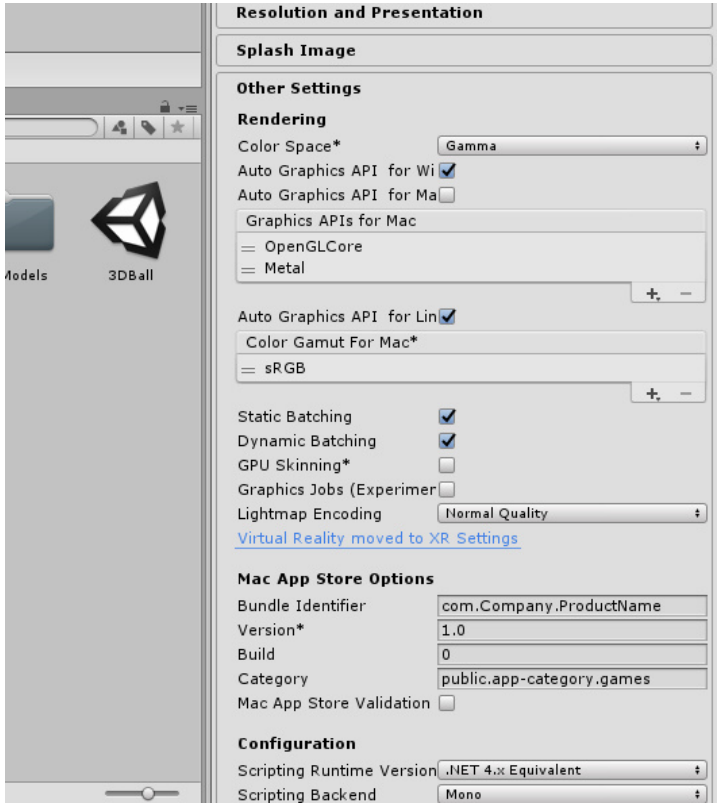
Setting up TensorFlowSharp Support

- 먼저 TensorFlowSharp를 설치하여야 한다. 다음의 링크를 통하여 받을수 있다.

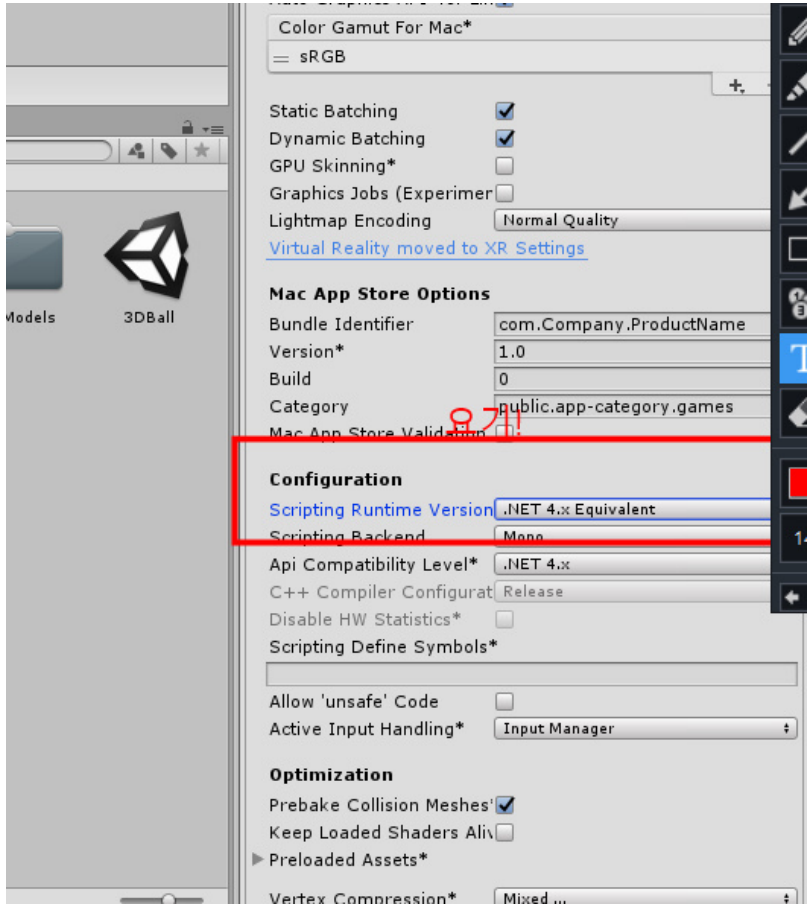
https://s3.amazonaws.com/unity-ml-agents/0.3/TFSharpPlugin.unitypackage

- 다운받은 파일을 실행한다
- 패키지를 import한다.

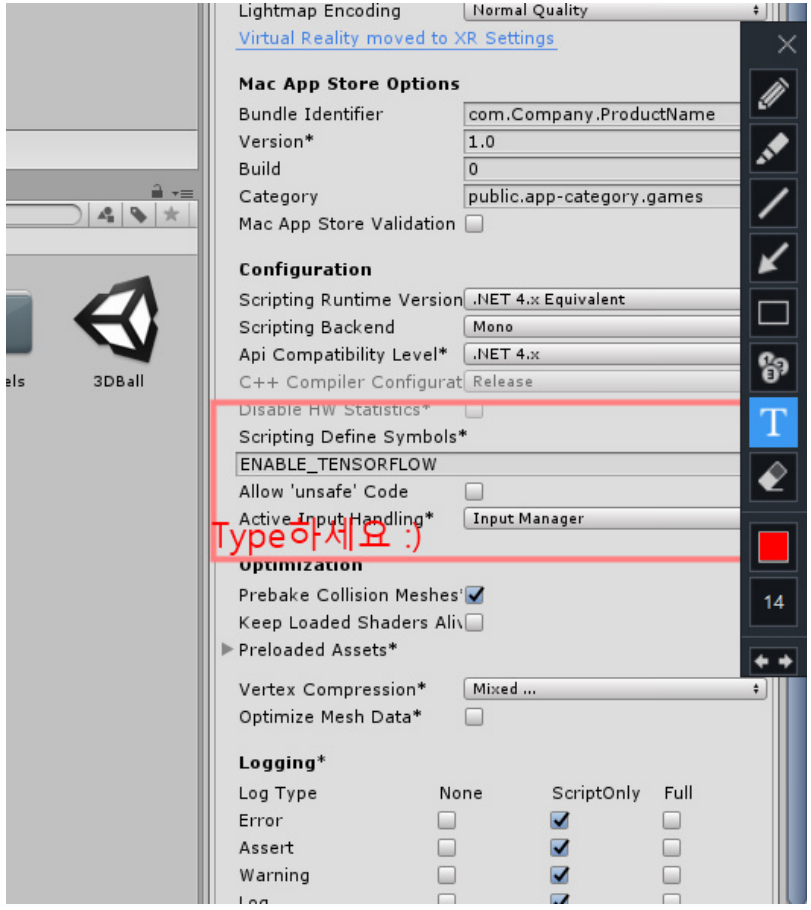
- Edit -> Project Settings -> Player
- For each of the platforms you target (PC, Mac and Linux Standalone, iOS or Android):
  Go into Other Settings.



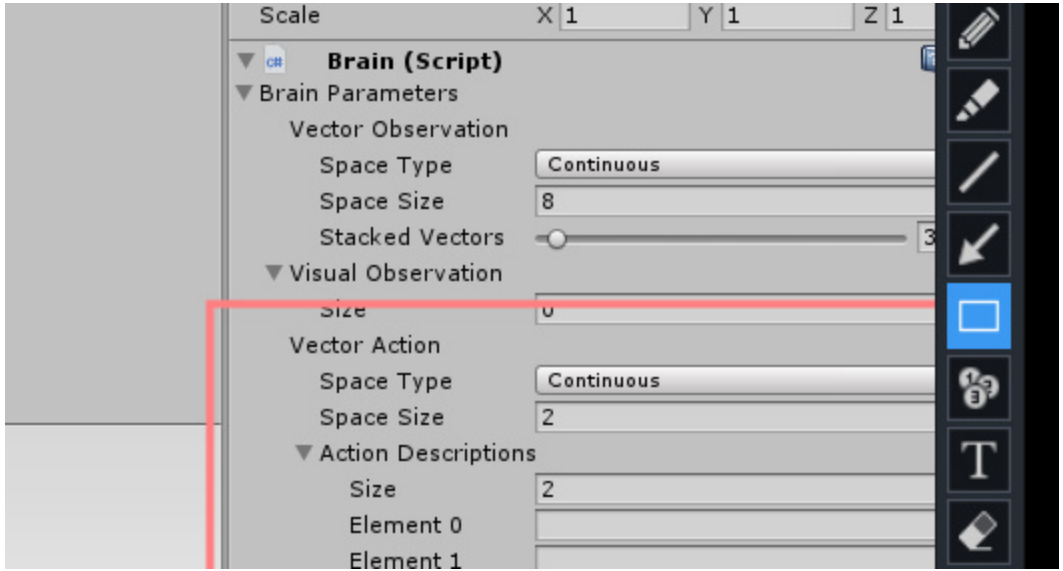
- Select Scripting Runtime Version to Experimental (.NET 4.6 Equivalent)



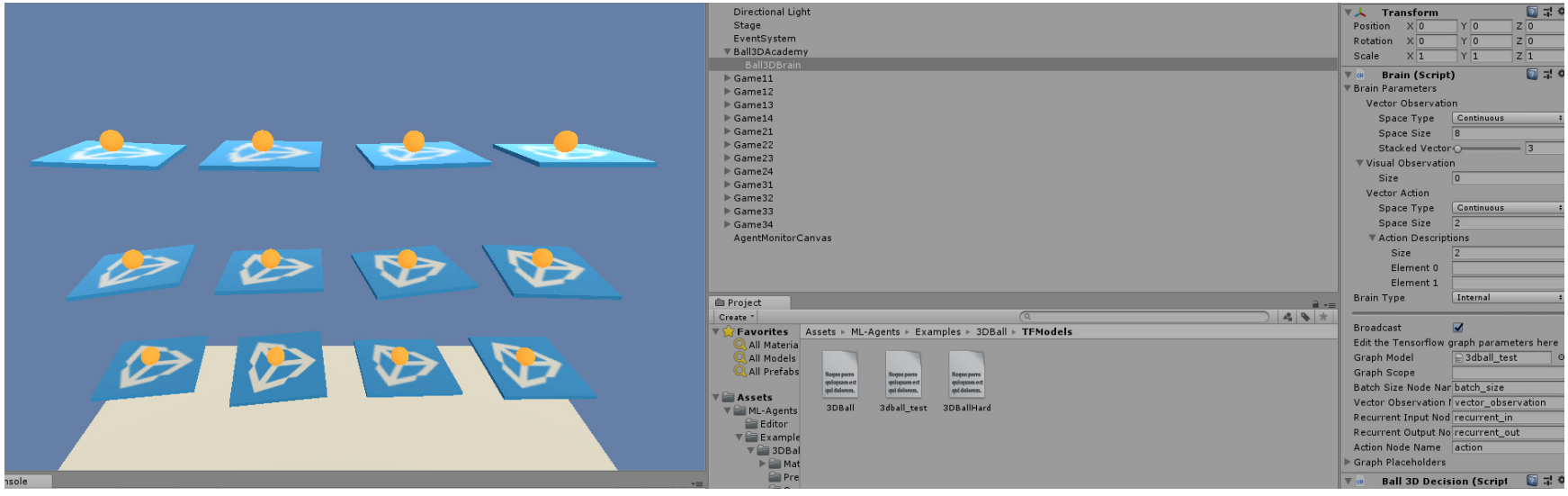
- In Scripting Defined Symbols, add the flag ENABLE_TENSORFLOW. After typing in, press Enter.



- Go to File -> Save Project
- Restart the Unity Editor.
- The trained model is stored in models/ in the ml-agents folder. Once the training is complete, there will be a <env_name>.bytes file in that location where <env_name> is the name of the executable used during training.
- Move <env_name>.bytes from python/models/ppo/ into unity-environment/Assets/ML-Agents/Examples/3DBall/TFModels/.


Scale
X 1
Y 1
Z 1
Brain (Script)
Brain Parameters
Vector Observation
Space Type
Continuous
Space Size
8
Stacked Vectors
Visual Observation
Vector Action
Space Type
Continuous
Space Size
2
Action Descriptions
Size
2
Element 0
Element 1

- Graph Model에서 3Dball을 트레이닝 시킨 <env_name>.bytes 모델로 변경



성공!!!

References

https://github.com/Unity-Technologies/ml-agents/blob/master/docs/Getting-Started-with-Balance-Ball.md

https://github.com/wonseokjung