

# Unity로 Machine learning Agent 만들기 :

## First story : 기본구성 설명

wonseok Jung

## **Getting Started with the 3D Balance Ball Environment**

# Understanding a Unity Environment (3D Balance Ball)

- An agent observes and interacts with an environment
- In Unity, an environment contains :
  - Academy
  - One or more Brain
  - Agent objects



# 1. Academy

- The Academy object for the scene is placed on the Ball3DAcademy Gameobject
- Several properties that control how the environment works
  - Training and Inference Configuration : set the graphics and timescale
  - Training Configuration : academy uses it during training
  - Inference Configuration : when not training

## 1.1 Setting graphics and time

- Training configuration : low graphics quality, high time scale
- Inference Configuration : High graphics quality, time scale 1.0
- Observing the environment during training
  - Adjust the Inference Configuration : use larger window, timescale closer to 1:1

## 1.2 Three functions

- There are three functions you can implement
  1. `Academy.InitializeAcademy()` : Called once when the environment is launched
  2. `Academy.AcademyStep()` : Called at every simulation step before `Agent.AgentAction()` and after the agents collect observation
  3. `Academy.AcademyReset()` : when Academy starts or restarts simulations

## 2. Brain

- Brain doesn't store information about an agent
- Routes the agent's collected observations to the decision making process and returns the chosen action to the agent
- All agents can share the same brain, but act differently

## 2.1 Type of Brains

- Brain Type : how an agent makes its decisions
  - External type : when you train your agents
  - Internal type : when you use the trained model
  - Heuristic brain : allow you to handcode the agent's logic
  - Player brain : lets you map keyboard commands to actions,

You can also implement your own type of brain



### 3. Vector Observation Space

- ML-Agents classifies vector observation into two types:
  - Continuous : vector of floating point of numbers
  - Discrete : index into a table of states

## 4. Vector Action Space

- An agent is given instructions from the brain in the form of actions.
- Two types of action:
  - Continuous :vector of numbers
  - Can vary continuously (force, torque)
  - Discrete : action space defines its actions as a table(index to this table)

## 5. Agent

- Agent is the actor that observes and takes action in the Environment
- Agent object has few properties that affect behavior:
  - Brain : Every agent must have a brain. Brain determines how an agent make decisions
  - Visual observations - Camera objects, used by agent to observe the environment.
  - Max Step - How many simulation steps can occur before the agent decides it is done.
  - Reset On Done - Defines whether an agent over when it is finished

## 5. Agent subclass implementation

- `Agent.AgentReset()` - When the Agent resets and beginning of a session.
- `Agent.CollectObservations()` - Called every simulation step, Collecting the agent's observation
- `Agent.AgentAction()` - Called every simulation step, receive
- action chosen by the brain