Rise of the **Reinforcement Learning**

Wonseok Jung

소개

정원석

뉴욕시립대 - Baruch college (Data Science Major)

ConnexionAI Researcher

CTRL (Contest in RL) 리더

DeepLearningCollege 강화학습 연구원

Project : Object Detection, Chatbot, Reinforcement Learning

Github:
https://github.com/wonseokjung

Facebook:
https://www.facebook.com/ws.jung.798

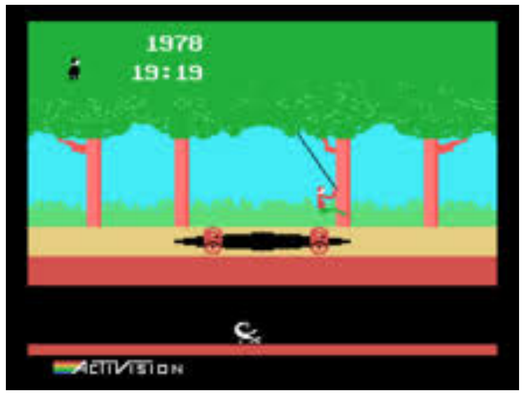
Blog:
https://wonseokjung.github.io/

목차

CREATING ENVIRONMENTS

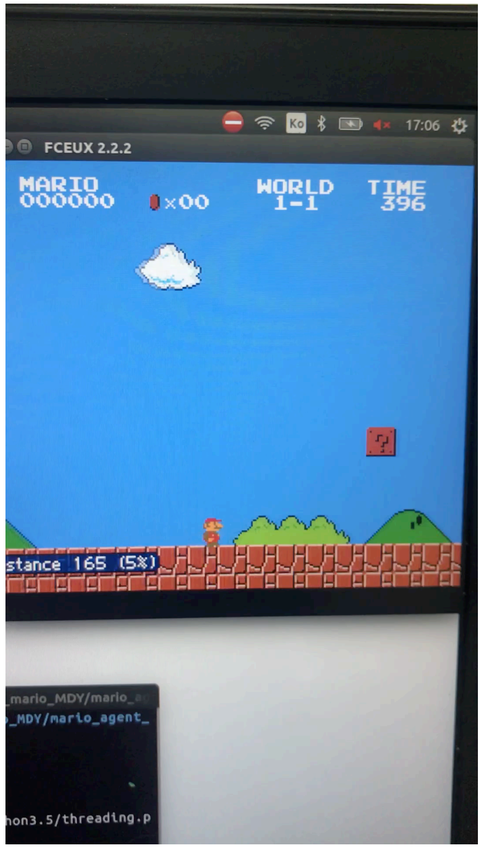
OpenAI-gym

DQN



Supermario

DDQN(tuned)

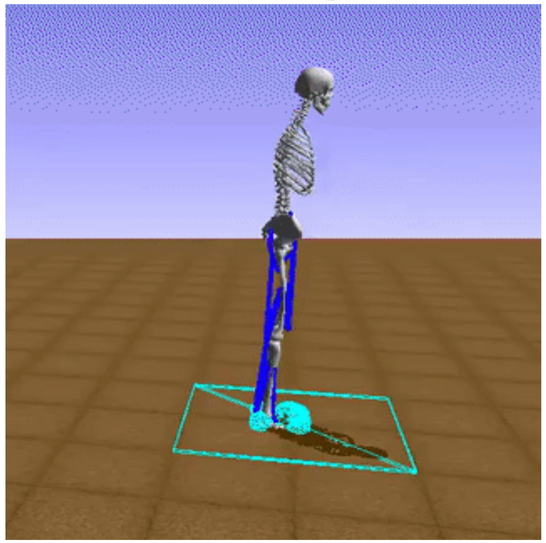


Sonic

Rainbow DQN(tuned)

OpenSim

DDPG



- 여러가지 환경에서 그 환경에 맞는 강화학습 알고리즘을 적용해 보았다.
- 이 과정에서 여러가지 이슈가 발생했다.

QUESTIONS

Issues :

1. 학습시간이 너무너무 오래걸린다.

I 8, 1080기준 :

OpenAI GYM : 최소 5분 ~ 일주일 이상

SuperMario Level 1 : 6일 소요

Sonic : OpenAI 제공 서버 사용 : 7시간

Prosthetics : 1달 이상 예상

2. 강화학습은 학습이 필수다.

제공하는 환경만 사용하는것이 가능하다.

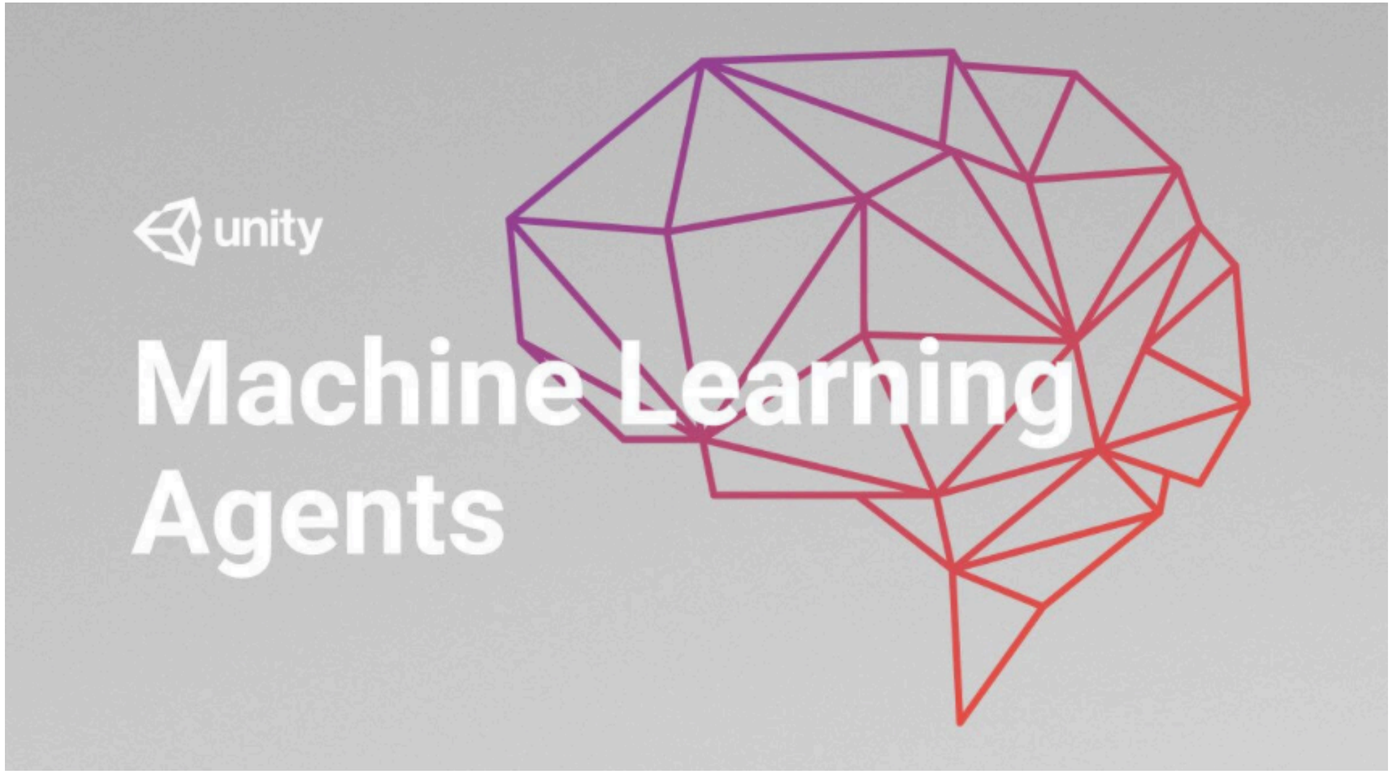
질문 :

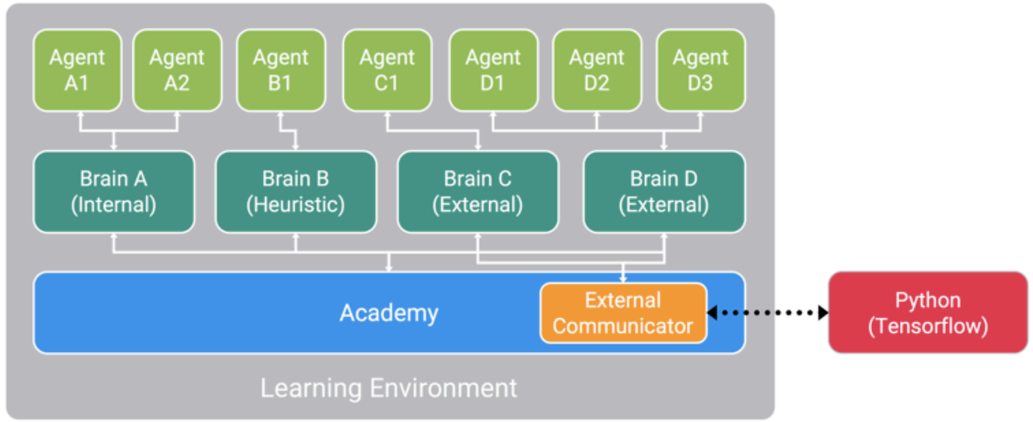
주어진 환경이 아닌 내가 궁금한 문제를 풀기 위해

환경을 만드는 것이 가능할까?

학습시간을 줄일수 있는 방법이 있을까?

UNITY ML-AGENTS





- Unity를 사용하여 개인이 강화학습 환경을 만들수 있다.
- 또한 Machine Learning Agents 기능으로 학습을 보다 효과적이게 하는것이 가능하다.

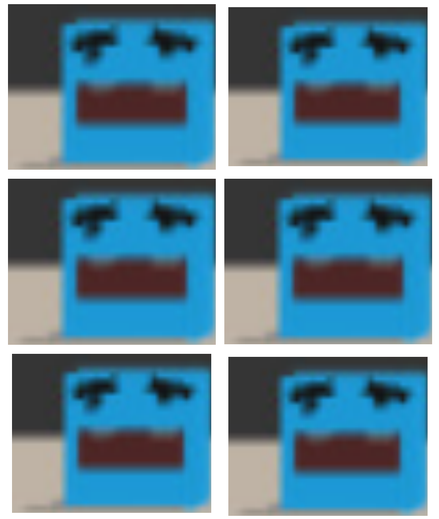
MULTI-AGENT ENVIRONMENT

MULTI-AGENTS?

-Intelligent human agents(사람)은 사회에서 다른 agents와 정보를 공유한다.

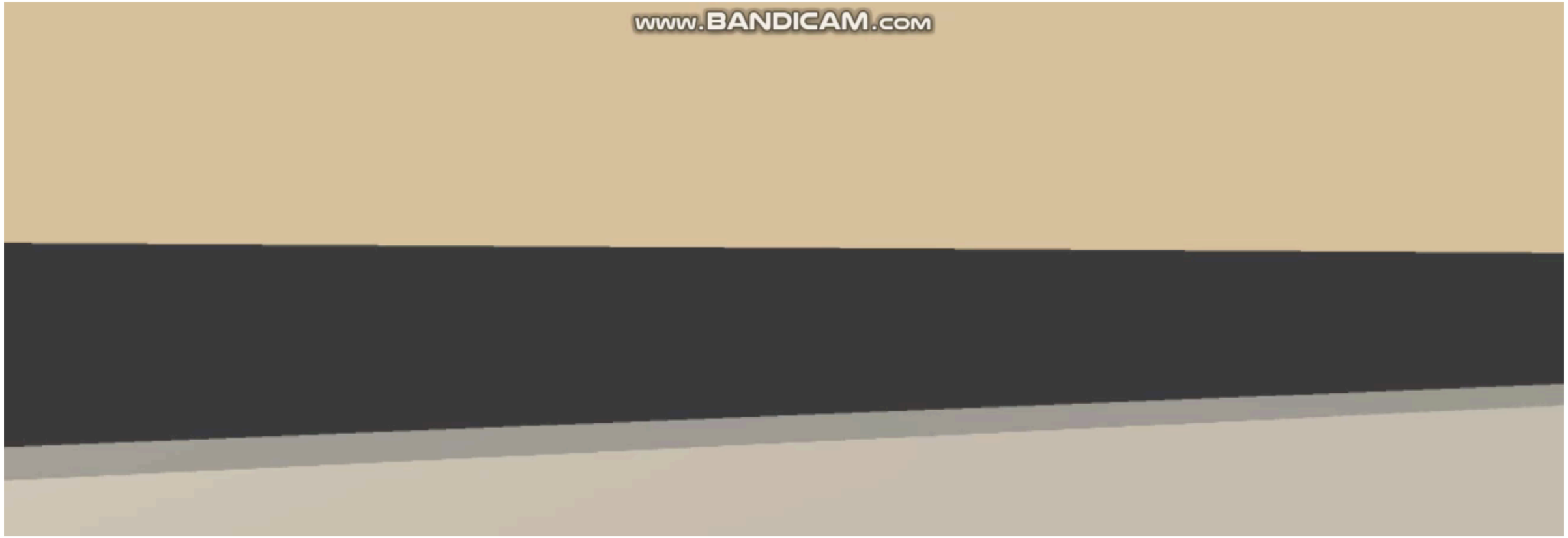
-정보를 공유하며 cooperation(협력)을 하거나 Independent(독립적으로) 하게 목표를 달성한다.

MULTI-AGENTS



- 총 여섯 Agents가 있다.
- 파란색 바나나를 획득했을때 penalty를 받으며 , 노란색 바나나를 획득시 Reward를 받는다.
- 각 Agent는 독립된 Brain을 가지고 있으며 독립적으로 action을 선택한다.

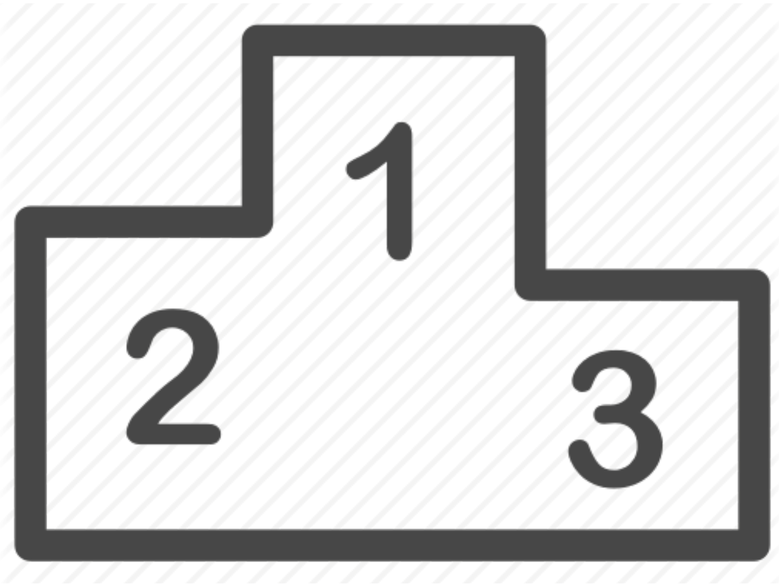
TRAINING USING IMITATION LEARNING



- 여섯 Agents는 Independent하게 노란색Banana를 찾기위해 여러가지 action을 선택하며 배운다.

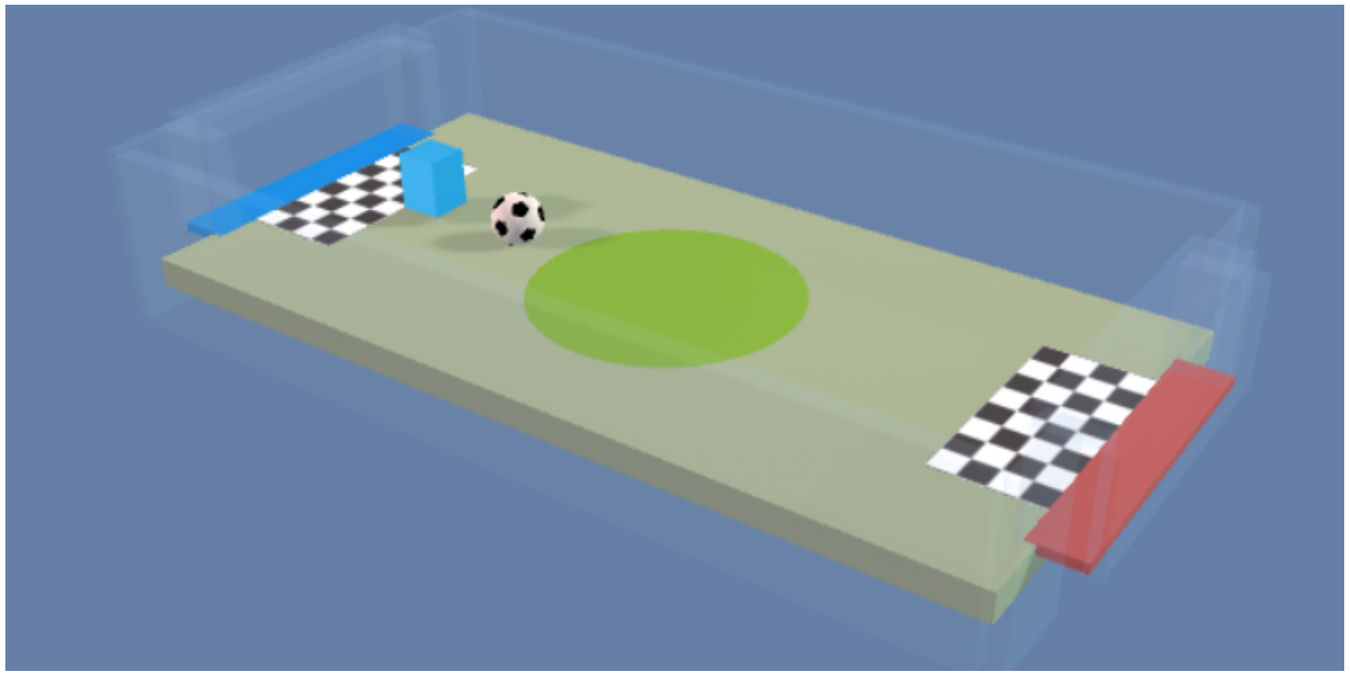
ADVERSARIAL SELF-PLAY

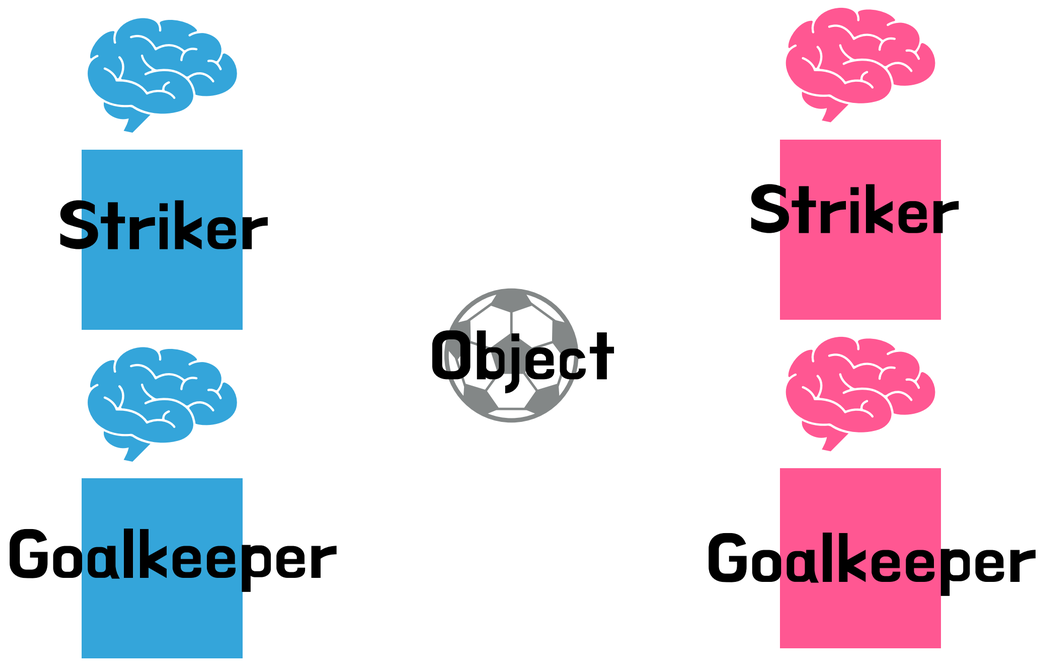
ADVERSARIAL LEARNING?



- 공통된 목표를 달성하기 위해 정보를 공유하며 Cooperation 할때도 있지만, 복싱, 축구, 탁구, 테니스 등과 같이 승패가 확실한 경우도 있다.

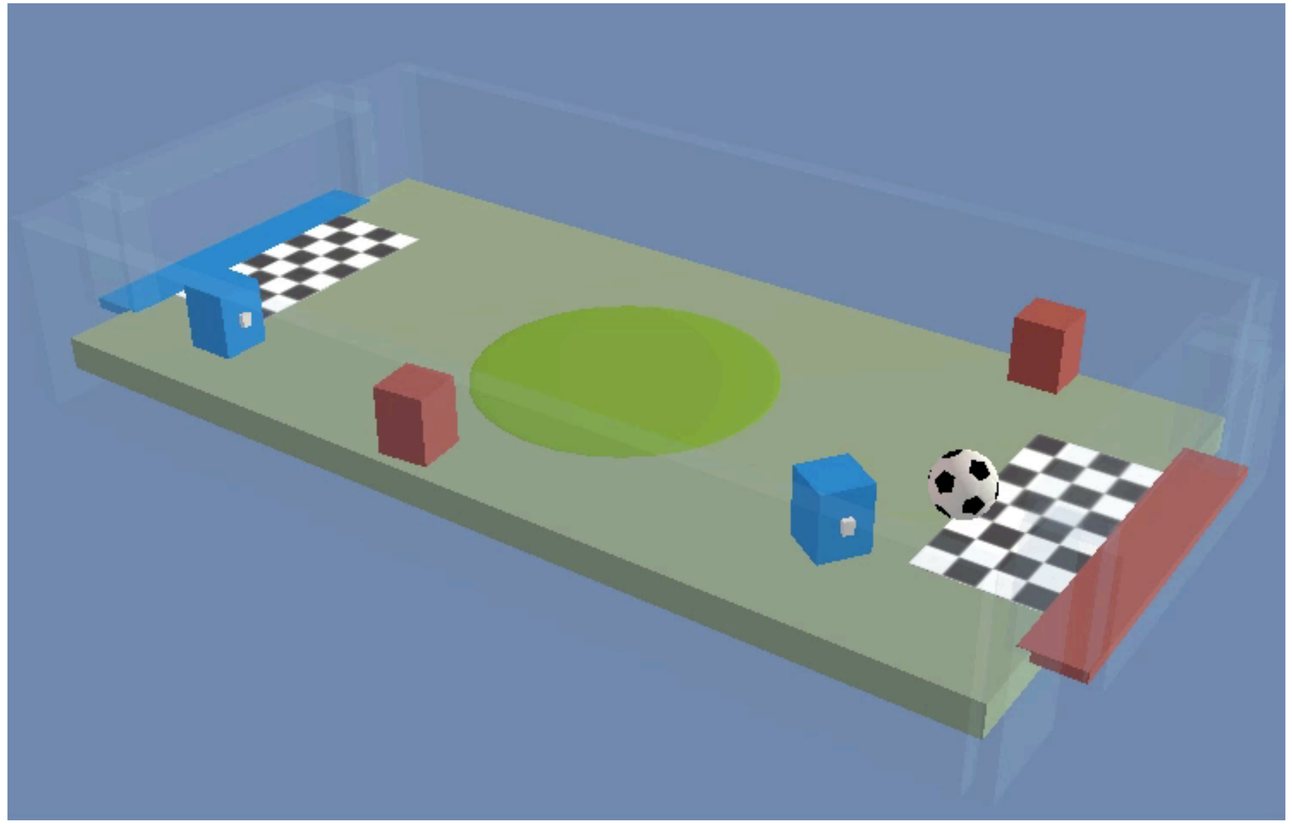
ADVERSARIAL LEARNING

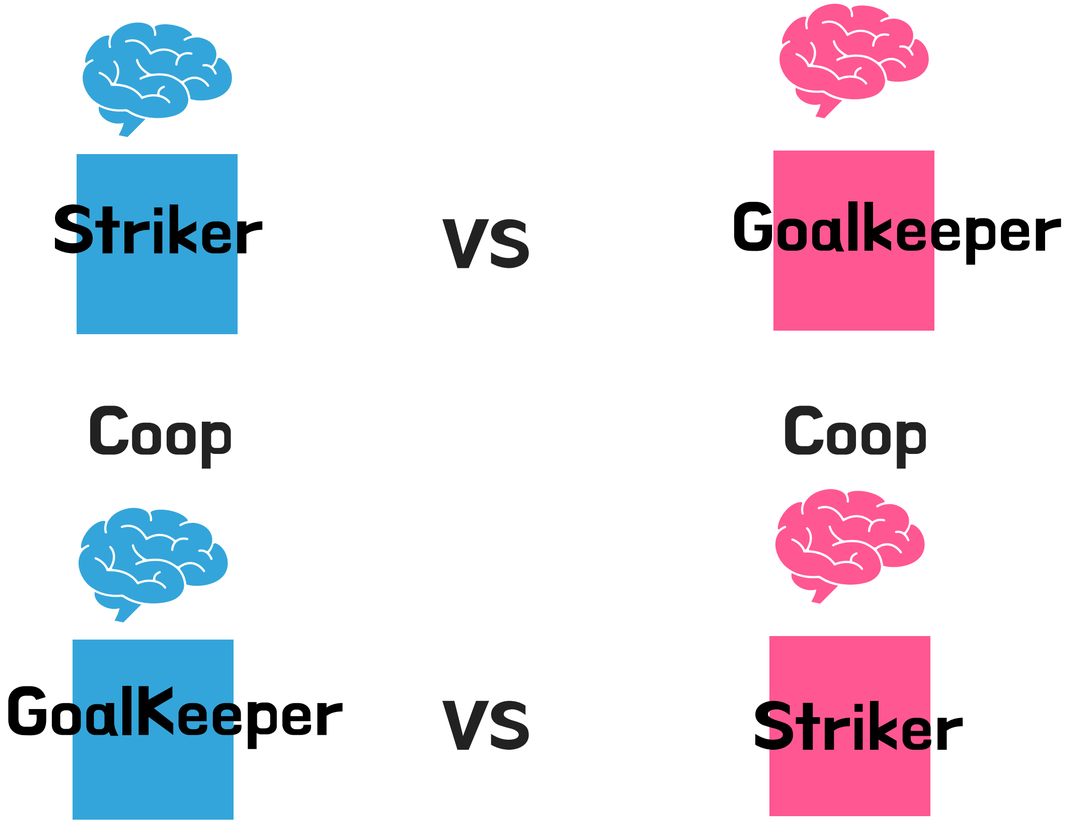




- 공통된 목표를 달성하기 위해 정보를 공유하며 Cooperation 할때도 있지만, 복싱, 축구, 탁구, 테니스 등과 같이 승패가 확실한 경우도 있다.
- 팀당 골을 넣는 striker와 골을 막는 Goalkeeper로 구성되어 있다.

ADVERSARIAL LEARNING





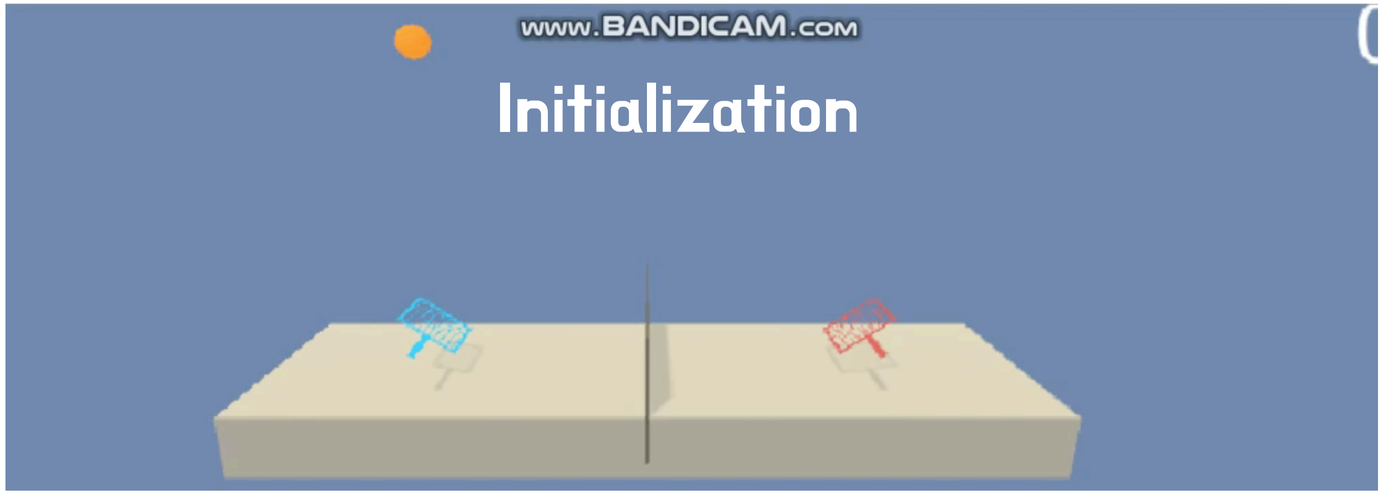
- Striker와 Goalkeeper는 상대편에 공을 넣고 막기 위해 Cooperation하며 행동한다.

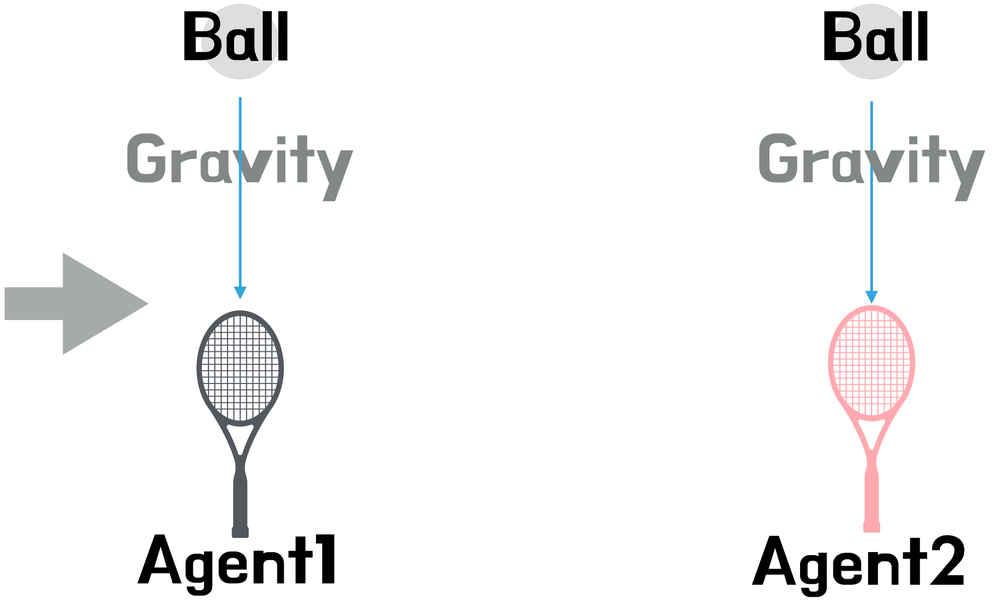
IMITATION LEARNING

IMITATION LEARNING?

- 사람과 동물은 어떠한 목표물을 보고 그들이 하는 behavior을 보고 배운다.
- 타겟이 하는 행동을 모방하며 배우는 방법을 Imitation Learning이라고 한다.

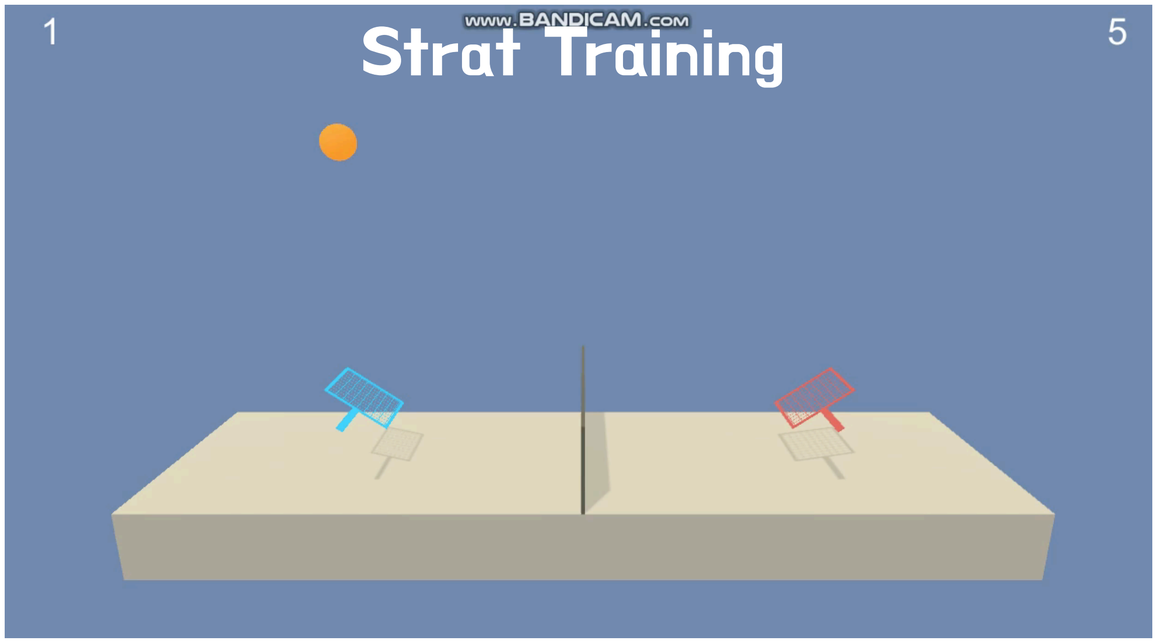
TRAINING USING IMITATION LEARNING

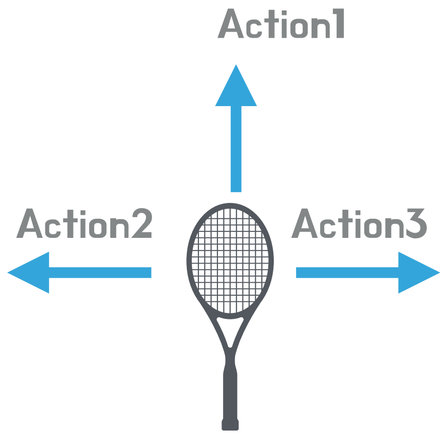




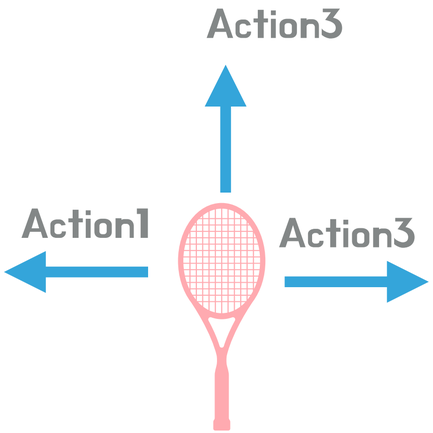
- 공은 중력에 의해 떨어지며, 각 Agent는 공을 받아 반대 Agent 영역으로 넘겨야한다.
- 반대에 위치한 Agent는 공을 보고 다시 넘긴다.

TRAINING USING IMITATION LEARNING





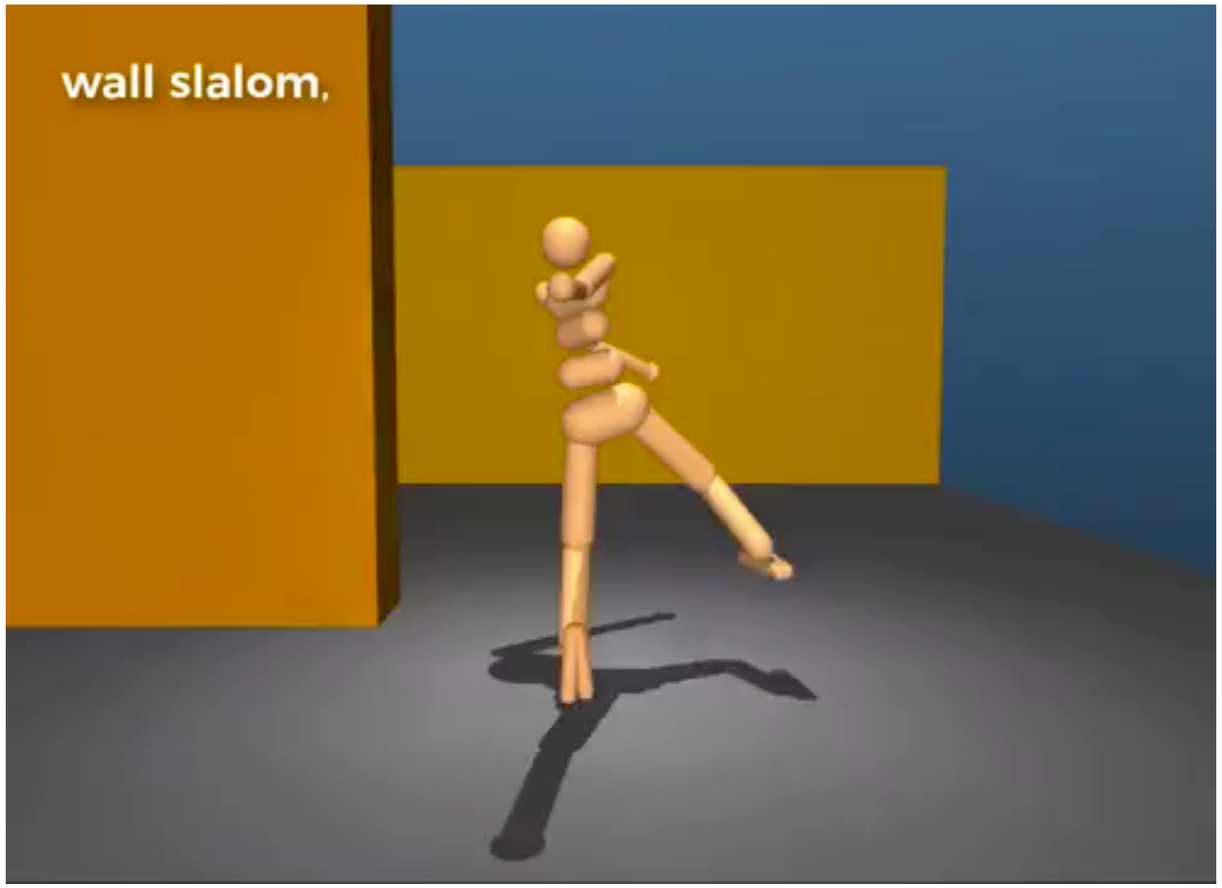
Agent1



Agent2

- Agent는 여러가지 action을 선택하며 더 많은 Reward를 받을수 있는 action을 탐색
- 이런 방식은 학습되는데 시간이 많이 소요된다.

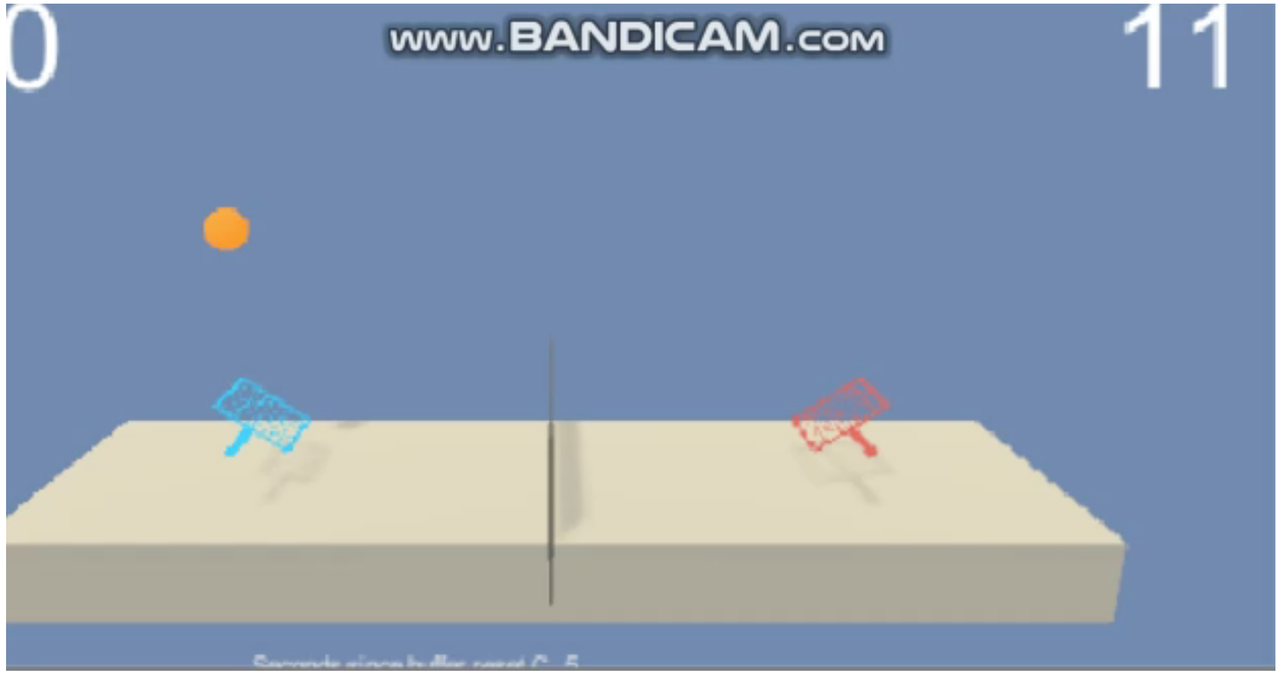
TRAINING WITHOUT IMITATION LEARNING

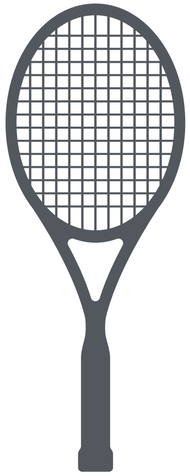


- 로봇이 걷고 뛰게 만드는것과 같은 어려운 환경에서는 학습시간이 매우 길며 자연스럽지 않은 문제가 발생한다.

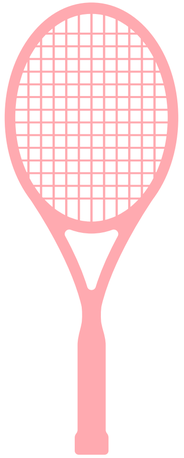
TRAINING USING IMITATION LEARNING

Imitation Learning





Teacher

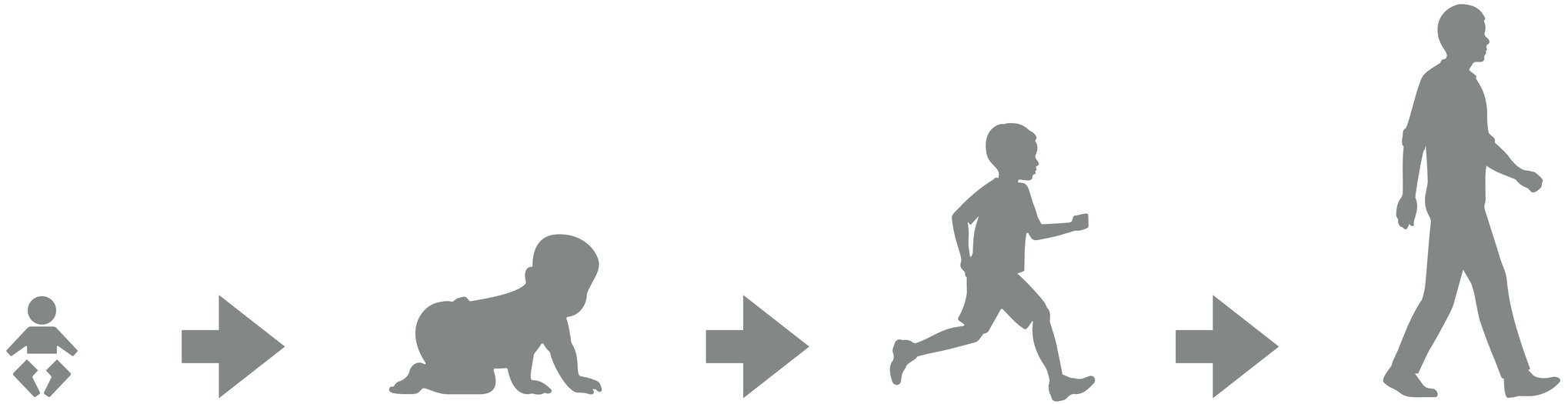


Student

- 더 빠르고 효과적으로 배울수 있게 전문가의 행동을 보고 배우는 Imitation Learning을 사용한다.
- Student는 Teacher을 보고 배움
- Teacher(Player)는 적절한 action을 선택하며 student가 빨리 배울수 있게 도와준다.

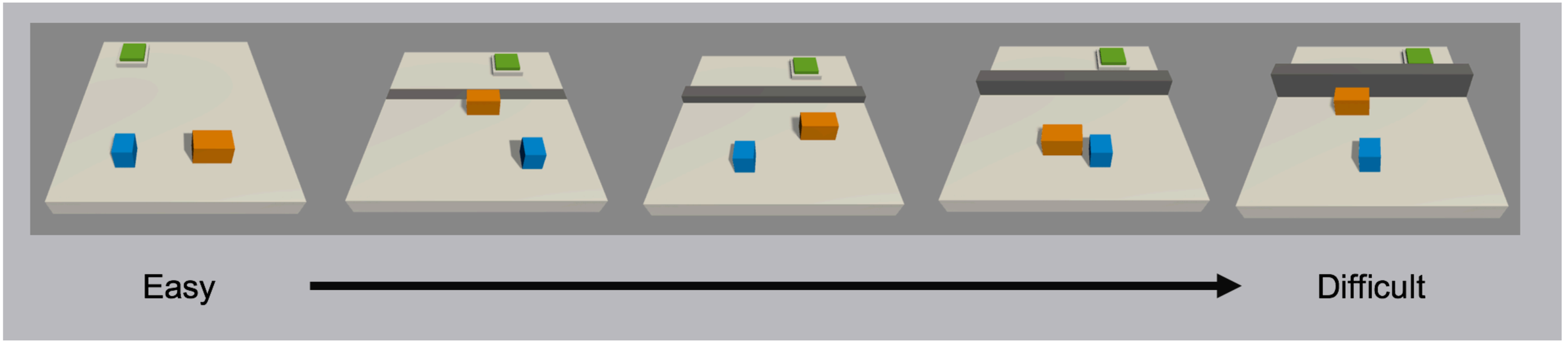
CURRICULUM LEARNING

CURRICULUM LEARNING



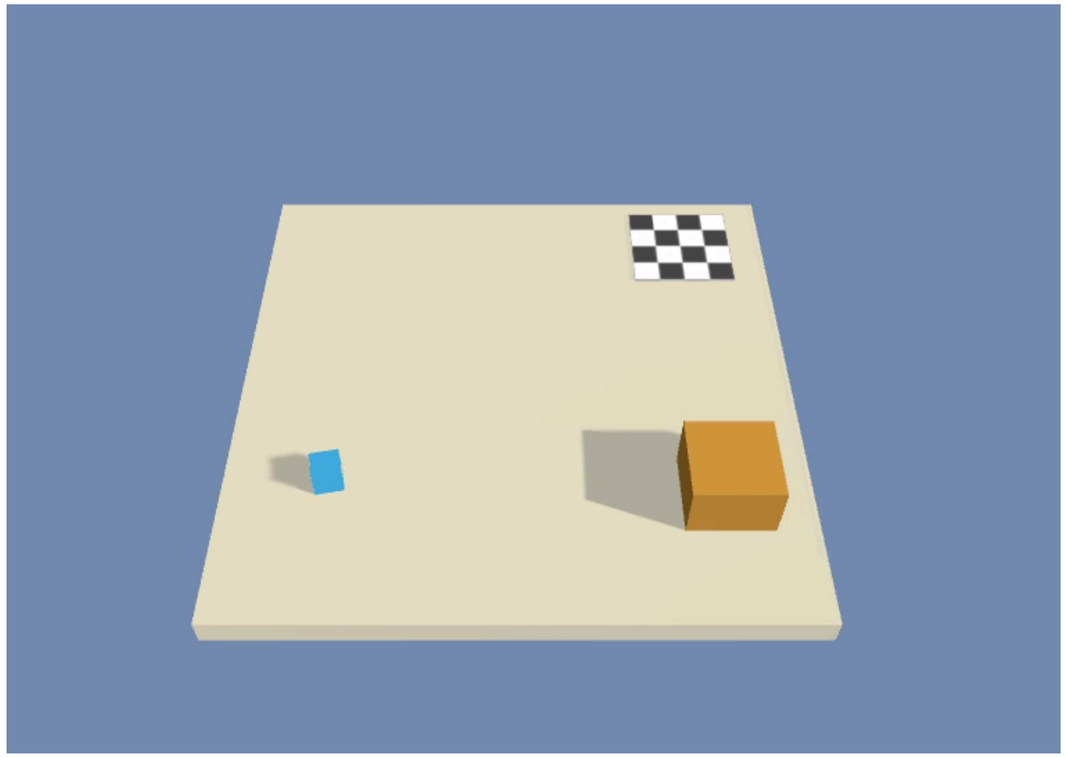
- 사람은 태어나서 바로 달리는 것이 가능하지 않다.
- 뒤집고, 기고, 서고, 걷고, 달리는 것처럼 단계별로 학습한다.
- 이런 학습 방법을 강화학습에서 Curriculum Learning이라고 한다.

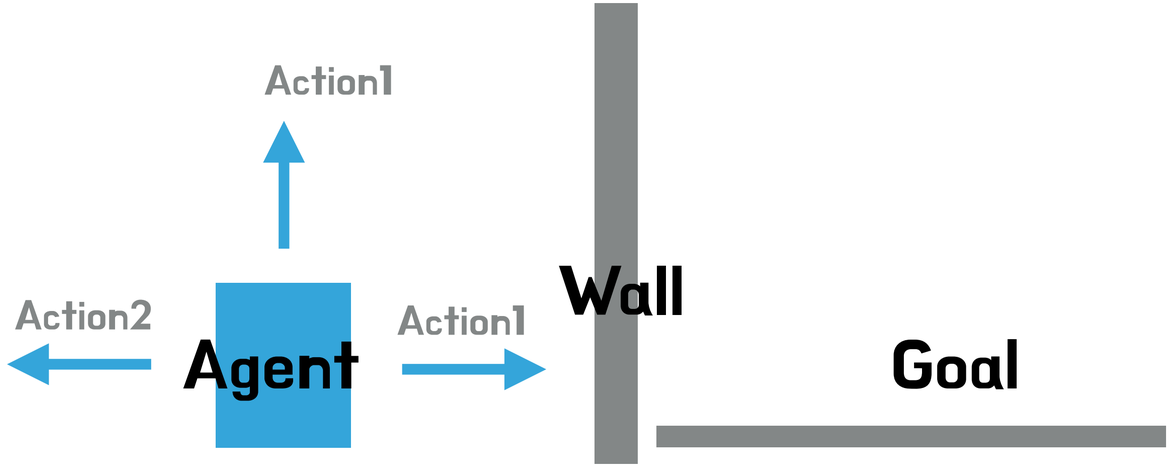
CURRICULUM LEARNING



- Agent는 간단한 task부터 배우기 시작하여 단계별로 학습을 한다.
- 한번에 어려운 task를 학습하기 힘들기 때문에 점점 단계를 높여 학습을 시킨다.

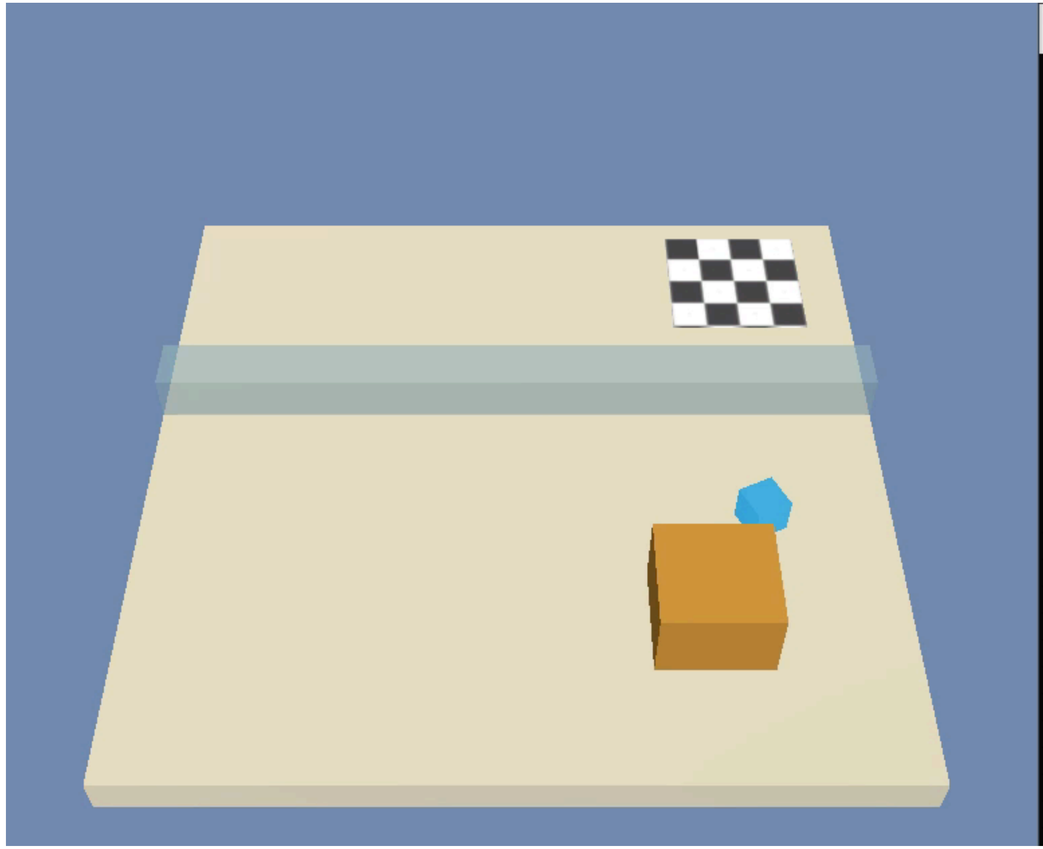
ENVIRONMENTS





- Agent는 Goal에 도착하기 위해 action을 선택한다.
- Wall은 다섯가지 높이로 바뀐다.

TRAINING USING IMITATION LEARNING



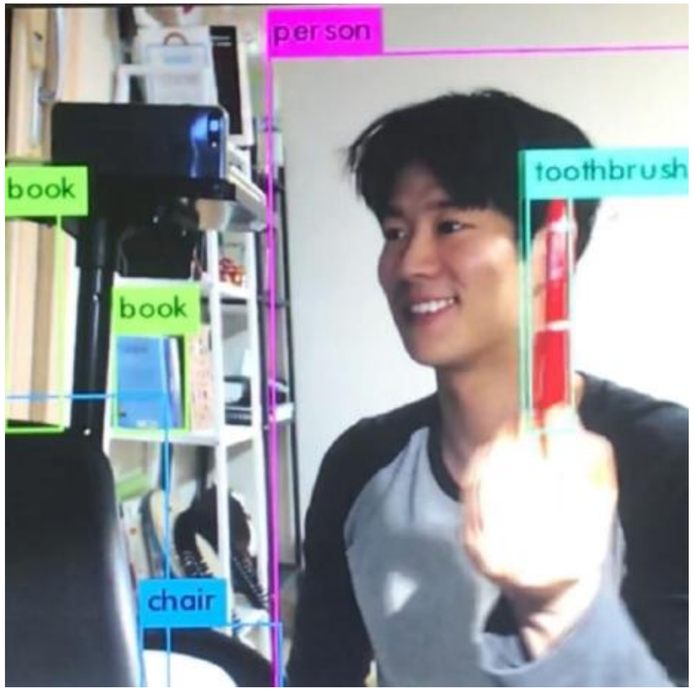
```
Anaconda Prompt - python  learn.py wall_curricu/wall_curricu.exe --run-id=walljump1 --train --curriculum=wall_curricu/curricula.jso
        hidden_units:   256
        lambd:  0.95
        learning_rate:  0.0003
        max_steps:      2.0e5
        normalize:      False
        num_epoch:      3
        num_layers:     2
        time_horizon:   128
        sequence_length:        64
        summary_freq:   2000
        use_recurrent:  False
        graph_scope:    BigWallBrain
        summary_path:   ./summaries/walljump1_BigWallBrain
        memory_size:    256
        use_curiosity:  False
        curiosity_strength:     0.01
        curiosity_enc_size:     128
INFO:unityagents: SmallWallBrain: Step: 2000. Mean Reward: -1.174. Std of Reward: 0.180.
INFO:unityagents: BigWallBrain: Step: 2000. Mean Reward: -1.155. Std of Reward: 0.194.
INFO:unityagents: SmallWallBrain: Step: 4000. Mean Reward: -1.146. Std of Reward: 0.090.
INFO:unityagents: BigWallBrain: Step: 4000. Mean Reward: -1.142. Std of Reward: 0.152.
INFO:unityagents: SmallWallBrain: Step: 6000. Mean Reward: -1.225. Std of Reward: 0.271.
INFO:unityagents: BigWallBrain: Step: 6000. Mean Reward: -1.141. Std of Reward: 0.211.
INFO:unityagents: SmallWallBrain: Step: 8000. Mean Reward: -1.329. Std of Reward: 0.249.
INFO:unityagents: BigWallBrain: Step: 8000. Mean Reward: -1.079. Std of Reward: 0.464.
INFO:unityagents: SmallWallBrain: Step: 10000. Mean Reward: -1.080. Std of Reward: 0.115.
INFO:unityagents: BigWallBrain: Step: 10000. Mean Reward: -0.933. Std of Reward: 0.357.
INFO:unityagents: SmallWallBrain: Step: 12000. Mean Reward: -1.483. Std of Reward: 0.480.
INFO:unityagents: BigWallBrain: Step: 12000. Mean Reward: -0.875. Std of Reward: 0.611.
```

- 파란색 agent는 small wall과 large wall을 피하거나 뛰어넘어 목표로 가기 위해 학습한다.

SUMMARY

1. Create **Environments**
2. **Multi-Agent Environment**
3. **Adversarial self-play**
4. **Imitation** Learning
5. **Curriculum** Learning

SUMMARY



감사합니다.

Github:
https://github.com/wonseokjung

Facebook:
https://www.facebook.com/ws.jung.798

Blog:
https://wonseokjung.github.io/