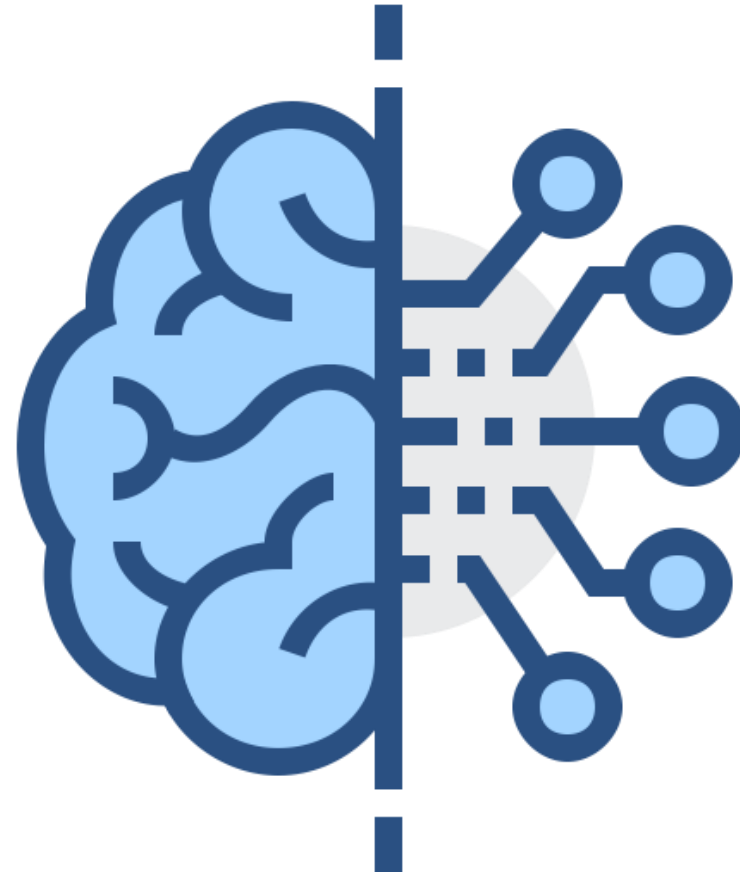


ML-Agents를 이용한 강화학습 에이전트 개발 사례 소개

2020. 06. 10. 민규식

kyushikmin@gmail.com



목차



- 강화학습 개요
- ML-Agents 소개
- ML-Agents를 이용한 연구

발표자 소개

발표자 소개

민규식 (한양대학교 석박사 통합과정)

연구 주제

- 자율주행차, 운전자 보조 시스템, 인공지능, 강화학습

경력사항

- Unity Masters 3, 4기
- 강화학습 관련 페이스북 페이지 RLKorea 운영진
- ML-Agents Challenge에서 Creative Application Award 수상
- ML-Agents를 이용한 연구 수행 및 논문 발표
- ML-Agents에 대한 다수의 세미나 강의 수행

RLKorea Unity ML-Agents Team



민규식

정규석

신명재

이현호

윤성훈

조동헌

텐서플로와
유니티
ML-Agents로 배우는
강화학습

다섯 가지
게임들
제작하며
살펴보는
심층강화학습

민규식, 신명재, 윤성훈,
이현호, 정규석, 조동헌 지음

DS 데이터 사이언스 시리즈 046

우리복스

Reinforcement Learning

Machine Learning



Supervised Learning



Unsupervised Learning



Reinforcement Learning

Reinforcement Learning

강화학습



Agent

Action (a)
Jump, forward, backward, run, ...

State (s)
Position of agent, enemy, coins

Reward (r)



Good



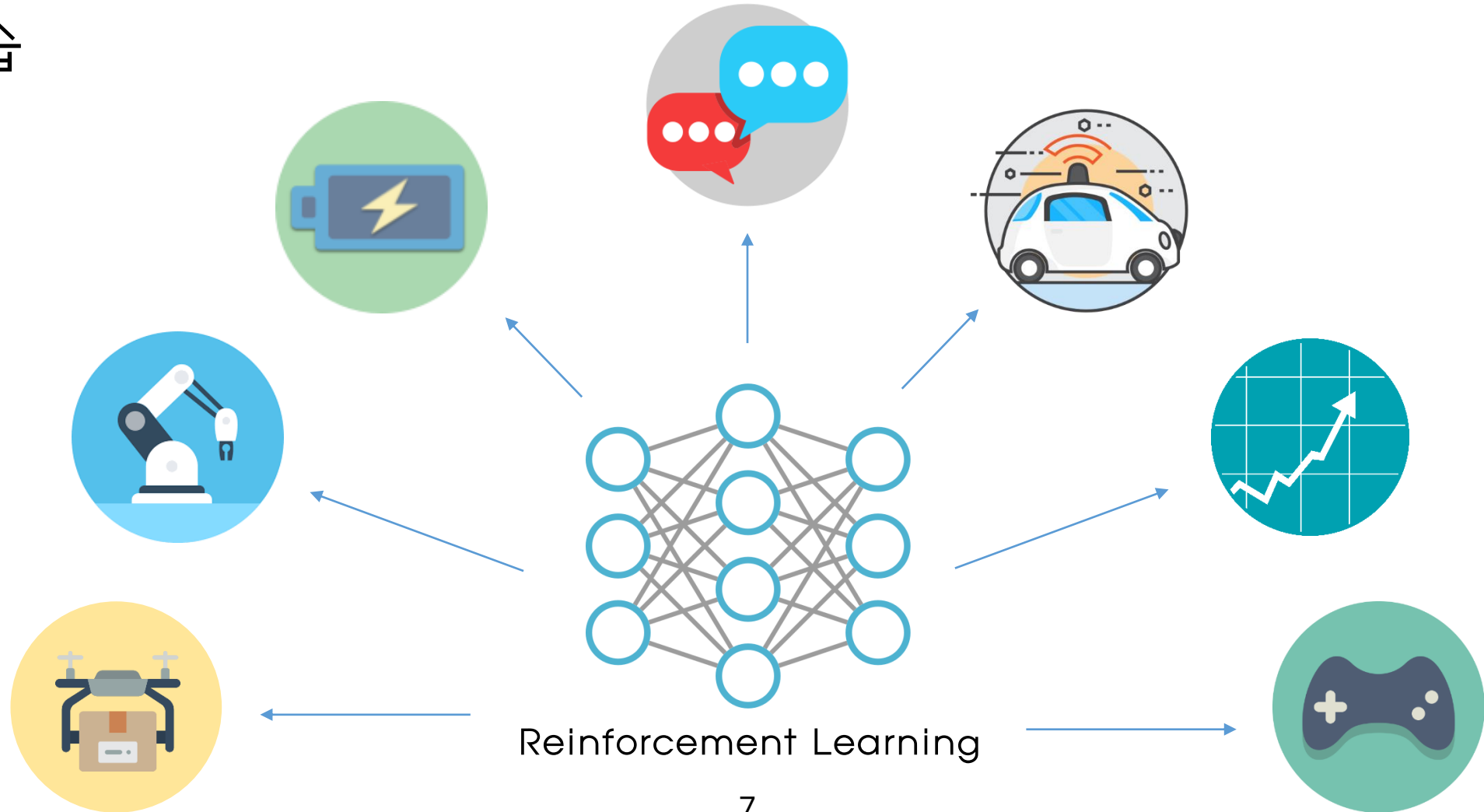
Bad



Environment

Reinforcement Learning

강화학습



Reinforcement Learning

강화학습



Deep Q Network

Rainbow DQN

Deep Deterministic Policy Gradient

Trust Region Policy Optimization

Proximal Policy Optimization

Soft Actor Critic



OpenAI GYM

Atari

GTA 5

Super Mario

Mujoco

Malmo

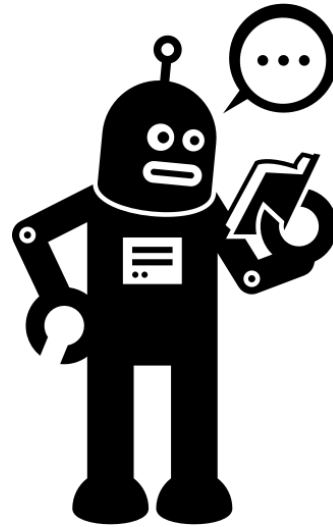


Reinforcement Learning

만들어진 환경을 사용하는 경우



환경에 대한 수정이 어려움!



환경마다 사용법이 다름



필요한 환경이 없을 수도 있음!

Reinforcement Learning

강화학습



Environment ???

Reinforcement Learning

강화학습



강화학습을 하는 사람들의 고민



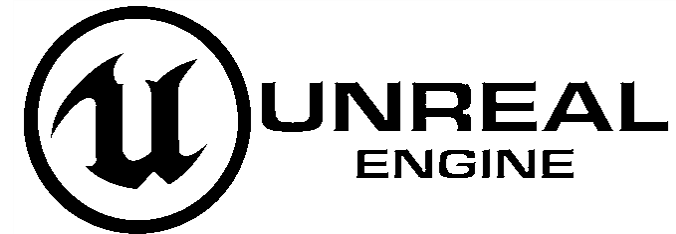
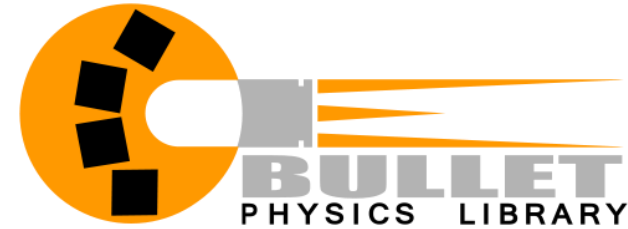
강화학습을 테스트할 환경 제작

Unity ML-Agents

강화학습



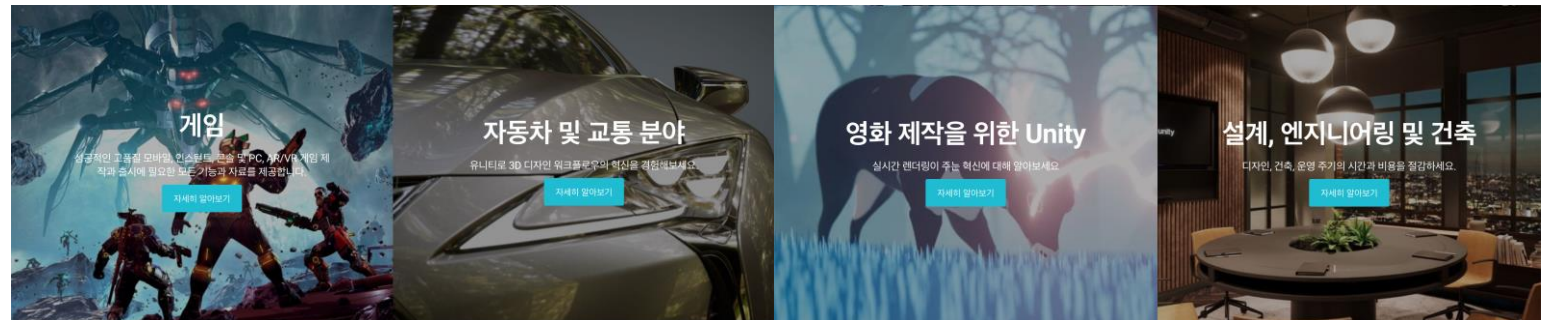
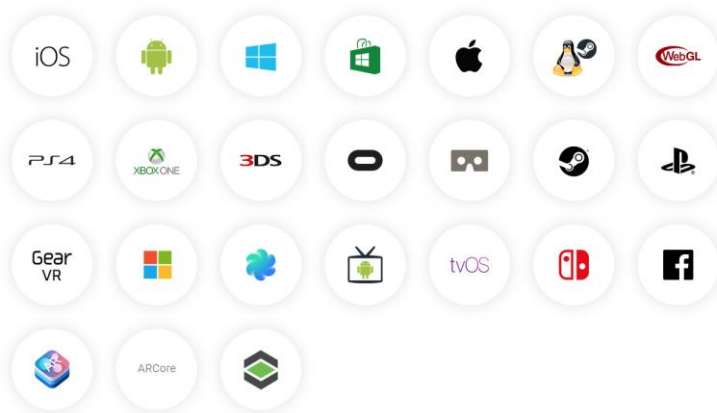
강화학습을 테스트할 환경 제작



Unity ML-Agents



- 3D 및 2D 비디오 게임의 개발 환경을 제공하는 게임 엔진
- 3D 애니메이션과 건축 시각화, 가상현실(VR) 등 인터랙티브 콘텐츠 제작
- 게임 엔진 시장의 45% 이상 차지, 등록 개발자 수 500만 명 이상
- 물리법칙 구현, 다양한 3D 모델 및 기능 구매 가능, 비교적 간단하게 환경 제작이 가능



Unity ML-Agents

Unity 



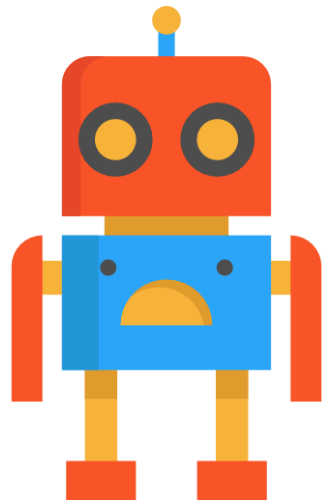
Unity ML-Agents



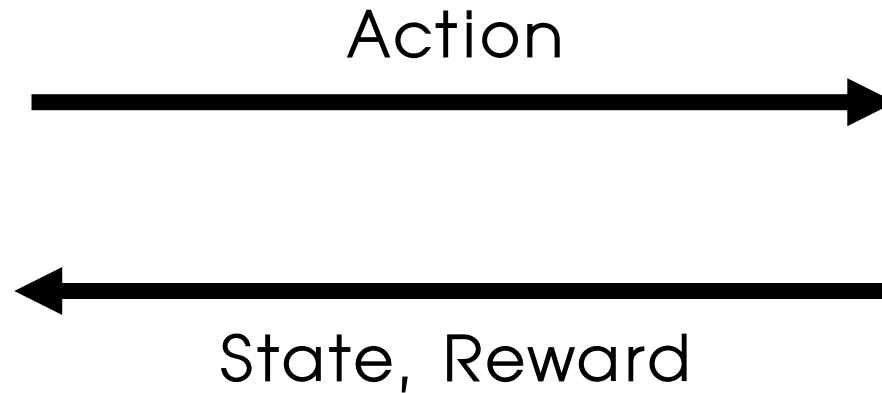
2017.09.19 v0.1 -> 2020.04.30 v1.0 -> 현재 1.0 Release 2

Unity ML-Agents

Unity ML-Agents



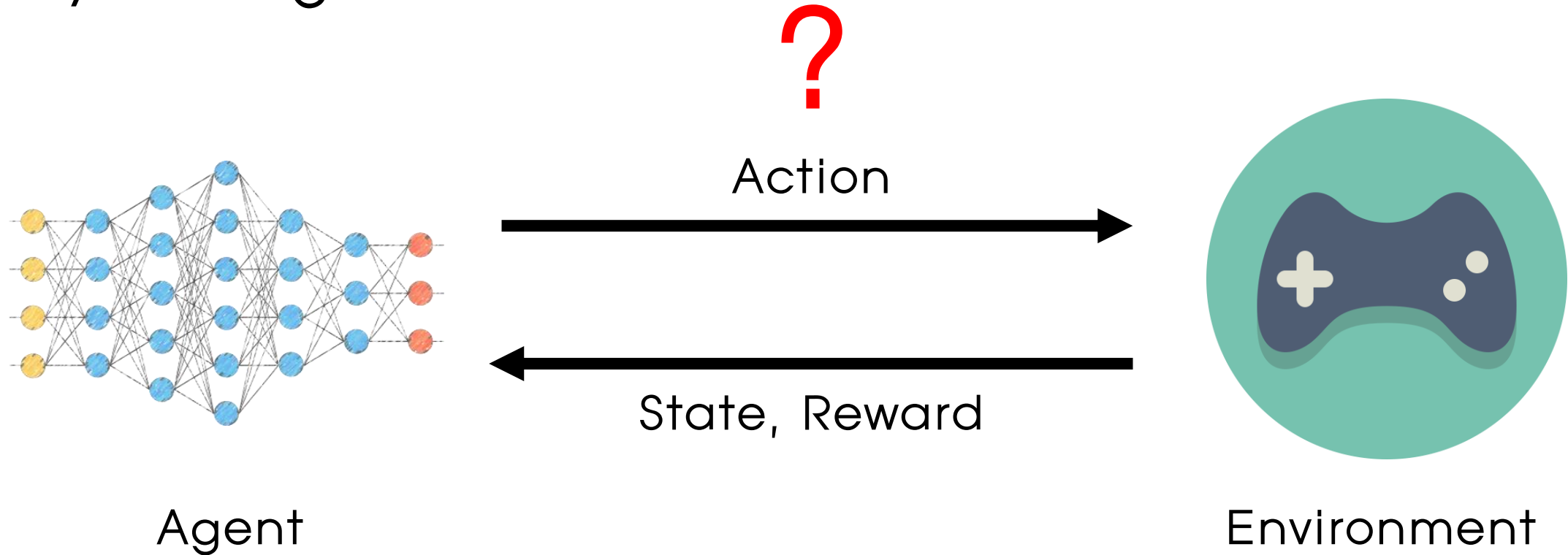
Agent



Environment

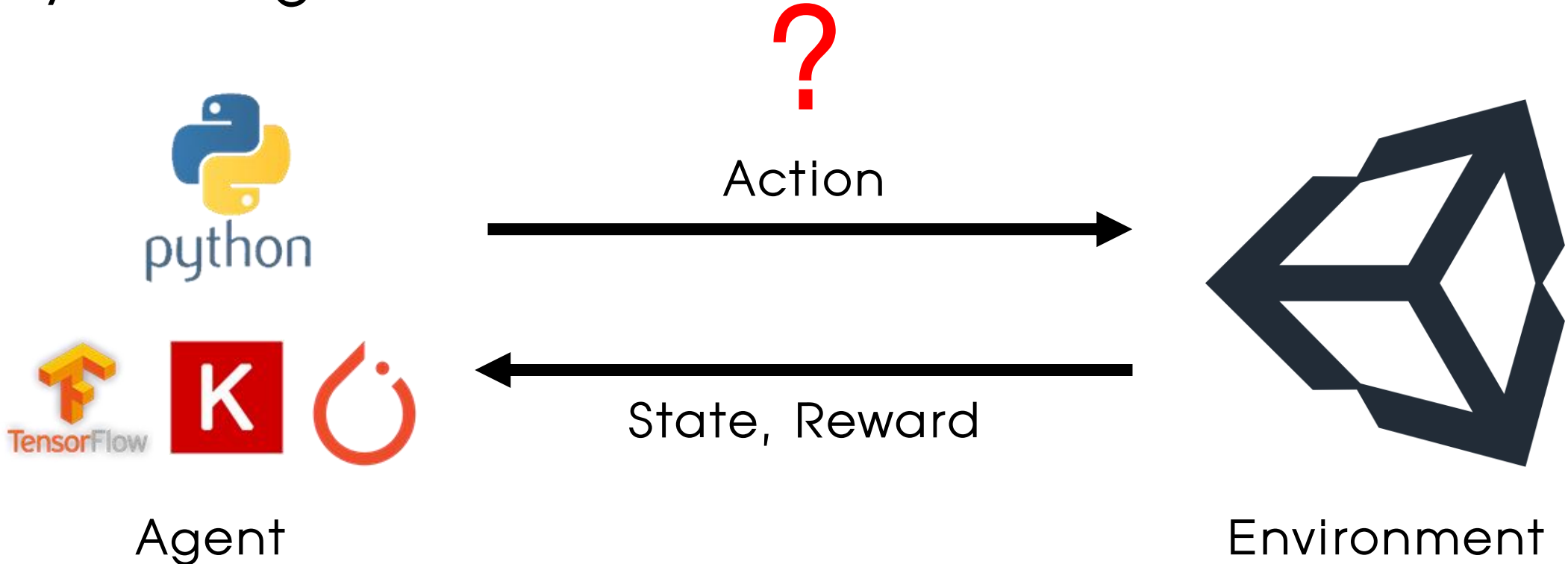
Unity ML-Agents

Unity ML-Agents



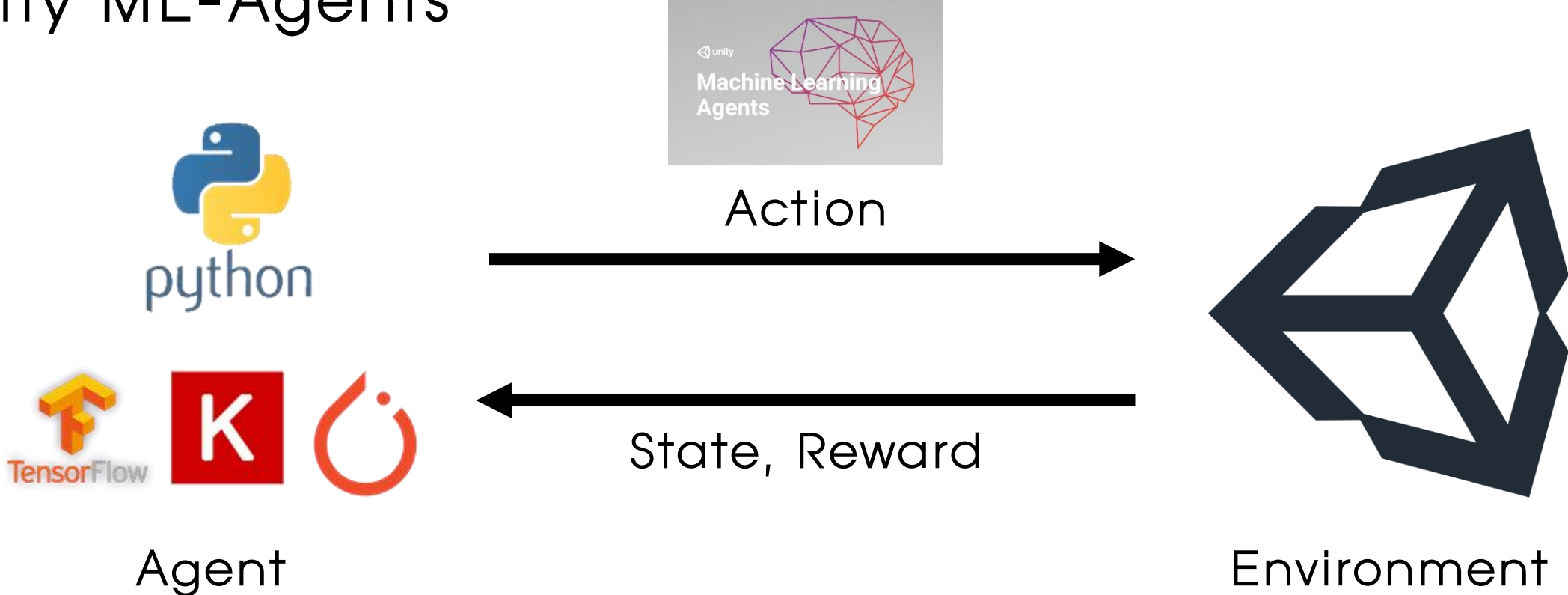
Unity ML-Agents

Unity ML-Agents



Unity ML-Agents

Unity ML-Agents



Unity ML-Agents

Unity ML-Agents v1.0



Unity ML-Agents

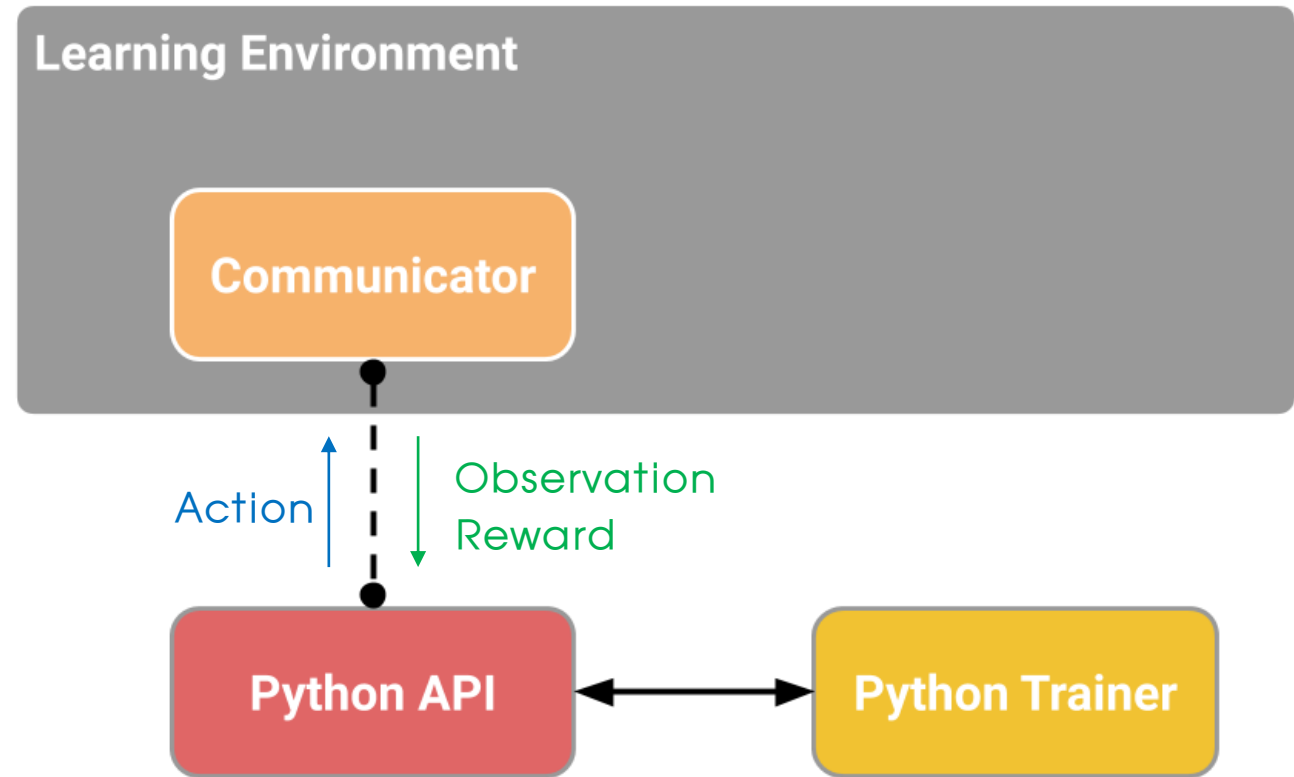
Unity ML-Agents v1.0

- ML-Agents의 구성 요소
 - Learning Environment: 학습을 위한 환경
 - Python Low-Level API: 유니티와 독립적인 python low-level interface로 communicator를 통해 유니티 환경과 통신
 - External Communicator: Learning environment에 포함되며 learning environment와 python low-level API를 연결
 - Python Trainers: 에이전트를 학습시킬 수 있는 머신러닝 알고리즘들을 포함, python을 통해 실행
 - Gym Wrapper: Learning environment를 OpenAI Gym 형태로 만들어줌

Unity ML-Agents

Unity ML-Agents v1.0

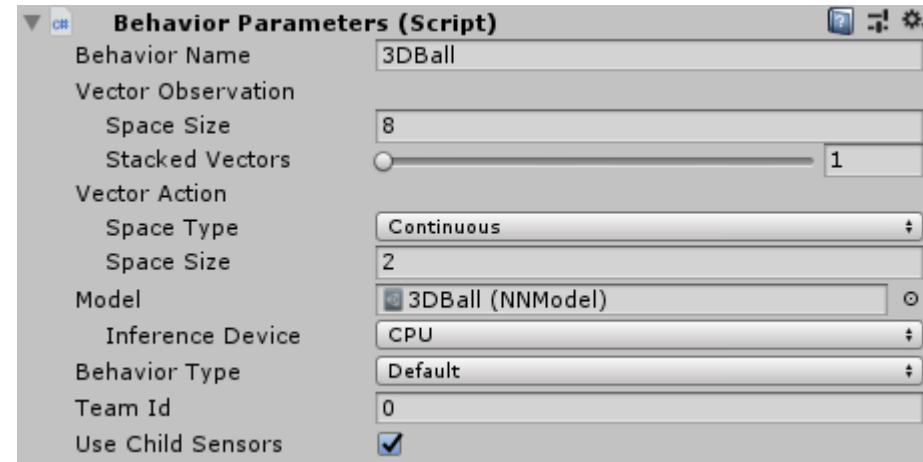
- ML-Agents의 구성 요소
 - Learning Environment
 - Python Low-Level API
 - External Communicator
 - Python Trainers
 - Gym Wrapper



Unity ML-Agents

Unity ML-Agents v1.0

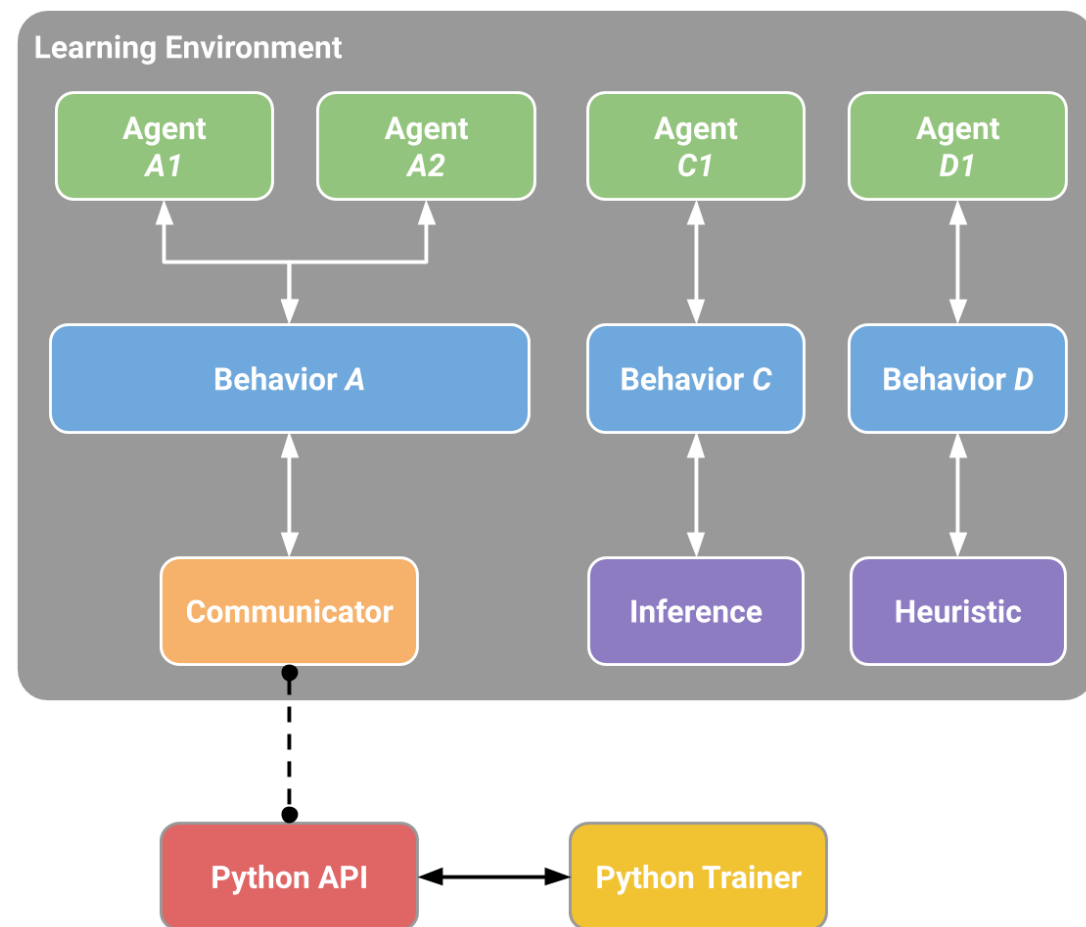
- Learning Environment의 구성 요소
 - Agents
 - Unity GameObject (게임 내의 캐릭터)에 붙이는 스크립트
 - Observation에 대한 설정, 각 action에 대한 행동 결정, 보상 및 게임 종료 조건 결정
 - Behavior
 - 에이전트에 대한 구체적인 설정을 수행
 - Vector observation의 수, Action의 종류, Action의 수, ...
 - 각 환경에는 하나 이상의 에이전트가 존재
 - 각 에이전트는 behavior script를 포함



Unity ML-Agents

Unity ML-Agents v1.0

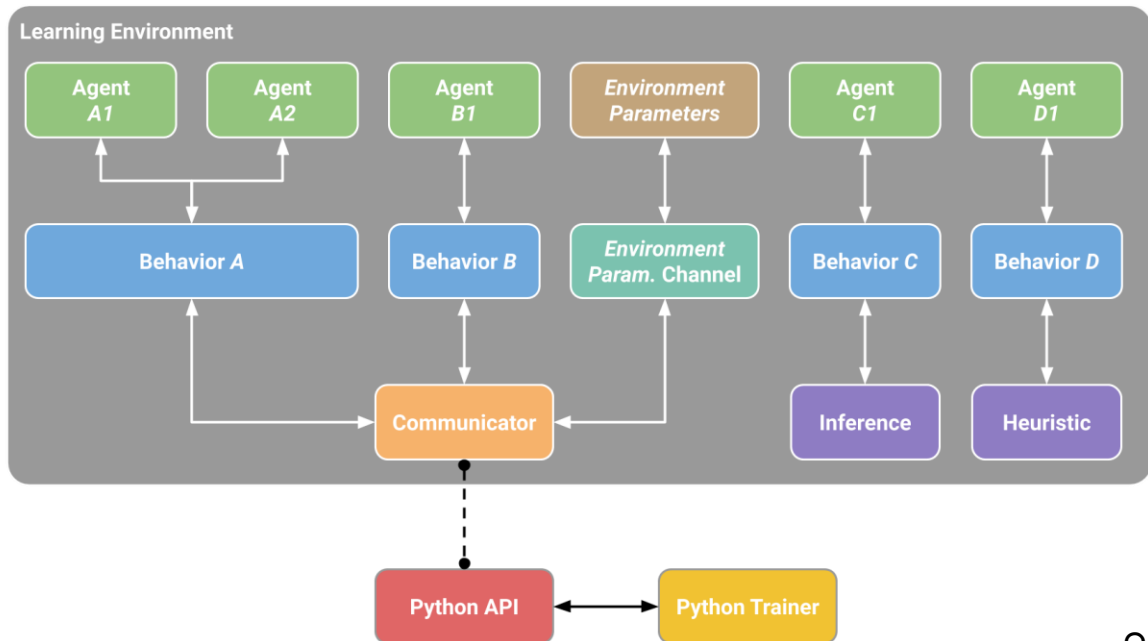
- Learning Environment의 구성 요소
 - Learning Environment 내에 다수의 Agents가 존재하는 경우
 - 각 에이전트의 행동을 다양한 방식으로 선택 가능
 - Agent A1, A2
 - 동일한 behavior A를 사용, Python API를 통해 학습
 - Agent C1
 - Inference: 학습된 모델을 이용하여 행동을 결정
 - Agent D1
 - Heuristic: 정해진 규칙에 따라 행동 (사람이 컨트롤 or rule-based)



Unity ML-Agents

Unity ML-Agents v1.0

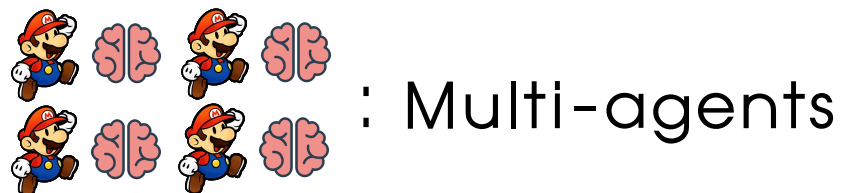
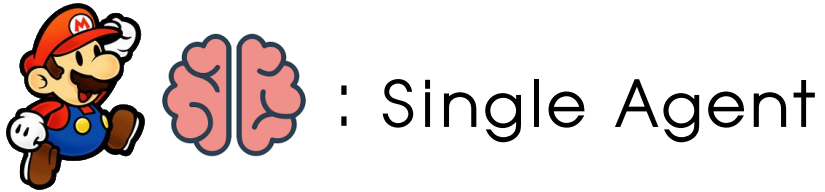
- Learning Environment의 구성 요소
 - Side Channel을 통해 Unity와 Python 사이에 학습에 직접적으로 연관이 없는 데이터도 교환하는 것이 가능 (ex. 환경의 정보, Plotting을 위한 정보)



Unity ML-Agents

Unity ML-Agents v1.0

- ML-Agents를 통해 구현 가능한 Learning Environment의 종류



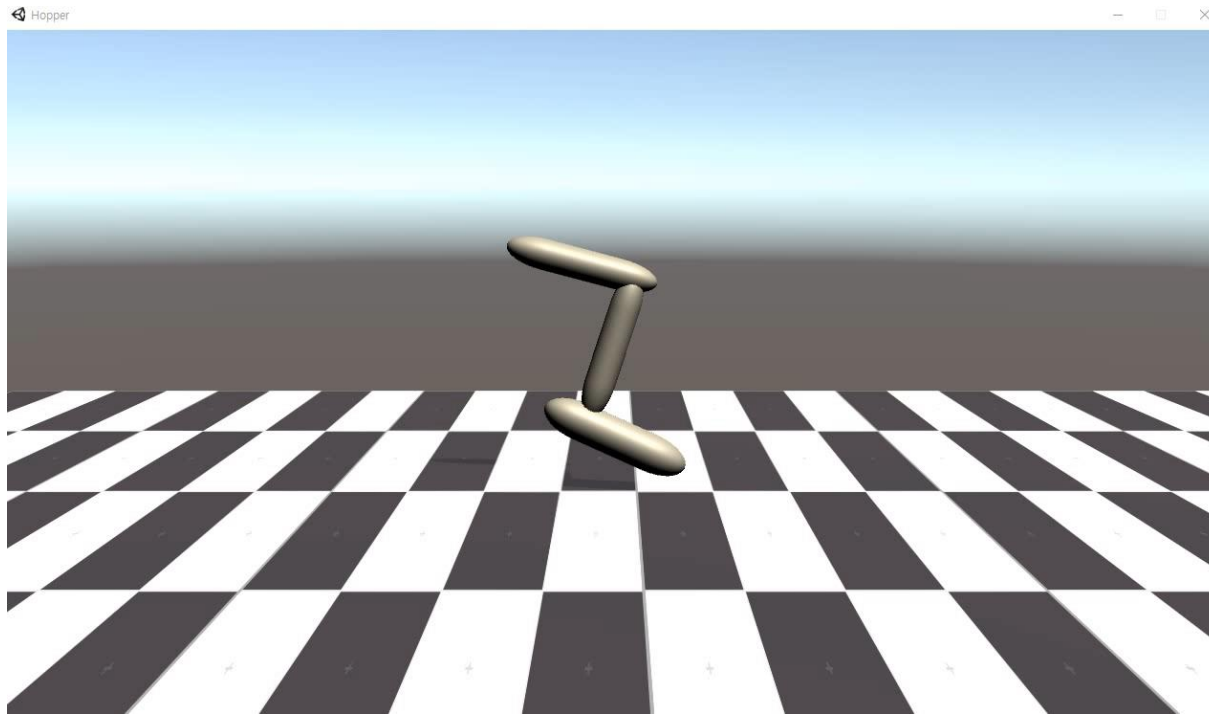
Unity ML-Agents

Unity ML-Agents v1.0

- ML-Agents를 통해 구현 가능한 Learning Environment의 종류



: Single Agent

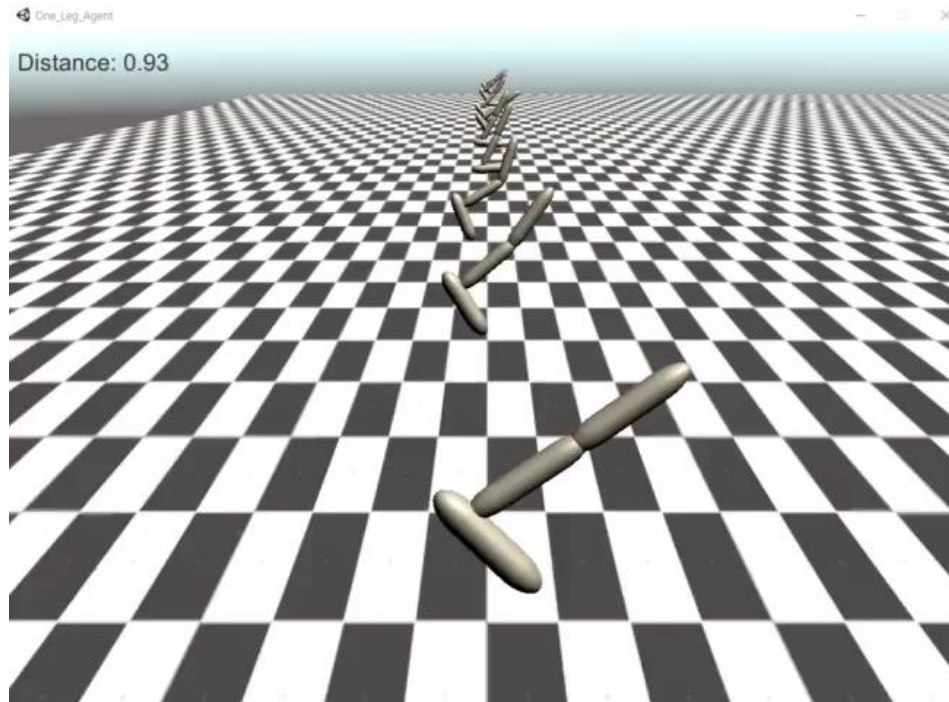


Unity ML-Agents

Unity ML-Agents v1.0

- ML-Agents를 통해 구현 가능한 Learning Environment의 종류

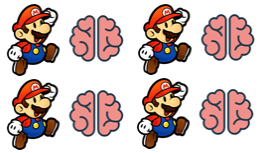
 : Distributed Agents



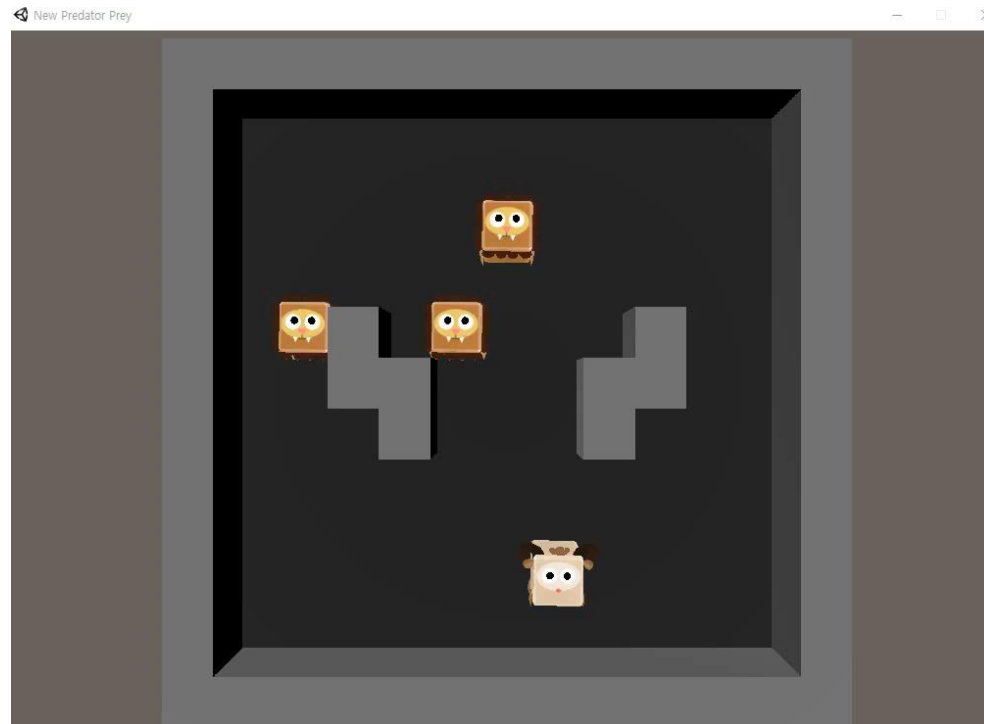
Unity ML-Agents

Unity ML-Agents v1.0

- ML-Agents를 통해 구현 가능한 Learning Environment의 종류



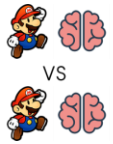
: Multi-agents



Unity ML-Agents

Unity ML-Agents v1.0

- ML-Agents를 통해 구현 가능한 Learning Environment의 종류



vs

: Adversarial Agents

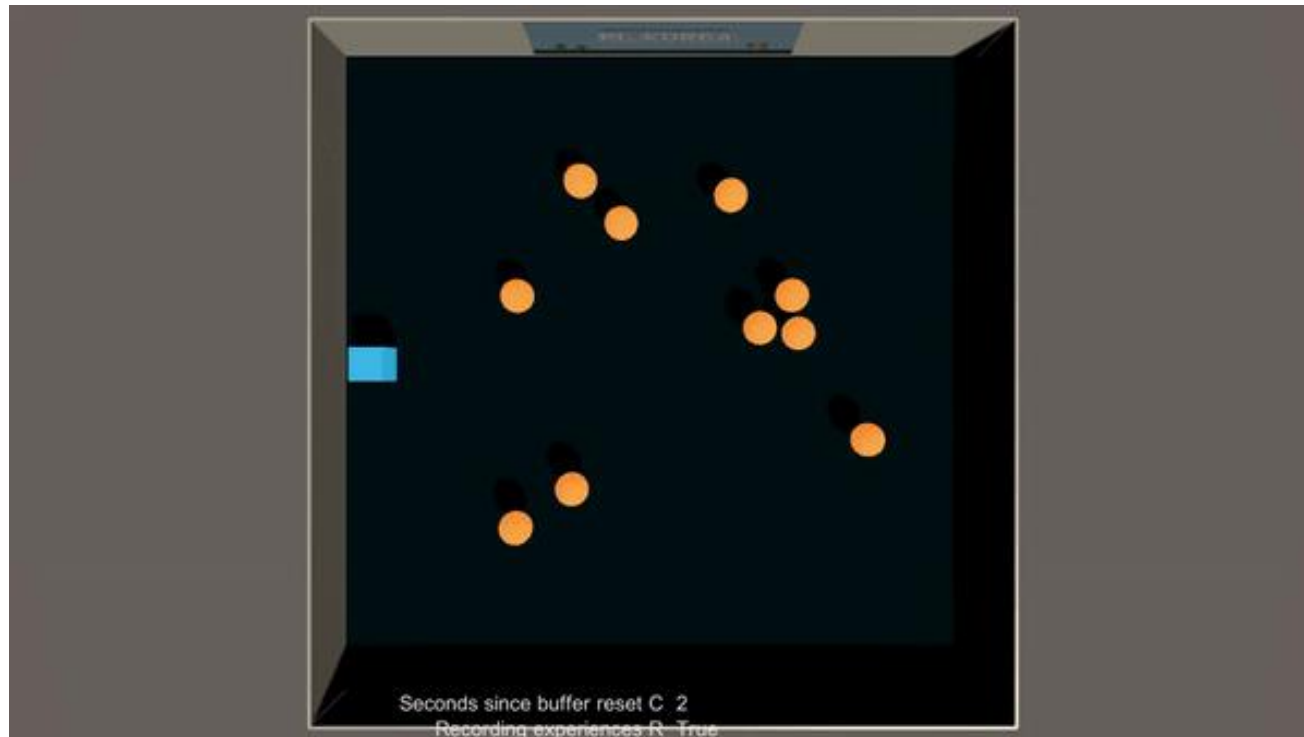


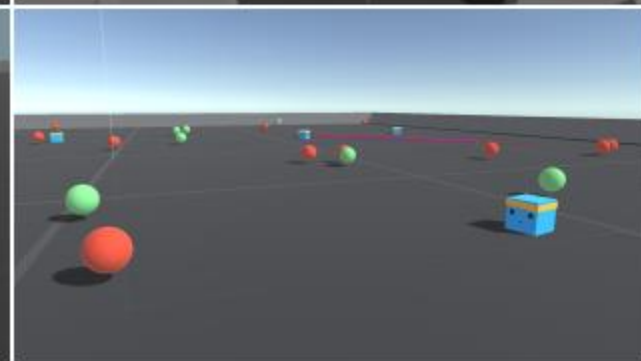
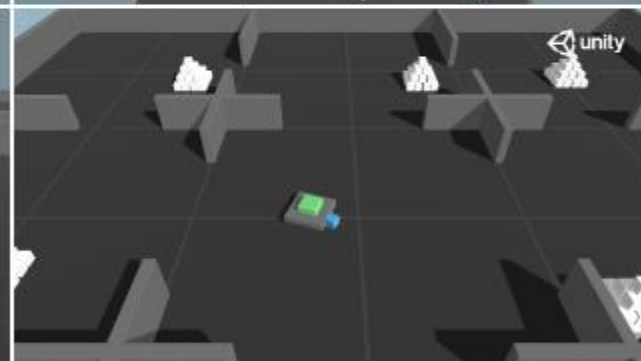
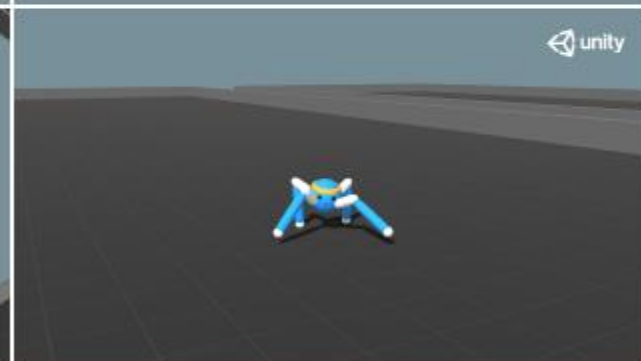
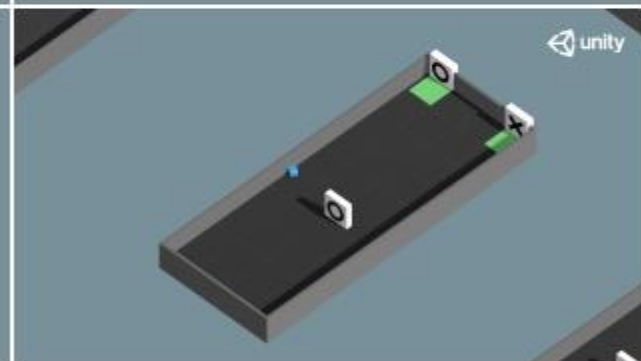
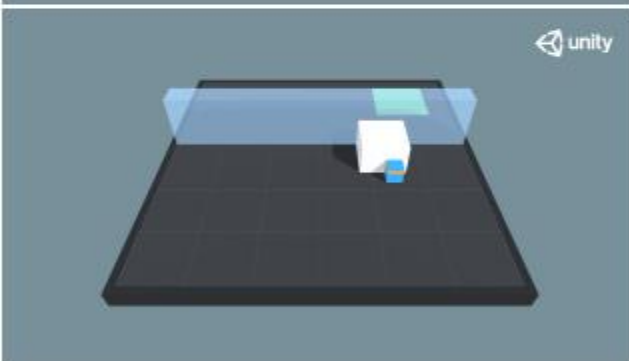
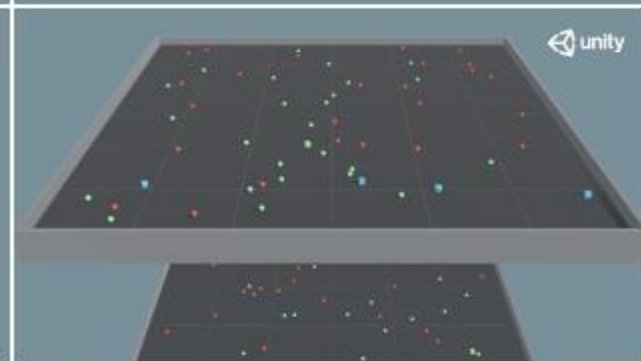
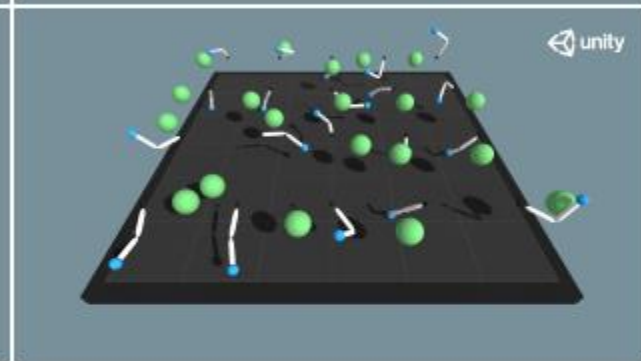
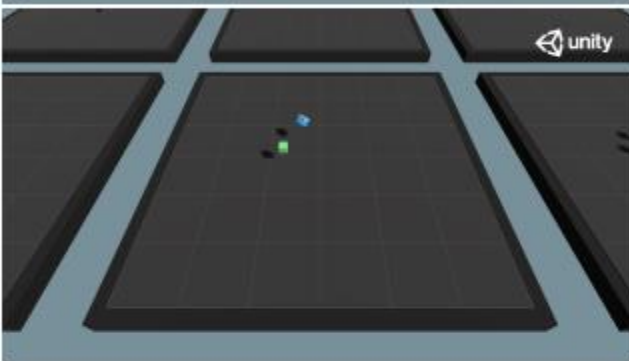
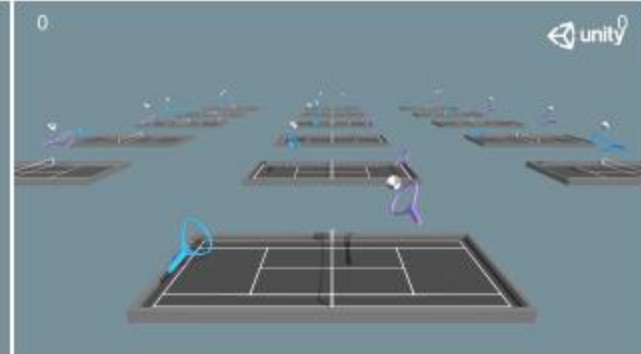
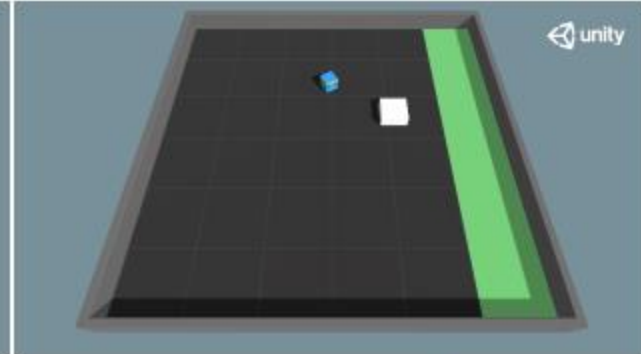
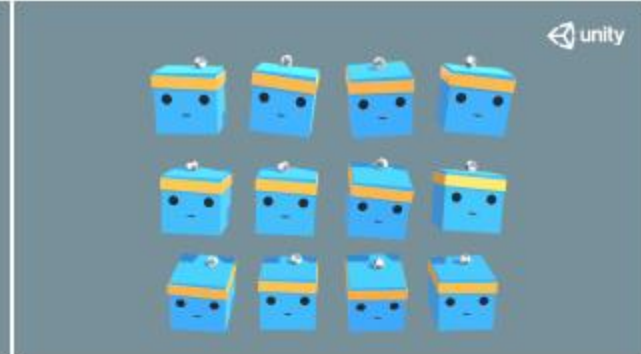
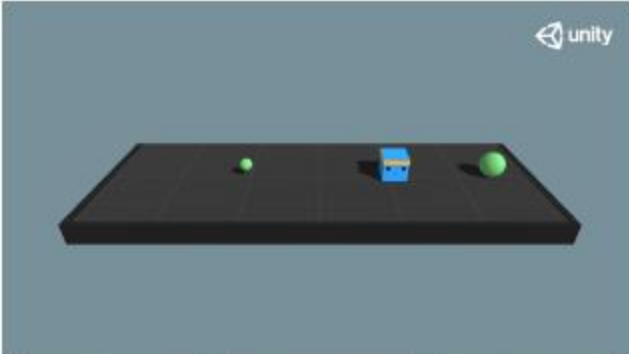
Unity ML-Agents

Unity ML-Agents v1.0

- ML-Agents를 통해 구현 가능한 Learning Environment의 종류

  : Imitation Learning

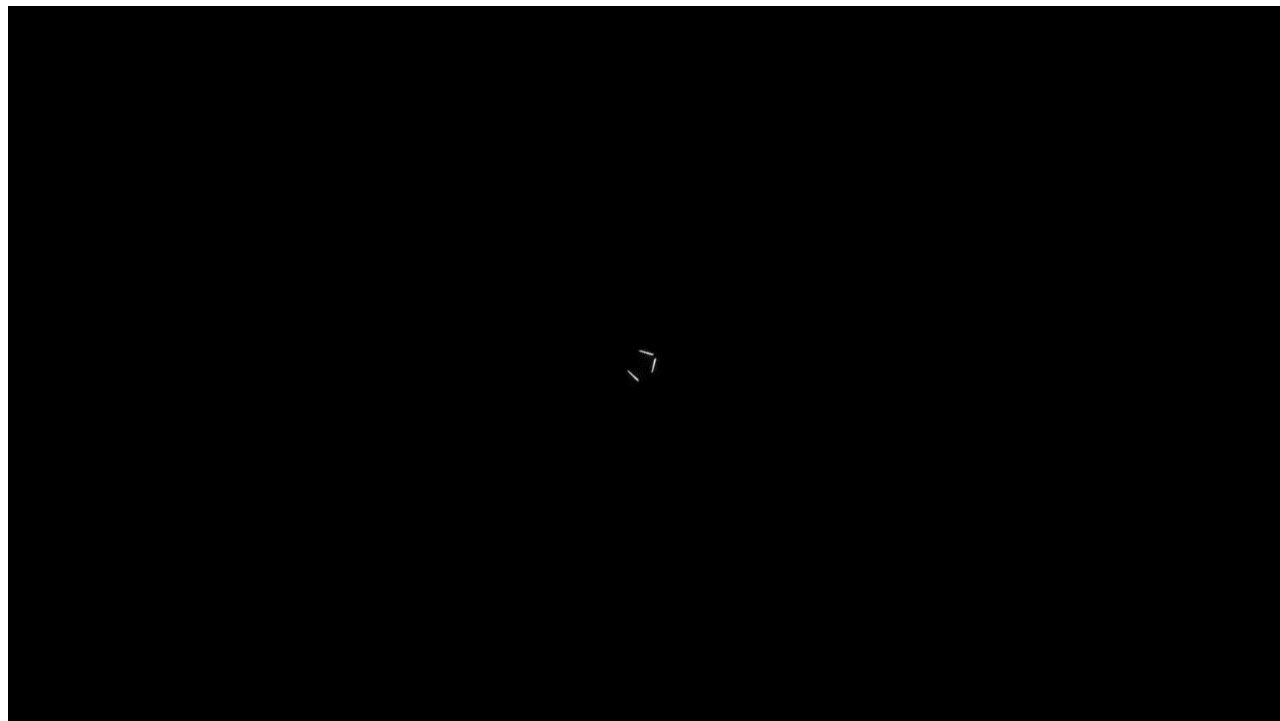
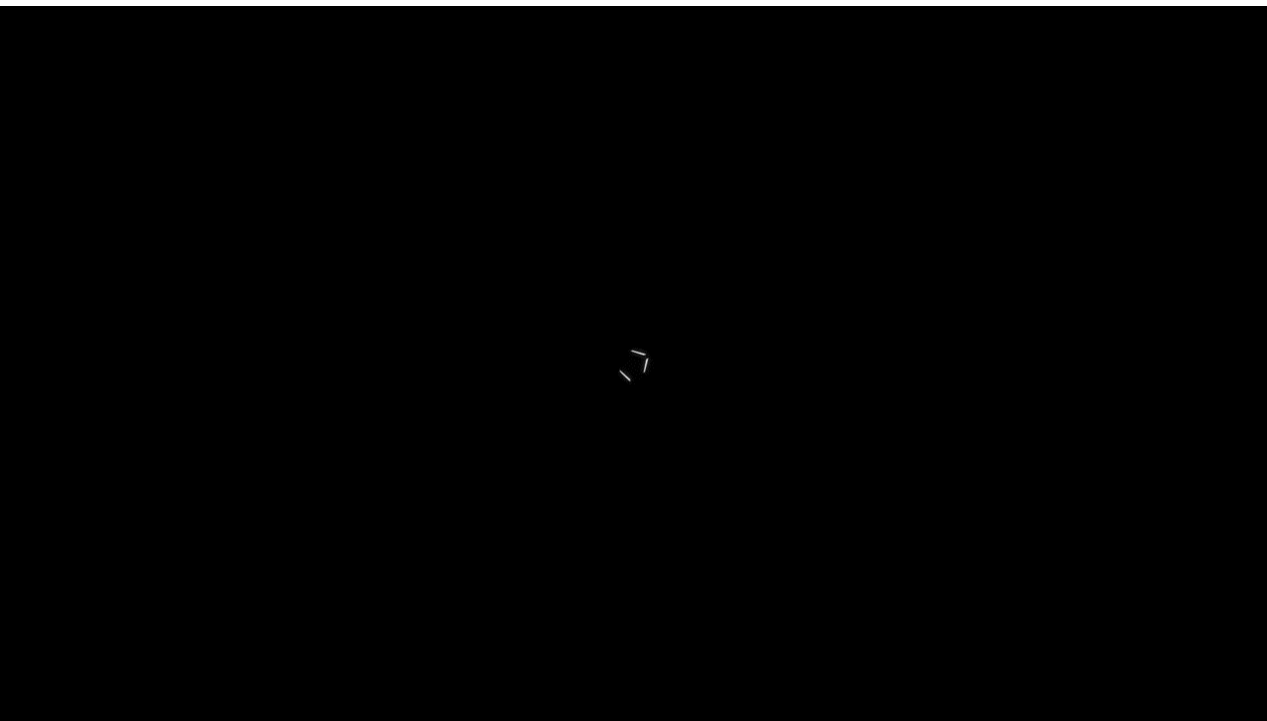




Unity ML-Agents

Unity ML-Agents v1.0

- Learning Environment 예시



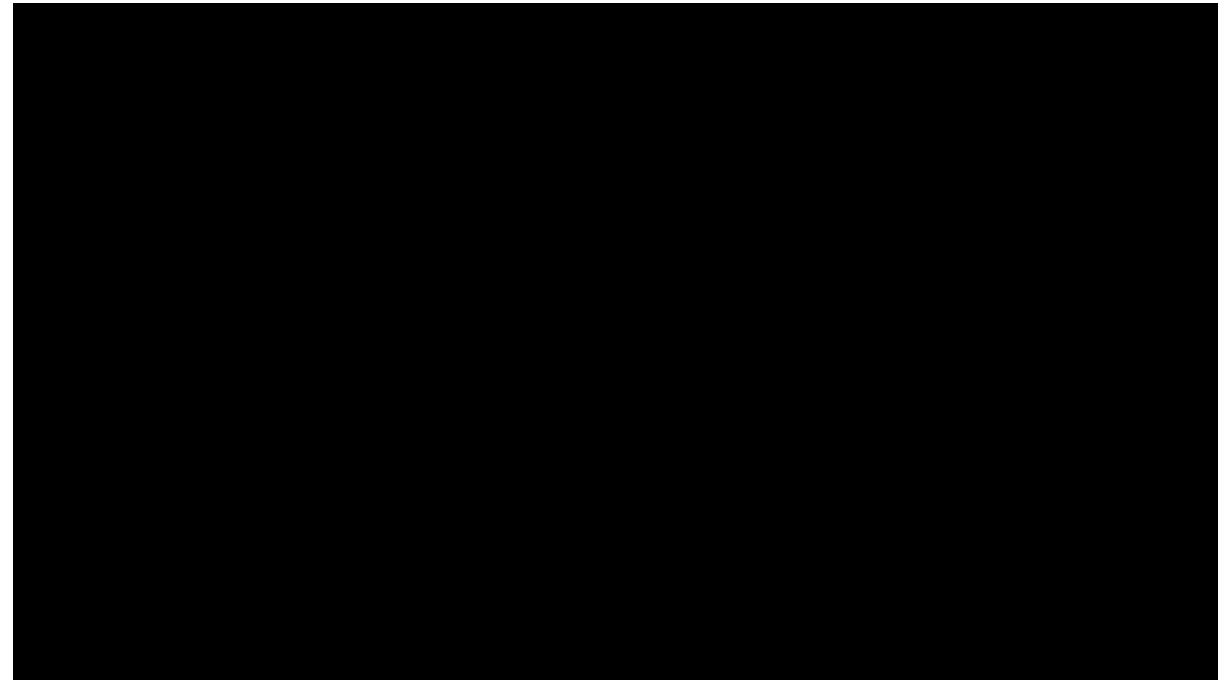
Unity ML-Agents

Unity ML-Agents v1.0

- Learning Environment 예시



<https://www.youtube.com/watch?v=aapPR-OalwY&t=9s>

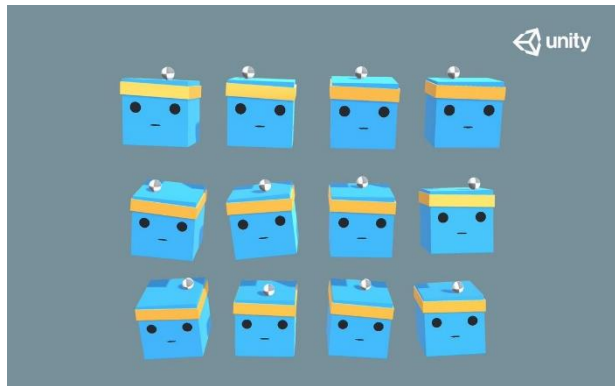


https://www.youtube.com/watch?v=VMp6pq6_QjI&t=532s

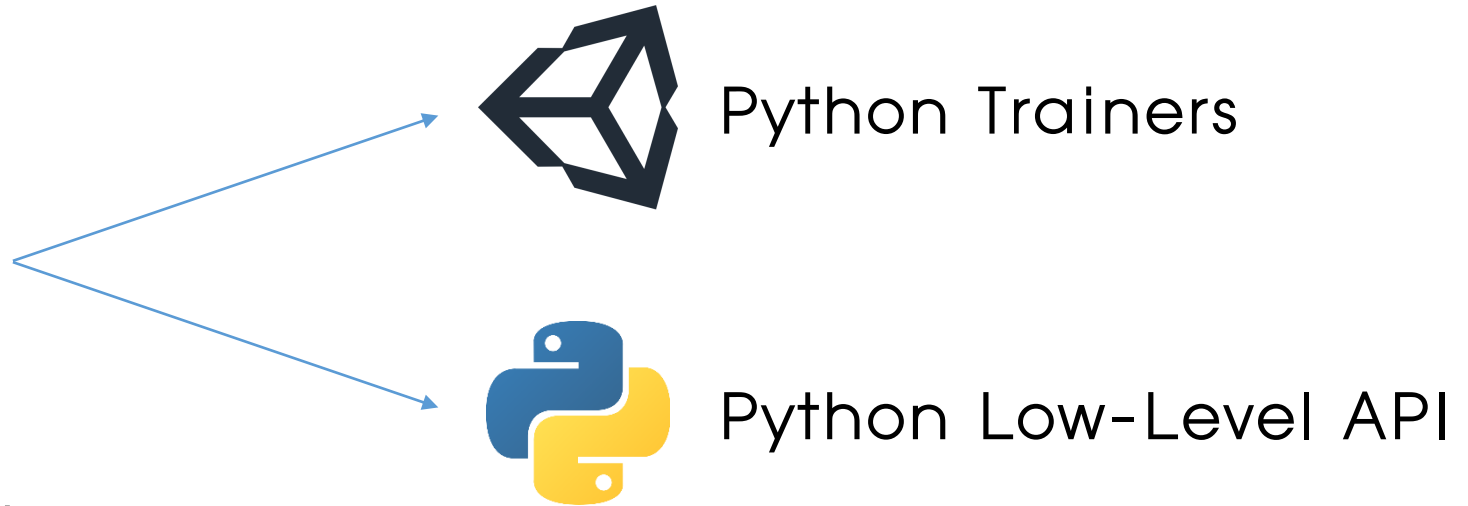
Unity ML-Agents

Unity ML-Agents v1.0

- ML-Agents를 이용한 학습 방법
 - 직접 python으로 짠 강화학습 코드 이용하기 (Python Low-Level API)
 - ML-Agents에서 제공하는 강화학습 코드 이용하기 (Python Trainers)



Learning Environment



Python Trainers

Python Low-Level API

Unity ML-Agents

Unity ML-Agents v1.0

■ ML-Agents를 이용한 학습 방법

- 직접 python으로 짠 강화학습 코드 이용하기 (Python Low-Level API)
 - 강화학습 알고리즘을 직접 개발할 필요가 있는 연구자나 개발자가 사용하는 방식
 - mlagents-envs를 통해 제공되는 ML-Agents toolkit low-level API를 통해 python 코드를 이용하여 Unity 환경과 통신 가능
 - 직접 강화학습 알고리즘을 구현하고 해당 기능을 이용하여 에이전트를 학습
 - 참고: <https://github.com/Unity-Technologies/ml-agents/blob/master/docs/Python-API.md>
- ML-Agents에서 제공하는 강화학습 코드 이용하기 (Python Trainers)
 - 강화학습 알고리즘에 대한 개발 없이 학습된 인공지능 에이전트를 얻고 싶은 경우 사용하는 방식
 - Unity ML-Agents에서 제공하는 강화학습 알고리즘을 통해 에이전트 학습
 - 학습 완료 후 생성되는 .nn 파일을 이용하여 학습된 에이전트를 유니티 환경 내에 내장 후 빌드 가능 -> 빌드된 환경에서 에이전트가 학습된대로 행동
 - 참고: <https://github.com/Unity-Technologies/ml-agents/blob/master/docs/Getting-Started.md>

Unity ML-Agents


Unity ML-Agents v1.0

- ML-Agents에서 제공하는 강화학습 알고리즘
 - Baseline Deep Reinforcement Learning Algorithms
 - Proximal Policy Optimization (<https://openai.com/blog/openai-baselines-ppo/>)
 - Soft Actor-Critic (<https://bair.berkeley.edu/blog/2018/12/14/sac/>)
 - Curiosity for Sparse-reward Environments
 - Curiosity-driven Exploration by Self-supervised Prediction (<https://pathak22.github.io/noreward-rl/>)
 - Imitation Learning
 - Behavioral Cloning
 - GAIL: Generative Adversarial Imitation Learning (<https://arxiv.org/abs/1606.03476>)

Unity ML-Agents

Unity ML-Agents v1.0

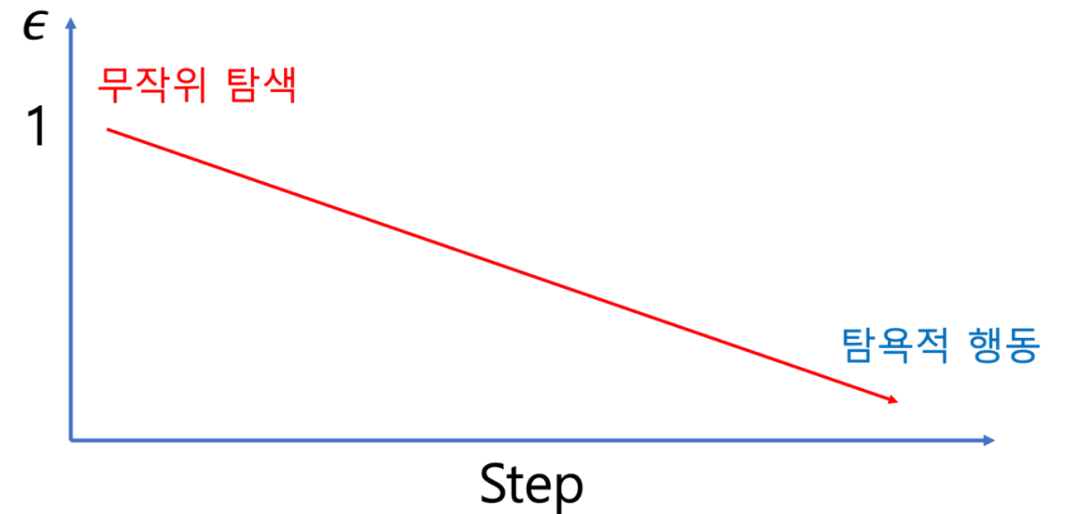
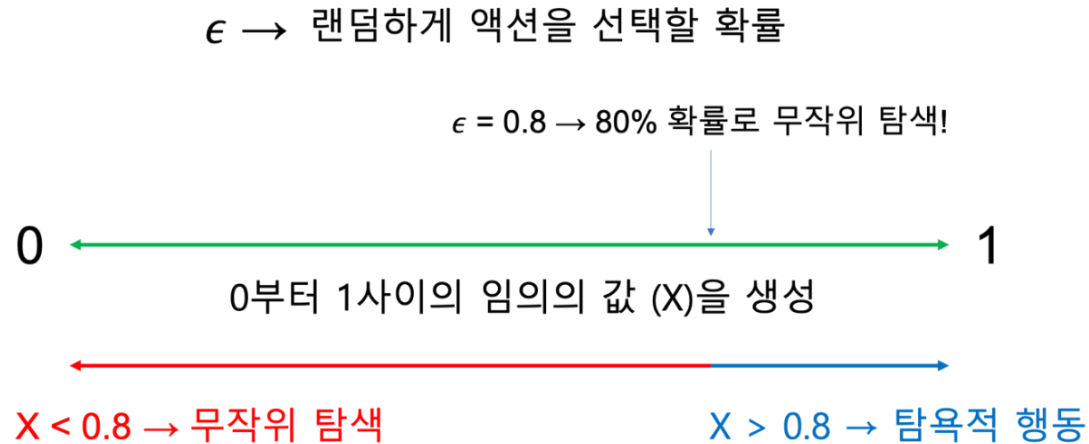
- ML-Agents에서 제공하는 강화학습 알고리즘
 - Curiosity for Sparse-reward Environments
 - Curiosity-driven Exploration by Self-supervised Prediction (<https://pathak22.github.io/noreward-rl/>)
 - 보상을 받기까지 많은 과정이 필요한 환경 (Sparse reward 환경)을 풀기 위한 알고리즘
 - 강화학습에서 중요한 문제! -> Exploration & Exploitation
 - Exploration: 에이전트가 다양한 경험을 할 수 있도록 행동 결정 (ex. Random exploration)
 - Exploitation: 에이전트가 학습된 대로 행동을 결정 (ex. Greedy method)
 - 최적의 행동을 찾기 위해서는 다양한 경험을 해봐야 한다!

	+50	
+10		+5
	+1	

Unity ML-Agents

Unity ML-Agents v1.0

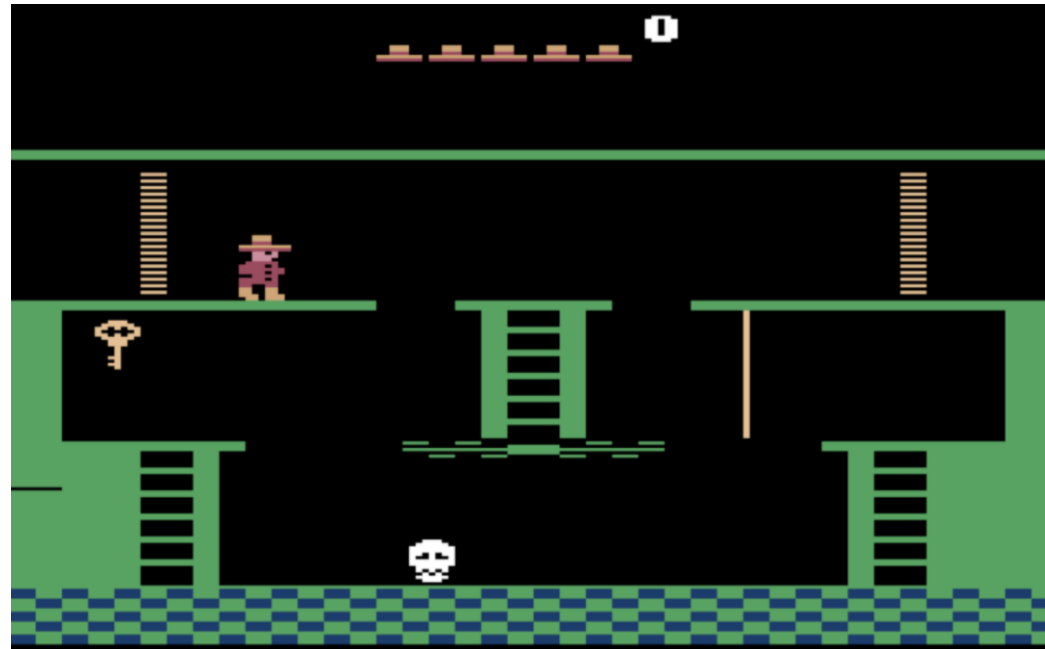
- ML-Agents에서 제공하는 강화학습 알고리즘
 - Curiosity for Sparse-reward Environments
 - ϵ -greedy: exploration과 exploitation을 적절하게 밸런스를 맞춰 수행하는 기법
 - ϵ : 무작위 행동을 취할 확률



Unity ML-Agents

Unity ML-Agents v1.0

- ML-Agents에서 제공하는 강화학습 알고리즘
 - Curiosity for Sparse-reward Environments



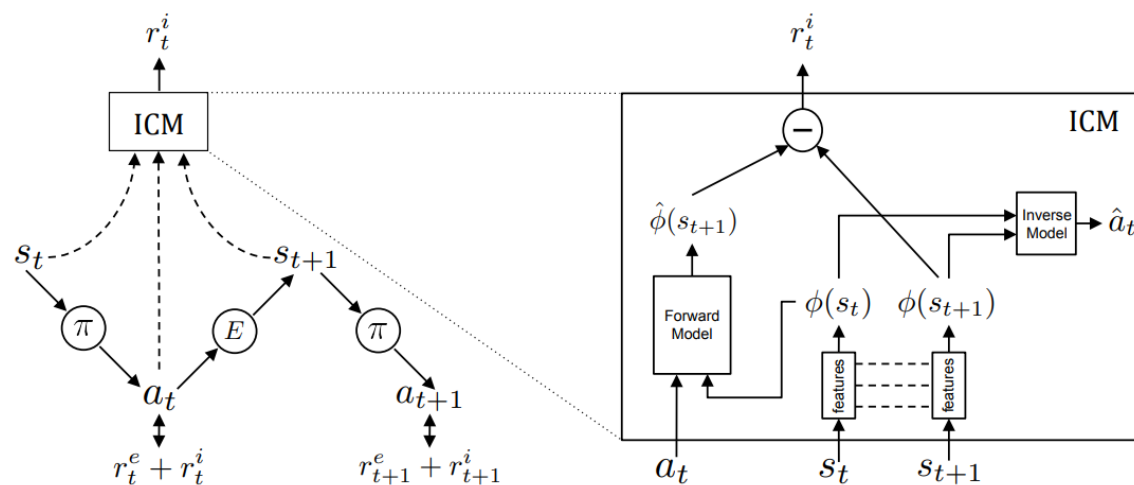
Unity ML-Agents

Unity ML-Agents v1.0

■ ML-Agents에서 제공하는 강화학습 알고리즘

■ Curiosity for Sparse-reward Environments

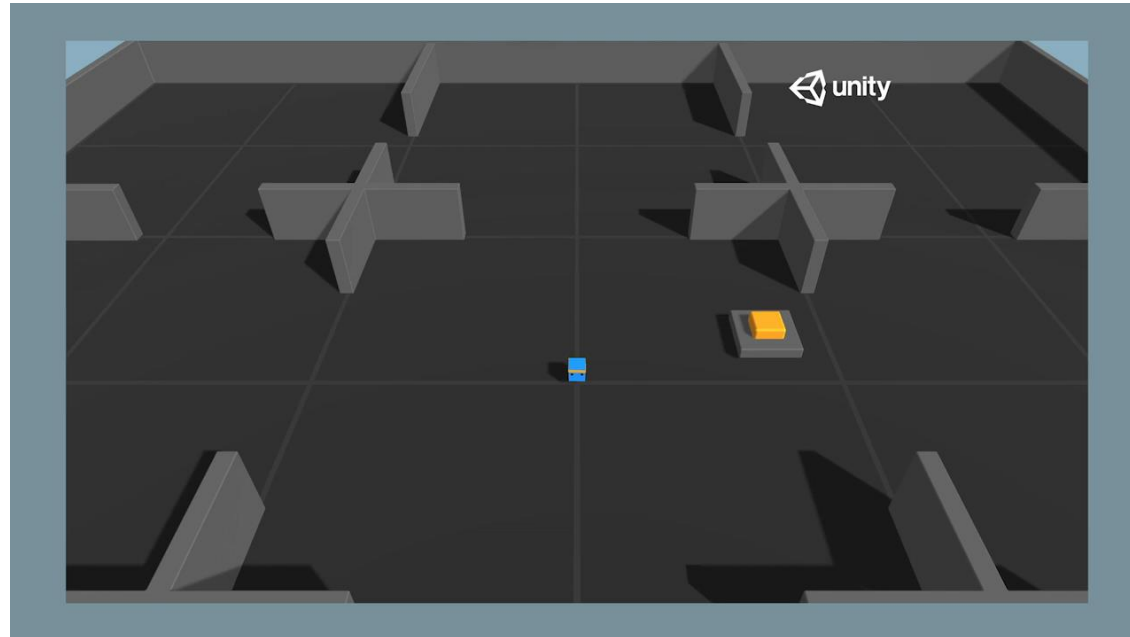
- 해당 기법은 다음 상태를 예측하기 어려운 상태에 대해 더 큰 내부 보상을 부여함 -> 새로운 경험을 하도록 에이전트를 유도 -> 다양한 경험 가능
- 내부 보상과 외부 보상 (실제 게임의 보상)을 합쳐서 학습을 위한 보상으로 사용
- 3가지 네트워크 모델 사용: Feature, Forward Model, Inverse Model
- Feature -> 상태에서 feature를 뽑아냄
- Forward model -> 현재 상태의 feature와 action을 통해 다음 상태의 feature를 예측
- Inverse model -> 현재 상태의 feature와 다음 상태의 feature를 통해 action을 추정
- Forward model을 통해 예측된 다음 상태의 feature와 실제 다음 상태의 feature간 차이를 내부 보상으로 사용



Unity ML-Agents

Unity ML-Agents v1.0

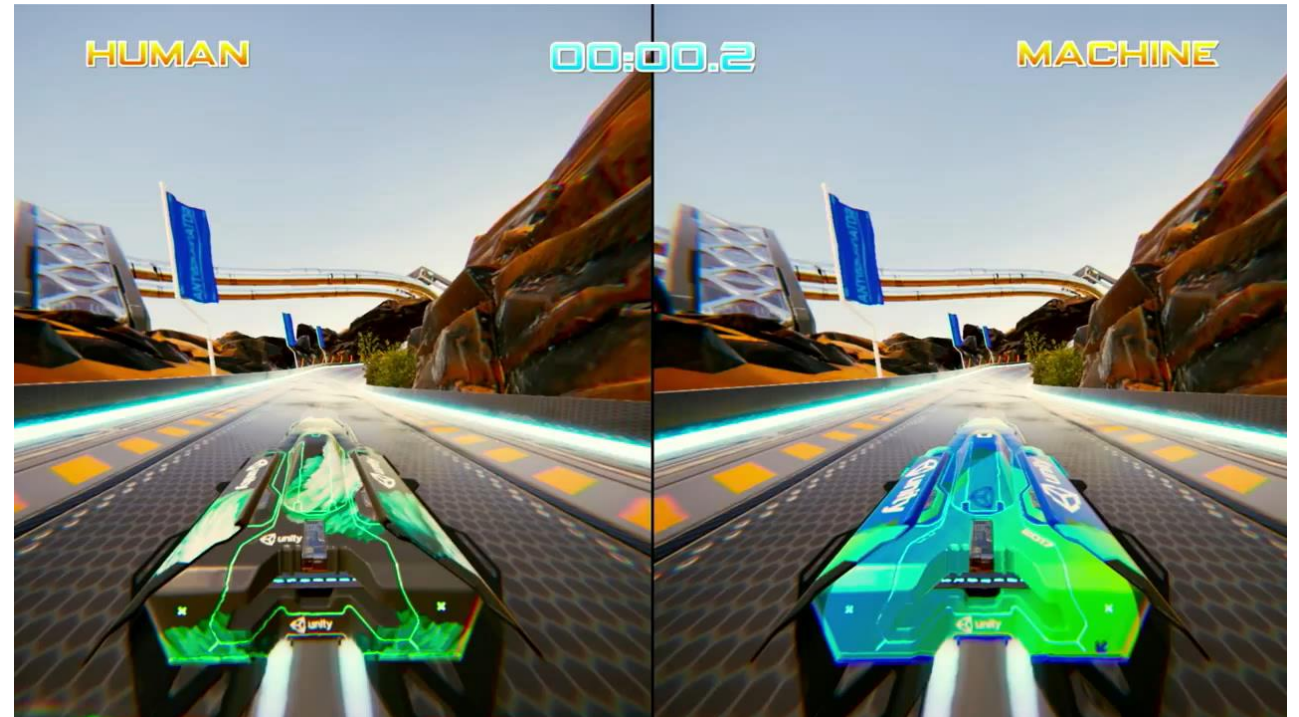
- ML-Agents에서 제공하는 강화학습 알고리즘
 - Curiosity for Sparse-reward Environments



Unity ML-Agents

Unity ML-Agents v1.0

- ML-Agents에서 제공하는 강화학습 알고리즘
 - Imitation Learning
 - 사람의 플레이를 모방하는 학습 방식
 - 사람의 플레이 데이터를 기반으로 학습 수행
 - Behavioral Cloning
 - Generative Adversarial Imitation Learning

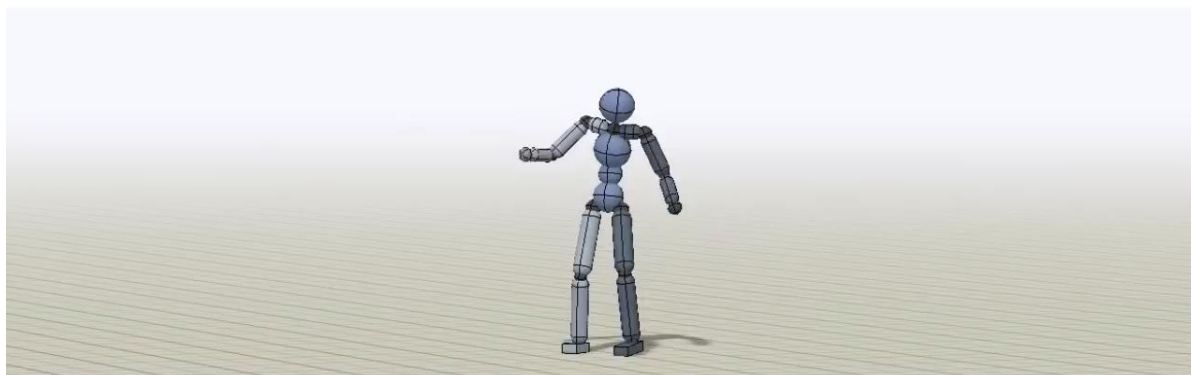


Unity ML-Agents


Unity ML-Agents v1.0

- ML-Agents에서 제공하는 강화학습 알고리즘
 - Imitation Learning

DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills



Xue Bin Peng¹, Pieter Abbeel¹, Sergey Levine¹, Michiel van de Panne²

¹ University of California
Berkeley 

² University of British
Columbia 

Unity ML-Agents

Unity ML-Agents v1.0

- ML-Agents에서 제공하는 학습 방법
 - Self-Play
 - 에이전트들이 서로 대결하며 학습하는 방법
 - Symmetric game: observation, action space, 보상 함수도 동일 -> Policy를 공유할 수 있음
 - Asymmetric game: observation, action space, 보상 함수가 다름
 - AlphaGo 등에서 사용된 방식
 - Curriculum Learning
 - 어려운 문제를 쉬운 난이도부터 순차적으로 학습하는 방식
 - Environment Parameter Randomization
 - 환경내의 다양한 파라미터를 변경해가면서 강인한 에이전트를 학습하는 방식

Unity ML-Agents

Unity ML-Agents v1.0

- ML-Agents에서 제공하는 학습 방법
 - Self-Play (Symmetric games)



Unity ML-Agents

Unity ML-Agents v1.0

- ML-Agents에서 제공하는 학습 방법
 - Self-Play (Asymmetric games)

A rectangular box with a vertical gradient from light red at the top to light blue at the bottom. The text "Multi-Agent Hide and Seek" is centered in a dark blue font.

Multi-Agent
Hide and Seek

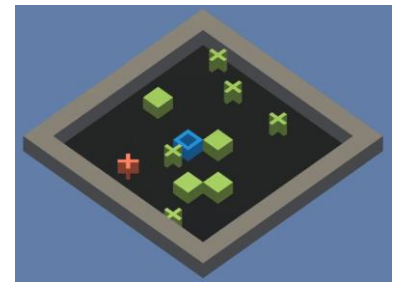
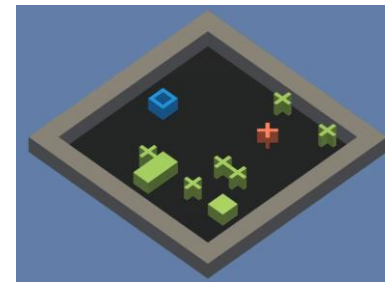
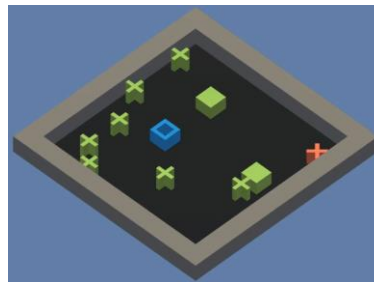
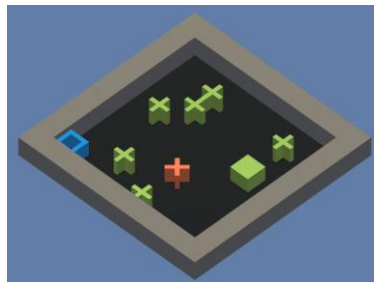
Unity ML-Agents

Unity ML-Agents v1.0

- ML-Agents에서 제공하는 학습 방법

- Curriculum Learning

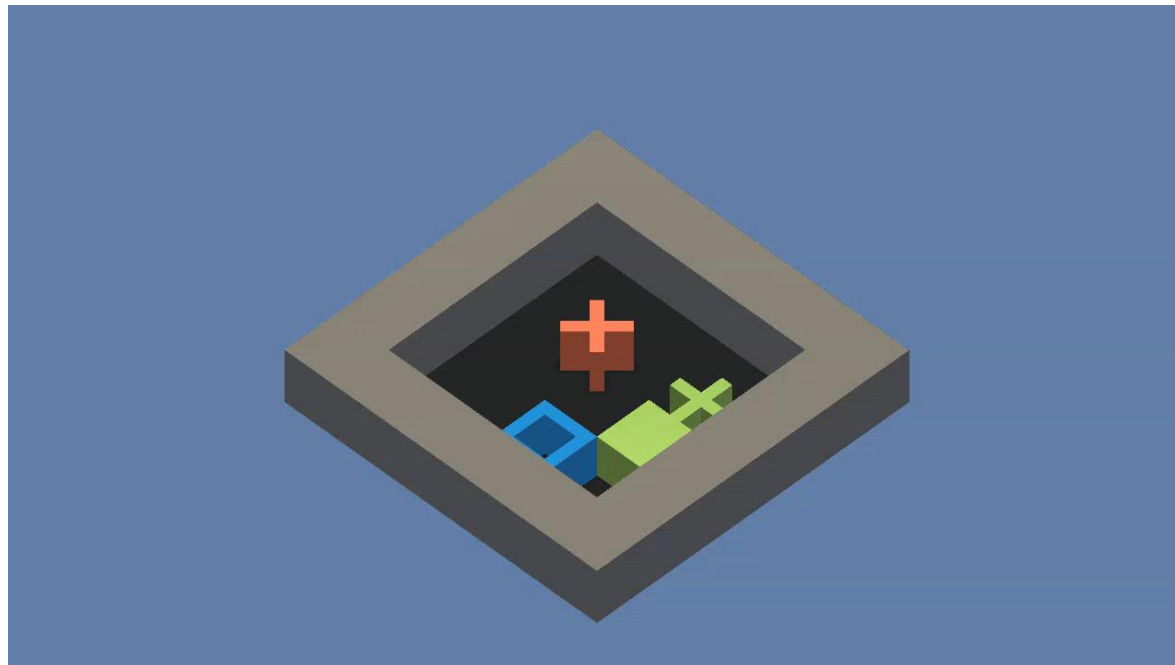
- Sokoban 환경
 - 학습이 진행될수록 게임판이 커지며 박스 수가 늘어남
 - 처음부터 가장 어려운 난이도 문제를 학습하기 어렵기 때문에 순차적으로 단계를 높여가면서 학습 수행



Unity ML-Agents

Unity ML-Agents v1.0

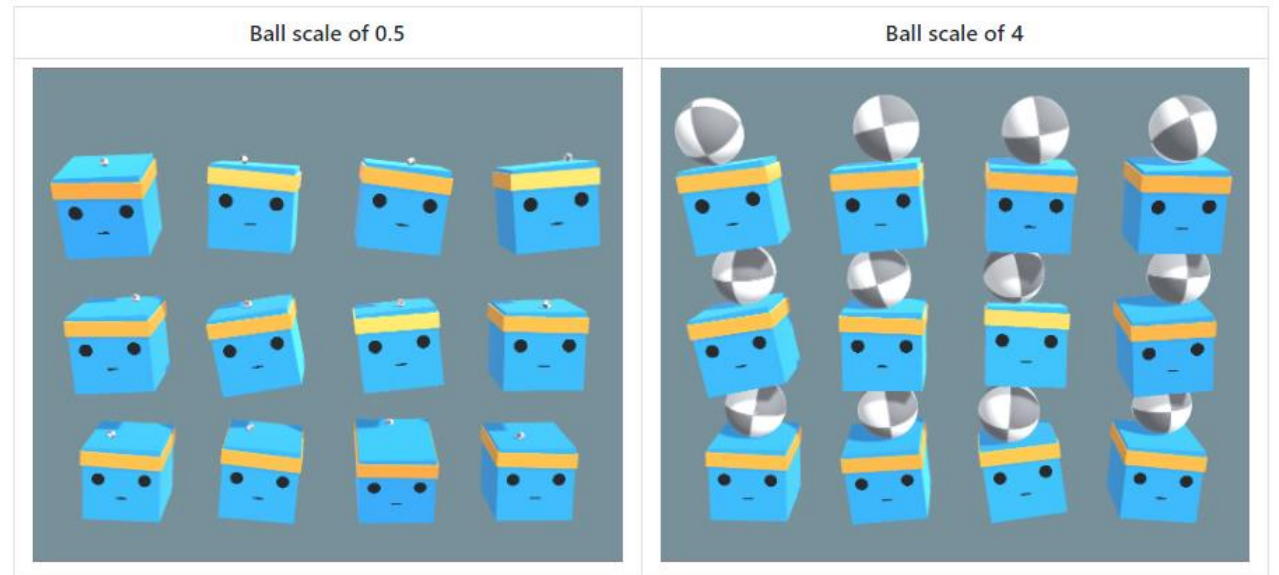
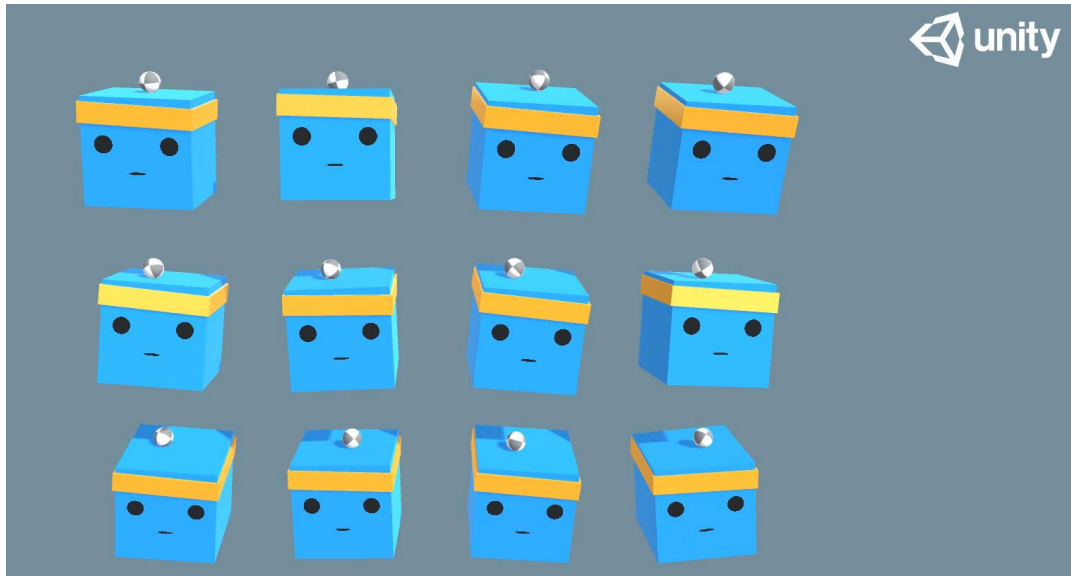
- ML-Agents에서 제공하는 학습 방법
 - Curriculum Learning



Unity ML-Agents

Unity ML-Agents v1.0

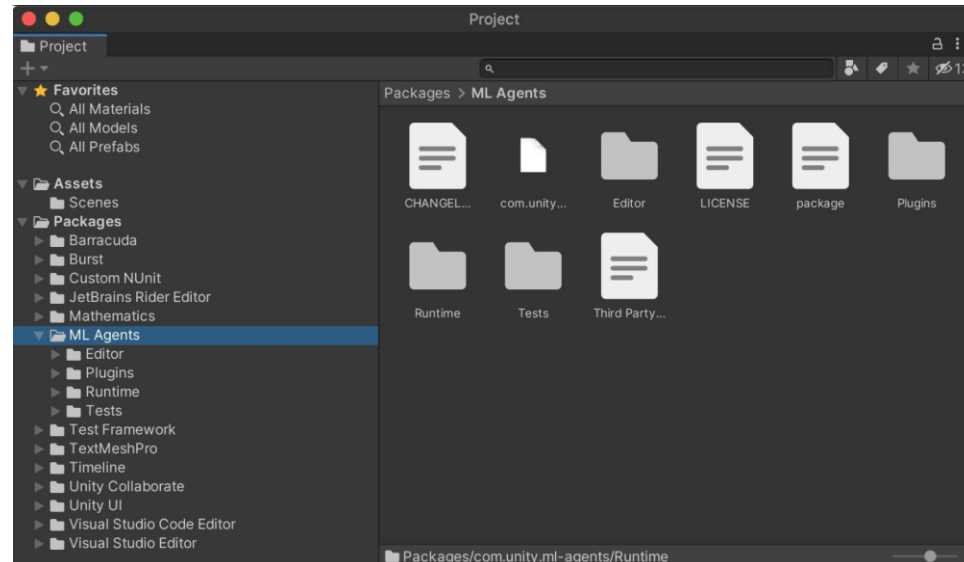
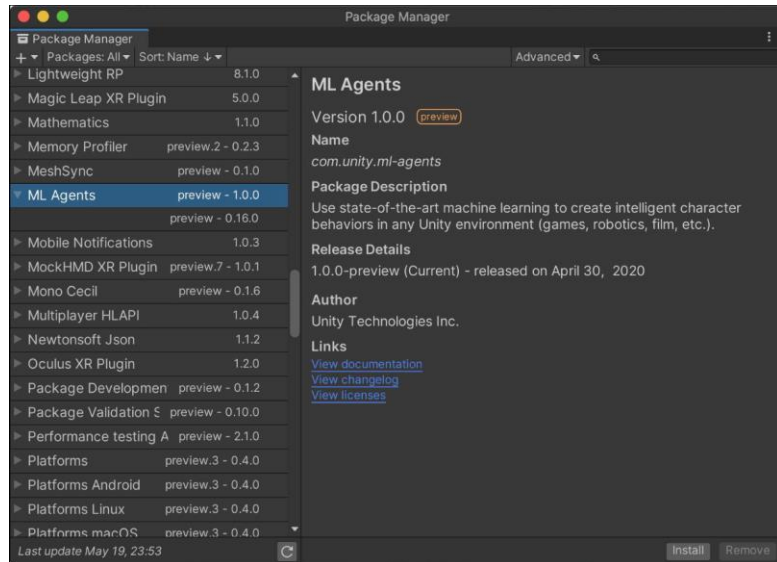
- ML-Agents에서 제공하는 학습 방법
 - Environment Parameter Randomization
 - 환경내의 다양한 파라미터를 변경해가면서 강인한 에이전트를 학습하는 방식



Unity ML-Agents

Unity ML-Agents v1.0

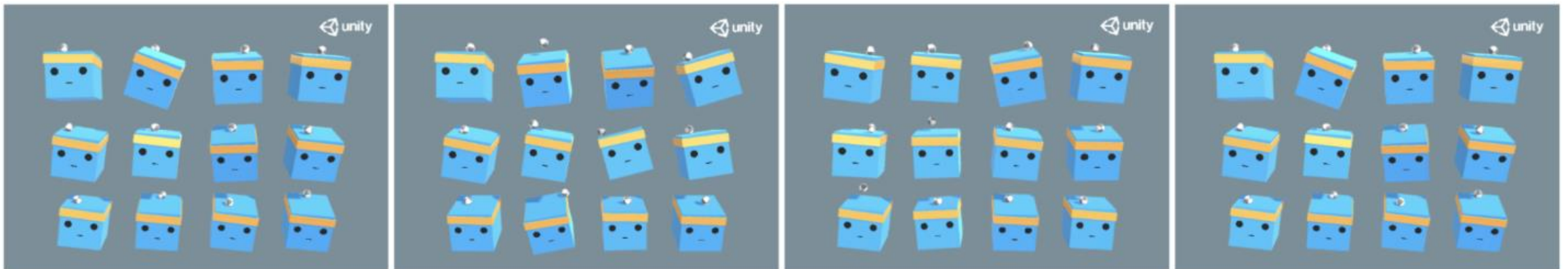
- 앞으로 ML-Agents의 발전 방향
 - Unity 2020.2 버전부터 패키지로 내장되어 출시될 예정
 - 깃허브에서 Unity ML-Agents를 다운받을 필요 없이 유니티 설치 후 바로 사용 가능



Unity ML-Agents

Unity ML-Agents v1.0

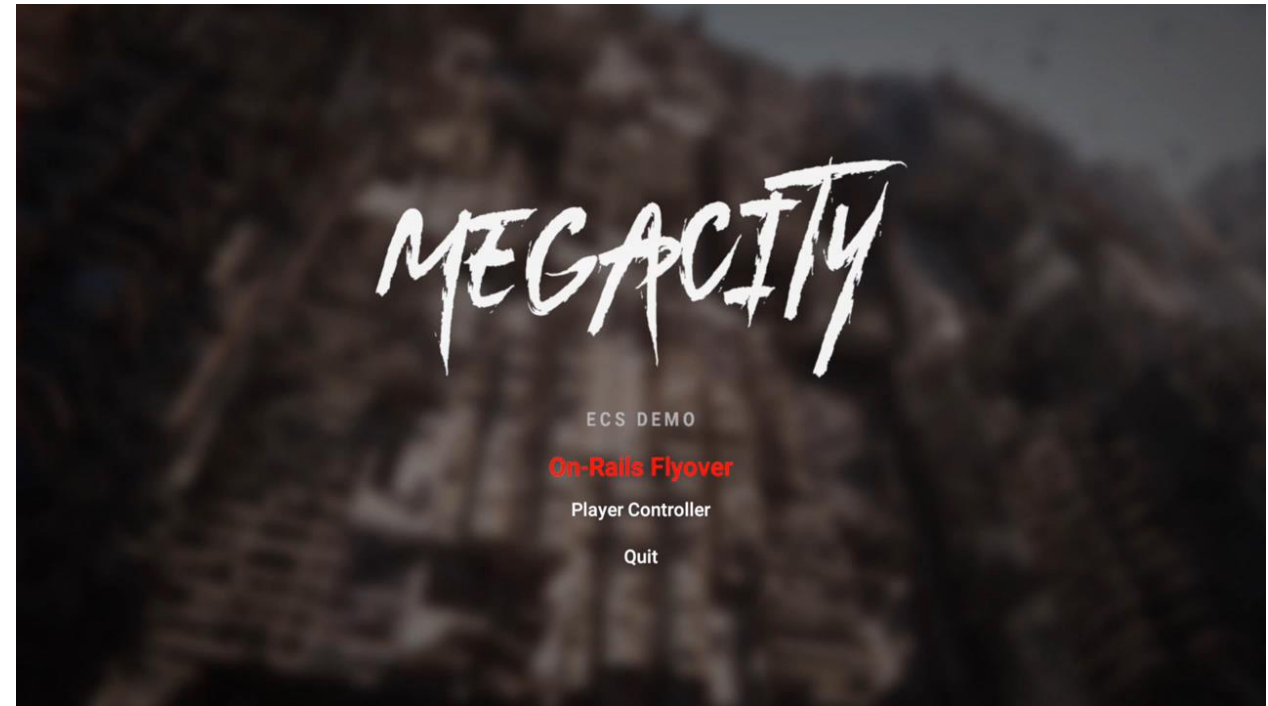
- 앞으로 ML-Agents의 발전 방향
 - 연말에 ML-Agents cloud 출시 예정
 - Cloud 환경으로 ML-Agents 사용자들이 에이전트를 학습시킬 수 있도록 서비스를 제공할 예정
 - https://create.unity3d.com/ml-agents-cloud-training?_ga=2.109394821.1490974290.1591342055-654746196.1533086514



Unity ML-Agents

Unity ML-Agents v1.0

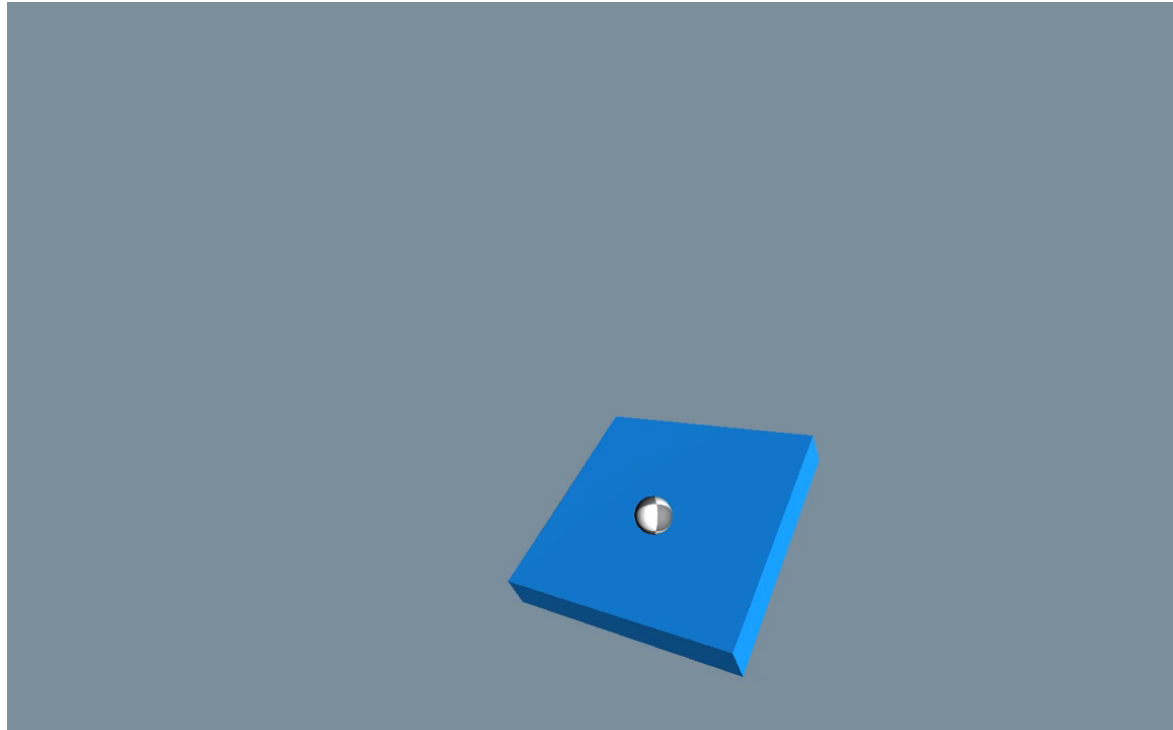
- 앞으로 ML-Agents의 발전 방향
 - Data-Oriented Technology Stack (DOTS)를 사용하여 Unity의 core를 rebuild
 - 특별한 프로그래밍 없이 멀티 코어 프로세서를 이용해 게임을 더 빠르게 실행 -> 성능 향상
 - 복잡하게 환경의 경우도 일반 노트북으로 수월하게 실행할 수 있음
 - ML-Agents for DOTS를 연말에 출시할 예정



Unity ML-Agents

Unity ML-Agents v1.0

- 앞으로 ML-Agents의 발전 방향



Unity ML-Agents

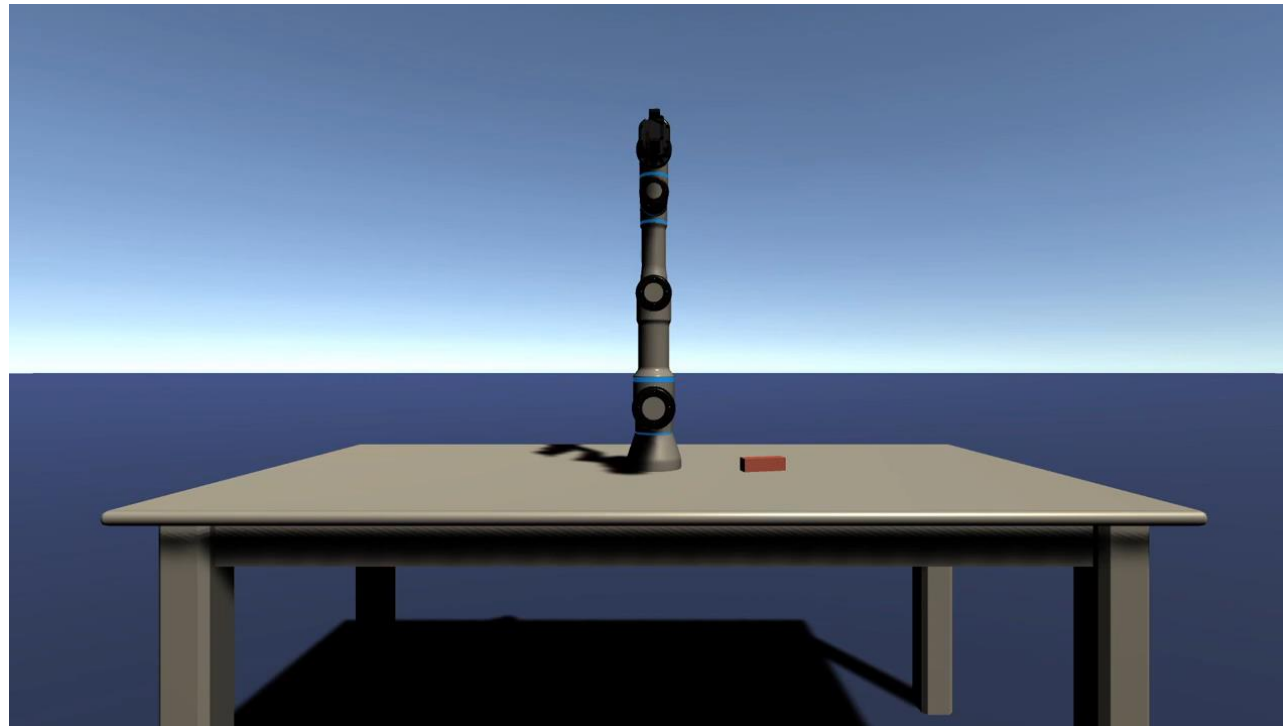
Unity ML-Agents v1.0

- 앞으로 ML-Agents의 발전 방향
 - NVIDIA PhysX 4.0을 이용해 robot simulation의 퀄리티를 향상시켰음 (Unity 2019.3)
 - Unity 2020.1에 Nvidia PhysX 4.1을 이용하여 새로운 articulation joint system 적용 -> 로봇팔과 continuous joint의 성능을 크게 향상시킴
 - Demo github: <https://github.com/Unity-Technologies/articulations-robot-demo/tree/mlagents>
 - Unity 2020.1 (beta)
 - ML-Agents Release 1 적용
 - ML-Agents 내부에 robotics나 continuous control 환경을 더 많이 포함하여 공개 예정

Unity ML-Agents

Unity ML-Agents v1.0

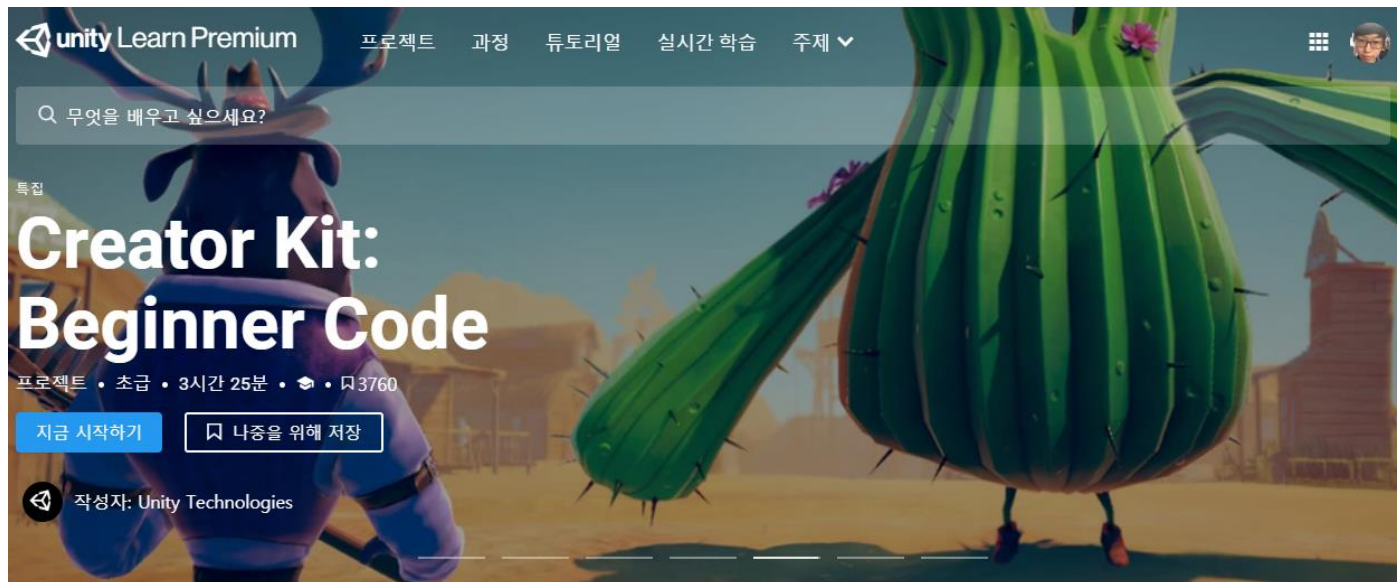
- 앞으로 ML-Agents의 발전 방향



Unity ML-Agents

Unity ML-Agents v1.0

- ML-Agents를 통한 환경 만드는 법 배우기
 - Unity Learn
 - <https://learn.unity.com/>



Unity ML-Agents

Unity ML-Agents v1.0

- ML-Agents Release 1 실행법 배워보기
 - ML-Agents: Hummingbirds 환경
 - https://learn.unity.com/course/ml-agents-hummingbirds?_ga=2.144029877.1490974290.1591342055-654746196.1533086514



Unity ML-Agents

Unity ML-Agents Github

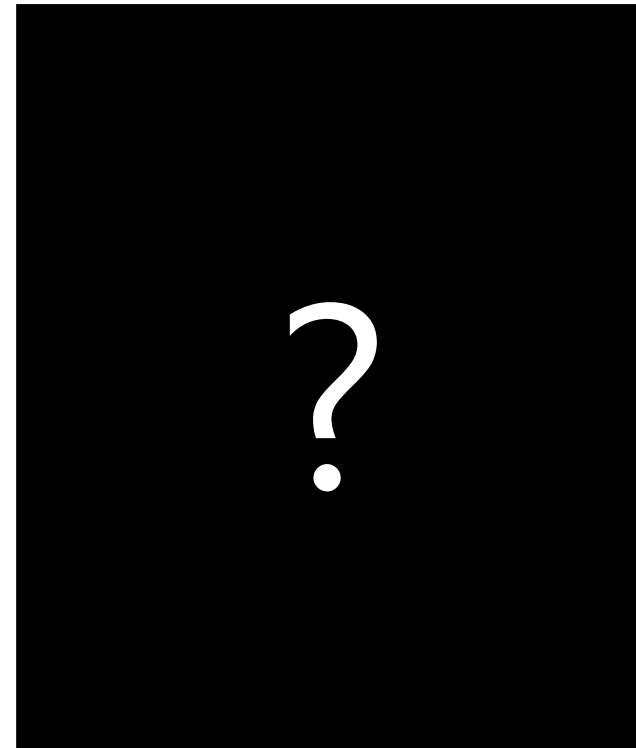
- <https://github.com/Unity-Technologies/ml-agents>

The screenshot shows the GitHub repository page for Unity ML-Agents. At the top, the repository name "Unity-Technologies / ml-agents" is displayed. To the right, there are statistics: "Used by 156", "Watch 519", "Unstar 8.8k", and "Fork 2.4k". Below this, a navigation bar includes "Code", "Issues 109", "Pull requests 21", "Actions", "Projects 0", "Security 0", and "Insights". The repository description is "Unity Machine Learning Agents Toolkit" with the URL "https://unity3d.ai". Below the description are tags: "reinforcement-learning", "unity3d", "deep-learning", "unity", "deep-reinforcement-learning", and "neural-networks". At the bottom, a summary bar shows: "2,294 commits", "103 branches", "0 packages", "58 releases", "99 contributors", and "Apache-2.0".

Unity ML-Agents

Unity ML-Agents

<https://wikibook.co.kr/tensorflow-mlagents/>



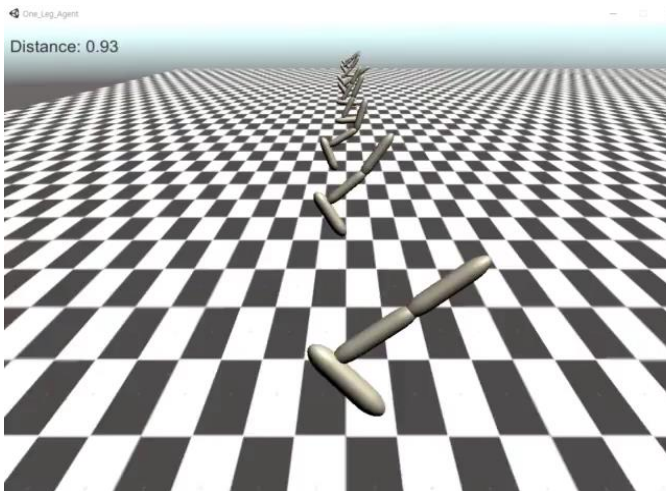
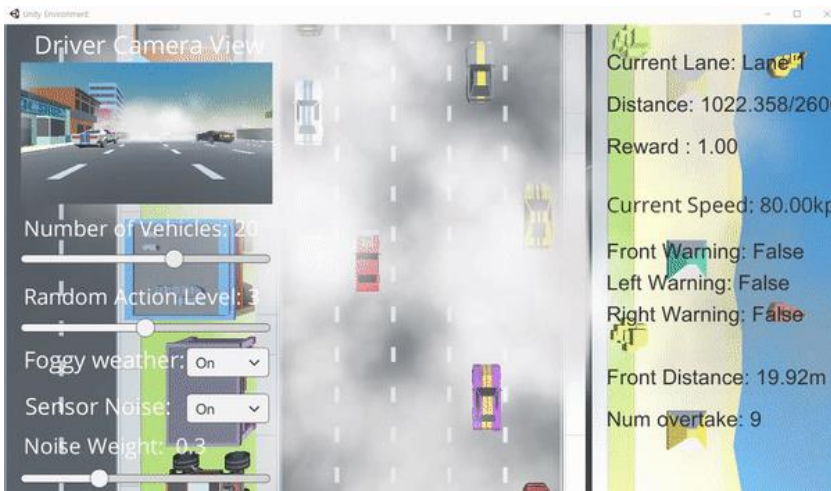
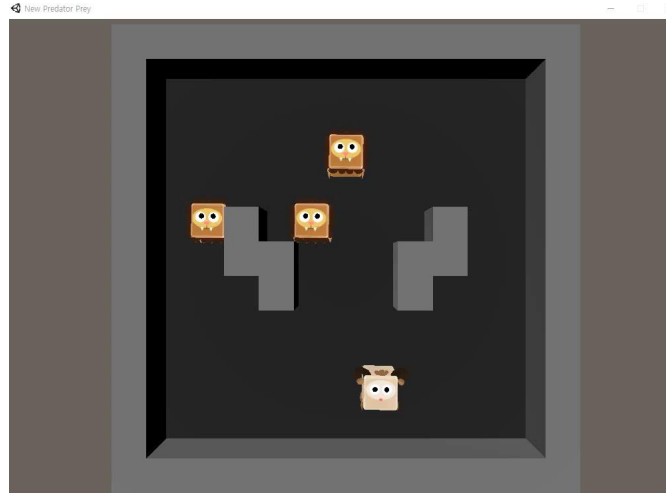
Unity ML-Agents

Reinforcement Learning Korea

<https://www.facebook.com/groups/ReinforcementLearningKR/?ref=bookmarks>



ML-Agents를 이용한 개발 사례 소개



ML-Agents를 이용한 개발 사례 소개

MACHINE LEARNING CAMP JEJU 2017

Google 한국 블로그

Google에 대한 다양한 소식을 확인해 보세요!

구글·카카오 등 '머신러닝 캠프 제주 2017' 공동 개최..."AI 연구 활성화 지원"

조선비즈 | 김범수 기자



입력 2017.07.03 17:39

'머신러닝 캠프 제주 2017' 인공지능기술 대중화 캠프 개최

정동희 기자 | 승인 2017.07.03 15:01 | 댓글 0



구글이 머신러닝 캠프 제주 2017과 함께 합니다!

2017년 7월 3일 월요일

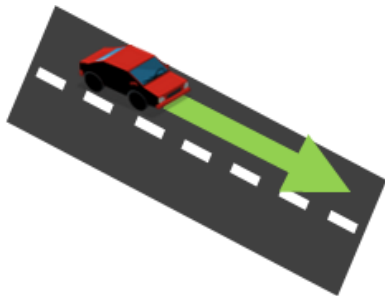


ML-Agents를 이용한 개발 사례 소개

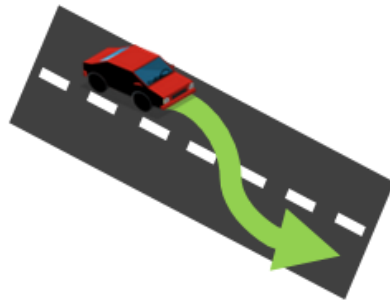
Machine Learning Camp Jeju 2017

■ Project Proposal

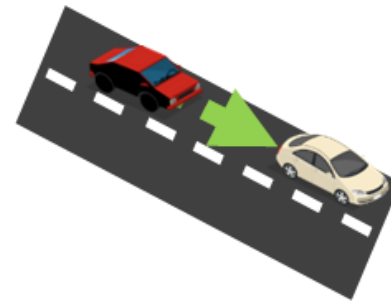
- 차선 유지, 차선 변경 등 다양한 운전자 보조 시스템(ADAS)이 이미 상용화
- 운전자 보조 시스템의 조합만으로도 자율주행이 가능하지 않을까?
- 문제는 매 상황마다 어떤 ADAS를 통해 차량을 제어할지 결정하는 것!
=> 강화학습을 이용하자!



차선 유지



차선 변경



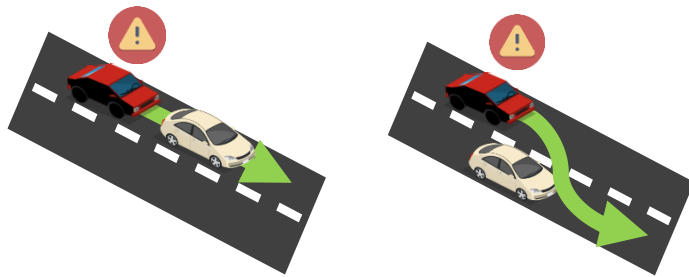
앞 차량과 거리 유지

ML-Agents를 이용한 개발 사례 소개

Machine Learning Camp Jeju 2017

■ Project Proposal

- 강화학습의 경우 액션 선택에 대한 예측이 불가능
- 긴급 제동, 차선 변경 방지와 같은 충돌방지 시스템 도입
- 자율주행차량에서 많이 사용하는 카메라, LIDAR, RADAR 같은 센서 사용
- 하지만 원하는 조건을 만족하는 시뮬레이터가 없는 상황



충돌 방지 시스템

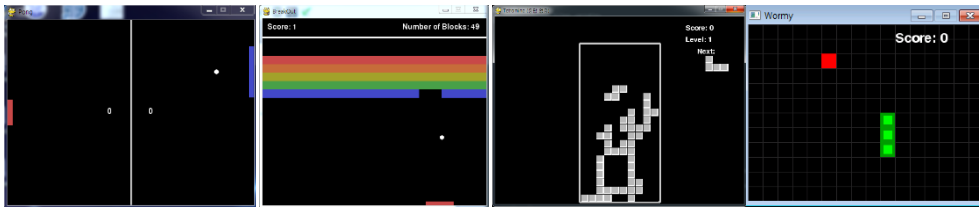


카메라, LIDAR, RADAR

ML-Agents를 이용한 개발 사례 소개

Machine Learning Camp Jeju 2017

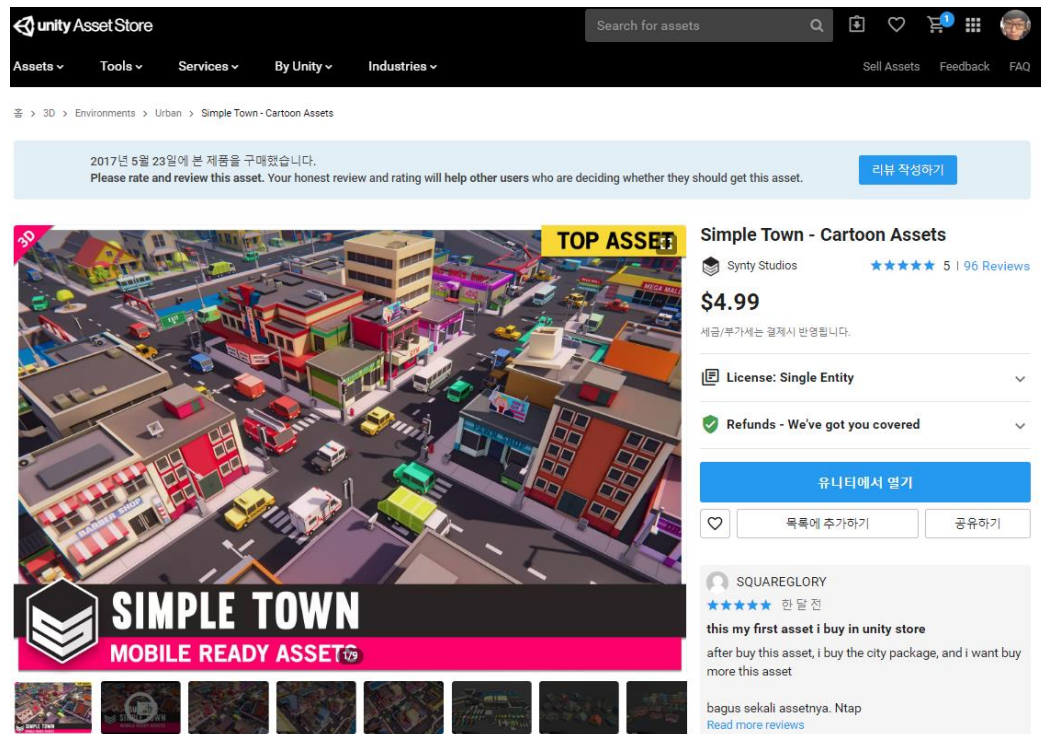
- Simulator
 - Pygame의 경우 3D 게임에 대한 제작이 불가능
 - 센서 제작 및 충돌에 대한 코딩이 어려움
 - 물리계에 대한 구현이 없음



ML-Agents를 이용한 개발 사례 소개

Machine Learning Camp Jeju 2017

■ Simulator



The screenshot shows the Unity Asset Store interface. At the top, there's a search bar and navigation tabs for Assets, Tools, Services, By Unity, and Industries. Below that, the breadcrumb path is '3D > Environments > Urban > Simple Town - Cartoon Assets'. A notification banner states: '2017년 5월 23일에 본 제품을 구매했습니다. Please rate and review this asset. Your honest review and rating will help other users who are deciding whether they should get this asset.' The main product listing is for 'Simple Town - Cartoon Assets' by Synty Studios, priced at \$4.99. It features a 'TOP ASSET' badge, a 5-star rating with 5,196 reviews, and a license of 'Single Entity'. A blue button says '유니티에서 열기'. Below the main image, there are smaller thumbnails and a testimonial from user SQUAREGLORY: 'this my first asset i buy in unity store after buy this asset, i buy the city package, and i want buy more this asset'. Another testimonial from Ntap says 'bagus sekali assetnya. Ntap'.

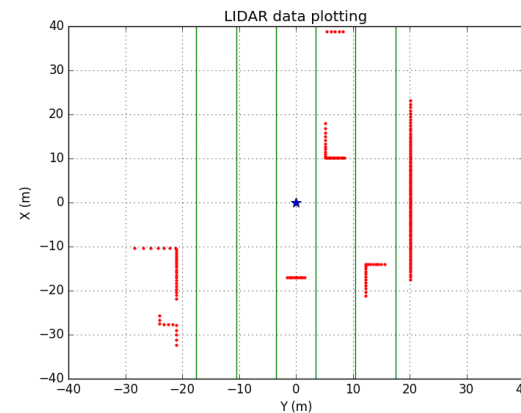
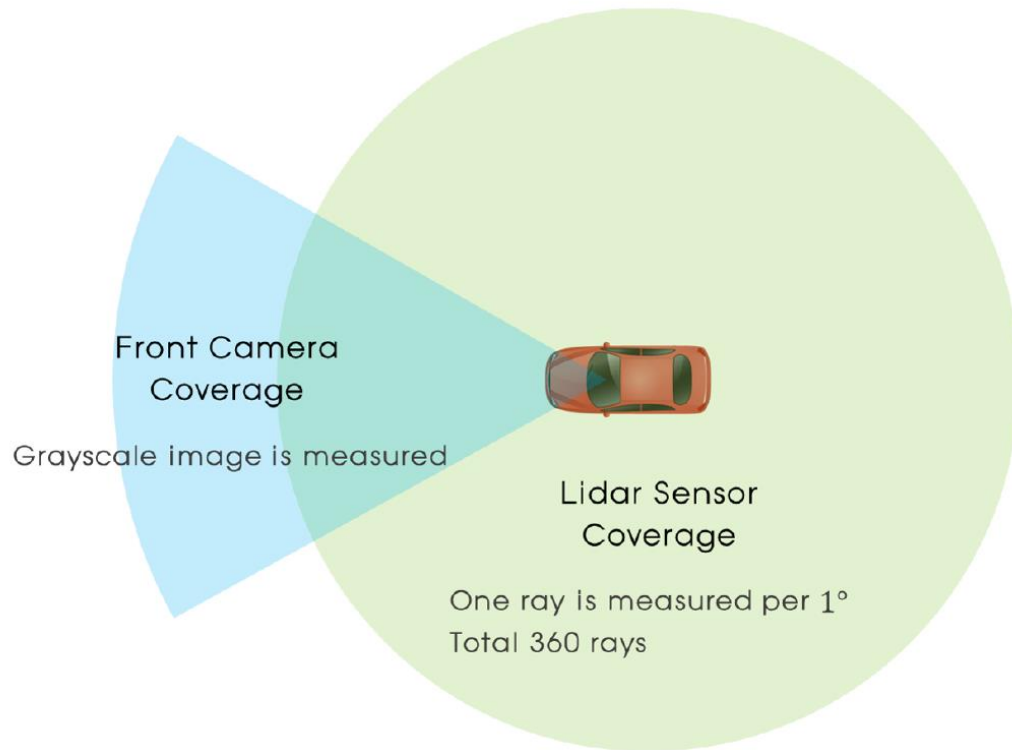


The screenshot displays the 'Vehicle Simulator' interface. On the left, there are two camera viewports: 'Front Camera View' and 'Rear Camera View'. The central part of the screen shows a top-down view of a road with several cars in various colors (purple, yellow, red, white) and a green car with the number '32'. On the right side, there is a data panel with the following information: 'Longitudinal Velocity Range: 20 - 40', 'Lateral Velocity Range: 5 - 10', 'Current Speed: 20', and 'Current Lane: Lane 3'. The background of the right panel shows a stylized road layout with a yellow center line and blue lanes.

ML-Agents를 이용한 개발 사례 소개

Machine Learning Camp Jeju 2017

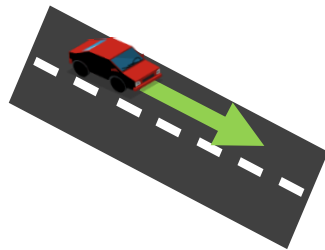
- Simulator (Observations)



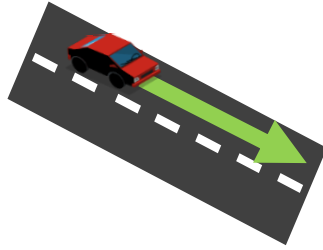
ML-Agents를 이용한 개발 사례 소개

Machine Learning Camp Jeju 2017

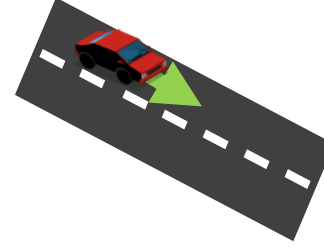
- Simulator (Actions)



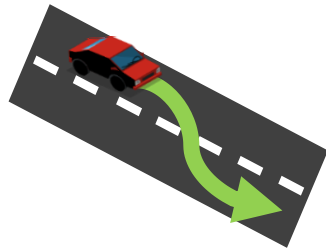
현재 상태 유지



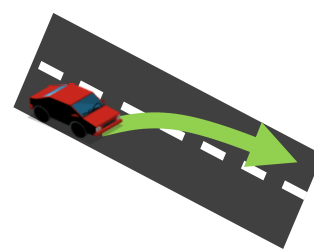
가속



감속



차선 변경 (오른쪽)



차선 변경 (왼쪽)

ML-Agents를 이용한 개발 사례 소개

Machine Learning Camp Jeju 2017

- Simulator (Reward)

$r_{v,max}$	1
v_{max}	80km/h
v_{min}	40km/h
$r_{collision}$	-10
r_{lc}	-0.25
$r_{overtake}$	0.5

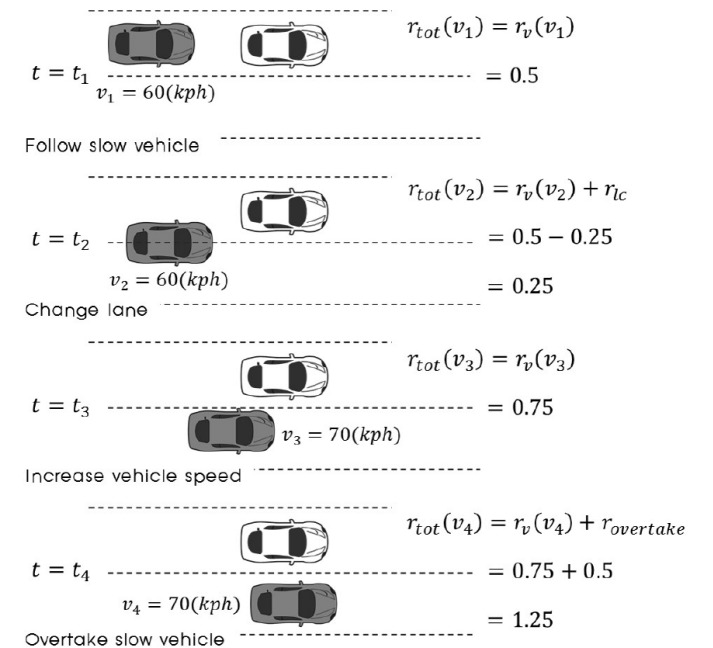
$$r_v(v) = \frac{v - v_{min}}{v_{max} - v_{min}} r_{v,max} \quad (1)$$

$$r_{col} = \begin{cases} -r_{collision} & \text{if host vehicle colides} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$r_{lc} = \begin{cases} -r_{lanechange} & \text{if host vehicle changes lane} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$r_{overtake} = \begin{cases} r_{overtake} & \text{if host vehicle overtake other vehicle} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

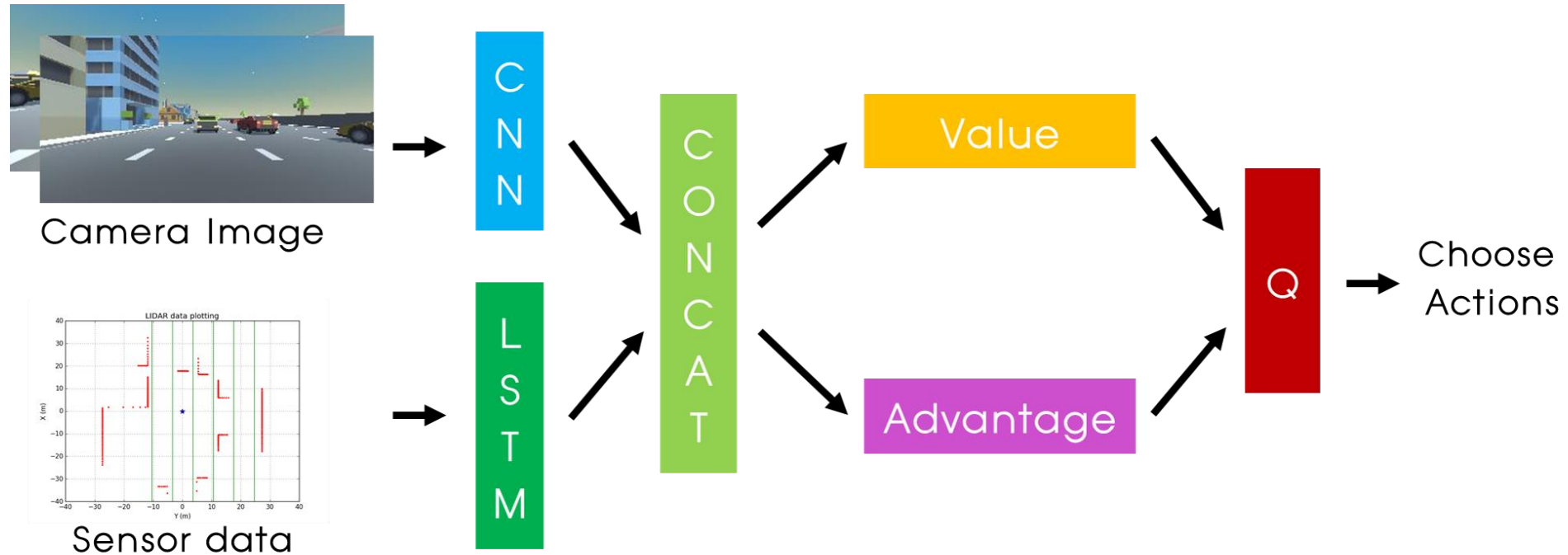
$$r_{tot}(v) = r_v(v) + r_{col} + r_{lc} + r_{overtake} \quad (5)$$



ML-Agents를 이용한 개발 사례 소개

Machine Learning Camp Jeju 2017

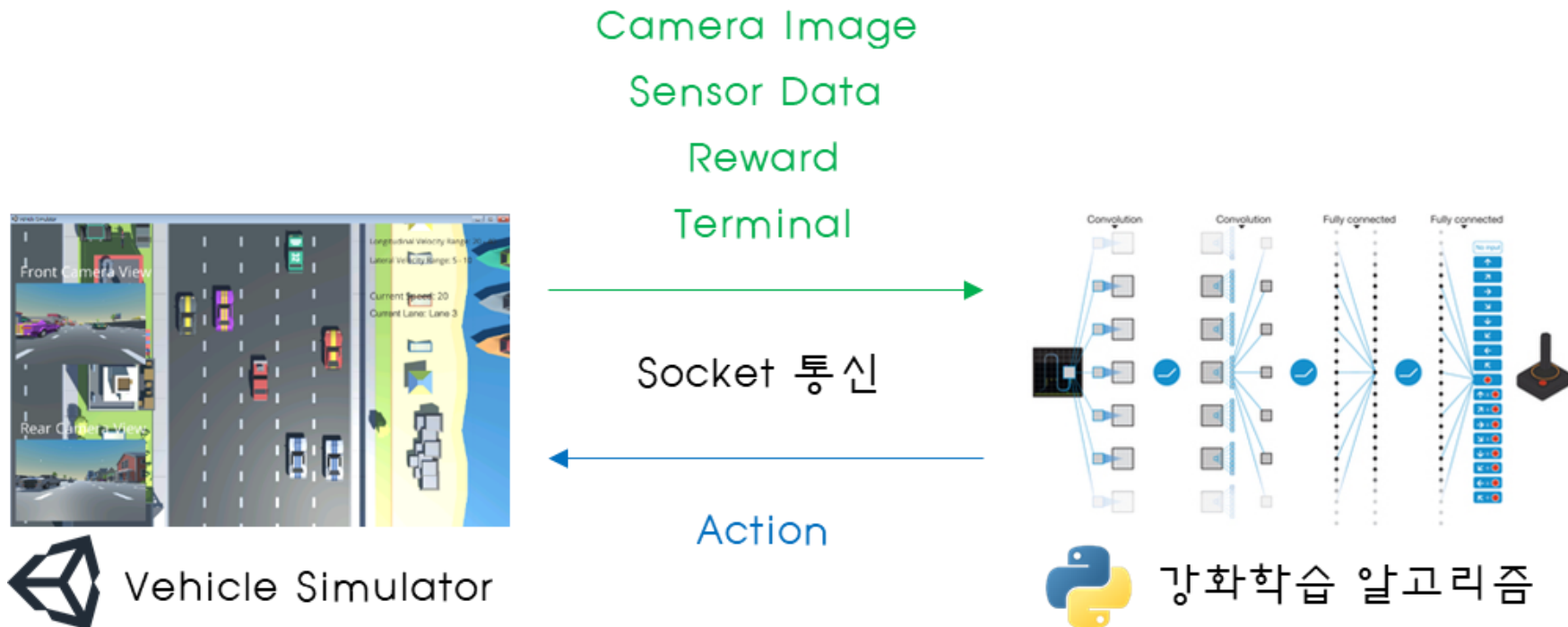
■ Network Architecture



ML-Agents를 이용한 개발 사례 소개

Machine Learning Camp Jeju 2017

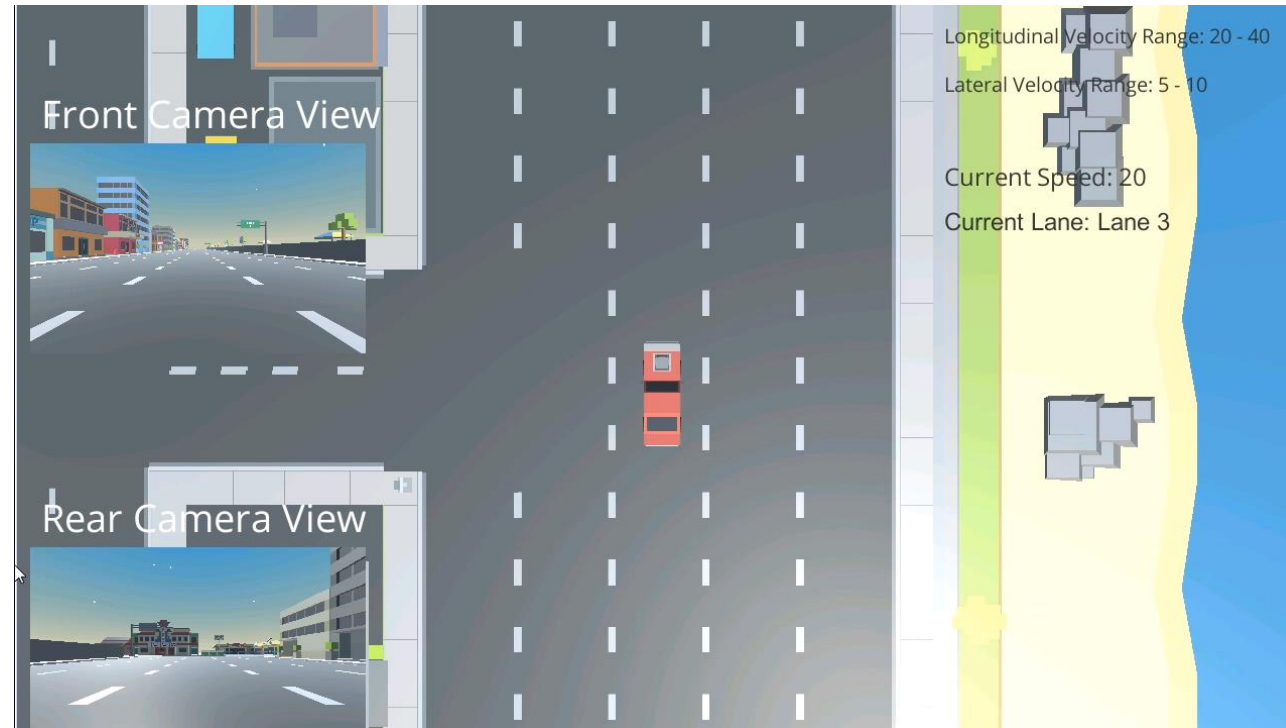
- Python과 Unity의 통신



ML-Agents를 이용한 개발 사례 소개

Machine Learning Camp Jeju 2017

- 결과 영상



ML-Agents를 이용한 개발 사례 소개

Machine Learning Camp Jeju 2017

- Unity와 Python의 통신
 - Socket 통신을 이용하여 구현했지만 불안한 부분 및 버그 다수 존재
 - 통신이 일정 이상 진행되면 끊기는 문제 발생
 - 환경 상의 작은 변화를 위해 많은 코딩이 필요
 - 통신 간의 동기화 문제
 - Unity와 Python 코드 간 실행 속도 차이 문제
 - 약 1~2 개월 동안 문제들을 해결하려 노력
 - 전체 문제의 약 70% 정도 해결
 - Github에 배포 예정

ML-Agents를 이용한 개발 사례 소개

Machine Learning Camp Jeju 2017

- Unity ML-Agents 출시 (2017.09.19)



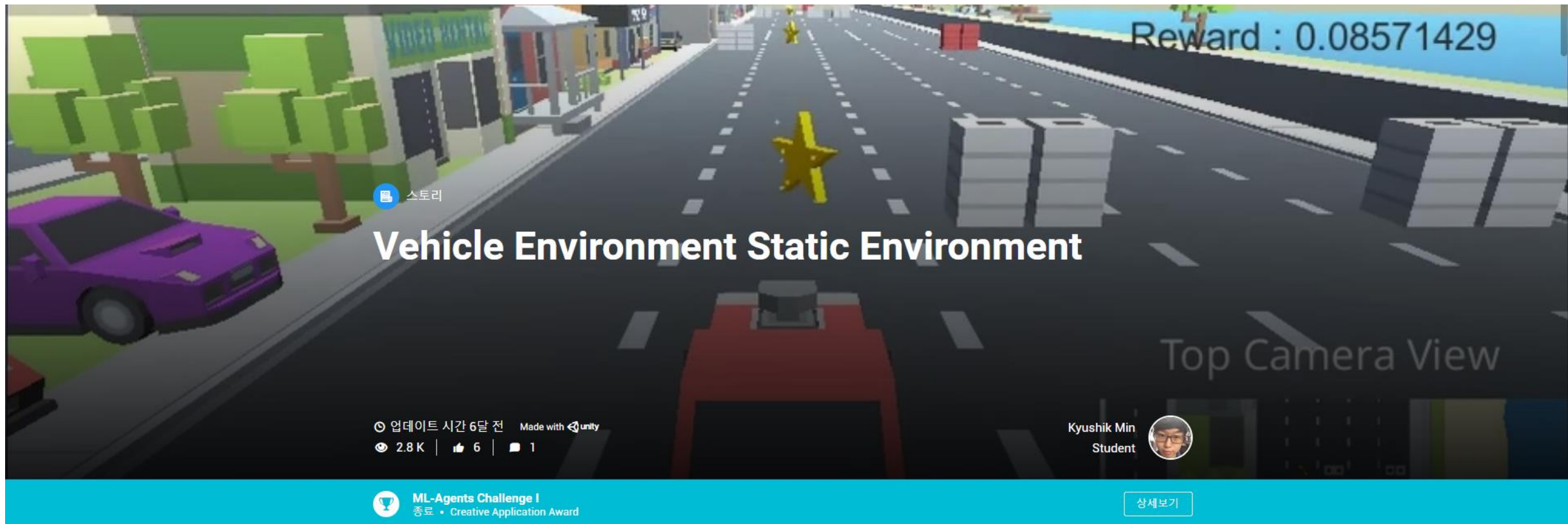
ML-Agents를 이용한 개발 사례 소개

Unity ML-Agents Challenge



ML-Agents를 이용한 개발 사례 소개

Unity ML-Agents Challenge



ML-Agents를 이용한 개발 사례 소개

Unity ML-Agents Challenge



ML-Agents를 이용한 개발 사례 소개

Unity ML-Agents Challenge



ML-Agents를 이용한 개발 사례 소개

2018 IEEE IV Conference



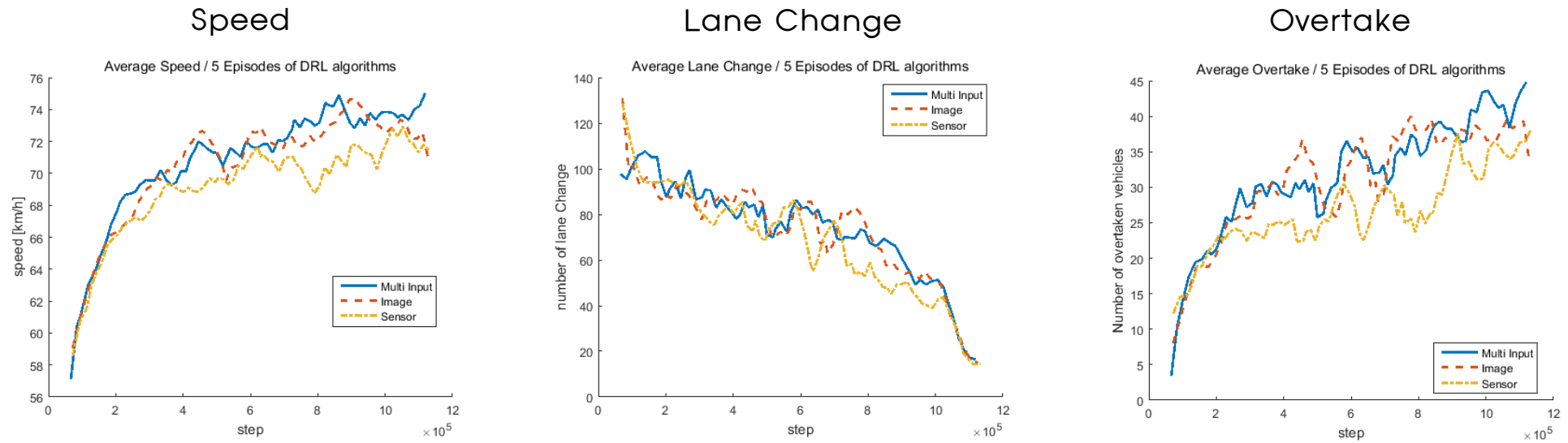
Vehicle Simulator



Unity ML-agents

ML-Agents를 이용한 개발 사례 소개

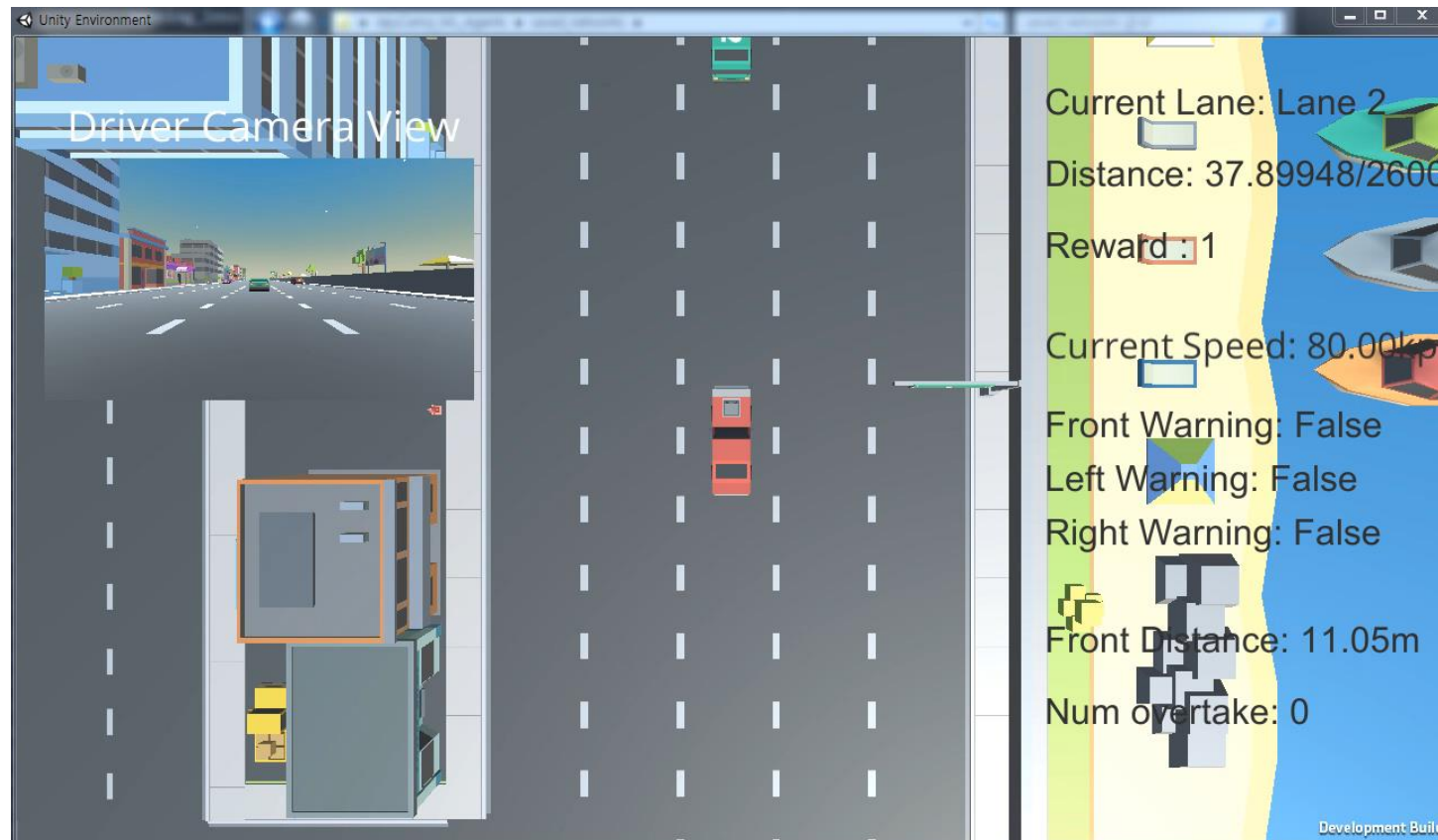
2018 IEEE IV Conference



Input Configuration	Average Speed (km/h)	# of Average Lane Change	# of Average Overtaking
Camera Only	71,0776	15	35,2667
LIDAR Only	71,3758	14,2667	38,0667
Multi-Input	75,0212	19,4	44.8

ML-Agents를 이용한 개발 사례 소개

2018 IEEE IV Conference



ML-Agents를 이용한 개발 사례 소개

2018 IEEE IV Conference

2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Deep Q Learning Based High Level Driving Policy Determination

Kyushik Min, Hayoung Kim and Kunsoo Huh, *Member, IEEE*

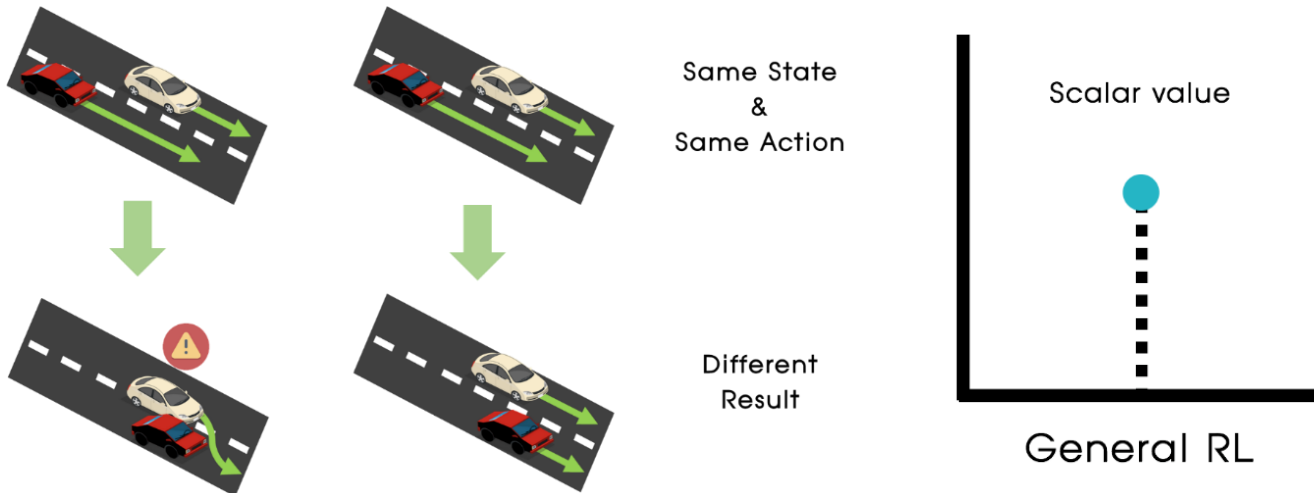
<https://ieeexplore.ieee.org/abstract/document/8500645>

ML-Agents를 이용한 개발 사례 소개

IEEE T-IV

■ IEEE T-IV

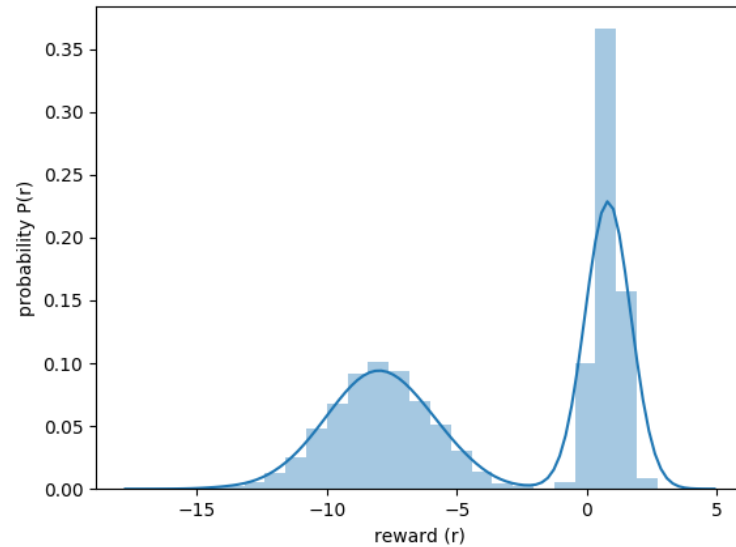
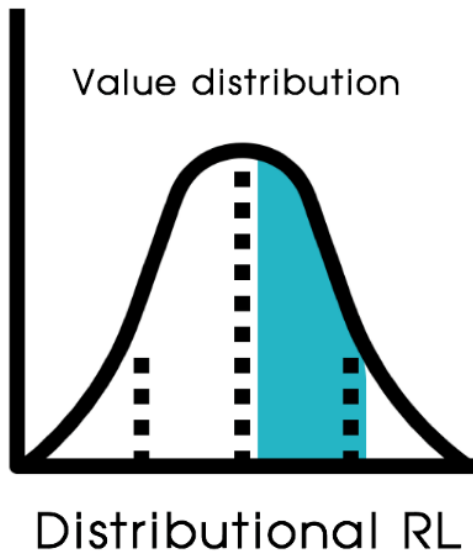
- 운전 상황은 확률적인 환경
- 동일한 환경에서 동일한 행동을 해도 결과가 다를 수 있다!
- 일반적인 강화학습: 미래에 받을 평균 보상의 합을 하나의 값으로 예측



ML-Agents를 이용한 개발 사례 소개

IEEE T-IV

- Distributional RL
 - 미래에 받을 평균 보상의 합을 확률 분포로 예측
 - 확률적인 환경에서 더 좋은 성능



ML-Agents를 이용한 개발 사례 소개

IEEE T-IV

■ Network Architecture

- Distributional RL 알고리즘 중 하나인 QR-DQN 사용

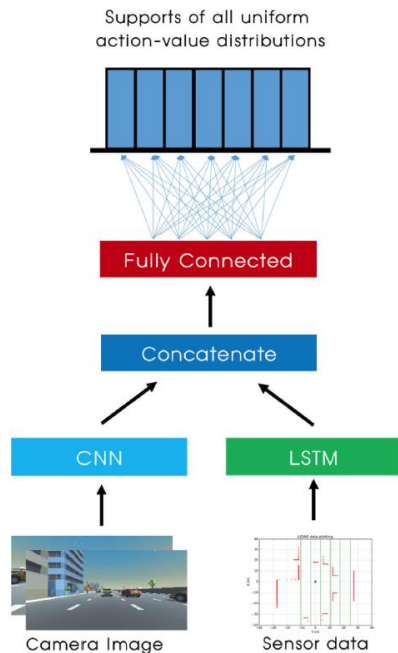


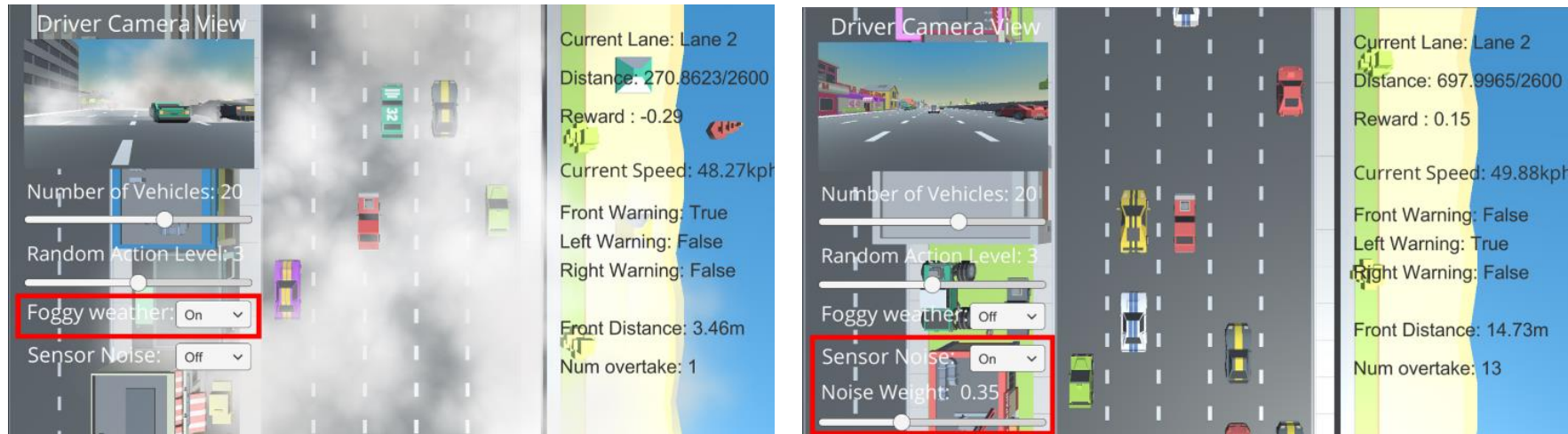
TABLE II
HYPERPARAMETERS OF DRIVING POLICY NETWORK

Data	Type	Actuation	Hyperparameters
Camera data	Convolution	ReLU	patch size = (8x8) stride = 4 # of filters = 32
	Convolution	ReLU	patch size = (4x4) stride = 2 # of filters = 64
	Convolution	ReLU	patch size = (3x3) stride = 1 # of filters = 64
Sensor data	LSTM	-	time steps = 4 # of cell states = 256
Concatenated data	Fully Connected	ReLU	# of units = 512

ML-Agents를 이용한 개발 사례 소개

IEEE T-IV

- Simulation 환경
 - Distributional RL의 강인성 검증을 위하여 안개와 센서 노이즈 추가
 - 센서 노이즈 식: $d = d + \alpha * \text{Random.Range}(-d, d)$ (α : noise weight)
 - 학습 후 알고리즘 검증을 수행할 때만 센서 노이즈와 안개 추가

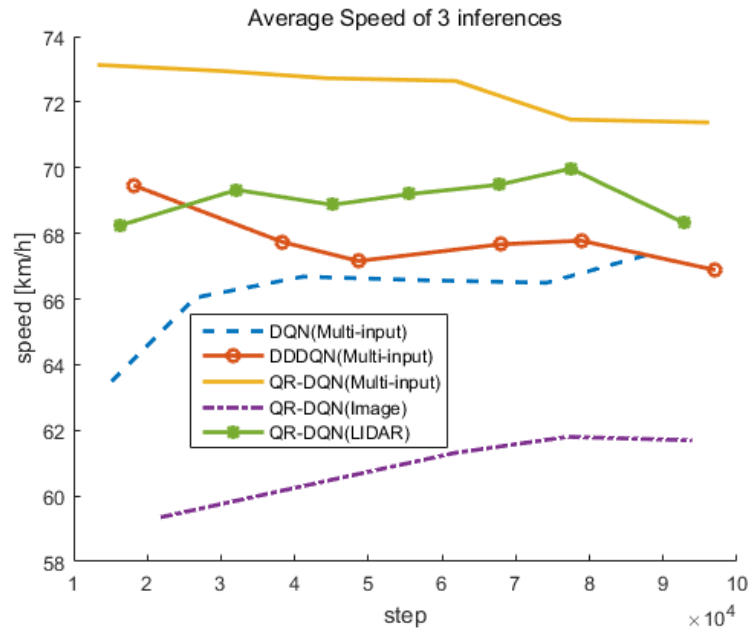
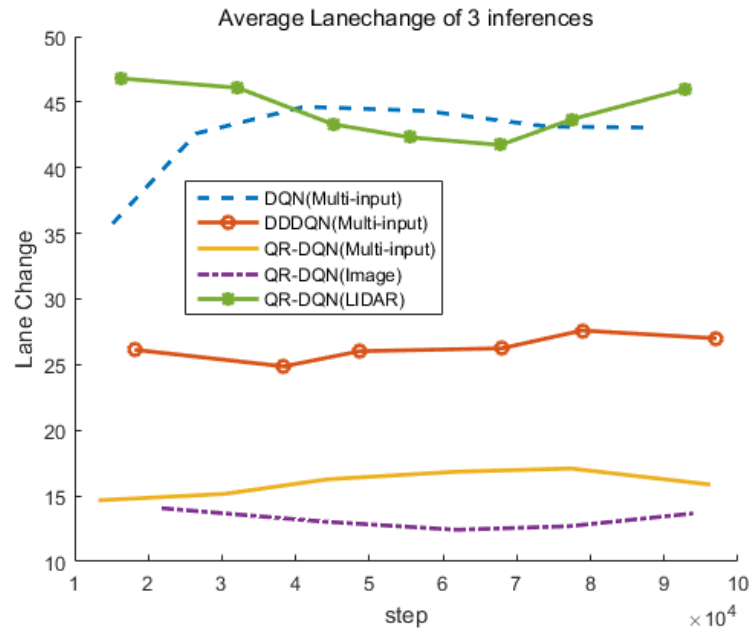


ML-Agents를 이용한 개발 사례 소개

IEEE T-IV

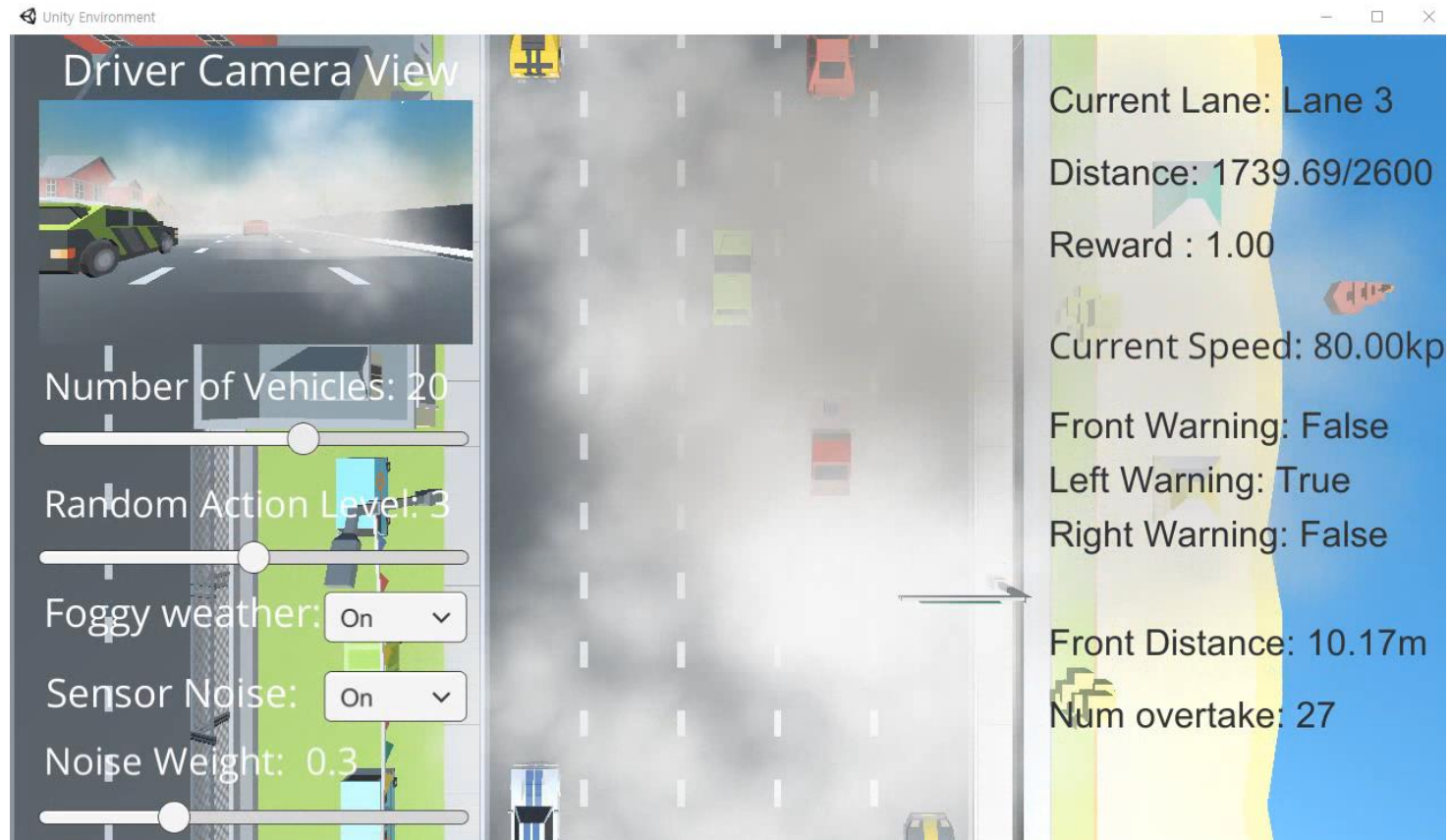
■ Result

- 빠른 평균 속도, 불필요한 차선 변경 최소로!



ML-Agents를 이용한 개발 사례 소개

IEEE T-IV






ML-Agents를 이용한 개발 사례 소개

IEEE T-IV

416

IEEE TRANSACTIONS ON INTELLIGENT VEHICLES, VOL. 4, NO. 3, SEPTEMBER 2019

Deep Distributional Reinforcement Learning Based High-Level Driving Policy Determination

Kyushik Min , Hayoung Kim , and Kunsoo Huh , *Member, IEEE*

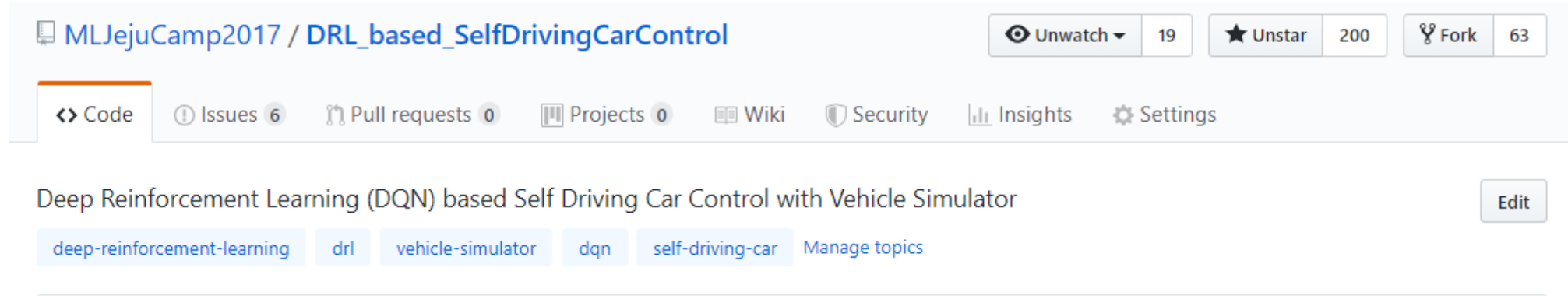
<https://ieeexplore.ieee.org/abstract/document/8723635>

ML-Agents를 이용한 개발 사례 소개

IEEE T-IV

■ Github

- 다음의 항목들을 Github에 업로드!
 - RL 알고리즘
 - 빌드된 유니티 환경
 - 유니티 파일



The screenshot shows the GitHub repository page for MLJejuCamp2017 / DRL_based_SelfDrivingCarControl. The repository name is displayed at the top left, followed by navigation buttons for Unwatch (19), Unstar (200), and Fork (63). Below the repository name is a navigation bar with tabs for Code, Issues (6), Pull requests (0), Projects (0), Wiki, Security, Insights, and Settings. The repository description is "Deep Reinforcement Learning (DQN) based Self Driving Car Control with Vehicle Simulator", and there is an Edit button. Below the description are topic tags: deep-reinforcement-learning, drl, vehicle-simulator, dqn, self-driving-car, and a Manage topics link.

https://github.com/MLJejuCamp2017/DRL_based_SelfDrivingCarControl

ML-Agents를 이용한 개발 사례 소개

2019 IEEE ISPACS

- Multi-Agent Traffic Control Environment
 - 복잡한 도로 상황에서 원하는 차선으로 차선 변경하는 것이 어려움
 - 차선 변경으로 인해 전체적인 교통의 흐름이 느려질 수 있음
 - 복잡한 상황에서 무리한 차선 변경은 사고를 유발할 수 있음
- Goal
 - 다수의 차량을 동시에 제어!
 - 특정 차량을 타겟 차선으로 이동시키는 문제!
 - 전체적인 차량 속도 감소는 최소로!
=> Multi-Agent Reinforcement Learning

ML-Agents를 이용한 개발 사례 소개

2019 IEEE ISPACS

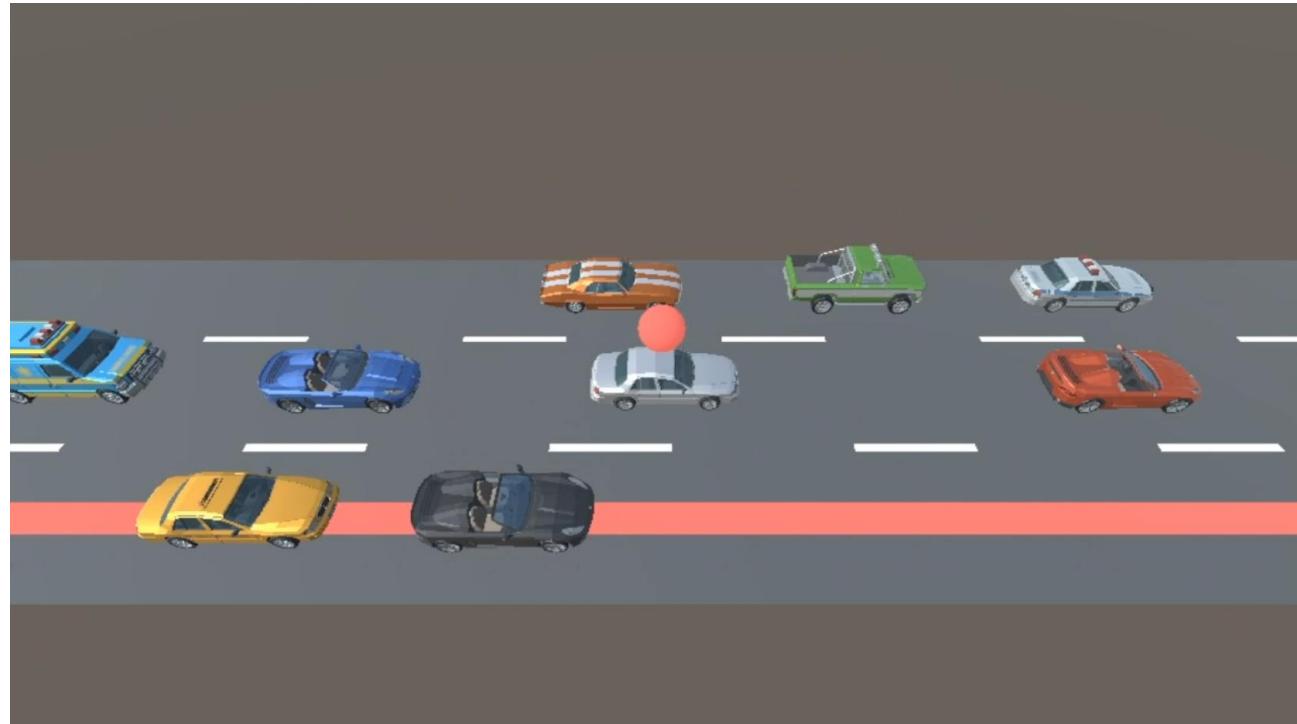
- Zombie Defense 환경



ML-Agents를 이용한 개발 사례 소개

2019 IEEE ISPACS

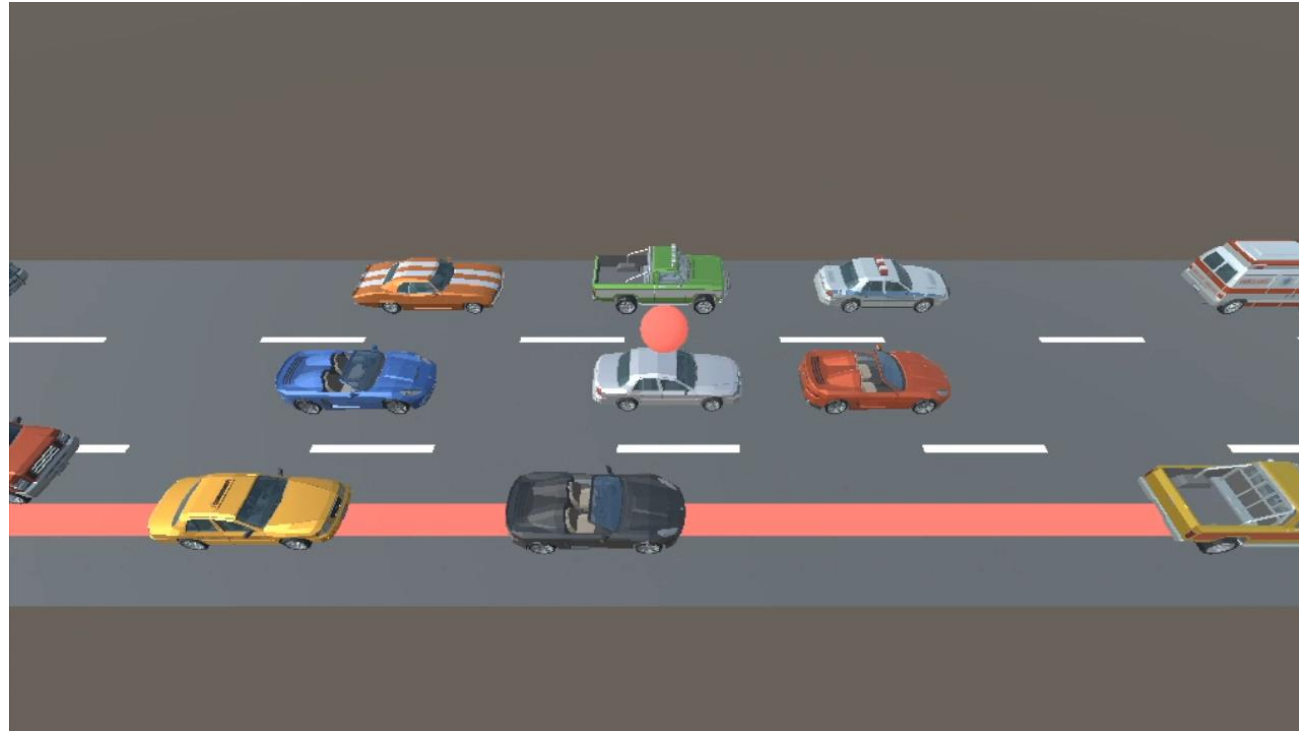
- Simulator



ML-Agents를 이용한 개발 사례 소개

2019 IEEE ISPACS

- Result



ML-Agents를 이용한 개발 사례 소개

2019 IEEE ISPACS

- Result

Multi-Agent Deep Reinforcement Learning for Cooperative Driving in Crowded Traffic Scenarios

Jongwon Park
Hanyang University
Seoul, Republic of Korea
pjlw2091@hanyang.ac.kr

Kyushik Min
Hanyang University
Seoul, Republic of Korea
mks0813@hanyang.ac.kr

Kunsoo Huh
Hanyang University
Seoul, Republic of Korea
khuh2@hanyang.ac.kr

<https://ieeexplore.ieee.org/abstract/document/8986374>

ML-Agents를 이용한 개발 사례 소개

2019 IEEE ISPACS

- 기차 환경



Conclusion

Conclusion

- Unity ML-Agents를 사용한 강화학습 환경 제작!
 - 유니티를 이용한 손쉬운 환경 제작
 - 유니티 환경과 Python 코드 간의 안정적인 통신
 - 다양한 종류의 강화학습 환경 제작 지원 (Multi-Agent, Curriculum, Self-play, Sparse-reward, ...)
 - 제공된 강화학습 알고리즘을 통해 에이전트 학습 가능 -> 학습된 에이전트를 내장한 환경 빌드 가능
 - 추후 ML-Agents cloud, DOTS 적용, Nvidia PhysX 적용 등 다양한 서비스 제공 및 개선이 이루어질 예정
- ML-Agents를 이용한 다양한 연구 수행
 - 원하는 조건에 맞는 다양한 게임 및 시뮬레이션 환경 제작 가능
 - ML-Agents를 이용하여 강화학습 수행 및 성능 검증
 - ML-Agents를 사용하여 구현한 환경에서 수행한 연구에 대해 논문 발표 및 준비 중

Reference

Reference

- Unity ML-Agents 공식 깃허브
 - <https://github.com/Unity-Technologies/ml-agents>
- Blog: Announcing ML-Agents Unity Package v1.0
 - <https://blogs.unity3d.com/2020/05/12/announcing-ml-agents-unity-package-v1-0/>
- RLKorea Unity ML-Agents Team Github
 - https://github.com/reinforcement-learning-kr/Unity_ML_Agents