



---

# Democratizing Deep Learning with Unity ML-Agents

Arthur Juliani

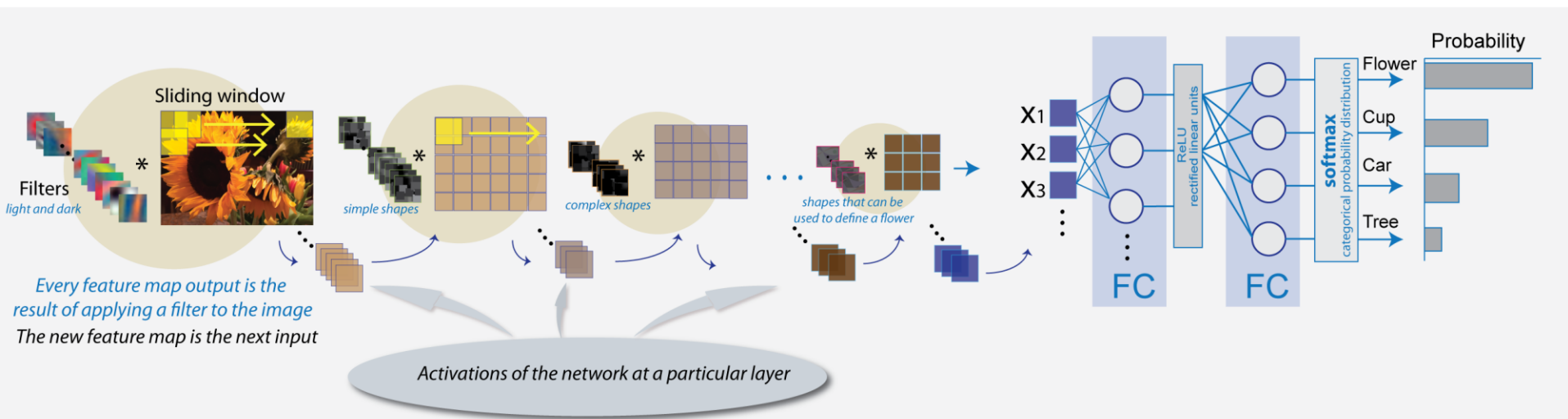
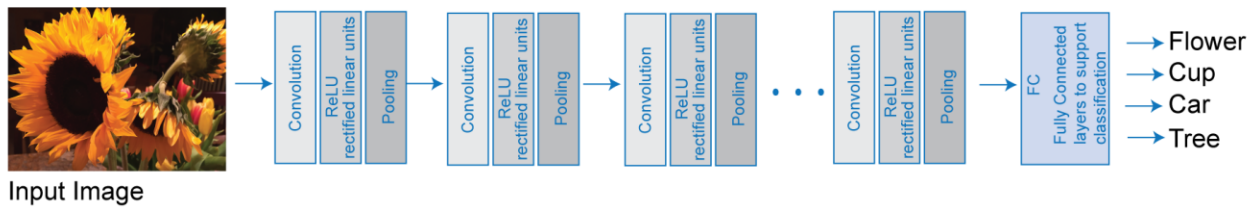
# About Unity

“Creation Engine”

- Games
- AR/VR
- Cinematics
- Simulations
- 40+ Platforms
- Free for personal use



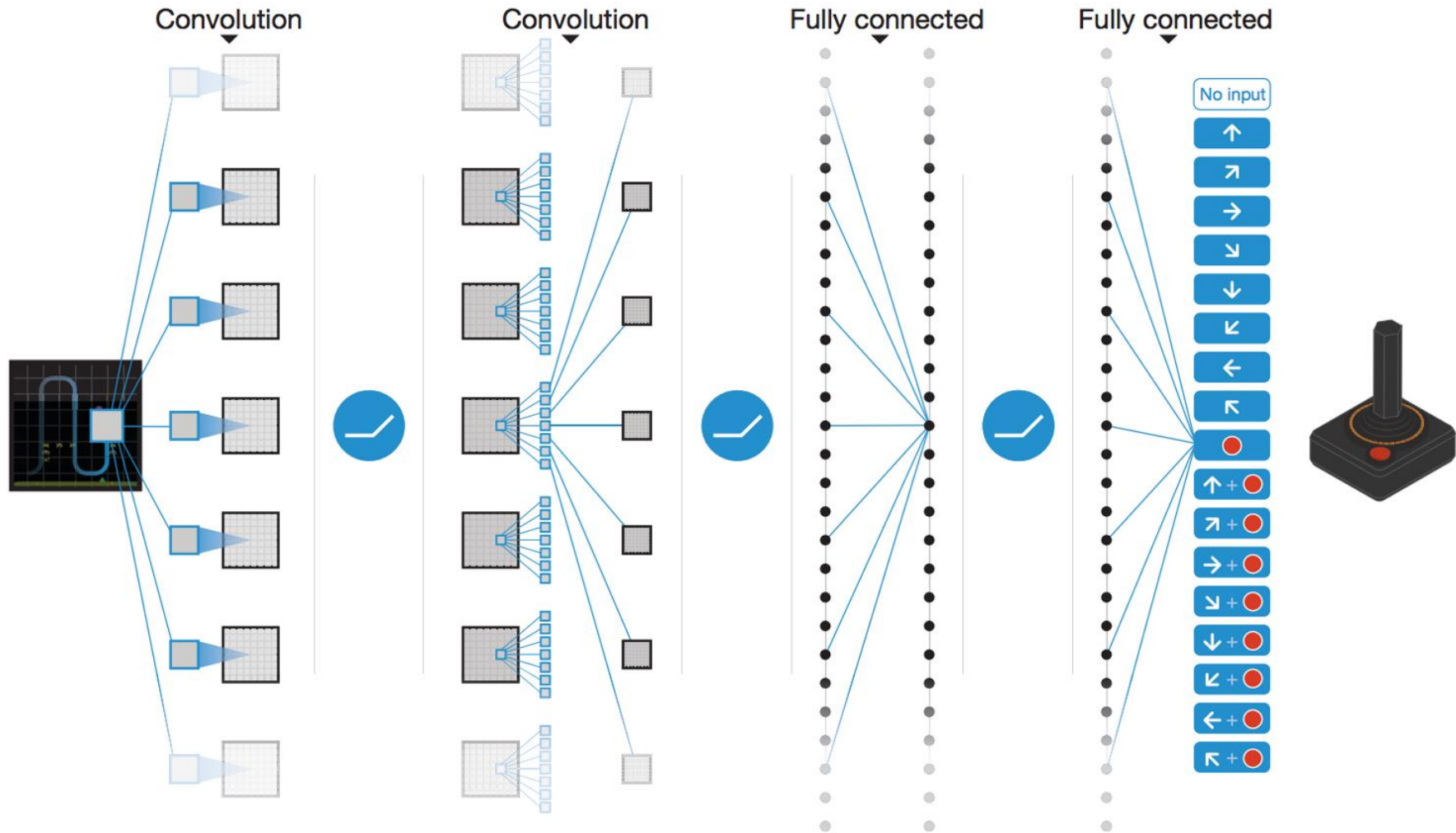
# Machine Learning



---

# Machine Learning And Games?







HUMAN

00:00.2

MACHINE





# Machine Learning Agents



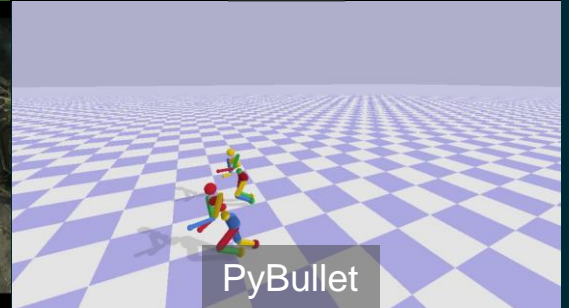
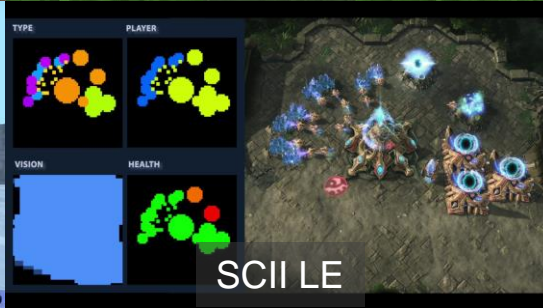
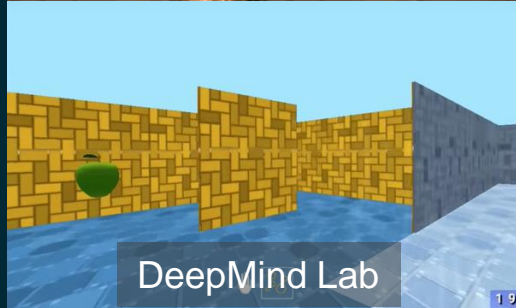
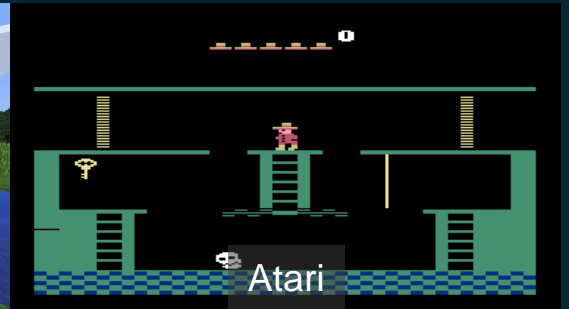
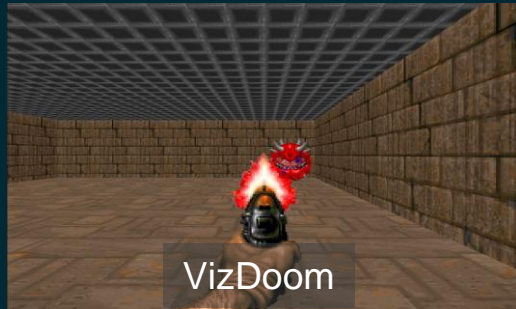
---

**Yet another ML training platform?**





# ML Training Platforms



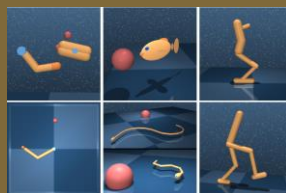
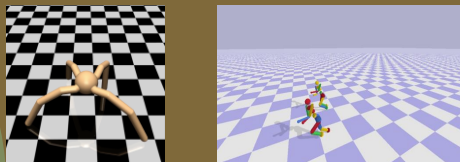
---

# ML Training Environments



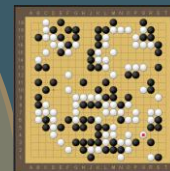
---

Visual Complexity



---

Physical Complexity



---

Cognitive Complexity

# The Unity Ecosystem



---

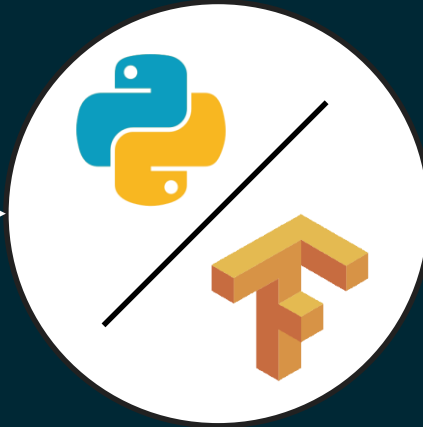
# How does it work?



# Unity ML-Agents Workflow



Create Environment



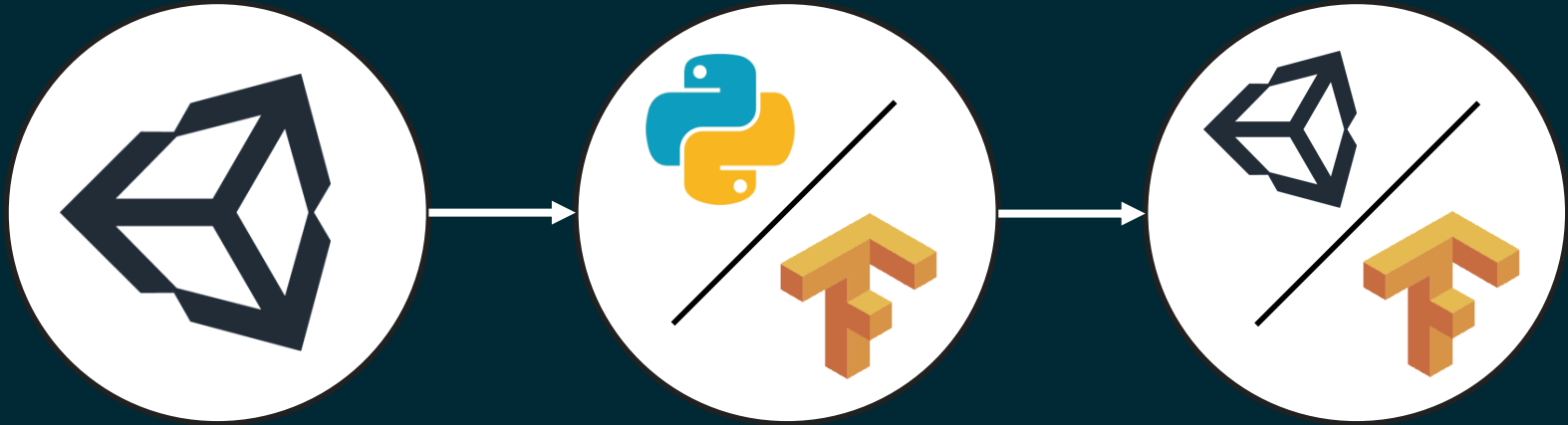
Train Agents



Embed Agents



# Unity ML-Agents Workflow



**Create Environment**

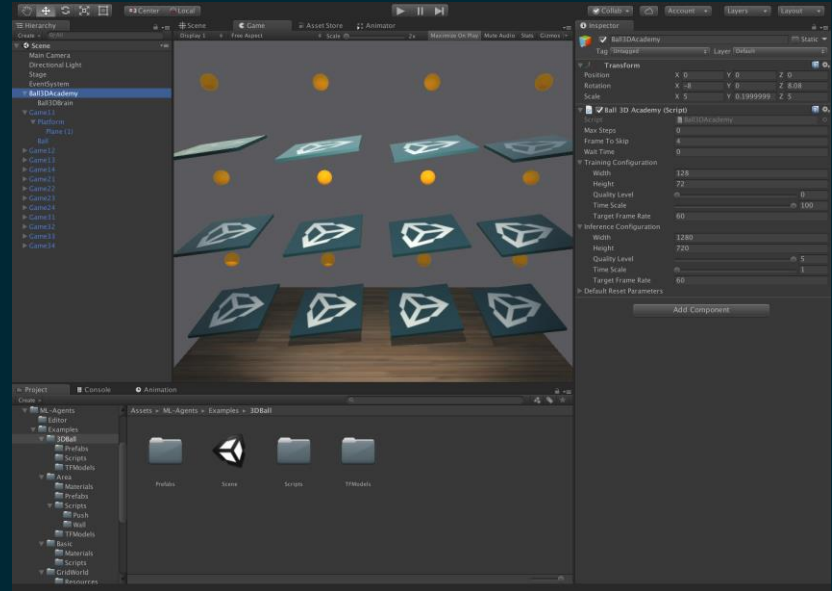
**Train Agents**

**Embed Agents**

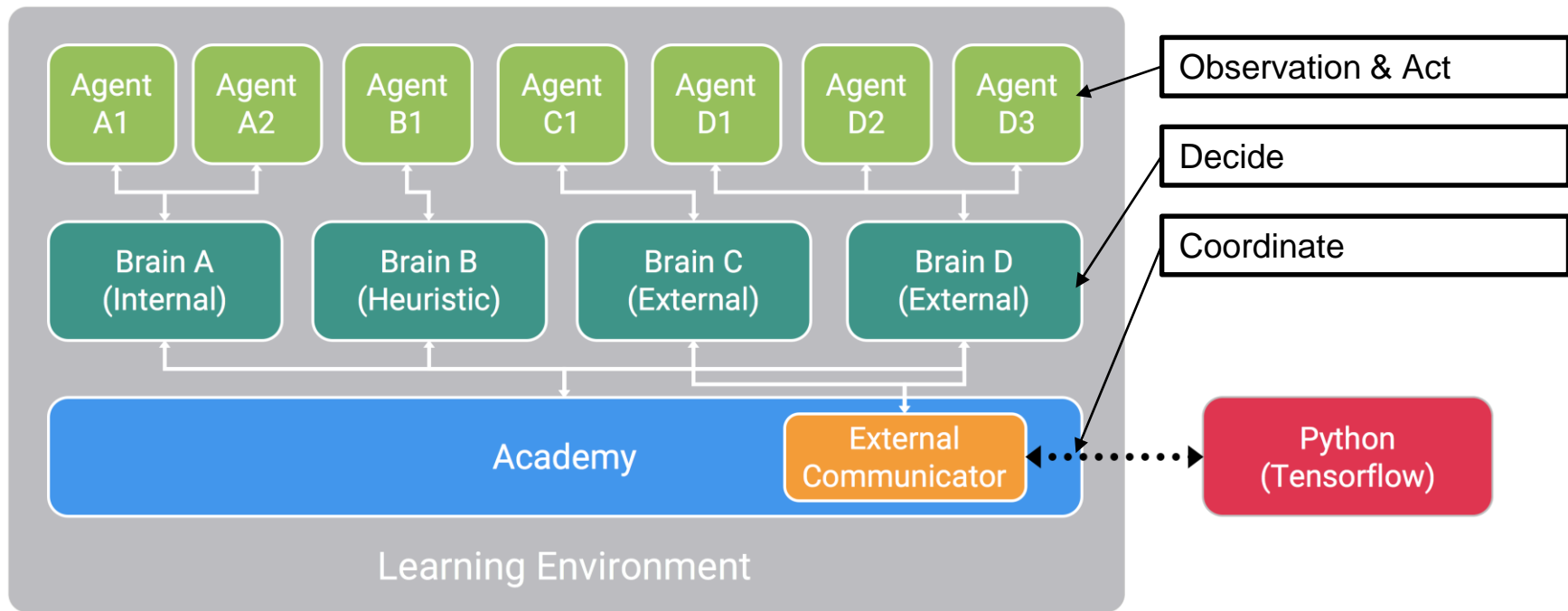


# Create Environment (Unity)

1. Create Scene
2. Add Academy, Brain(s), and Agent(s)
3. Define Observations, Actions, and Rewards
4. Build Executable



# Create Environment (Unity)



---

# Agents

- Agents are GameObjects within the Unity scene.
- They perceive the environment via ***observations***, take ***actions***, and optionally receive ***rewards***.
- Each agent is linked to a brain, which makes decisions for the agent.

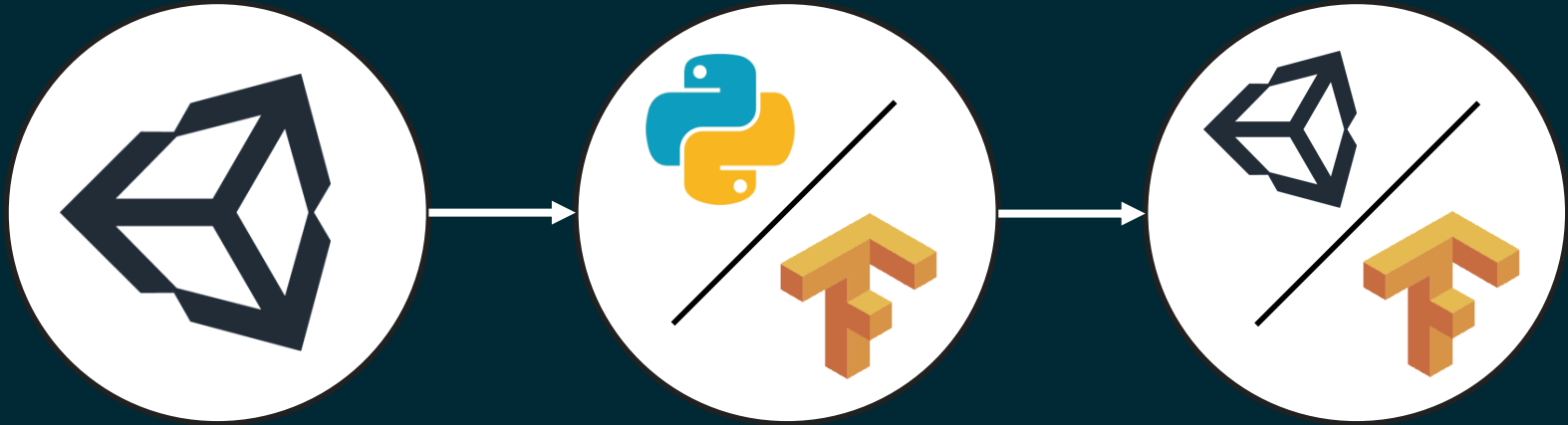
---

# Brains

- **Player** – Actions are decided by user input through keyboard or gamepad.
- **Heuristic** – Actions are decided by C# script using state input.
- **External** – Actions are decided using Tensorflow via Python interface.
- **Internal** – Actions are decided using Tensorflow model embedded into project.



# Unity ML-Agents Workflow



Build Environment

Train Agents

Embed Agents



# Training Methods

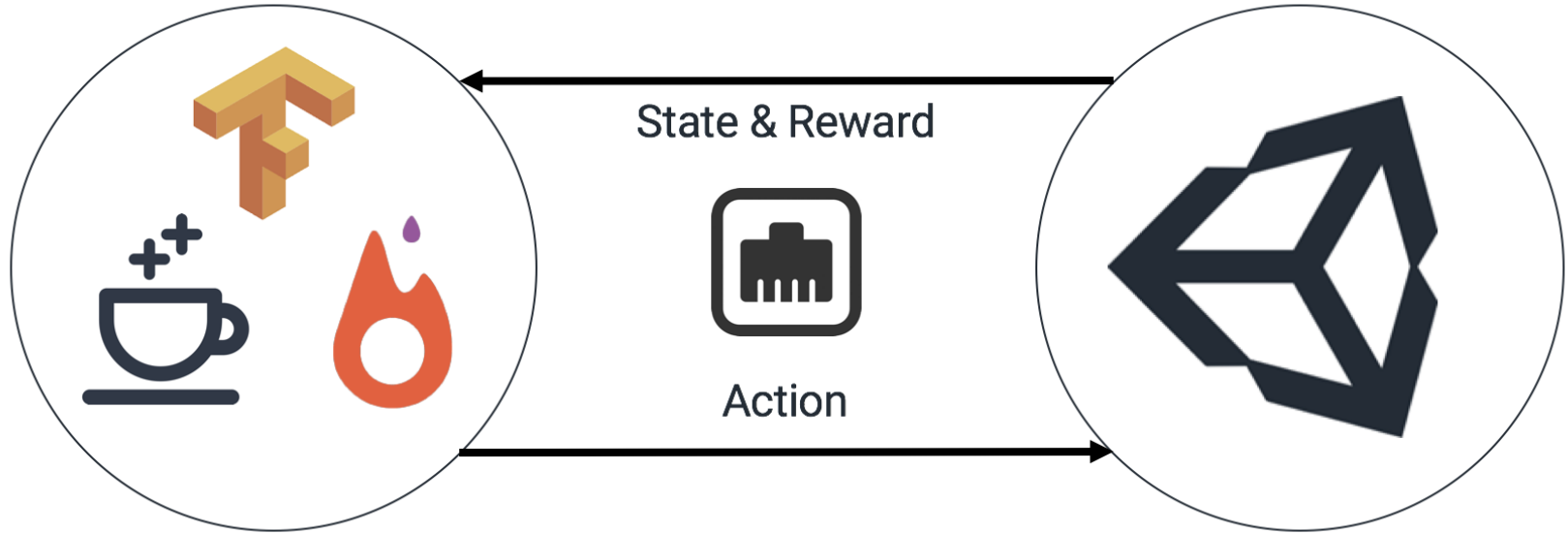
## Reinforcement Learning

- Learn through rewards
- Trial-and-error
- Super-speed simulation
- Agent becomes “optimal” at task

## Imitation Learning

- Learn through demonstrations
- No rewards necessary
- Real-time interaction
- Agent becomes “human-like” at task

# Agent Training Process



# Train Agents (Python)

- Launch an environment from python with  
`env = UnityEnvironment("my_environment")`
- Interact with gym-like interface:

```
env.reset()
```

```
env.step()
```

```
env.close()
```

```
1 env = UnityEnvironment(file_name=env_name)
2
3 # Examine environment parameters
4 print(str(env))
5
6 # Set the default brain to work with
7 default_brain = env.brain_names[0]
8 brain = env.brains[default_brain]
```

```
INFO:unityagents.environment:
'Ball3DAcademy' started successfully!
```

```
Unity Academy name: Ball3DAcademy
Number of brains: 1
Reset Parameters :
```

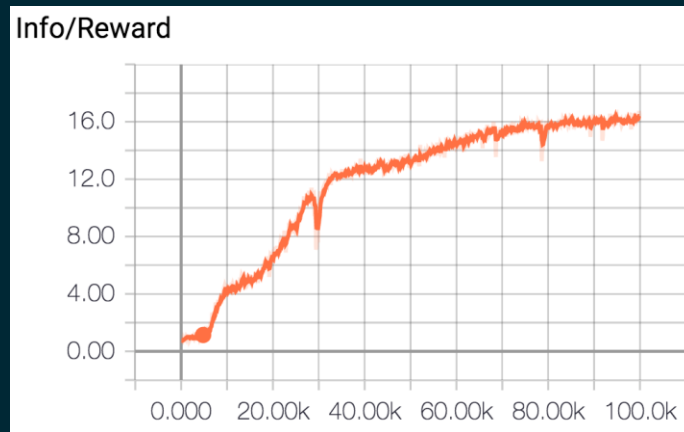
```
Unity brain name: Ball3DBrain
Number of observations (per agent): 0
State space type: continuous
State space size (per agent): 8
Action space type: continuous
Action space size (per agent): 2
Memory space size (per agent): 0
```



---

# Train Agents (Python)

- PPO and Behavioral Cloning algorithms included by default.
- Works with Continuous and Discrete Control, plus image and/or vector inputs.
- Monitor progress with TensorBoard.

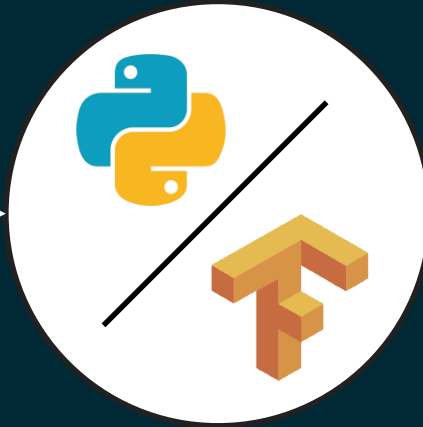




# Unity ML-Agents Workflow



Build Environment



Train Agents

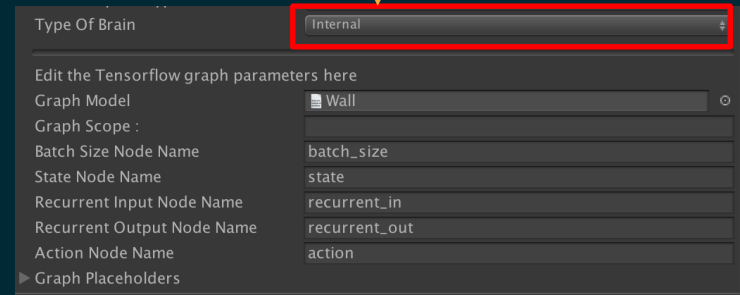
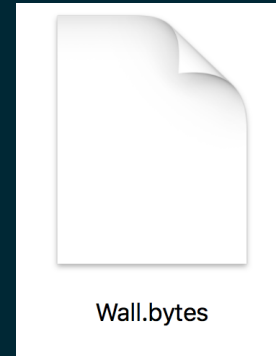


Embed Agents



# Embed Agents (Unity)

- Once a model is trained, it can be exported into the Unity project.
- Simply drop *.bytes* file into Unity project, and use it in corresponding Brain with “Internal” mode.
- Support for Mac, Windows, Linux, iOS, and Android.



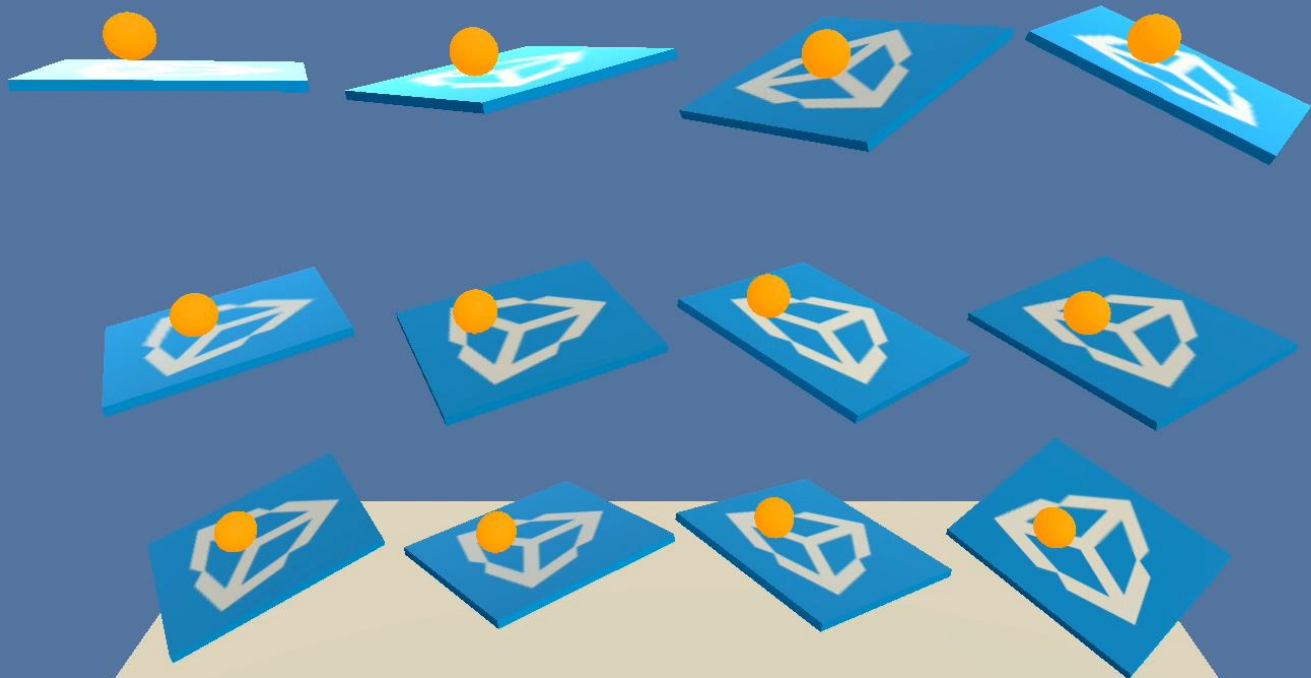
---

# Learning Scenarios



---

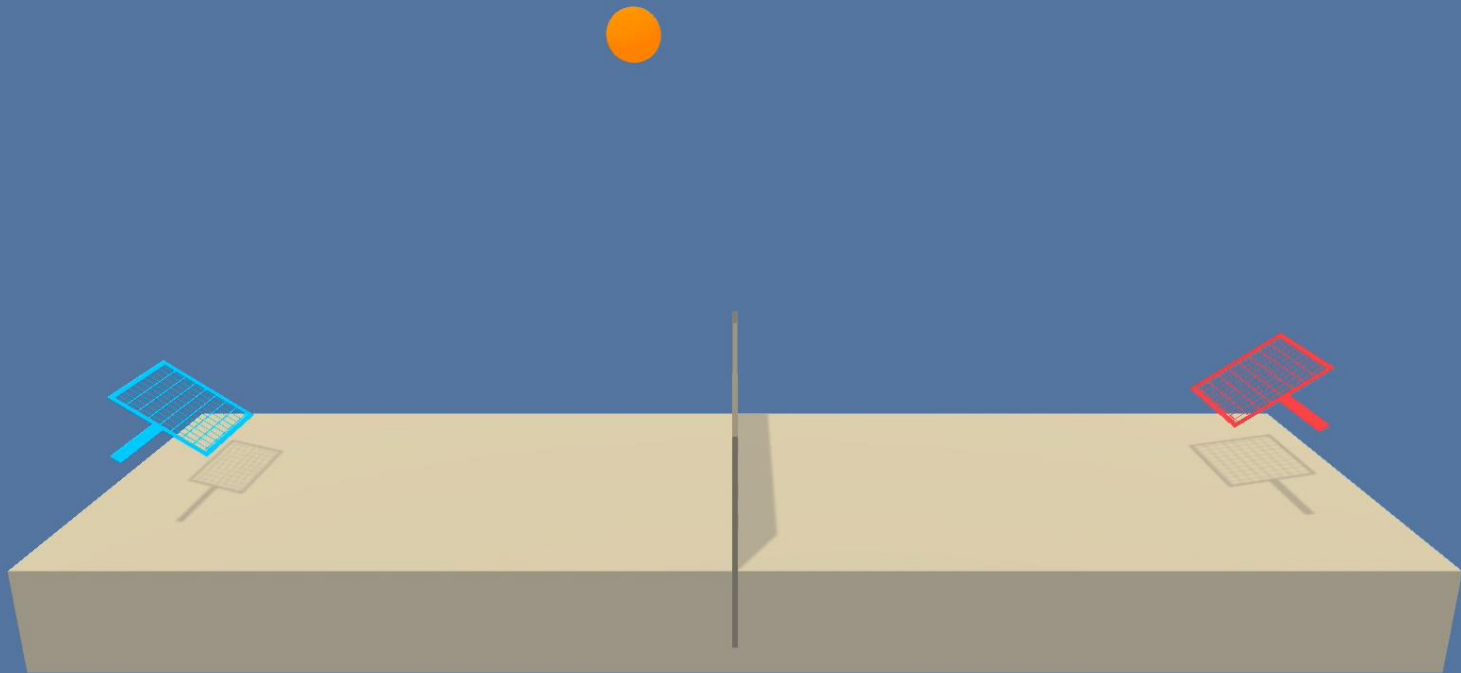
# Twelve Agents, One Brain, Independent Rewards





---

# Two Agents, One Brain, Cooperative Rewards



---

# Four Agents, Two Brains: Competitive Multi-Agent

---

# Multi-Stage Soccer Training

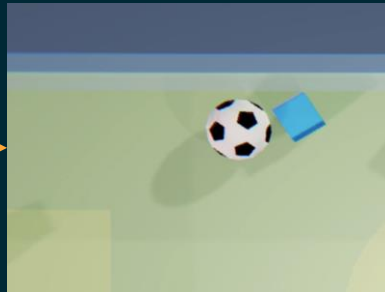
## Defense

Train one brain with negative reward for ball entering their goal



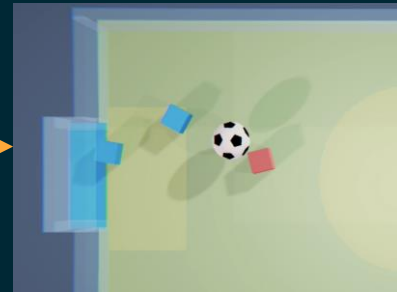
## Offense

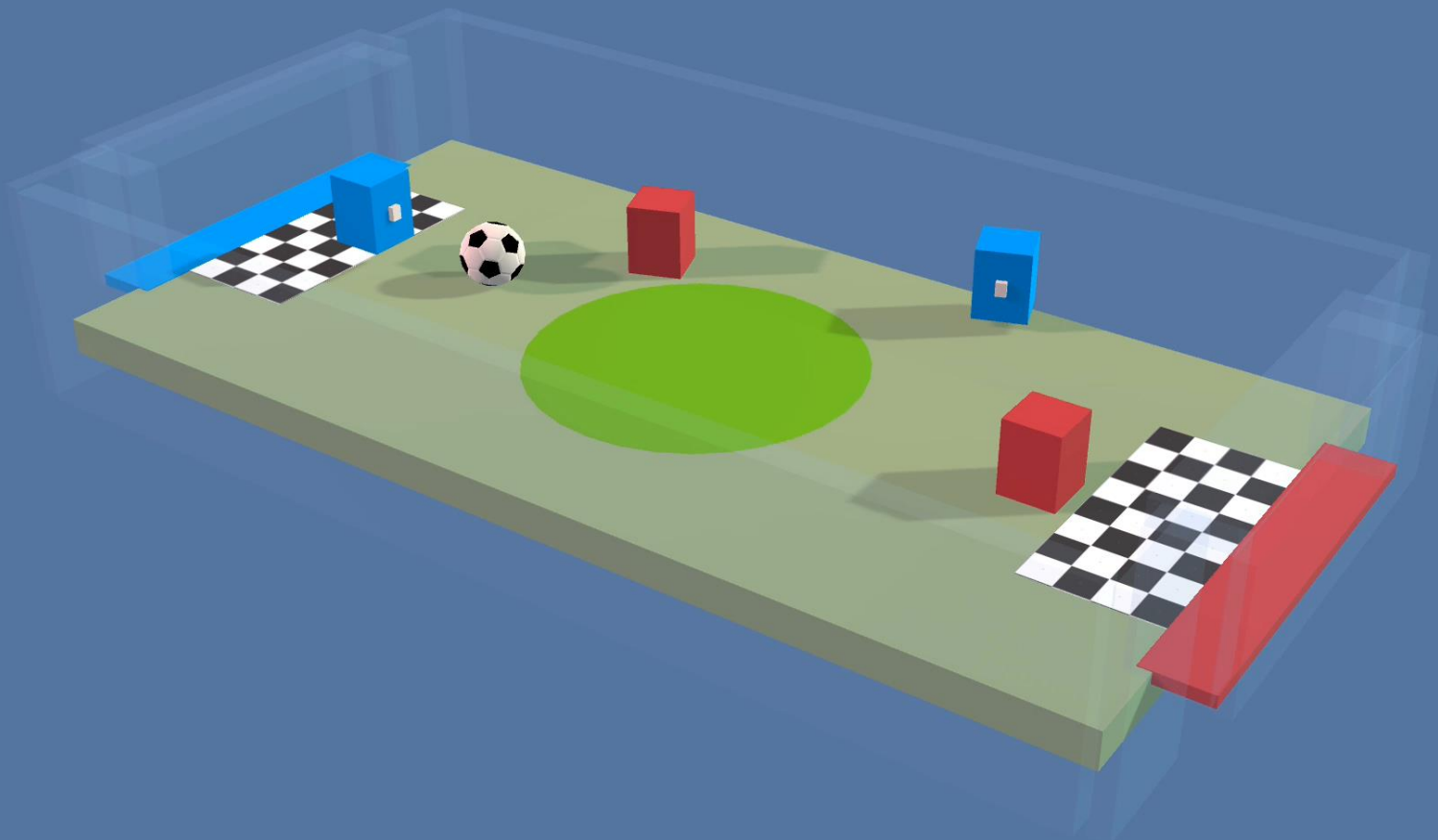
Train one brain with positive reward for ball entering opponents goal



## Combined

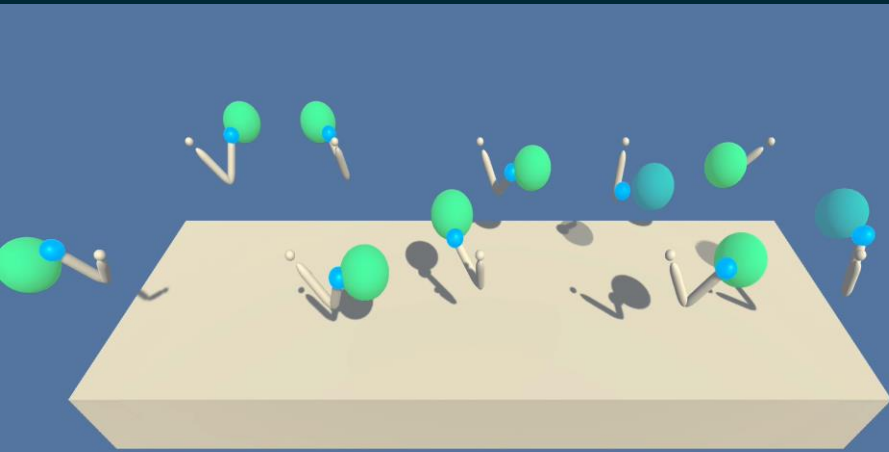
Train both brains together to play against opponent team



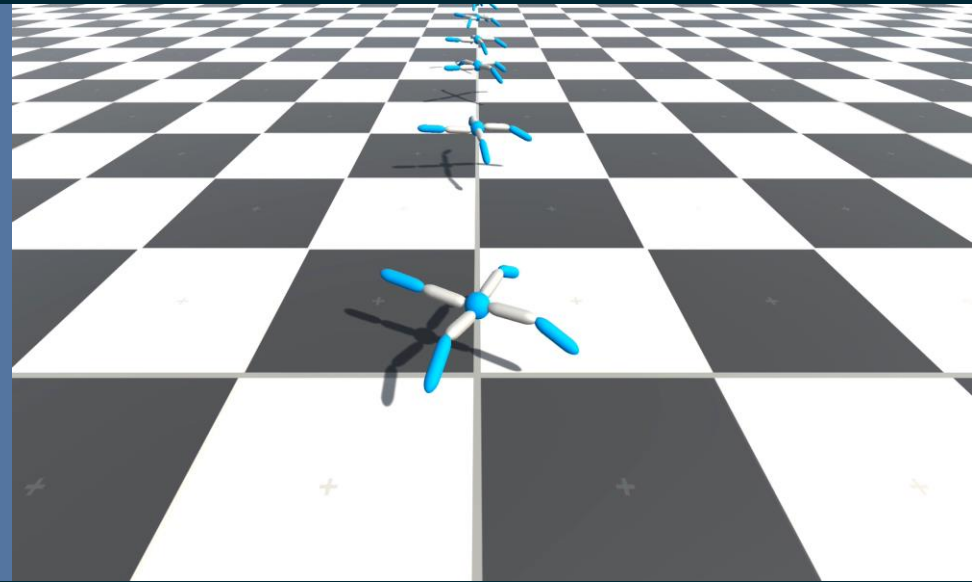


---

# Physical Manipulation



Reacher



Crawler

---

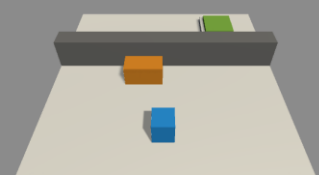
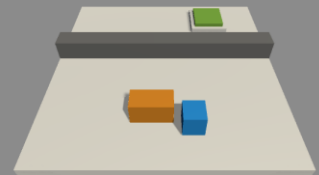
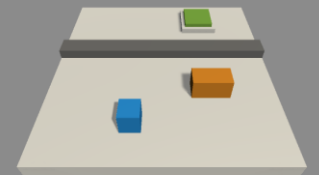
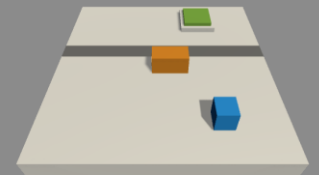
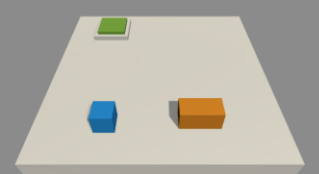
# Additional Features





---

# Curriculum Learning



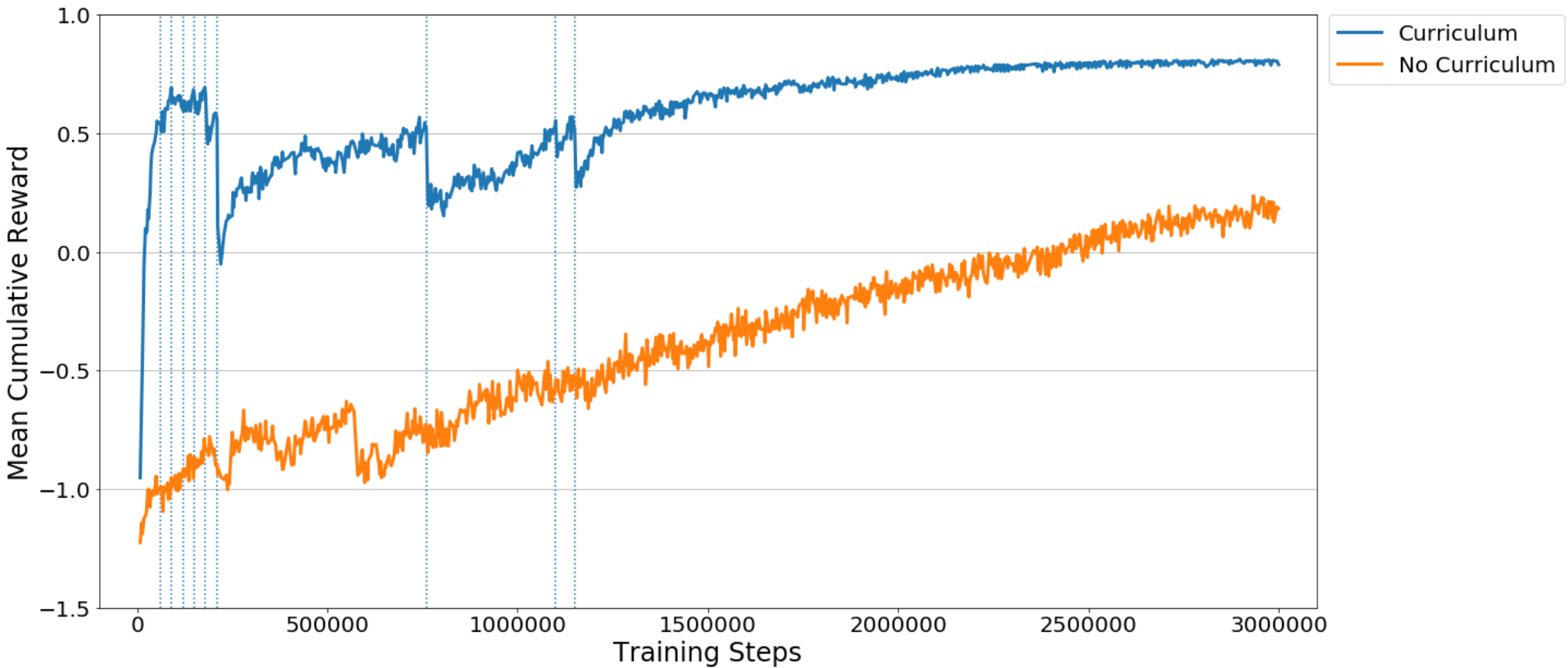
Easy



Difficult

## Curriculum Learning

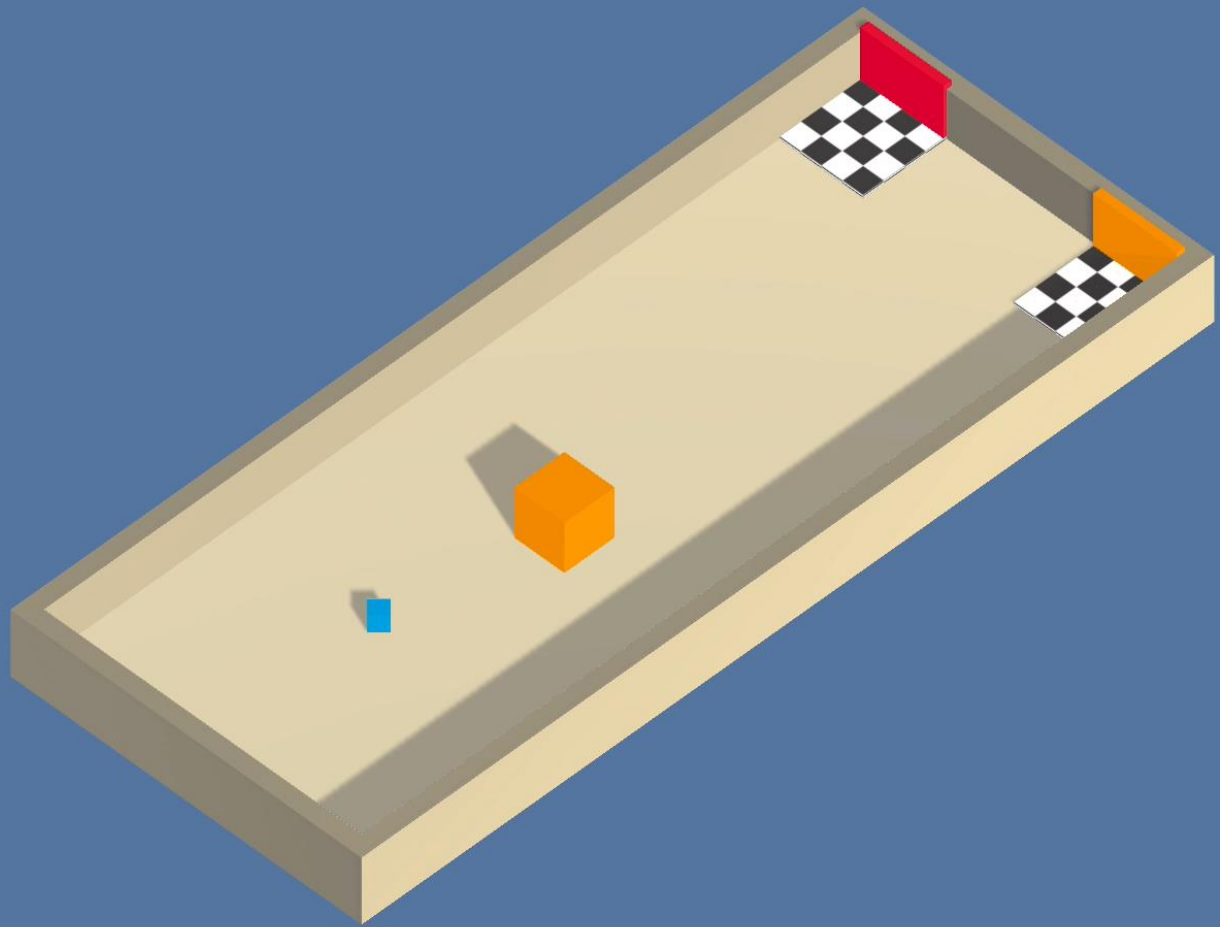
- Bootstrap learning of difficult task with simpler task
- Utilize custom reset parameters
- Change environment task based on reward or fixed progress

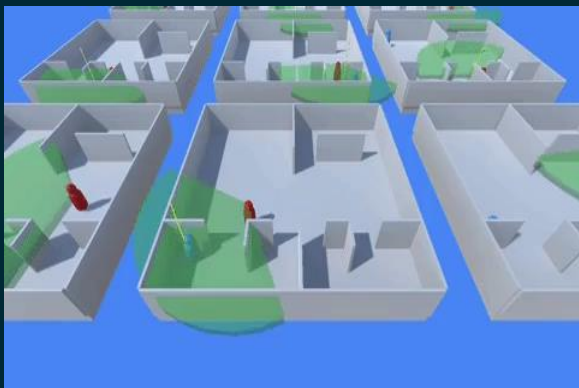
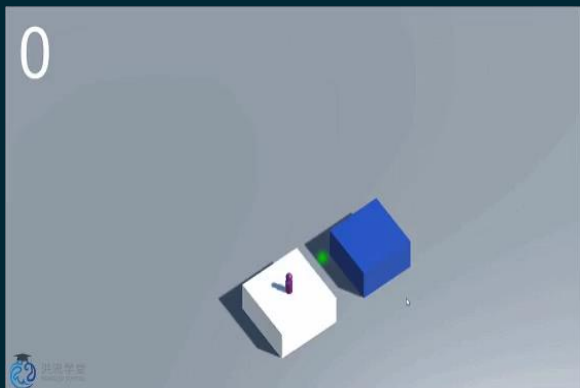
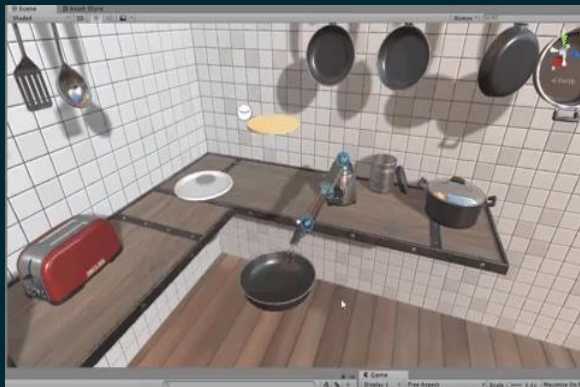




V0.3 New Feature

# Memory-enhanced agents





# Try it Now

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

Unity-Technologies / ml-agents

Unwatch

230

Unstar

1,831

Fork

328

Code

Issues 60

Pull requests 7

Projects 0

Insights

Settings

Unity Machine Learning Agents

Edit

reinforcement-learning

unity3d

deep-learning

unity

deep-reinforcement-learning

neural-networks

Manage topics

105 commits

6 branches

4 releases

18 contributors

Apache-2.0







---

# Thank You!

# Questions?



[arthurj@unity3d.com](mailto:arthurj@unity3d.com)