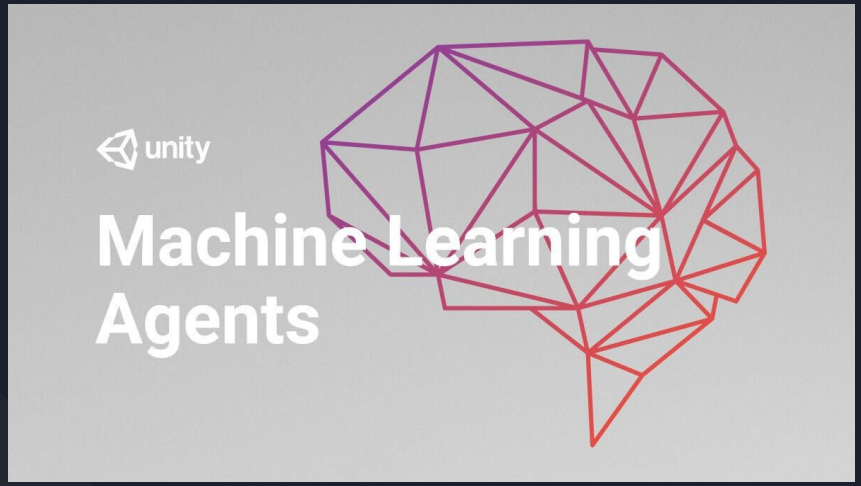
Unity ML Agents

The Unity Toolkit used to train agents



By Lea Wölf, Xabier Arriaga & Jon Ander Ruiz

Advanced Techniques in Artificial Intelligence

Donostia, 09-22-2021

Contents

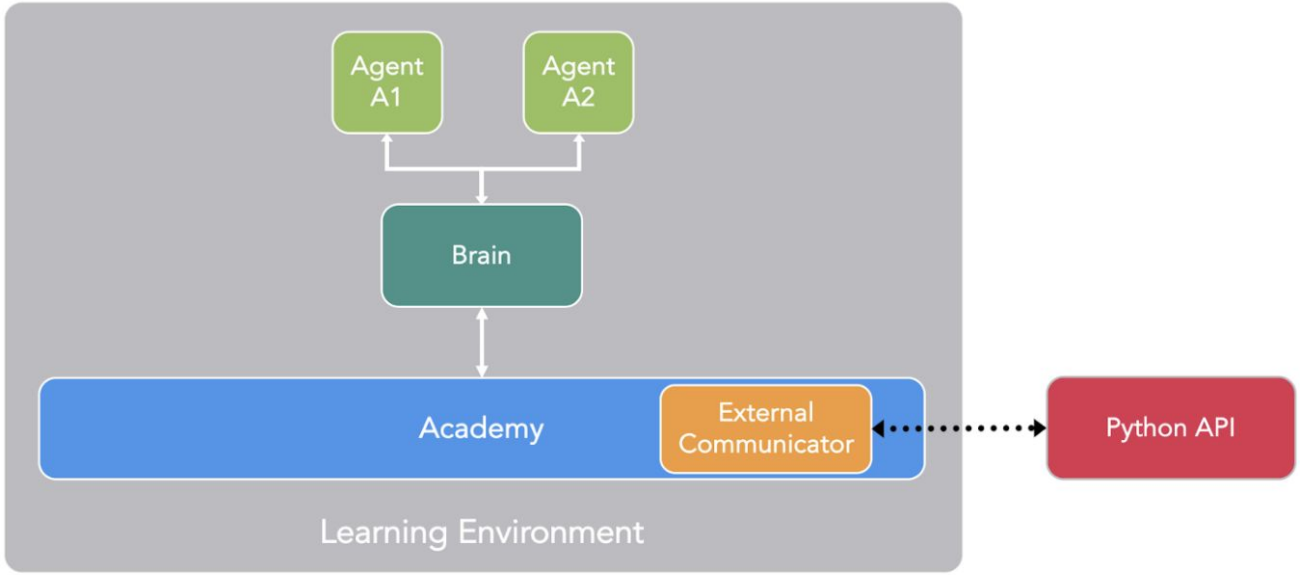
Introduction

- New plugin to Unity game engine
- What is Unity?
- IDE (Integrated Development Environment)
- Useful tool
- Knowledge

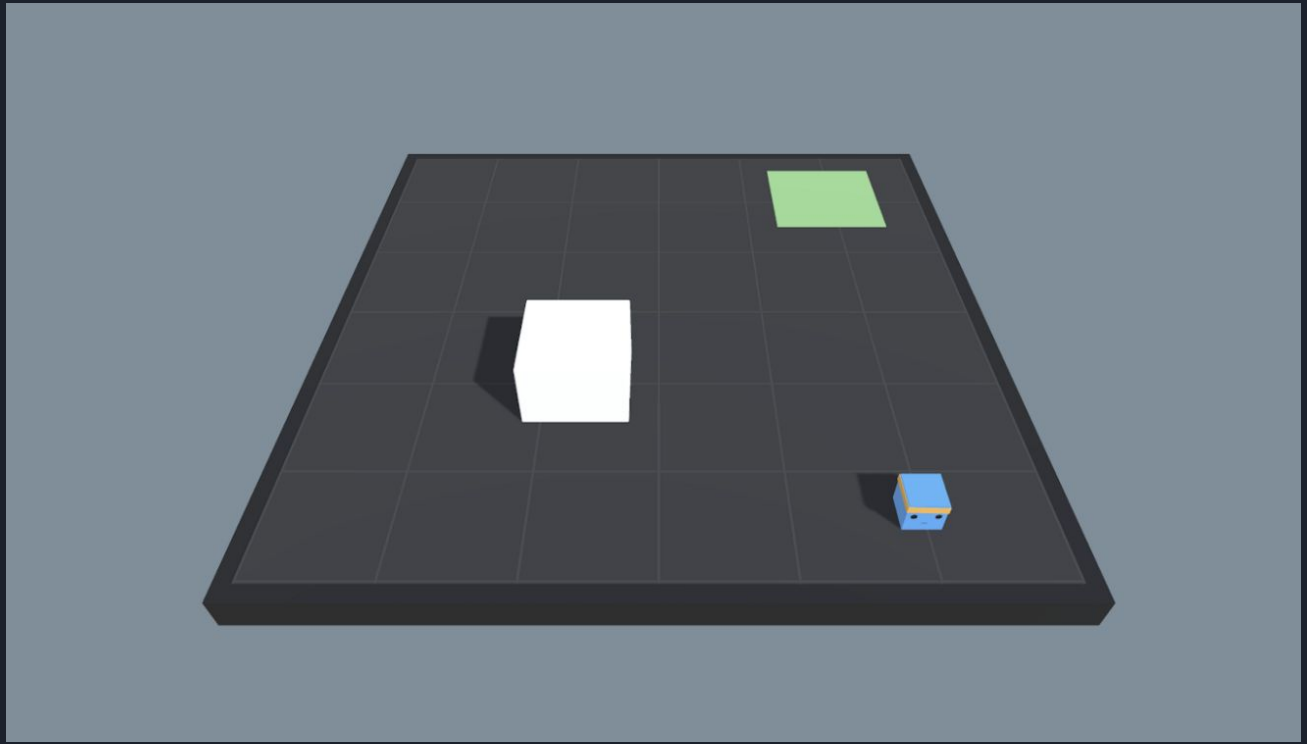
How does it work?

- toolkit is based on Python and TensorFlow
- multiple training techniques, e.g.: imitation learning, neuroevolution
- used here: reinforcement learning
  - give a reward

The components



Source: Unity ML-Agents Documentation



Example of an unity environment with the blue agent and some obstacles

The Reinforcement Training Loop

- environment is in state 0
- based on this our agent can take action 1 - e.g. step forward
- afterwards the environment has a different state
- if we reached our goal: give an reward and reset the environment

The agent will try to maximize the reward.

reset as well when a maximum amount of steps is reached.

Demonstration

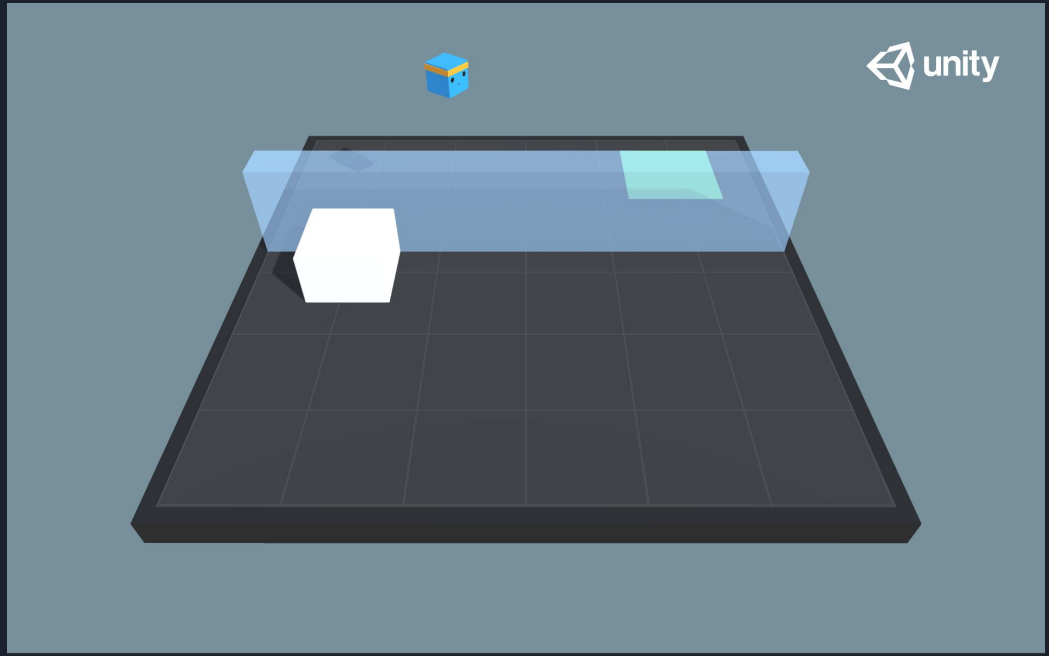
For this demonstration, the examples on the Unity ML Agents github page were used, as we could use them immediately after downloading them. You can obtain them by downloading their github repository here.

WallJump

Objective: Get to the green zone with as many points as possible -> as quick as possible

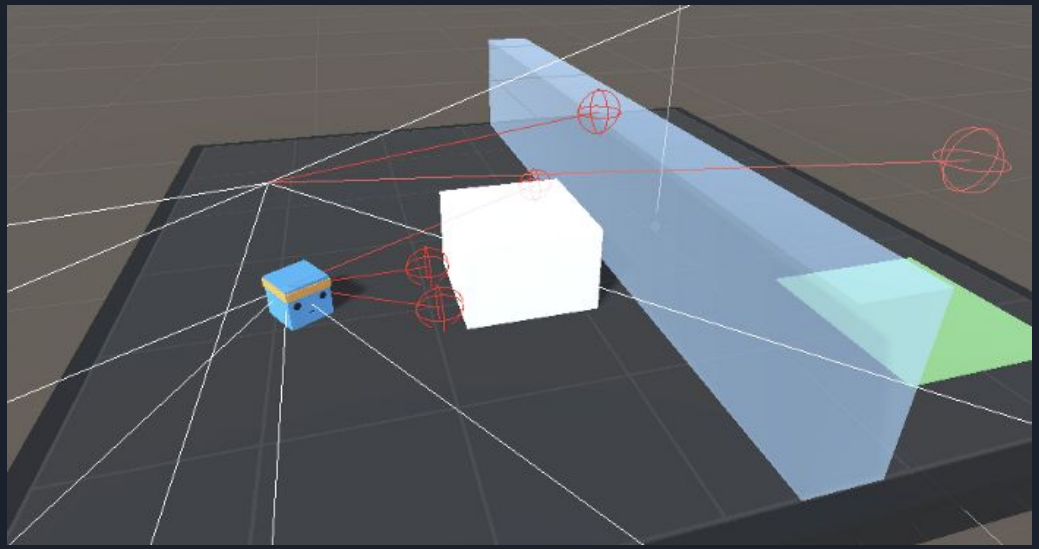
Reward system:

1.- -0.005 for every step so it moves as fast as possible to the green zone
2.- +1 if the agent reaches the goal
3.- -0.01 if the agent falls of the platform



WallJump

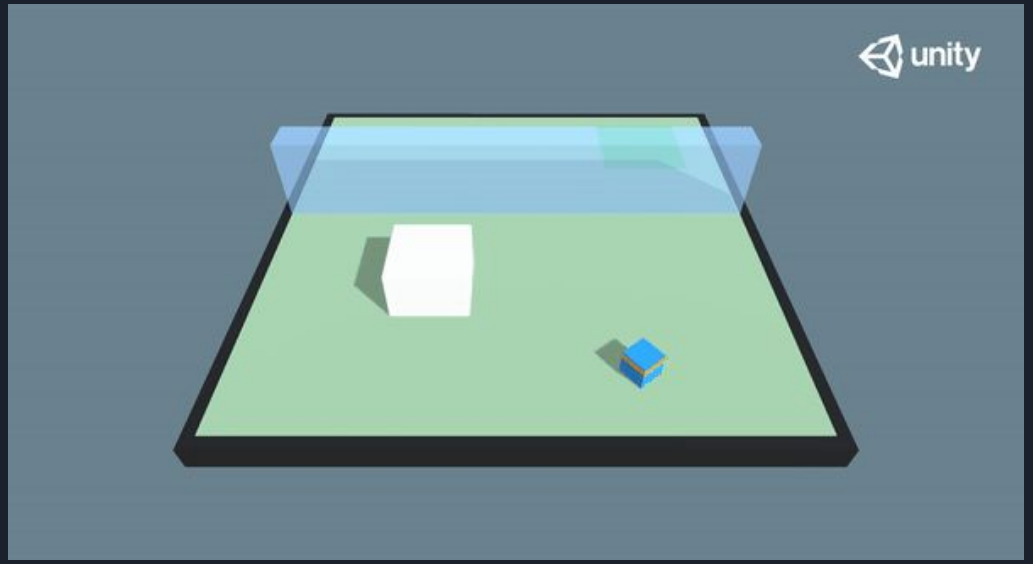
To observe it uses 14 raycasts that each detect 4 possible objects. The global position of the agent is also used.



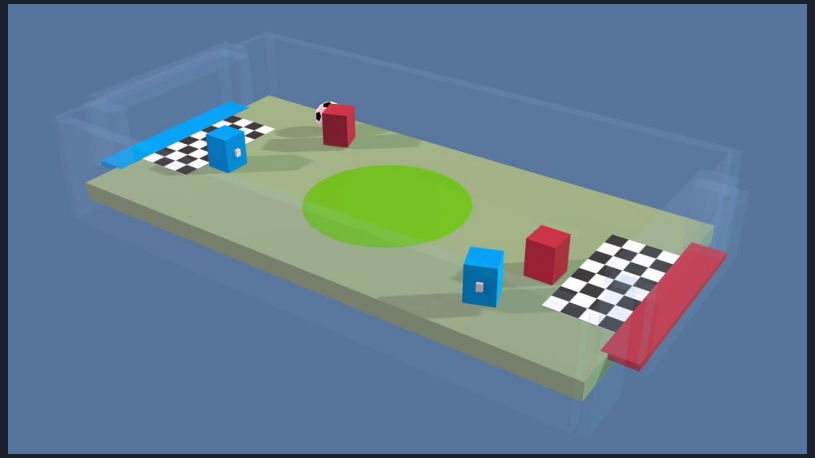
WallJump

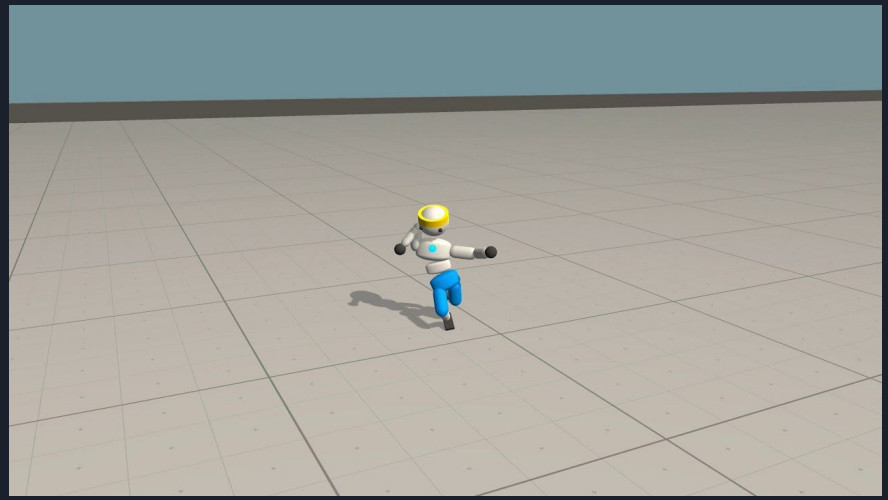
The action space has 4 branches:

- Forward Motion: UP / DOWN / NO ACTION
- Rotation: ROTATE LEFT / ROTATE RIGHT / NO ACTION
- Side Motion: LEFT / RIGHT / NO ACTION
- Jump: JUMP / NO ACTION



More Examples





How to prepare our system to use this toolkit

- Install Python and Pytorch versions that are in the documentation in GitHub, as the project is constantly being updated. Recommended Python 3.6 and 3.7.
- Recommended to create a python virtual environment:
  python -m venv venv (windows)
- Install PIP.
- Iinstall ml-agents. Compatibility errors? → Fix them by installing asked dependencies. (usually happens with the GPU dependencies).
- In Unity install ML Agents package: package manager (window → Package Manager) → Enable Unity registry → ML Agents.

Bibliography

- [GitHub repository]: https://github.com/Unity-Technologies/ml-agents
- [Article]: An Introduction to Unity ML-Agents | by Thomas Simonini
- [Video]: How to use Machine Learning AI in Unity! (ML-Agents)
- [Video] Reinforcement Learning: Crash Course AI
- [Video]: Unity at GDC
- [Video]: Unity ML-Agents Environment - SoccerTwos
- [Video]: Unity ML-Agents Environment - Walker