Erik Angulo
Xabier Lahuerta
Oihane Roca

REINFORCEMENT LEARNING IN

MINECRAFT

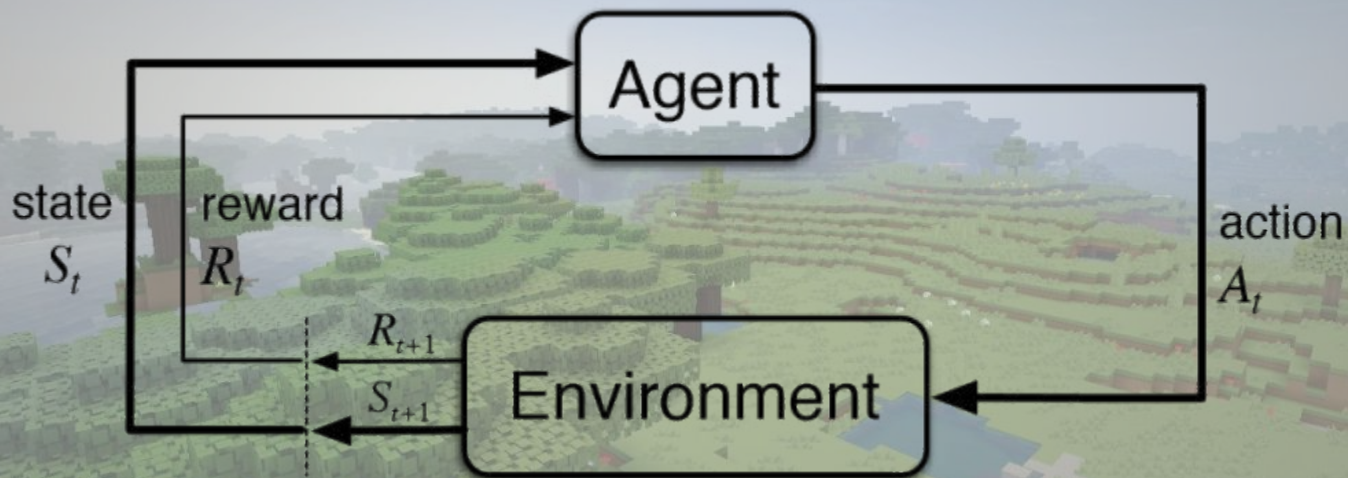# What is minecraft?

Reinforcement Learning (RL)

MARLÖ

MINERL

Reinforcement Learning

state S_t

reward R_t

Agent

action A_t

R_{t+1}

S_{t+1}

Environment

States, observations, policies and trajectories

Policies

Deterministic:
$$a_t = \mu(s_t)$$

Stochastic:
$$a_t \sim \pi(\cdot | s_t)$$

Trajectories / Episodes

$$\tau = (s_0, a_0, s_1, a_1, \dots)$$

State transitions

$$s_0 \sim \rho_0(\cdot)$$

Deterministic:
$$s_{t+1} = f(s_t, a_t)$$

Stochastic:
$$s_{t+1} \sim P(\cdot | s_t, a_t)$$

Returns and rewards

$$r_t = R(s_t, a_t, s_{t+1})$$

$$P(\tau|\pi) = \rho_0(s_0) \prod_{t=0}^{T-1} P(s_{t+1}|s_t, a_t) \pi(a_t|s_t)$$

$$R(\tau) = \sum_{t=0}^{T} r_t$$

$$J(\pi) = \int_\tau P(\tau|\pi) R(\tau) = \mathop{\mathbf{E}}_{\tau \sim \pi}[R(\tau)]$$

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t$$

Optimal Policy

$$\pi^* = arg \max_\pi J(\pi)$$

Environment generator (XML)

```xml
<?xml version="1.0" encoding="UTF-8" ?>

<Mission xmlns="http://ProjectMalmo.microsoft.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <About>

        <Summary>Everyday Minecraft life: survival</Summary>

    </About>


    <ServerSection>

        <ServerHandlers>

            <DefaultWorldGenerator />

            <ServerQuitFromTimeUp timeLimitMs="3000000" description="out_of_time"/>

            <ServerQuitWhenAnyAgentFinishes />

        </ServerHandlers>

    </ServerSection>

    <AgentSection mode="Survival">
        <Name>Rover</Name>
        <AgentStart>
            <Placement x="-203.5" y="81.0" z="217.5"/>
        </AgentStart>
        <AgentHandlers>
            <ContinuousMovementCommands/>
            <ObservationFromFullStats/>
            <VideoProducer want_depth="false">
                <Width>480</Width>
                <Height>320</Height>
            </VideoProducer>
            <RewardForMissionEnd rewardForDeath="-10000">
                <Reward description="found_goal" reward="1000" />
                <Reward description="out_of_time" reward="-1000" />
            </RewardForMissionEnd>
            <AgentQuitFromTouchingBlockType>
                <Block type="gold_block diamond_block redstone_block" description="found_goal" />
            </AgentQuitFromTouchingBlockType>
        </AgentHandlers>
    </AgentSection>
</Mission>
```
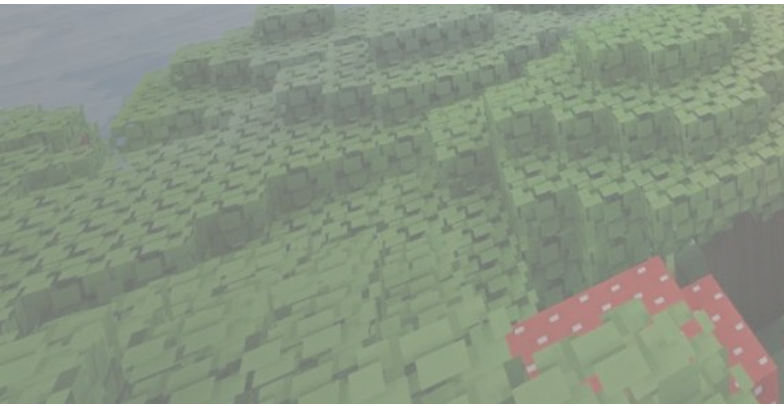
Functions / agents objectives / environments

- **MarLo-MazeRunner-v0**

- **MarLo-CatchTheMob-v0**

- **MarLo-Attic-v0**

Usage

Launching client

```
$MALMO_MINECRAFT_ROOT/launchClient.sh -port 10000
```

Creating the environment

```
1   import marlo
2   client_pool = [('127.0.0.1', 10000)]
3   join_tokens = marlo.make('MarLo-MazeRunner-v0',
4                            params={
5                                "client_pool": client_pool,
6                                "agent_names" : ["MarLo-Agent-0"]
7                            })
8   assert len(join_tokens) == 1
9   env = marlo.init(join_tokens[0])
10  env.reset()
```

Game loop

```
1   done = False
2
3   while not done:
4       _action = env.action_space.sample()
5       obs, reward, done, info = env.step(_action)
6
7   env.close()
```

Conclusions

- A good approach to understand Reinforcement Learning

- Not much documentation and code examples (MarLÖ)

Thanks OpenAI !

Demos

Questions?

Bibliography

- https://marlo.readthedocs.io/en/latest/modules.html
- https://github.com/crowdAI/marLo
- https://minerl.io/docs/
- https://github.com/minerllabs/minerl
- https://www.aicrowd.com/challenges/neurips-2020-minerl-competition
- https://arxiv.org/pdf/1912.08664v2.pdf (our initial demo, no time to train :( )
- https://spinningup.openai.com/en/latest/