QS STARS RATED FOR EXCELLENCE 2012

FPT Education

FPT UNIVERSITY

Building Machine Learning Bot with ML-Agents in Tank Battle

Author: Van Duc Dung
Thesis Supervisor: Assoc. Prof. Phan Duy Hung

Authors

Van Duc Dung

Author

Assoc. Prof. Phan Duy Hung

Supervisor

Table of Contents

01

Introduction

Reinforcement Learning in Video Game

Introduction

Video Games

are inherently an extremely fertile ground for AI research, especially Reinforcement Learning

History

There have been many AIs that beat humans in video games, from very simple to highly complex.

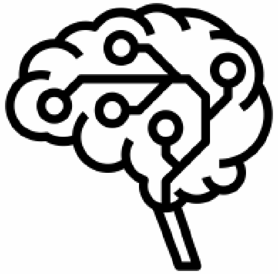
Commercial

AI in video games is still for research only.

Introduction

Unity

is a 3D ultimate game engine development platform



ML-Agents Toolkit

is an open-source project that enables games and simulations to serve as environments for training intelligent agents

QS STARS RATED FOR EXCELLENCE 2012

FPT Education

FPT UNIVERSITY

02

Background

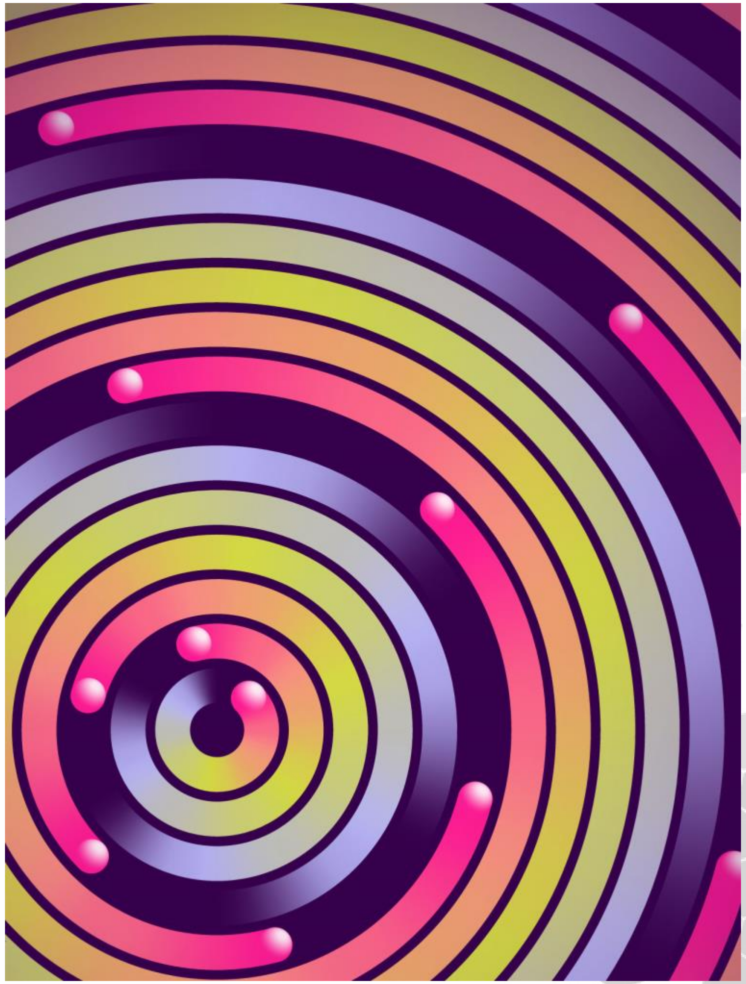
Proximal Policy Optimization, Curriculum Learning, Self-play, Unity3D and ML-Agents Toolkit

Background

Proximal Policy Optimization

”PPO has become the default reinforcement learning algorithm at OpenAI because of its ease of use and good performance”

OpenAI



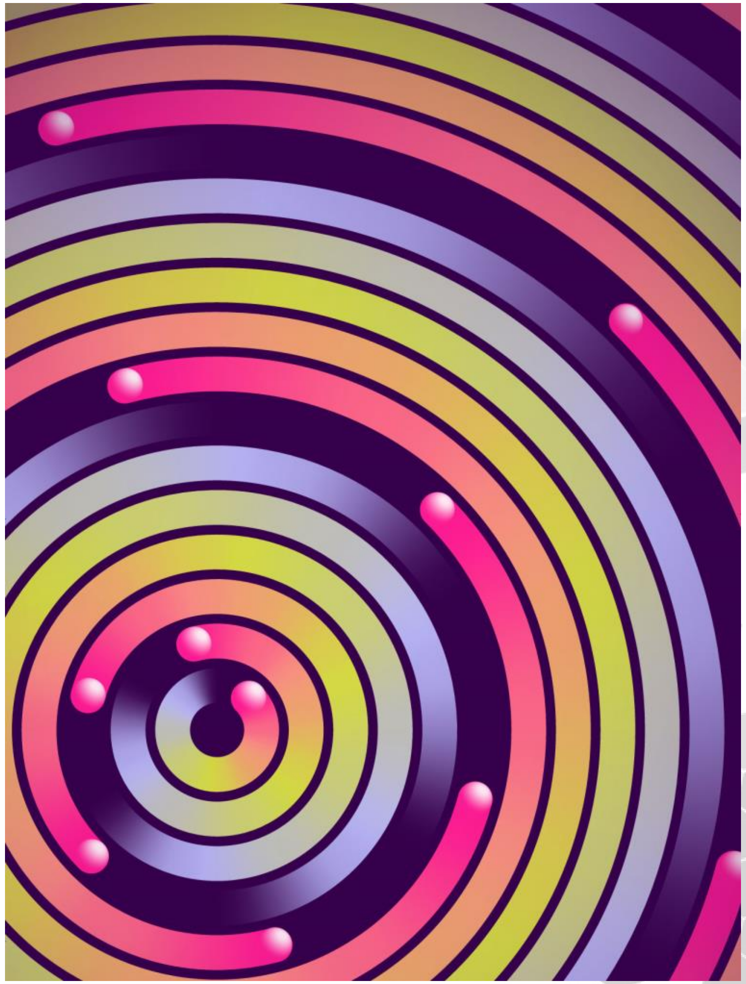
Background

Proximal Policy Optimization

Policy Gradiant

$$L^{PG}(\theta) = \hat{\mathbb{E}}_t\left[\log \pi_\theta(a_t|s_t)\hat{A}_t\right]$$

A destructively large policy.



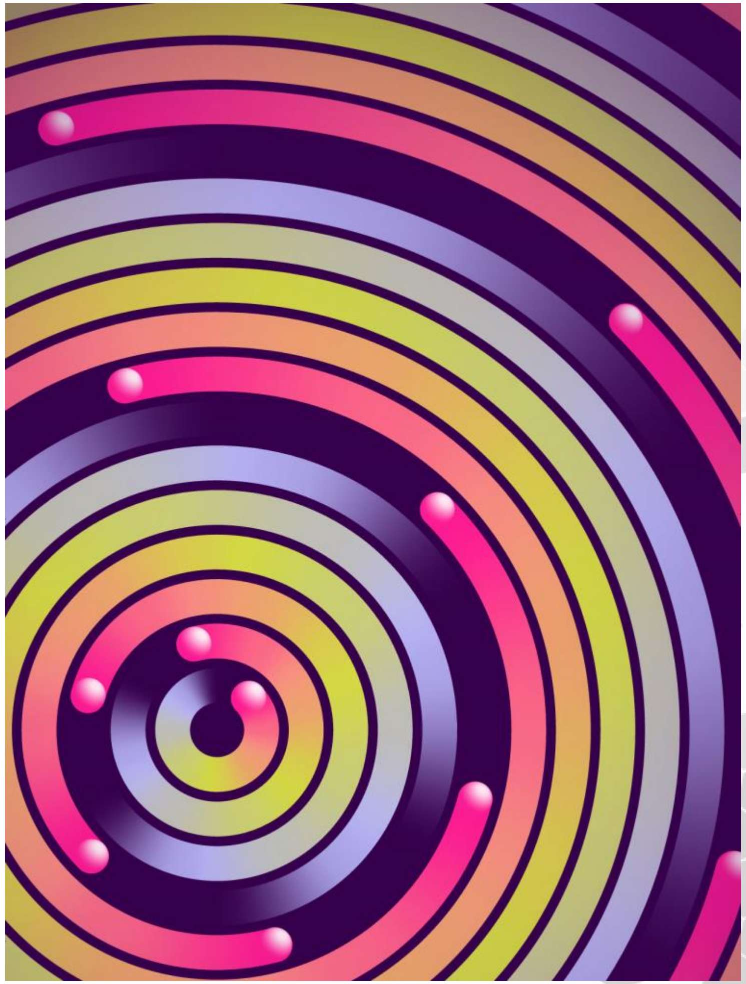
Background

Proximal Policy Optimization

Trust Region Policy Optimization

$r_t(\theta)$: probability ratio between the action under the current policy and the action under the previous policy.

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}, \qquad so\ r(\theta_{old}) = 1$$



Background

Proximal Policy Optimization

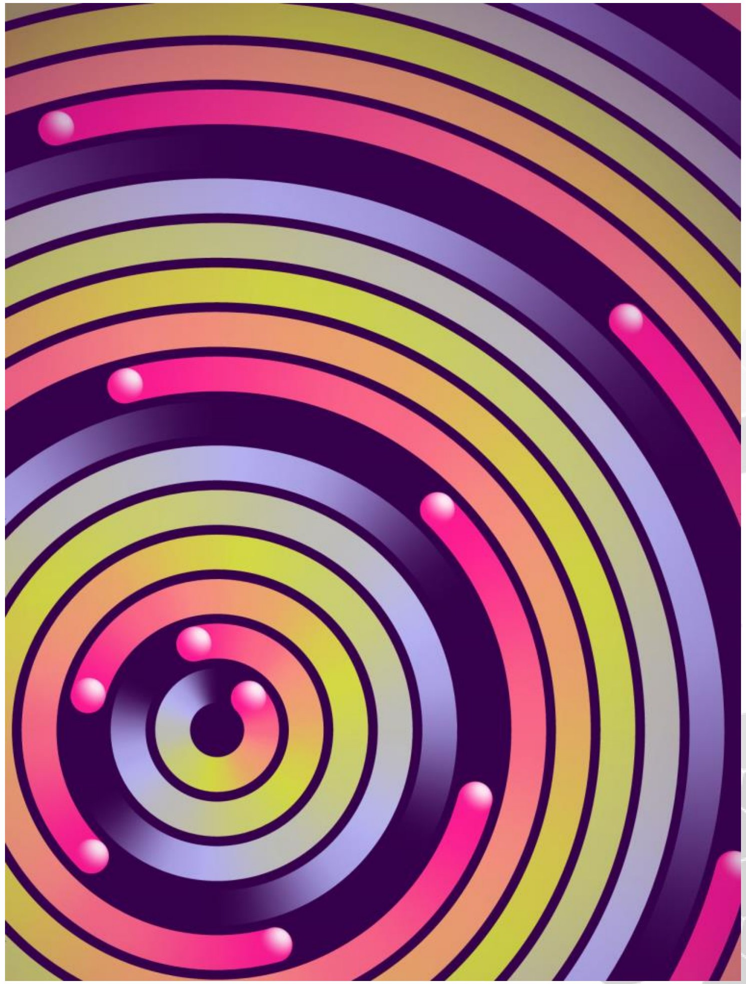
Trust Region Policy Optimization (TRPO)

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}, \qquad so\ r(\theta_{old}) = 1$$

TRPO object’s function:

$$\underset{\theta}{\text{maximize}} \quad \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\hat{A}_t\right]$$

$$subject\ to \quad \hat{\mathbb{E}}_t\left[KL\left[\pi_{\theta_{old}}(\cdot\,|s_t), \pi_\theta(\cdot\,|s_t)\right]\right] \le \delta$$



Background

Proximal Policy Optimization

Clipped Surrogate Objective

With the motives mentioned above, Proximal Policy Optimization attempts to simplify the optimization process while retaining the advantages of TRPO.

$$L^{CLIP}(\theta) = \hat{E}_t\left[\min(r_t(\theta)\,\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$

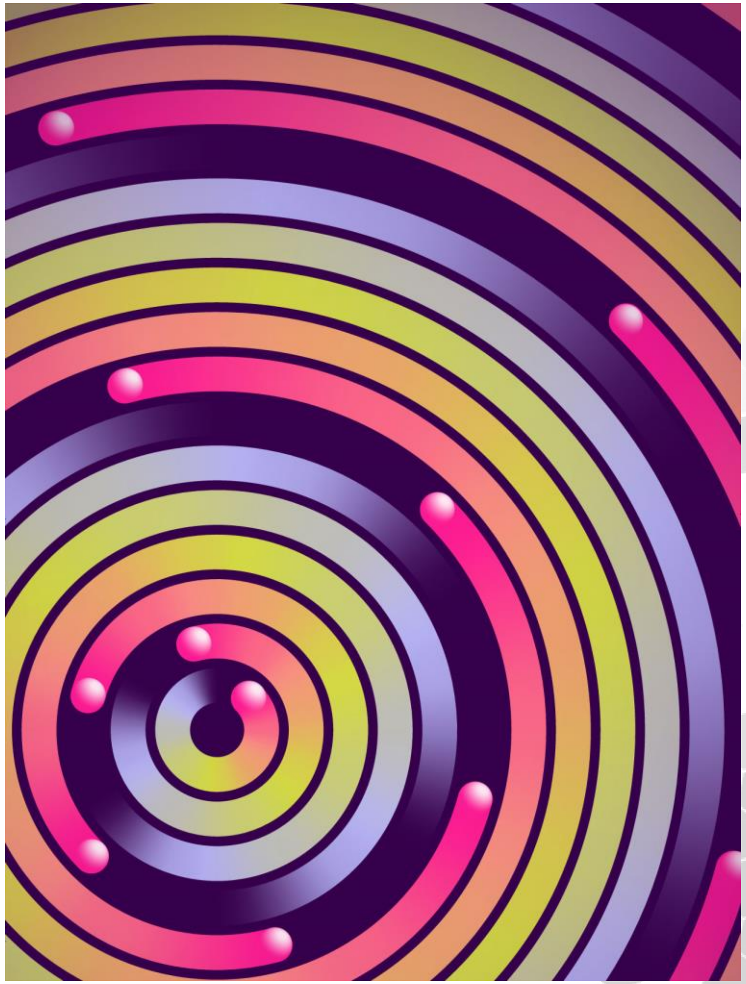
θ is the policy parameter

$\hat{E}_t$ denotes the empirical expectation over timesteps

r_t is the ratio of the probability under the new and old policies, respectively

$\hat{A}_t$ is the estimated advantage at time t

ϵ is a hyperparameter, usually 0.1 or 0.2



Background

Proximal Policy Optimization

Clipped Surrogate Objective

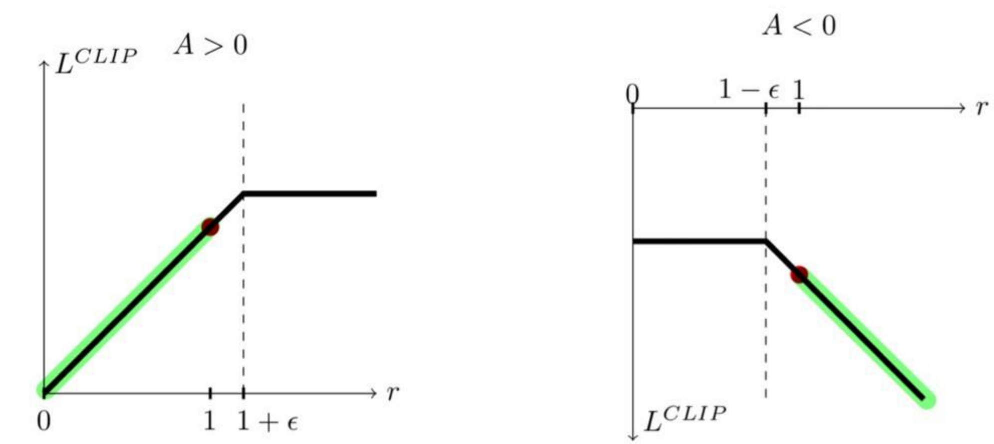
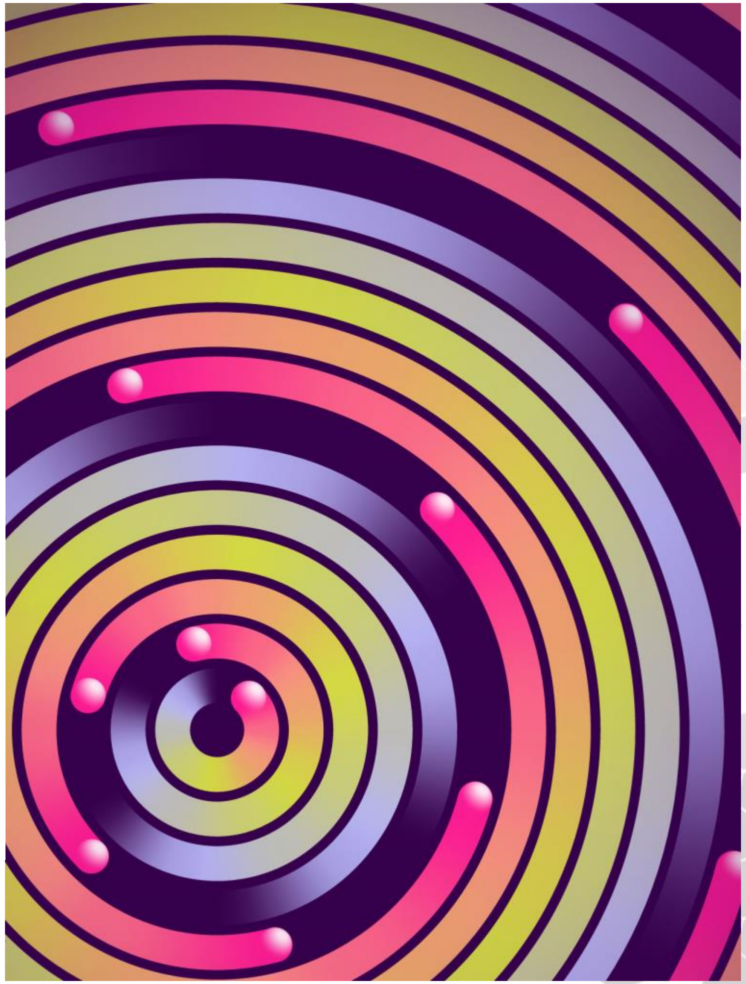


Figure 1: Plots showing one term (i.e., a single timestep) of the surrogate function L^{CLIP} as a function of the probability ratio r, for positive advantages (left) and negative advantages (right). The red circle on each plot shows the starting point for the optimization, i.e., $r = 1$. Note that L^{CLIP} sums many of these terms.



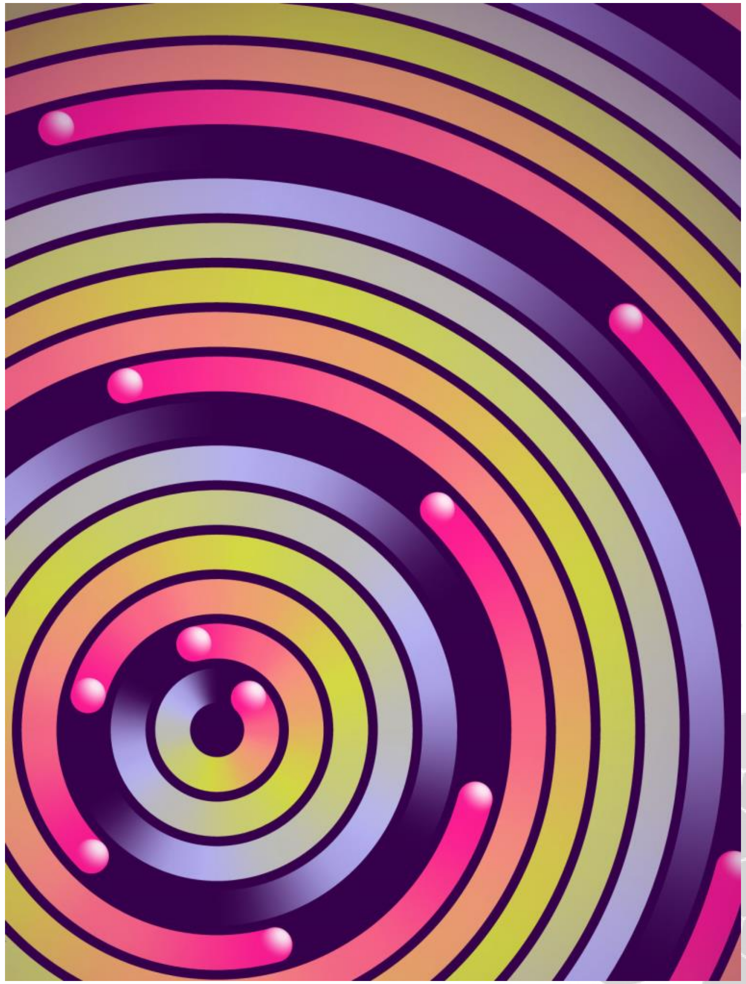
Background

Proximal Policy Optimization

Clipped Surrogate Objective

Final Objective:

$$L_t^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta(s_t)]$$



Background

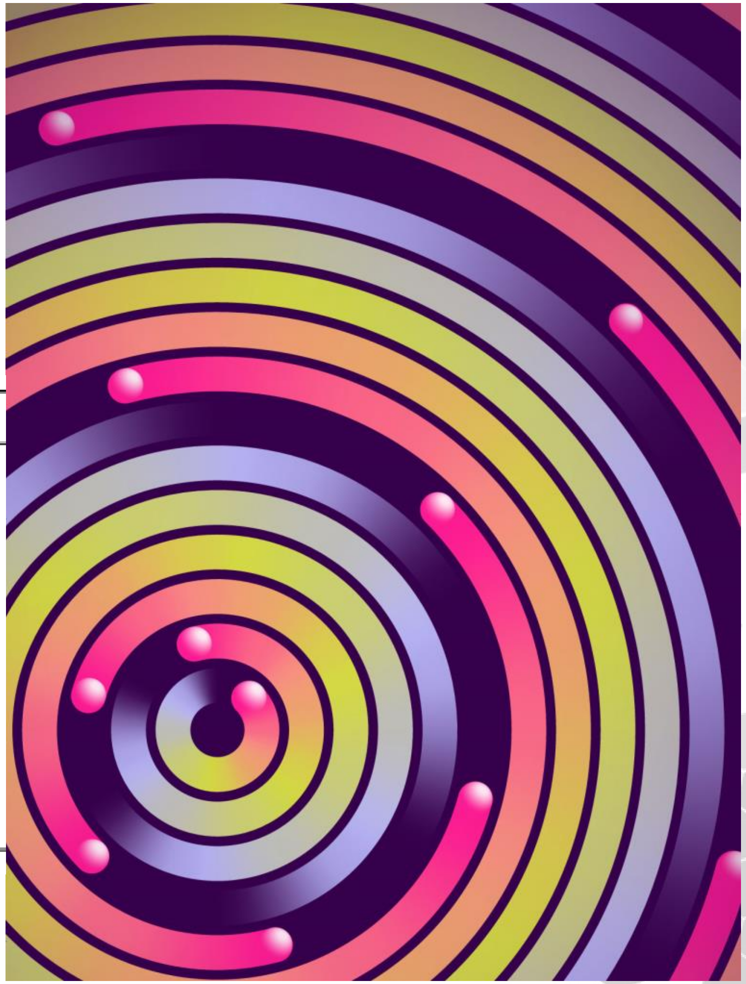
Proximal Policy Optimization

Multiple Epochs for Policy Updating

The algorithm altogether:

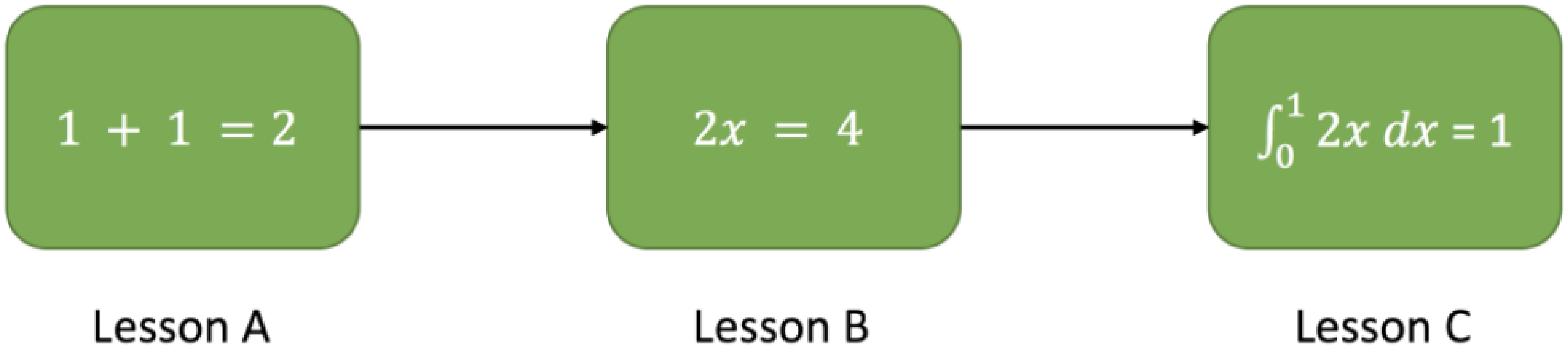
Algorithm 1 PPO, Actor-Critic Style

for iteration=$1, 2, \ldots$ **do**
  for actor=$1, 2, \ldots, N$ **do**
    Run policy $\pi_{\theta_{\text{old}}}$ in environment for T timesteps
    Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$
  end for
  Optimize surrogate L wrt θ, with K epochs and minibatch size $M \leq NT$
  $\theta_{\text{old}} \leftarrow \theta$
end for



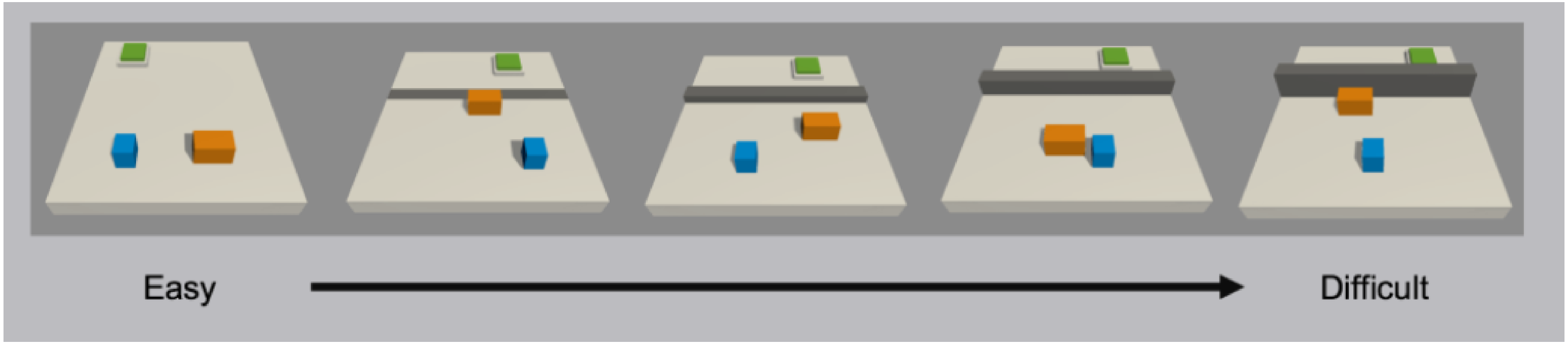
Background

Curriculum Learning



Background

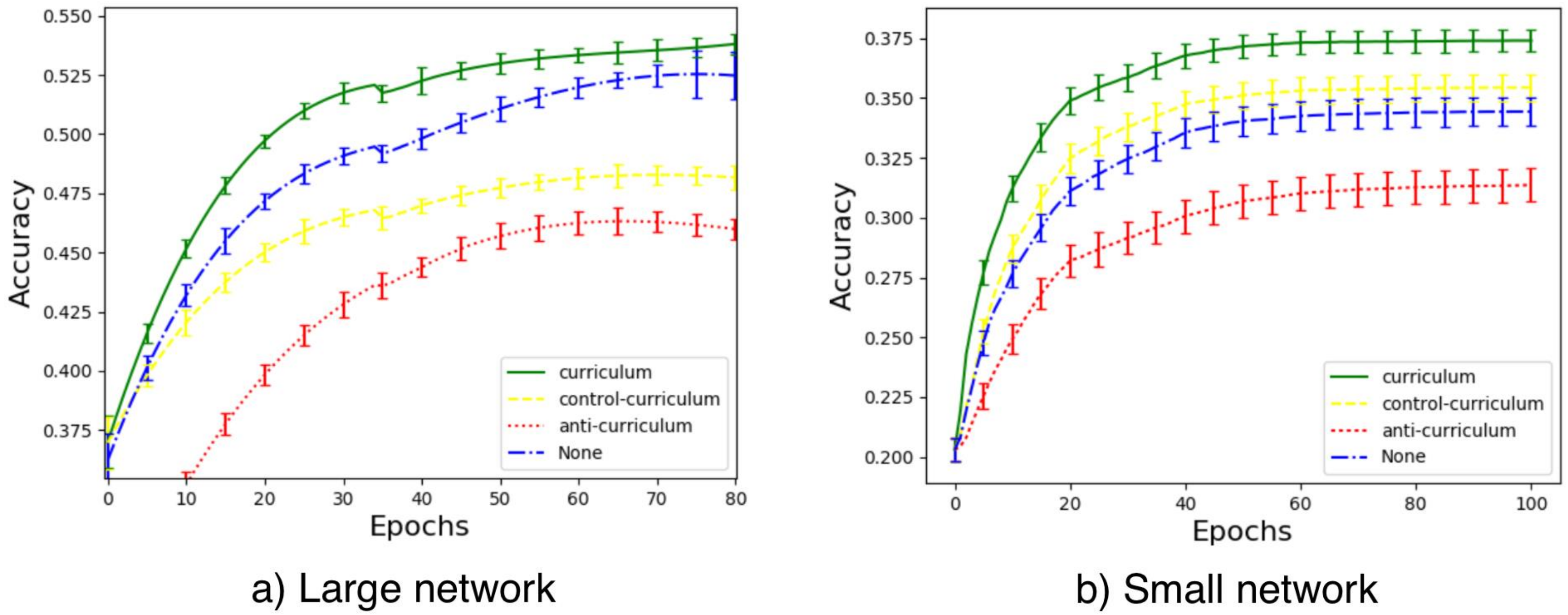
Curriculum Learning



Example of Curriculum Learning in Reinforcement Learning

Background

Curriculum Learning



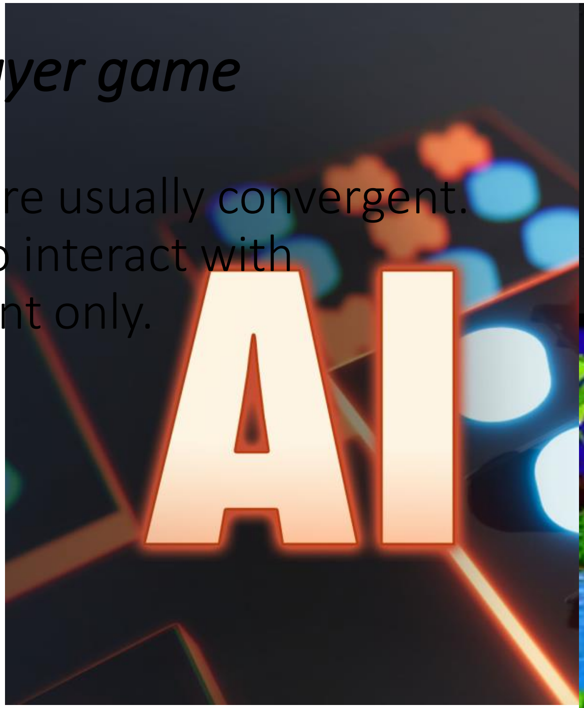
(Image source: [Weinshall, et al. 2018](#))

Background

Self-Play

In Single-player game

- Solutions are usually convergent.
- Agents also interact with environment only.

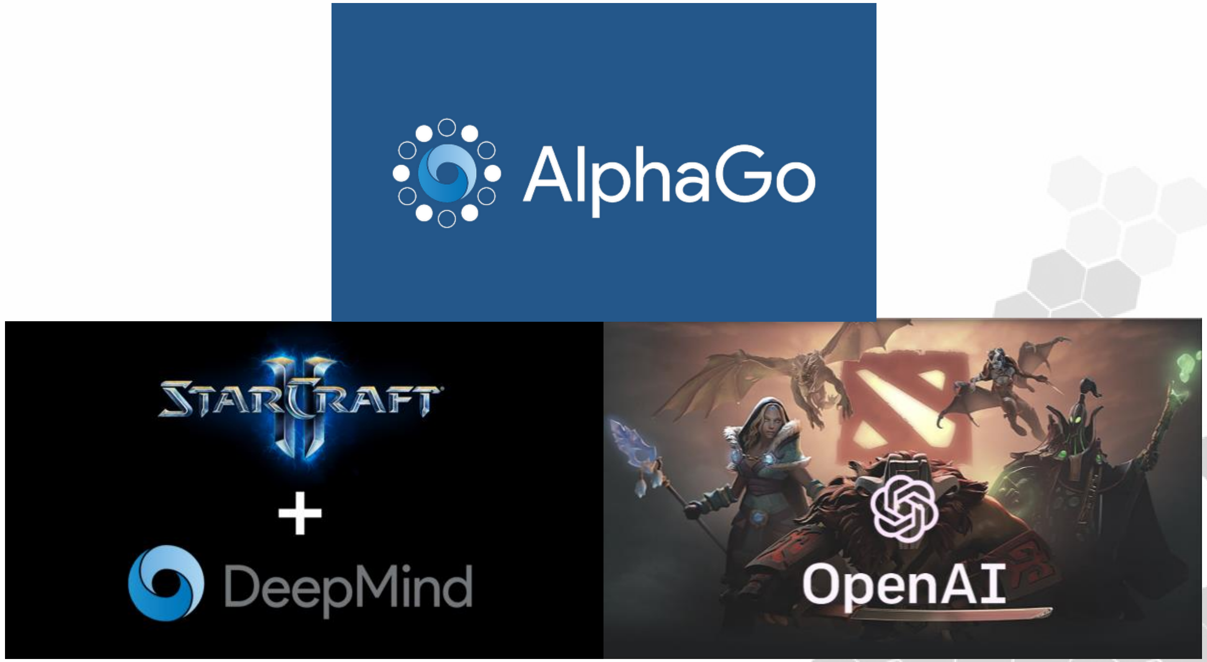


Background

Self-Play

In Multiplayer Game:

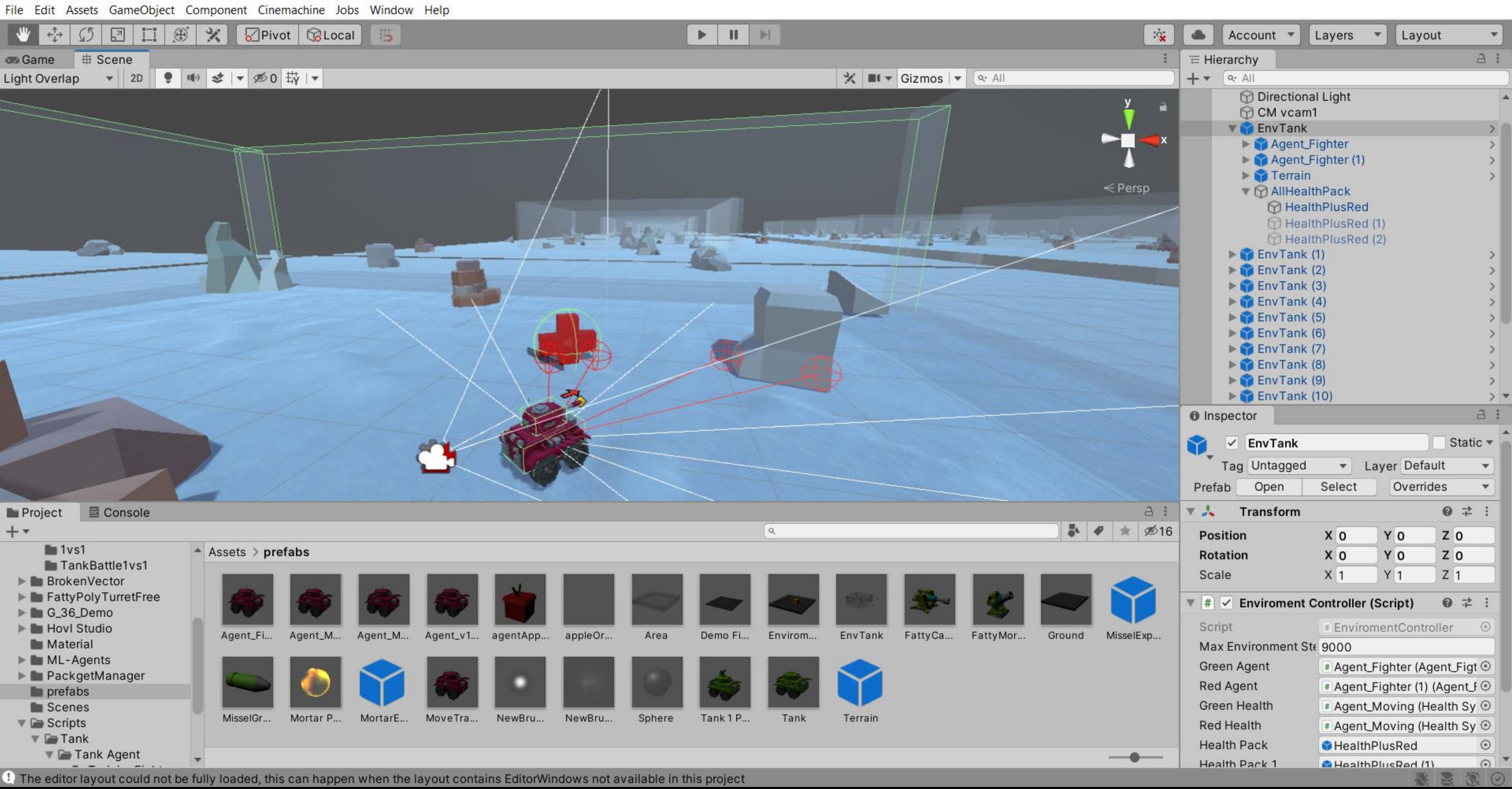
- Interact with both environment and opponent
- Data from bot-handscripted:
  => Not diverse
- Data from human:
  => Not possible



Background

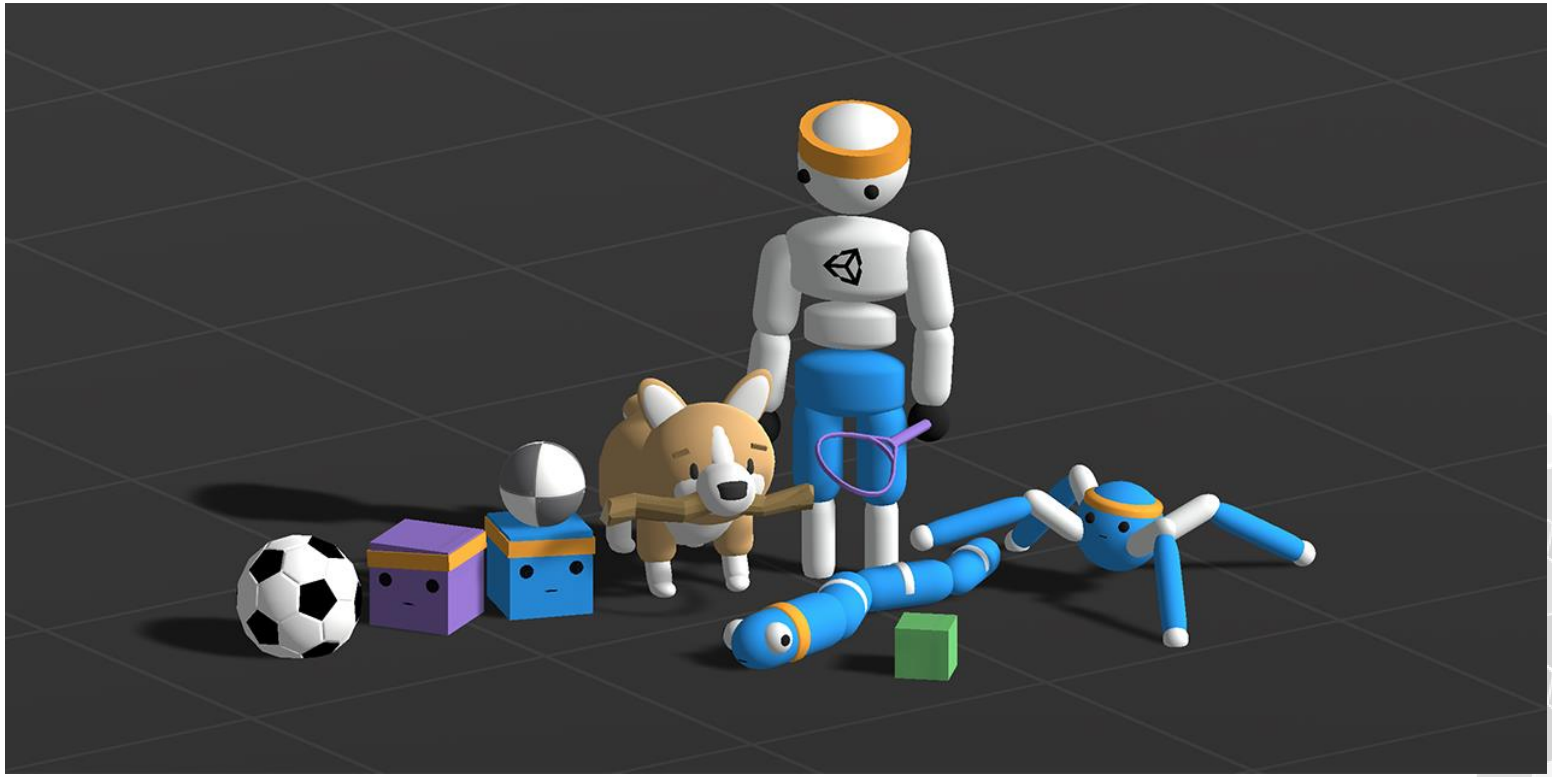
Unity

- Easy enough for the beginner
- Powerful cross-platform 3D engine for experts


File Edit Assets GameObject Component Cinemachine Jobs Window Help
Pivot
Local
Account
Layers
Layout
Game
Scene
Light Overlap
2D
Gizmos
All
Persp
Hierarchy
Directional Light
CM vcam1
EnvTank
Agent_Fighter
Agent_Fighter (1)
Terrain
AllHealthPack
HealthPlusRed
HealthPlusRed (1)
HealthPlusRed (2)
EnvTank (1)
EnvTank (2)
EnvTank (3)
EnvTank (4)
EnvTank (5)
EnvTank (6)
EnvTank (7)
EnvTank (8)
EnvTank (9)
EnvTank (10)
Inspector
EnvTank
Static
Tag Untagged
Layer Default
Prefab Open Select Overrides
Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Enviroment Controller (Script)
Script EnviromentController
Max Environment Ste 9000
Green Agent Agent_Fighter (Agent_Figt
Red Agent Agent_Fighter (1) (Agent_I
Green Health Agent_Moving (Health Sy
Red Health Agent_Moving (Health Sy
Health Pack HealthPlusRed
Health Pack 1 HealthPlusRed (1)
Project
Console
Assets > prefabs
1vs1
TankBattle1vs1
BrokenVector
FattyPolyTurretFree
G_36_Demo
Hovl Studio
Material
ML-Agents
PackgetManager
prefabs
Scenes
Scripts
Tank
Tank Agent
Agent_Fi...
Agent_M...
Agent_M...
Agent_v1...
agentApp...
appleOr...
Area
Demo Fi...
Envirom...
EnvTank
FattyCa...
FattyMor...
Ground
MisselExp...
MisselGr...
Mortar P...
MortarE...
MoveTra...
NewBru...
NewBru...
Sphere
Tank 1 P...
Tank
Terrain
The editor layout could not be fully loaded, this can happen when the layout contains EditorWindows not available in this project

Background

ML-Agents Toolkit



Background

ML-Agents Toolkit

The basic idea of training agents

- Observations
- Actions
- Rewards signal

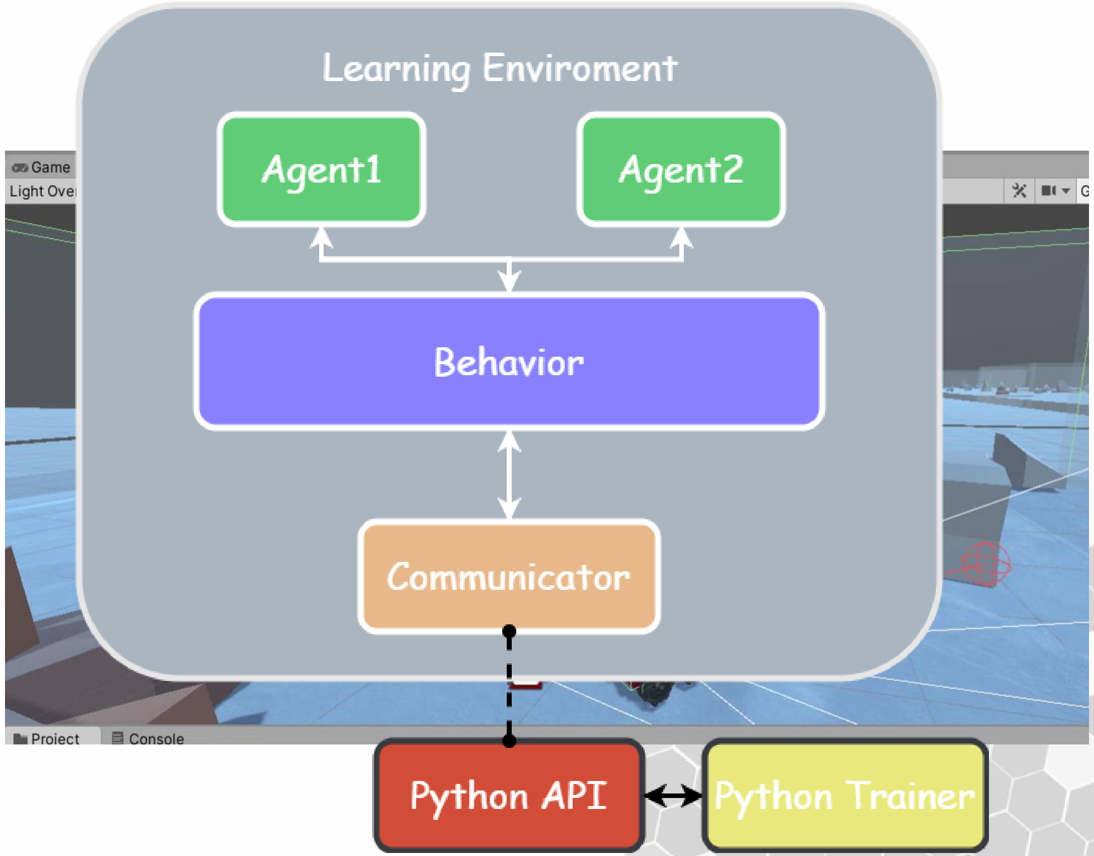
Reinforcement Learning

Background

ML-Agents Toolkit

Key Components

- **Learning Environment:** Unity scence create through Unity Editor
- **Python Low-Level API:** connects the environment agent to the learning trainers
- **External Communicator:** connection between the Low-level Python API and agent's policy.
- **Python Trainers:** provides the learning algorithm



A diagram of ML-Agents Toolkit in an Environment Learning.

03

Methodology

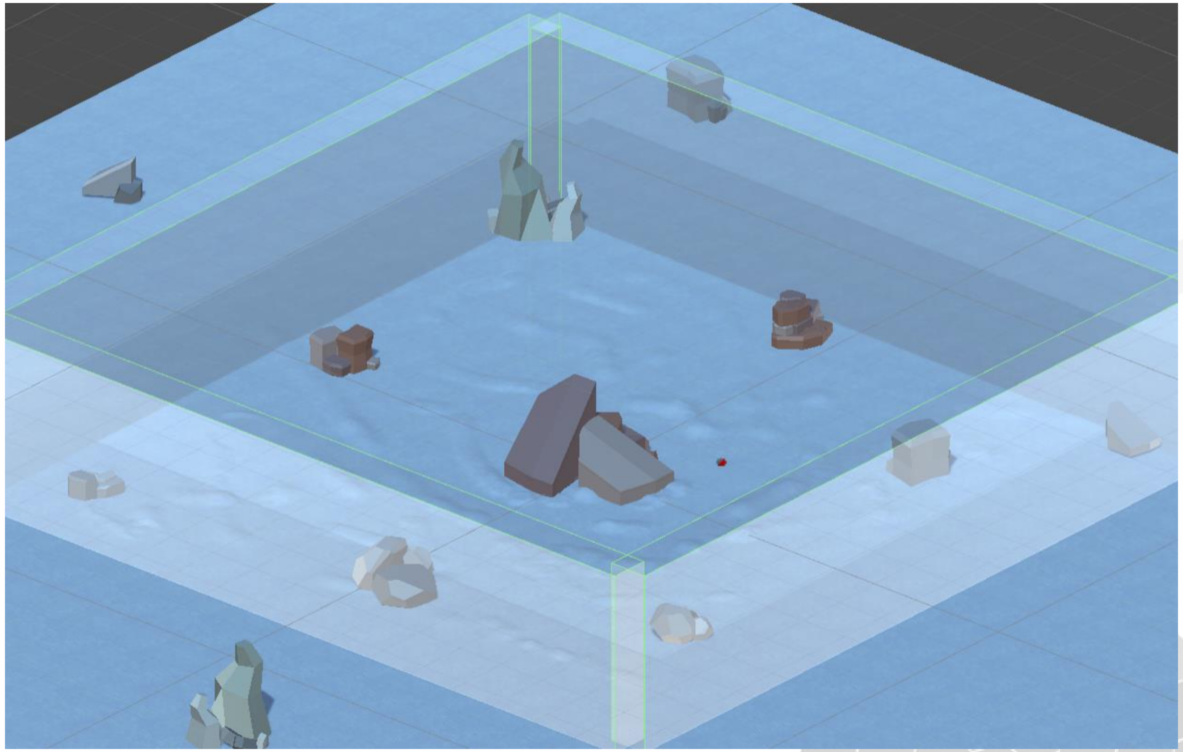
Environment Setup and Agent Design

Methodology

Agent Environment

Tank Environment

- 4 walls
- Obstacles
- Even terrain



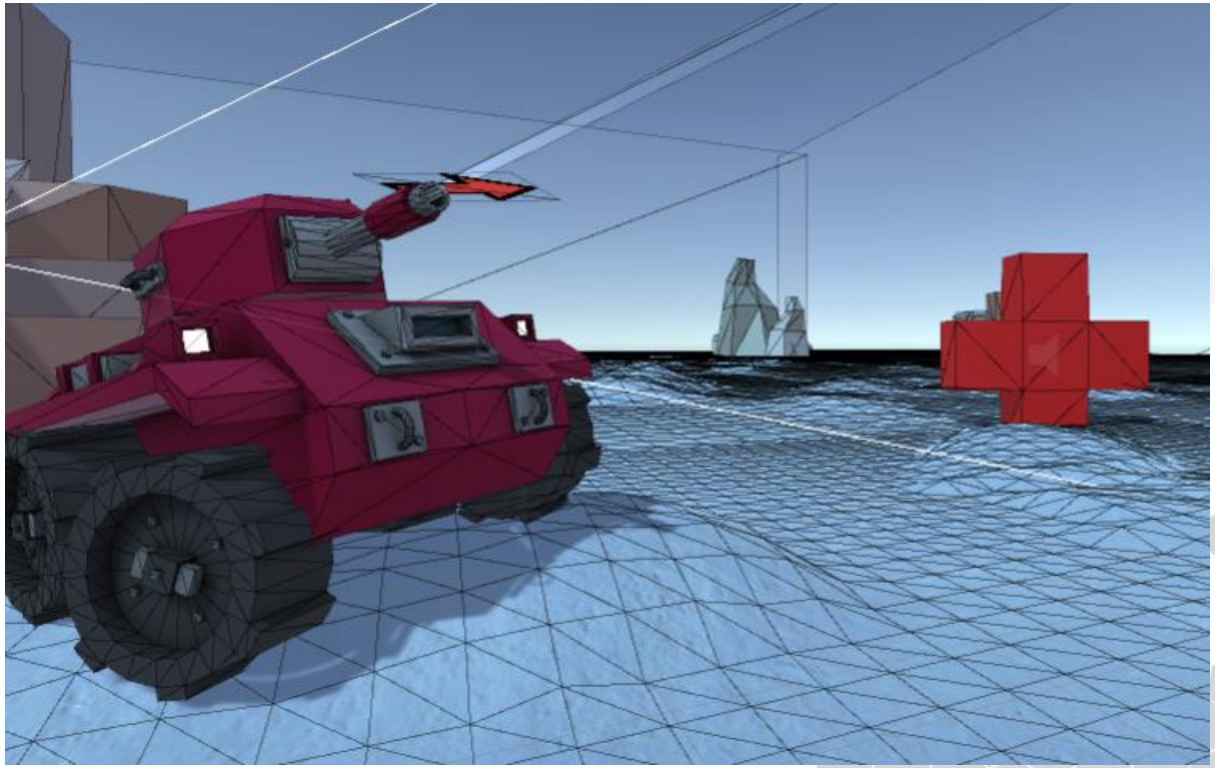
Tank Environment

Methodology

Agent Environment

Tank Environment

- 4 walls
- Obstacles
- Even terrain
- Health Packs



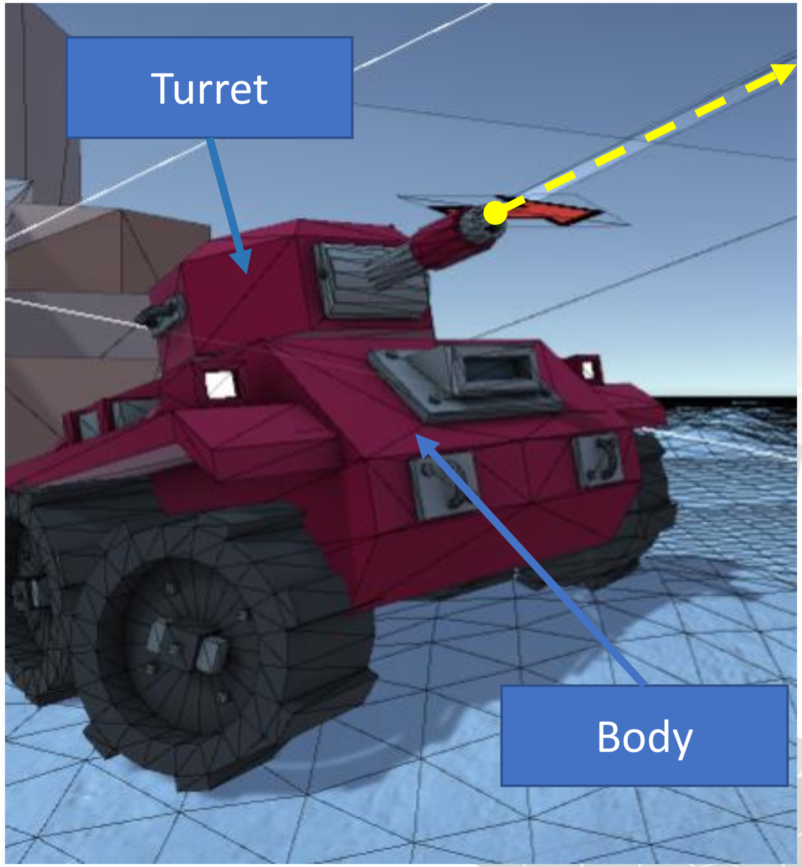
Even Terrain & Health Pack

Methodology

Agent Environment

Tank Design

- Body:
  - Four-wheel car
  - Actions: Forward, backward, turn left, turn right
  - Can only go forward
  - Backward when colliding obstacle or wall
- Turret:
  - Rotate clockwise and counterclockwise
  - Cannon control
  - Shooting control
  - Raycast for aiming



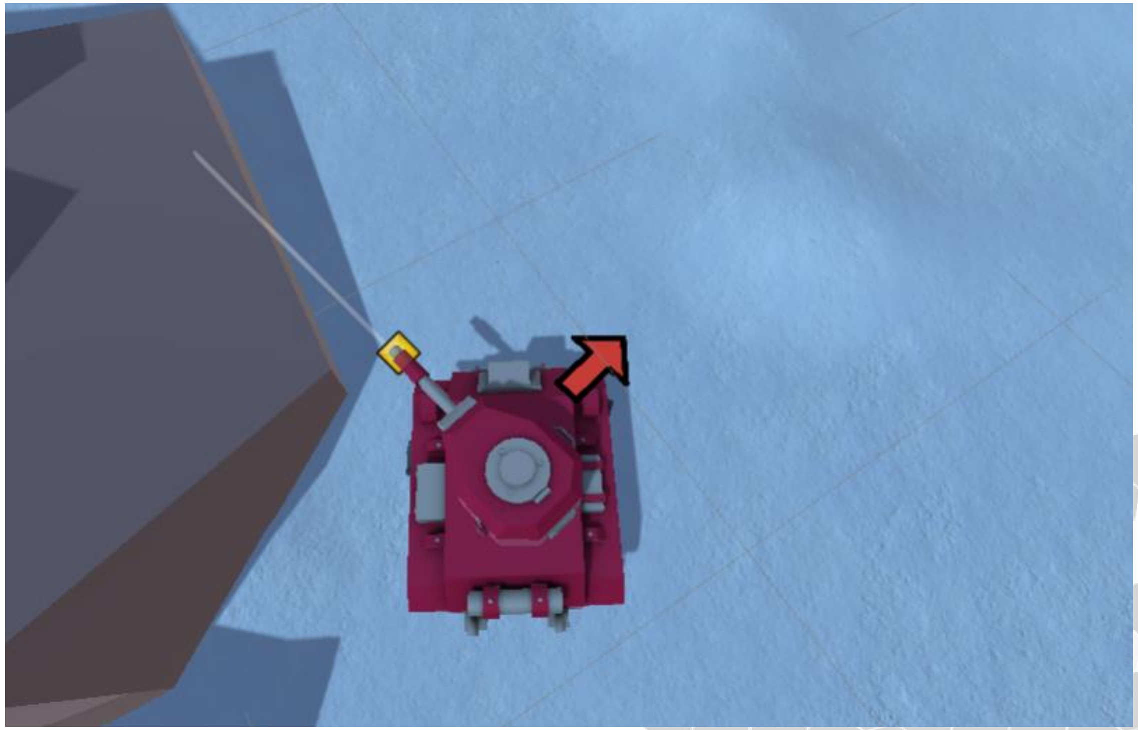
Tank Design

Methodology

Agent Environment

Tank Design

- Vector Moving:
  - The agent try to make its body direction match the Vector Moving
  => the behavior of turning left, turning right naturally

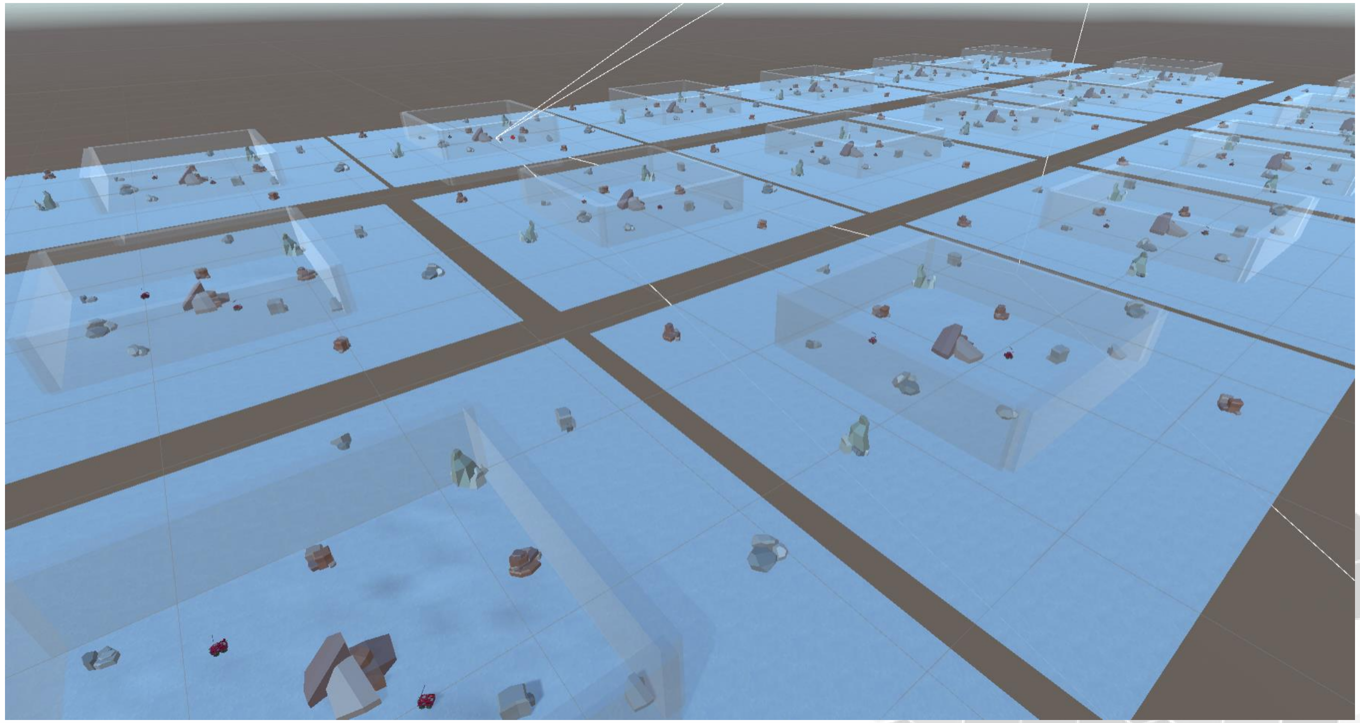


Vector Moving

Methodology

Enviroment Learning

- There are many agent that trained in duplicated environments.
- Time scale = 20;



Duplicated Environment

Methodology

Enviroment Learning

Current Position (x, z)	2
Current health percent	1
Turret's vector direction (x, z)	2
Vector from itself to enemy (x, z)	2
Fire bullet cooldown	1
Distance from the cannon to the first object that raycast hits	1
Cannon angle	1
Enemy's current health percent	1
Enemy's velocity (x, z)	2
Distance to enemy	1
Total	14

Vector Observation

Methodology

Enviroment Learning

Current Position (x, z)	2
Current health percent	1
Turret's vector direction (x, z)	2
Vector from itself to enemy (x, z)	2
Fire bullet cooldown	1
Distance from the cannon to the first object that raycast hits	1
Cannon angle	1
Enemy's current health percent	1
Enemy's velocity (x, z)	2
Distance to enemy	1
Total	14

Vector Observation

Normalizations Vector Observation

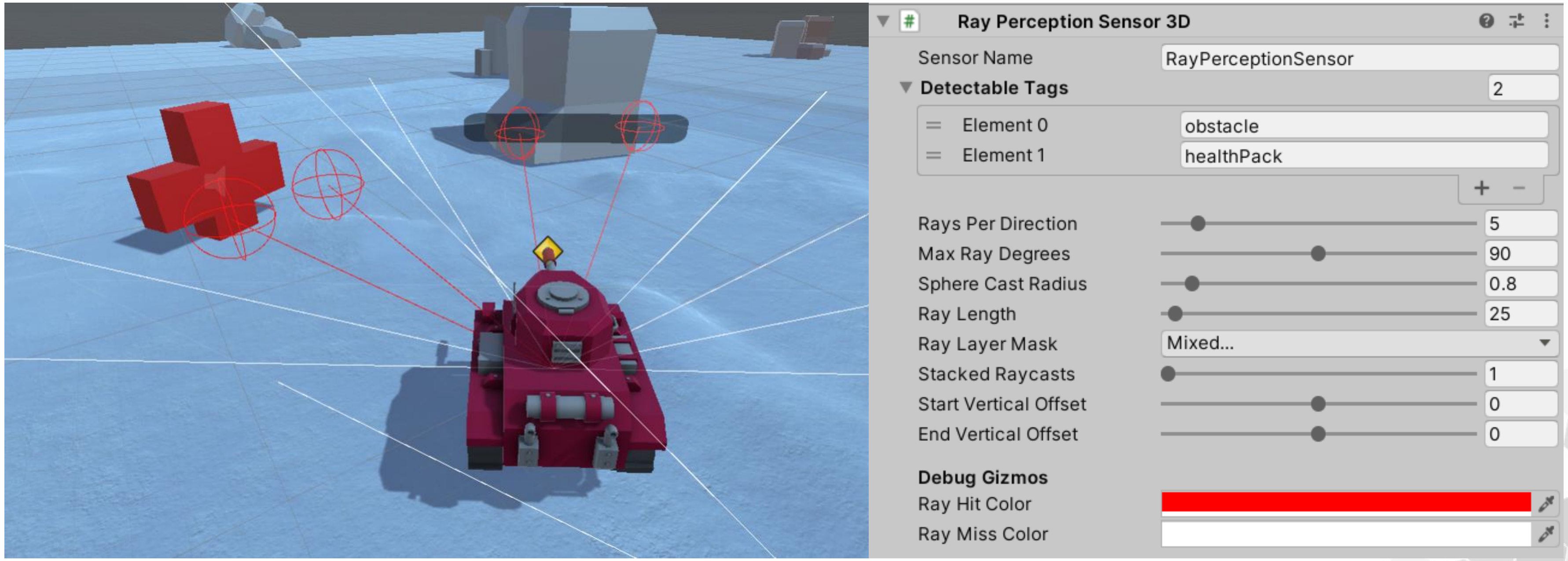
$$normalizedValue = (currentValue - minValue)/(maxValue - minValue)$$

Example

$$currentPos.x = (40 - 0)/(80 - 0) = 0.5$$

Methodology

Enviroment Learning



Ray Perception Sensor

RayPerception Sensor whose total size of:
(Observation Stacks) * (1 + 2 * Rays Per Direction) * (Num Detectable Tags + 2) = 1 * (1 + 2 * 5) * (2 + 2) = 44

Methodology

Enviroment Learning

Name	reward	Description
Shooting accurately	0.1	Each bullet that hits the enemy will get a reward.
Collect a health pack	3	
Collide with obstacle	-1	Collide with walls or rocks.
Turret direction	0.003	Every step if the turret's direction is facing the enemy.
Penalty per step	-0.0001	This penalty is applied every step for making the Agent kill the enemy faster.
Win	2	

Shaped Reward Weights

Methodology

Enviroment Learning

Self-play hyperparameter

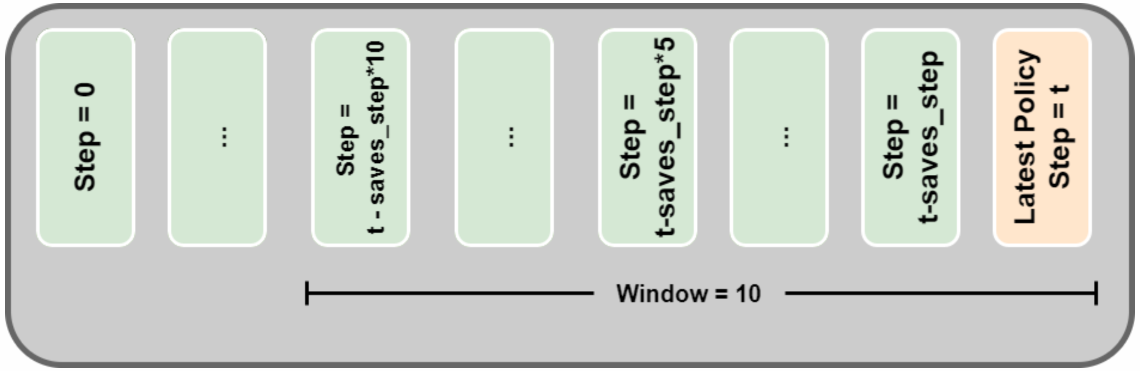
```yaml
self_play:
    save_steps: 50000
    team_change: 200000
    swap_steps: 10000
    window: 10
    play_against_latest_model_ratio: 0.6
```

Methodology

Enviroment Learning

Self-play hyperparameter

```
self_play:
    save_steps: 50000
    team_change: 200000
    swap_steps: 10000
    window: 10
    play_against_latest_model_ratio: 0.6
```



Self-Play Snapshots

- 1 snapshot = 50k step
- Change policy every 10k step
- Opponents are using 10 latest snapshot
- 60% Opponents are latest model.

04

Experiments and Result

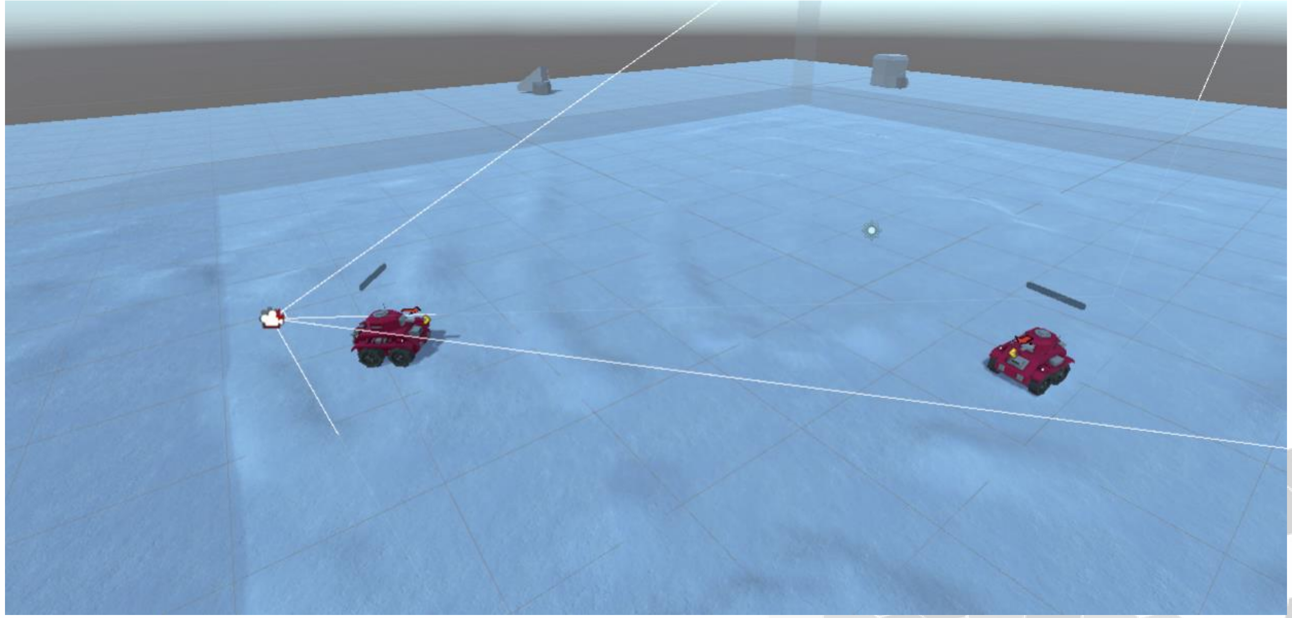
Statistics and Agent’s Behavior

Experiments and Result

Curriculum Learning

Step 0 to 3M:

- No Rocks
- Turret Direction Reward: On
- Opponent: Latest policy



No Obstacle

Experiments and Result

Curriculum Learning

Step 0 to 3M:

- No Rocks
- Turret Direction Reward: On
- Opponent: Latest policy

Above step 3M:

- Adding Obstacle proportionally to the mean reward.
- Turret Direction Reward: Off
- Opponent: 60% Latest policy

Cumulative Reward

Experiments and Result

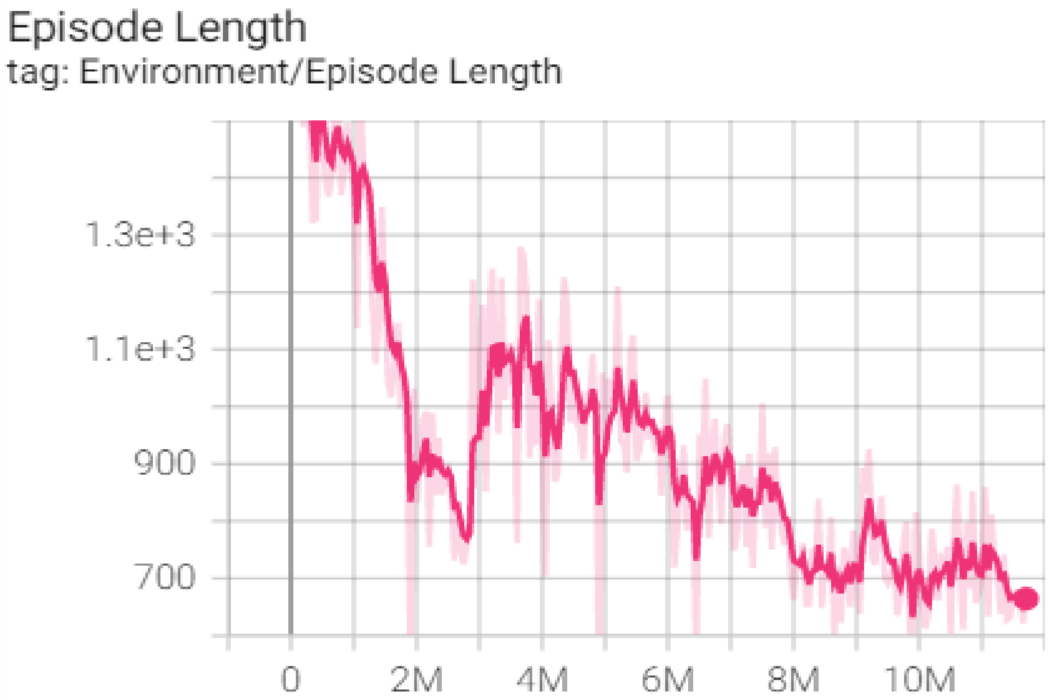
Curriculum Learning

Step 0 to 3M:

- No Rocks
- Turret Direction Reward: On
- Opponent: Latest policy

Above step 3M:

- Adding Obstacle proportionally to the mean reward.
- Turret Direction Reward: Off
- Opponent: 60% Latest policy

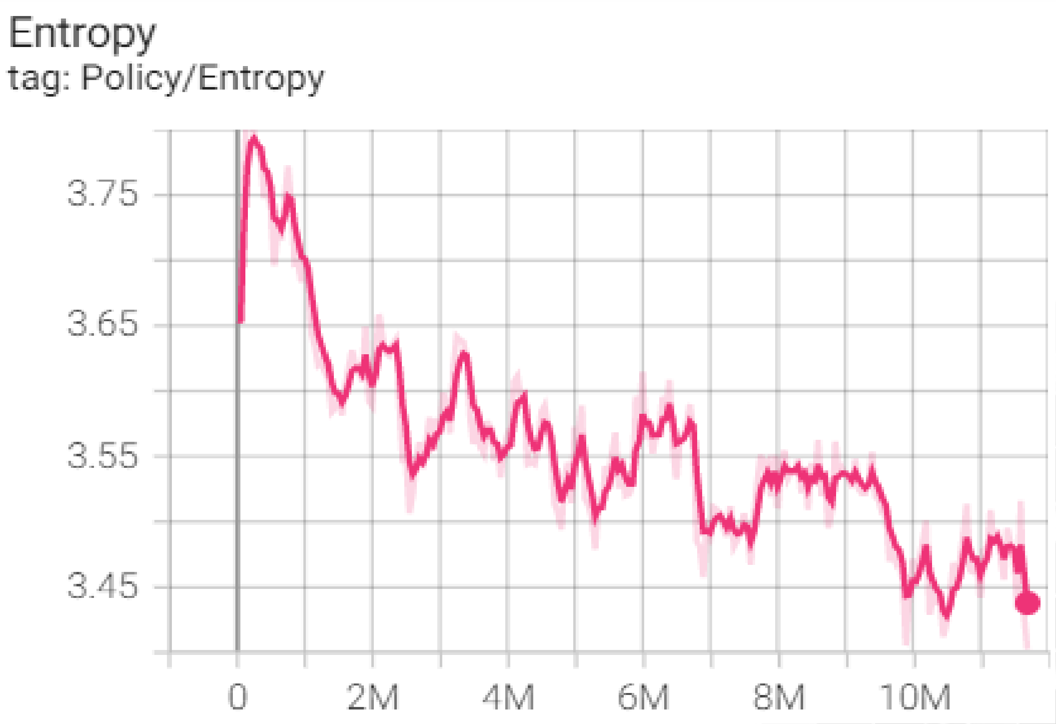


Episode Length

Experiments and Result

Entropy

- How random of the agent's decision
- Decreasing meaning the agent are learning well



Entropy

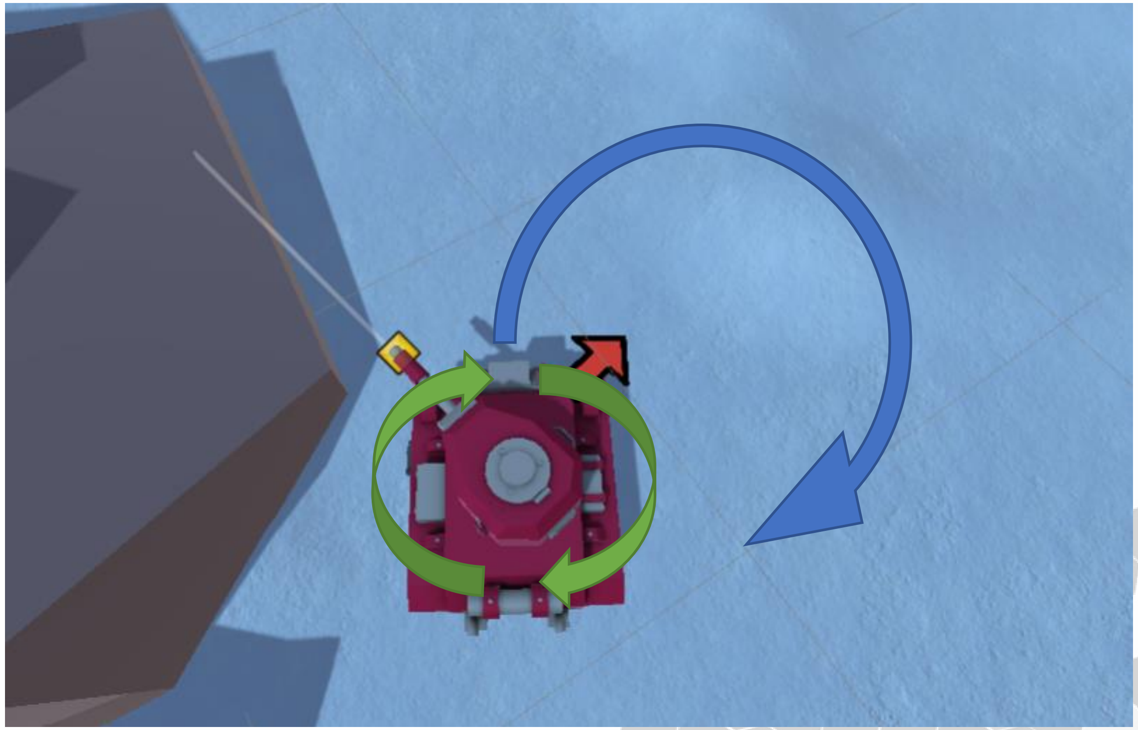
Experiments and Result

Normalize: False

Strange behavior

- Go in a circle
- Turret rotate 1 direction only
- Not shooting

The neuron network somehow converges fast to some weird local minimum.



Strange behavior

Experiments and Result

Live Demo

- Self-play
- Human vs Agent

05

Conclusion and Future Works

Potential and Oriented Development

QS STARS RATED FOR EXCELLENCE 2012

FPT Education

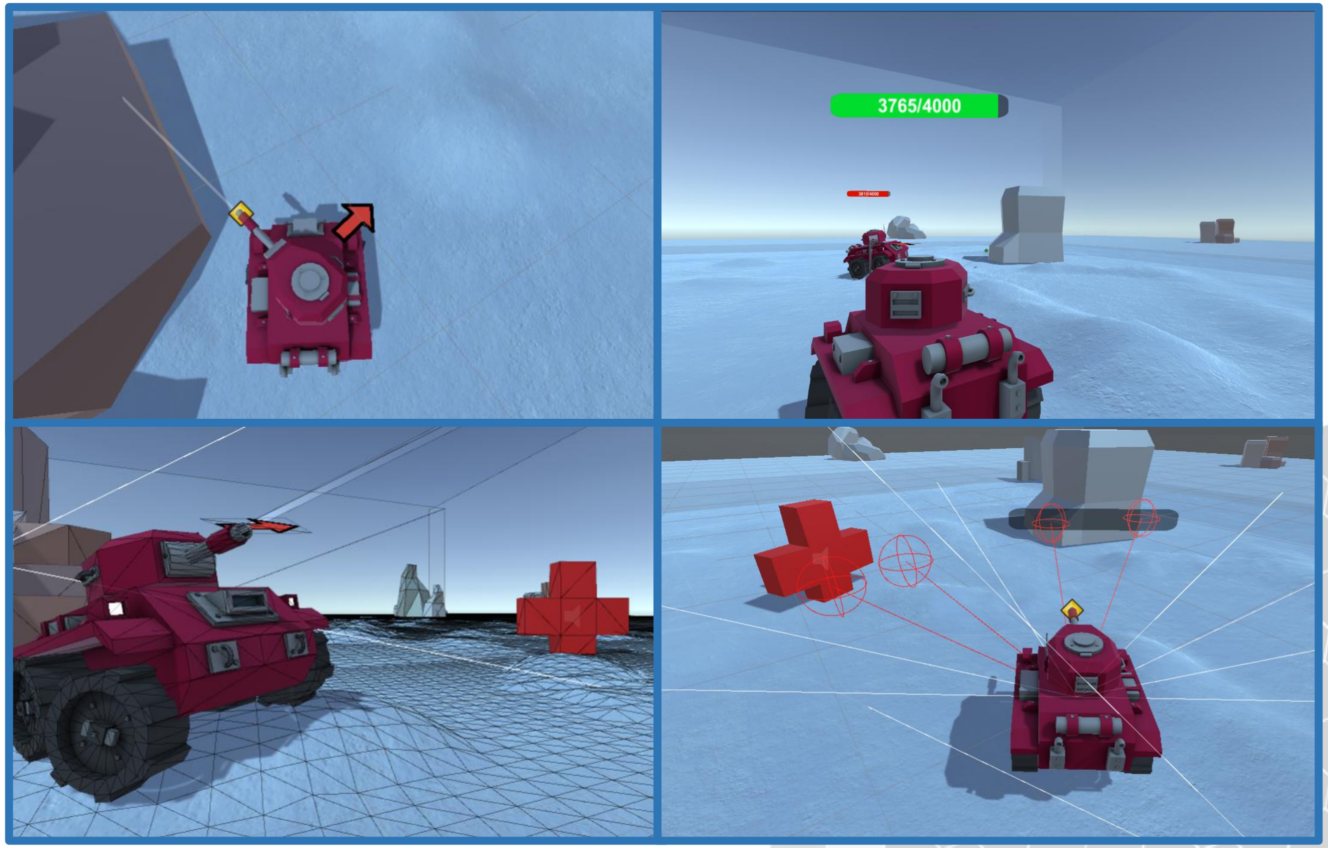
FPT UNIVERSITY

Conclusion and Future Works

Conclusion

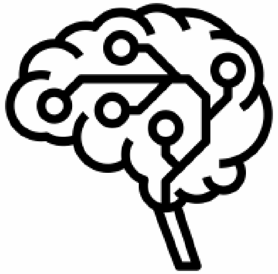
Agent learned the basic task

- Avoid obstacles and walls
- Collect health packs
- Know to shooting and aiming



Conclusion and Future Works

Conclusion



ML-Agents Toolkit

- Non academy, PPO is treated as black box.
- Any user should be able to set and train an agent
- Has potential in casual game in Commercial.

Conclusion and Future Works

Future Works

Visual Observations

Snoopy Pop Environment

Conclusion and Future Works

Future Works

Visual Observations

- Adding camera follow the tank
- Input image from camera as Observations.

Conclusion and Future Works

Future Works

Cooperative Game

- Adding more tanks.
- Allies can heal each other.
- Increased tactical and interactive.

Human observations

Thank you!

Do you have any question?

Contact info:
dungvdhe141196@fpt.edu.vn