SCALARR

Deep Reinforcement Learning (DRL) - a new AI-disrupt

Serhiy Protsenko | Lead data scientist

scalarr.ai

Agenda

1. **About Scalarr**
2. **Reinforcement Learning - development and accomplishments**
3. **Reinforcement Learning - concept and technical details**
4. **Traditional Playtesting vs Scalarr GameAI**
5. **Free Design Partnership with Scalarr**

About Us

Scalarr harnesses the power of AI to create progress and accelerate your revenue

Our AI-driven solutions:

Advertising:
Protection Suite

Gaming:
GameAI

Edge Computing:
AI Edge Labs

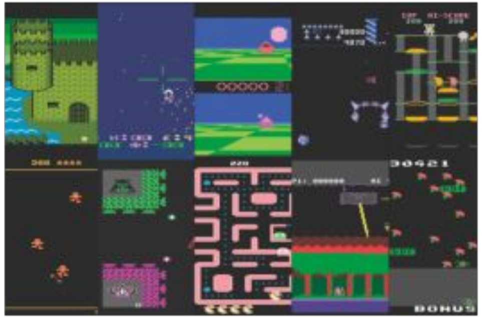
HUUUGE PIXONIC GOOD GAME GAMEHIVE ZiMAD

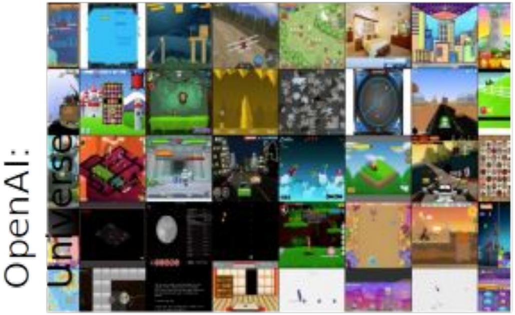
SCALARR

Reinforcement Learning - development and accomplishments

scalarr.ai

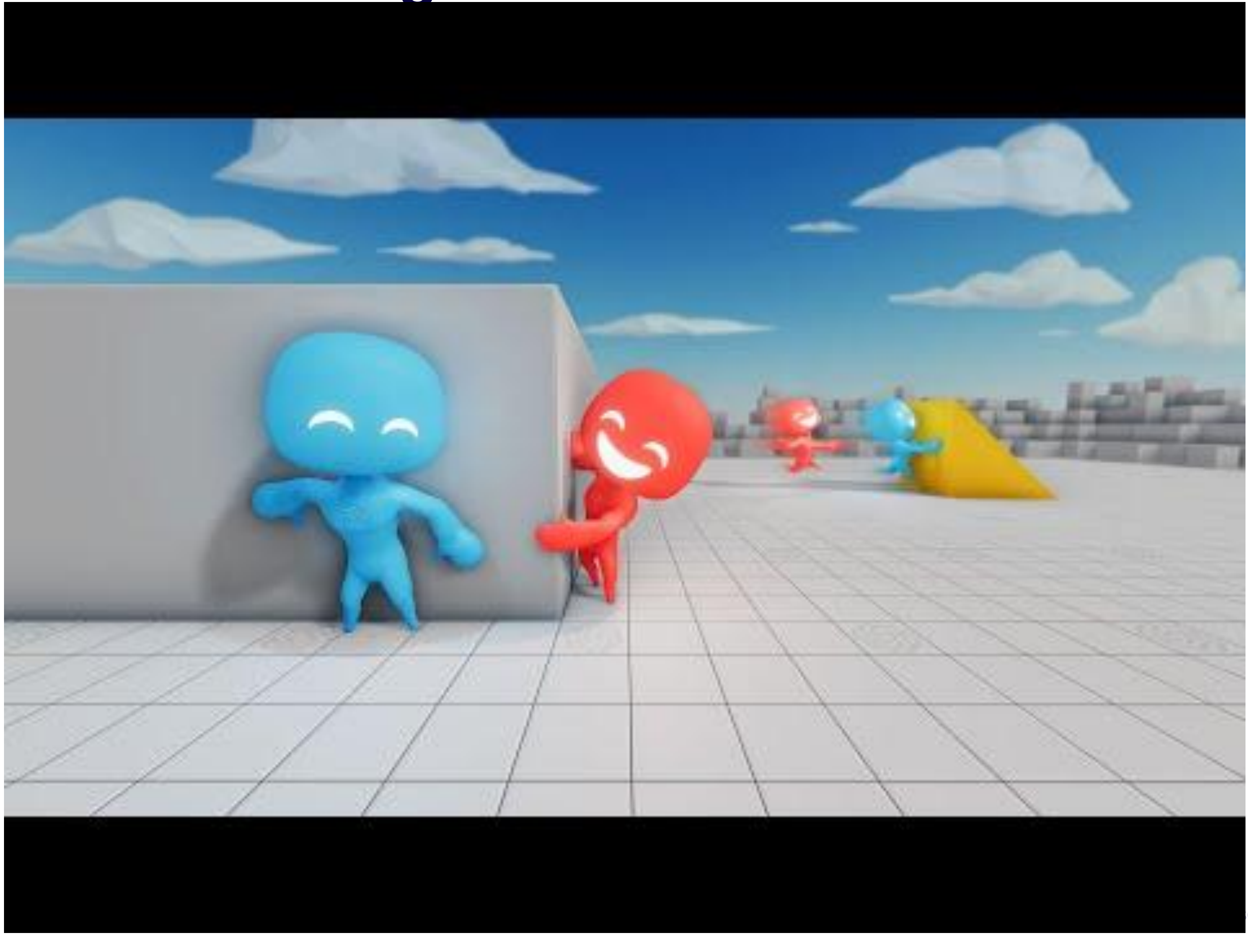
Reinforcement Learning





[illegible]

Reinforcement Learning



Reinforcement Learning

RL - Development and Achievements

2016

AlphaGo

DeepMind

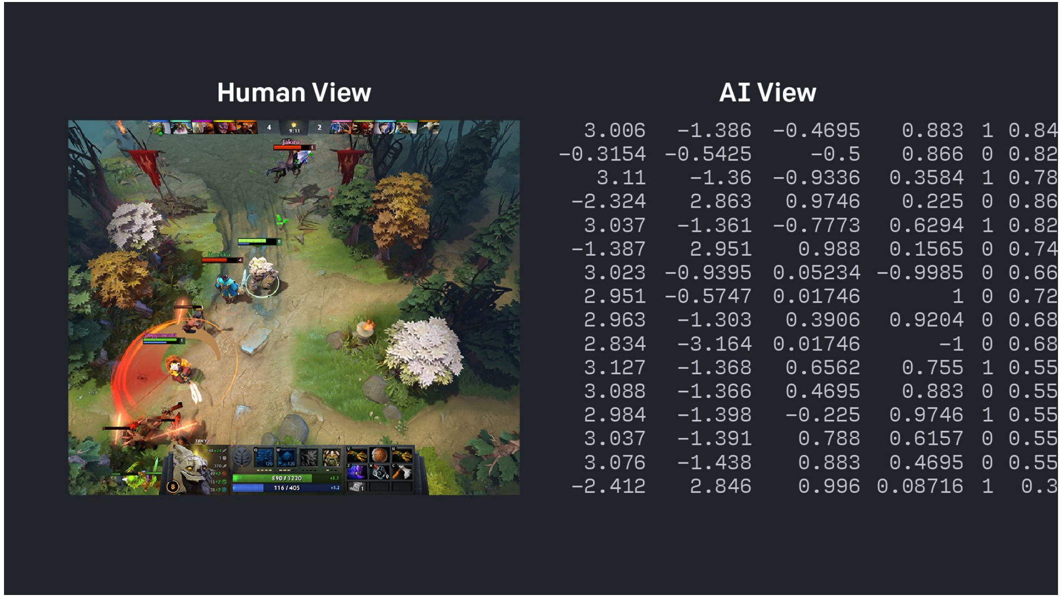
Source: https://deepmind.com/research/case-studies/alphago-the-story-so-far

RL - Development and Achievements

2019

OpenAI Five Defeats Dota 2 World Champions

OpenAI



```
  3.006  -1.386 -0.4695   0.883 1 0.84
-0.3154 -0.5425    -0.5   0.866 0 0.82
   3.11   -1.36 -0.9336  0.3584 1 0.78
 -2.324   2.863  0.9746   0.225 0 0.86
  3.037  -1.361 -0.7773  0.6294 1 0.82
 -1.387   2.951   0.988  0.1565 0 0.74
  3.023 -0.9395 0.05234 -0.9985 0 0.66
  2.951 -0.5747 0.01746       1 0 0.72
  2.963  -1.303  0.3906  0.9204 0 0.68
  2.834  -3.164 0.01746      -1 0 0.68
  3.127  -1.368  0.6562   0.755 1 0.55
  3.088  -1.366  0.4695   0.883 0 0.55
  2.984  -1.398  -0.225  0.9746 1 0.55
  3.037  -1.391   0.788  0.6157 0 0.55
  3.076  -1.438   0.883  0.4695 0 0.55
 -2.412   2.846   0.996 0.08716 1  0.3
```

Source: https://openai.com/blog/openai-five-defeats-dota-2-world-champions/

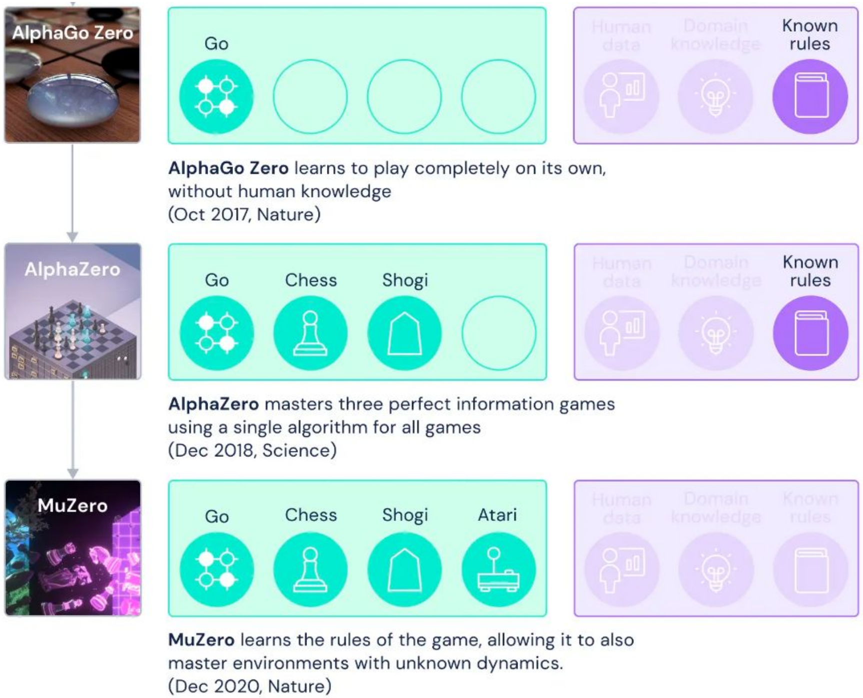
SCALARR

RL - Development and Achievements

2020

MuZero: Mastering Go, chess, shogi and Atari without rules

DeepMind



AlphaGo Zero learns to play completely on its own, without human knowledge
(Oct 2017, Nature)

AlphaZero masters three perfect information games using a single algorithm for all games
(Dec 2018, Science)

MuZero learns the rules of the game, allowing it to also master environments with unknown dynamics.
(Dec 2020, Nature)

Source: https://deepmind.com/blog/article/muzero-mastering-go-chess-shogi-and-atari-without-rules

SCALARR

RL - Development and Achievements

2021

Open-Ended Learning Leads to Generally Capable Agents

DeepMind

Source: https://deepmind.com/research/publications/2021/open-ended-learning-leads-to-generally-capable-agents

SCALARR

Reinforcement Learning - concept and technical details

scalarr.ai

Technical Details, RL Concept

What type of RL can we use?

Offline RL

Hierarchical RL

Off-Policy RL

Inverse RL

On-Policy RL

Model-Free RL

Meta-Learning RL

Model-Based RL

GAIL

With Model RL

Without Model RL

Transfer Learning and Multi-Task Learning RL

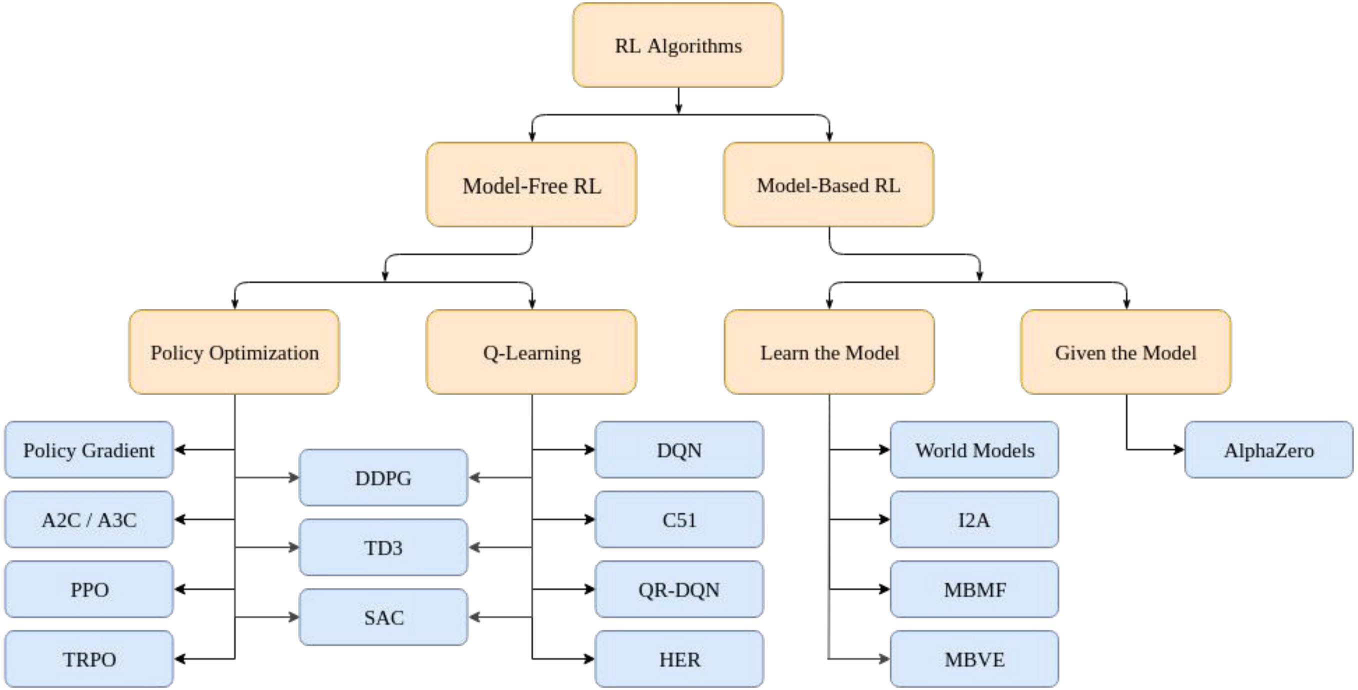
Imitation Learning

Multi-Agent Reinforcement Learning (MARL)

SCALARR

Technical Details, RL Concept

What models are available?

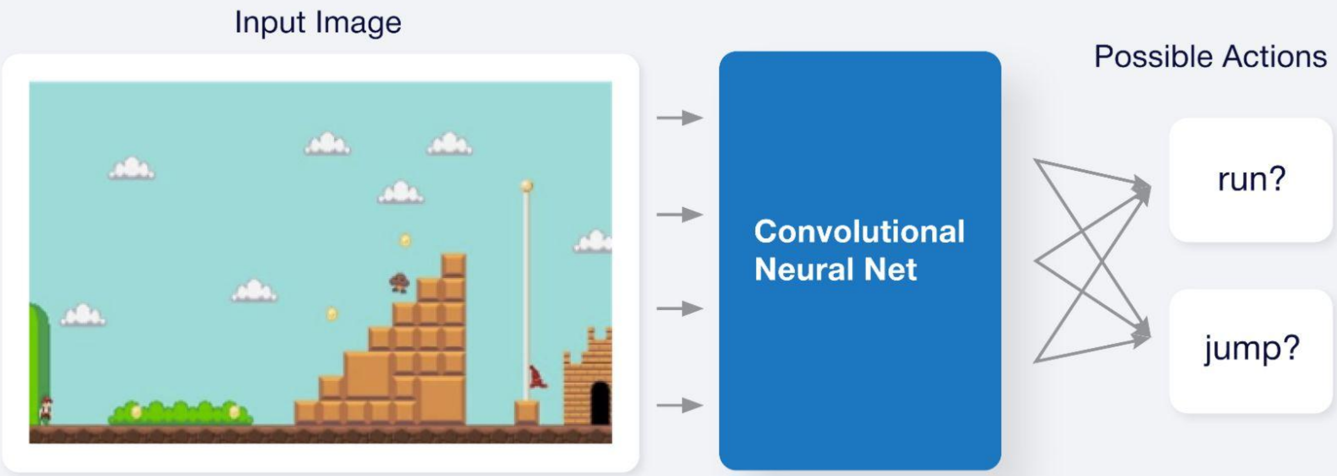


Source: https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html

SCALARR

Technical Details, RL Concept

Convolutional Agent



Input Image

Convolutional Neural Net

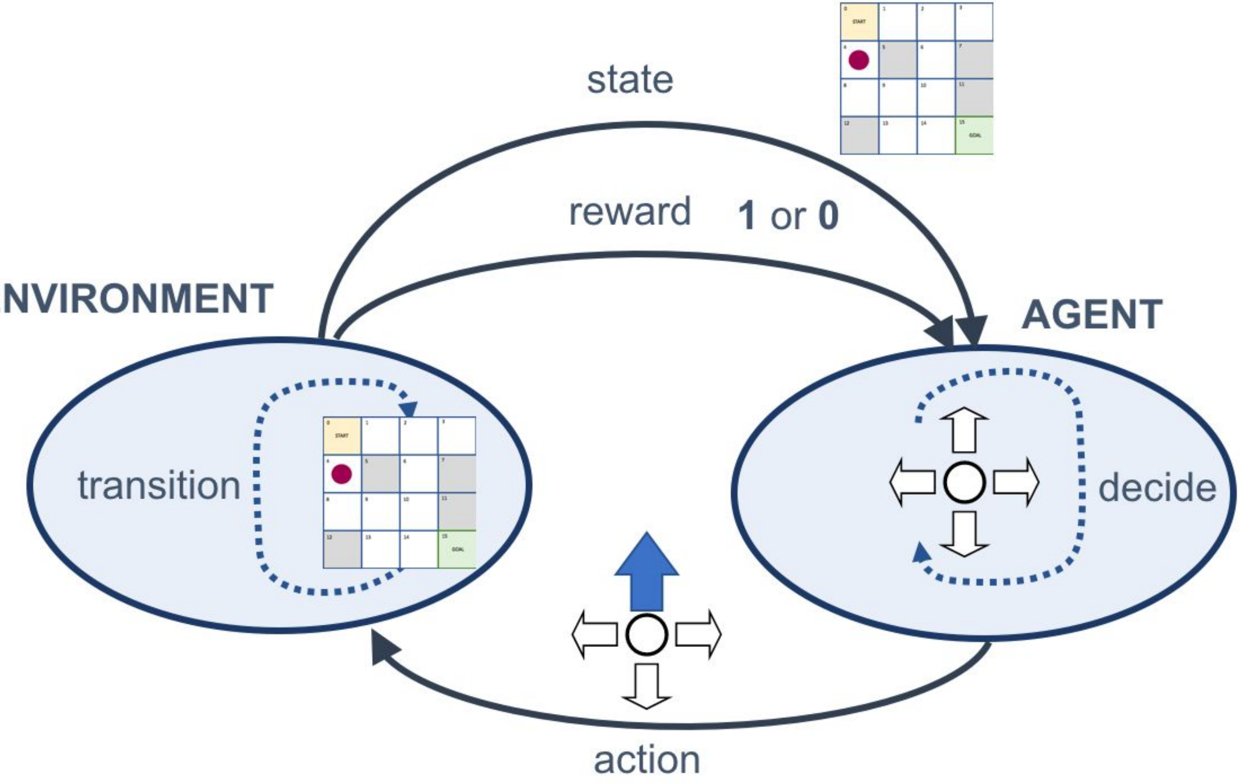
Possible Actions

run?

jump?

SCALARR

Technical Details, RL Concept

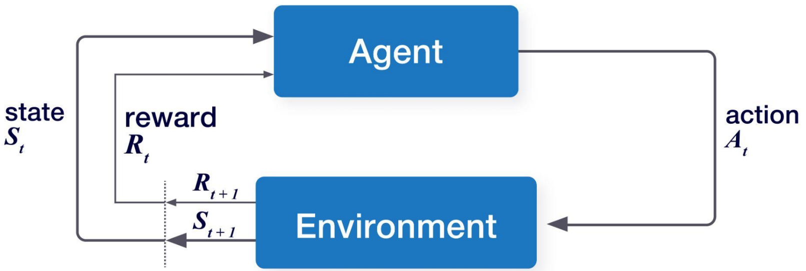


SCALARR

Technical Details, RL Concept

- s_t - state
- a_t - action
- $r_t = r(s_t, a_t)$ - reward
- $R = \sum_t \gamma^t r_t$ - return
- γ - discount factor
- $V(s) = E(R \mid s_0=s)$ - value
- $Q(s, a) = E(R \mid s_0=s, a_0=a)$ - Q-value
- $A(s, a) = Q(s, a) - V(s)$ - advantage
- $\pi(s, a)$ - policy
- $\tau = (s_0, a_0, r_0, s_1, \ldots)$ - trajectory or episode

Interaction loop of the Agent and the Environment

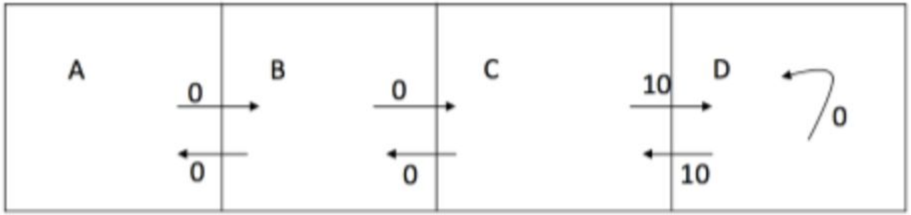


Single-Agent RL: Q-learning (Watkins and Dayan, 1992)

- Learn Q-values in each state
- Use Temporal Difference learning to update Q-values

$$Q^{new}(s_t,a_t) \leftarrow \underbrace{Q(s_t,a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\bigg(\underbrace{\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1},a)}_{\text{estimate of optimal future value}}}_{\text{new value (temporal difference target)}} - \underbrace{Q(s_t,a_t)}_{\text{old value}}\bigg)}^{\text{temporal difference}}$$

- Act by choosing the action that maximizes Q-value, sometimes deviating for exploration

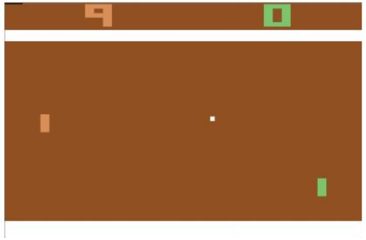


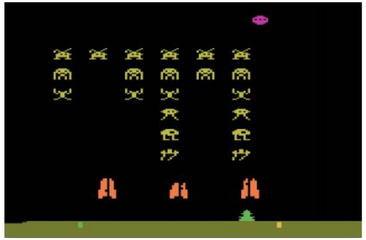
Single-Agent RL: Deep Q-Networks (Mnih et al., 2013)

- *Approximate* Q-values in each state with a *Neural Network*
- Use Temporal Difference learning to update *Networks' weights*

$$L_Q = (r_t + \gamma max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))^2$$

- Store transitions in Experience Replay (ER) buffer to reuse later





Single-Agent RL: Advantage Actor-Critic (Mnih et al., 2016)

- Train two networks: Actor and Critic
- Actor predicts agent's policy **π**(s, a)
- Critic predicts value V(s) to assist Actor's training
- May or may not use Experience Replay
- Train either via Temporal Difference (left) or Monte-Carlo (right)

$$L_{critic} = (r_t + \gamma V_{t+1} - V_t)^2$$

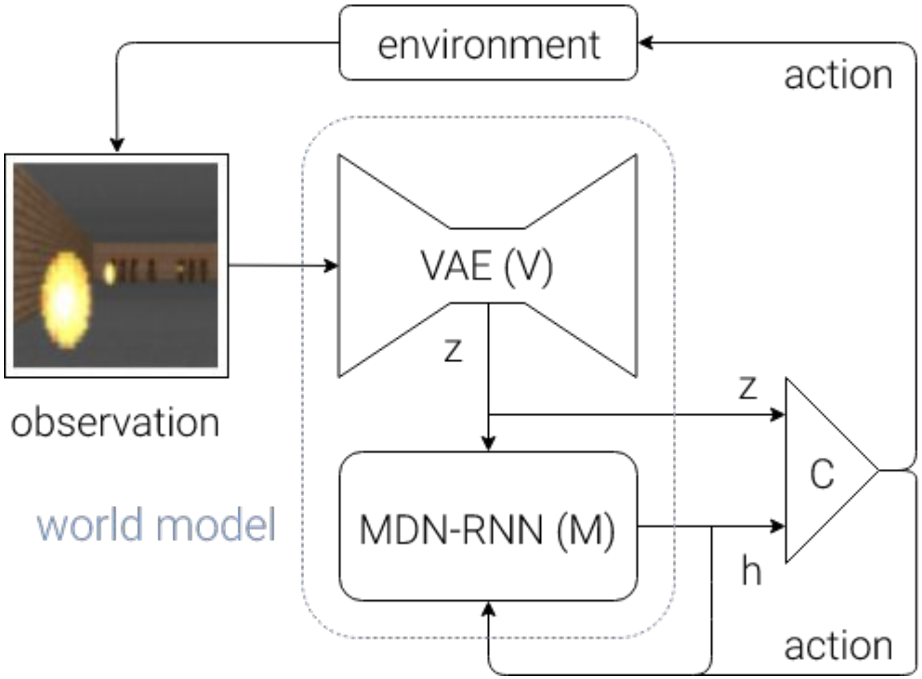
$$L_{actor} = \log \pi(s_t, a_t) * (r_t + \gamma V_{t+1} - V_t)$$

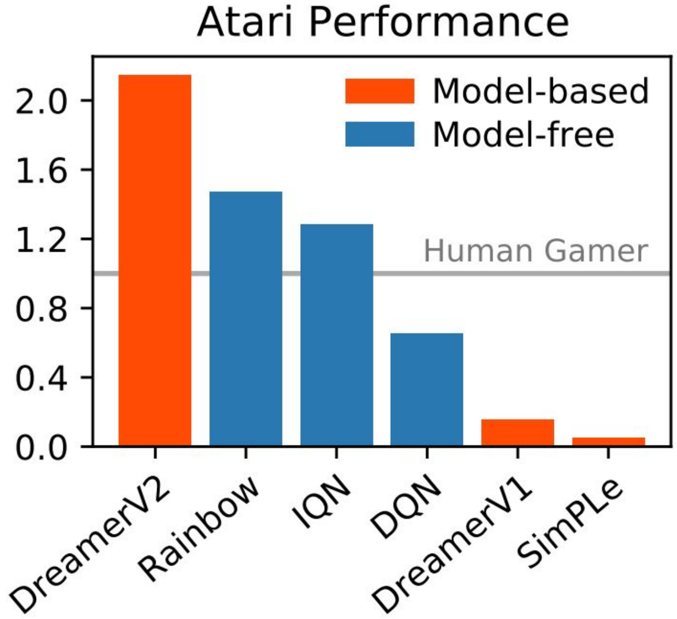
$$L_{critic} = (R_t - V_t)^2$$

$$L_{actor} = \log \pi(s_t, a_t) * (R_t - V_t)$$

Technical Details, RL Concept

World models, Dreamer, Dreamer V2 (DeepMind 2021)

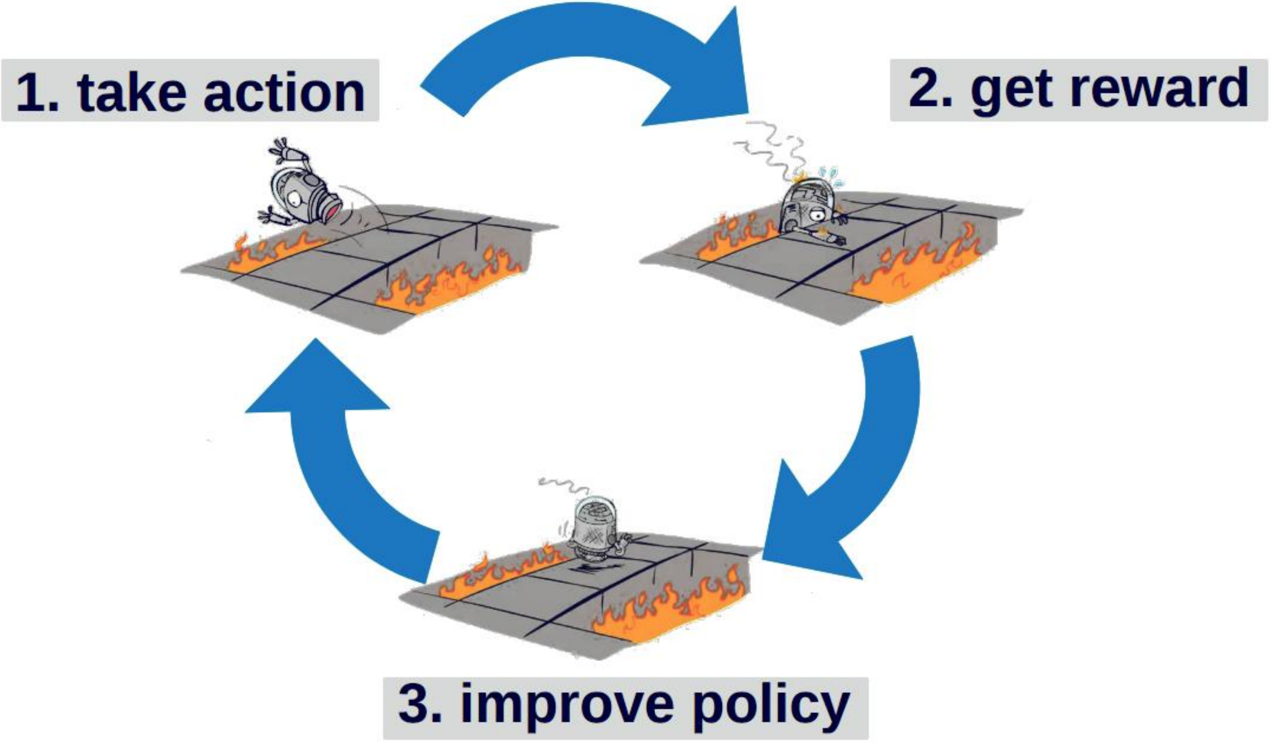




Source: https://ai.googleblog.com/2021/02/mastering-atari-with-discrete-world.html

SCALARR

Reinforcement learning is easy!

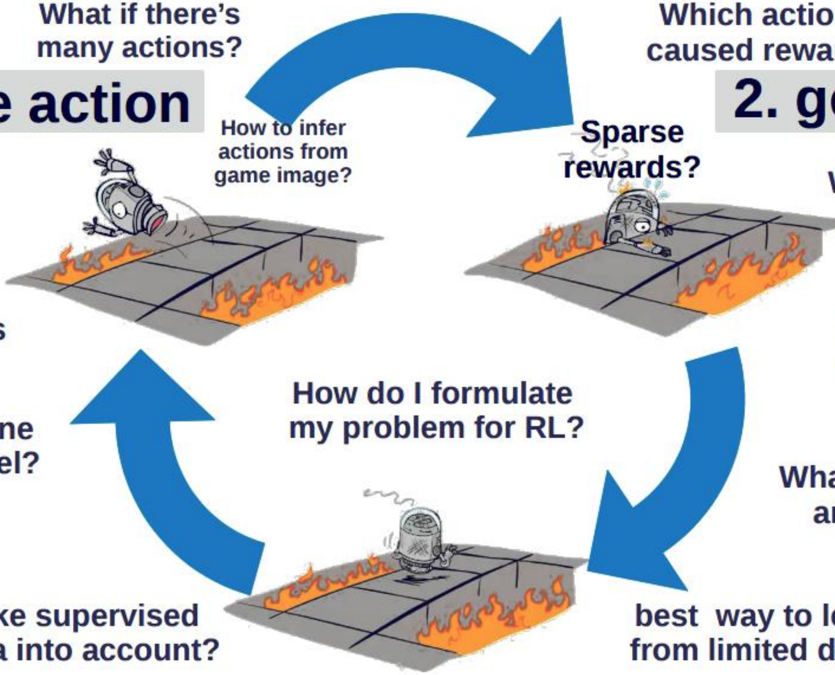


1. take action

2. get reward

3. improve policy

Reinforcement learning is challenging!



How to explore?

What if there's many actions?

1. take action

How to infer actions from game image?

How not to break anyting when acting?

Continuous actions?

How to define policy model?

When not to use RL?

take supervised data into account?

did my algorithm overfit to simulation?

Which actions caused reward?

Can agent cheat reward

2. get reward

Sparse rewards?

Which rewards are easier to learn?

What if there are multiple agents?

What if observations are incomplete?

How do I formulate my problem for RL?

3. improve policy

best way to learn from limited data?

How do I know if it converged or not?

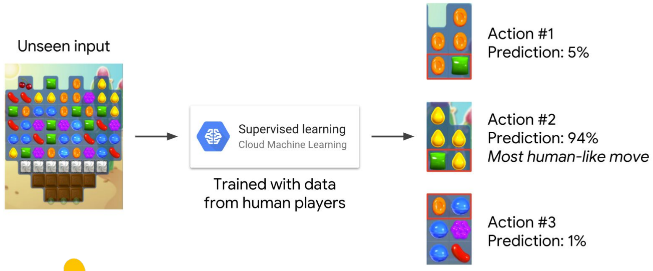
SCALARR

SCALARR

Traditional Playtesting vs Scalarr GameAI

scalarr.ai

World Game Industry Leaders and RL



Unseen input

Supervised learning
Cloud Machine Learning

Trained with data
from human players

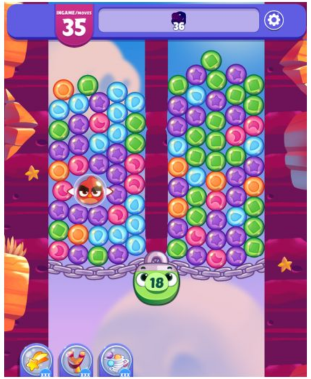
Action #1
Prediction: 5%

Action #2
Prediction: 94%
Most human-like move

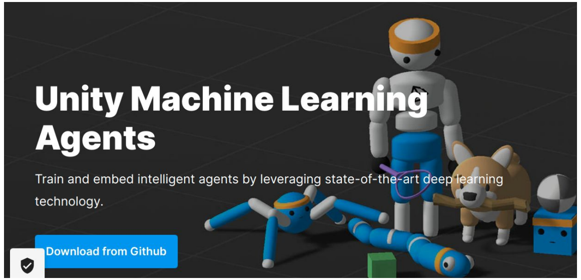
Action #3
Prediction: 1%

King

ROVIO



unity

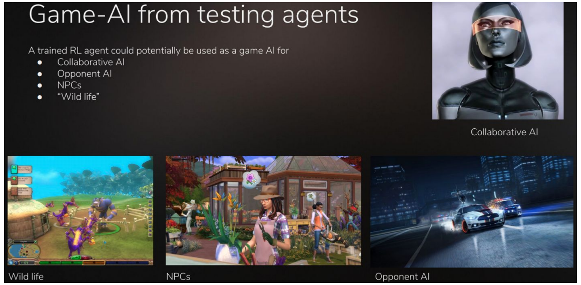


Unity Machine Learning Agents

Train and embed intelligent agents by leveraging state-of-the-art deep learning technology.

Download from Github

EA



Game-AI from testing agents

A trained RL agent could potentially be used as a game AI for
- Collaborative AI
- Opponent AI
- NPCs
- "Wild life"

Collaborative AI

Wild life

NPCs

Opponent AI

Game Testing & Level Design Challenges

☒ Gamers constantly demand new content (new levels)

For Match-3 games, it's nearly 100 new levels every two weeks!

☒ Game/Level designers spend *65%+* of their work time on playtesting & balancing,

☒ Current method of testing is time-consuming & inaccurate

In most cases the designers themselves are testing the game

☒ Badly balanced game levels negatively impact ***Retention*** and ***Monetization***

SCALARR

SCALARR

WHY REINFORCEMENT LEARNING?

Accuracy

Remove the biases that come with human testing & eliminate the need to test your levels internally

Speed

Drastically cut the testing period. Our AI can test a game/level in a couple hours compared to weeks

Balance

Balance your game/level using Game AI's outputs. i.e. Fail / Win rate, # of attempts before winning, etc.

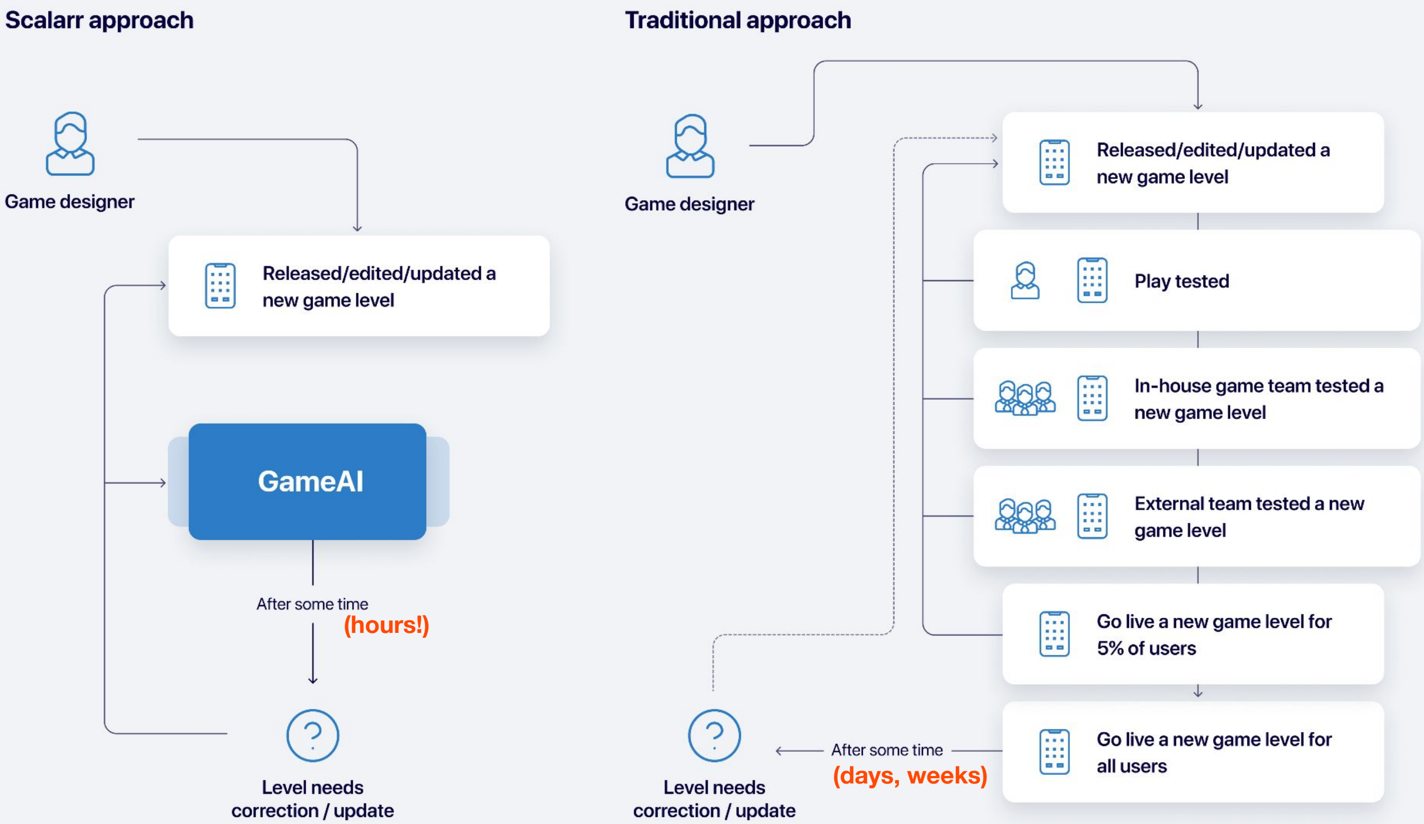
Engage

A properly balanced game will keep your users engaged and excited!

Monetize

More engaged users that feel challenged will spend more time and money $$$

What is GameAI?



Scalarr approach

Game designer

Released/edited/updated a new game level

GameAI

After some time (hours!)

Level needs correction / update

Traditional approach

Game designer

Released/edited/updated a new game level

Play tested

In-house game team tested a new game level

External team tested a new game level

Go live a new game level for 5% of users

Go live a new game level for all users

After some time (days, weeks)

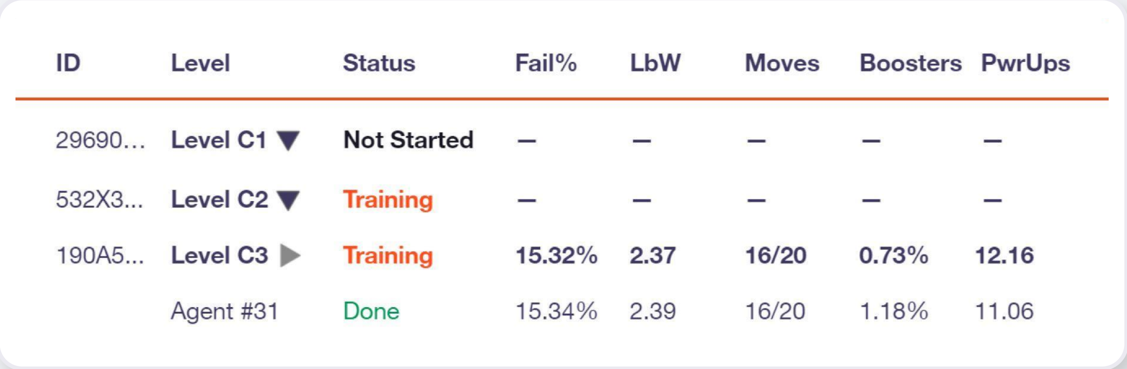
Level needs correction / update

SCALARR

RL Avatars playtesting - output data *(Match-3 Example)*

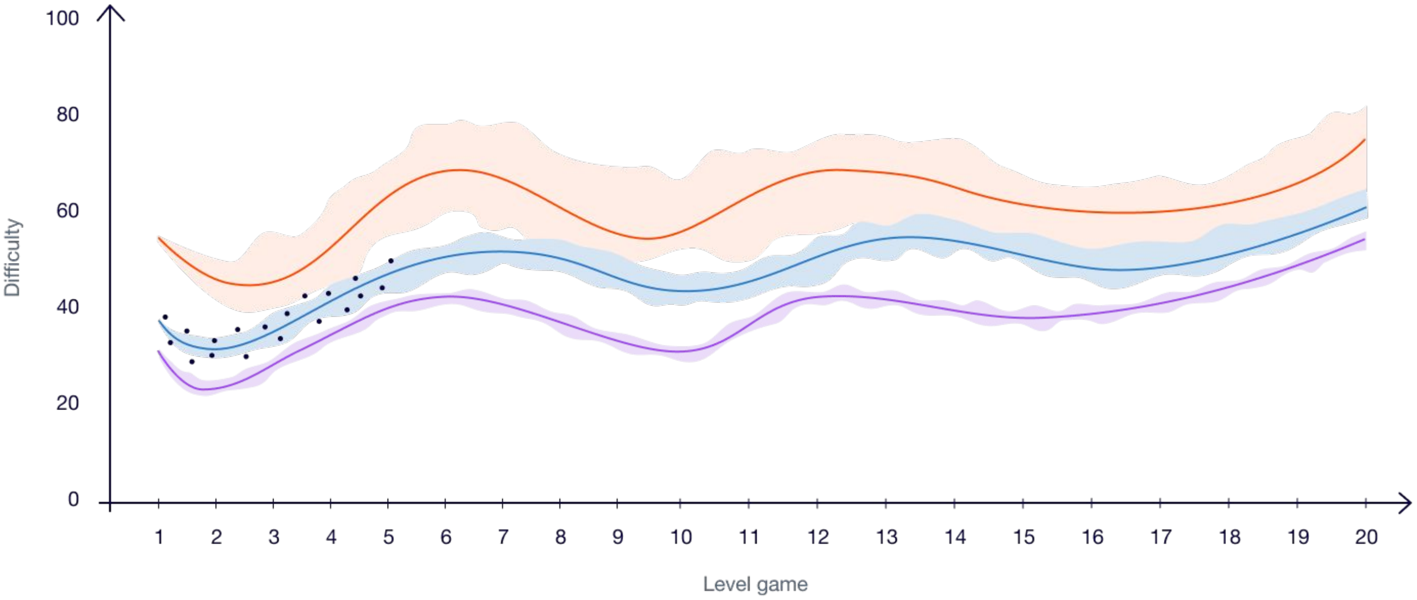
Data available for newly loaded levels and every RL Avatar (total and average):

- **Fail Rate / Win Rate**
- **# of attempts lost before winning**
- **Scatter of the results around the avg, depending on random factors**
- **# Moves spent / left**
- **Boosters & power-ups use rate**
- **# of collected / used field bombs**
- **Other custom metrics**



ID	Level	Status	Fail%	LbW	Moves	Boosters	PwrUps
29690...	Level C1 ▼	Not Started	—	—	—	—	—
532X3...	Level C2 ▼	Training	—	—	—	—	—
190A5...	Level C3 ▶	Training	15.32%	2.37	16/20	0.73%	12.16
	Agent #31	Done	15.34%	2.39	16/20	1.18%	11.06

Difficulty Assessment



🟧 **Newbie agent** - RL agent 10M step

🟦 **Expert agent** - RL agent 20M step

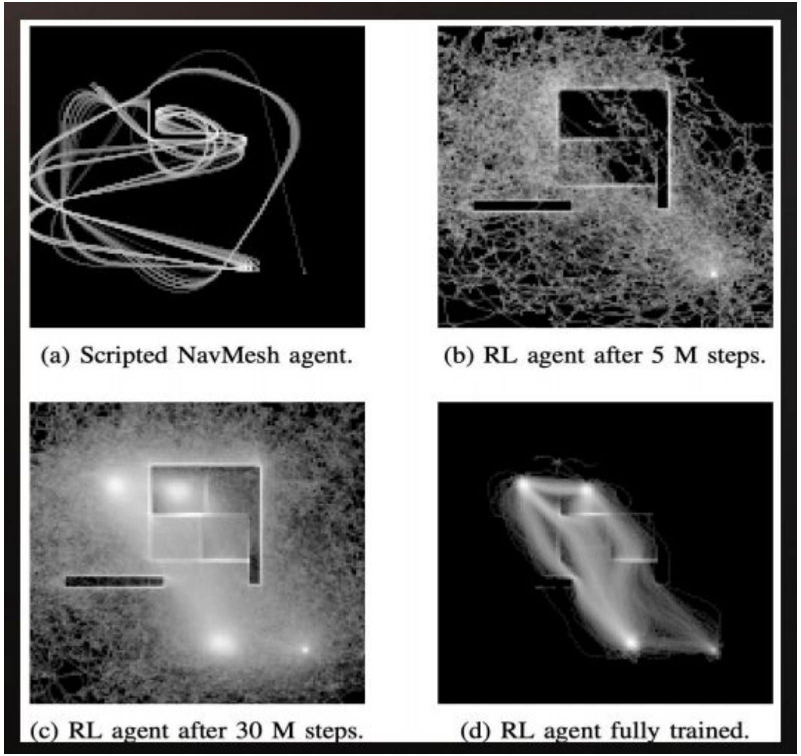
🟪 **Skilled agent** - RL agent 30M+ step

● **Real gamers**

SCALARR

RL - What Components are Needed?

- Integration components (SDK, etc.) & infrastructures;
- RL Agents (set of models);
- Analytical tool;
- etc.



(a) Scripted NavMesh agent.

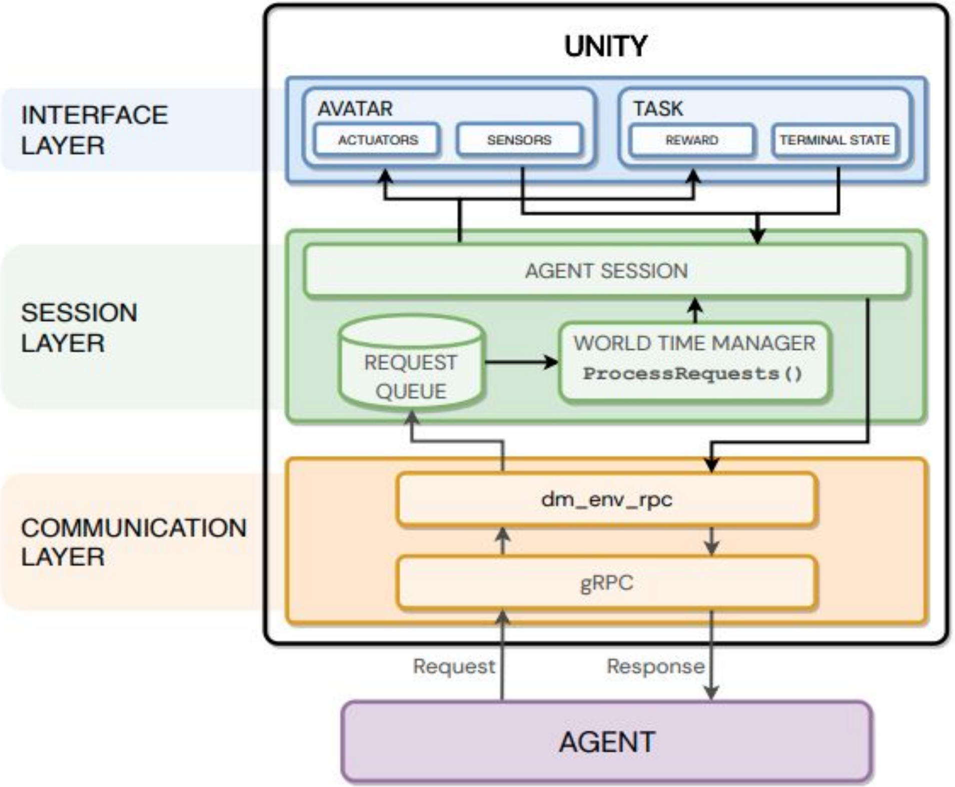
(b) RL agent after 5 M steps.

(c) RL agent after 30 M steps.

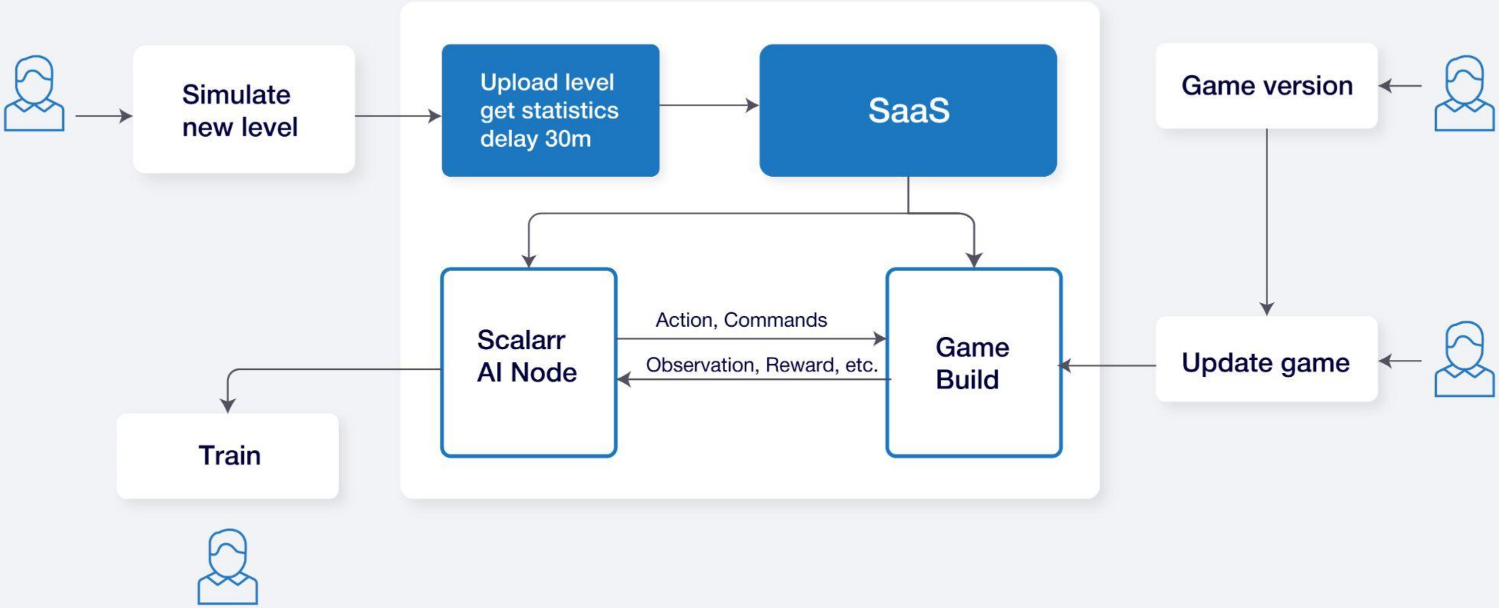
(d) RL agent fully trained.

Integration - Free of charge

- SDK aka Unity Plugin
- 7 days for full integration (approximately)

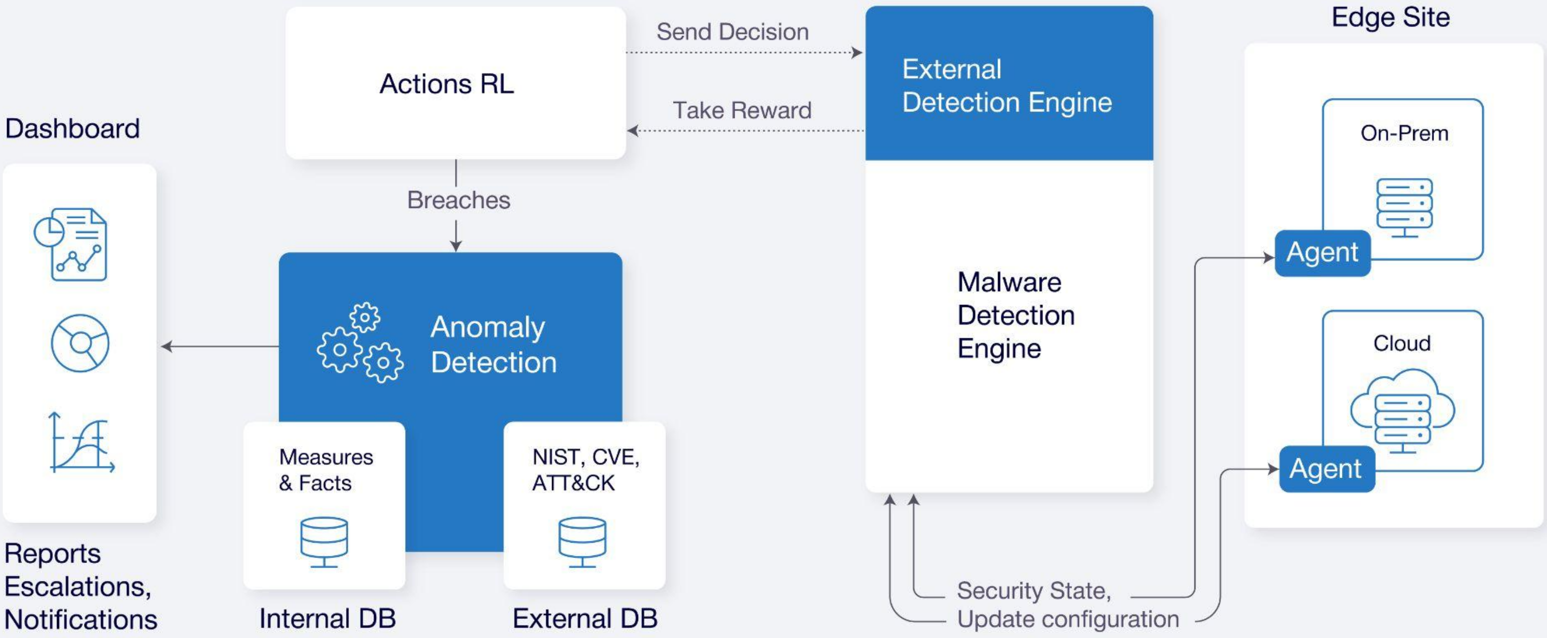


SaaS


Simulate new level
Upload level get statistics delay 30m
SaaS
Game version
Scalarr AI Node
Action, Commands
Observation, Reward, etc.
Game Build
Update game
Train

Demo - RL Avatar plays Match-3 game like a Human

Scalarr Edge Labs AI



SCALARR

SCALARR

Contact us today to progress tomorrow

Serhiy Protsenko | Lead data scientist

scalarr.ai serhiy.protsenko@scalarr.io