

강화학습 환경 개발

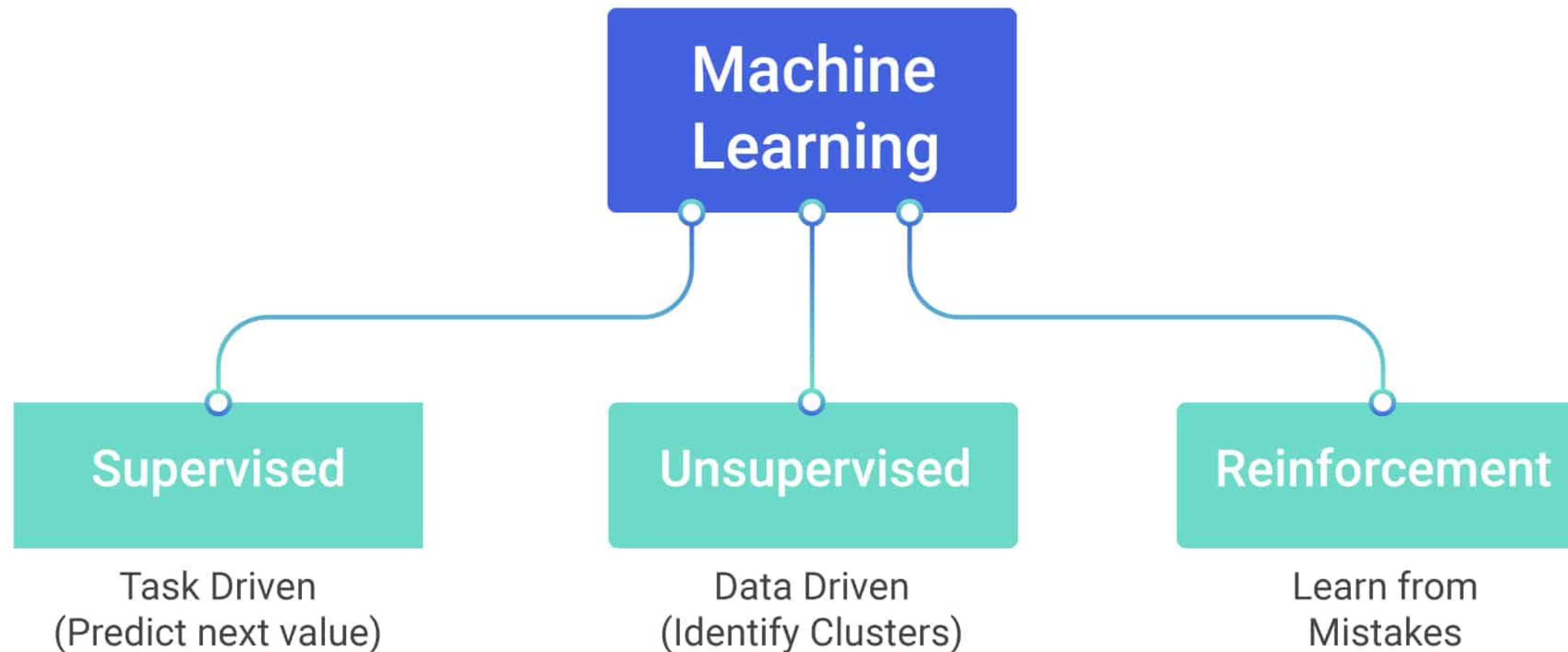
Reinforcement Learning Environment Development

옥찬호 (Chris Ohk)
utilForever@gmail.com

- 세미나의 발표 자료와 예제 코드는 다음 저장소에서 확인 가능합니다.
<https://github.com/utilForever/2021-AIFrenz-RLEnv>
- 강화학습 환경 개발에 관심이 있다면 다음 프로젝트를 참고하세요.
 - 하스스톤 : <https://github.com/utilForever/RosettaStone>
 - Baba Is You : <https://github.com/utilForever/baba-is-auto>
 - 레전드 오브 룬테라 : <https://github.com/utilForever/Runeterra>
 - 포켓몬 배틀 : <https://github.com/utilForever/PokeMaster>
 - Unrailed! : <https://github.com/utilForever/Corailed>

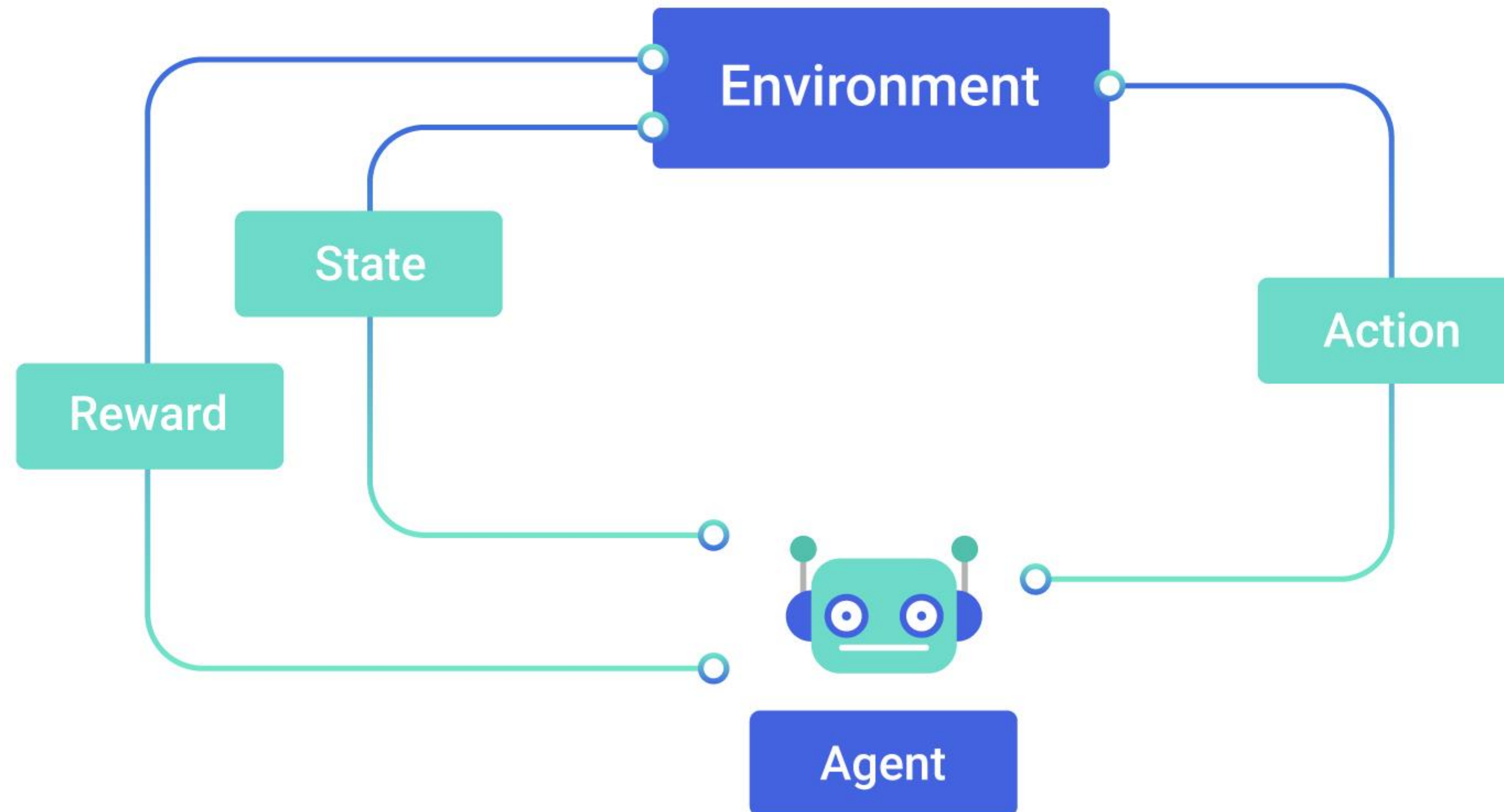
Reinforcement Learning

AI Frenz Seminar
RL Environment Development



Reinforcement Learning

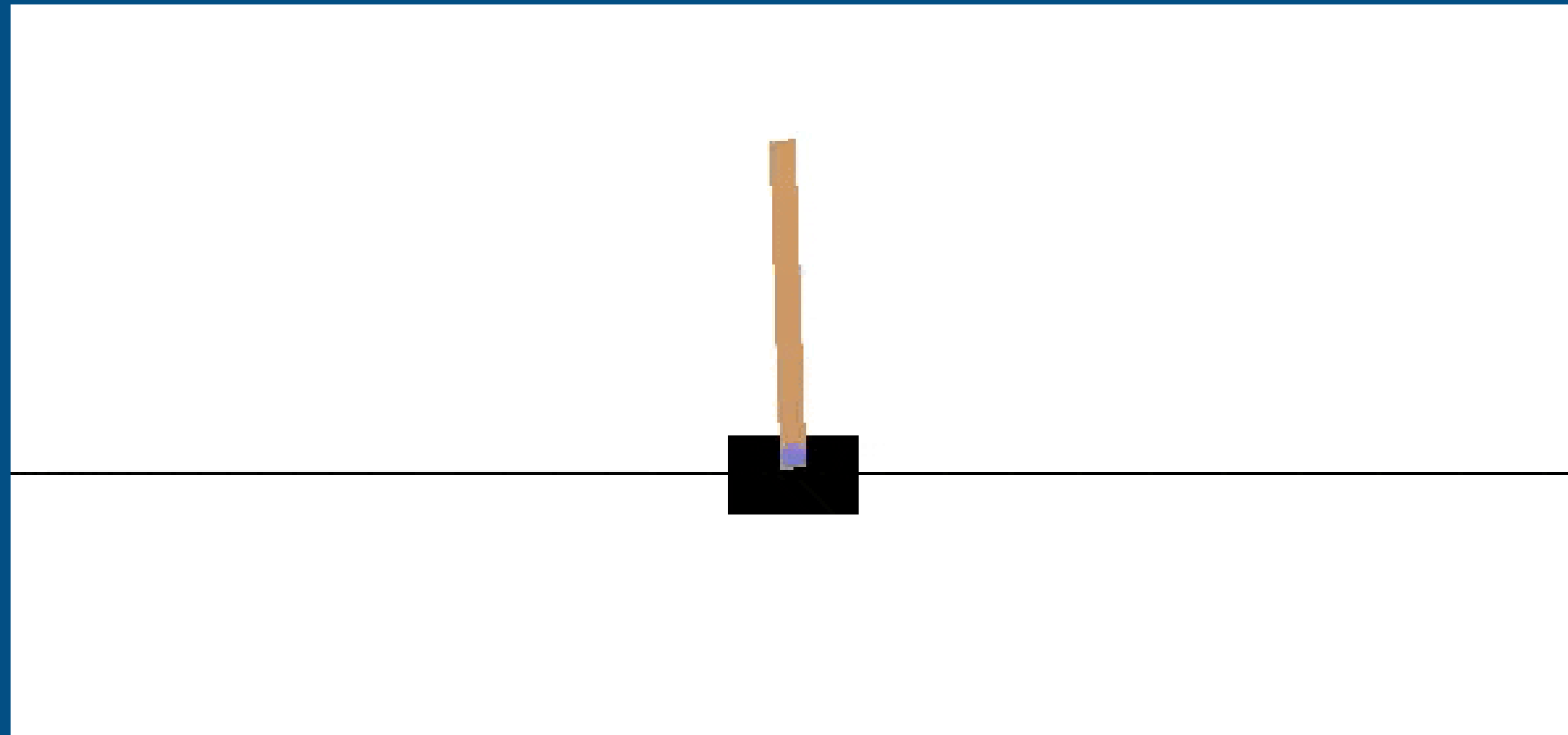
AI Frenz Seminar
RL Environment Development



RL Environments

AI Frenz Seminar
RL Environment Development

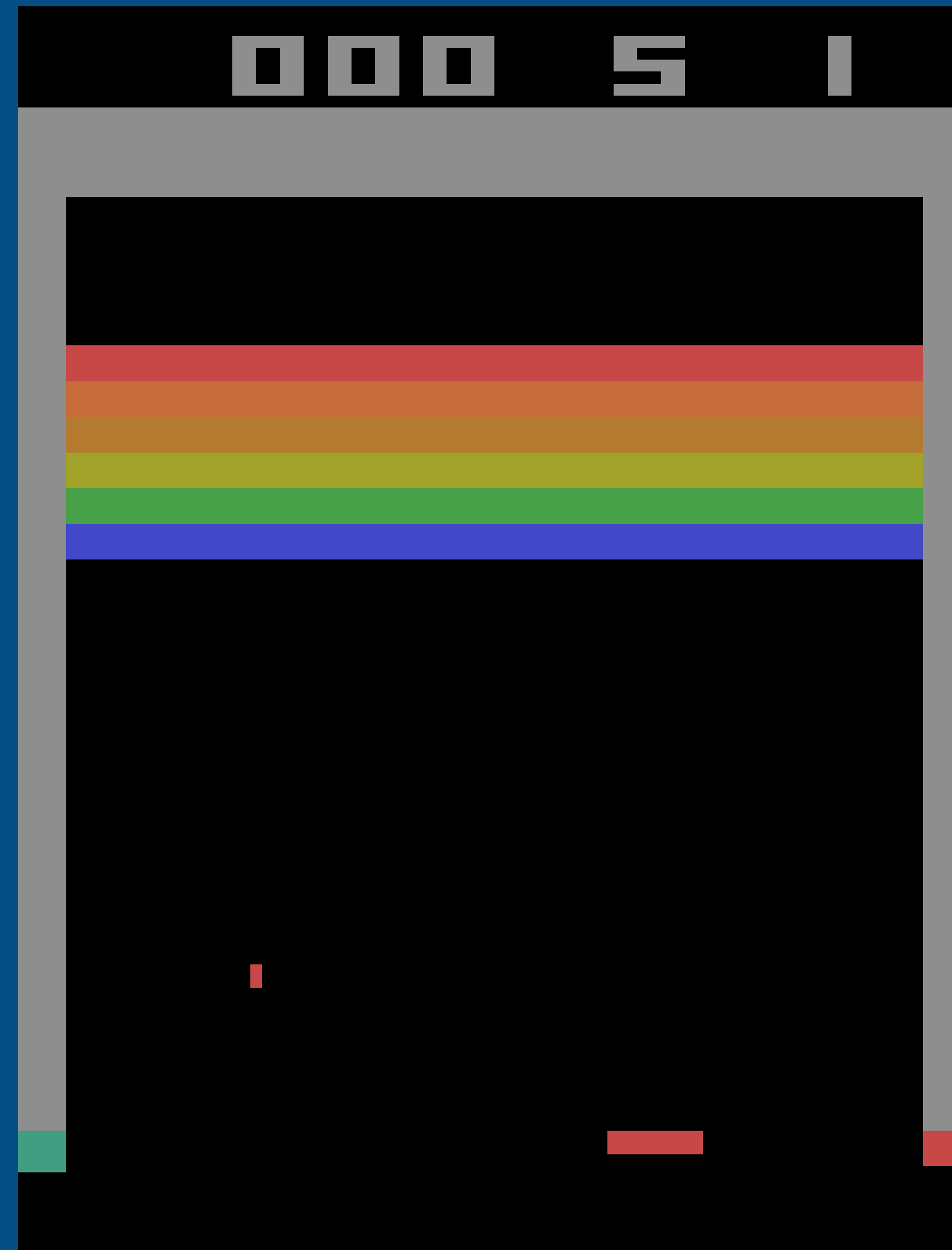
- Cart-Pole



RL Environments

AI Frenz Seminar
RL Environment Development

- Breakout



RL Environments

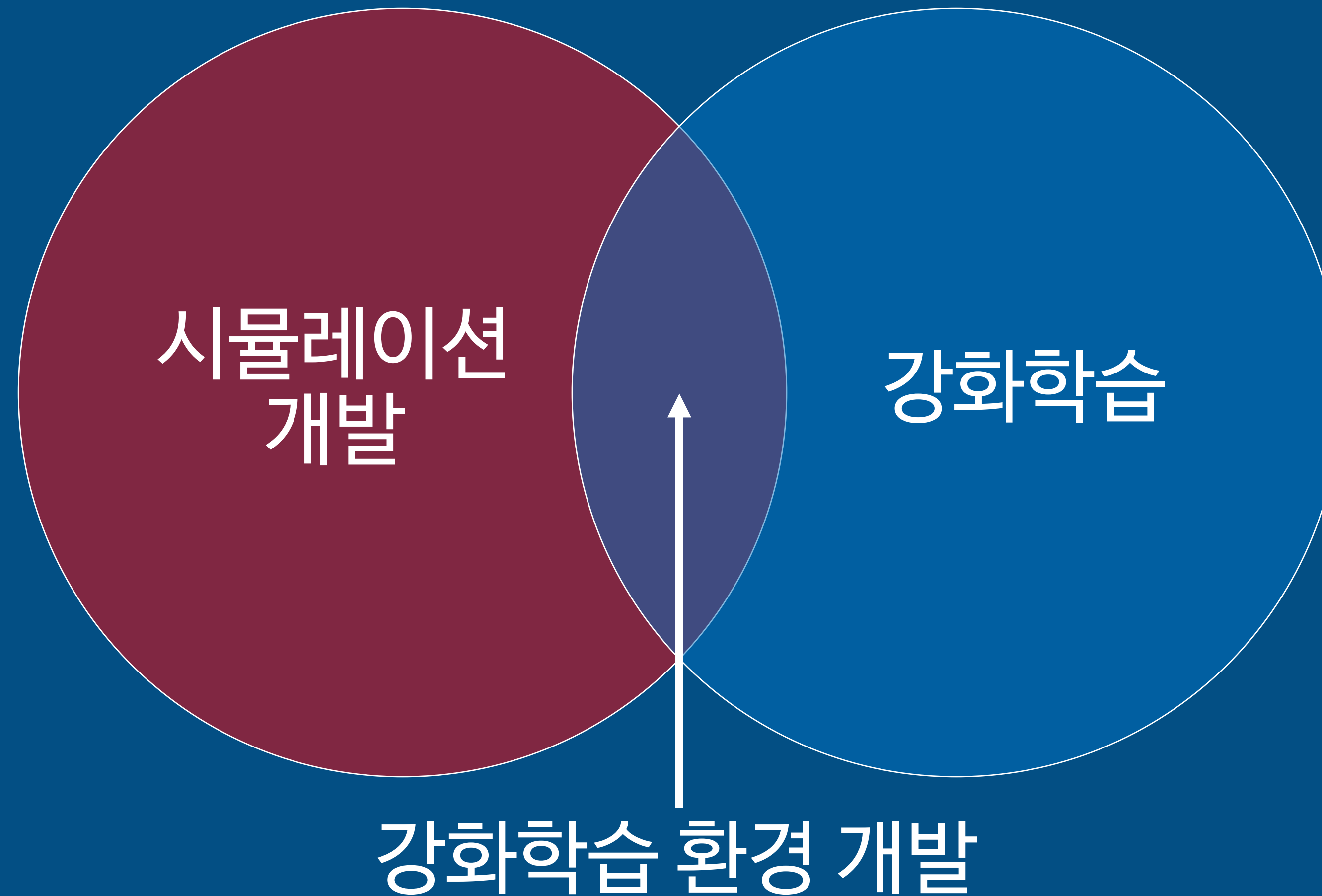
AI Frenz Seminar
RL Environment Development

- StarCraft 2

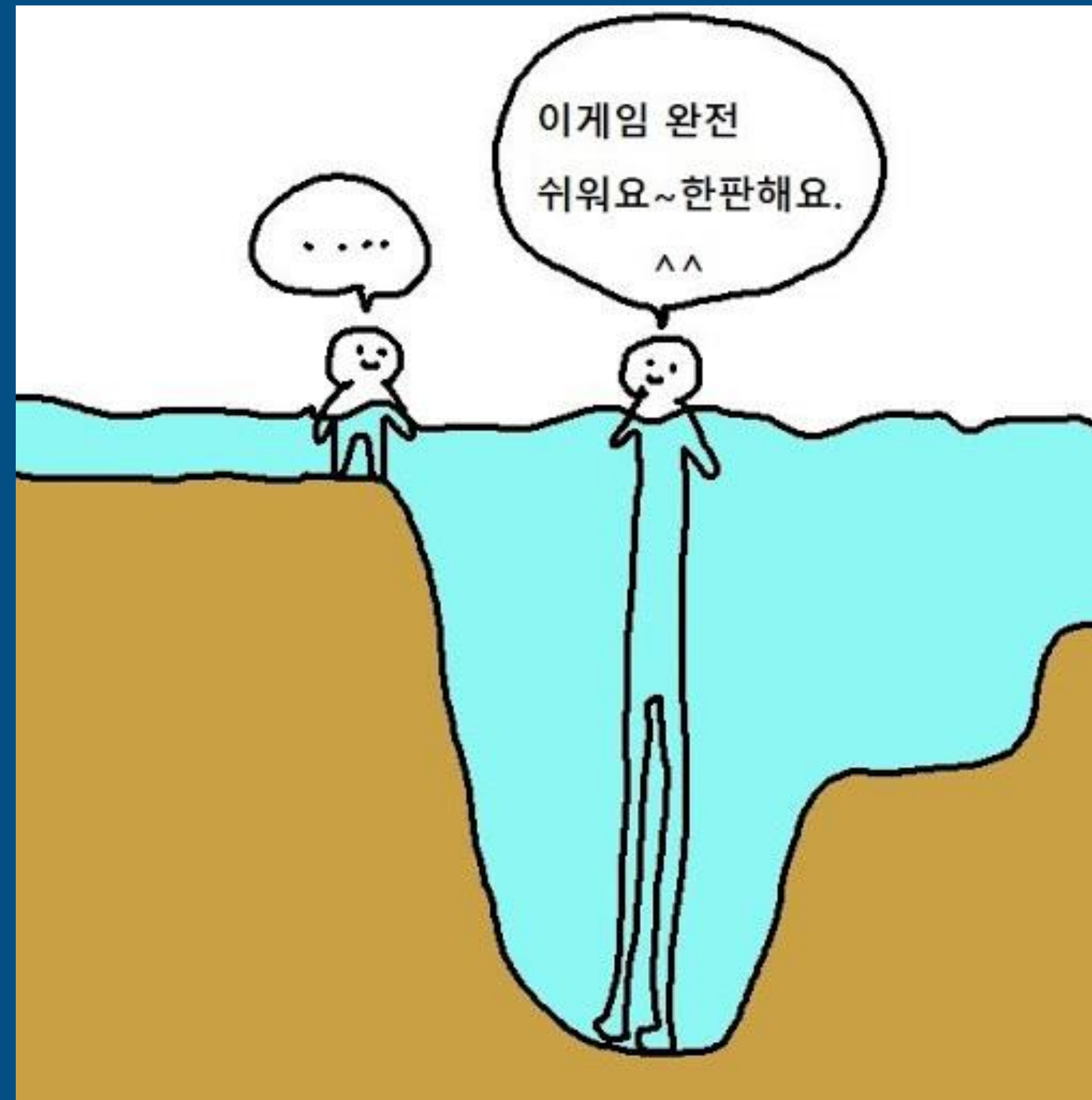


- 시간이 지나면서 강화학습 환경이 다양해지고 있다.
 - 강화학습 환경 목록 : <https://github.com/clvrai/awesome-rl-envs>
- 많은 강화학습 환경들이 갖고 있는 한계
 - 규칙이 단순하다.
 - 할 수 있는 행동의 종류가 적다.
 - 결정적이다. (Deterministic)
 - 모든 상태를 관찰할 수 있다. (Fully-observable)

- 강화학습 환경 개발이 어려운 이유



- 강화학습 환경 개발이 어려운 이유



Make RL Environment!

AI Frenz Seminar
RL Environment Development

- 강화학습 환경을 만들 때 주의할 점
 - 어떤 환경을 만들지 생각해 보기
 - 로보틱스
 - 게임
 - 네비게이션
 - 멀티 에이전트
 - 자율주행
 - 물리 시뮬레이터
 - ...

Make RL Environment!

- 강화학습 환경을 만들 때 주의할 점
 - MDP를 잘 생각해 보기

Markov Decision Process (MDP)

$$M = \{\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma\}$$

\mathcal{S} : The state space

\mathcal{A} : The action space

\mathcal{P} : The transition kernel

R : The bounded reward function

$\gamma \in [0, 1)$: The discount factor

Make RL Environment!

AI Frenz Seminar
RL Environment Development

- 강화학습 환경을 만들 때 주의할 점
 - MDP를 잘 생각해 보기
 - 게임의 상태를 어떻게 나타낼 것인가
 - 에이전트가 어떤 행동을 할 수 있는가
 - 모든 상태를 관찰할 수 있는가 (MDP vs POMDP)
 - 보상을 주기 위해 어떤 정보가 필요한가

Make RL Environment!





AI Frenz Seminar
RL Environment Development





- 강화학습 환경을 만들 때 주의할 점
 - 사전에 저작권 관련 문의하기
 - 게임을 강화학습 환경으로 만들기 전에 저작권에 문제가 없는지 반드시 확인해야 한다.
 - 게임 홈페이지에 있는 연락 수단을 통해 문의해보도록 하자.
 - 실제 사례
 - 1) Hearthstone : 프로젝트 이름에 'Hearthstone'이 들어가지 않게 할 것, 게임의 디자인을 그대로 사용하지 말 것 (아이콘, 카드 이미지 등), 카드 수치나 규칙과 같은 부분은 사용 가능
 - 2) Baba Is You : 비상업적인 목적으로 개발할 것, 초기 레벨에 해당하는 맵만 환경으로 제공할 것
 - 3) Unrailed! : 원본 게임 파일 업로드하지 말 것, 다른 플레이어에게 짜증을 유발할 수 있는 훈련되지 않은/테스트하지 않은/이상한 행동을 하는 모델을 배포하지 말 것

Make RL Environment!

AI Frenz Seminar
RL Environment Development

- 강화학습 환경을 만들 때 주의할 점
- 사전에 저작권 관련 문의하기

Request for making RL environment of "Unrailed!"  Inbox   

 **Chris Ohk** <utilforever@gmail.com> Tue, Aug 11, 2020, 10:44 AM   
to contact ▾





Hello, I'm a big fan of yours. :)

I am interested in creating a variety of Reinforcement Learning(RL) environments. I'm currently creating RL environment for Hearthstone (<https://github.com/utilForever/RosettaStone>). I want to make RL environment for Unrailed!, too.

Before I begin to develop, I'd like to ask your permission. Do you mind if I develop it? And let me know if there is anything I should do about copyright.

I'll be waiting for a reply. :)

Thanks in advance,
Best regards,
Chris.

 **Hendrik Baatz** <hb@indoorastronaut.ch> Wed, Aug 12, 2020, 6:34 PM   
to me ▾

Hi Chris,

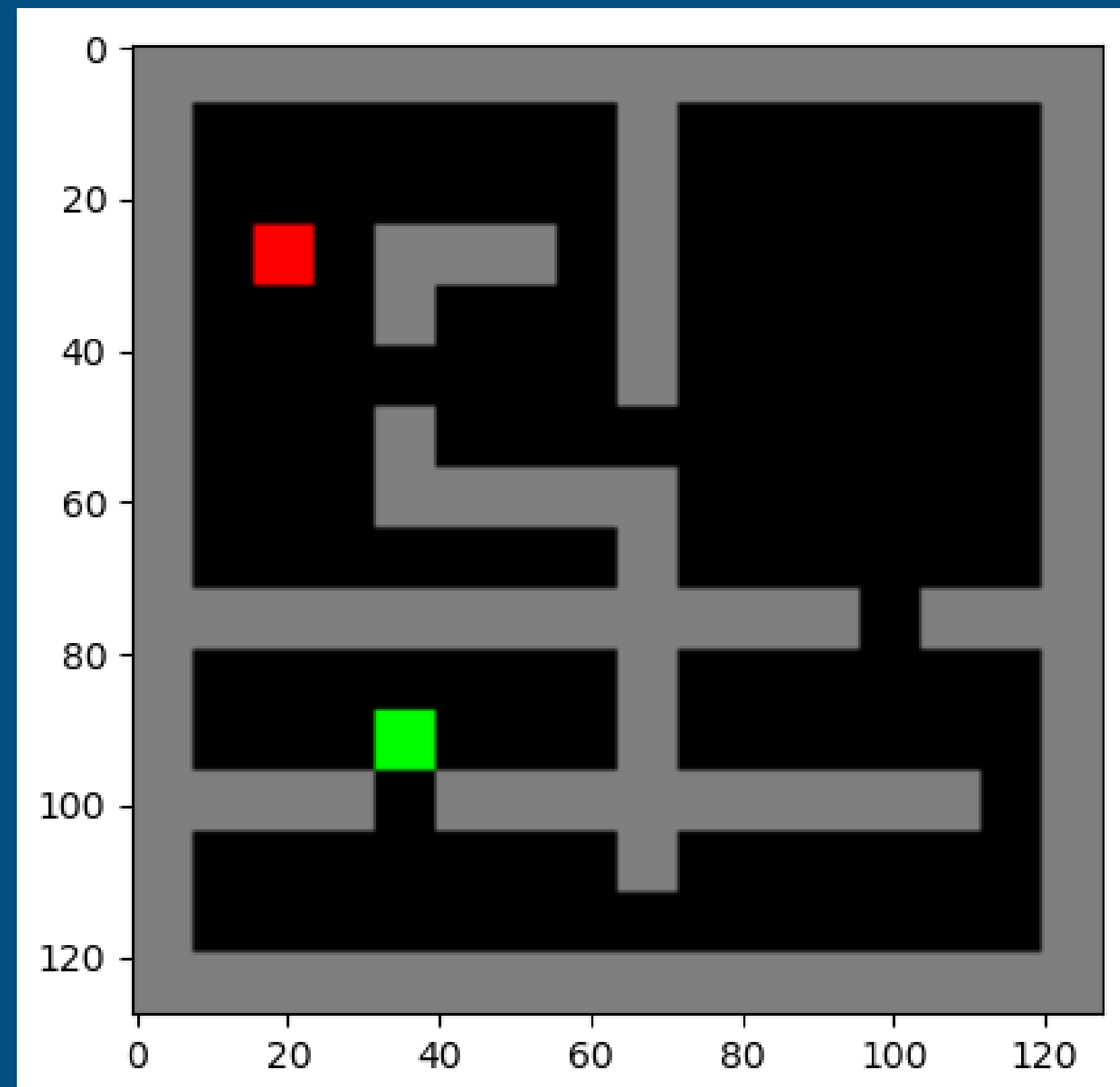
just do not upload any part of our game files and try not to disturb online sessions with untrained/un-tested/poorly performing models that might annoy other players. Other than that we think it's a cool idea and would love to see the progress you make. So go ahead.

Best,
Hendrik

Make RL Environment!

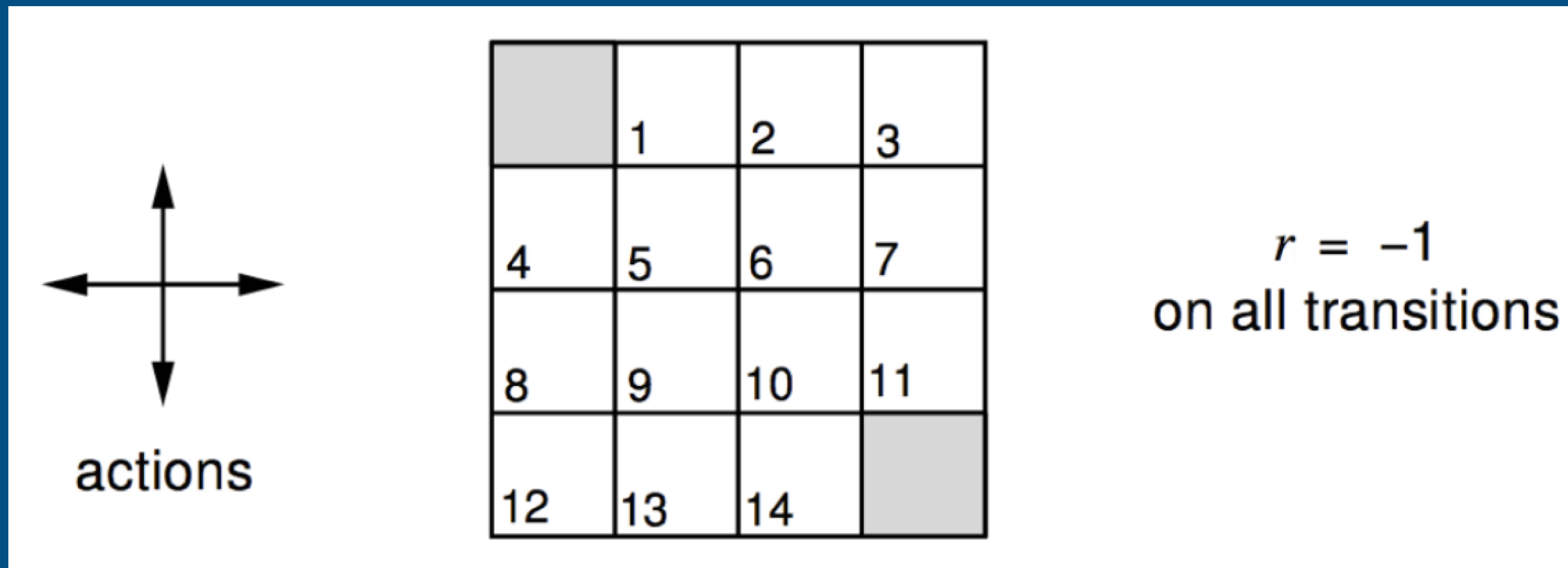
AI Frenz Seminar
RL Environment Development

- 간단한 강화학습 환경을 만들어 봅시다.
- GridWorld



Make RL Environment!

- 간단한 강화학습 환경을 만들어 봅시다.
- GridWorld



Make RL Environment!

AI Frenz Seminar
RL Environment Development

- 간단한 강화학습 환경을 만들어 봅시다.
- OpenAI Gym 기반 환경 클래스를 구현하려면
 - `gym.Env`를 상속받는다.
 - `reset()/step()/render()` 메소드를 구현한다.
- OpenAI Gym에서 사용하는 각종 함수 설명
 - `gym.make()` : 강화학습 환경을 불러온다.
 - `env.reset()` : 환경을 초기화한다.
 - `env.render()` : 환경을 렌더링해 화면으로 출력한다.
 - `env.action_space.sample()` : 임의의 행동을 선택한다.
 - `env.step()` : 선택한 행동을 환경으로 보낸다.

Make RL Environment!

AI Frenz Seminar
RL Environment Development

- 간단한 강화학습 환경을 만들어 봅시다.
- 사용할 패키지 가져오기

```
import gym
import sys
import os
import copy
import numpy as np

from gym import spaces

from PIL import Image as Image
import matplotlib.pyplot as plt
```

Make RL Environment!

AI Frenz Seminar
RL Environment Development

- 간단한 강화학습 환경을 만들어 봅시다.
- 격자에 사용할 색상 정의하기

```
# Define colors
# 0: Black; 1: Gray; 2: Green; 3: Red, 4: Blue
COLORS = {0: [0.0, 0.0, 0.0], 1: [0.5, 0.5, 0.5],
          2: [0.0, 1.0, 0.0], 3: [1.0, 0.0, 0.0],
          4: [0.0, 0.0, 1.0]}
```

Make RL Environment!

- 간단한 강화학습 환경을 만들어 봅시다.
- 초기화 : 행동 공간, 관찰 공간 정의

```
class GridworldEnv(gym.Env):
    num_env = 0

    def __init__(self):
        # Action space
        self.actions = [0, 1, 2, 3, 4]
        self.inv_actions = [0, 2, 1, 4, 3]
        self.action_space = spaces.Discrete(5)
        self.action_pos_dict = {
            0: [0, 0], 1: [-1, 0], 2: [1, 0], 3: [0, -1], 4: [0, 1]}

        # Observation space
        self.obs_shape = [128, 128, 3] # Observation space shape
        self.observation_space = spaces.Box(
            low=0, high=1, shape=self.obs_shape, dtype=np.float32)
```

Make RL Environment!

- 간단한 강화학습 환경을 만들어 봅시다.
- `step()` : 다음 상태를 받아오고 아무 행동도 하지 않은 경우 및 경계 검사

```
def step(self, action):  
    # Return next observation, reward, finished, success  
    action = int(action)  
    next_state = (self.agent_state[0] + self.action_pos_dict[action][0],  
                  self.agent_state[1] + self.action_pos_dict[action][1])  
  
    # Stay in place  
    if action == 0:  
        return (self.observation, 0, False)  
  
    # Out of bounds condition  
    if next_state[0] < 0 or next_state[0] >= self.grid_shape[0]:  
        return (self.observation, 0, False)  
    if next_state[1] < 0 or next_state[1] >= self.grid_shape[1]:  
        return (self.observation, 0, False)
```

Make RL Environment!

- 간단한 강화학습 환경을 만들어 봅시다.
- `step()` : 격자 색깔에 따른 처리

```
# Successful behavior
cur_color = self.current_map[self.agent_state[0], self.agent_state[1]]
new_color = self.current_map[next_state[0], next_state[1]]
if new_color == 0: # Black - empty
    if cur_color == 3: # Red - agent
        self.current_map[self.agent_state[0], self.agent_state[1]] = 0
        self.current_map[next_state[0], next_state[1]] = 3
        self.agent_state = copy.deepcopy(next_state)
elif new_color == 1: # Gray - obstacle
    return (self.observation, 0, False)
elif new_color == 2: # Green - target
    self.current_map[self.agent_state[0], self.agent_state[1]] = 0
    self.current_map[next_state[0], next_state[1]] = 4
    self.agent_state = copy.deepcopy(next_state)
self.observation = self.gridmap_to_observation(self.current_map)
if next_state[0] == self.target_state[0] and next_state[1] == self.target_state[1]:
    target_observation = copy.deepcopy(self.observation)
    return (target_observation, 1, True)
else:
    return (self.observation, 0, False)
```


Make RL Environment!

AI Frenz Seminar
RL Environment Development

- 간단한 강화학습 환경을 만들어 봅시다.
- `reset()`, `render()` : 환경 초기화 및 렌더링 후 화면 출력

```
def reset(self):
    self.agent_state = copy.deepcopy(self.start_state)
    self.current_map = copy.deepcopy(self.initial_map)
    self.observation = self.gridmap_to_observation(self.initial_map)
    return self.observation

def render(self):
    img = self.observation
    plt.clf()
    plt.imshow(img)
    self.fig.canvas.draw()
    plt.pause(0.00001)
```

Make RL Environment!

AI Frenz Seminar
RL Environment Development

- 간단한 강화학습 환경을 만들어 봅시다.
- 유틸리티 함수 : 파일로부터 격자 정보 읽어오기

```
def read_grid_map(self, grid_map_path):  
    with open(grid_map_path, 'r') as f:  
        grid_map = f.readlines()  
        grids = np.array(list(map(lambda x:  
                                   list(map(lambda y: int(y),  
                                             x.split(' '))), grid_map))  
                           )  
    return grids
```

Make RL Environment!

- 간단한 강화학습 환경을 만들어 봅시다.
- 유틸리티 함수 : 에이전트의 시작 위치 및 목표 위치 구하기

```
def get_agent_states(self, initial_map):  
    start_state = None  
    target_state = None  
    start_state = list(map(  
        lambda x: x[0] if len(x) > 0 else None,  
        np.where(initial_map == 3)  
    ))  
    target_state = list(map(  
        lambda x: x[0] if len(x) > 0 else None,  
        np.where(initial_map == 2)  
    ))  
    if start_state == [None, None] or target_state == [None, None]:  
        sys.exit('Start or target state not specified')  
    return start_state, target_state
```

Make RL Environment!

- 간단한 강화학습 환경을 만들어 봅시다.
- 유틸리티 함수 : 격자 맵을 관찰 공간으로 변환

```
def gridmap_to_observation(self, grid_map, obs_shape=None):
    if obs_shape is None:
        obs_shape = self.obs_shape
    observation = np.zeros(obs_shape, dtype=np.float32)
    gs0 = int(observation.shape[0]/grid_map.shape[0])
    gs1 = int(observation.shape[1]/grid_map.shape[1])
    for i in range(grid_map.shape[0]):
        for j in range(grid_map.shape[1]):
            observation[i*gs0:(i+1)*gs0, j*gs1:(j+1)*gs1] = np.array(COLORS[grid_map[i, j]])
    return observation
```

Make RL Environment!

AI Frenz Seminar
RL Environment Development

- 간단한 강화학습 환경을 만들어 봅시다.
- OpenAI Gym 환경 등록

```
from gym.envs.registration import register

register(
    id='gridworld-v0',
    entry_point='gym_gridworld.envs:GridworldEnv',
)
```


Make RL Environment!

- 만든 강화학습 환경이 잘 동작하는지 확인해 봅시다.

[illegible]

Make RL Environment!

AI Frenz Seminar
RL Environment Development

- 만든 강화학습 환경이 잘 동작하는지 확인해 봅시다.

```
from __future__ import unicode_literals

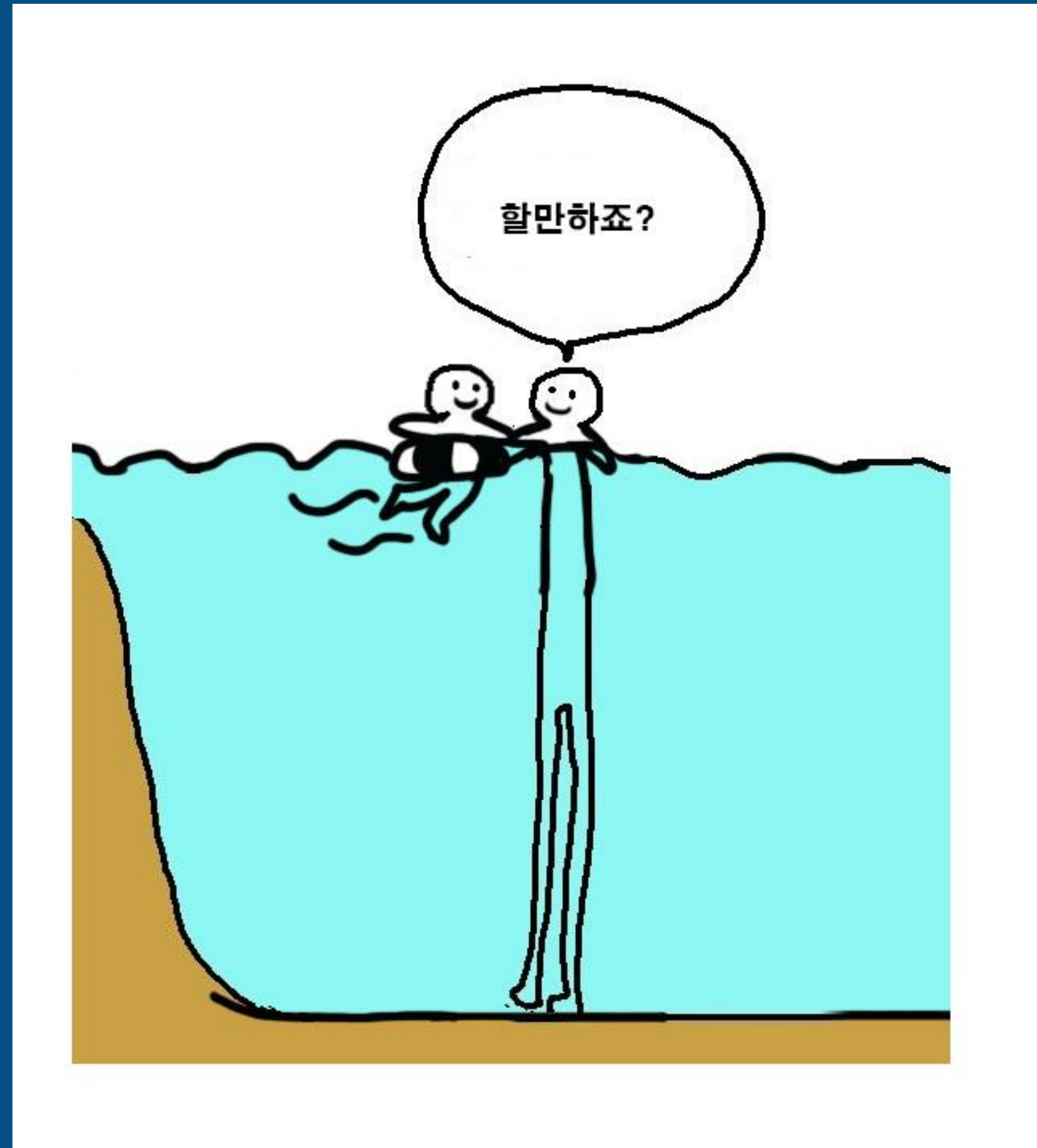
import gym
import gym_gridworld

# To make the environment
env = gym.make('gridworld-v0')

while True:
    env.render()
    _ = env.step(env.action_space.sample())
```

Make RL Environment!

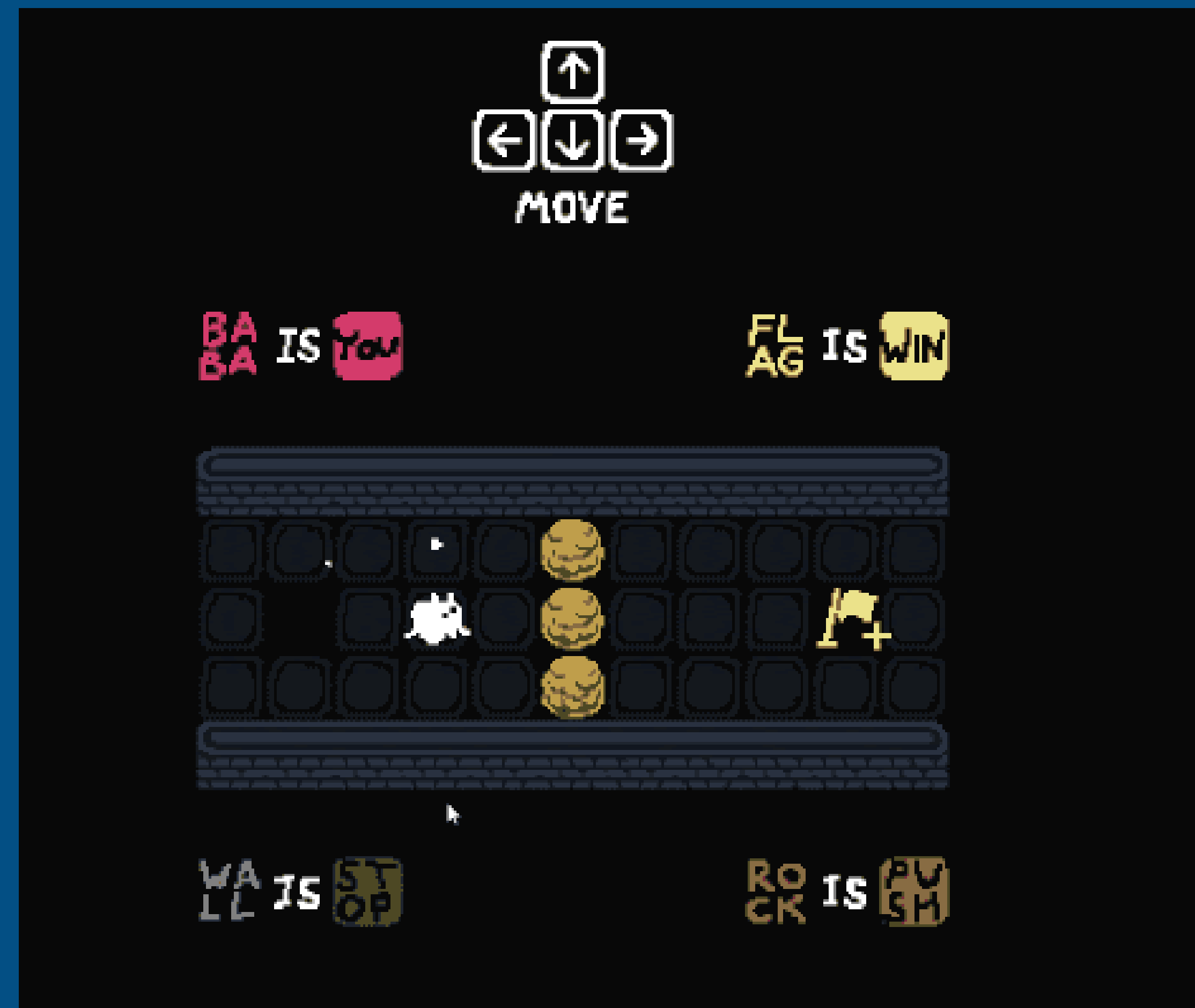
AI Frenz Seminar
RL Environment Development



Case : baba-is-auto

AI Frenz Seminar
RL Environment Development

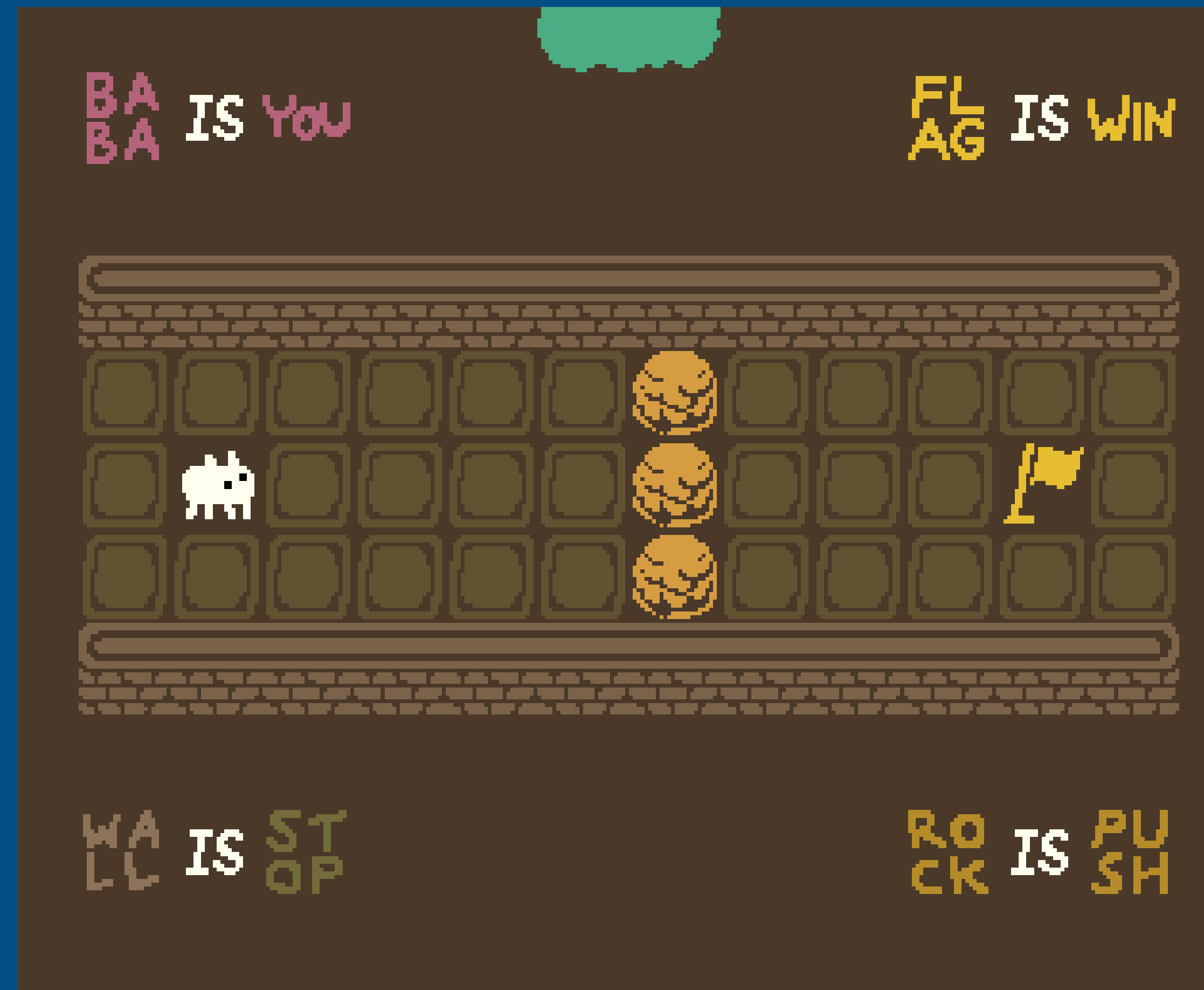
- Baba Is You



Case : baba-is-auto

AI Frenz Seminar
RL Environment Development

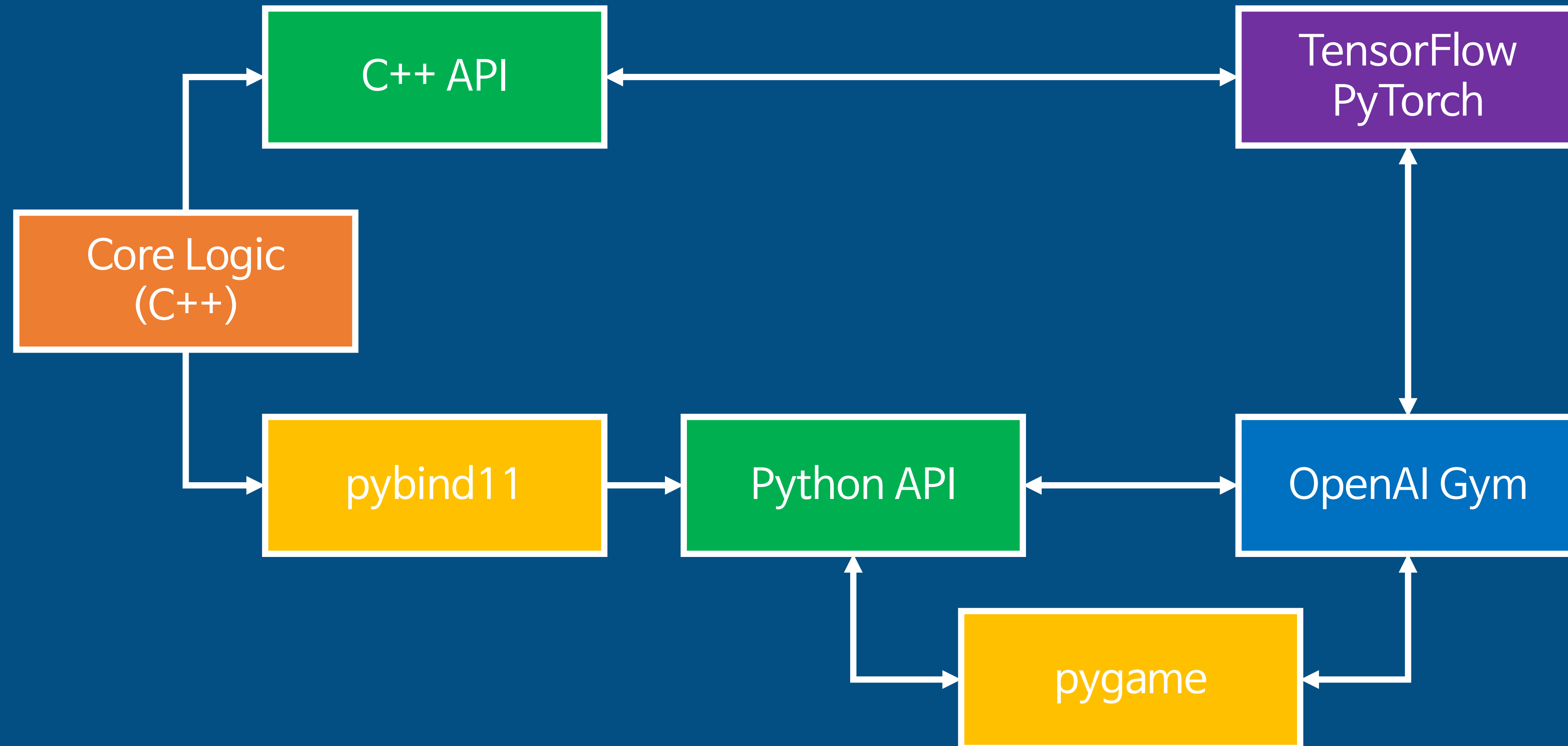
- Baba Is You



Case : baba-is-auto

AI Frenz Seminar
RL Environment Development

- Architecture



Case : baba-is-auto

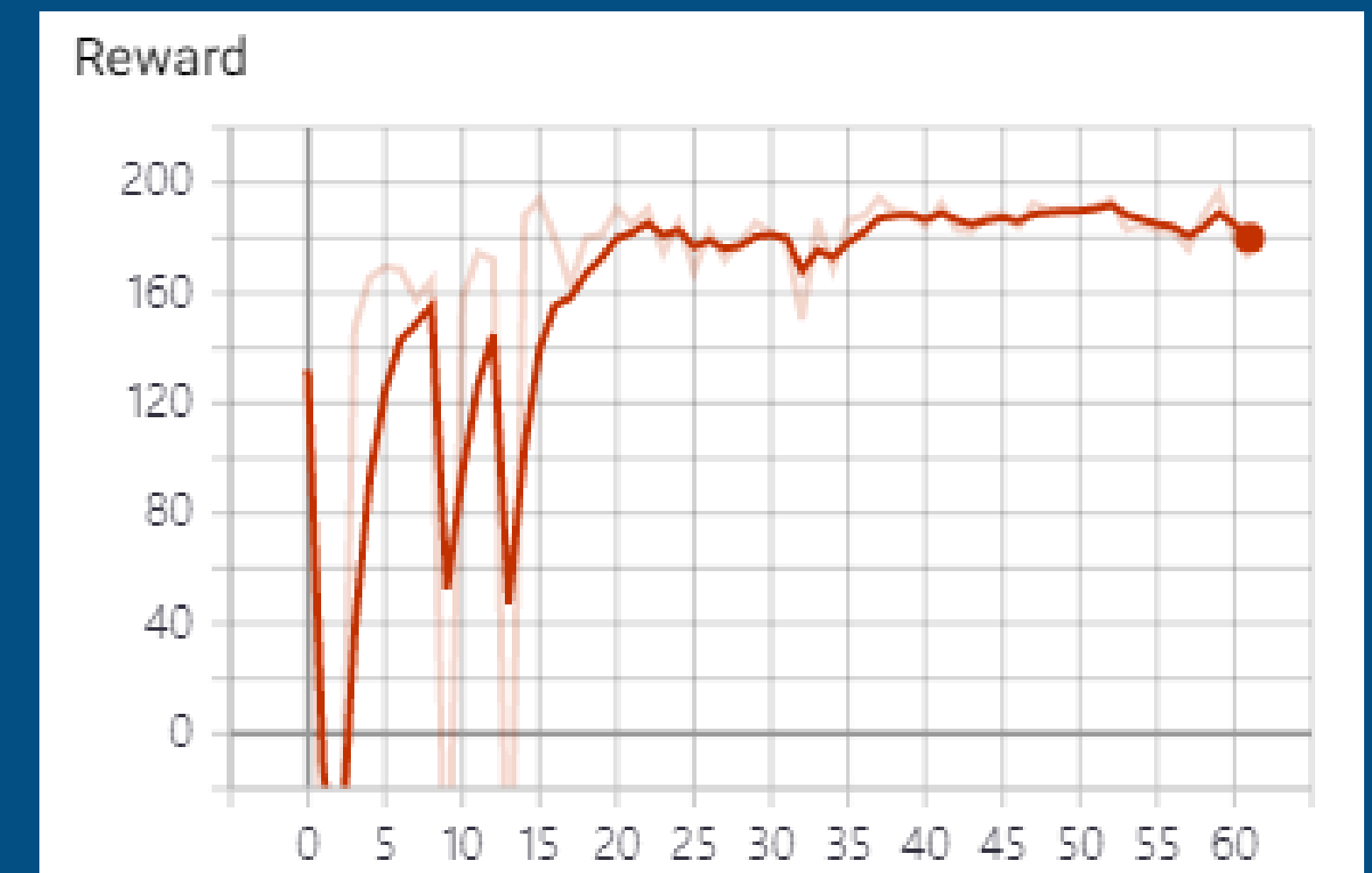
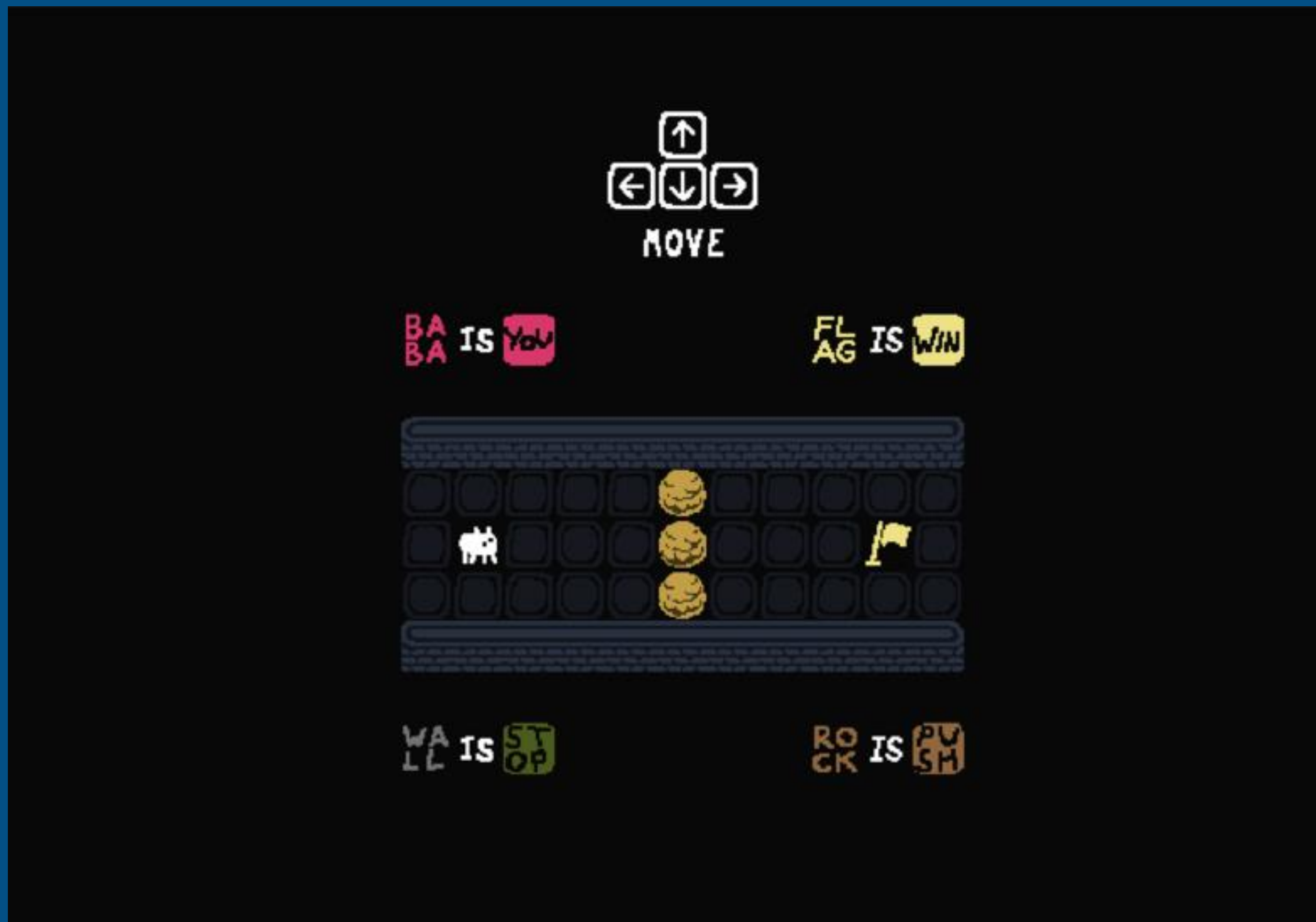
AI Frenz Seminar
RL Environment Development

- 환경 구성
 - Action Space
 - UP
 - DOWN
 - LEFT
 - RIGHT
 - Reward
 - Failed : -100 points
 - Solved : +200 points
 - Each action : -0.5 points

Case : baba-is-auto

AI Frenz Seminar
RL Environment Development

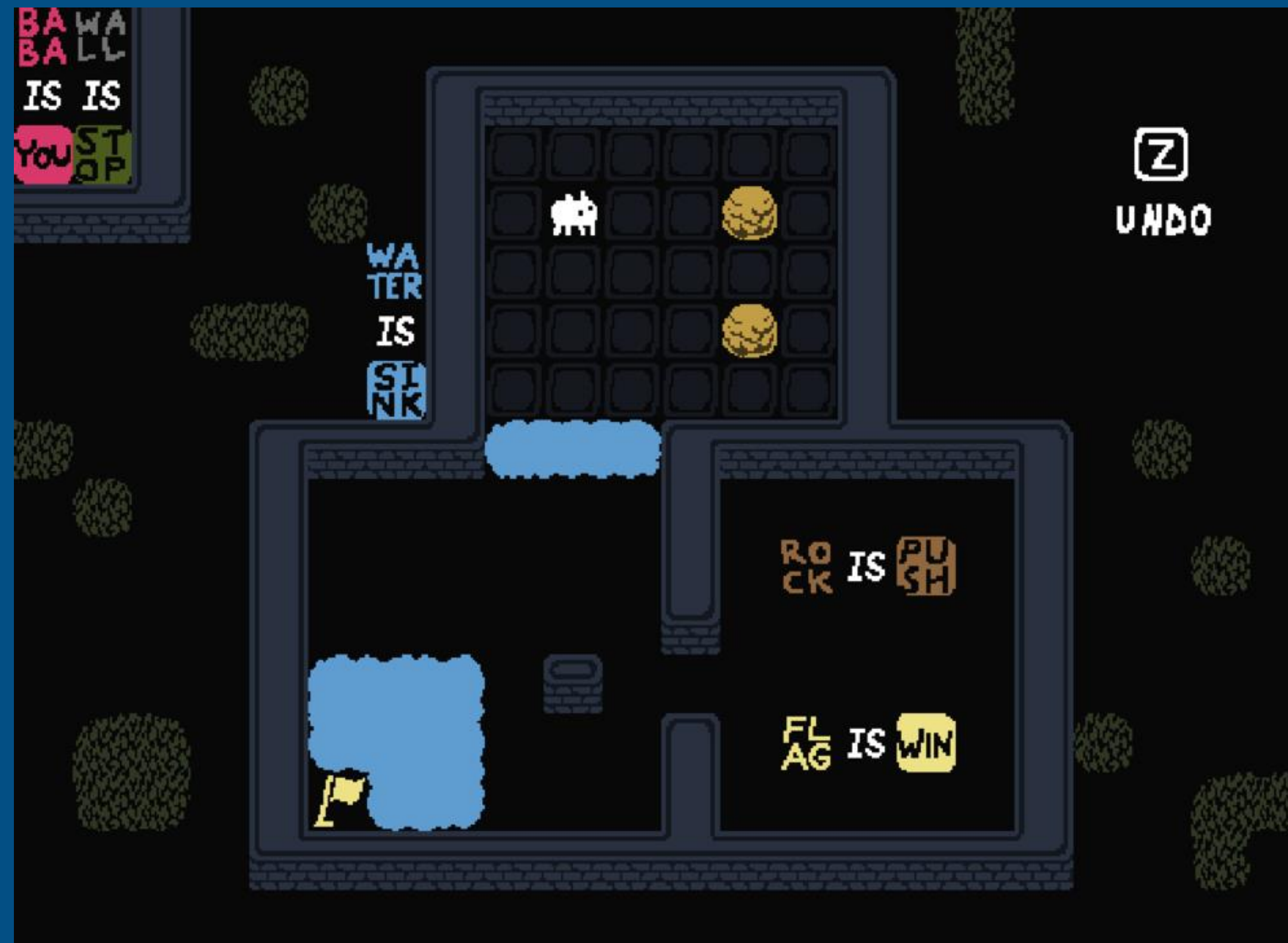
- baba-babaisyou-v0



Case : baba-is-auto

AI Frenz Seminar
RL Environment Development

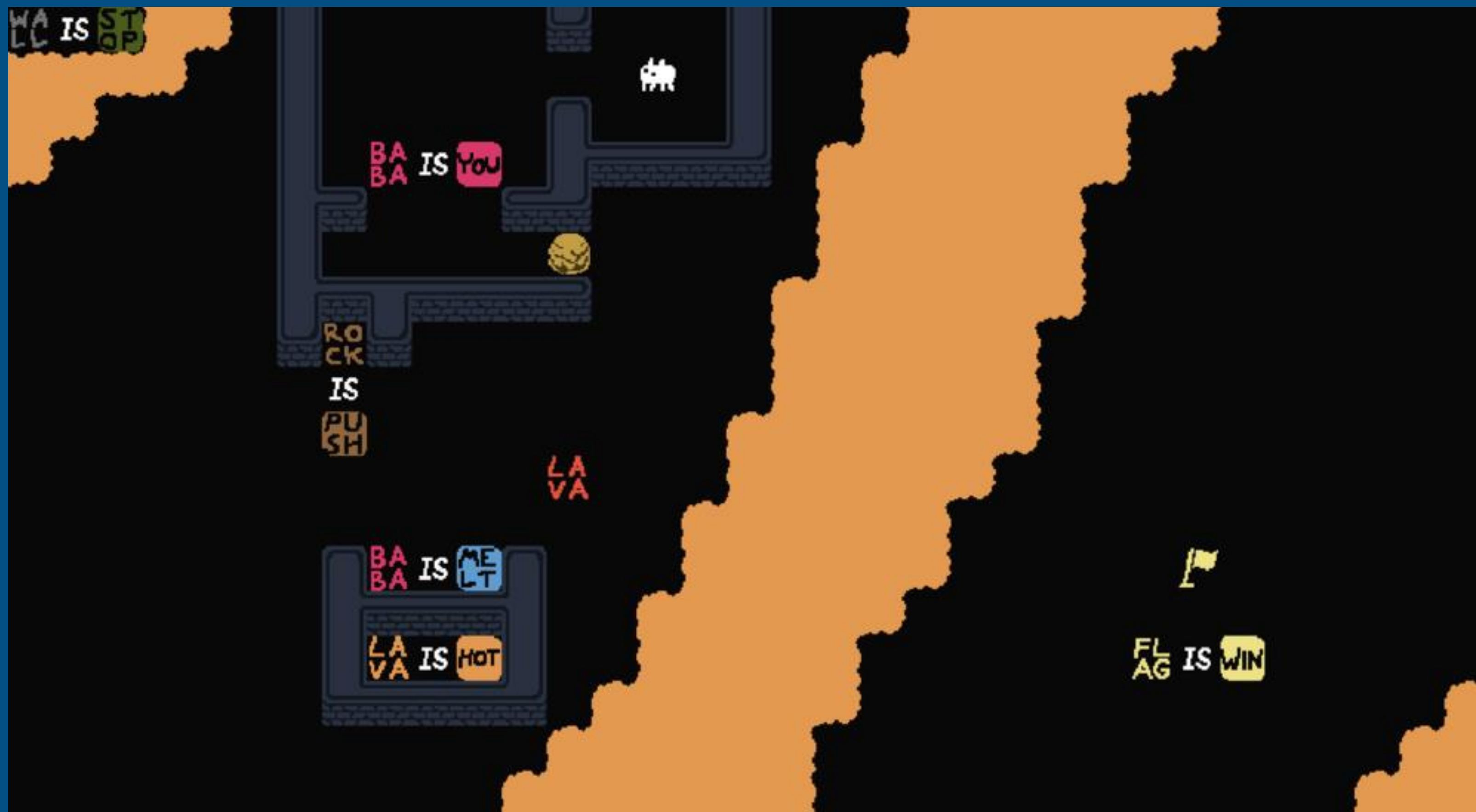
- baba-outofreach-v0



Case : baba-is-auto

AI Frenz Seminar
RL Environment Development

- baba-volcano-v0



Case : RosettaStone

AI Frenz Seminar
RL Environment Development

- Hearthstone



Case : RosettaStone

AI Frenz Seminar
RL Environment Development

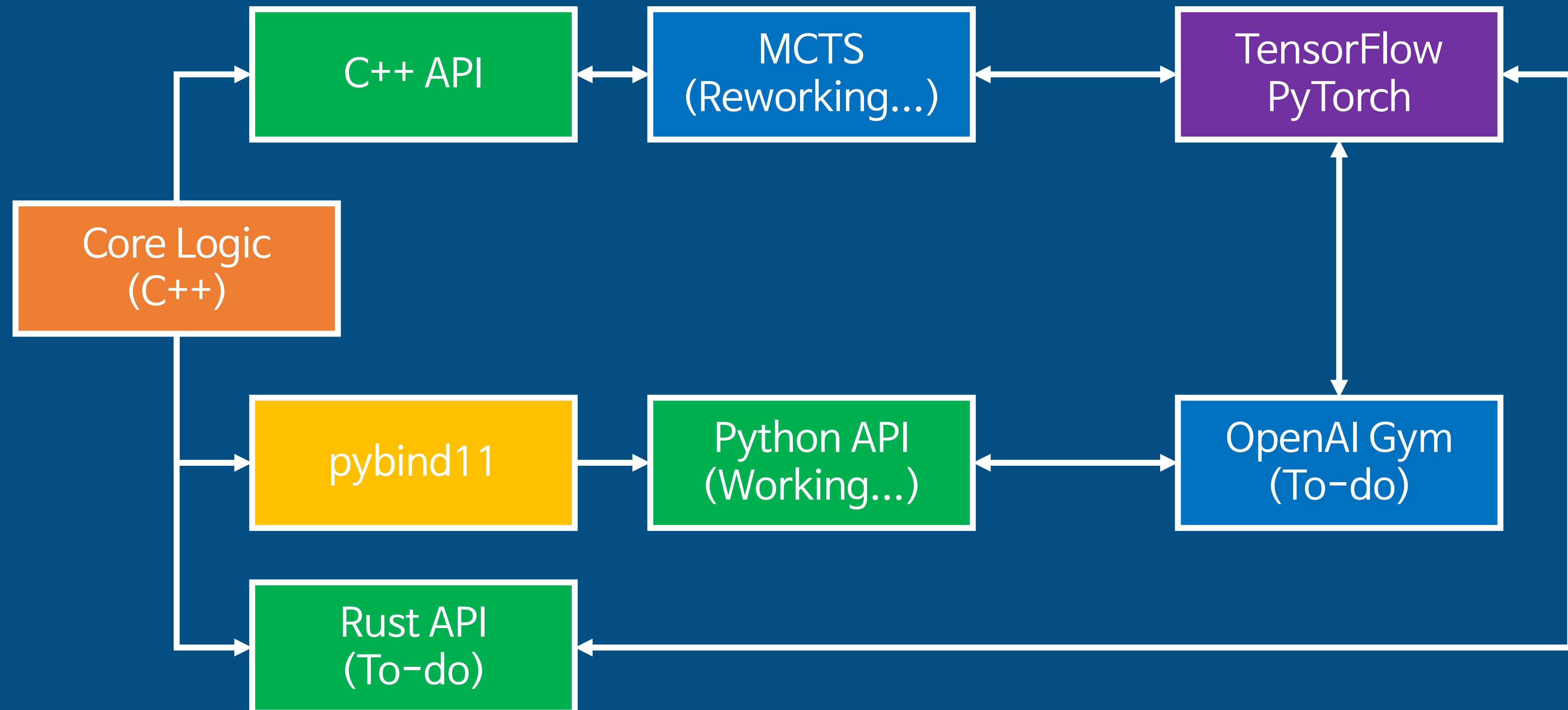
- Hearthstone



Case : RosettaStone

AI Frenz Seminar
RL Environment Development

- Architecture



Case : RosettaStone

AI Frenz Seminar
RL Environment Development

- RosettaStone 2.0 준비
 - ECS (Entity-Component System)로 변경해 시뮬레이션 성능 향상
 - Python API 완벽 지원
 - OpenAI Gym 환경 구현
 - MARL (Multi-Agent RL) 지원 준비
 - Rust로 재구현해 메모리 안전한 강화학습 환경 제공

Conclusion

AI Frenz Seminar
RL Environment Development



Conclusion

AI Frenz Seminar
RL Environment Development



*Behind every great agent
there's a great environment*

Thank you!