강화학습 기반
유저 적응형 협동 전투 NPC

User-Adaptive Cooperative Compat NPC System Based on Reinforcement Learning

손승현 (V2024105)
박준형 (V2025114)

https://github.com/junHyeong7083/rl-multiagent-combat-unity

프로젝트 개요

문제 정의

- **기존 규칙 기반 AI의 한계**

  Behavior Tree, FSM 등의 정해진 패턴으로 인해 행동이 예측 가능해져 몰입감 저하 발생

- **수동 난이도 조절의 비효율성**

  다양한 유저 수준에 맞추기 위해 개발자가 일일이 규칙을 튜닝해야 하는 번거로움

- **실시간 적응 불가**

  플레이어의 스타일이나 숙련도 변화에 즉각적으로 대응하지 못함

프로젝트 목표

- **숙련도 적응형 NPC 개발**

  플레이어의 실력을 실시간으로 파악하여 협동 방식을 스스로 조절하는 AI 학습

- **초보자 대상 전략**

  적극적으로 전투를 주도하고 플레이어를 보호하며 탱킹하는 리더형 NPC

- **숙련자 대상 전략**

  플레이어의 판단을 존중하며 보조와 서포트에 집중하여 시너지를 극대화하는 NPC

제안 방법 : 2단계 파이프라인

강화학습 기반 유저 적응형 협동 전투 NPC

STEP 01
Self-Play 학습

- **PPO + 파라미터 공유**
  동일한 정책을 사용하는 에이전트 간 대전으로 기본 전투 능력 학습

- **균형 AI 생성 목표**
  양 팀 승률이 50:50에 수렴하도록 하여 공정한 실력 확보

OUTPUT

숙련도별 체크포인트 모델

추출

Player Models

Lv.1 초보자(Novice)

Lv.2 중급자(Intermediate)

Lv.3 숙련자(Expert)

에피소드별 저장된
정책 모델 활용

배치

STEP 02
협동 NPC 학습

- **플레이어 모델 고정**
  체크포인트 모델을 플레이어 역할로 배치

- **협동 보상 추가**
  해당 숙련도 플레이어와의 시너지를 위한 별도 보상 체계 적용

OUTPUT

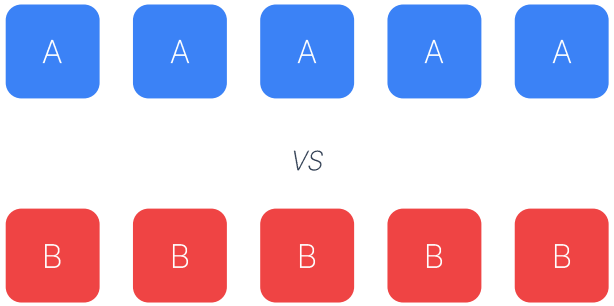
유저 적응형 협동 NPC

설계 : 20x20 격자 5vs5 전투

강화학습 기반 유저 적응형 협동 전투 NPC

전투 환경 설정

맵 크기	20 × 20 격자
팀 구성	5명 vs 5명
최대 스텝	200 스텝
타일 구성	벽 (장애물) 10% 위험 지역 5% 버프 지역 3%



역할별 스탯 (Role Stats)

역할	HP	MP	공격	방어	사거리	고유 스킬
탱커	150	30	10	15	1	도발
딜러	80	50	25	5	1	범위공격
힐러	70	100	8	5	2	치유
레인저	60	60	20	3	4	관통샷
서포터	90	80	12	8	2	버프

State 설계(관찰 공간)

229차원 관찰 벡터 구성

구성 요소	차원	세부 설명
자기 상태	10	HP, MP, 위치(x,y), 생존여부, 역할(one-hot)
아군 상태	40	4명 × 10차원 (자신 제외, 거리순 정렬)
적군 상태	50	5명 × 10차원 (거리순 정렬)
지형 정보	121	본인 중심 11×11 지역 정보 (벽, 위험, 버프)
전역 정보	2	현재 턴 수 (정규화), 팀 ID (-1 or 1)
플레이어 정보	6	협동 대상 유닛의 상대좌표, 거리, 상태(Step2에서만 사용)

→

229 Dimensions

Action 설계:행동 공간(Discrete 12)

강화학습 기반 유저 적응형 협동 전투 NPC

행동 공간 개요

이산 행동 공간 (Discrete): 총 12개의 행동 중 하나를 선택

공통 행동: 이동, 기본 공격 (모든 역할)

특수 행동: 역할별 고유 스킬 1종

동시 행동 처리

12개 이산 행동 목록 (Discrete Action List)

ID	행동명	대상	설명 및 사용 조건
0	**제자리 대기**	All	아무 행동도 하지 않음 (MP 회복)
1-4	**이동 (Move)**	All	상하좌우 1칸 이동
5	**가까운 적 공격**	All	사거리 내 가장 가까운 적을 공격 (기본 공격)
6	**최저 HP 적 공격**	All	사거리 내 HP가 가장 낮은 적을 우선 공격 (마무리)
7	**범위 공격**	딜러	타겟 주변 십자(+) 범위에 광역 피해
8	**치유(Heal)**	힐러	가장 HP 비율이 낮은 아군 1명 회복
9	**도발**	탱커	주변 2칸 내 적들이 자신을 공격하게 함
10	**관통샷**	레인저	직선상에 있는 모든 적을 관통하여 피해
11	**버프 (Buff)**	서포터	주변 아군의 공격력/방어력을 3턴간 증가

Reward 설계(1):계층적 보상 구조

강화학습 기반 유저 적응형 협동 전투 NPC

총 보상 함수 (Total Reward)

$$R_{total} = R_{sparse} + R_{dense} + R_{role}$$

희소 보상(결과)과 밀집 보상(행동 유도), 역할 보상(협동)을 결합하여 학습 효율성과 역할 수행 능력을 동시에 극대화합니다.

희소 보상 (Sparse Reward)

조건 (CONDITION)	보상 (VALUE)	설명
승리 (Victory)	+25.0	게임 목표 달성
패배 (Defeat)	-15.0	패배 회피 유도
무승부 (Draw)	-10.0	지연 행위 억제

밀집 보상 (Dense Reward)

학습 가속화

행동 (ACTION)	보상 (VALUE)	설계 목적 (PURPOSE)
적 처치	+15.0	결정적인 기여에 대한 높은 보상으로 공격적 행동 강화
데미지	+0.5 /HP	지속적인 공격 유도 (Shaping Reward)
적 접근	+0.3	전투 참여를 위해 적 방향으로 이동 유도 (초기 학습용)
제자리 대기	-0.5	무의미한 턴 소모 방지 및 소극적 플레이 억제
거리 이탈	가변	적과의 거리 8칸 초과 시 -0.1 × (d-8) 패널티 부여 → 전장을 이탈하여 도망가는 행위 방지

설계 의도: 초기에는 희소 보상(승/패)만으로 학습이 어렵기 때문에, 밀집 보상을 통해 에이전트가 '공격', '접근' 등의 유의미한 행동을 하도록 유도합니다.

Reward 설계(2):역할별/협동 보상

강화학습 기반 유저 적응형 협동 전투 NPC

역할별 보상 (R_role)

각 역할의 고유한 행동 특성을 강화하기 위한 추가 보상 체계입니다.

역할	보상 조건	보상 값
탱커	**피격 시 (어그로 담당)** 사망 시 패널티 부여	+0.15(어그로) / -2.0 (사망)
딜러	**적에게 데미지 가함**	+0.3 / HP
힐러	**아군 치유 (유효 힐량)**	+0.15 / HP
레인저	**원거리 공격 (2칸 이상 거리)**	+0.1 / HP
서포터	**아군 근처 위치 (3칸 이내)**	+0.03 / 명

협동 보상 (2단계 학습용)

플레이어와의 협력을 유도하기 위한 보상입니다.

조건	보상
플레이어 근접 유지 플레이어와 3칸 이내 거리 유지	+2.5 / 칸
플레이어 이탈 패널티 플레이어와 5칸 이상 거리 벌어짐	-1.5 / 칸
사거리 내 전투 참여 플레이어 공격 대상 협공	+15.0
플레이어 보호 (탱커) 플레이어 피격 대신 맞음 / 도발	+0.2

설계 의도

초보 플레이어일수록 NPC가 더 적극적으로 보호하고 전투를 주도하도록, 숙련된 플레이어일수록 NPC가 보조하고 시너지를 내도록 보상 가중치를 조절합니다.

알고리즘:PPO(Proximal Policy Optimization)

강화학습 기반 유저 적응형 협동 전투 NPC

클리핑 목적 함수 (Clipped Objective Function)

정책 업데이트가 너무 크게 변하는 것을 방지하여 학습 안정성 확보

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

$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta old}(a_t|s_t)}$ 확률 비율 (Probability Ratio)

$\hat{A}_t$ GAE (Generalized Advantage Estimation)로 추정된 어드밴티지

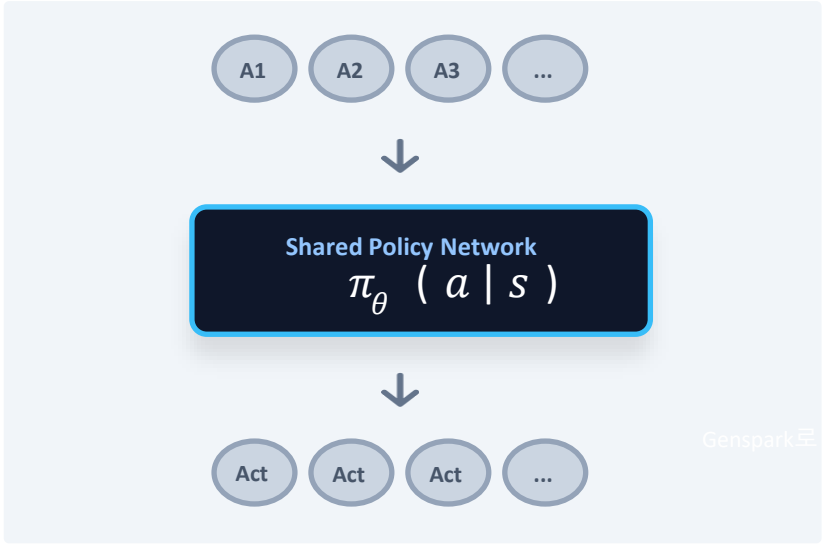
ϵ 클리핑 범위 파라미터 (본 연구에서는 0.2 사용)

Why PPO?

TRPO의 복잡한 연산 없이도 안정적인 정책 업데이트가 가능하며, 하이퍼파라미터에 덜 민감하여 멀티 에이전트 학습에 적합함.

파라미터 공유 (Parameter Sharing)

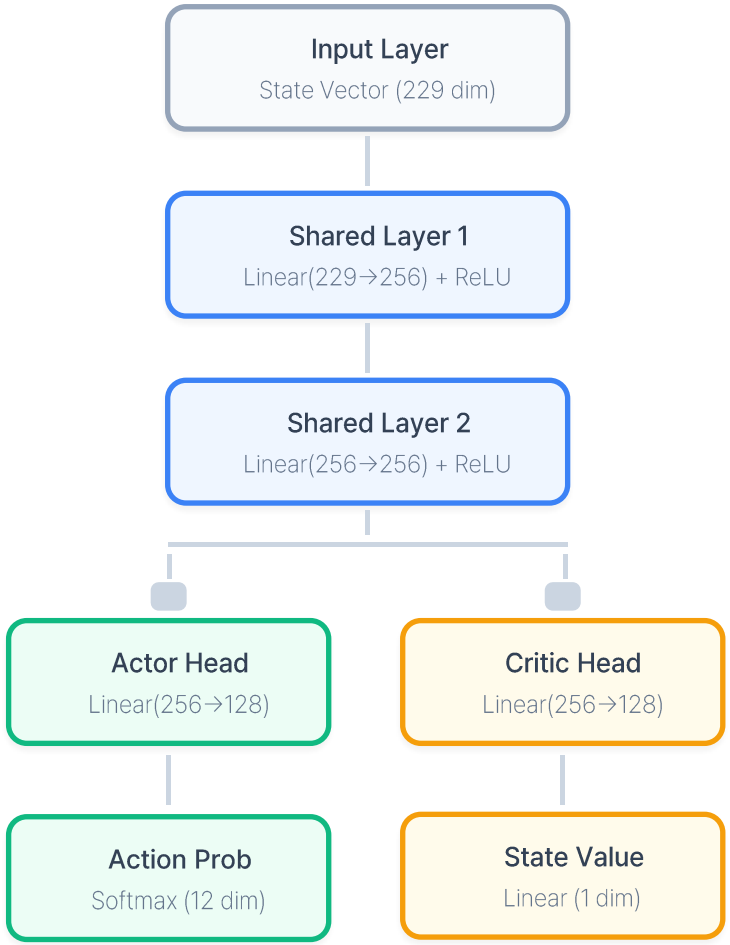
- 단일 네트워크 공유: 모든 에이전트가 동일한 정책 네트워크 πθ 사용
- 입력의 차별화: 각 에이전트는 자신의 관점에서 상태를 관찰하므로 동일 정책이어도 다른 행동 도출
- 학습 효율성 증대: 샘플 효율성(Sample Efficiency)이 높고 학습 수렴 속도 향상



네트워크 구조 및 Hyperparameter

강화학습 기반 유저 적응형 협동 전투 NPC

Actor-Critic Network Architecture



PPO Hyperparameters

PARAMETER	VALUE
Learning Rate	3×10^{-4}
Discount Factor (γ)	0.99
GAE Lambda (λ)	0.95
Clip Epsilon (ε)	0.2
Entropy Coefficient	0.01
Value Coefficient	0.5
Batch Size	256
Epochs per Update	4
Optimizer	Adam

실험 환경

하드웨어 사양 (Hardware Specs)

CPU	Intel Core i7-14700F
GPU	NVIDIA GeForce RTX 4060 Ti
RAM	32GB DDR5
OS	Windows 11

소프트웨어 환경 (Software Specs)

Language	Python 3.10
Framework	PyTorch 2.0
Engine	Unity 2022.3 LTS
Communication	Socket (UDP/TCP)

평가 지표 (Evaluation Metrics)

Team A/B 승률

양 팀의 승리/패배/무승부 비율을 측정하여 에이전트 성능의 균형과 우위를 평가

평균 에피소드 보상

에피소드 당 획득한 총 보상의 평균값으로 학습의 수렴 여부와 정책 개선도를 확인

평균 에피소드 길이

전투가 종료될 때까지 소요된 평균 스텝 수로 전투의 치열함과 지속성을 평가

학습 FPS

초당 처리되는 프레임(스텝) 수로 학습 환경의 효율성과 최적화 수준을 측정

실험 결과(1단계):셀프 플레이

강화학습 기반 유저 적응형 협동 전투 NPC

12,000 에피소드 학습 결과

평가 지표 (METRIC)	결과값 (VALUE)
총 학습 스텝	19,936,010 steps
Team A 승률	50.2%
Team B 승률	49.4%
무승부 비율	0.4%

평균 FPS

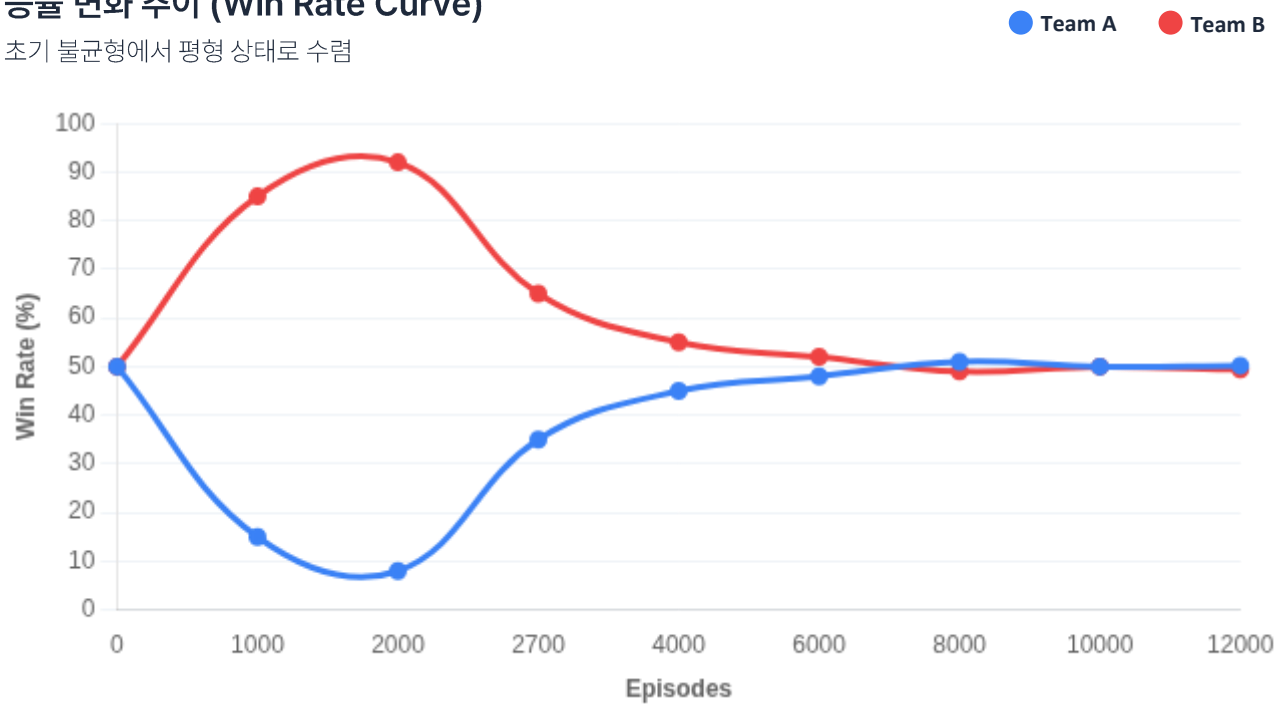
~905

총 학습 시간

6.1 hours

승률 변화 추이 (Win Rate Curve)

초기 불균형에서 평형 상태로 수렴



최종 목표 달성

양 팀 승률이 50:50에 근접하여 어느 한 쪽으로 치우치지 않는 균형 잡힌 AI 모델(Balanced AI) 생성에 성공함.

Insight: 학습 양상 분석

초기에는 Team B가 90% 이상 우세했으나, 약 2,700 에피소드(4.8M 스텝) 이후 전략적 대응을 학습하며 승률 균형(Equilibrium)에 도달함.

실험 결과(2단계):협동 학습

강화학습 기반 유저 적응형 협동 전투 NPC

12,750 에피소드 구간별 성능 (Performance by Epoch)

구간 (EPISODES)	평균 보상 (MEAN REWARD)	TEAM A 승률 (WIN RATE)
초기 (10~1,000)	-1,297 ~ -664	10 ~ 20%
중기 (1,000~6,000)	-664 ~ +752	20 ~ 32%
후기 (6,000~12,750)	+752 ~ +1,739	32 ~ 47%

보상 증가율

▲ 3,036

최종 승률 균형

47:52

AVG TANK DIST

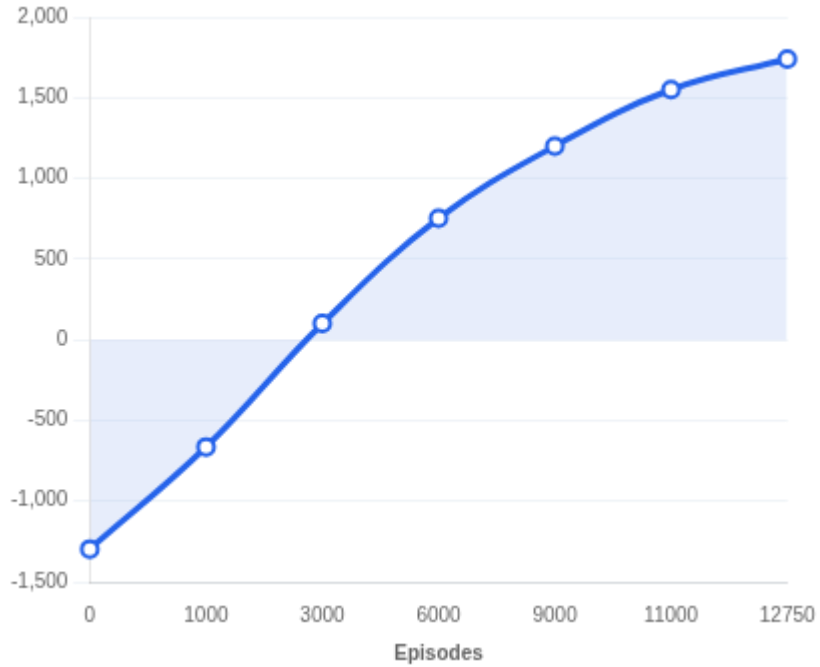
2.91 (10.0→2.91)

학습 성공: 탱커(플레이어) 협동 행동 발현

평균 탱커 거리(avg_tank_dist)가 초기 10.0에서 최종 2.91로 감소함. 이는 NPC 탱커가 플레이어를 효과적으로 따라다니며 보호하는 협동 행동(Cooperative Behavior)을 성공적으로 학습했음을 의미함.

학습 곡선 (Reward Curve)

Total: 12,750 Eps



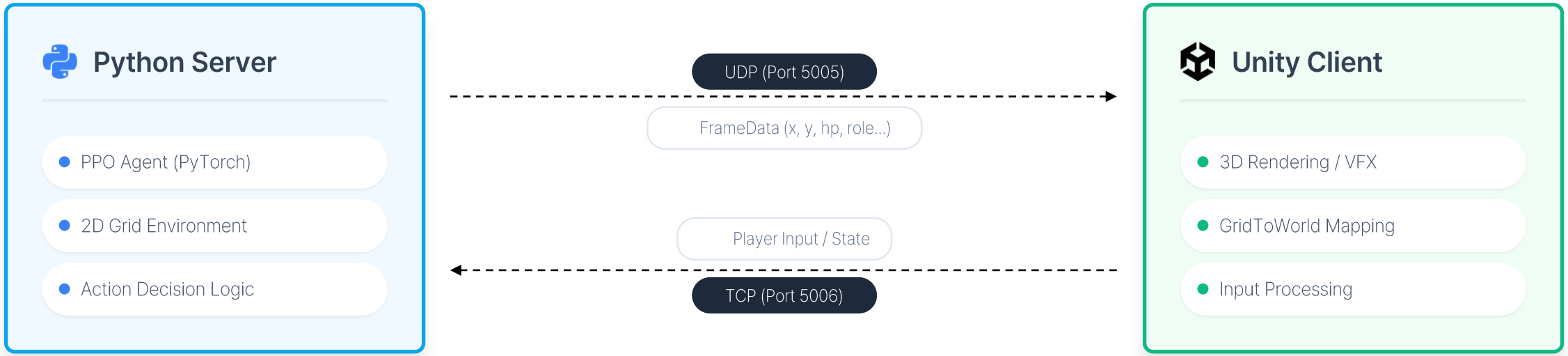
그래프 분석

초기(-1,297)에서 급격한 성능 향상 후 후기(+1,739)에 안정적으로 수렴하는 양상을 보임.

Unity 3D 데모:UDP 미러링 아키텍처

강화학습 기반 유저 적응형 협동 전투 NPC

SYSTEM ARCHITECTURE



Low Latency (~1ms)

ONNX 변환 없이 PyTorch 모델을 직접 사용하여 실시간 추론 및 제어 가능

Decoupled Architecture

학습(Python)과 렌더링(Unity)의 분리로 리소스 효율화 및 빠른 학습 가능

Dual Mode Support

AI vs AI 관전 모드 및 Human vs AI 플레이어 모드 즉시 전환 지원

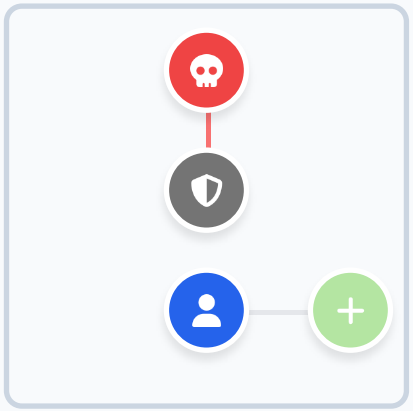
데모:학습된 NPC 행동 패턴

강화학습 기반 유저 적응형 협동 전투 NPC

협동 모드 예시

SCENARIO CONDITION

플레이어(User)
Role: 딜러 (Dealer) 선택 시



Tactical Formation Overview

NPC 적응 행동
탱커: 플레이어 전방 2칸 앞에서 보호
힐러: 플레이어 기준 3칸 이내 유지

역할별 학습된 전략 (Learned Strategies)

역할 (ROLE)	주요 행동 패턴 및 전략
탱커 (Tanker)	전방 배치 어그로 관리 적과 아군 사이에 위치하여 피해를 대신 흡수. 체력이 낮아지면 후방으로 일시 후퇴하여 생존 도모.
딜러 (Dealer)	집중 공격 안전 거리 탱커의 후방에서 적의 주요 타겟을 지속적으로 공격(DPS). 범위 공격 스킬 사용 시 아군 오폭 방지.
힐러 (Healer)	후방 지원 유저 밀착 전투 라인 최후방에 위치하며 체력이 낮은 아군 우선 치유. 협동 모드 시 플레이어 주변을 배회하며 생존 지원.
레인저 (Ranger)	카이팅(Kiting) 최대 사거리 적이 접근하면 거리를 벌리며(Kiting) 원거리에서 견제. 지형지물(벽)을 활용하여 엄폐 사격 수행.
서포터 (Supporter)	버프 활용 진형 유지 아군이 밀집된 구역 중앙에 위치하여 버프 효율 극대화. 적의 진입 경로를 차단하는 보조 역할 수행.

토의:핵심 기여(Key Contributions)

1. 셀프플레이 체크포인트의 창의적 활용

별도의 인간 플레이어 데이터 수집 없이, 학습 과정의 부산물인 체크포인트 모델을 활용하여 숙련도별(초보~숙련) 플레이어 모델을 자동으로 생성함. 데이터 수집 비용을 획기적으로 절감.

2. '이기는 AI'에서 '협동하는 AI'로의 전환

기존 강화학습 연구가 인간을 능가하는 경쟁적 성능에 초점을 맞춘 반면, 본 연구는 유저의 실력에 맞춰 최적의 시너지를 발휘하는 동료 NPC 구현에 집중하여 게임 경험(UX) 향상에 기여.

3. 실용적이고 확장 가능한 배포 파이프라인

복잡한 ONNX 변환 과정 없이 UDP 미러링을 통해 PyTorch 학습 환경과 Unity 렌더링 환경을 연동함으로써, 다양한 게임 엔진(Unreal, Godot 등)으로 쉽게 확장 가능한 구조 제시.

단순 성능 최적화를 넘어, 플레이어 경험을 고려한 적응형 AI 설계 방법론을 제시함

한계점 및 향후 연구

연구의 한계점

- 격자 기반 환경의 단순성 이산적인 20x20 격자 환경은 실제 MMORPG와 같은 복잡한 연속 공간과 물리 엔진을 완전히 반영하지 못하여 일반화에 한계가 있음
- 네트워크 통신 지연 이슈 Python 서버와 Unity 클라이언트 간의 UDP 통신 과정에서 패킷 유실이나 미세한 지연이 발생할 경우 학습된 정밀 행동과 불일치 발생 가능
- 체크포인트 모델 검증 미흡 셀프플레이 체크포인트가 실제 다양한 숙련도의 인간 플레이어 행동 양식을 완벽하게 대변하는지에 대한 정량적 상관관계 검증 부족

향후 연구 방향

- 숙련도 판별 시스템 정교화 단순 승률뿐만 아니라 APM, 이동 패턴, 스킬 적중률 등 다양한 지표를 종합하여 실시간으로 유저 숙련도를 정밀하게 판별하는 모델 연구
- 실사용자 대상 UX 평가 실제 게이머들을 대상으로 한 AB 테스트 및 설문 조사를 통해, 적응형 NPC가 게임의 몰입감과 재미에 미치는 영향을 정성적으로 평가
- 다른 장르 및 환경 확장 전투 중심의 시나리오를 넘어 탐험, 퍼즐 해결 등 다양한 협동 장르로 확장하고, 3D 연속 공간에서의 적응형 에이전트 학습 연구

결론(Conclusion)

1. 효율적인 2단계 학습 파이프라인 제안

셀프플레이를 통해 균형 잡힌 기본 AI를 생성하고, 학습 부산물인 체크포인트를 활용하여 별도의 데이터 수집 없이 숙련도별 협동 NPC를 학습시키는 방법론 확립.

2. 성공적인 실험 결과 및 성능 향상

12,000 에피소드 학습을 통해 50:50 승률 균형을 달성했으며, 협동 학습 단계에서 평균 보상이 -139에서 +76으로 대폭 상승하며 유의미한 협동 행동 발현.

3. 실용적인 Unity 연동 시스템 구축

UDP 기반 미러링 아키텍처를 통해 실시간 데모를 구현하였으며, Python 학습 환경과 Unity 렌더링 환경을 효율적으로 결합하여 실제 게임 적용 가능성 입증.

플레이어 숙련도에 따라 협동 방식을 능동적으로 조절하는
지능형 NPC 시스템 개발 성공

마무리 및 참고자료

Thank You!

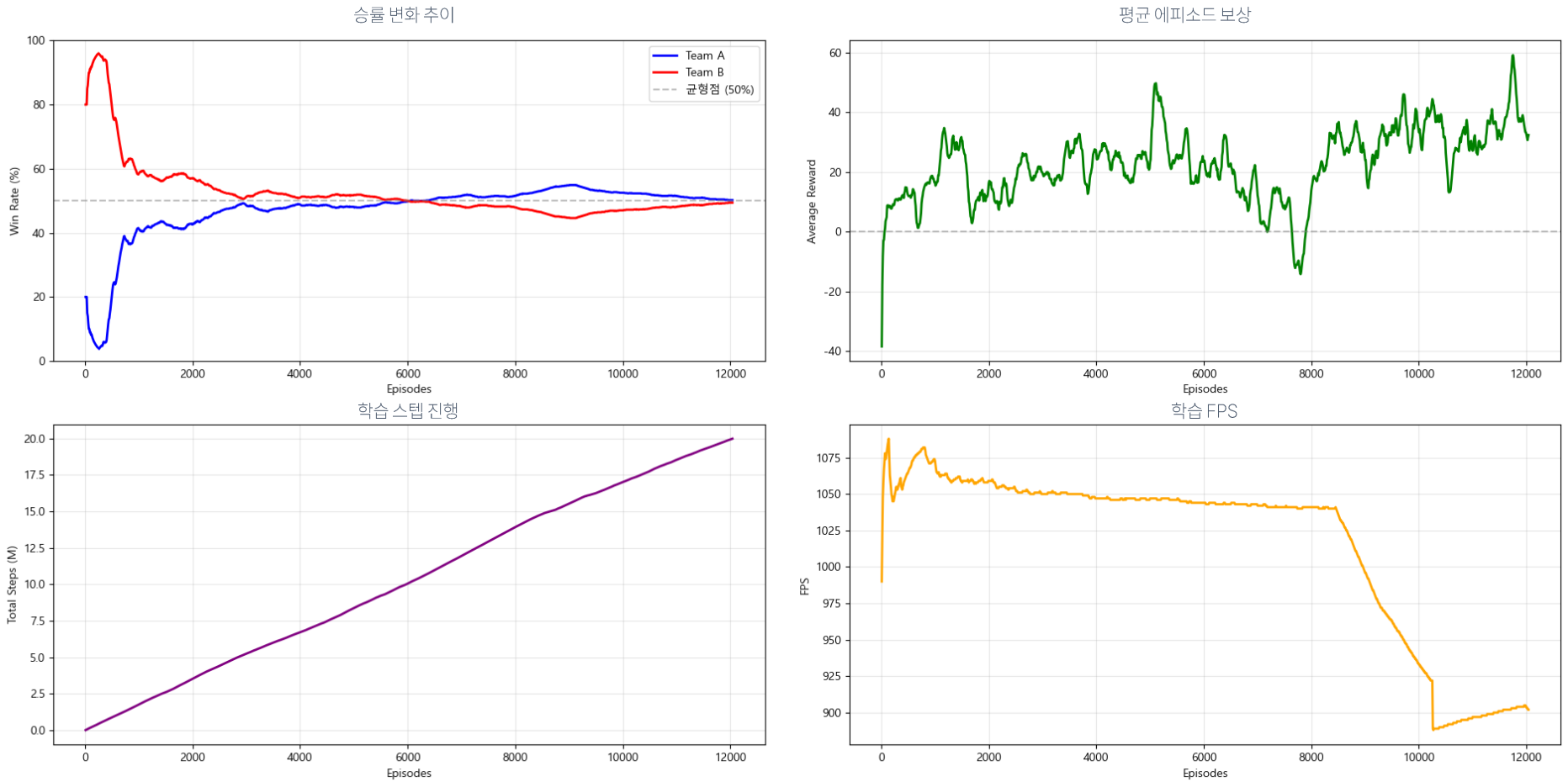
GITHUB REPOSITORY

github.com/junHyeong7083/rl-multiagent-combat-unity

bash — 80x24

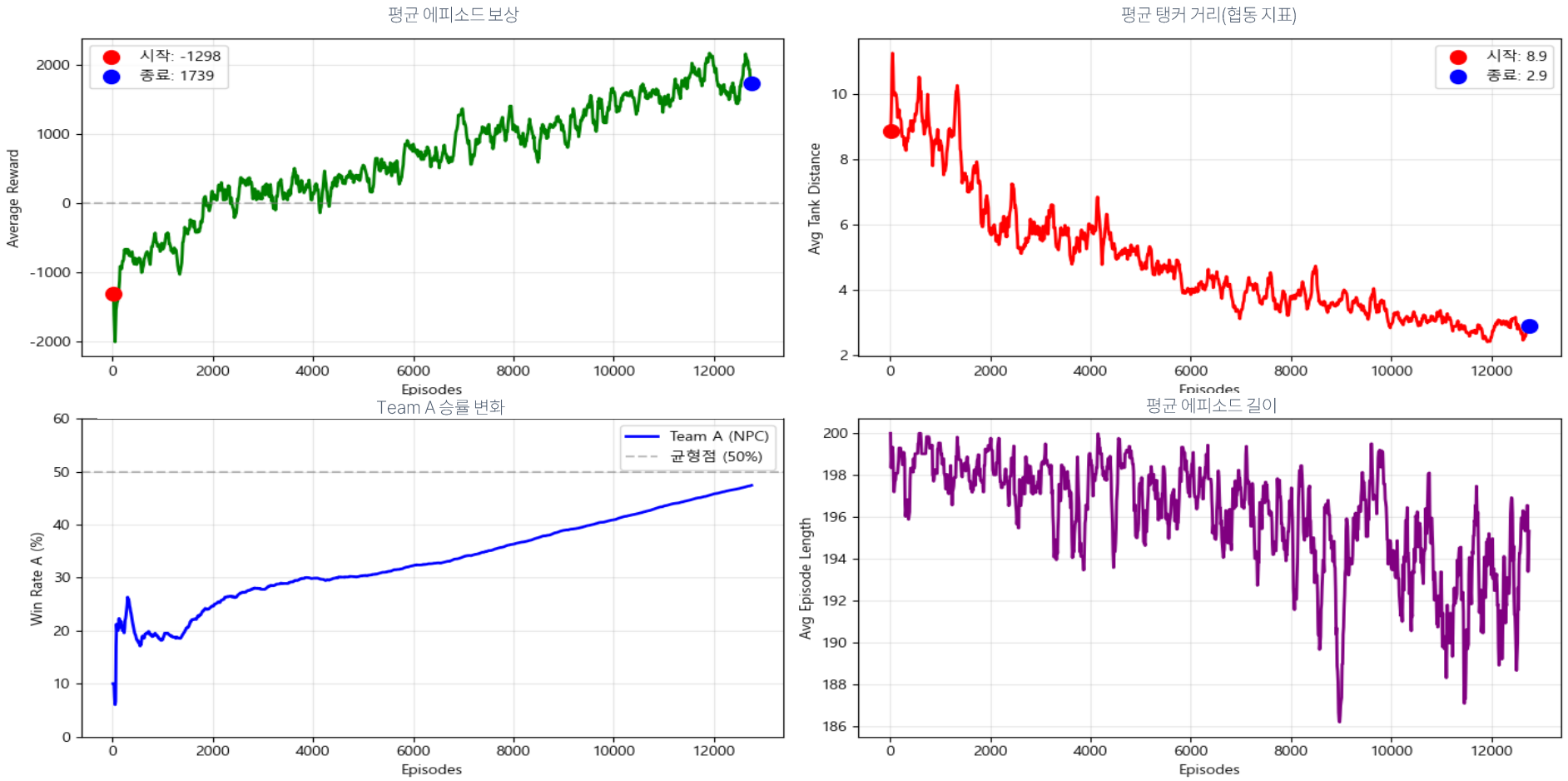
```
# 1. 강화학습 모델 학습 시작 (PPO)
python train.py --total-steps 500000

# 2. Unity 연동 (AI vs AI 관전 모드)
python unity_streamer.py --mode spectator

# 3. Unity 연동 (Human vs AI 플레이어 모드)
python player_mode_streamer.py --role dealer

Running server on localhost:5005...
Waiting for Unity client connection...
Connected! Starting episode 1.
```

별첨1

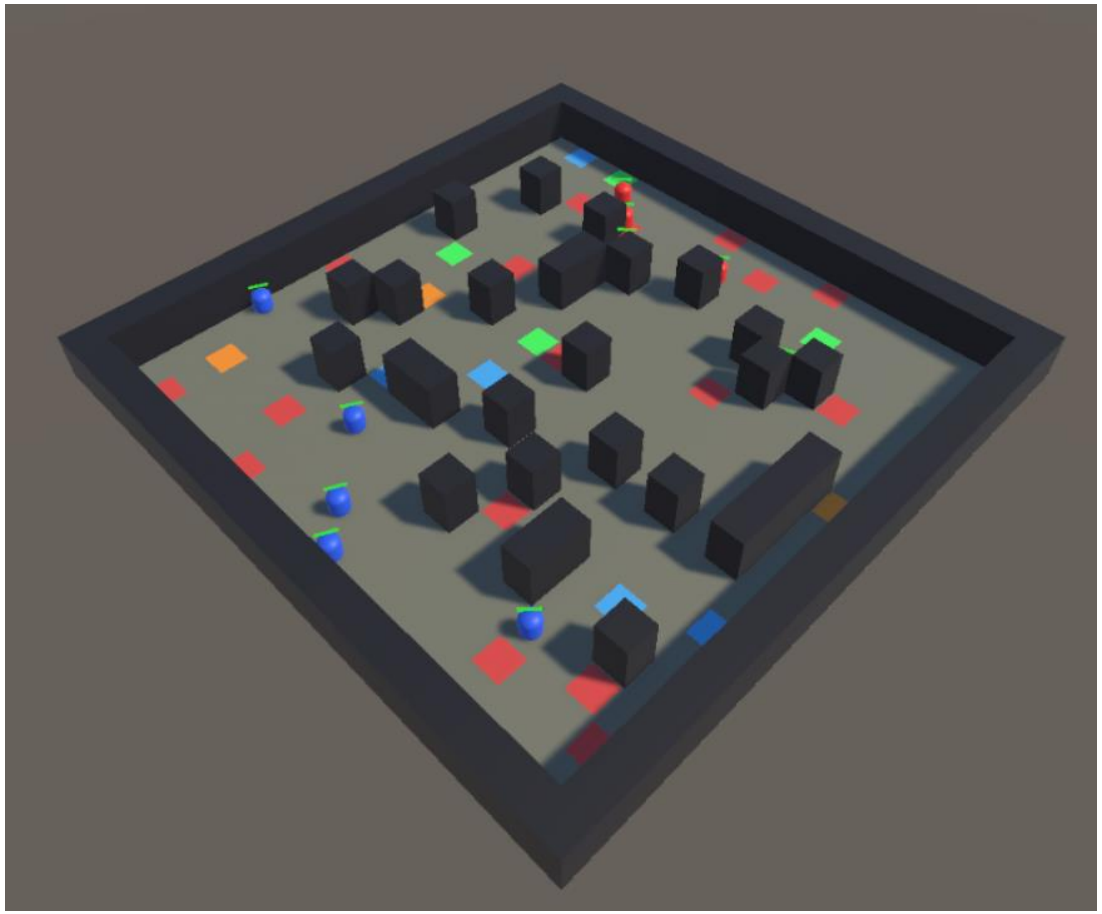


별첨 1. 실제 학습에서 사용된 1단계 셀프플레이(Self-Play) 학습 결과

별첨2



별첨 2. 실제 실험에서 사용된 2단계 협동 학습(Cooperative Training)결과



별첨 3. 실제 테스트한 유니티 3D 데모 환경