JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 26, No. 2, pp. 511-524, Feb. 2025

Check for updates

원거리 군대 간 전투 전략 탐구: 적대적 멀티 에이전트 군사 시뮬레이션에서의 MADDPG와 CTDE 활용

현 기 정[1*] · 윤 병 현[2] · 송 지 원[2] · 조 래 현[2]
[1]서강대학교 메타버스전문대학원 박사과정 [2]서강대학교 메타버스전문대학원 석사과정

Exploring Combat Strategies Between Ranged Armies: Application of MADDPG and CTDE in Adversarial Multiagent Military Simulation

Ki-Jeong Hyun[1*] · Byeong-Hyun Yoon[2] · Ji-Won Song[2] · Rae-Hyun Jo[2]
[1]Ph.D. Student, Graduate School of Metaverse, Sogang University, Seoul 04107, Korea
[2]Master's Course, Graduate School of Metaverse, Sogang University, Seoul 04107, Korea

[요 약]

본 연구는 적대적 멀티 에이전트 군사 시뮬레이션 환경에서 MADDPG와 CTDE를 활용한 원거리 공격 군사 시뮬레이션을 개발하여 에이전트들의 협력적 행동과 전투 전략 학습을 탐구한다. 공격적 성향의 블루 팀과 방어적 성향의 레드 팀으로 구성된 두 팀은 협력적 및 경쟁적 행동을 학습하며, 중앙 집중식 Critic 네트워크는 팀 전체 정보를 활용한 학습을, 분산된 Actor 네트워크는 독립적인 실행을 통해 실시간 의사결정을 수행한다. 실험 결과, 블루 팀의 공격적 전략은 보상 수렴 속도와 승률에서 우위를 보였으나, 레드 팀의 방어적 전략은 보상 설계와 Critic 네트워크의 불안정성으로 인해 학습 속도가 더디게 진행되었다. 본 연구는 보상 구조 설계, 협력적 행동 강화, 전략적 적응성의 중요성을 강조하며, 향후 연구를 위해 보상 체계 개선, 훈련 시나리오 다양화, 강화학습 알고리즘의 확장을 제안한다.

[Abstract]

This study developed a long-range attack military simulation for an adversarial multiagent environment using MADDPG and CTDE to explore cooperative behaviors and combat strategy learning among agents. Two teams—an offensive Blue Team and a defensive Red Team—learned cooperative and competitive behaviors. The centralized Critic network leveraged team-wide information for training, whereas the decentralized Actor networks made real-time independent decisions. Experimental results indicate that the Blue Team's offensive strategy outperformed in terms of reward convergence speed and win rates, whereas the Red Team's defensive strategy exhibited slower learning progress due to the reward design challenges and instability in the Critic network. This study highlights the importance of reward structure design, reinforcement of cooperative behaviors, and strategic adaptability. Future research may focus on improving reward mechanisms, diversifying training scenarios, and extending reinforcement learning algorithms.

색인어 : 적대적 강화학습, MADDPG, CTDE, 멀티 에이전트 강화학습, 군사 시뮬레이션
Keyword : Adversarial Reinforcement Learning, MADDPG, CTDE, MARL, Military Simulation

http://dx.doi.org/10.9728/dcs.2025.26.2.511

Received 08 December 2024; **Revised** 14 January 2025
Accepted 11 February 2025

***Corresponding Author; Ki-Jeong Hyun**
Tel: +82-70-4235-1495
E-mail: vennyhyun@gmail.com

Ⅰ. 서 론

1-1 연구의 배경 및 목적

멀티 에이전트 강화 학습(Multi-Agent Reinforcement Learning, MARL)[1],[2]은 복잡한 의사결정 문제를 해결하기 위한 효과적인 도구로, 여러 에이전트가 상호작용하며 최적의 전략을 학습하는 방식으로 발전하고 있다[3]. 군사 전략에서 병사들의 협력과 적응성은 전투의 중요한 요소로, 이를 강화 학습을 활용한 시뮬레이션 환경에서 재현하고 분석하는 것은 유의미한 연구 주제이다[4]. 본 연구는 이러한 시뮬레이션 환경에서 강화 학습 알고리즘이 복잡한 전술적 의사결정 문제를 어떻게 해결할 수 있는지를 탐구한다.

기존의 군사 시뮬레이션 연구들은 개별 병사의 성능 최적화에 중점을 두고 있다. 하지만, 병사 간 협력적 행동은 전장에서 중요한 전략적 요소로, 이를 강화 학습으로 구현하는 것은 군사 전략의 효율성을 높이는 데 필수적이다[5]. SMAC(StarCraft Multi-Agent Challenge)와 GRF(Google Research Football)와 같은 환경에서 멀티 에이전트 강화학습(MARL)을 활용하여 협력적 행동을 학습하는 데 중점을 두었다. 특히, MADDPG(Multi-Agent Deep Deterministic Policy Gradient)와 CTDE(Centralized Training with Decentralized Execution) 접근법은 중앙 집중 Critic 네트워크와 분산 Actor 네트워크를 활용하여 에이전트 간 협력을 효과적으로 학습할 수 있음을 보여주었다. 이러한 연구들은 협력적 행동 학습의 가능성을 제시했지만, 적대적 환경에서의 전략적 경쟁과 협력을 학습하는 연구는 상대적으로 부족했다.

본 연구는 기존 연구를 확장하여, 적대적 군사 시뮬레이션 환경에서 두 팀 간의 협력적 및 경쟁적 행동을 학습하는 데 초점을 맞춘다. 기존 연구가 단일 팀 내 협력 강화에 초점을 두었다면, 본 연구는 두 팀의 상반된 전략(공격적 vs. 방어적)을 비교함으로써 MARL의 적용 가능성을 더욱 현실적인 시나리오로 확장한다. 특히, 기존 연구의 협력 중심 시뮬레이션 접근법을 보완하여, 적대적 환경에서의 협력과 경쟁 전략의 학습 효과를 실증적으로 분석하고, 강화학습 알고리즘의 실제 적용 가능성을 높이는 데 기여한다.

본 연구는 공격적 성향의 블루 팀과 방어적 성향의 레드 팀으로 구성된 원거리 군사 시뮬레이션 환경을 구축하고, 이를 통해 두 팀의 협력적 행동과 경쟁적 전략 학습의 효율성을 비교 분석한다. 이를 위해 MARL의 대표적 알고리즘인 MADDPG와 CTDE 접근 방식을 적용하였다. 블루 팀은 협력적 공격 전략을 통해 전장을 장악하려는 반면, 레드 팀은 방어적 협력을 통해 생존 가능성을 극대화하는 전략을 택한다.

이 연구의 주요 목적은 다음과 같다:

- 공격적 및 방어적 전략의 학습 효율성을 비교하고, 협력적 행동이 전투 성과에 미치는 영향을 분석한다.
- Critic 네트워크의 설계가 방어적 학습의 성능에 미치는 영향을 평가한다.
- 군사 시뮬레이션 환경에서 협력적 행동을 강화하기 위한 보상 구조를 제안한다.

본 연구는 기존 SMAC와 GRF 환경에서 수행된 연구와는 달리, 적대적 군사 시뮬레이션에서 두 팀 간 협력 및 경쟁 전략을 직접적으로 비교 분석한다는 점에서 차별성을 가진다. 특히, Critic 네트워크의 설계를 통해 방어적 전략의 불안정성을 개선하고, 강화 학습 환경에서 협력적 행동을 학습할 수 있는 구조를 제시한다.

1-2 연구 범위와 방법

본 연구는 MADDPG와 CTDE 알고리즘을 적용하여 중앙 집중 Critic 네트워크를 통해 팀 전체 정보를 학습하고, 분산 Actor 네트워크를 통해 각 에이전트가 독립적으로 행동할 수 있도록 한다. 블루 팀은 적을 빠르게 무력화하는 공격적 전략을, 레드 팀은 거리 유지와 협력적 방어를 통한 생존 전략을 학습한다. 학습 과정은 총 600 에피소드로 진행되며, 학습 속도, 보상 수렴 속도, Critic 손실, 승률 등의 지표를 통해 성과를 평가한다. 그림 1은 연구 모델 구성도이다.

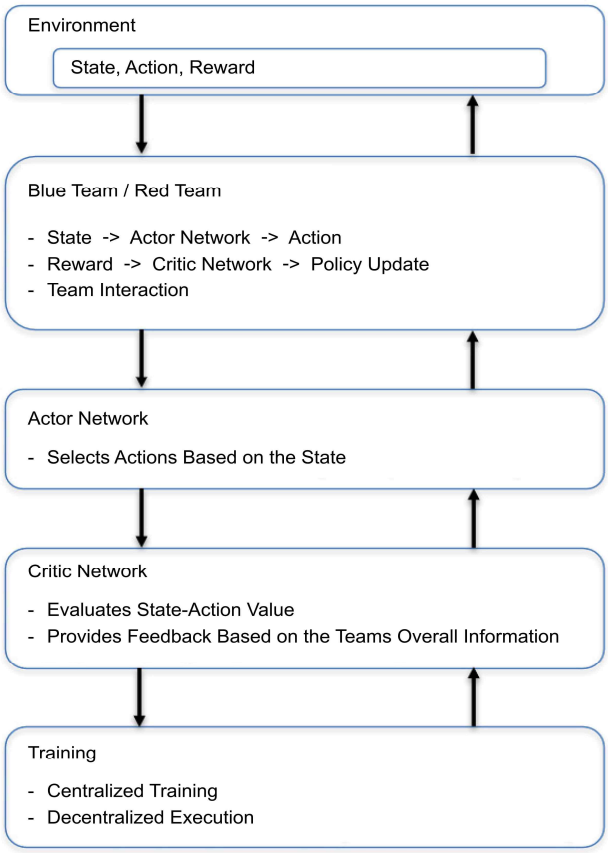

그림 1. 연구 모델 구성도
Fig. 1. Research model diagram

이 논문의 기여점은 다음과 같다:

- **협력적 전투 시뮬레이션 구현** – MADDPG와 CTDE를 활용하여 적대적 환경에서 블루 팀(공격적 성향)과 레드 팀(방어적 성향) 간 협력 및 경쟁 전략을 학습하는 시뮬레이션을 구현하였다. 기존의 협력 중심 연구와 달리, 적대적 상황에서의 전략 학습과 효과를 실증적으로 분석하였다.
- **Critic 네트워크 설계 개선** – 방어적 전략 학습의 효율성을 높이기 위해 Critic 네트워크 구조와 보상 설계를 최적화하였다. 이는 기존 연구의 Critic 불안정성을 개선하고, 방어적 행동의 복잡성을 효과적으로 처리할 수 있도록 설계되었다.
- **공격적 및 방어적 전략 비교** – 두 팀의 상반된 전략 학습 결과를 비교 분석하여, 보상 구조가 전략 성과에 미치는 영향을 정량적으로 평가하였다. 기존 연구에서는 다루지 않은 방어적 전략의 세부 성과와 한계를 도출하였다.
- **적대적 환경에서 강화학습 평가** – 적대적 시뮬레이션 환경에서 MADDPG와 CTDE의 성능을 평가하며, 강화학습 알고리즘의 실제 적용 가능성을 확장하였다. 이는 단순 협력 환경을 넘어선 현실적인 응용 가능성을 제시한다.

II. 배경지식

멀티 에이전트 강화학습(Multi-Agent Reinforcement Learning, MARL)은 에이전트를 다중으로 사용하는 기법으로 단일 에이전트가 해결하기 어렵거나 불가능한 문제를 에이전트간 상호작용을 통해 협력하거나 경쟁하며 학습하고, 이 과정을 통해 공동 목표를 달성하거나 개별적으로 목표를 최적화하는 방법이다[1],[5]. 또한, MARL은 군사 시뮬레이션에서 NPC의 행동을 더욱 현실적이고 예측 불가능하게 만들어 훈련의 효과를 높이는 데 기여한다. 기존 규칙 기반 시스템으로는 구현하기 어려웠던 동적인 전투 환경에서 다수의 에이전트가 상호작용하며 발생하는 복잡한 상황들을 효과적으로 모델링하고 학습할 수 있기 때문이다. MARL을 도입하면 NPC들이 단순한 명령에 따라 움직이는 것이 아니라, 스스로 상황을 판단하고 최적의 행동을 선택하며, 에이전트들과 협력하여 목표를 달성하기 위한 전략을 수립할 수 있다. 이는 훈련 에이전트들에게 더욱 도전적이고 현실적인 환경을 제공하여 전술적 사고 능력과 의사 결정 능력을 향상시키는 데 도움을 준다. 특히, MARL은 팀워크와 협력 행동을 강화하는 데 매우 효과적이다. 다수의 에이전트가 상호 의존적인 관계를 형성하고 함께 학습하며, 이를 통해 팀 전체의 성과를 극대화할 수 있는 전략을 개발할 수 있다. 요약하면, MARL은 군사 훈련 시뮬레이션의 현실성을 높이고, 훈련 효과를 극대화하며, 미래 전장에서 요구되는 협력과 전략적 사고 능력을 함양하는 데 필수적인 기술이다[6]. 이러한 협력을 강화하기 위해서 다양한 방법이 연구되고 있으며, 중앙에서 모든 에이전트를 관리하는 중앙 집중형 방식과 각 에이전트가 독립적인 분산형 방식을 적절히 합쳐 훈련 과정에서 중앙 집중형, 실행 단계에서는 분산형으로 진행되는 CTDE 방식의 연구가 활발하게 진행되고 있다[14].

CTDE(Centralized Training with Decentralized Execution)는 MARL에서 중앙화 된 학습과 분산 실행을 결합한 대표적인 패러다임으로, 값 기반(Value-based), 정책 기반(Policy-based), 융합(Hybrid) 방식으로 구분된다. 값 기반 방법론 QMIX는 개별 에이전트의 Q-값을 중앙화 된 네트워크에서 결합함으로써 팀 전체의 최적화를 도모하며, 정책 기반 방법론의 대표인 MADDPG는 중앙화 된 Critic이 전체 정보를 활용하여 각 에이전트의 정책을 평가하되 실행 단계에서는 독립적으로 행동하도록 설계된다. 융합적 접근법에 해당하는 COMA(Counterfactual Multi-Agent)는 반사실적 기여도(Counterfactual Contribution)를 계산하여 에이전트 간 협력을 강화하는 데 초점을 맞춘다. 이러한 CTDE 접근법은 중앙화 된 학습을 통해 에이전트 간 상호작용을 효과적으로 모델링하면서도 실행 단계에서의 독립성을 보장함으로써, 다양한 환경에서의 효율적이고 확장 가능한 학습을 가능하게 한다[7]. MARL에 LNS(Large Neighborhood Search)을 추가하여 모든 전체 에이전트를 바로 학습시키는 것이 아닌 부분집합으로 이웃을 구성하고 선택된 에이전트를 훈련하는 방식을 통해 강화 학습의 효율을 높이는 방식으로 CTDE를 보완한 MARL-LNS라는 방식을 제안하였으며[15], Zhou외 6인은 기존 CTDE 방식에서 훈련 진행 과정에 에이전트 간 통신 채널을 만들어 서로 통신하는 과정을 추가하여 협력을 진행하다가 이를 점점 줄여 나가고 독립성을 강화하는 방식을 추가한 CADP(Centralized Advising and Decentralized Pruning) 방식을 제안하였다[16]. 이러한 연구들은 모두 기존의 CTDE를 개선하는 방향으로 진행되고 있으며, 동일하게 SMAC(StarCraft Multi-Agent Challenge)과 GRF(Google Research Football)를 통해 멀티 에이전트 환경에서 강화학습을 통한 시뮬레이션이 진행되었으나, 많이 사용된 CTDE와 다르게 다양한 시나리오의 검증을 거친 것이 아니다. 본 연구의 에이전트의 종류와 수, 실험 환경의 차이로 인해 안정성의 문제를 고려하여 본 연구에서는 기존의 CTDE를 사용한다.

MADDPG(Multi-Agent Deep Deterministic Policy Gradient)는 MARL 알고리즘으로, 혼합된 협력-경쟁 환경에서 에이전트 간의 상호작용을 효과적으로 학습할 수 있도록 설계되었다. 기존의 단일 에이전트 강화학습 알고리즘은 멀티 에이전트 환경에서 발생하는 비정상성(non-stationarity) 문제를 다루는 데 한계가 있었지만, MADDPG는 이를 해결하기 위해 중앙 집중 학습과 분산 실행(CTDE) 접근법을 채택하였다. 이를 통해, non-stationarity 문제를 해소하고, CTDE를 통해 학습 시에는 전체적인 팀 목표를 위해 정보를 공유하고, 실행 시에는 각 에이전트가 독립적으로 행동하여 유연성을 확보하였다. Actor-Critic 구조로 Actor는 행동을 선택하고, Critic은 행동의 가치를 평가하여 학습을 안내한다. 또한, 다양한 상황에서

협력과 경쟁을 동시에 학습하여 복잡한 환경에 적응할 수 있다. 이러한 이유로, MADDPG는 군사 시뮬레이션, 멀티로봇 협력, 경제적 에이전트 시뮬레이션 등 다양한 응용 분야에서 높은 성능을 발휘할 수 있는 강력한 MARL 알고리즘이다[8].

Ⅲ. 관련 연구

강화학습을 활용한 시뮬레이션 연구는 다양한 알고리즘을 적용하여 시뮬레이션 환경에서 에이전트의 성능을 평가하는 방식으로 발전하고 있다. PPO(Proximal Policy Optimization), DQN, VPG(Vanilla Policy Gradient)[17] 등 다양한 알고리즘이 멀티 에이전트를 활용한 시뮬레이션에서 그 성능 테스트를 받았으며, 각 알고리즘의 성과를 개별적인 실험 환경에서 비교하거나, 기존 알고리즘의 개선된 성능을 증명하는 방식으로 연구가 진행되고 있다[1],[9]-[11],[17]. 적대적 강화학습(Adversarial Reinforcement Learning)은 방해하는 적대적인 에이전트(adversary) 와 특정 목표를 수행하는 에이전트(protagonist)를 동시에 학습시키는 방식으로 적대적 에이전트가 방해를 하면 목표 수행 에이전트가 이를 극복하는 방향으로 진행되며 방해와 극복을 교대로 최적화를 진행하여 방해 속에서도 목표 수행 에이전트가 성공할 수 있는 환경을 만드는 것이다[12]. Lantao Yu 외 2인은 강화학습에서는 보상 함수가 중요하고 만일 잘못 정의된 경우 에이전트가 예상하지 못한 행동을 하며, 멀티 에이전트 환경에서는 에이전트 간의 상호작용이 있기에 합리적인 행동 정의가 복잡하고 보상 함수 설계가 복잡하다는 문제를 제기하였다. 이에 멀티 에이전트 환경에서 적대적 역 강화학습(Multi-Agent-Adversarial Inverse Reinforcement Learning, MA-AIRL)을 제안하고 다양한 환경 시나리오를 적용하였고 이 중 한 가지는 적대적 보상 학습 방식을 사용하여 적대적 훈련을 통해 보상 함수를 얻는 실험 환경으로 진행하였다. 에이전트는 공유된 환경에서 각자 다른 보상 함수를 가지고 있으며 에이전트 간 경쟁적인 Competitive Keep-Away 태스크를 구성하여 한 에이전트는 목표 지점에 도달하려고 시도하며 이에 다른 에이전트는 목표 달성을 방해 시도하는 방식으로 실험하였다. 다만, 해당 연구는 MA-AIRL 알고리즘을 제안하고 이를 검증하는 방식으로 적대적인 상황을 가정하고 이 상황에 따른 보상 함수를 사용하였다. 그리고, 하나의 멀티 에이전트에서 각 에이전트가 적대적인 상황이지만 본 논문에서는 군대 간 서로 다른 멀티 에이전트가 적대적인 상황에서 강화학습을 진행하고, 개별 에이전트의 수치 비교가 아닌 멀티 에이전트 간 직접 비교라는데 차별점이 존재한다[18].

CTDE는 멀티 에이전트 환경에서의 학습 방법으로 특히 로봇 분야에서 활용되고 있다. Leroy et al.은 SMAC 환경에서 CTDE, QMIX, MAVEN, QVMix 방법을 활용해 협력 및 경쟁을 시뮬레이션하며, 대칭적 상황을 가정한 연구를 진행하였다[22]. 반면, 본 논문에서는 대칭적 상황에 대한 별도의 가정 없이도 적용 가능한 방법을 제시하며, 군사 시뮬레이션에 특화된 보상 설계를 적용하였다. 또한, Zhou et al.은 다중 로봇 시뮬레이션에서 장애물 회피와 같은 과제를 수행하기 위해 리더-팔로워(leader-following) 방식을 기반으로 CTDE를 활용하였다. 본 연구는 이러한 접근과 달리 에이전트 간 협동을 평가할 수 있는 구조를 설계한 점에서 차별성을 가진다[19].

MADDPG는 멀티 에이전트 강화학습 환경에서 비정상성 문제를 해결하고 협력과 경쟁적 시나리오에서 높은 성능을 보이는 알고리즘으로, 다양한 전투 시뮬레이션에서 활용되고 있다[8]. Kouzeghar 외 3인은 MADDPG를 변형하여 UAV (Unmanned Aerial Vehicle) 군집이 복잡한 환경에서 목표를 탐지하고 추적하며, 탐색과 활용 간의 균형을 유지하도록 보로노이 기반 보상 정책을 설계하였다. 시뮬레이션 환경은 2D 공간에서 랜덤 장애물과 빠르게 움직이는 목표들이 포함된 동적인 환경으로 구성되었으며, UAV 간 역할 분담(추적자와 정찰자)을 통해 효율적인 탐색과 추적 성능을 입증하였다. 실제 드론 실험에서도, 6m×6m의 물리적 환경에서 UAV 군집이 목표를 성공적으로 포획하고 협력적 행동을 학습하는 결과를 보였다[20]. Chen 외 1인은 해상 전투 시뮬레이션 환경에서 MADDPG 기반의 멀티 에이전트 대립 알고리즘을 제안하였다. 이 연구는 350km×350km의 해상 공간을 시뮬레이션 환경으로 설정하였으며, 붉은 팀(공격 측)과 파란 팀(방어 측)으로 나뉘어 진행되었다. 붉은 팀은 방공 시스템을 회피하며 지정된 목표를 파괴해야 하고, 파란 팀은 이를 방어하는 임무를 수행한다. 이 알고리즘은 LSTM(Long Short-Term Memory)을 통합하여 과거 상태 정보를 활용하며, 대규모 상태 공간에서도 높은 수렴성과 효율성을 보였다. 또한, 모방 학습을 결합하여 초기 학습 속도를 개선하고, 평균 승률 90%를 달성하며 전술적 유연성과 강건성을 입증하였다[21]. Zhao 외 6인은 MW-MADDPG(Meta-Weight MADDPG) 알고리즘을 제안하여 UAV 군집의 협력적 의사결정 문제를 해결하였다. POMDP(Partially Observable Markov Decision Process) 기반으로 설계된 시뮬레이션 환경에서 UAV 군집은 제한된 관찰 정보만으로 임무를 수행해야 했으며, 환경의 동적인 특성을 반영하였다. 이 알고리즘은 Reward-TD 기반 경험 리플레이와 경험 망각 메커니즘을 도입하여 데이터를 효율적으로 학습하고, 새로운 환경에서도 안정적인 학습을 가능하게 하였다. 시뮬레이션 환경은 복잡한 임무 목표와 장애물이 포함된 240km×240km 공간으로 구성되었으며, 실험 결과 학습 속도와 강건성이 기존 방법 대비 크게 향상되었음을 확인하였다[13]. 이와 같은 연구들은 다양한 시뮬레이션 환경에서 MADDPG가 높은 적응성과 학습 효율성을 제공하며, 멀티 에이전트 시스템의 협력적 행동과 전략적 의사결정에 중요한 기여를 하고 있음을 보여준다. 각 연구는 시뮬레이션 환경의 복잡성과 특성을 고려하여 알고리즘의 확장 가능성을 입증하며, 실제 전투 시뮬레이션에서도 실용적으로 활용될 수 있음을 보여준다.

Ⅳ. 제안 알고리즘

Algorithm 1은 MARL을 통해 두 팀의 에이전트가 협력적 또는 경쟁적 전략을 학습하는 전투 시뮬레이션 환경을 구현하고, 공격적 및 방어적 전략의 성과를 분석하는 데 목적이 있다. 각 팀의 에이전트들은 협력하고 행동을 적응시켜 각자의 역할에 맞는 보상을 최대화하려고 한다. MADDPG 와 CTDE 접근 방식을 사용하여, 팀 단위의 협력적 학습과 개별 에이전트의 독립적 실행을 동시에 구현하였다.

표 1. 전투 전략 알고리즘
Table 1. Combat Strategies Algorithm

Initialize environment with env_size and N_{team} agnets per team Initialize MADDPG agents for both teams (Blue team, Red team) with learning rates lr_{actor} and lr_{critic} Set each agent's initial health $HP = 100$, movement speed $SOLDIER\_SPEED$, and projectile speed $PROJECTILE\_SPEED$

Reset: Reset environment and agent positions Initialize agent rewards $r_{i=0}$ and projectiles

Action Selection: Select action $a_i = \pi_\theta(s_i)$ using actor network for each agent in the blue team Select action $a_j = \pi_\theta(s_j)$ using actor network for each agent in the red team

Environment Update: Update position of each agent:

$$pos_i(t+1) = pos_i(t) + a_i \times SOLDIER\_SPEED$$

where a_i is the selected action, and pos_i represents the position of agent i. Update positions of all projectiles:

$$pos_p(t+1) = pos_p(t) + PROJECTILE\_SPEED \times d$$

where $\hat{d}$ is the unit vector in the direction of the target. Handle collisions between agents and projectiles:

$$HP_i(t+1) = HP_i(t) - \Delta HP$$

if hit by a projectile

Firing Projectiles: Fire projectile from agent i targeting closest enemy:

$$target = \arg\min_{j \in enemies} \| pos_i - pos_j \|$$

update $last\_fire\_time$

Reward Calculation: Calculate reward r_i for each agent in the blue team based on:

$$r_i = -\alpha d_{(i,enemy)} + \beta \sum_{j \neq i} II(distance(i,j) < d_{coop}) + r_{success}$$

where $d_{i,enemy}$ is the distance to the nearest enemy, and $r_{success}$ represents reward for a successful attack. Calculate reward r_j for each agent in the red team based on:

$$r_{j=+}\alpha d_{j,enemy} + \beta \sum_{k \neq j} II(distance(j,k) < d_{coop}) + r_{\text{defense}}$$

where $d_{j,enemy}$ is distance to the nearest enemy, and r_{defense} represents reward for successful defense.

Store $(s_i, a_i, r_i, s_i', done)$ in replay buffer for each agent i in both teams

Step: Perform a step in the environment with given actions:

- Fire Projectiles: At time t, if $t - last\_fire\_time \geq T_{fire}$, fire projectiles from agents targeting the closest enemy:
  $target = \arg\min_{j \in enemies} \| pos_i - pos_j \|$
  Update $last\_fire\_time$
- Blue Team Agent Movement: For each agent i in blue team:

a. Find the closest enemy agent in the red team, $closest\_enemy$

b.if $\| pos_i - closest\_enemy.position \| < 100$:

- Find nearby allies within distance 100
- If nearby allies exist, calculate:

$$a_i \leftarrow \frac{closest\_enemy.position - pos_i}{\| closest\_enemy.position - pos_i \|}$$

- Move agent i according to a_i

■ Continue to the next agent

c. Otherwise, move agent i according to the selected action a_i

- Red Team Agent Movement: For each agent j in red team:

a. if $HP_j \leq 50$:

■ Find the closest enemy agent in the blue team, $closest\_enemy$

■ if $closet\_enemy$ exists, calculate the direction away from the enemy:

$direction\_away \leftarrow pos_j - closest\_enemy.position$

■ if $\| direction\_away \| > 0$, update:

$$a_j \leftarrow \frac{direction\_away}{\| direction\_away \|}$$

b. Otherwise, find nearby weak allies within distance 100.

c. If nearby weak ally exists calculate:

$$a_j \leftarrow \frac{target_{ally}.position - pos_j}{\| target_{ally}.position - pos_j \|}$$

d. Find nearby allies within distance 100

e. If nearby allies exists, calculate the average position of nearby allies and agent j:

$$avg\_position \leftarrow \frac{(\Sigma_{k \in allies + \{j\}} \ pos_k)}{|allies + \{j\}}$$

f. Calculate the direction to the average position:

$$a_j \leftarrow \frac{\text{direction\_to\_avg}}{\| direction\_to\_avg \|}$$

g. if $\| direction\_to\_avg \| > 0$, update.

h. Move agent j according to a_j.

- **Projectile** Movement and Collision Handling: Move projectiles and handle collisions Handle agent collisions
- **Agent** Removal: Remove agents that have reached their goal or are dead from blue team and red team
- **Reward Calculation**: Calculate rewards r_i for blue team and r_j for red team using centralized reward function

○ For each agent i in blue team:

■ If $HP_i > 0$ and

$\| pos_i - closest_{enemy}.position \| \leq 100 : r_i \leftarrow r_i + 20$ (Additional reward for attempting an attack)

- **Training Update**: Update critic network by minimizing TD error:

$$L_\phi = E\ [(Q_\phi(s_i, a_i) - (r_i + \gamma Q_\phi(s_i', a_i')))^2]$$

where Q_ϕ is the target critic network. Update actor network to maximize expected return:

$$J(\theta) = E[Q_\phi(s_i, \pi_\theta(s_i))]$$

The actor network is updated to maximize the value Q_ϕ as estimated by the Critic.

Calculate and store training metrics (e.g., critic loss, average reward)

Output training progress, win/loss statistics, and reward metrics

4-1 알고리즘의 주요 단계

1) 초기화 단계

알고리즘은 환경 크기, 팀별 에이전트 수, 학습률, 보상 할인 계수 등 기본 매개변수를 초기화하는 것으로 시작한다. 두 팀의 MADDPG 에이전트를 초기화하며, 각 에이전트의 체력, 이동 속도, 발사체 속도 등의 매개변수를 설정한다.

2) 환경 재설정 및 초기화

환경을 재설정하여 에이전트의 위치를 초기화하고, 보상을 0으로 설정하며, 발사체 정보를 초기화한다.

3) 행동 선택

각 에이전트는 현재 상태를 기반으로 Actor 네트워크를 사용하여 행동을 선택한다. 블루 팀과 레드 팀 모두 학습된 정책을 사용해 각 팀의 에이전트들이 독립적으로 행동을 결정한다.

4) 환경 업데이트

에이전트들은 선택된 행동에 따라 이동하며, 이에 따라 위치를 업데이트한다. 발사체는 목표를 향해 이동하며, 위치가 업데이트된다. 에이전트와 발사체 간의 충돌이 발생하면 해당 에이전트의 체력이 감소한다.

5) 스텝 함수

- **발사체 발사:** 정해진 시간 간격이 지나면 에이전트들은 가장 가까운 적을 향해 발사체를 발사한다.
- **블루 팀 이동:** 블루 팀 에이전트들은 가장 가까운 적을 공격하려고 시도한다. 근처에 아군이 있을 경우 협력하여 더 강력한 공격을 수행한다. 근처에 아군이 없을 경우,

에이전트들은 선택된 행동에 따라 독립적으로 이동한다.

- **레드 팀 이동:** 레드 팀 에이전트들은 특히 체력이 일정 수준 이하로 떨어졌을 때 방어적인 행동을 한다.에이전트들은 체력이 낮을 때 적과의 거리를 유지하려 하고, 약한 아군을 지원하며 더 안전한 위치로 이동한다.
- **발사체 이동 및 충돌 처리:** 발사체가 환경 내에서 이동하고, 발사체와 에이전트 간의 충돌을 처리한다.
- **에이전트 제거:** 목표에 도달했거나 체력이 0이 된 에이전트는 팀에서 제거된다.
- **보상 계산:** 각 에이전트의 보상은 중앙 집중식 기준에 따라 계산되며, 예를 들어 적과의 거리나 공격 성공 여부 등을 고려한다.

6) 훈련 업데이트

- **Critic 네트워크 업데이트:** Temporal Difference (TD) 오차를 최소화하여 Critic 네트워크를 업데이트한다. 타겟 Critic 네트워크는 상태와 행동 쌍에 기반해 예상되는 미래 보상을 추정한다.
- **Actor 네트워크 업데이트:** Critic 네트워크의 평가를 사용해 예상 반환값을 최대화하도록 Actor 네트워크를 업데이트한다.
- **학습 성과 기록:** 크리틱 손실, 평균 보상 등 학습 성과를 기록하여 훈련 성능을 모니터링한다.

7) 협력 행동과 팀 전략

- **블루 팀:** 블루 팀은 공격적인 전략을 채택한다. 에이전트들은 적을 공격할 때 특히 근처에 아군이 있는 경우 협력하여 공격 성공률을 높이고 적의 체력을 빠르게 감소시키려 한다.
- **레드 팀:** 레드 팀은 방어에 중점을 둔다. 체력이 낮은 에이전트들은 적으로부터 거리를 벌려 생존 확률을 높이려 하고, 약한 아군을 보호하기 위해 더 안전한 위치로 이동하며 방어적인 행동을 강화한다.
- **보상 설계:** 각 팀의 에이전트들은 자신의 행동에 따라 보상을 받으며, 바람직한 행동에는 추가 보상이 부여된다. 블루 팀 에이전트들은 근처의 적을 공격하는 경우 추가 보상을 받아 공격적인 행동을 유도한다. 이러한 보상 구조는 에이전트들이 팀 목표에 맞는 효과적인 행동을 학습하도록 설계되었다. 예를 들어, 블루 팀의 경우 공격적인 행동을, 레드 팀의 경우 보호적인 행동을 강화한다.
- **학습과 협력:** MADDPG는 각 에이전트가 중앙 집중식 Critic 네트워크와 자신의 Actor 네트워크를 사용해 협력적 행동을 학습할 수 있도록 한다. Critic 네트워크는 전체 상태 정보를 사용해 에이전트들의 행동을 평가하므로, 에이전트 간의 협력 능력을 향상시킨다.

4-2 CTDE 적용

CTDE 접근법은 중앙집중식 Critic 네트워크를 사용하여 팀 내 모든 에이전트의 행동을 평가하고, 각 에이전트는 분산된 Actor 네트워크를 통해 독립적으로 행동을 선택한다.

1) Critic 네트워크 업데이트 (중앙 집중 학습)

각 에이전트의 Critic 네트워크는 팀 내 모든 에이전트의 상태(state)와 행동(action) 정보를 활용하여 학습된다. Critic 네트워크의 손실 함수는 다음과 같이 정의된다:

$$L_{\phi} = E[(Q_{\phi}(s,a) - y)^2] \quad (1)$$

수식 (1)에서,

- $Q_{\phi}(s,a)$: Critic 네트워크에서 현재 상태 s 와 모든 에이전트의 행동 a 에 대한 추정값
- $y = \gamma + \gamma Q_{\phi'}(s',a')$: 타겟 값으로 보상 γ, 다음 상태 s', 그리고 다음 행동 a' 에 대한 타겟 Critic Value의 할인 합으로 구성됨
- ϕ : Critic 네트워크의 파라미터
- ϕ' : 타겟 Critic 네트워크의 파라미터

Critic 네트워크는 모든 에이전트의 정보를 중앙에서 학습하여, 에이전트 간 협력 행동과 상호작용을 효율적으로 학습할 수 있도록 지원한다.

2) Actor 네트워크 업데이트 (분산 실행)

Actor 네트워크는 각 에이전트별로 개별적으로 학습된다. Actor 네트워크는 자신의 상태를 기반으로 행동을 선택하며, Critic 네트워크로부터 받은 피드백을 바탕으로 업데이트된다. Actor 네트워크의 목표는 Critic 네트워크가 평가한 Q 값을 최대화하는 것이다. Actor 네트워크의 손실 함수는 다음과 같이 정의된다:

$$J(\theta_i) = E(Q_{\phi}(s,\pi_{\theta_i}(s_i)] \quad (2)$$

수식 (2)에서,

- $Q_{\phi}(s,\pi_{\theta_i}(s_i))$: Critic 네트워크가 현재 상태 s 와 Actor 네트워크의 정책 π_{θ_i}가 선택한 행동 a_i에 대해서 추정한 값
- θ_i : Actor 네트워크의 파라미터
- 정책 그래디언트 업데이트 : Actor 네트워크는 Critic 네트워크로부터 받은 피드백을 바탕으로 Gradient Ascent 방식으로 업데이트된다:

$$\theta_i \leftarrow \theta_i + \alpha \nabla_{\theta_i} J(\theta_i) \quad (3)$$

수식 (3)에서,

- α : 학습률
- $\nabla_{\theta_i} J(\theta_i)$: 손실함수 $J(\theta_i)$의 θ_i에 대한 그래디언트

이 접근법은 Actor 네트워크가 각 에이전트에 대해 독립적으로 학습하도록 보장하며, 이를 통해 분산된 실행을 실현한다.

정리하자면, 이 알고리즘은 팀 기반의 멀티 에이전트 환경에서 에이전트들이 협력과 경쟁을 통해 최적의 전략을 학습하도록 돕는 것을 목표로 한다. 블루 팀은 공격적인 전략을 통해 협력된 공격을 강조하고, 레드 팀은 방어적 행동과 협력을 통한 생존을 강조한다. 중앙 집중식 훈련과 분산 실행을 통해 에이전트들은 팀 성과를 극대화하는 최적의 전략을 학습하며, MARL 시나리오에서 협력의 중요성을 보여준다.

V. 실험 및 분석

그림 2의 전투 시뮬레이션 환경은 300×300 크기이며, 블루 팀(공격 전략)과 레드 팀(방어 전략)으로 구성된 각 팀은 5개의 에이전트로 이루어졌다. 각 에이전트는 MADDPG를 사용하여 중앙 집중적 Critic으로 팀 전체의 상태를 학습하며, 분산된 Actor를 통해 개별 행동을 결정한다. 블루 팀은 높은 학습률과 낮은 할인율을 사용해 공격적인 전략을 유도하고, 레드 팀은 낮은 학습률과 높은 할인율을 통해 방어적인 전략을 학습하였다.

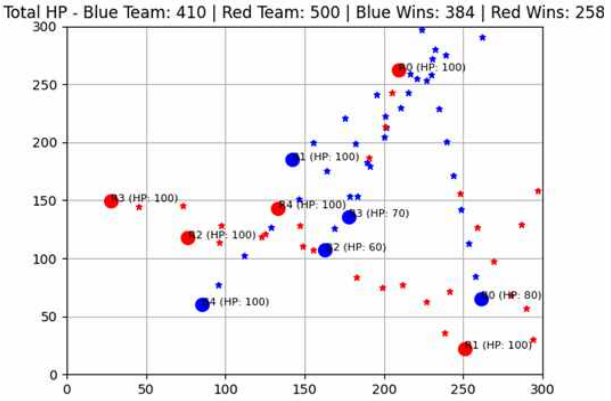

그림 2. 전투 환경
Fig. 2. Combat environment

보상 구조는 팀의 전략을 반영하여 설계되었다. 블루 팀은 적과의 거리를 줄이며 공격적인 행동을 취할 때 보상을 얻고, 레드 팀은 적과의 거리를 유지하거나 늘리는 방어적 행동에 보상을 받는다. 팀 내 에이전트 간 근접할 경우 협력 보상이 추가된다. Actor와 Critic 네트워크는 각각 은닉층 2개(64 유닛, ReLU)를 적용해 학습 안정성을 높였다.

총 600 에피소드 동안 학습을 진행하며, Critic Loss, Critic Value, 평균 보상 등을 기록하고 시각화하였다. 이를 통해 각 팀의 전략적 학습 성과를 평가하고, 협력적 학습의 중요성과 공격적 및 방어적 전략의 효과를 분석하였다.

표 2의 하이퍼 파라미터들은 전투 시뮬레이션 환경에서 각 팀이 공격적 혹은 방어적 전략을 학습하도록 설계되었다. 블루 팀은 공격적 학습을 유도하기 위해 상대적으로 높은 학습률을 사용하고, 레드 팀은 방어적 학습을 위해 낮은 학습률과 높은 감마 값을 사용하고 있다. 또한 CTDE 접근 방식을 활용하여, Critic은 팀 전체의 상태를 학습하지만, Actor는 분산된 방식으로 실행한다. 이러한 하이퍼파라미터 설정은 에이전트들이 전투 상황에서 협력적 혹은 경쟁적 행동을 효과적으로 학습하는 데 도움을 준다.

표 2. 전투 에이전트 하이퍼파라미터
Table 2. Combat agent hyperparameters

Hyperparameter	Value
Environment size	300×300
Agent # in team	5
Projectile Speed	10
Agent Movement Speed	1
Agent HP(Health Point)	100
Actor Learning Rate (lr_actor)	Blue Team: 0.0015
	Red Team: 0.0003
Critic Learning Rate (lr_critic)	Blue Team: 0.0015
	Red Team: 0.0008
Gamma	Blue Team: 0.92
	Red Team: 0.95
Batch_Size	32
Update Repetition #	5
Fire Interval	3.0 (seconds)
Co-op Compensation Distance	up to 100
# of Episodes	600
Distance of Projectile Hit	1.5
Neural Network Architecture(Actor)	Dense (64, ReLU) x 2, Output Layer (Softmax)
Neural Network Architecture(Critic)	Dense (64, ReLU) x 2, Output Layer (Linear)
Centralized Critic	Use Entire Team Status

표 3의 계산 방식은 블루 팀과 레드 팀의 상반된 전투 전략을 모델링하는 데 핵심적인 역할을 한다. 블루 팀은 적을 공격하고 전장을 장악하는 데 중점을 두며, 레드 팀은 적과의 거리를 유지하고 방어적인 위치를 고수하는 데 초점을 맞춘다. 이러한 전략적 목표를 강화 학습을 통해 달성하기 위해 수식적으로 정의된 보상 구조와 Critic/Actor 네트워크의 업데이트 방식을 사용하여, 각 팀이 목표한 전략을 최적화할 수 있도록 설계되었다.

표 3. 전투 에이전트의 계산 방식
Table 3. Calculation method of combat agents

Factor	Equation
Agent Status (s_i)	$s_i = [x_i, y_i, HP_i]$
Behavior Selection (a_i)	$a_i = \pi_\theta(s_i)$
Reward Function (r_i)	$r_i = r_d + r_c + r_s$
Distance Reward (r_d)	$r_d = r_d \begin{cases} -ad & for\ BlueTeam \\ ad & for\ RedTeam \end{cases}$
Co-op Reward (r_c)	$r_c = \beta \sum_{j \neq i} II(dist(i,j) < d_{coop})$
Critic Loss (L_ϕ)	$L_\phi = E\ [(Q_\phi(s,a) - y)^2],$ $(where\ \ y = \gamma + \gamma Q_\phi(s', a'))$
Critic Network Update	$\phi \leftarrow \phi - \eta_{critic} \nabla_\phi L_\phi$
Actor Loss ($J(\theta)$)	$J(\theta) = E[Q_\phi(s, \pi_\theta(s))]$
Actor Network Update	$\theta \leftarrow \theta + \eta_{actor} \nabla_\theta J(\theta)$
ProjectilePosition Update	$(x_p, y_p) \leftarrow (x_p, y_p) +,$ $PROJECTILE\_SPEED \times \hat{d}$
Reward Distribution(c_j)	$c_j \leftarrow c_j + \sum_{sk \in S_{selected}}^{s_j} \times R_{total}$

다음 그래프를 기반으로 블루 팀과 레드 팀의 강화 학습의 결과와 의미를 평가하고 분석한다. 각 그래프는 Critic Loss, Critic Value, 보상, 평균 이동 거리, 전투 후 잔여 HP와 승패 횟수를 보여준다. 각 팀의 성과를 비교하면서 학습 과정과 전략의 효과를 살펴본다.

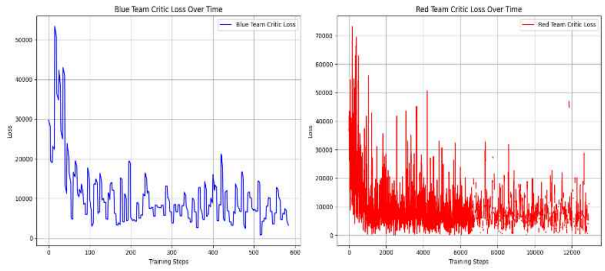

그림 3. 평가 손실
Fig. 3. Critic loss

그림 3의 블루 팀 Critic Loss와 레드 팀 Critic Loss 그래프는 학습 과정 동안 Critic 네트워크가 에이전트의 예측 오차를 얼마나 줄였는지 보여준다. 블루 팀의 Critic Loss는 초기에는 높은 값에서 시작해 빠르게 감소하고, 이후에는 안정적인 수준을 유지한다. 이는 블루 팀의 에이전트들이 주어진 환경에서의 보상을 효과적으로 예측하는 법을 빠르게 학습하고 있는 것을 의미한다. 레드 팀의 Critic Loss는 블루 팀보다 더 많은 변동성을 보이면서 점차 감소한다. 이러한 차이는 레드 팀의 전략이 블루 팀의 공격에 대응하기 위한 더 복잡한 정책을 요구하기 때문에 학습 속도가 더 느리게 나타난 것으로 해석할 수 있다.

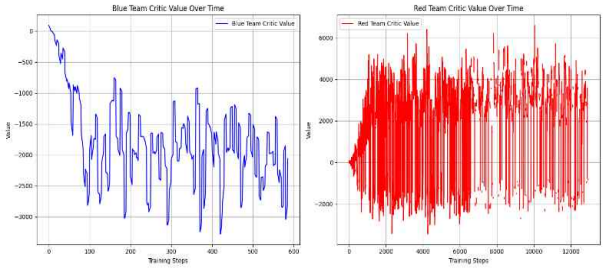

그림 4. 크리틱 값
Fig. 4. Critic value

그림 4의 블루 팀 Critic Value와 레드 팀 Critic Value 그래프는 각 팀의 Critic 네트워크가 예측하는 상태-행동 쌍의 가치를 보여준다. 블루 팀은 Critic Value가 점차적으로 감소하는 경향을 보이는데, 이는 블루 팀의 에이전트가 공격적 행동을 취할 때 리스크가 크다는 것을 의미한다. 이 과정에서 보상의 불확실성이 높아진 것으로 볼 수 있다. 반면 레드 팀은 변동 폭이 더 크고 전반적으로 높은 Critic Value 값을 유지하는데, 이는 레드 팀이 방어적 전략을 통해 에이전트의 생존을 목표로 하고 있으며, 그에 따른 보상이 더 일관되게 유지되고 있는 것을 의미한다.

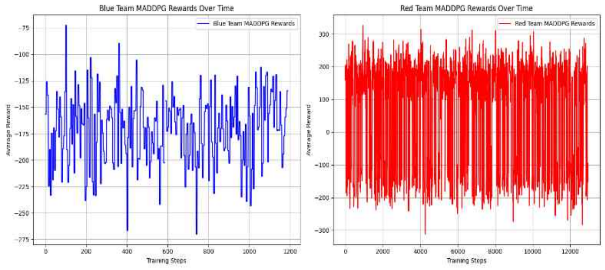

그림 5. MADDPG 보상
Fig. 5. MADDPG rewards

그림 5의 MADDPG Rewards 그래프는 각 에이전트가 얻는 평균 보상을 나타낸다. 블루 팀의 경우 보상의 변화가 상당히 큰데, 이는 블루 팀의 공격적인 전략이 보상에 영향을 많이 미치는 동적인 환경임을 시사한다. 공격 전략은 적과의 거리, 적의 체력 등 다양한 요소에 따라 큰 영향을 받기 때문에 보상이 매우 변동적인 패턴을 보인다. 레드 팀의 보상은 더 높은 변동성을 가지며, 보상 역시 공격 전략에 비해 안정적이지 않는 모습을 보여준다. 레드 팀이 방어적인 전략을 택하지만, 보상 변동성이 큰 이유는 블루 팀의 공격이 강해짐에 따라 방어에 실패할 가능성이 높기 때문으로 보인다.

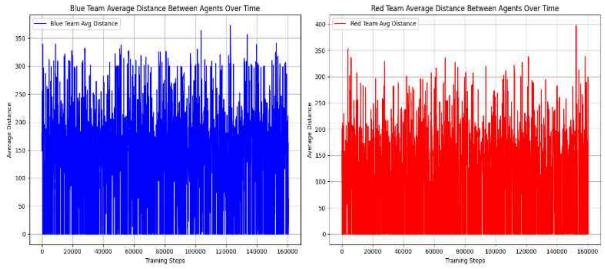

그림 6. 평균 거리
Fig. 6. Average distance

그림 6의 Average Distance Between Agents 그래프는 각 팀의 에이전트들 간의 평균 거리를 나타낸다. 블루 팀과 레드 팀 모두 에이전트들 간의 평균 거리는 크게 변동하고 있다. 블루 팀은 공격을 위해 적에게 접근하려는 전략을 사용하기 때문에 종종 거리 값이 낮아진다. 반면 레드 팀은 방어적인 전략을 택하고 있어 적에게 거리를 유지하려 하며, 그 결과 거리가 더 크게 나타날 수 있다. 이러한 거리는 협력적 방어의 일부로 해석될 수 있다.

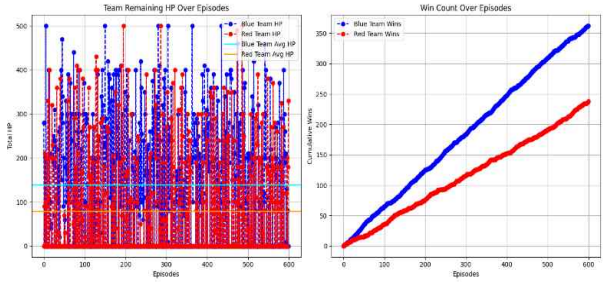

그림 7. 각 팀의 남은 체력과 승리 횟수
Fig. 7. Team remaining HP and win count

그림 7의 Team Remaining HP는 각 에피소드 종료 시점에서 각 팀의 총 체력을 나타내며, 팀이 얼마나 효과적으로 생존했는지를 보여준다. 블루 팀의 잔여 체력 그래프는 공격적인 전략으로 인해 변동성이 크다. 이는 많은 에피소드에서 블루 팀이 공격적인 전술을 사용하지만, 그로 인해 체력을 소진하는 경우가 많다는 것을 의미한다. 레드 팀은 방어적 성향을 통해 생존을 목표로 하고 있음에도 불구하고, 에피소드에서 더 적은 체력을 보유하고 있음을 알 수 있다. 방어적 전략을 통해 생존하려고 노력했지만, 블루 팀의 지속적인 공격에 대응하는 데 어려움을 겪고 있다. 방어만으로는 승리를 보장할 수 없으며, 공격과 방어의 균형이 필요함을 보여준다. Win Count Over Episodes 그래프는 각 팀의 누적 승리 횟수를 보여준다. 블루 팀은 초기부터 우세를 보이며 꾸준히 승리 횟수를 쌓아 나가는 반면, 레드 팀은 방어적인 전략으로 인해 승리 횟수가 상대적으로 적다. 블루 팀의 공격력이 성공적으로 적을 무력화하는 경우가 더 많다는 것을 나타낸다.

종합적으로 보면, 블루 팀은 공격적인 전략을 취하며, 높은 Critic Loss 감소와 큰 보상 변동성을 보인다. 이는 블루 팀의 공격적인 성향이 보상에 큰 영향을 주고 있으며, 매번 적과의 전투에서 상황이 크게 달라질 수 있다는 것을 시사한다. 레드 팀은 방어적인 전략을 사용하며, Critic Value가 더 안정적으로 높은 경향을 보인다. 하지만 승리 횟수에서는 블루 팀에 밀리는 모습을 보이는데, 이는 방어적인 전략이 적의 공격을 견디는 데는 유리하지만, 전반적인 전투에서 승리로 이어지기에는 어려움이 있다는 것을 의미한다. 각 팀의 에이전트들 간의 협력은 거리 유지 그래프에서 잘 나타나며, 블루 팀의 에이전트들은 공격 시 뭉쳐서 움직이는 반면, 레드 팀은 서로 거리를 두고 방어를 강화하려는 모습을 보인다. 이러한 분석 결과는 멀티 에이전트 환경에서 협력 및 경쟁 전략의 효율성을 평가하는 데 중요한 인사이트를 제공한다. 또한, 전투 시뮬레이션에서 공격과 방어의 균형이 중요하며, 공격적인 접근이 높은 승률을 기록할 수 있는 효과적인 방법일 수 있음을 시사한다.

Ⅵ. 논의 및 향후 연구 방향

본 연구는 MARL을 기반으로 공격적 성향의 블루 팀과 방어적 성향의 레드 팀이 서로 협력하며 경쟁하는 전투 시뮬레이션 환경을 구축하고, 이를 MADDPG와 CTDE 접근을 통해 학습시키는 방법을 제안하였다. 실험 결과를 바탕으로 연구의 주요 시사점을 논의하고, 향후 연구에서 해결해야 할 점과 개선 방향을 다음과 같이 제시한다.

6-1 주요 논의 점

1) 공격적 전략의 우위와 방어적 전략의 한계

실험 결과, 블루 팀(공격적 성향)의 승리 비율과 보상 수렴 속도가 레드 팀(방어적 성향)보다 우수한 것으로 나타났다. 이는 공격적 행동이 강화학습 초기 단계에서 단기적인 성과를 더 효과적으로 달성할 수 있음을 시사한다. 특히, 블루 팀의 Critic 네트워크는 비교적 빠르게 손실 값이 감소하여 안정적으로 수렴하였으며, 보상 체계의 설계가 공격적 행동을 더 강하게 강화한 결과로 해석할 수 있다. 반면, 레드 팀의 Critic 네트워크는 학습 초기 단계에서 상당한 변동성을 보였으며, 손실 값의 안정화가 늦게 이루어졌다. 이는 방어적 행동이 상대적으로 복잡한 환경 요인을 고려해야 하고, 전략적으로 아군을 보호하거나 거리를 유지하는 행동이 적절히 강화되지 않았음을 의미한다. 방어적 전략의 학습을 개선하기 위해서는 장기적인 보상을 반영하는 보상 구조가 필요하다. 예를 들어, 적의 공격을 방어하는 행동이나 아군을 보호하는 행동에 대해 지속적인 보상을 제공하고, 협력적인 방어 전략이 장기적으로 우수한 결과를 가져오도록 유도하는 방식이 필요하다. 이를 위해서는 할인율(Gamma)을 높여 미래 보상이 중요하게 다뤄지도록 하고, 보상의 누적 효과가 강조될 수 있도록 해야 한다. 그리고, 방어적 전략에서 발생하는 보상 변동성

을 해결하기 위해서는 경험 리플레이 및 우선 순위 샘플링(Prioritized Experience Replay, PER) 기법을 활용하여 중요 경험을 우선 학습하도록 할 수 있다. 또한, 타겟 네트워크와 같은 기법을 사용하여 Critic 네트워크의 불안정성을 줄이고, 가중치 업데이트의 안정성을 높이는 방법도 중요하다.

2) 보상 설계의 중요성

보상 설계는 에이전트의 행동을 결정짓는 핵심적인 요소로, 블루 팀과 레드 팀의 성능 차이를 이해하는 데 중요한 역할을 한다. 본 연구에서 블루 팀은 공격적인 협력과 적의 제압을 보상받았지만, 레드 팀은 방어 구역 형성, 거리 유지, 아군 보호와 같은 복합적 목표를 충분히 반영하지 못하였다. 특히, 방어 전략의 성공 여부를 구체적으로 평가하고 강화할 수 있는 보상 설계가 부족했으며, 이는 레드 팀의 학습 성과가 제한적이었던 주요 원인으로 작용하였다.

3) Critic 네트워크의 설계와 안정성

Critic 네트워크는 중앙 집중형 학습의 핵심으로, 각 에이전트의 행동과 팀의 전략적 협력을 통합적으로 평가한다. 본 연구에서는 MADDPG의 Critic 네트워크를 활용하여 학습했으나, 레드 팀의 Critic 네트워크는 학습이 느리게 진행되었으며, 변동성이 큰 학습 패턴을 보였다. 이는 방어적 성향의 복잡한 행동을 평가하는 Critic의 설계가 충분히 최적화되지 않았음을 나타낸다.

4) 팀 간 거리와 협력성

팀 내 에이전트 간 평균 거리는 블루 팀과 레드 팀 모두 일정 수준에서 유지되었으나, 공격적 협력(블루 팀)과 방어적 협력(레드 팀)이 명확히 구분되지 않았다. 이는 에이전트 간 상호작용과 팀워크가 보상 체계에 의해 충분히 강화되지 않았거나, Critic 네트워크가 협력적 행동을 학습하기에 적합하지 않았을 가능성을 시사한다.

5) 군사 시뮬레이션에서 적대적 강화학습의 한계

본 연구와 같이 적대적 강화학습을 적용한 군사 시뮬레이션에서는 다음과 같은 한계점이 존재한다:

- **장기적 학습의 불안정성:** 적대적 강화학습에서는 학습 중 양 팀의 정책이 상호 영향을 주며 비정상성 학습 환경을 형성한다. 이는 양 팀 모두 학습 안정성에 부정적인 영향을 미칠 수 있다.
- **의도적 과 적응 문제:** 한 팀의 정책이 특정 상대의 행동에 과도하게 적응 할 경우, 새로운 상대에 대해서는 비효율적인 행동을 보일 수 있다.

6-2 향후 연구 방향

1) 보상 설계의 다각화

방어적 성향을 학습하는 레드 팀의 성능을 개선하기 위해, 방어 구역 형성, 협력적 이동, 취약 에이전트 보호와 같은 목표를 명시적으로 반영하는 보상 구조를 도입해야 한다. 또한, 블루 팀의 경우 공격적 협력성을 더 강화하기 위해 팀원 간 거리 최소화나 집중 공격을 보상으로 설정하는 것이 효과적일 수 있다. 이와 함께, 특정 시나리오에 따라 가변적인 보상 체계를 설계하여 다양한 전략적 행동을 유도할 필요가 있다.

2) Critic 네트워크 구조의 개선

Critic 네트워크의 구조를 개선하기 위해, 먼저 입력 피처 공간을 확장하여 방어적 행동의 복잡성을 정확히 반영하도록 설계한다. 이를 위해 에이전트 간 거리, 적군 및 아군의 위치 밀도, 아군 보호 상태와 같은 정보를 Critic 네트워크의 입력으로 추가한다. 이러한 확장은 방어적 전략의 세부적인 행동 평가를 가능하게 한다. 또한, Critic 네트워크의 깊이와 구조를 최적화하여 복잡한 방어적 행동 패턴을 학습할 수 있도록 한다. 은닉층의 개수를 3~4개로 늘리고 각 층의 노드 수를 확장하며, 활성화 함수로 ReLU를 사용해 네트워크의 비선형성을 강화한다. Dropout과 Batch Normalization 기법을 적용하여 과적합을 방지하고 학습의 안정성을 확보한다. 학습 초기 단계의 불안정을 줄이기 위해 가중치를 Xavier 초기화나 He 초기화를 통해 초기화하며, 동적 학습률 조정을 도입해 학습 초기에는 낮은 학습률로 안정성을 높이고 이후에는 학습률을 점진적으로 증가시켜 학습 속도를 가속화한다. 또한, 타겟 Critic 네트워크의 업데이트를 Polyak 평균화 방식을 사용해 점진적으로 이루어지도록 하여 네트워크 간의 불일치를 완화한다. 마지막으로, 학습 과정의 효율성을 높이기 위해 우선 순위 샘플링를 적용한다. 이를 통해 보상 변화가 큰 경험을 우선적으로 학습하며, 방어적 행동 학습의 효율성을 증대시킨다. 동시에 보상 설계를 다중 목적 형태로 조정하여 거리 유지, 아군 보호, 적군 회피와 같은 방어적 행동의 세부 목표를 명확히 반영한다. 이러한 개선 방안은 Critic 네트워크의 안정성과 효율성을 높여 방어적 전략의 학습 성과를 극대화하고, 초기 학습 과정에서의 불안정성을 효과적으로 완화할 수 있다.

3) 다양한 강화학습 알고리즘의 적용

본 연구에서 사용된 MADDPG 외에도 PPO, SAC (Soft Actor-Critic)과 같은 알고리즘을 실험적으로 적용하여 성능을 비교할 필요가 있다. 특히 PPO는 정책 업데이트의 안정성을 제공하며, SAC는 탐색-활용 균형을 효과적으로 다룰 수 있어 방어적 전략 학습에 적합할 가능성이 높다.

4) 환경의 복잡도 증가

현재 연구에서는 단순한 이동, 공격 및 방어 행동으로 구성된 환경을 가정하였다. 향후 연구에서는 장애물 추가, 제한된 자원(예: 체력 회복 지역), 적응형 목표(예: 특정 구역 점령)와 같은 요소를 추가하여 환경의 복잡도를 증가시킬 필요가 있다. 이러한 확장은 에이전트의 전략적 행동을 유도하고, 학습

결과의 현실성을 높이는 데 기여 할 것이다.

• 실시간 협력 강화

팀원 간의 실시간 통신과 협력을 강화하기 위해 GNN(Graph Neural Network)을 활용할 수 있다. 이를 통해 에이전트 간 정보 공유를 효율적으로 수행하고, 협력적 의사 결정을 보다 정교하게 구현할 수 있다.

• 메타버스 내 시뮬레이션 적용

본 연구의 결과를 메타버스 상에서 군사 시뮬레이션, NPC(Non-Playable Character) 군사 협력 등 실제 응용 분야로 확장할 가능성을 검토해야 한다.

Ⅶ. 결 론

본 연구는 MARL에서 CTDE 접근을 활용하여 두 팀의 에이전트들이 협력과 경쟁을 통해 상반된 전략을 학습하고 전투 시나리오에서 적용하는 방법을 탐구하였다. MADDPG와 CTDE를 활용하여 에이전트의 행동 정책을 학습시켰다. 그럼에도 불구하고, 적대적 강화학습은 특정 상대에 과 적응하는 문제가 발생할 수 있으며, 학습 환경이 비 정상성이기 때문에 안정적인 학습이 어렵다는 한계를 보였다. 본 연구의 주요 결과와 시사점을 다음과 같이 요약할 수 있다.

7-1 주요 연구 결과

1) 팀 전략 차별화 성과

블루 팀(공격적 전략)은 높은 학습과 낮은 할인율을 통해 빠른 학습과 높은 공격 성공률을 성공하였고, 레드 팀(방어적 전략)은 낮은 학습률과 높은 할인율을 통해 장기적 방어 성공률을 목표로 학습하였으나, 초기 학습 속도가 느리고 변동성 높은 결과를 보였다.

2) Critic 네트워크 중심 협력 학습

중앙 집중 Critic 네트워크를 활용하여 에이전트 간 협력적 행동을 효과적으로 학습하였다. Critic 네트워크의 안정성이 팀 간 성과 차이에 중요한 영향을 미쳤으며, 레드 팀의 Critic 네트워크는 상대적으로 학습 안정성이 부족하였다.

3) 보상 설계의 영향

공격적 전략에서는 단순하고 직관적인 보상 체계가 효과적이었으나, 방어적 전략에서는 복잡한 목표를 충분히 반영하지 못한 보상 체계로 인해 학습 효율성이 제한되었다. 보상 설계가 팀 간 균형 유지 및 학습 성능 향상에 결정적인 역할을 하였다.

4) 성공률 비교

블루 팀은 공격적 전략으로 평균 성공률이 높았으며, 레드 팀은 장기적 방어 성공률에서 일부 개선된 성과를 보였으나, 초기 학습 단계에서 어려움을 겪었다.

7-2 시사점

1) 전략적 학습 파라미터의 중요성

- 학습률과 할인율 설정은 각 팀의 성향(공격적 또는 방어적)에 맞는 학습 전략을 유도하는 데 핵심적인 역할을 한다.
- 강화학습 환경에서 전략적 목표에 따라 파라미터를 조정하는 방식이 효과적임을 입증하였다.

2) Critic 네트워크 설계의 필요성

- 방어적 전략의 학습 안정성을 높이기 위해 Critic 네트워크의 설계를 개선할 필요가 있다.
- 팀 간 협력을 강화하는 데 있어 Critic 네트워크의 안정적 학습이 필수적이다.

3) 보상 체계의 정교화 필요

- 방어적 전략에서 다양한 목표(지역 보호, 팀 생존율 등)를 충분히 반영한 보상 체계가 필요하다.
- 협력적 행동과 팀워크를 강화하기 위한 보상 메커니즘이 중요한 역할을 한다.

4) 다양한 환경 및 시나리오 적용

- 본 연구는 제한된 환경(특정 크기의 맵과 에이전트 수)에서 실험되었으므로, 환경의 복잡도를 증가시키고 다양한 시나리오를 추가해 연구 결과의 일반화를 높이는 것이 필요하다.

5) 실제 응용 가능성

- 본 연구의 결과는 군사 시뮬레이션, 메타버스, 자율 로봇 등 다양한 분야에서의 협력 및 경쟁적 전략 학습에 적용 가능성을 제시한다.
- 적대적 환경에서의 강화학습 성능을 높이기 위한 새로운 알고리즘 설계로 확장 가능하다.

결론적으로, 본 연구는 MARL을 기반으로 한 적대적 환경에서의 팀 전략 학습에 대한 중요한 통찰을 제공한다. 블루 팀과 레드 팀의 전략적 차이를 유도하기 위해 학습률과 할인율을 차별화한 접근법은 각 팀의 목표에 적합한 학습 결과를 도출하였으며, 적대적 환경에서 협력적 및 경쟁적 행동 학습의 가능성을 입증하였다.

감사의 글

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 메타버스 융합대학원의 연구의 (IITP-2023-RS-2022-00156318)와 문화체육관광부 및 한국콘텐츠 진흥원의 2023년도 문화기술 연구개발사업(RS-2023-00219237)으로 수행된 연구로서, 관계부처에 감사드립니다.

참고문헌

[1] D. Huh and P. Mohapatra, “Multi-Agent Reinforcement Learning: A Comprehensive Survey,” arXiv:2312.10256v1, December 2023. https://doi.org/10.48550/arXiv.2312.10256

[2] L. Wan, Z. Liu, X. Chen, H. Wang, and X. Lan, “Greedy-Based Value Representation for Optimal Coordination in Multi-Agent Reinforcement Learning,” arXiv:2112.04454v1, December 2021. https://doi.org/10.48550/arXiv.2112.04454

[3] J. K. Gupta, M. Egorov, and M. Kochenderfer, “Cooperative Multi-Agent Control Using Deep Reinforcement Learning,” in *Proceedings of AAMAS 2017 Workshops*, São Paulo, Brazil, pp. 66-83, May 2017. https://doi.org/10.1007/978-3-319-71682-4_5

[4] G. Zhang, Y. Li, X. Xu, and H. Dai, “Efficient Training Techniques for Multi-Agent Reinforcement Learning in Combat Tasks,” *IEEE Access*, Vol. 7, pp. 109301-109310, August 2019. https://doi.org/10.1109/ACCESS.2019.2933454

[5] V. Ustun, R. Kumar, A. Reilly, S. Sajjadi, and A. Miller, “Adaptive Synthetic Characters for Military Training,” arXiv:2101.02185, January 2021. https://doi.org/10.48550/arXiv.2101.02185

[6] L. Liu, N. Gurney, K. McCullough, and V. Ustun, “Graph Neural Network Based Behavior Prediction to Support Multi-Agent Reinforcement Learning in Military Training Simulations,” in *Proceedings of 2021 Winter Simulation Conference (WSC)*, Phoenix: AZ, pp. 1-12, December 2021. https://doi.org/10.1109/WSC52266.2021.9715433

[7] C. Amato, “An Introduction to Centralized Training for Decentralized Execution in Cooperative Multi-Agent Reinforcement Learning,” arXiv:2409.03052, September 2024. https://doi.org/10.48550/arXiv.2409.03052

[8] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, “Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS ’17)*, Long Beach: CA, pp. 6382-6393, December 2017.

[9] J. Boron and C. Darken, “Developing Combat Behavior through Reinforcement Learning in Wargames and Simulations,” in *Proceedings of 2020 IEEE Conference on Games (CoG)*, Osaka, Japan, pp. 728-731, August 2020. https://doi.org/10.1109/CoG47356.2020.9231609

[10] H. van Hasselt, A. Guez, and D. Silver, “Deep Reinforcement Learning with Double Q-Learning,” in *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16)*, Phoenix: AZ, pp. 2094-2100, February 2016. https://doi.org/10.1609/aaai.v30i1.10295

[11] N. de la Fuente and D. A. V. Guerra, “A Comparative Study of Deep Reinforcement Learning Models: DQN Vs PPO Vs A2C,” arXiv:2407.1415, July 2024. https://doi.org/10.48550/arXiv.2407.14151

[12] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, “Robust Adversarial Reinforcement Learning,” in *Proceedings of the 34th International Conference on Machine Learning (ICML ’17)*, Sydney, Australia, pp. 2817-2826, August 2017. https://doi.org/10.48550/arXiv.1703.02702

[13] M. Zhao, G. Wang, Q. Fu, X. Guo, Y. Chen, T. Li, and X. Y. Liu, “MW-MADDPG: A Meta-Learning Based Decision-Making Method for Collaborative UAV Swarm,” *Frontiers in Neurorobotics*, Vol. 17, 1243174, September 2023. https://doi.org/10.3389/fnbot.2023.1243174

[14] L. Yuan, Z. Zhang, L. Li, C. Guan, and Y. Yu, “A Survey of Progress on Cooperative Multi-Agent Reinforcement Learning in Open Environment,” arXiv:2312.01058, December 2023. https://doi.org/10.48550/arXiv.2312.01058

[15] W. Chen, S. Koenig, and B. Dilkina, “MARL-LNS: Cooperative Multi-Agent Reinforcement Learning via Large Neighborhoods Search,” arXiv:2404.03101, April 2024.

[16] Y. Zhou, S. Liu, Y. Qing, K. Chen, Y. Huang, J. Song, and M. Song, “Is Centralized Training with Decentralized Execution Framework Centralized Enough for MARL?,” arXiv:2305.17352, May 2023. https://doi.org/10.48550/arXiv.2305.17352

[17] B. Grooten, M. Poot, J. Wemmenhove, and J. Portegies, “Is Vanilla Policy Gradient Overlooked? Analyzing Deep Reinforcement Learning for Hanabi,” arXiv:2203.11656, March 2022. https://doi.org/10.48550/arXiv.2203.11656

[18] L. Yu, J. Song, and S. Ermon, “Multi-Agent Adversarial Inverse Reinforcement Learning,” in *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, Long Beach: CA, pp. 7194-7201, June 2019. https://doi.org/10.48550/arXiv.1907.13220

[19] C. Zhou, J. Li, Y. Shi, and Z. Lin, “Research on Multi-Robot Formation Control Based on MATD3 Algorithm,” *Applied Sciences*, Vol. 13, No. 3, 1874, February 2023. https://doi.org/10.3390/app13031874

[20] M. Kouzeghar, Y. Song, M. Meghjani, and R. Bouffanais, “Multi-Target Pursuit by a Decentralized Heterogeneous UAV Swarm Using Deep Multi-Agent Reinforcement Learning,” in *Proceedings of 2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, pp. 3289-3295, May-June 2023. https://doi.org/10.1109/ICRA48891.2023.10160919

[21] W. Chen and J. Nie, “A MADDPG-Based Multi-Agent Antagonistic Algorithm for Sea Battlefield Confrontation,” *Multimedia Systems*, Vol. 29, No. 5, pp. 2991-3000, October 2023. https://doi.org/10.1007/s00530-022-00922-w

[22] P. Leroy, J. Pisane, and D. Ernst, “Value-Based CTDE Methods in Symmetric Two-Team Markov Game: From Cooperation to Team Competition,” arXiv:2211.11886v1, November 2022. https://doi.org/10.48550/arXiv.2211.11886

현기정(Ki-Jeong Hyun)

2005년 : 서강대학교 정보통신대학원 (공학석사)

2004년～현 재: 엔씨소프트 팀장
2024년～현 재: 서강대학교 메타버스전문대학원 박사과정
※관심분야 : 메타버스 게임, 지능형 블록체인, 인공지능

윤병현(Byeong-Hyun Yoon)

2017년 : 한국공업대학교 게임공학과 (공학사)

2016년～2016년 : ㈜드림로스팅
2017년～2019년 : 윤뱅게임즈
2019년～2020년 : ㈜플레이솔루션
2020년～2023년 : ㈜알타바그룹
2023년～2024년 : ㈜맥스트
2023년～현 재 : 서강대학교 메타버스전문대학원 석사과정
※관심분야 : 메타버스 게임, 생성형AI, 디지털저작권

송지원(Ji-Won Song)

2019년 : 성균관대학교 문헌정보학과, 통계학과 (문헌정보학사, 경제학사)

2020년～2023년: 인천국제공항공사
2023년～2024년: 스파르타코딩클럽
2024년～현 재: 유니티 테크놀로지스 코리아
2024년～현 재: 서강대학교 메타버스전문대학원 석사과정
※관심분야 : 게임, 그래픽스

조래현(Rae-Hyun Jo)

2023년 : 서강대학교 게임교육원 (미디어공학사)

2024년～현 재: 서강대학교 메타버스전문대학원 석사과정
※관심분야 : 게임, 디지털 저작권(DRM), 유비쿼터스 컴퓨팅