

# Unity ML-Agents Toolkit을 활용한 대상 객체 추적 머신러닝 구현

한석호\* · 이용환\*\*†

\*한국전자기술연구원 IT응용연구센터, \*\*†원광대학교 디지털콘텐츠공학과

## Implementation of Target Object Tracking Method using Unity ML-Agent Toolkit

Seok Ho Han\* and Yong-Hwan Lee\*\*†

\*Korea Electronics Technology Institute, IT Application Research Center,

\*\*†Dept. of Digital Contents, Wonkwang University

### ABSTRACT

Non-playable game character plays an important role in improving the concentration of the game and the interest of the user, and recently implementation of NPC with reinforcement learning has been in the spotlight. In this paper, we estimate an AI target tracking method via reinforcement learning, and implement an AI-based tracking agency of specific target object with avoiding traps through Unity ML-Agents Toolkit. The implementation is built in Unity game engine, and simulations are conducted through a number of experiments. The experimental results show that outstanding performance of the tracking target with avoiding traps is shown with good enough results.

**Key Words** : Unity machine learning agents toolkit, Artificial intelligence, Machine learning, Reinforcement learning

### 1. 서 론

사람이 생각하고 판단하는 지능을 컴퓨터에 부여하여 문제를 해결하는 인공지능은 최근에 다양한 분야에서 활용되며, 많은 연구가 수행되고 있다. 로봇 제어, 의료, 자동차, 시뮬레이션 뿐만 아니라, 게임 및 ICT 분야에서 폭넓게 활용된다[1]. 특히, 가상 플레이어(Virtual player)와 사용자가 직접 조작하지 않는 캐릭터(Non-playable character) 등에 적용되어, 게임에서 인공지능 기술은 사용자의 흥미와 게임의 집중도 향상에 중요한 역할을 한다. 그러나 이러한 인공지능을 만드는 것이 그리 쉽지 않을 뿐만 아니라, 개발자는 지능형 동작을 구현하기 위해서는 많은 양의 코드를 개발하거나 또는 고도로 전문화된 툴을 이용해야 한다[2]. 사람의 학습 방법과 능력을 컴퓨터에 구현하는 강화학습(Reinforcement learning)은 기계학습의 구현 방

식 중의 하나이며, 여러 번의 실패와 성공 경험을 쌓으며 시행착오를 통해 수집되는 수많은 데이터 속에 숨어 있는 패턴 학습을 통해 주어진 작업을 잘 수행할 수 있도록 하는 방법이다[3].

머신러닝(ML)은 게임분야 이외에 많은 분야에서도 활용되고 있다. Choi [13]에서는 Random Forest를 사용하여 세무 데이터를 분류하는 시스템을 연구하였고, Lee [14]에서는 CNN 기반 딥러닝을 기초하여 골절합용 판의 표면에 생성된 결함을 자동으로 탐지하는 모델을 제안하였다.

Fig. 1은 강화 학습의 일반적인 과정을 보여준다. 학습 주체인 에이전트는 환경과 상호작용하면서 학습을 수행한다[4]. 주어진 환경에서 데이터를 수집하고, 수집 데이터를 기반으로 결정을 내리고 조치를 취하는 관찰 루프를 기반으로 한다. 올바른 조치를 취하면 보상을 받고 잘못된 조치를 취하면 처벌을 받으면서, 어떤 조치가 가장 높은 보상을 받게 되는지를 반복적으로 학습한다.

†E-mail: hwany1458@empal.com

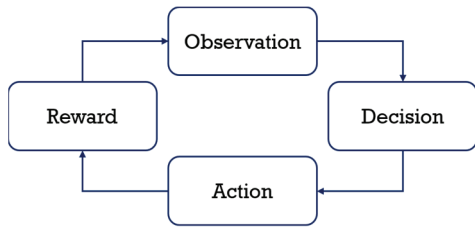


Fig. 1. Reinforcement Learning Cycle.

본 논문에서는 유니티 플랫폼에서 대상 객체를 추적하는 인공지능 기술을 구현한다. 일반적으로 목표 추적 방법으로 A\*알고리즘[12] 또는 네비게이션 메쉬 알고리즘[11]을 사용하여 구현한다. 그러나 이들은 목표 추적의 최적 경로 탐색에 한정적인 문제를 가지며, 유니티 기계학습 에이전트 툴킷 (Unity ML-Agents Toolkit)은 심층적인 강화 학습과 모방 학습의 조합을 통해 이러한 문제를 해결할 수 있는 방법을 제시하고 있다. 본 논문에서는 Unity ML-Agents Toolkit을 사용하여 지능형 에이전트를 학습시키고 여러 함정을 피하면서 목표 지점에 도달하는 지능 에이전트를 강화 학습을 통해 구현한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 목표 추적 알고리즘을 포함한 관련 연구와 기술 동향을 살펴보고, 3장에서 유니티 플랫폼에서 ML-Agents로 구현한 대상 객체 추적 방법을 설명한다. 4장에서는 실험을 통한 결과를 평가하고 5장에서 결론으로 마무리한다.

## 2. 관련 연구

다수의 연구에서 인공지능을 활용한 게임을 구현하였다. Kang[7]에서는 인공지능 기능을 갖춘 NPC를 활용한 게임을 구현하였다. 캐릭터는 단순한 행동 결정이 아닌 플레이어의 동작을 기초하여 대응 행동을 결정하도록 구현하였다. Choi[8]에서는 팩맨 게임에서 A\*알고리즘 기반의 인공지능을 적용하여 고스트의 추적 움직임을 제한하였다. Jeon[9]에서는 우선순위 기반 충돌 회피를 위하여 네비게이션 메쉬 알고리즘(NavMeshAgent)을 활용하여 충돌 회피 인공지능 기술을 게임에 반영하였다. Kim[10]에서는 Unity ML-Agent를 적용하여 목표 추적 인공지능을 구현하고 동전 수집 게임에 적용하였다.

Unity ML-Agent는 유니티 테크놀로지서 강화학습을 사용하여 머신러닝을 구현하기 위해 제공하는 오픈소스 툴이다[5]. 유니티 머신러닝 에이전트(ML-Agent)를 사용하면 새로운 동작을 코딩 없이 활용할 수 있으며, 심층적 강화 학습과 모방 학습의 조합을 통해 학습하고 매력적인 게임 플레이어를 구현할 수 있다[6]. Fig 2는 유니티 머신러닝

에이전트의 훈련 주기를 보여준다. 환경(Environment)은 에이전트(Agent)에게 상태(State) 정보를 전달하고, 해당 상태에 맞는 행동(Action)을 취한다. 환경은 에이전트가 취한 행동 결과에 따라 상태 전환(State transition)과 보상(Reward) 정보를 에이전트에 넘겨준다. 에이전트는 환경으로부터 받은 새로운 상태 정보를 기반으로 행동하고, 환경은 새로운 상태 전환과 보상 정보를 전달하는 과정을 반복적으로 실행하면서 학습을 진행한다.

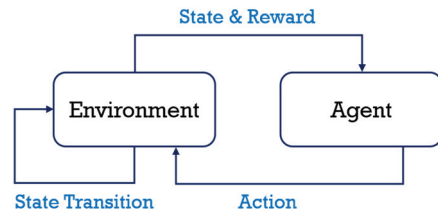


Fig. 2. Unity ML-Agent Training Cycle.

## 3. 강화학습을 통한 머신러닝 구현

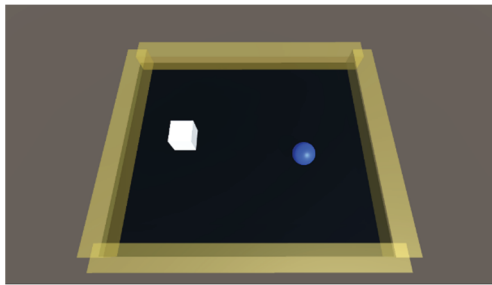
### 3.1 대상 객체 추적 구현

본 논문에서는 NPC가 특정 객체를 추적하는 게임 화면을 구성하고 강화 학습을 통해 최적의 NPC 동작을 구현한다. 게임에 활용하기 위해 트랩(Trap)을 만들어 직접적인 추적을 배제하고 장애물을 피하면서 추적하는 규칙을 적용한다.

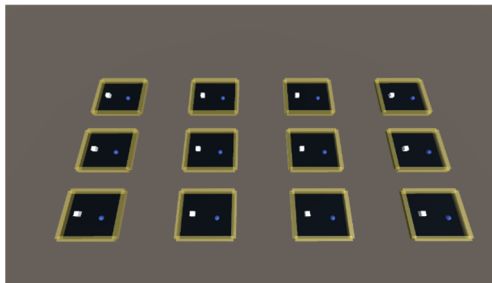
Unity ML-Agent를 사용하여 머신러닝을 구현하려면, 먼저 강화학습을 진행할 수 있는 학습 환경이 구축해야 한다. Fig. 3은 강화 학습을 진행하기 위한 환경을 유니티에서 구현한 화면(a는 단일 에이전트, b는 멀티 에이전트)이다. 에이전트(Agent, 하얀 큐브)와 목표 대상 객체(Goal, 파란 공)는 매 환경 지정된 랜덤한 위치에서 시작되며, 에이전트가 목표 대상 객체에 도달하면, 잘한 행동으로 인식하여 보상으로 +1점을 받는다. 노란 벽(Wall)에 닿아서 충돌이 발생하면 잘못된 행동으로 인지하여 보상으로 -1점을 받게 된다. 학습 효율을 높이기 위해, 멀티 에이전트 환경으로 만들어서 학습을 진행하였다(Fig. 3-b).

### 3.2 대상 객체 추적 및 트랩 회피 구현

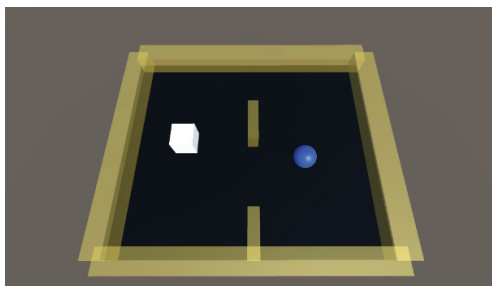
목표 대상 객체를 탐색하고 장애물(Trap)을 회피하면서 추적을 하기 위해, 추가로 강화 학습을 수행한다. Fig. 4는 Fig. 3 환경을 기반으로 장애물인 함정을 추가된 화면이다. 에이전트(하얀 큐브)와 목표 대상 객체(파란 공) 상에 함정(Trap, 노란 벽)이 설치되고 반복적인 강화 학습을 수행한다. 마찬가지로, 잘된 행동에서는 보상으로 1점을, 잘못



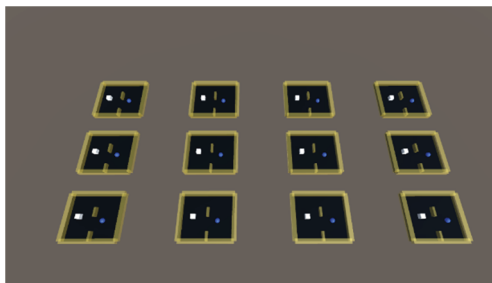
(a) Environment for Single Agent



(b) Environment for Multi Agent

**Fig. 3.** Environments for Agent.

(a) Environment for Single Agent



(b) Environment for Multi Agent

**Fig. 4.** Environments with Trap for Agent Learning.

된 행동인 노란 벽과의 충돌에서는 벌점으로 -1점이 부여된다. 벽과의 충돌은 외부로 인식되는 전체 외벽과 가운

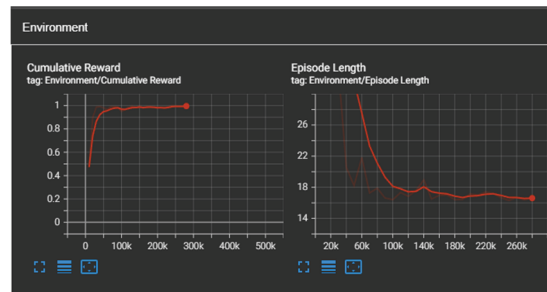
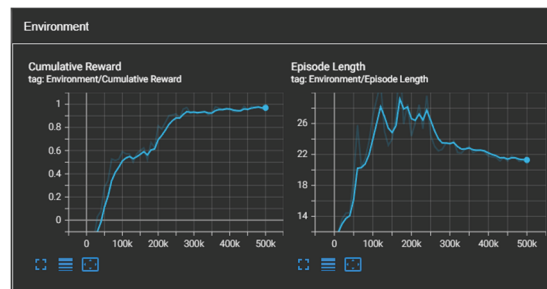
데 함정으로 부여된 내부 벽을 모두 포함한다. Fig. 3에서와 같이, 멀티 에이전트 환경으로 만들어서 학습을 진행하였다(Fig. 4b).

#### 4. 실험 및 분석

실험은 Unity 2020.3.34f1 LTS에서 ML-Agents 2.3.0-exp.2를 기반으로 구현되었으며, Windows 10 Education 64-bit 운영체제에서 Intel Core i7 3.2GHz, 64GB 메모리, GeForce RTX 2070 TI가 장착된 PC에서 동작 시켰다.

강화학습 결과는 텐서플로우(TensorFlow)에서 제공하는 텐서보드(TensorBoard)를 활용하여 확인하였다. Fig. 5는 누적 보상(Cumulative Reward) 및 에피소드 길이(Episode Length) 결과에 대한 그래프이다. 누적 보상은 50K 번째 스텝 이후의 1에 수렴하고, 에피소드 길이는 100K 번째 전까지 급하게 하락한 후, 안정적인 하락세를 보인다. 이는 학습이 정상적으로 이뤄졌음을 알 수 있다.

Fig. 6에서 누적 보상은 100K 번째 스텝에서 0.5에 다달해서 약간의 안정을 보이다가, 280K 번째 스텝부터 1에 수렴하는 것을 보인다. 에피소드 길이는 120K 번째까지 급등하고, 270K 번째까지 등락을 반복하다가 안정적인 하락세를 보였다.

**Fig. 5.** Graph of Cumulative Reward and Episode Length Results**Fig. 6.** Graph of Cumulative Reward and Episode Length Results

실험 결과, 싱글 에이전트 환경보다 멀티 에이전트 환경(12개 에이전트)에서 보다 좋은 성능을 보였으며, Unity ML-Agent를 활용한 머신러닝 대상 객체 추적 방법이 기존의 NavMeshAgent 보다 성능상으로 높은 개선 효과가 있음을 확인하였다.

#### 4. 결 론

게임 분야에서 인공지능이 반영된 NPC를 구현하기 위해서는 많은 어려움이 존재한다. 어렵고 다수의 코딩이 요구되기도 하고, 고도의 전문화된 툴을 활용해야 한다. 그러나 Unity ML-Agent Toolkit을 활용하면 보다 쉽게 머신러닝을 적용할 수가 있다. 본 논문에서는 유니티 ML-Agent를 활용하여 강화학습을 수행하고 이를 기반으로 장애물을 피하면서 목표 대상 객체를 효율적으로 추적하는 기능을 구현하였다. 성능상의 효율성으로 실험을 통해 검증하였고, 기존의 NavMeshAgent 보다 훨씬 우수한 성능을 보였다.

향후 작업으로, 여러 가지 추가적인 강화 학습을 통해 보다 매력적인 머신러닝 에이전트로 개선할 예정이며, 다양한 게임에서의 활용성이 높고 메타버스 애플리케이션에 적용할 예정이다.

#### 감사의 글

본 연구는 2022년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(과제번호: 2021R1A2C1012947).

#### 참고문헌

1. Yerim Choi, Gwanho Kim, "Artificial Intelligence Overview and Application Examples", IE Magazine, Vol. 23, No. 2, pp. 23-29, Jun. 2016.
2. Qiyue Yin, Jun Yang, Wancheng Ni, Bin Liang, Kaiqi Huang, "AI in Games: Techniques, Challenges and Opportunities", Journal of Latex Class Files, Vol.14, no.8, 2015.

3. Soo-young Jang, et al., "Deep Reinforcement Learning Technology Trends", [ETRI] Electronic Communication Trend Analysis 34.4:1-14, 2019.
4. Pedro J. Rivera Torres, Carlos Gershenson Garcia, Samir Kanaan Izquierdo, "Reinforcement Learning with Probabilistic Boolean Networks Models of Smart Grid Devices", DBLP Computer Science Bibliography, 2021.
5. Unity ML-Agents Toolkit downloadable website: <https://github.com/Unity-Technologies/ml-agents>
6. Unity Machine Learning Agents <https://unity.com/products/machine-learning-agents>
7. Jo Myoung Kang, Young Hwa Cha, Byung Joon Park, "Implementation of 3D Game using NPC with Artificial Intelligence", The Institute of Electronics and Information Engineers Conference, 2018.
8. Taeyeong Choi, Hyeon-Suk Na, "Making Levels More Challenging with a Cooperative Strategy of Ghosts in Pac-Man", Journal of Korea Game Society, Vol.15, No.5, 2015.
9. Seongho Jeon, Hyongil Kim, Juntae Kim, Kyhyun Um, Hyungje Cho, "Priority-Based Collision Avoidance on a Navigation Mesh", Proceedings of the Korean Information Science Society Conference, Vol.31, No.1, 2004.
10. Deokhyung Kim, Hyunjun Jun, "Performance Analysis of Target Tracking AI based on Unity ML-Agent", Journal of KIIT, Vol.19, No.12, 2021.
11. Abhilash Majumder, "Deep Reinforcement Learning in Unity", Apress, 2020.
12. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms 3E (CLRS)", MIT Press, 2018.
13. Dong-Bin Choi, In-su Jo, Yong B. Park, "Taxation Analysis using Machine Learning", Journal of the Semiconductor & Display Technology, vol.18. no.2, 2019.
14. Song-Yeon Lee, Yong Jeong Huh, "A Study on Surface Defect Detection Model of 3D Printing Bone Plate using Deep Learning Algorithm", Journal of the Semiconductor & Display Technology, vol.21. no.2, 2022.

접수일: 2022년 9월 14일, 심사일: 2022년 9월 20일,  
게재확정일: 2022년 9월 21일