JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 29 No. 2, pp. 61-66, February 2024
https://doi.org/10.9708/jksci.2024.29.02.061

Design of track path-finding simulation using Unity ML Agents

In-Chul Han*, Jin-Woong Kim*, Soo Kyun Kim*

*Student, Department of Computer Engineering, Jeju National University, Jeju, Korea
*Student, Department of Computer Engineering, Jeju National University, Jeju, Korea
*Professor, Department of Computer Engineering, Jeju National University, Jeju, Korea

[Abstract]

This paper aims to design a simulation for path-finding of objects in a simulation or game environment using reinforcement learning techniques. The main feature of this study is that the objects in the simulation are trained to avoid obstacles at random locations generated on a given track and to automatically explore path to get items. To implement the simulation, ML Agents provided by Unity Game Engine were used, and a learning policy based on PPO (Proximal Policy Optimization) was established to form a reinforcement learning environment. Through the reinforcement learning-based simulation designed in this study, we were able to confirm that the object moves on the track by avoiding obstacles and exploring path to acquire items as it learns, by analyzing the simulation results and learning result graph.

▸ **Key words**: Reinforce Learning, Simulation, Proximal Policy Optimization, Path-Finding, Unity Engine

[요 약]

본 연구에서는 강화학습 기술을 이용하여, 시뮬레이션이나 게임 환경 내에서 개체의 경로 탐색을 위한 시뮬레이션을 개발하는 것을 목표로 한다. 본 연구에서는 주어진 트랙 위에 생성된 임의 위치의 장애물을 회피하고, 아이템을 획득할 수 있는 경로를 자동으로 탐색할 수 있도록 시뮬레이션 내 개체를 학습시킨 점이 주된 특징이다. 해당 시뮬레이션을 구현하기 위해 유니티 게임 엔진에서 제공하는 ML 에이전트 (Machine Learning Agents)를 사용하였고, PPO(Proximal Policy Optimization)에 기반을 둔 학습 정책을 수립하여 강화학습 환경을 구성한다. 본 논문에서 제안한 강화학습 기반의 시뮬레이션을 통해, 개체가 학습을 진행할수록 장애물을 회피하고, 아이템을 획득할 수 있는 경로를 탐색해 트랙 위를 움직이고 있다는 점을 시뮬레이션 결과와 학습 결과 그래프를 분석하여 확인할 수 있다.

▸ **주제어**: 강화학습, 시뮬레이션, 근위 정책 최적화, 경로 탐색, 유니티 엔진

• First Author: In-Chul Han, Corresponding Author: Soo Kyun Kim
*In-Chul Han (ironin0923@gmail.com), Department of Computer Engineering, Jeju National University
*Jin-Woong Kim (0802dragon@naver.com), Department of Computer Engineering, Jeju National University
*Soo Kyun Kim (kimsk@jejunu.ac.kr), Department of Computer Engineering, Jeju National University
• Received: 2024. 01. 16, Revised: 2024. 02. 13, Accepted: 2024. 02. 13.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

유니티를 이용한 강화 학습 [1]에서는 물리적 공간에서의 시뮬레이션은 제약이 다양하며, 가상공간에 별도의 환경을 설정하고, 시뮬레이션을 진행하는 것이 효율적이라 할 수 있다. 2019년 구글의 딥마인드는 블리자드 사의 스타크래프트 게임을 플레이하는 게임 인공지능 알파스타[2]를 공개했다. 해당 사례를 참고하여 게임 공간에 자동으로 플레이 방법을 탐색하고 학습을 진행하는 인공지능 개체를 추가하면 보다 나은 사용자 경험을 줄 수 있다는 점을 생각해볼 수 있다. ML-Agents[3, 4]는 현재 상용되고 있는 게임 개발 엔진인 유니티 엔진에서 제공하는 기계 학습 기반 패키지이다. 따라서 유니티 엔진 내 가상공간에서 ML-에이전트를 통해 학습한 개체를 추가하면 게임과 함께 다양한 시뮬레이션에 활용할 수 있으며, 본 논문에서는 유니티 ML-에이전트를 통해 개체를 학습시키고, 강화학습 기반의 경로 탐색 시스템을 개발한다.

본 논문은 다음과 같이 구성한다. 2장에서는 유니티 엔진과 ML-에이전트에 관해 소개하고, 관련 연구 사례를 설명한다. 또한 본 논문에서 이용한 PPO 알고리즘을 소개한다. 3장에서는 제안하는 시스템의 목적과 강화학습 시의 보상 정책, 실제로 구성한 시뮬레이션에 관해 설명한다. 4장에서는 실험 결과와 시뮬레이션 진행에 대해 다루고, 5장에서는 4장의 내용을 기반으로 결론을 서술하고, 제안 방법의 기대효과와 향후 연구에 관해 설명한다.

II. Preliminaries

1. Unity Engine

유니티 엔진은 게임 개발 작업을 목적으로 상용되는 게임 엔진이다. 타 게임 엔진보다 비교적 쉬운 진입 장벽을 가지고 있어 많은 사람이 입문하는 게임 엔진이라 할 수 있다. 기본적으로 렌더링 시스템, 물리 엔진, 텍스쳐, 이펙트, UI와 같은 시스템을 가지고 있다. 또한, [5, 6, 7] 와 같이 여러 장르와 플랫폼을 가리지 않고 게임 콘텐츠를 만들어낼 수 있다. 유니티 엔진의 직관성과 낮은 진입 장벽으로, [8]의 경우는 유니티 엔진을 이용하여 메타버스(Metaverse)를 구현하려는 시도를 보였다. 그림 1과 같이, 유니티 엔진은 다양한 게임 개발에 있어 직관적인 조작과 핵심 기능을 제공한다는 것을 알 수 있다.

Fig 1. Examples of game development with Unity Engine[5]

[9, 10]의 사례가 유니티 게임 엔진으로 구성한 가상공간 내에서, 목적에 맞는 시뮬레이션을 진행할 수 있고, 그 가능성을 보여주고 있다. 드론 장애물 회피 알고리즘을 이용한 방법[10]의 경우, 유니티 게임 엔진의 ML 에이전트를 이용한 드론 시뮬레이션을 구성하였는데, 가상 카메라를 구현하여 드론이 장애물을 피하도록 구성했다. 이때 드론에 강화학습을 반복하고, 최종적으로 높은 정확도를 보였다고 밝히고 있다. 또한, 경로 탐색을 위해 사용되는 방법들은 제한적인 처리 능력을 보이거나, LiDAR와 같은 센서는 성능은 높으나 높은 비용을 요구한다는 점[10] 또한 알 수 있다. 본 연구는 앞선 유니티 게임 엔진을 이용한 사례들과 시뮬레이션에서의 활용성을 고려하여, 트랙 위의 개체가 경로를 탐색해 움직일 수 있도록 유니티 ML 에이전트 툴킷(Unity ML Agents Toolkit) 기반 강화학습 환경을 구성하고 그 결과를 분석한다.

2. ML-Agents Toolkit

ML-에이전트[3]는 새로운 동작을 일일이 코딩하거나 설정하지 않고, 지능형 에이전트의 심층 강화학습을 통해 지능적으로 반응하는 가상의 개체나 캐릭터를 만들 수 있도록 지원하는[4], 유니티 게임 엔진의 강화학습 기반 툴킷이다. 파이썬 API를 통해 강화학습과 모방학습 등의 기계학습 기법을 이용할 수 있고, 현재 2.0.1버전 기준 파이토치 라이브러리에 기반한 개발환경을 제공한다. 많은 게임 개발자와 연구자들이 게임 제작 및 강화학습 연구에 유니티 엔진과 ML-에이전트 툴킷을 이용하고 있다[11]. 강화학습을 통한 목표 추적 인공지능 기법[12]에서 밝히는 바와 같이, 기계 학습 기반의 기술을 구현하는 것에 있어 ML-에이전트의 성능이 유니티 엔진의 기본 컴포넌트이자 알고리즘인 NavMeshAgent보다 높다는 것을 보여주었다. 또한 최적의 성능을 위해 에이전트에 대해 적절한 파라미터값을 설계해야 함을 보여주었다. 본 연구에서도 3장에서 이야기할 주요 파라미터들을 사용하는 경우, 최적의 성능을 위해 파라미터 값을 별도 지정하여 학습을 진행한다.

3. Proximal Policy Optimization

강화학습은 최대 보상을 위해 최적의 행동을 하게끔 학습하는 것을 목적으로 한다. 이때 최적의 정책을 위해 누적 보상을 최대화하는 방식을 PG(Policy Gradient) 기법이라 한다[13, 16]. Proximal Policy Optimization[13] 알고리즘은 이러한 PG 기법의 하나라 할 수 있다. PPO 기법은 에이전트가 환경과 상호작용하며 학습하는 것을 전제로 하고, 에이전트의 정책(Policy)을 업데이트해 최적 행동을 선택하도록 한다. 이때 이전 정책과의 KL(Kullback-Leibler) 발산 값을 제한하고, 대체 목적 함수(Surrogate Objective Function)를 최적화하여 Agent로 하여금 이전 정책과 너무 다른 행동을 하지 않도록 유도해 안정성을 높인다. PPO 알고리즘의 의사코드는 표1과 같다.

Table 1. Pseudo-code of proximal policy optimization (PPO) algorithm[13]

Algorithm 1 PPO, Actor-Critic Style
for iteration=1, 2, ... **do**
for actor=1, 2, ..., N **do**
Run policy $\pi_{\theta_{old}}$ in environment for T timesteps
Compute advantage estimates $\hat{A}_1, ..., \hat{A}_T$
End for
Optimize surrogate L wrt θ, with K epochs and minibatch size $M \leq NT$
$\theta_{old} \leftarrow \theta$
End for

의사코드의 진행은 다음과 같다. 반복 횟수만큼의 반복 루프 동안 N만큼의 액터(actor)에 대한 정책 $\pi_{\theta_{old}}$를 환경(Environment) 내에서 T번의 시간 간격만큼 진행한다. 그동안 추정 보상인 $\hat{A}_1, ..., \hat{A}_T$를 계산하도록 한다. actor에 대한 연산을 마친 뒤에는 정책(Policy)에 대한 손실(Loss)을 계산하며 정책을 새롭게 갱신하는 순서로 진행된다. PPO는 타 강화학습 기법보다 안정적이고, 적은 샘플을 가지고도 높은 성능을 보여 로봇 제어 등 여러 응용분야[14]에서도 사용되고 있다. 본 연구에서도 ML-에이전트 학습 기법을 PPO로 설정해 시뮬레이션을 진행한다.

III. Proposed method

본 연구는 앞선 장에서 밝힌 바와 같이 에이전트가 트랙 위에 무작위로 생성되는 장애물을 피하고, 점수를 획득하여 도착지까지 이동하는 시뮬레이션의 제작을 목표한다.

1. Object to be Detected

본 연구에서 에이전트는 자동으로 전진하는 과정에서 좌, 우의 입력만 이용한다. 이때 레이(Ray) 센서를 통해 트랙, 평면, 방해물(Obstacle), 스코어(Score), 목표(Goal) 오브젝트를 감지하고, 감지된 오브젝트의 정보를 바탕으로 강화학습을 진행한다. 트랙(Track)은 에이전트가 움직이는 길이며, 평면의 경우 트랙 외부의 공간으로 에이전트가 지나가는 경우 보상을 감소하도록 설정한다. 에이전트는 방해물을 피해야 하므로 장애물 개체와 충돌 시 보상이 감소하도록 설정한다. 스코어의 경우, 에이전트가 획득해야 하는 요소이므로 충돌 시 보상이 증가하도록 설계한다. 마지막 목표(Goal)의 경우, 에이전트가 트랙을 한 바퀴만 돌 수 있게 하도록 설정한다.

2. Learning Policies and Learning Parameters

본 연구에서 학습이 진행되는 동안 적용되는 정책은 다음과 같다. 에이전트가 평면과 방해물에 충돌 시 보상을 1점 감소시킨다. 그리고 스코어와 충돌 시 보상을 1점 증가시킨다. 에이전트가 트랙을 따라 움직이는 것이 올바른 학습인 것으로 인지할 수 있도록 에이전트가 움직이는 동안 점수가 0.01점씩 증가하게끔 설계한다. 또한 에이전트가 과도하게 회전하지 않도록 회전할 때마다 보상이 0.001점씩 감소하도록 정의하였다. 모든 스코어를 획득하고 목표에 충돌하면 1점의 보상을 획득하고, 아무 스코어도 획득하지 못한 채로 목표에 충돌한다면 1점의 보상을 감소하도록 한다. 에이전트가 학습하는 알고리즘의 주요 파라미터[15]는 표 2의 내용과 같이 구성하여 사용하였다.

Table 2. Configuring ML-Agent Learning Parameters

Hyper Parameter	Value
trainer type	ppo
batch size	1,024
buffer size	50,000
learning rate	0.003
lambd	0.95
num epoch	3
max steps	10,000,000

먼저, 트레이너 타입(trainer type)의 값을 ppo로 설정하여 PPO 알고리즘을 사용하여 강화학습을 진행하도록 설정하였다. 배치 사이즈(batch size)는 경사하강법 진행 시 정책 수용 학습 경험의 수로, 연속으로 입력되는 좌, 우 방향을 충분히 받도록 1,024로 설정한다. 버퍼 사이즈

(buffer size)는 정책을 결정할 때 사용하는 경험의 수로, 안정적인 학습을 위해 배치 사이즈의 50배 수준인 50,000으로 설계하였다. 학습률(learning rate)는 경사하강법의 학습 강도로, 안정적인 학습을 진행할 수 있도록 0.003이라는 수치를 적용하였다. lambd는 기존 정책과 환경에 대한 에이전트의 의존도를 조절하며, 0.95의 값으로 설정하여 정책과 환경의 중간에서 학습하도록 설계한다. num epoch는 경사하강법 최적화에 이용되는 패스 수를 말하며, 안정적인 학습을 위해 3의 값으로 설정하였다. 마지막으로, 에이전트가 해당 환경을 충분히 학습할 수 있도록 1,000만 번의 경험을 하도록 설정 후 학습을 진행하였다.

3. Development Environment

본 연구에서, 유니티 게임 엔진을 사용해 그림 2와 같은 씬(Scene)을 구성한다. 학습 시마다 트랙 위에 지정된 포인트 지점에서 무작위로 방해물과 스코어 개체가 생성된다. 트랙의 경우 유니티의 스플라인 오브젝트를 활용하여 곡선으로 구현하였다. 에이전트는 좌측 중앙 흰색 출발선을 기준으로 스스로 경로를 탐색해 트랙을 이동한다. 스플라인 (Spline) 개체를 통해 트랙을 구성하기 위하여 2022년 버전의 유니티 엔진을 기반으로 진행되었다. 표 3, 4의 내용을 통해 본 연구의 개발환경을 알 수 있다.

Table 3. H/W Environment

classification	spec.
CPU	Intel(R) Core(TM) i7-8700K (3.70GHz)
GPU	NVIDIA Geforce RTX 3060Ti (VRAM 8GB)
RAM	DDR4-2400 64GB (SK Hynix 32GB) (SpecTek Incorporated 32GB)

Table 4. S/W Environment

classification	Version
OS	Windows 11 Pro
Unity Game Engine	Unity 2022.3.10f1
Python	Python 3.9
PyTorch	2.0.1
ML-Agents	2.0.1

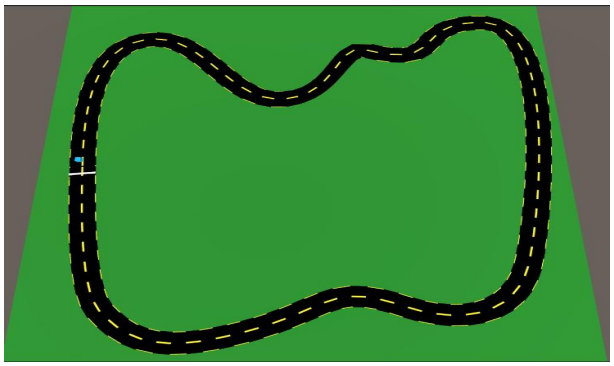

Fig. 2. **Our results,**
The track created using a Spline object and an agent trained using reinforcement learning (blue box)

IV. Result

다음은 본 연구의 구현 결과이다. 에이전트 학습 후, 텐서 보드 (Tensor Board)를 통해 결과를 시각화하였다.

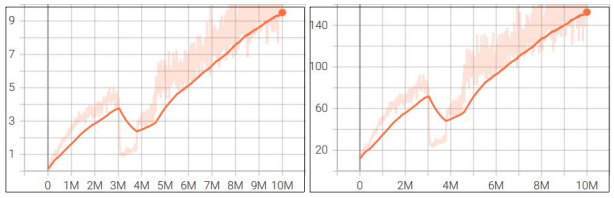

Fig. 3-(a) Fig. 3-(b)

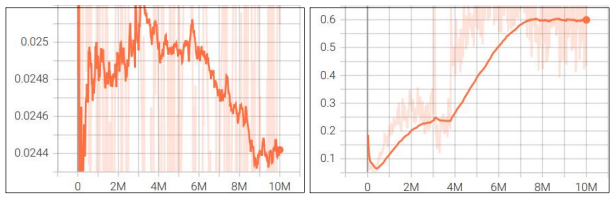

Fig. 3-(c) Fig. 3-(d)

Fig. 3. Four graphs showing the final results after the simulation proposed in this study

그림의 3-(a)는 누적 보상 (Cumulative Reward), 즉 Agent가 학습하는 보상이 누적되고 있는 모습을 통해, 매 학습마다 얻는 보상의 편차를 알 수 있다. 학습이 진행될수록 보상이 증가하는 모습을 보였다. 3-(b)는 각 학습이 진행될 때마다 걸리는 평균 길이를 의미하며, 이를 에피소드 길이(Episode Length)라 부른다. 에이전트가 장애물을 피하며 오래 살아남고 있어 매번 학습에 소요되는 시간의 길이가 증가하는 모습을 보였다. 3-(c)는 정책 손실 함수의 크기를 나타내는 정책 손실(Policy Loss)를 말하며, 정책이 얼마나 변경되는지에 따라 값이 변하게 된다. 학습을 진행하는 단계에서는 증가하였다가 보상이 안정화되면서

정책 변화량도 점차 감소하는 모습을 보였다. 3-(d)는 가치 함수의 손실을 나타내며 가치 손실(Value Loss)이다. 에이전트의 상태 예측에 따라 값이 변화하게 되며 본 연구에서 해당 그래프는 점차 증가하였다가 0.6에 점차 수렴하는 모습을 보였다. 학습 그래프를 통해 본 연구에서 진행한 시뮬레이션 결과를 분석하면, 에이전트의 누적 보상(Cumulative Reward)은 증가하면서 가치 함수의 손실(Value Loss)은 증가하다 수렴하는 모습을 보였다고 할 수 있다. 보상이 증가하면서 가치 손실은 증가하지 않는 결과를 통해, 본 연구에서 목표한 대로 PPO 알고리즘을 이용한 강화학습 기반 Agent의 학습이 긍정적으로 진행되었다는 것을 알 수 있었다. 그림 4는 실제로 진행된 시뮬레이션의 모습이다.

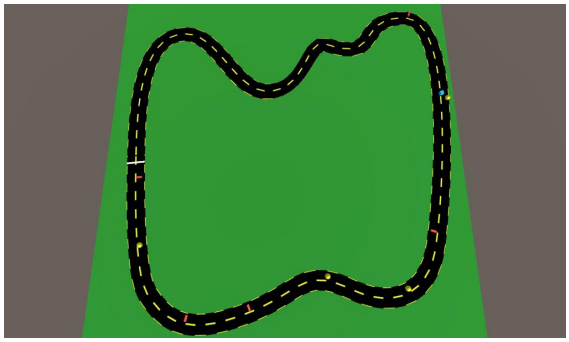

Fig. 4. The simulation implemented in this study in action

에이전트를 의미하는, 파란 박스 개체가 트랙을 이탈하지 않고 움직이는 모습을 확인할 수 있었다. 또한, 그와 동시에 임의로 생겨난 장애물은 피하고, 스코어 개체는 획득(충돌)하여 트랙을 완주하는 결과를 확인할 수 있었다. 단, 가장 이상적인 결과의 학습에서는 가치 손실(Value Loss)의 값은 낮아지는 그래프를 띄어야 하는데 본 연구에서의 학습은 오차가 특정한 값으로 수렴하는 점을 발견할 수 있었다. 이를 완화하기 위해 파라미터를 재조정하고, 보상 정책을 수정하여 완화해야 할 것으로 분석할 수 있었다.

V. Conclusions

제안 방법은 유니티 엔진의 ML-에이전트와 근위 정책 최적화(Proximal Policy Optimization) 알고리즘을 활용하여 트랙 위 에이전트가 트랙을 벗어나지 않고 장애물은 회피하며, 그와 동시에 아이템을 추적, 획득할 수 있도록 강화학습 기반의 시뮬레이션을 구축한 점이 주된 특징이다. 본 연구를 진행하기 위해 기존 유니티 게임 엔진을 통한 시뮬레이션 사례와 ML-에이전트 기반의 연구 사례를 조사하였다. 제안된 시뮬레이션에 대해 학습을 진행한 이후, 긍정적인 결과 그래프와 함께 에이전트의 장애물 회피와 아이템 획득, 경로 탐색에 있어 뚜렷한 성능 향상이 이루어진 점을 확인할 수 있었다.

향후 학습된 에이전트를 다른 트랙이나, 게임 공간에 배치하여 제안한 시뮬레이션의 활용성을 추가로 확인할 예정이며, 학습을 마친 개체를 타 게임이나 시뮬레이션에도 적용하여 활용성을 높일 수 있도록 할 계획이다.

ACKNOWLEDGEMENT

본 연구는 중소벤처기업부와 중소기업기술정보진흥원의 “지역특화산업육성+(R&D, S3365329)”사업의 지원을 받아 수행된 연구결과임.

REFERENCES

[1] H.-B. Choi, C.-M. Kim, J.-B. Kim, and Y.-H. Han, “Design and Implementation of Reinforcement Learning Environment Using Unity 3D-based ML-Agents Toolkit”, Proceedings of the Korea Information Processing Society Conference, pp. 548-551, May 2019.

[2] Vinyals, O., Babuschkin, I., Czarnecki, W.M. et al., “Grandmaster level in StarCraft II using multi-agent reinforcement learning”, Nature 575, pp. 350-354 October 2019.

[3] Ml-agents github page, Github, “https://github.com/Unity-Technologies/ml-agents”

[4] Unity Machine Learning Agents, Unity Technologies, “https://unity.com/products/machine-learning-agents”

[5] Jin-Woong Kim, Jee-Sic Hur, Hyeong-Geun Lee, Ho-Young Kwak, and Soo Kyun Kim, "Design of Action Game Using Three-Dimensional Map and Interactions between In-Game Objects", Journal of The Korea Society of Computer and Information , vol. 27, no. 12, pp. 85-92, 2022.

[6] Lee Young-Joon, Hwang bong-su, Son ha-yeon, Han Seung-yeon, Jo kyong-jin, and Kim Eun-han, "Development of Rhythm Game using Unity Engine", in Proceedings of KIIT Conference, 2022, pp. 530-533.

[7] Bae Jae Hwan, "Design and Development for Unity3D Game Engine using the Shooting Game", Journal of The Korean Society for Computer Game, vol. 29, no. 1, pp. 93-100, 2016.

[8] Gang In Lee, Seok Ho Han, and Yong-Hwan Lee, “Implementation of Metaverse Virtual World using Unity Game Engine”, Journal

of the Semiconductor & Display Technology, Vol. 22, No. 2, pp. 120-127 June 2023.

[9] Z. Wang, X. Liao, C. Wang, David Oswald, Guoyuan Wu et al., "Driver Behavior Modeling Using Game Engine and Real Vehicle: A Learning-Based Approach", in IEEE Transactions on Intelligent Vehicles, vol. 5, no. 4, pp. 738-749, December 2020. DOI: 10.1109/TIV.2020.2991948.

[10] S. Jo and T.-Y. Kim, “Drone Obstacle Avoidance Algorithm using Camera-based Reinforcement Learning”, Journal of the Korea Computer Graphics Society, vol. 27, no. 5. Korea Computer Graphics Society, pp. 63−71, November 2021.

[11] Č. Livada and D. Hodak, "Advanced Mechanisms of Perception in the Digital Hide and Seek Game Based on Deep Learning", 2022 International Conference on Smart Systems and Technologies (SST), Osijek, Croatia, 2022, pp. 135-140, DOI: 10.1109/SST55530.2022.9954814.

[12] D. Kim and H. Jung, “Performance Analysis of Target Tracking AI based on Unity ML-Agents”, The Journal of Korean Institute of Information Technology, vol. 19, no. 12. pp. 19−26, December 2021.

[13] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov, “Proximal Policy Optimization Algorithms”, OpenAI, arxiv.org/pdf/1707.06347, 2017.

[14] Sung Gwan Park and Dong Hwan Kim, "Autonomous Flying of Drone Based on PPO Reinforcement Learning Algorithm", Journal of Institute of Control, Robotics and Systems, vol. 26, no. 11, pp. 955-963, 2020. DOI: 10.5302/J.ICROS.2020.20.0125

[15] Parameters In ML-Agents Toolkit, Unity Technologies “https://unity-technologies.github.io/ml-agents/Training-Configuration-File/”

[16] Y.-H. Liang, S.-J. Kang, and S. H. Cho, “A Study about the Usefulness of Reinforcement Learning in Business Simulation Games using PPO Algorithm”, Journal of Korea Game Society, vol. 19, no. 6. Korea Academic Society of Games, pp. 61−70, December 2019.

Authors

In-Chul Han is currently a Student in the Department of Computer Engineering, Jeju National University. He interested in game developing, 3D rendering in computer graphics, and artificial intelligence in game contents.

Jin-Woong Kim received the B.S. degree in Computer Engineering, Jeju National University, Jeju, Republic of Korea, in 2023, where he is currently pursuing the master’s degree in computer science. He interested in Artificial Intelligence Computing for Computer graphics, and Affordance of Graphic media such as game contents.

Soo Kyun Kim received Ph.D. in Computer Science & Engineering Department of Korea University, Seoul, Korea, in 2006. He joined the Telecommunication R&D Center at Samsung Electronics Co., Ltd., in 2006 and 2008. He is now a professor at the Department of Computer Engineering at Jeju National University, Korea. Dr. Kim has published many research papers in international journals and conferences. His research interests include multimedia, pattern recognition, image processing, mobile graphics, geometric modeling, and interactive computer graphics. He is a member of ACM, IEEE, IEEE CS, KACE, KMMS, KKITS, and KIIT.