게임개발 환경 개선을 위한 강화학습 기반 데이터 생성 환경 연구

이형석[1], 양희석[1], 박근희[1], 김만제[2*]
[1]전남대학교 소프트웨어공학과 학부생
[2]전남대학교 인공지능융합학과 교수
yoghyogh12@naver.com, sce728@gmail.com, pgh2045@naver.com, jaykim0104@jnu.ac.kr

Research on Reinforcement Learning-Based Data Generation Environments for Games

Hyeong-Seok Lee[1], Hui-Seok Yang[1], Geun-Hi Park[1], Man-Je Kim[2]
[1]Dept. of Software Engineering, Chonnam National University
[2]Dept. of AI Convergence, Chonnam National University

요 약

최근 인공지능은 게임 분야에서 적극적으로 활용되고 있는데, 움직임, 패턴, 데이터 처리 등 대부분 과정에서 빠짐없이 사용되고 있다. 그러나 일반적으로 인공지능은 다루기 어렵다는 인식과 개발 비용이 많이 든다는 문제로 인해 중소 게임사에서는 이를 활용하기 어렵다. 이에 본 연구에서는 게임사에서 손쉽게 데이터를 생성하여 게임 내 밸런스 등을 고려할 수 있는 강화학습 기반 데이터 생성 환경을 제안하여 중소규모의 게임사의 게임개발 능률 향상에 도움을 주고자 한다.

1. 서론

최근 인공지능(Artificial Intelligence)의 눈부신 발전을 바탕으로 제조업, 유통 등을 비롯한 기반산업 뿐만 아니라, 여가 산업으로 분류되는 게임 산업까지 인공지능이 활용되고 있다. 이를 방증하듯이 국내 8대 게임사에서만 이미 수백 명의 인공지능 연구 인력을 보유하고 있으며, 이들 8사의 작년 한해 누적 AI 연구 개발비는 약 2조에 달한다.[1] 그러나 8대 게임사를 제외한 대부분의 게임사는 중소규모로 이루어져 있어 인공지능을 활용하기에 어려움을 겪고 있다. 본 연구에서는 투자가 어려운 중소게임사에서 손쉽게 게임 플레이 데이터를 생성하고 활용할 수 있는 강화학습 기반 플레이어 데이터 생성 환경을 제안하고자 한다.

2. 학습 환경

실험에 사용할 게임은 서비스되는 게임을 활용하는 것이 가장 좋으나, 게임 저작권 등의 문제로 직접 구축하여 활용하고자 한다. 뱀서라이크 게임, 일명 역탄막 게임으로 몰려오는 적을 처치하여 생존하는 게임이다. 해당 장르는 중소형 개발사의 개발이 많기 때문에 뱀서라이크 환경을 활용하였다. 또한, 연구에서 개발하는 모델의 학습환경은 2D 유니티 환경으로 설정하였다. 유니티 엔진은 일반적으로 간편한 환경 구성과 복잡성을 줄일 수 있다고 알려져 있으며, 2D에서 2.5D와 3D로의 이식이 용이하기에 2D 유니티를 선택하였다.[2] 게임 산업 데이터 분석 프로젝트인 게임데이터크런치의 조사에 따르면 게임 전 분야에서 개발 중인 개발자 중 49.48%가 Unity 엔진을 사용하는 것으로 나타났다.[3]

마지막으로 강화학습을 수행하기에 앞서, 에이전트(Agent)가 학습할 공간을 오픈월드로 구성하고, 다양한 크기의 장애물을 배치하였다. 에이전트는 무한히 생성되는 적(Enemy)을 피해 이동하거나 처치하면서, 확률적으로 보상을 획득할 수 있는 환경을 제공하였다.

3. 실험

적(Enemy)은 에이전트를 추적하고 무한히 생성되도록 하였다. 효율적인 추적을 위해 최단경로 탐색에 주로 활용되는 A^*알고리즘을 적용하고자 하였으나, 학습 공간에 존재하는 다양한 형태의 장애물로 인해 적들이 유저에게 몰리면서 병목 현상이 발생하였다. 해당 연구에서는 장애물을 감지하게 되면 법선 벡터를 계산하여 반대 방향으로 우회하는 방식을 적용하여 문제를 해결하였다. 그 결과, <표1>과 같은 장애물 회피 알고리즘을 적용한 후 적의 경로 선택이 더 효율적으로 이루어졌기에 실험 환경의 완성도가 향상되었다.[4]

<표 1> 장애물 회피 알고리즘

Algorithm 1 Modified A* with Obstacle Avoidance

```
if neighbor is "Obstacle" then
    normal_vector := calculate_normal_vector(current, neighbor)
    goal := goal + normal_vector
    continue
end if
```

이러한 환경을 바탕으로 유저가 적을 피해 생존에 집중

하는 회피(Passive) 모델과 적을 처치(Kill)하고 보상(Exp)을 획득하는 데 집중하는 공격(Aggressive) 모델을 설정하고, 보상과 벌점 조건을 달리하여 학습을 수행하였다. 각 에피소드의 길이는 5분으로 설정되었으며, PPO(Proximal Policy Optimization)알고리즘을 사용하여 학습을 진행하였다.

Passive Model (Avoidance):

$$R(s,a) = \begin{cases} +5f, & \text{if survives 5 minutes} \\ +4f, & \text{if survives 4 minutes} \\ +3f, & \text{if survives 3 minutes} \\ +2f, & \text{if survives 2 minutes} \\ +x, & \text{every 20 seconds of survival} \\ +0.7f, & \text{if agent moves away from spawned enemy} \end{cases}$$

Penalties:

$$P(s,a) = \begin{cases} -1f/x, & \text{if contacts a monster every 20 seconds} \\ -5f, & \text{if HP} < 50 \end{cases}$$

x increases by 0.5 every 20 seconds.

Aggressive Model (Attack):

Common Rewards:

$$R(s,a) = \begin{cases} +3f, & \text{if survives 5 minutes} \\ +3f, & \text{if survives 4 minutes} \end{cases}$$

EXP Count:

$$R_{EXP}(s,a) = \begin{cases} +x, & \text{for multiples of 2 EXP gained} \\ +2f, & \text{for multiples of 5 EXP gained} \\ +0.4f, & \text{if agent approaches spawned EXP} \end{cases}$$

Kill Count:

$$R_{KC}(s,a) = \begin{cases} +0.01f, & \text{for kills 0-100} \\ +0.02f, & \text{for kills 100-250} \\ +0.03f, & \text{for kills 250-700} \\ +0.04f, & \text{for kills 700-1000} \\ +0.05f, & \text{for kills >1000} \end{cases}$$

Penalties:

$$P(s,a) = \begin{cases} -0.2f/x, & \text{if contacts a monster every 20 seconds} \\ -3f, & \text{if HP} < 50 \end{cases}$$

x increases by 0.2 every 20 seconds.

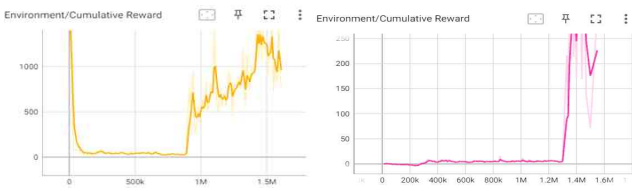
Common End Conditions:

- The episode ends if HP < 50 or if 5 minutes have passed.

(그림 1) 두 모델의 보상 및 벌점 함수

(그림 1)과 같이 보상 신호를 설정하고 정책을 개선하였다. 구축된 학습환경에서는 100만 스텝을 기준으로 보상 신호의 학습이 이루어졌고 에이전트의 성능이 점차 개선되었는데, 회피 모델과 공격 모델 모두 약 150만 스텝 학습을 거쳐 성능이 향상되는 것을 확인할 수 있었다. 이를 통해 에이전트가 최적의 행동 경로를 찾아가는 과정이 (그림 2)와 (그림 3)을 통해 시각적으로 나타났다.

(그림 2) 회피 모델의 보상 그래프



(그림 3) Exp 선호모델(좌)과 Kill 선호모델(우)의 보상 그래프

공격 모델은 게임사에서 다양한 게임 내에 필요한 모델을 만들 수 있도록 더 세부적인 검증을 위해 Exp 선호, Kill 선호로 나누어 학습을 진행하였다. 앞선 회피 모델과 마찬가지로 100만 스텝 전후로 성능이 우상향하는 모습을 (그림 3)과 같이 확인할 수 있었다. 이 결과는 제안한 방법이 게임 내 다양한 목표를 학습할 수 있음을 보여준다. 또한, 게임사에서 원하는 플레이 데이터를 수집하고자 할 때, 활용한다면 굳이 많은 실험자를 사용하지 않고도 다양한 플레이 데이터를 수집할 수 있음을 의미한다.

4. 결론

본 연구에서는 게임 환경에서 강화학습을 활용한 데이터 생성 환경을 제안하였다. 특이성이 강하고 수정이 어려운 환경 대신 범용적인 오픈월드 학습환경을 사용하여, 장애물과 적과의 상호작용을 통해 다양한 행동패턴을 학습하도록 설계하였으며, 유저가 취할 수 있는 행동(회피, 공격)에 따라 보상 및 벌점을 설정하여 학습 수행하였다. 이를 통해 게임사는 플레이 데이터를 원하는 목표에 맞추어 손쉽게 수집할 수 있으며, 게임개발 및 난이도 조절 등에 활용할 수 있을 것으로 사료된다.

다만, 본 연구에서는 높은 수준의 인간 플레이어 데이터가 없었기에 이를 실제적으로 비교하기 어려웠다. 그로 인해 실제 사용자들이 느끼는 게임의 흥미 요소를 자극할 수 있는지와 플레이어 수준을 가늠하기 어렵다는 문제가 존재한다. 이러한 문제를 보완하기 위해

추후 연구로 실제 유저 데이터를 수집하여 생성된 데이터와의 직접적인 비교 및 다양한 실험을 통한 검증과 보완을 수행하고자 한다.

Acknowledgment

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 소프트웨어중심대학사업(2021-0-01409)과 대학ICT연구센터사업(IITP-2024-RS-2024-00437718) 그리고, 인공지능융합혁신인재양성사업(IITP-2023-RS-2023-00256629)의 연구결과로 수행되었습니다.

참고문헌

[1] 김환영, "[게임업계 상반기 실적]④ 게임업계, AI 연구개발에 진심...엔씨 '바르코' 연구실적 올려", 조세일보, 2023.08.24., https://m.joseilbo.com/news/view.htm?newsid=495714

[2] Adams, Ernest "Fundamentals of Game Design" 1st Ed. Prentice Hall, 2006

[3] Lars Doucet, Anthony Pecorella "Game engines on Steam: The definitive breakdown", GameDeveloper, 2021.09.02., https://www.gamedeveloper.com/business/game-engines-on-steam-the-definitive-breakdown#close-modal

[4] David Silver "Cooperative pathfinding" AIIDE'05: Proceedings of the First AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, Pages 117 - 122, 2005