

강화학습을 활용한 레이싱 게임에서의 동선 최적화 및 사용자 가이드라인 생성 연구

김 정 민¹ · 우 탁^{2*}¹경희대학교 소프트웨어융합학과 학사과정²경희대학교 일반대학원 메타버스학과 교수

The Study on Path Optimization and User Guidance Generation in Racing Games Using Reinforcement Learning

Jung-Min Kim¹ · Tack Woo^{2*}¹Bachelor's Course, Department of Software Convergence, Kyung Hee University, Yongin 17104, Korea²Professor, Department of Metaverse, Graduate School, Kyung Hee University, Yongin 17104, Korea

[요 약]

게임 인공지능(AI)은 최근 몇 년간 급격히 발전하여 실시간 전략 게임 등 다양한 장르에서 활용되고 있다. 그러나 레이싱 게임에서 AI는 주로 사용자와의 경쟁에 집중되어 있으며, 사용자는 처음 접하는 트랙에서 주행에 어려움을 느끼는 경우가 많다. 만약 AI가 사용자에게 주행 가이드를 제공할 수 있다면, 사용자는 익숙하지 않은 트랙에서도 편안하게 주행할 수 있을 것이다. 강화학습은 다양한 게임 AI에 적용되며, 여러 문제 해결의 도구로 주목받고 있다. 본 연구에서는 Unity의 Machine Learning-Agents와 강화학습을 활용하여 레이싱 게임에서 트랙을 완주할 수 있는 AI를 개발하고, 이를 통해 사용자에게 주행 가이드를 제공하는 방안을 제안한다. 또한 가이드 제공 유무에 따른 사용자의 주행 기록을 비교 분석하여 가이드의 효율성을 평가하고자 한다. 이를 통해 AI 기반 가이드의 실용성을 확인하고, 향후 레이싱 게임에 활용될 수 있는 AI 가이드 시스템의 가능성을 탐구하고자 한다.

[Abstract]

In recent years, game AI has advanced rapidly and is now widely used across various genres, including real-time strategy games. However, in racing games, AI is primarily designed to compete with players, often leaving them struggling to navigate unfamiliar tracks. Providing an AI-guided system could help players drive with ease on new tracks. Reinforcement learning (RL) has emerged as a promising approach to in-game AI, addressing various challenges. This study leverages Unity's ML agents and RL to develop an AI capable of completing racing tracks and providing real-time guidance to players. Furthermore, by comparing player performance with and without guidance, the study evaluates the influence of AI-based guidance on user experience and driving performance. Through this analysis, we aim to validate the practicality of AI guidance and explore its potential for future application in racing games.

색인어 : 강화학습, 레이싱 게임, 게임 인공지능, 인공지능 가이드, 사용자 경험**Keyword** : Reinforcement Learning, Racing Game, Game AI, AI Guide, User Experience<http://dx.doi.org/10.9728/dcs.2025.26.1.203>

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 14 November 2024; Revised 13 December 2024

Accepted 23 December 2024

*Corresponding Author; Tack Woo

Tel: +82-31-201-3890

E-mail: twoo@khu.ac.kr

1. 서론

게임 인공지능(AI)은 최근 몇 년간 급격히 발전하여 다양한 게임 장르에서 널리 사용되고 있다. 실시간 전략 게임에서는 다양한 AI가 연구되고 있지만[1], 레이싱 게임에서는 AI가 상대적으로 적게 활용되고 있다. 현재 레이싱 게임에서 사용되는 AI는 주로 사전에 설정된 경로를 따르거나, 단순한 경로 탐색 알고리즘을 사용해 주행하는 것이 일반적이다. 그리고 이러한 방식은 대부분 사용자와의 대결에서 많이 쓰인다. 그렇기에 사용자가 처음 보는 트랙을 주행할 때는 무작정 부딪히며 트랙을 익히거나, 공략 영상을 통해 학습하는 비효율적인 방법을 주로 사용하게 된다. 만약 AI가 실시간으로 최적의 경로를 탐색하여 사용자에게 가이드를 제공한다면 훨씬 효율적이고 몰입감이 있는 경험을 제공할 수 있을 것이다.

강화학습(Reinforcement Learning, RL)[2]은 이러한 문제를 해결할 수 있는 강력한 도구로 주목받고 있다. 강화학습은 에이전트가 환경과 상호작용하며 스스로 최적의 행동을 학습하는 기법으로 다양한 상황에서 적응력과 학습 능력을 발휘한다. 특히 레이싱 게임과 같이 실시간 의사결정이 중요한 환경에서는 에이전트가 주행 경로, 속도 제어, 회전각도 등의 다양한 요소를 학습하여 동적인 경로 탐색을 실현할 수 있다. 이를 활용하면 AI는 단순히 경로를 주행하는 데 그치지 않고, 실시간으로 사용자에게 경로의 가이드를 제공하는 역할을 할 수 있을 것으로 기대된다.

기존 레이싱 게임에서는 주로 사전 설정된 경로를 따르는 AI가 사용되었지만, 본 연구에서는 Proximal Policy Optimization (PPO)[3] 알고리즘을 이용한 강화학습을 통해 동적 경로 탐색과 사용자 가이드 제공을 구현하고자 한다. PPO는 안정적인 정책 탐색과 빠른 수렴을 제공하여 복잡한 환경에서도 효과적으로 학습할 수 있는 알고리즘으로 이를 통해 에이전트가 다양한 경로와 속도 상황에 맞는 최적의 주행 전략을 학습할 수 있다.

본 논문에서는 PPO 알고리즘을 적용한 레이싱 게임 AI의 동선 최적화 방법을 연구하고 이를 통해 사용자에게 실시간으로 경로 가이드를 제공하는 방법을 제시하고자 한다. 또한 실제 사용자를 대상으로 한 평가를 통해 제안된 방법의 성능을 검증할 계획이다. 이를 통해 레이싱 게임에서의 사용자 경험을 향상시키는 새로운 AI 시스템을 제안한다.

II. 이론적 배경 및 분석

2-1 강화학습

강화학습은 에이전트(Agent)가 환경(Environment)과 상호작용하며 최적의 행동을 학습하는 방법이다. 에이전트는 환경으로부터 상태(State)를 받아들이고 그에 따라 행동(Action)을 선택한 후 보상(Reward)을 받는다. 이러한 과정

을 반복하면서 에이전트는 보상을 최대화할 수 있는 정책(Policy)을 학습한다. 또한, 에이전트는 탐색(Exploration)과 활용(Exploitation)을 적절히 활용하여, 새로운 행동을 시도하는 동시에 이미 학습한 최적의 행동을 반복적으로 수행한다. 그림 1은 강화학습의 기본적인 학습 과정을 시각적으로 표현한 것이다. 여기서 S_t 는 현재의 상태를 나타내고, a_t 는 현재 상태에서 선택하는 행동이다. 행동을 선택한 후 에이전트는 r_t 라는 보상을 얻고 S_{t+1} 의 상태가 된다.

1) 구성 요소

• 에이전트(Agent)

학습을 수행하는 주체로 환경에서 행동을 선택하고 결과에 따라 보상을 받는다.

• 환경(Environment)

에이전트가 상호작용하는 시스템으로, 에이전트의 행동에 따라 상태와 보상을 반환한다.

• 정책(Policy)

에이전트가 각 상태에서 취할 행동을 결정하는 규칙이다. 강화학습의 목표는 최적의 정책을 학습하는 것이다.

• 보상(Reward)

에이전트의 행동이 얼마나 좋은지 평가하는 기준이다.

• 가치함수(Value Function)

상태 또는 상태-행동 쌍의 장기적인 보상 기대치를 나타내며 에이전트는 가치함수를 기반으로 최적의 행동을 선택한다.

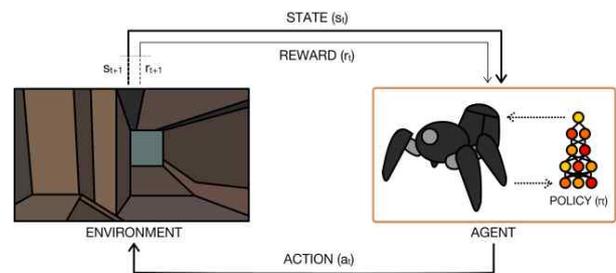


그림 1. 강화학습의 기본 구조[4]
Fig. 1. of reinforcement learning[4]

2) 가치 기반 알고리즘

가치 기반 알고리즘은 상태나 상태-행동 쌍의 가치를 추정하여 최적의 행동을 학습하는 방법이다. 이 알고리즘은 주어진 상태에서 행동의 가치를 나타내는 Q-함수 또는 V-함수를 학습하고, 이를 통해 최적의 행동을 선택한다.

• Q-learning

Q-learning[5]은 가치 기반 학습의 알고리즘으로 에이전

트는 각 상태-행동 쌍의 Q값(장기적인 보상 기대치)을 학습한다. 학습이 완료되면 에이전트는 상태에서 Q값이 가장 큰 행동을 선택함으로써 최적의 정책을 수행한다.

• Deep Q-Network(DQN)

DQN[6]은 Q-learning에 딥러닝을 결합한 방법으로, 신경망을 사용하여 Q값을 추정한다. 이를 통해 고차원적인 상태 공간에서도 효과적으로 학습할 수 있는 장점이 있다.

3) 정책 기반 알고리즘

정책 기반 알고리즘은 에이전트가 직접 정책(Policy)을 학습하여 주어진 상태에서의 행동 확률을 추정하는 방식이다. 가치 기반 방법과 달리 에이전트는 상태-행동의 가치를 추정하지 않고 최적의 정책을 직접 찾는다.

• Trust Region Policy Optimization (TRPO)

TRPO[7]는 정책 기반 알고리즘 중 하나로, 에이전트가 학습 도중 급격한 정책 변화를 방지하면서 더 안정적인 학습을 할 수 있도록 설계된 방법이다. TRPO는 기존 정책과 새로운 정책 간의 차이를 제한하는 대신 두 정책 간의 차이가 지나치게 크지 않도록 제약을 건다. 이를 통해 학습 과정에서 정책이 크게 변하는 것을 막아 안정적인 성능 향상과 수렴을 도모한다.

• PPO

PPO는 정책 기반 알고리즘 중 하나로 에이전트가 극단적인 정책 변화를 방지하면서 안정적으로 학습할 수 있도록 만들어진 방법이다. 기존의 정책을 새로운 정책으로 변경할 때, 그 차이가 일정 범위를 넘지 않도록 제한함으로써 안정적인 학습을 제공한다. 이를 위해 PPO는 신뢰가 가능한 영역을 간단히 구현하는 클리핑 방식을 도입하여 정책 업데이트가 과도하게 이루어지는 것을 방지한다. 이러한 접근은 이전의 TRPO의 이론에서 나왔던 것을 활용한 것이며 TRPO보다 구현이 간단하지만 비슷한 성능을 제공한다는 점을 주목받았다.

4) 혼합형 알고리즘

혼합형 알고리즘은 가치 기반과 정책 기반 방법을 결합하여 더 효율적으로 학습할 수 있는 방법이다. 에이전트는 정책을 학습하는 정책 네트워크(Actor)와 가치를 학습하는 비판 네트워크(Critic)를 함께 사용하여 최적의 행동을 결정한다.

• Actor-Critic

Actor-Critic[8] 알고리즘은 정책을 결정하는 ‘Actor’와 행동의 가치를 평가하는 ‘Critic’으로 구성된다. Actor는 행동을 선택하고 Critic은 해당 행동의 가치가 얼마나 좋은지를 평가하여 Actor의 정책을 업데이트한다.

• Advantage Actor-Critic (A2C)

A2C와 Asynchronous Advantage Actor-

Critic(A3C)[9]는 Actor-Critic 구조의 강화된 버전으로 Advantage 함수를 사용하여 정책을 업데이트한다. A3C는 여러 에이전트가 병렬로 학습하면서 정책을 공유해 학습 속도를 향상시킬 수 있다.

2-2 레이싱 게임에서의 AI

레이싱 게임에서의 AI는 주로 플레이어와 경쟁하거나 게임 내 차량을 제어하는 역할을 수행한다. 전통적인 레이싱 게임에서는 AI가 사전에 정의된 경로를 따라 주행하며, 주어진 트랙을 벗어나지 않고 일정한 주행 패턴을 유지하는 방식이 일반적이다. 이러한 방식은 미리 설정된 경로를 따라가며 차량의 가속, 회전, 제동을 조정하는 단순한 알고리즘을 기반으로 한다[10]. 예를 들어 마리오 카트와 같은 아케이드 레이싱 게임에서는 AI가 미리 정의된 경로를 따르며 기본적인 충돌 감지와 속도 조절을 통해 경주를 완료한다[11].

일부 게임에서는 AI의 난이도를 조절할 때 주행 속도만 변화시키는 방식으로 난이도를 조절한다[11]. 이러한 방식은 플레이어에게 기본적인 게임 경험을 제공하지만, AI가 고정된 경로를 따르기 때문에 게임의 도전 과제가 낮아지며, 플레이어가 경로를 직접 학습하거나 전략적으로 AI와 상호작용할 기회가 부족하다.

1) 카트라이더

카트라이더는 넥슨에서 개발한 대표적인 레이싱 게임으로, 다양한 캐릭터와 트랙을 제공하며, 간단한 조작으로 누구나 즐길 수 있는 게임이다. 카트라이더에서는 도전자 모드라는 AI와의 대결 모드가 존재하며 AI는 미리 설정된 경로를 따라 주행한다. 난이도에 따라 AI의 속도가 조정되며, 상위 플레이어의 움직임을 저장해 AI로 활용하기도 한다. 그림 2는 카트라이더의 도전자 모드 화면을 나타낸다.



*The image was written in Korean

그림 2. 카트라이더 도전자 모드

Fig. 2. Kartrider challenge mode

2) 마리오 카트 8 디럭스

마리오 카트 8 디럭스는 닌텐도에서 출시된 인기 레이싱

게임 시리즈의 최신작으로 다양한 캐릭터가 등장하며 아이템을 활용한 플레이가 특징인 게임이다. 이 게임에서 AI는 플레이어가 부족할 때 자리를 채우기 위해 주로 사용되며 아이템을 활용해 플레이어를 방해하거나 속도를 조절하여 경쟁한다. 난이도에 따라 AI의 속도와 아이템 사용 빈도가 조정된다. 그림 3은 마리오 카트8 디럭스의 게임 화면을 나타낸다.



그림 3. 마리오 카트8 디럭스 게임 화면
 Fig. 3. Mario Kart8 deluxe game screen

3) 기존 연구

기존 연구들은 주로 플레이어에게 경로를 안내하기보다는 AI가 주어진 경로를 완주하는 데 초점을 맞춰왔다. 예를 들어, NeuralKart 연구에서는 CNN을 활용해 장애물을 인식하고, AI가 마리오 카트 64에서 경로를 완주할 수 있도록 하는 시스템을 개발했다[12]. Wang과 Lin의 연구에서는 동적 탐색 알고리즘을 수정하여 경로 탐색의 효율성을 높이는 데 중점을 두었다[13]. Pujianto의 연구에서는 유전 알고리즘을 최적화하여 레이싱 AI가 경로를 효율적으로 주행할 수 있도록 학습하는 시스템을 개발하였다. 이 연구에서는 학습률과 모멘텀을 조정하여 훈련 속도를 가속화하고 정확도를 향상시키는 기법을 적용하였다[14].

이처럼 기존 연구들은 AI의 경로 탐색과 완주에 중점을 두었지만 본 연구는 AI가 최적의 경로를 탐색하고 사용자에게 경로를 안내함으로써 사용자 경험을 더욱 향상시키고자 한다.

III. 연구 방법 및 구현

3-1 레이싱 게임 구현

본 연구에서는 플레이어가 한 바퀴를 도는 것을 목표로 하는 간단한 레이싱 게임을 Unity를 사용해 구현하였다. 그림 4는 이 논문에서 구현한 레이싱 게임의 기본 화면과 게임 환경을 구성하는 주요 요소들을 보여준다.

게임 환경에는 네 가지 주요 오브젝트가 포함된다. 첫째로, 경로 역할을 하는 로드 오브젝트와 경로 외부의 잔디 오브젝트가 있으며 둘째로, 인공지능의 경로를 방해하는 장애물 오브젝트가 존재한다. 셋째로, 중간 경유지 역할을 하는 체크포인트 오브젝트가 있으며 마지막으로, 인공지능이 적용된 플레

이어 오브젝트가 있다.

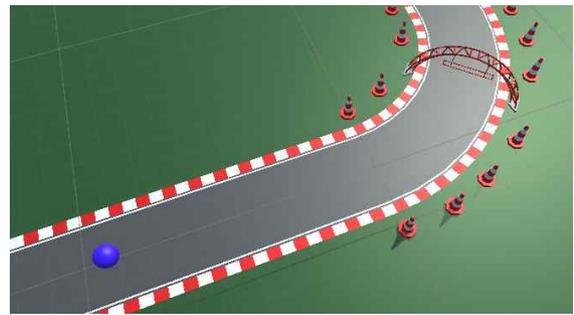


그림 4. 레이싱 게임 환경
 Fig. 4. Racing game environment

본 레이싱 게임에서 플레이어 오브젝트는 Unity의 AddForce 함수를 통해 이동한다. 이 함수는 물리적인 힘을 구현해주며 플레이어는 실제 주행과 유사한 환경에서 이동할 수 있다. 또한, 마찰력과 원심력의 공식을 기반으로 추가적인 물리 함수를 구현하였기 때문에 플레이어는 이러한 힘들고 고려하며 주행하게 된다[15]. 이러한 물리적 요소는 게임에서 더 현실감 있는 주행 경험을 제공한다.

3-2 AI 구현

1) ML-Agents

Unity에서는 AI를 구현하기 위한 도구로 ML-Agents를 지원한다. ML-Agents는 강화학습을 사용하여 AI를 구현할 수 있는 오픈소스 툴로 학습 가속화, 학습 안정성 향상, 자원 사용 감소 등 다양한 기능을 제공한다[16]. 본 논문에서는 이러한 ML-Agents를 사용하여 AI를 구현했다.

2) PPO 알고리즘

PPO는 정책 기반 강화학습 알고리즘 중 하나로 정책의 급격한 변화를 제한하여 안정적인 학습을 가능하게 한다. Clipping 기법을 사용하여 정책 업데이트 시 현재 정책과 이전 정책 간의 차이를 제한함으로써 안정성을 유지하며, 이러한 특성 덕분에 PPO는 다른 알고리즘보다 구현이 간편하고 학습 속도가 빠르다. 본 연구에서는 PPO의 안정적인 탐색과 효율적인 정책 업데이트 기능을 활용하여 Agent가 실시간으로 주행 경로를 최적화하고 환경에 적응할 수 있도록 하였다.

3) 학습 환경

에이전트는 매 상태마다 행동을 선택한다. 본 논문에서는 에이전트의 행동을 2가지로 나누었다. 첫 번째는 이동이고 두 번째는 회전이다. 이동은 전진, 후진, 정지의 3가지 행동이 가능하며 회전은 왼쪽, 오른쪽, 정지의 3가지 행동이 가능하다. 이렇게 에이전트는 총 6가지의 행동을 통해 학습을 진행한다.

에이전트는 여러 가지의 관찰값을 받는다. 매 상태마다 다음 체크포인트의 위치, 에이전트의 위치, 에이전트의 속도, 에이전트의 방향을 받는다. 매 상태마다 이러한 관찰값을 이용

해 에이전트는 다음 행동을 선택한다. 선택한 행동에 따라 다음 상태가 정해지고 에이전트는 이러한 행동을 계속 반복한다.

학습이 시작되면 에이전트는 무작위 방향으로 이동한다. 에이전트는 매초 보상을 잃기 때문에 다양한 곳을 가보려고 시도한다. 올바른 경로로 진행할 경우 보상의 값이 조금 더 올라가기 때문에 에이전트는 경로로 이동하려고 노력하고 그러다가 우연히 체크포인트에 닿을 경우 보상을 2점 얻는다. 학습을 계속 하면서 에이전트는 본인만의 최적의 경로를 찾아가며 학습한다.

에이전트는 올바른 체크포인트에 도달할 경우 2점, 모든 체크포인트를 돌고 결승점에 도달할 경우 5점을 획득하고 장애물에 부딪히면 1.5점을 잃는다. 또한 에이전트가 가만히 있는 것을 방지하기 위해 매 상태마다 0.005점을 잃는다. 에이전트의 이동방향과 로드 오브젝트의 방향을 내적을 이용하여 계산한 후 해당 값에 작은 값을 곱해 에이전트는 보상을 획득한다. 그리고 에이전트는 다음으로 이동해야할 체크포인트까지의 거리를 보상으로 획득한다. 보상들을 통해 에이전트는 완주를 향해 달려가게 된다. 이러한 보상 구조는 표 1에 정리되어 있다. 표 2는 연구에서 사용된 주요 파라미터와 설정값을 보여준다.

표 1. 에이전트의 보상

Table 1. Reward of agents

State	Reward
Per Second	-0.005
Stationary	-0.1
Checkpoint(correct)	2
Checkpoint(wrong)	-1
Obstacle	-1.5
Goal	5

표 2. 학습의 파라미터

Table 2. Parameter of learning

Parameter	Description	Value
Batch Size	The number of data samples used in a single update	1024
Buffer Size	The maximum size of the buffer that stores training data	4096
Learning Rate	Determines how quickly the policy is updated during training	0.0003
Beta	Adjusts the entropy bonus of the policy to balance exploration and exploitation	0.005
Num Epoch	The number of training iterations per cycle	3
Gamma	The discount factor for rewards, determining the impact of future rewards	0.95
Max Steps	The maximum number of steps before training termination	1,000,000
Epsilon	Controls the allowable range of changes when updating the policy	0.2

학습이 완료된 AI의 보상과 손실 값은 그림 5와 그림 6에서 나타난다.

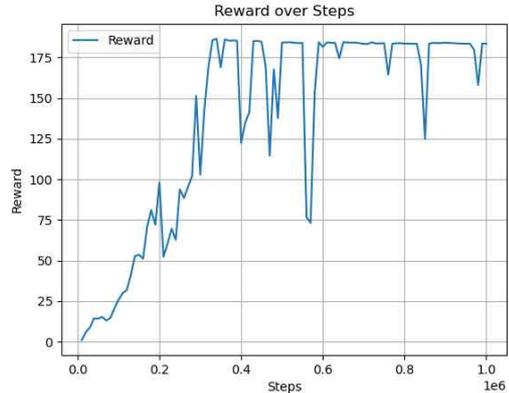


그림 5. Agent의 보상 그래프
Fig. 5. Agent's reward graph

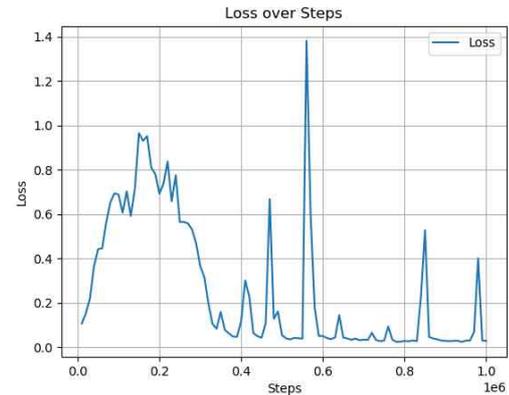


그림 6. Agent의 손실 그래프
Fig. 6. Agent's loss graph

그림 5는 학습 과정 중 에이전트가 얻은 보상값의 변화를 나타낸다. 초기에는 보상이 급격히 변동하며 불안정한 모습을 보이지만, 학습이 진행됨에 따라 점차 상승하고 수렴하는 경향을 보인다. 중간에 보상이 일시적으로 하락하는 패턴이 관찰되는데 이는 탐색과 활용과정에서 에이전트가 새로운 경로를 시도하거나 정책이 변경되었을 때 발생하는 자연스러운 현상이다. 이후 다시 보상이 회복되고 수렴하는 것을 통해 에이전트가 최적 경로를 학습하면서 전반적인 성능이 향상되었음을 확인할 수 있다.

그림 6은 학습 과정 중 손실의 변화를 보여준다. 초기에는 손실이 높게 나타나지만 학습이 진행됨에 따라 손실 값이 점차 감소하고 수렴하는 양상을 보인다. 특히 학습 도중 손실 값이 일시적으로 증가하는 구간이 관찰되는데 이는 정책 업데이트 과정에서 발생할 수 있는 현상으로 에이전트가 새로운 시도를 하거나 탐색 과정에서 불안정해질 때 나타난다. 그러나 학습 후반부로 갈수록 손실 값이 줄어들며 에이전트가 안정적으로 학습하고 경로 최적화에 성공했음을 나타낸다.

3-3 테스트 환경 구현

에이전트의 동선이 가이드를 제공하기 위해 학습이 완료된 에이전트가 이동한 경로를 하얀색으로 표시하는 기능을 추가 하였다. 또한, 사용자가 직접 조작할 수 있도록 새로운 게임 오브젝트를 만들어 에이전트가 사용한 가속도와 동일한 가속도를 적용하였다. 사용자는 3초의 대기 시간을 가지고 출발하며 에이전트와 마찬가지로 모든 체크포인트를 지나 도착지점에 도달하면 게임이 종료된다. 게임이 끝났을 때 소요 시간이 결과에 표시된다.

그림 7은 학습이 완료된 에이전트가 주행한 경로를 하얀색으로 시각화한 모습을 보여준다. 이 동선은 사용자가 AI의 경로를 따라 주행할 수 있도록 가이드를 제공하는 역할을 한다. 그림 8에서는 사용자가 직접 조작하는 플레이어 오브젝트가 게임 환경 내에 있는 모습을 나타낸다. 그림 9에서는 테스트 환경의 전체 경로를 나타낸다.



그림 7. 에이전트의 가이드 동선
Fig. 7. Agent's guide movement line

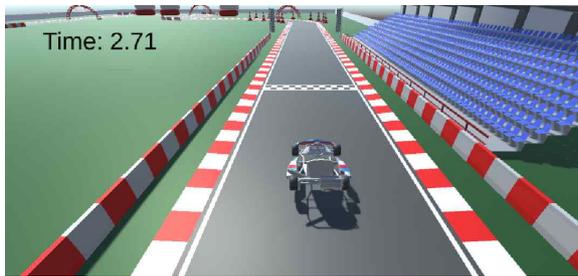


그림 8. 플레이어 오브젝트
Fig. 8. Player object

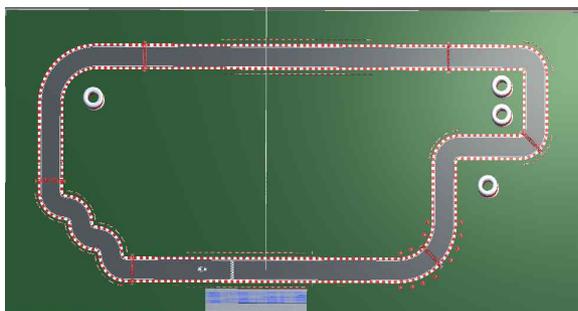


그림 9. 테스트 환경 경로
Fig. 9. Test path

IV. 실험 결과

에이전트의 동선이 사용자에게 도움이 되었는지를 판단하기 위해 테스트를 진행했다. 그러나 동선을 보지 않고 진행한 뒤 동선을 보고 진행하게 된다면 사용자의 시간 단축이 동선 덕분인지 아니면 트랙에 대한 익숙함 덕분인지 분명하지 않기 때문에 본 테스트에서는 첫 테스트와 두 번째 테스트를 모두 동선을 보지 않고 진행하여 더 빠른 기록을 측정했다. 그 후 학습된 에이전트가 그린 동선을 보고 따라가는 방법으로 기록을 측정했다.

4-1 테스트 결과



그림 10. 실험을 진행하는 참여자
Fig. 10. Image of an experiment participant

표 3. 참가자 테스트 결과
Table 3. Participant's test result

Participant	Test1	Test2	Test3
1	55.81	55.53	54.35
2	59.17	56.74	54.80
3	57.71	56.29	54.66
4	56.01	55.40	54.59
5	60.38	56.14	54.83
6	56.31	56.41	55.31
7	57.24	56.27	54.27
8	57.91	56.87	53.97
9	64.93	57.80	54.42
10	55.33	54.89	54.13
11	56.72	56.05	54.87
12	57.27	55.26	54.26
13	59.93	56.46	54.62
14	56.01	56.08	54.61
15	55.67	55.61	54.45
16	56.74	56.37	54.42
17	55.97	56.25	54.09
18	56.31	56.03	54.94
19	55.70	55.75	54.45
20	55.92	55.55	54.33

20대 남녀 구분 없이 총 20명의 참가자를 모집하여 실험을 진행했다. 참가자들에게는 간단한 조작법만 알려진 채로 첫 번째와 두 번째 실험을 진행했고, 그 뒤에 AI를 따라가라는 것만 알려준 채로 세 번째 실험을 진행했다. 그림 10은 실험하는 참가자의 모습을 찍은 것이고 표 3은 참가자 20명의 결과를 나타낸다.

4-2 결과 분석

표 4는 첫 번째 테스트에서 두 번째 테스트 사이, 그리고 두 번째 테스트에서 세 번째 테스트 사이에 감소한 주행을 나타냈다. 첫 번째에서 두 번째 테스트로 넘어가면서 대부분의 참가자가 조작감에 익숙해짐에 따라 시간이 평균적으로 1.2645초가 감소했고, 두 번째에서 세 번째 테스트에서는 AI의 동선을 참고하여 주행하면서 평균적으로 1.569초가 감소했다. 다만 첫 번째 테스트 때 조작을 어려워하던 참가자들의 경우에는 조작감에 의한 감소가 훨씬 크게 나타났다.

표 4. 테스트 결과 분석

Table 4. Analysis of test result

Participant	Test1 - Test2	Test2 - Test3
1	0.28	1.18
2	2.43	1.94
3	1.42	1.63
4	0.61	0.81
5	4.24	1.31
6	-0.1	1.1
7	0.97	2
8	1.04	2.9
9	7.13	3.38
10	0.44	0.76
11	0.67	1.18
12	2.01	1
13	3.47	1.84
14	-0.07	1.47
15	0.06	1.16
16	0.37	1.95
17	-0.28	2.16
18	0.28	1.09
19	-0.05	1.3
20	0.37	1.22
AVG	1.2645	1.569

처음부터 좋은 기록을 내던 1번과 10번 참가자의 경우 AI의 동선이 효과를 주었지만, 그 효과가 상대적으로 작았다. 5번과 9번 참가자의 경우 첫 번째 시도에서는 1분이 넘는 기록이 걸렸지만 테스트를 진행하면서 기록을 5초 이상 단축하는 모습을 보였다. 반면 6번, 14번, 17번, 19번 참가자는 첫 번째 테스트에서 조작에 어려움을 느끼지 않아 오히려 두 번째

테스트에서 시간이 증가하는 현상을 보였다.

이 실험 결과는 AI 가이드가 사용자 주행 시간을 단축하는데 유의미한 도움을 줄 수 있음을 나타낸다. 모든 사용자가 AI 가이드가 제공하는 최적의 동선을 따름으로써 주행 시간을 단축할 수 있었고 이를 통해 사용자 경험이 더욱 향상될 가능성을 확인할 수 있었다.

그러나 마지막 테스트의 경우, AI 가이드 동선에 따라 시간이 감소했는지 조작감에 의해 감소했는지 정확하게 알 수 없다는 한계점이 존재한다. 난이도가 비슷한 다른 경로를 이용하여 테스트를 추가로 진행하거나 레이싱 게임에 익숙한 참가자로 하는 테스트가 추후에 필요하다.

V. 결 론

본 논문에서는 레이싱 게임 사용자에게 AI를 활용한 주행 가이드를 제공하는 연구를 수행했다. AI는 Unity와 ML-Agents 기반의 강화학습을 통해 학습되었으며 그 결과 대부분의 참가자가 AI 가이드를 통해 주행 시간을 단축하는 데 성공하였다. 조작감이 익숙해짐에 따른 시간 감소 효과를 고려하더라도 AI 가이드는 긍정적인 영향을 미쳤으며 특히 초기에 게임 조작에 어려움을 느꼈던 일부 참가자에게 더 큰 효과를 나타냈다.

본 연구는 AI 기반 가이드 시스템이 사용자 주행 성능 향상에 기여할 수 있음을 보여주었으며 이는 사용자 맞춤형 지원 제공에 중요한 의미를 가진다. 또한 본 연구의 결과는 레이싱 게임에 국한되지 않고 AR/VR 환경에서의 주행 훈련, 시뮬레이션 기반 교육 및 훈련 프로그램 등 다양한 분야에서 AI 가이드의 활용 가능성을 제시한다.

그러나 일부 사용자는 AI 가이드를 통해 주행 시간을 크게 단축하지 못하는 경우도 있었으며 가이드가 사용자 만족감 향상과 직접적인 연관성을 입증하기 어려운 문제가 있다. 또한 사용자의 표본이 20명으로 적기 때문에 결과를 전체에 적용하기에는 한계가 존재한다. 또한 조작감에 의한 시간 감소와 AI 가이드 동선에 의한 시간 감소가 어느정도인지 알 수 없기 때문에 추가적인 테스트가 필요하다.

향후 연구에서는 난이도에 따른 다양한 트랙에서의 테스트를 진행하여 지금의 결과와 비교를 진행할 예정이다. 그리고 일반 참가자가 아닌 레이싱 게임에 익숙한 참여자들을 따로 모집해 비교를 진행할 예정이다. 또한 현재 AI보다 더욱 최적화된 알고리즘이 있을 가능성을 고려하여 이에 대한 연구도 병행할 예정이다.

감사의 글

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 정보통신방송통신인재양성(메타버스융합대학원)사업 연구 결과로 수행되었습니다(IITP-2024-RS-2024-00425383).

참고문헌

[1] G. Robertson and I. Watson, "A Review of Real-Time Strategy Game AI," *AI Magazine*, Vol. 35, No. 4, pp. 75-104, 2014. <https://doi.org/10.1609/aimag.v35i4.2478>

[2] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement Learning: A Survey," *Journal of Artificial Intelligence Research*, Vol. 4, pp. 237-285, May 1996. <https://doi.org/10.1613/jair.301>

[3] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," arXiv:1707.06347v2, August 2017. <https://doi.org/10.48550/arXiv.1707.06347>

[4] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "A Brief Survey of Deep Reinforcement Learning," arXiv:1708.05866, August 2017. <https://doi.org/10.48550/arXiv.1708.05866>

[5] C. J. C. H. Watkins and P. Dayan, "Q-Learning," *Machine Learning*, Vol. 8, No. 3-4, pp. 279-292, May 1992. <https://doi.org/10.1007/BF00992698>

[6] M. Roderick, J. MacGlashan, and S. Tellex, "Implementing the Deep Q-Network," arXiv:1711.07478, November 2017. <https://doi.org/10.48550/arXiv.1711.07478>

[7] J. Schulman, S. Levine, P. Moritz, M. Jordan, and P. Abbeel, "Trust Region Policy Optimization," arXiv:1502.05477v1, February 2015. <https://doi.org/10.48550/arXiv.1502.05477>

[8] J. Peters and S. Schaal, "Natural Actor-Critic," *Neurocomputing*, Vol. 71, No. 7-9, pp. 1180-1190, March 2008. <https://doi.org/10.1016/j.neucom.2007.11.026>

[9] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Harley, T. P. Lillicrap, ... and K. Kavukcuoglu, "Asynchronous Methods for Deep Reinforcement Learning," arXiv:1602.01783v1, February 2016. <https://doi.org/10.48550/arXiv.1602.01783>

[10] A. Nguyen and J. Kittur, "Exploring the Use of Artificial Intelligence in Racing Games in Engineering Education: A Systematic Literature Review," in *Proceedings of 2024 ASEE Annual Conference & Exposition*, Portland: OR, June 2024. <https://doi.org/10.18260/1-2--47040>

[11] S. Tomlinson and N. Melder, An Architecture Overview for AI in Racing Games, in *Game AI Pro: Collected Wisdom of Game AI Professionals*, Boca Raton, FL: CRC Press, ch. 38, pp. 471-479, 2014.

[12] GitHub. NeuralKart: A Real-Time Mario Kart 64 AI [Internet]. Available: <https://github.com/rameshvarun/NeuralKart>.

[13] J.-Y. Wang and Y.-B. Lin, "An Effective Method of Pathfinding in a Car Racing Game," in *Proceedings of the 2nd International Conference on Computer and*

Automation Engineering (ICCAE), Singapore, pp. 544-547, February 2010. <https://doi.org/10.1109/ICCAE.2010.5452074>

[14] Pujianto, A. "Implementation of Genetic Algorithms in the Application of Car Racing Games," *Indonesiang Journal of Artificial Intelligence and Data Mining*, Vol. 4, No. 1, pp. 29-34, March 2021. <https://doi.org/10.24014/ijaidm.v4i1.10835>

[15] J. D. González, J. H. Escobar, H. Sánchez, J. De la Hoz, and J. R. Beltrán, "2D and 3D Virtual Interactive Laboratories of Physics on Unity Platform," in *Proceedings of the 4th International Meeting for Researchers in Materials and Plasma Technology (IMRMPT)*, Santa Marta: CO, 012069, May 2017. <https://doi.org/10.1088/1742-6596/935/1/012069>

[16] D. Kim and H. Jung, "Performance Analysis of Target Tracking AI Based on Unity ML-Agents," *Journal of KIIT*, Vol. 19, No. 12, pp. 19-26, December 2021. <https://doi.org/10.14801/jkiit.2021.19.12.19>



김정민(Jung-Min Kim)

2018년~현 재: 경희대학교 소프트웨어융합학과 학부과정
※ 관심분야: 강화학습(Reinforcement Learning), 게임 인공지능(Game AI)

우탁(Tack Woo)

2002년 : University of Dundee (UK),
Electronic Imaging. BA
(Honours)

2004년 : University of Dundee (UK),
Electronic Imaging. MSc (이
학석사)

2010년 : University of Dundee (UK),
Electronic Imaging. (게임학),
PhD (이학박사)



2004년~2007년: University of Dundee, Lecturer
2007년~2010년: KAIST 엔터테인먼트 공학연구소, 연구원
(기능성 게임랩)
2010년~2012년: KAIST 문화기술대학원, Digital Art &
Entertainment Track 초빙교수 (게임)
2012년~2013년: 서울대학교 차세대융합기술연구원, 게임융합
미디어연구센터 센터장
2013년~2024년: 경희대학교 디지털콘텐츠학과 교수
2024년~현 재: 경희대학교 일반대학원 메타버스학과 교수
※ 관심분야: 기능성 게임, 게임화, 게임문화, VR/AR 콘텐츠