

사람과 강화학습 인공지능의 게임플레이 유사도 측정

허민구, 박창훈

hucce@imrlab.hoseo.edu, chpark@hoseo.edu

Measuring gameplay similarity between human and reinforcement learning artificial intelligence

Min-Gu Heo, Chang-Hoon Park

요약

최근, 사람 대신 인공지능 에이전트를 이용하여 게임 테스트를 자동화하는 연구가 관심을 모으고 있다. 본 논문은 게임 밸런싱 자동화를 위한 선행 연구로써 사람과 인공지능으로부터 플레이 데이터를 수집하고 이들의 유사도를 분석하고자 한다. 이때, 사람과 유사한 플레이를 할 수 있는 인공지능의 생성을 위해 학습 단계에서 제약사항을 추가하였다. 플레이 데이터는 14명의 사람과 60개의 인공지능을 대상으로 플리피버드 게임을 각각 10회 실시하여 획득하였다. 수집한 데이터는 코사인 유사도 방법으로 이동 궤적, 액션 위치, 죽은 위치를 비교 분석하였다. 분석 결과 사람과의 유사도가 0.9 이상인 인공지능 에이전트를 찾을 수 있었다.

ABSTRACT

Recently, research on automating game tests using artificial intelligence agents instead of humans is attracting attention. This paper aims to collect play data from human and artificial intelligence and analyze their similarity as a preliminary study for game balancing automation. At this time, constraints were added at the learning stage in order to create artificial intelligence that can play similar to humans. Play datas obtained 14 people and 60 artificial intelligence by playing Flippy bird games 10 times each. The collected datas compared and analyzed for movement trajectory, action position, and dead position using the cosine similarity method. As a result of the analysis, an artificial intelligence agent with a similarity of 0.9 or more with humans was found.

Keywords : 테스트 자동화(Test Automation), 강화학습(Reinforce Learning), 유사도 분석(Similarity analysis), 빅데이터(Big data), 게임 밸런스 디자인(Game balance design)

Received: Oct. 27. 2020. Revised: Dec. 10. 2020.
Accepted: Dec. 16. 2020.
Corresponding Author: Chang-Hoon Park(Hoseo University)
E-mail: chpark@hoseo.edu

ISSN: 1598-4540 / eISSN: 2287-8211

© The Korea Game Society. All rights reserved. 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.

1. 서론

게임 밸런스를 위해서는 테스트를 통해 많은 데이터를 얻어야 한다. 게임 밸런스는 이론적으로 모든 플레이어의 능력을 고려하여 적절한 과업을 제공해야 하지만 이는 현실적으로 어렵다. 따라서 대다수의 경우 사용자의 플레이 데이터를 분석하여 밸런스 디자인을 한다[1]. 이러한 방법은 정확한 분석을 위해서는 많은 수의 데이터가 필수이다. 이에 게임사들은 테스터를 고용하거나, 사람들을 선정하여 베타테스트를 진행한다[2,3,4].

이러한 테스트의 비용 절감을 위해 게임에서 테스트 자동화에 관한 연구가 이루어졌다. 테스트 자동화는 반복적인 작업과 수동으로 수행하기 어려운 테스트 작업을 자동화하는데 사용된다. 이는 테스트 스크립트를 작성하거나, 자동화 테스트 도구를 사용하여 수행한다[5]. 게임엔진인 유니티, 언리얼부터 AWS, 애저, 파이어베이스 같은 클라우드 플랫폼도 자동화 테스트 도구를 지원한다[6]. 최근에는 테스트 스크립트를 대신해서 딥러닝 인공지능을 이용하는 테스트 자동화에 관한 연구가 이루어지고 있다[7].

하지만 밸런스 테스트는 자동화가 어려웠는데, 이는 인공지능이 사람과 유사한 플레이를 해야 밸런스 테스트 자동화가 가능하기 때문이다. 밸런스 테스트 자동화는 사람 대신 인공지능이 게임플레이를 수행하는 방법으로 플레이데이터를 수집한다. 그런데 비디오게임에서 제약사항 없이 인공지능을 제작할 경우 초인적인 플레이를 하므로, 밸런스 테스트에서 인공지능을 적용할 수 없었다[8]. 초인적인 플레이 데이터를 밸런스에 적용하면 사람은 게임을 클리어 할 수 없을 것이다. 그렇기에 밸런스 테스트 자동화를 위해서는 사람과 유사한 플레이를 하는 인공지능에 관한 연구가 필요하다.

본 논문은 게임 밸런스 테스트 자동화를 위한 선행 연구로써 사람과 인공지능으로부터 플레이 데이터를 수집하고 이들의 유사도를 분석하고자 한다. 이때, 사람과 유사한 플레이가 가능한 인공지

능 생성을 위해 기존의 인공지능 연구에서 제안된 제약사항을 추가하여 인공지능을 생성한다. 생성된 강화학습 인공지능과 실제 사람의 게임플레이를 코사인 유사도를 이용한 평가방법으로, 인공지능과 사람의 게임플레이 유사도를 비교 분석한다.

2. 관련 연구

2.1 사람과 유사한 인공지능

인공지능의 초인적인 능력 문제를 해결하기 위해, 인공지능의 능력을 제약하는 방법이 제시되었다. 게임 인공지능 연구는 게임을 문제없이 플레이하는지 확인하는 성능중심이었다[9]. 예를 들어 대전형태의 게임은 높은 실력의 사람을 이기는 것을, 레벨형태의 게임은 높은 점수를 내는 것이 목표였다. 하지만 비디오게임에서 성능을 중심으로 인공지능을 제작한다면 인공지능은 초인적인 플레이를 한다[8].

이러한 초인적인 플레이를 하는 문제를 해결하기 위해 제약사항으로 인공지능의 능력을 제한하는 방법이 제시되었다. Ahmed Khalifa는 인공지능이 사람보다 뛰어난 이유를 세 가지로 설명했다. 첫째, 반응속도의 제약을 받지 않는다. 둘째, 행동의 정지가 없다. 셋째, 위험도가 높은 전략은 하지 않는다. 그리고 문제에 대한 해결법을 제시했다. 첫째, 액션 간 지연 추가. 둘째, 아무것도 하지 않는 NULL액션 추가[10]. 해결법을 제시하지 못한 세 번째 문제는, Random Network Distillation(RND) 알고리즘과 같은 강화학습 알고리즘의 발전으로 해결되었다[11]. 이러한 발전으로 복잡한 게임인 스타크래프트2와 도타2를 이용하여 인공지능이 사람과 유사한 게임플레이를 하도록 만들었다[8,12]. 스타크래프트2를 이용한 알파스타는 사람을 초월하는 문제를 해결하기 위해 제한사항으로 분당 명령 횟수의 제한, 지연 추가, 카메라 시야 제한을 제안했다.

2.2 사람과 인공지능 플레이 유사도 측정

사람과 인공지능을 유사도를 측정하는 방법으로 튜링테스트 방법이 제시됐다. 사람의 게임플레이와 유사한 인공지능을 만들려는 시도는 이전부터 있었으며, 이렇게 만들어진 인공지능이 사람과 유사한 플레이를 했는지 평가는 튜링테스트로 평가했다[13,14]. 튜링테스트는 1950년 앨런튜링이 개발한 방법으로 평가자가 기계와 사람을 테스트해 사람과 기계를 구분할 수 있는지 확인하는 방법이다[15]. 하지만 튜링테스트는 사람의 주관적 평가여서 신뢰성을 위해서는 많은 사람이 필요하다.

튜링테스트를 대신할 방법으로 사람과 인공지능을 구분하는 방법으로 코사인 유사도 방법이 제시됐다. 사람과 인공지능을 구분하는 방법은 인공지능 분야보다 오히려 온라인게임에서 주로 연구되었다. 사람 대신에 게임을 플레이하는 자동 게임 봇이 불법적으로 사용되고 있기 때문이다[16]. 게임 봇을 찾는 방법은 다양하게 제시되었지만, 최근에는 데이터를 이용한 방법으로 게임 봇을 찾고 있다. 데이터 분석 방법으로 게임 봇의 플레이 데이터와 일반 유저의 플레이 데이터를 벡터화하여 코사인 유사도 비교를 통해서 게임 봇을 가려내는 방법을 제안했고, 이를 리니지에서 사용 중에 있다[17]. 코사인 유사도는 내적 공간의 두 벡터 간 각도의 코사인값을 이용하여 측정된 벡터 간의 유사한 정도를 의미한다. 두 벡터의 방향이 완전히 같으면 1, 90°의 각을 이루면 0, 180°로 완전히 반대 방향일 때 -1의 값을 갖는다[18].

3. 연구 방법

3.1 인공지능 구성

본 논문은 카메라 시야 제한, 액션 간 지연 추가, 아무것도 하지 않는 NULL액션 추가의 제약사항과 특정 학습량별로 학습된 모델을 저장하도록 인공지능을 구성한다. 학습량별로 모델을 저장하는 이유는 일반적으로 강화학습 인공지능은 학습이 진

행될수록 성능이 올라간다. 따라서 학습이 계속된다면 언젠가 알파고처럼 사람을 초월할 것이다[19]. 그러므로 학습량도 사람과 유사한 플레이를 위해서는 고려할 점으로 볼 수 있다. 또한, 대전형태의 게임은 사실상 무한한 학습이 가능하지만, 스테이지를 진행하는 레벨형태의 게임은 한정된 레벨에서 학습하기 때문에 Overfitting¹⁾ 문제가 발생할 수 있다. 따라서 제안된 제약사항에 특정 학습량별로 학습된 모델을 저장하도록 인공지능을 구성한다.

3.2 연구절차

사람과 인공지능의 데이터 수집을 위해 게임을 제작한다. 제작된 게임은 데이터를 수집할 하나의 레벨을 제작한다. 사람과 특정 학습량의 인공지능당 각각 10회 플레이한 데이터를 수집한다.

수집된 데이터는 신뢰성을 위해 프레임별 위치, 액션 위치, 죽은 위치, 세 가지 방법의 유사도 평균을 낸 종합 네 가지 방법으로 유사도를 분석한다. 사람과 인공지능과의 분석과 사람과 사람 분석, 그리고 유사한 플레이의 사람들을 집단으로 만들어, 사람집단과 인공지능 분석으로 사람과 인공지능이 유사한 플레이를 하는지 3단계로 분석한다.

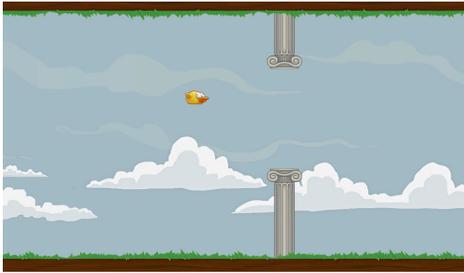
4. 강화학습 인공지능 제작

4.1 게임

본 실험을 위하여 유니티 게임엔진을 이용하여 '플리피버드' 게임을 제작한다. '플리피버드'는 횡스크롤 액션게임으로 새를 조종하여 기둥을 피해 최대한 오래 살아남아야 하는 게임이다. 이 게임의 레벨 디자인은 장해물인 기둥의 높이를 조절하는 것으로 가능하다. 플레이어는 중력으로 낙하하는 새가 기둥 사이로 이동할 수 있도록 타이밍에 맞게 점프동작을 해야 한다. 본래 플리피버드는 무한한 레벨을 가지고 있지만, 본 연구를 위해 클리어가 가능한 최종 목표 지점을 만들었다.

1) 학습 데이터를 과하게 학습하는 것

인공지능 에이전트가 학습할 때 레벨 학습의 다양화를 위해서 초기화 때마다 장애물인 기둥을 X축은 고정 간격으로, Y축은 3.5에서 -1의 범위 내에서 무작위로 생성했다.



[Fig. 1] Developed Flippy Bird

4.2 강화학습

인공지능을 개발하기 위해 유니티에서 제공하는 튜킷인 기계학습 에이전트를 사용했다. 이 튜킷은 강화학습, 모방학습, 신경진화 등의 최신 인공지능 알고리즘을 이용하여 지능형 에이전트를 교육할 수 있는 환경을 제공한다[20]. 우리는 이 환경에서

Proximal Policy Optimization(PPO) 알고리즘을 이용하여 10개의 에이전트를 병렬로 학습시켰다.

보상설정은 게임을 클리어 할 때마다 1점을, 죽을 때마다 -1점을 받는다. 인공지능이 움직이지 않는 것이 효율적이라고 판단하는 문제를 방지하기 위해 매 Step²⁾마다 -0.1점의 보상을 받는다. 빠른 학습을 위해서 기둥을 통과할 때마다 0.1점의 보상을 받도록 설정했다.

Hyper parameter는 인공지능의 문제였던 위험한 전략을 채용을 위해서 보상 신호와 호기심의 값을 높게 설정했다.

```
reward_signals:
  extrinsic:
    gamma: 0.99
    strength: 1.0
  curiosity:
    gamma: 0.99
    strength: 0.02
encoding_size: 256
learning_rate: 0.0003
```

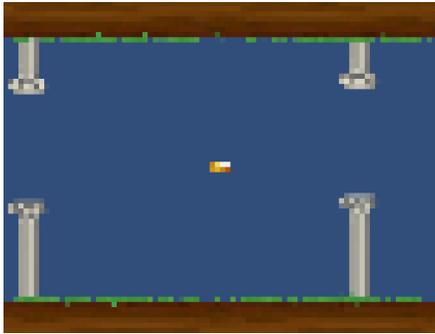
본 논문에서 사람과 유사한 플레이를 할 수 있는 인공지능을 생성하기 위하여 시야 제한, 액션 간 지연, NULL 액션의 제약사항을 반영하였다.

카메라 시야를 사람이 플레이하는 시야 범위와 같도록 설정했다. 관측 정보는 인공지능 제약사항인 시야 제한을 위해서 시각적 관찰을 이용하여 카메라의 정보를 준다. 시각적 관찰은 카메라에 렌더링 된 이미지 정보를 Convolutional Neural Network(CNN) 알고리즘을 이용하여 이미지의 특징을 분류한다. CNN알고리즘은 이미지를 분석하고 패턴을 찾는데 유용한 알고리즘이다[21]. CNN 알고리즘으로 분류한 환경 정보를 통해 학습을 진행하기 때문에, 학습의 효율성이 떨어지지만, 복잡한 문제를 해결할 때 사용한다[22]. 알파스타는 초기에는 게임 전체를 시야 정보로 관측했으나, 인공지능이 사람이 한눈에 볼 수 없는 영역까지 빠르게 조종, 판단하는 문제가 발생했다. 이에 최종버전에서는 인공지능의 시야 범위를 사람과 같은 시야 범위를 설정했다[8]. 따라서 본 논문도 인공지능의 시야정보를 사람과 같은 범위로 설정했다. 그리고 84*64 크기로 렌더링 이미지를 제공했으며, 학습에 불필요한 배경 이미지는 카메라 렌더링에서 제외했다. 카메라는 새를 화면 중앙에 위치하고 따라가도록 했다.

액션 간 5Step의 지연을 추가했다. 인공지능은 반응속도의 제약을 받지 않아, 사람보다 빠른 반응속도로 게임을 플레이한다. 그래서 사람의 반응속도를 고려하여 액션 간 지연을 추가했다[8,10]. 따

2) 학습 시 Weight와 Bias를 업데이트하는 단계

라서 본 논문도 액션 간 5Step의 지연을 추가했다.

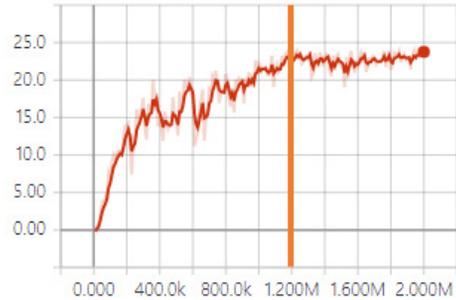


[Fig. 3] Visual observation Rendering image

새가 점프하는 점프액션과 아무것도 하지 않는 NULL 액션을 추가했다. 사람은 때때로 생각을 위해 행동을 정지하는데, 인공지능은 행동의 정지가 없어 사람보다 더 효율적으로 움직인다. 그러므로 아무것도 하지 않는 NULL액션을 추가하여 인공지능이 아무것도 하지 않는 액션을 할 수 있도록 하였다[10]. 따라서 본 논문도 아무것도 하지 않는 NULL액션을 추가했다. 또한, 플리피버드는 NULL액션의 활용이 중요한 게임으로 점프액션을 계속 하는 것이 아닌, 특정한 시점에서는 아무것도 하지 않고 새를 낙하시키는 것도 필요하다. 따라서 인공지능의 NULL액션 활용을 확인하기에 유용하다.

사람과 유사도가 높은 인공지능을 생성하기 위하여 에이전트의 학습량에 따라 서로 다른 모델을 생성하였다. 10개의 에이전트를 병렬로 2,000,000Step까지 학습을 진행하면서 10,000Step 간격으로 모델을 생성하여 총 200개의 인공지능 모델을 생성했다.

Tensor board를 보고 Overfitting이 올 수 있는 학습량을 확인하여, 최대 보상을 얻은 학습량인 1, 200,000Step을 데이터를 수집할 최대학습량으로 선택했다. 그리고 20,000Step 간격으로 총 200개의 모델 중 60개의 모델을 선택했다.



[Fig. 4] Environment/Cumulative Reward in Tensor board

5. 데이터 분석

5.1 데이터 수집

강화학습을 완료한 60개의 인공지능 모델과 14명의 사람을 대상으로 실험을 하여 플레이 데이터를 수집하였다. 이때, 인공지능과 사람의 비교 실험을 위하여 플리피버드 게임의 기동 위치를 고정 한 하나의 레벨을 제작하여 실시하였다.

데이터는 SQL을 이용하여 프레임별로 기록했다. 기록한 데이터 항목은 표1과 같다.

[Table 1] Play Database

Column	Explain
ID	Player ID, AI is one model, Human is one person.
Game	Number of game currently playing.
Frame	Number of frames per game.
Clear	Check if the game is cleared in that frame.
Over	Check if the game is over in that frame.
Jump	Check if jump to the frame.
X Position	X-axis value of bird.
Y Position	Y-axis value of bird.
Time stamp	Recorded time.

인공지능의 플레이 데이터는 60개의 모델을 대상으로 각 10회씩 총 600회를 실시하여 546,368개

[Table 2] AI Playdatas

ID	Game	Frame	Clear	Over	Jump	X Position	Y Position	Time stamp
2	1	1	False	False	False	-21.85	-2.452	PM 4:10:31
2	1	2	False	False	False	-21.75	-2.465	PM 4:10:31
2	1	3	False	False	False	-21.65	-2.488	PM 4:10:31
2	1	4	False	False	False	-21.55	-2.522	PM 4:10:31
2	1	5	False	False	False	-21.5	-2.543	PM 4:10:32
...								
120	10	1349	False	False	False	92.300	-0.449	PM 10:23:41
120	10	1350	False	False	False	92.350	-0.465	PM 10:23:41
120	10	1351	False	False	False	92.451	-0.504	PM 10:23:41
120	10	1352	False	False	False	92.548	-0.554	PM 10:23:41
120	10	1353	True	False	False	92.558	-0.557	PM 10:23:41

의 행데이터를 수집했다.

사람의 플레이 데이터는 14명의 사람을 대상으로 각 10회씩 총 140회를 실시하여 95,473개의 행 데이터를 수집했다. 실험은 2020년 10월 14일부터 18일까지 17명이 참여했다. 이 중 데이터가 문제가 있는 3명을 제외하고 14명으로 분석을 진행하였다. 참여자의 성별로는 남자 13명(92.9%), 여자 1명(7.1%), 연령대는 20대 11명(78.6%), 30대 3명(21.4%)이었다.

5.2 데이터 분석 방법

수집된 게임데이터는 프레임 별 위치, 액션 위치, 죽은 위치, 종합 네 가지 방법으로 코사인 유사도를 분석한다.

코사인 유사도는 속성 A, B의 벡터값이 각각 주어졌을 때, 벡터의 스칼라곱과 크기로 다음과 같이 계산할 수 있다.

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^{\pi} A_i \times B_i}{\sqrt{\sum_{i=1}^{\pi} (A_i)^2} \times \sqrt{\sum_{i=1}^{\pi} (B_i)^2}}$$

프레임별 위치를 이용한 두 게임데이터 사이의 유사도 계산은 프레임별로 X축 좌표값과 Y축 좌표값으로 코사인 유사도를 먼저 계산한다. 이때, 기둥이나 벽에 충돌하여 죽었기 때문에 비교할 좌표값이 없는 경우 0을 대입하여 진행한다. 그리고 두 게임데이터의 프레임 별 유사도들의 평균을 계산한다.

액션 위치를 이용한 두 게임데이터 사이의 유사도 계산은 점프했을 때의 X축 좌표값과 Y축 좌표값으로 코사인 유사도를 먼저 계산한다. 하나의 점프좌표에 비교할 게임의 모든 점프좌표의 코사인 유사도를 계산한다. 이 중 가장 높은 유사도를 기준으로 평균을 계산한다. 이때, 점프 수를 비교하여 수가 같지 않으면 나머지 값에 0을 대입하여 계산한다.

죽은 위치를 이용한 두 게임데이터 사이의 유사도 계산은 죽었을 때의 X축 좌표값과 Y축 좌표값으로 코사인 유사도를 계산한다. 죽지 않고 클리어한 경우 0을 대입하여 계산한다.

마지막으로, 종합은 앞의 세 가지의 유사도의 평균으로 계산한다.

60개의 인공지능 모델과 14명의 사람은 각각 10회의 게임플레이를 실시하였기 때문에, 10회 평균 유사도로 플레이어 사이의 유사도를 계산한다.

계산된 데이터를 히트맵 방식과 게임지도 방식으로 시각화하여 분석된 데이터를 확인한다.

그림5는 플레이어별 유사도를 히트맵으로 시각화하여 제시한다. 프레임별 위치, 액션 위치, 죽은 위치, 종합 각각 히트맵으로 표시한다. 색상의 표시범위는 가시성을 위해 0부터 1까지로 설정한다.

그림8은 게임지도에 위치를 표시하여 유사도가 실제 게임의 움직임과 유사한지 확인한다. 게임 지도에 선과 마커로 해당하는 좌표에 표시하여 시각화한다. 비교하는 플레이어의 색을 다르게 표시하고, 투명도를 주어 위치가 겹치면 색이 바뀌고 표

시가 전해지는 것으로 플레이의 유사함을 확인한다. 프레임별 위치, 액션 위치, 죽은 위치 세 가지 방법을 통해서 확인한다.

5.3 사람과 인공지능 분석

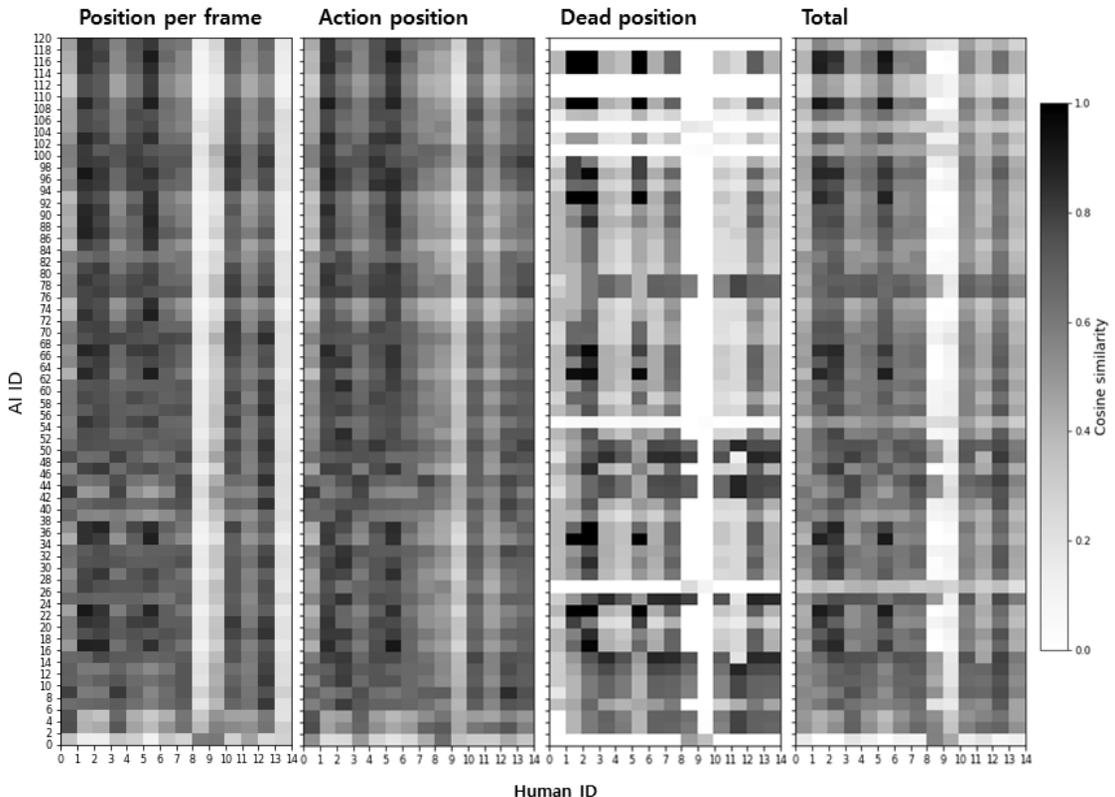
14명의 사람에게서 수집한 140개의 게임 데이터와 60개의 인공지능에서 수집한 600개의 게임 데이터, 총 84,000개의 게임 데이터를 비교했다.

히트맵 분석으로 6번 사람과 1,180,000Step 학습한 인공지능의 게임플레이가 유사하다고 할 수 있었다. 히트맵으로 유사도가 높은 사람과 인공지능을 확인했다. 종합 유사도가 높은 6번 사람과 1,180,000Step 학습한 인공지능 히트맵의 네 가지 분석 모두 높은 유사도를 보였다. 데이터 확인결과 프레임별 위치 0.910, 액션 위치 0.903, 죽음 위치 0.99

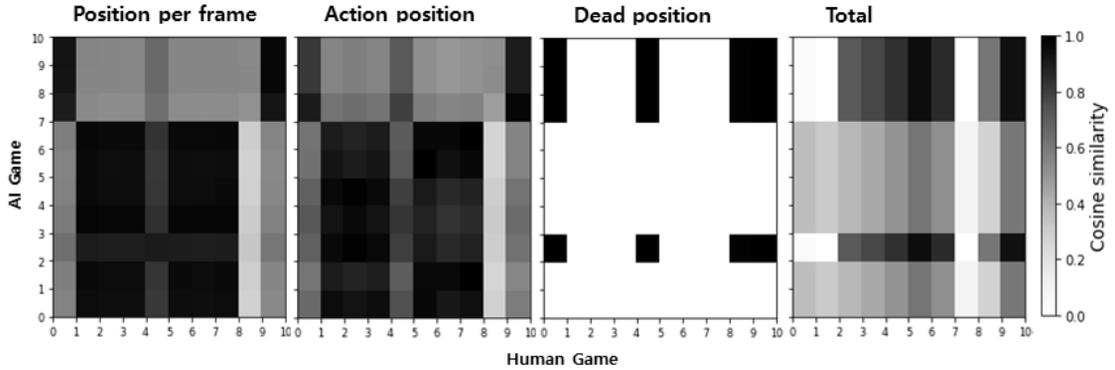
5, 종합 0.936로 높은 유사도를 확인했다.

죽은 위치는 다른 분석 히트맵과 달리 유사도 값끼리 차이가 큰데, 이는 죽은 횟수가 적어 비교군이 적기 때문이다. 또한, 일부 학습량의 경우 거의 모든 사람과 유사도가 낮게 나온다. 확인결과, 1,120,000 - 1,140,000Step 학습량의 인공지능은 모든 게임에서 죽지 않아 유사도가 0이었다. 또한, 특이 집단의 죽은 횟수가 평균 2.41회로 사람 평균 6.8회 인공지능 평균 6.6회보다 죽은 횟수가 낮으므로 비교군이 적었으며, 집단의 유사도 평균이 -0.03으로 적은 비교군 내에서 일반적이지 않은 위치에서 죽은 것으로 추정된다.

종합 유사도가 높은 6번 사람과 1,180,000Step 학습한 인공지능의 데이터를 게임지도 시각화 분석하여, 확인결과 게임플레이가 유사하다고 할 수 있었다. 프레임별 위치, 액션 위치, 죽은 위치를 게임지도 시각화하여 확인했다.



[Fig. 5] Data analysis heat maps of Human and AI



[Fig. 6] Data analysis heat maps of 6th human and AI learned 1,180,000 steps

5.4 사람과 사람 분석

14명 사람의 10개의 게임을 자신의 게임을 제외한 130회의 게임을 비교했다. 하나의 사람과 인공지능이 유사하다고 모든 사람과 인공지능이 유사하다고 볼 수 없다. 따라서 사람과 사람과의 유사도 분석을 통해 각 사람과의 유사도를 계산했다.

히트맵 분석으로 2번 사람과 6번 사람의 게임플레이가 유사하다고 할 수 있었다. 히트맵으로 유사도가 높은 사람들을 확인했다. 종합 유사도가 높은 2번 사람과 6번 사람의 게임플레이의 히트맵의 네 가지 분석 모두 높은 유사도를 보였다. 데이터 확인결과 프레임별 위치 0.867, 액션 위치 0.879, 죽음 위치 0.999, 종합 0.915로 높은 유사도를 확인했다.

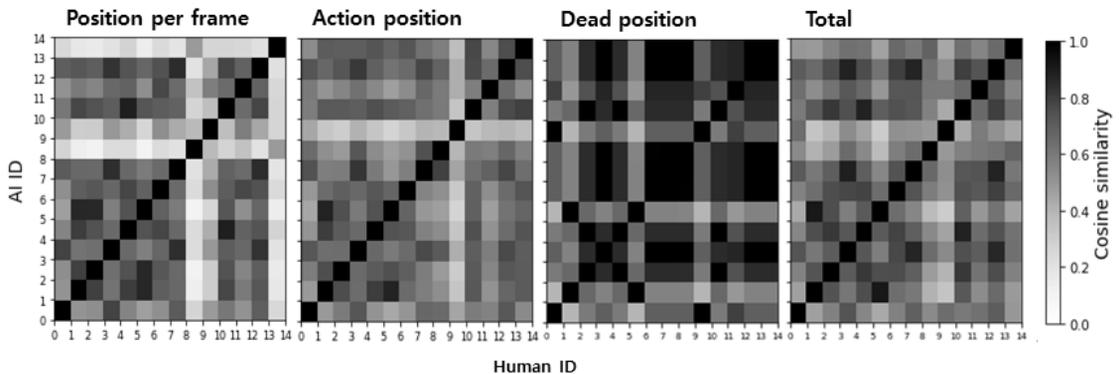
인공지능과 비교에서 일반적이지 않은 분포의

사람 9번, 10번, 14번 사람은 다른 사람과의 비교에도 같은 분포를 보여 이들이 평균적인 사람들과 다른 플레이를 한다는 것을 알 수 있었다.

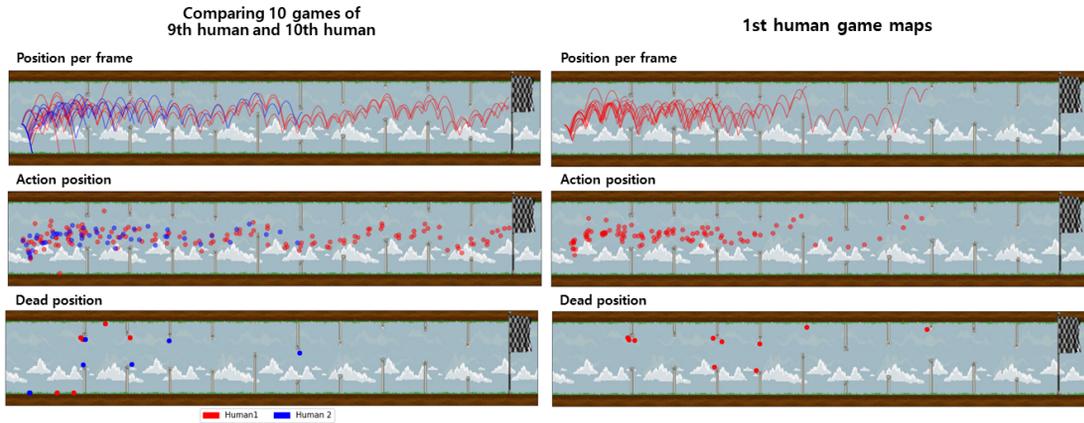
종합 유사도가 높은 2번 사람과 6번 사람의 데이터를 게임지도 시각화 분석하여, 확인결과 게임플레이가 유사하다고 할 수 있었다. 프레임별 위치, 액션 위치, 죽은 위치를 게임지도 시각화하여 확인했다.

5.5 사람 집단과 인공지능분석

사람과 사람의 유사도와 사람과 인공지능 유사도의 연관성을 확인하기 위해 유사도가 높은 사람끼리 집단을 만들었다. 사람끼리 유사도가 0.87 이상이고 다른 집단과 유사성이 적은 3개의 집단을 만들었다.



[Fig. 7] Data analysis heat maps of Human and Human



[Fig. 8] Data analysis game maps of Human and Human

[Table 3] Create human group

Group ID	Human1 ID	Human2 ID	Cosine similarity
1	2	6	0.914
2	4	8	0.881
2	8	13	0.874
2	4	13	0.871
3	5	11	0.870

사람집단과 유사한 인공지능 학습량을 확인했다. 사람집단에 속한 사람들의 인공지능의 플레이어별 종합 유사도의 평균을 계산하여, 사람집단과 유사한 인공지능 학습량을 확인했다.

[Table 4] Human group and AI player total data analysis.

Group ID	AI ID	Cosine similarity
1	110	0.922249
1	118	0.912534
1	24	0.903571
1	36	0.898453
1	116	0.897372
...		
2	112	0.325195
2	114	0.318173
3	2	0.131898
2	2	0.118941
1	2	-0.01384

종합 유사도가 높은 1번 집단과 1,100,000Step 학습한 인공지능의 데이터를 히트맵과 게임지도 시각화 분석하여, 확인결과 게임플레이가 유사하다고 할 수 있었다. 유사도를 중심으로 사람의 표시 순서를 바꿔서 히트맵을 재확인했다. 유사성이 강한 3개의 집단은 인공지능과의 히트맵도 분포가 유사했다. 이후 게임지도 방식으로 시각화했다. 프레임별 위치, 액션 위치, 죽은 위치 세 가지의 시각화하여, 분석결과 게임플레이가 유사하다고 할 수 있었다.

6. 결 론

본 논문은 사람과 유사한 게임플레이를 하는 인공지능 생성을 위해 제약사항을 추가해 강화학습 인공지능을 생성하고, 이를 코사인 유사도로 사람과 게임플레이를 비교 분석하여 사람과 유사한 게임플레이를 하는 인공지능 에이전트를 찾아냈다. 먼지 플리피버드 게임에 제약사항인 카메라 시야 제한, 액션 간 지연, 아무것도 하지 않는 NULL 액션을 추가해 강화학습 인공지능을 생성했다. 온라인을 통해 14명의 사람의 플레이데이터와 학습량 별로 생성한 60개의 강화학습 인공지능 모델로 게임플레이 데이터를 수집했다. 이를 코사인 유사도로 프레임별 위치, 액션 위치, 죽은 위치, 종합 네 가지 방법으로 측정했다. 그리고 유사도를 히트맵

과 게임지도 시각화를 통해 측정이 맞는지 확인했다. 또한, 정확한 분석을 위해서 사람과 인공지능 뿐만 아니라 사람과 사람, 사람과 사람 분석에서 유사한 집단을 사람집단을 찾아 사람집단과 인공지능을 분석하였다. 분석결과 사람과의 유사도가 0.9 이상인 인공지능 에이전트를 찾을 수 있었다.

사람과 인공지능의 게임플레이 비교평가 방법으로 코사인 유사도 방법을 사용하여, 튜링테스트보다 나은 결과를 보여줬다. 적은 수의 인원으로 사람과 인공지능이 유사한 정도를 확인할 수 있었으며, 다각도 분석을 통해 실제 사람의 플레이와 어떠한 플레이가 유사한지 분석할 수 있었다.

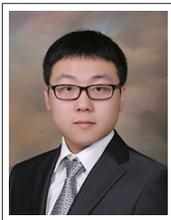
이러한 사람과 유사한 강화학습 인공지능 연구로 게임에서 밸런스 테스트 자동화를 통해 저비용으로 데이터 수집과 게임의 품질상승이 가능하다. 그러나 플리피버드 게임은 단순한 게임으로 복잡한 게임과 다른 장르의 게임에서도 유사도 측정방법을 적용할 수 있는지 추가적인 연구가 필요하다.

REFERENCES

- [1] Hyung-ryul Son, "A game balance story that makes a fun game", pp31-45, 2014
- [2] Luis Levy and Jeannie Novak, "Game development essentials: Game QA & testing. Cengage Learning", pp58-59, 2009.
- [3] A. Drachen and A. Canossa. "Towards game play analysis via gameplay metrics", Proceedings of the 13th international MindTrek conference: Everyday life in the ubiquitous era, pp202 - 209, 2009.
- [4] A.DRACHEN et al. "Game analytics: Maximizing the value of player data", Springer, London, pp205 - 253, 2013.
- [5] Huizinga et al., "Automated defect prevention: best practices in software management". Wiley-IEEE Computer Society Press, pp74, 2007.
- [6] Unreal Automation System Overview, <https://docs.unrealengine.com/ko/Programming/Automation/index.html>.
- [7] Suk-ki Lee and Ho-Young Kwak., "Game Test Automation with Reinforce Learning", Proceedings of the Korean Society of Computer Information Conference, Vol 26, No.1, pp25-28, 2018.
- [8] Oriol Vinyals et al., "Grandmaster level in StarCraft II using multi-agent reinforcement learning". Nature. Vol 575. pp350-572, 2019.
- [9] Julian Togelius and G. N. Yannakakis., "General general game AI", IEEE Conference on Computational Intelligence and Games (CIG), pp20-23, 2016.
- [10] Ahmed Khalifa et al., "Modifying MCTS for Human-like General Video Game Playing". IJCAI International Joint Conference on Artificial Intelligence. pp2514-2520, 2016
- [11] Y. Burda et al., "Exploration by Random Network Distillation,", in Int. Conf. Learning Representations, New Orleans, LA, USA, pp1-17, 2019.
- [12] BERNER, Christopher, et al. "Dota 2 with large scale deep reinforcement learning". arXiv preprint Vol 1912 No.06680, pp1-66, 2019.
- [13] P. Hingston, "A new design for a turing test for bots", Computational Intelligence and Games (CIG) 2010 IEEE Symposium on, pp345-350, 2010.
- [14] N. Shaker, J. Togelius, G. N. Yannakakis, L. Poovanna, V. S. Ethiraj, S. J. Johansson, et al., "The turing test track of the 2012 mario ai championship: entries and evaluation", Computational Intelligence in Games (CIG) 2013 IEEE Conference on, pp1-8, 2013.
- [15] Alan Turing, "Turing 1950", pp433, 1950.
- [16] Jiyoung Woo and Huy Kang Kim, "Trends of research on fraud detection in online games", Korea Institute Of Information Security And Cryptology, Vol 27, No. 4, pp14-21, 2017.
- [17] Eun-Jo Lee et al, "A Study on Game Bot Detection Using Self-Similarity in MMORPGs", Journal of the Korea Institute of Information Security & Cryptology, Vol 26, No.1, pp93-107, 2016.
- [18] Chang-wook Jeon and Taekyun Choi and Jonghyun Cho, "Natural language processing starting with TensorFlow and machine learning", pp122-123, 2019.
- [19] David Silver et al., "Mastering the game of go without human knowledge", Vol 550, No.

7676, Nature, pp354-359, 2017

- [20] Unity ML-Agents Toolkit, <https://github.com/Unity-Technologies/ml-agents>, Unity-Technologies.
- [21] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. pp 1097-1105, 2012.
- [22] Unity ML-Agents, <https://github.com/Unity-Technologies/ml-agents/blob/master/docs/Learning-Environment-Design-Agents.md>, Unity-Technologies.



허민구 (Heo, Min Gu)

약력 : 2013 호서대학교 게임공학과 학사
2015-2018년 DevArc 기업 대표
2019-현재 호서대학교 게임학과 석사과정

관심분야 : 게임디자인, 빅데이터, 인공지능



박창훈 (Park, Chang Hoon)

약력 : 1995 단국대학교 전자계산학과 학사
1997 단국대학교 전자계산학과 석사
2003 고려대학교 컴퓨터과학과 박사
2006-현재 호서대학교 게임애니메이션융합학부 교수

관심분야 : 게임, 가상현실
