

Match-3 게임 플레이 테스트를 위한 오브젝트 특성 분류 기반 상태 표현 방법

노진하^{0,1}, 백인창¹, 오승원², 문재영¹, 오송미¹, 김경중^{*,1}

¹광주과학기술원, ²승실대학교

noah9905@gm.gist.ac.kr, inchang.baek@gm.gist.ac.kr, sw980907@naver.com,

super_moon@gm.gist.ac.kr, songmi.ohh@gmail.com, kjkim@gist.ac.kr

Game Object Characteristic-based State Representation Method for Playtesting Match-3 Game

Jin-Ha Noh^{0,1}, In-Chang Baek¹, Seung-Won Oh², Jae-Young Moon¹, Song-Mi Oh¹, Kyung-Joong Kim^{*,1}

¹Gwangju Institute of Science and Technology, ²Soongsil University

요약

게임 플레이 테스트는 개발사가 콘텐츠 배포 전 검증하는 주요한 절차이다. 게임 개발사에서는 매주 새롭게 개발하는 다수의 레벨을 검증하기 위해서 자동화된 방법을 사용하고 있다. 최근, 인간 플레이어를 대신하여 콘텐츠를 검증하기 위해서, 기계학습 방법을 이용하여 플레이 테스트를 위한 에이전트를 학습하려는 시도들이 늘어나고 있다. 게임 에이전트를 학습하기 위한 심층 강화학습 방법에서는, 이산적인 게임의 상태를 표현하기 위해 One-hot 형태로 인코딩하는 방법이 대표적이다. 하지만, 이 방법을 적용할 경우에는 게임에 새로운 오브젝트가 추가될 때마다 이를 표현하기 위해 차원을 추가하여 모델을 재학습해야 하는 불편함이 있다. 이러한 문제를 해결하기 위해서, 본 논문에서는 게임 오브젝트들을 개별적인 차원으로 표현하는 대신에, 게임 오브젝트를 특성 별로 카테고리화 하여 매핑한 새로운 형식의 상태 표현 방식을 제안한다. 이러한 상태 표현 방법을 상용 Match-3 퍼즐게임에 적용하였을 때, 새로운 콘텐츠에 대해 6.24%의 성능향상이 있음을 확인하였다.

1. 서론

게임 플레이 테스트는 개발사가 새로운 콘텐츠를 배포하기 전 검증하는 주요한 절차이다. 유저의 흥미를 유지하기 위해서는 게임 난이도 조절 및 버그 해결을 통하여 바람직한 업데이트를 진행하는 것이 필수적이다. 이러한 업데이트는 지속적인 플레이 테스트를 통하여 유지할 수 있다. 다만, 퍼즐 게임의 경우 매주 새롭게 개발되는 레벨을 사람이 직접 검증하기 위해서는 시간 및 자원적인 한계가 매우 크다. 이에 게임 개발사에서는 다수의 레벨을 검증하기 위하여 자동화된 방법을 찾고 있으며, 이러한 자동화 방식을 찾기 위한 연구는 Match-3 형태의 퍼즐 게임에서도 활발히 진행되어 왔다 [1].

현재 자동화된 게임 플레이 테스트 에이전트를 만들기 위한 많은 시도 중 하나로는 심층 강화학습을 이용한 방식이 있다. 심층 강화학습에는 플레이중인 게임 상태가 요구되기에, 이산적인 게임의 상태를 표현하기 위한 방식이 필요하다. Match-3 장르 게임의 경우 One-hot 형태로 인코딩하는 방식이 대표적이다 [1]. 하지만, 이러한 인코딩 기법은 새로운 오브젝트의 추가가 곧 차원의 추가로 이루어지기에 모델의 재학습이 필수적으로 요구된다. 뿐만 아니라 오브젝트의 개수에 비례하여 차원 크기가 커지기 때문에 새로운 오브젝트가 추가될 수록 학습이 힘들어지는 단점이 있다.

본 연구에서는 게임 오브젝트의 특성을 기반으로 학습하는 새로운

상태 표현 방식을 제안하고자 한다. 기존의 One-hot 인코딩 방식과는 달리 특성 별 표현 방식은 새로운 오브젝트의 등장에도 기존의 학습 파라미터를 유지하며 게임 플레이를 진행할 수 있다. 우리는 이 방식을 상용 Match-3 게임 환경에 적용한다. 먼저 게임에 등장하는 오브젝트들이 가질 수 있는 특성들을 전문가 지식을 이용해 정의한 후 각 블록을 특성들의 조합으로 상태를 표현하여 학습에 사용한다. 또한, 새로운 콘텐츠를 개발하는 상황을 가정하여 테스트 세트를 구성하였으며, 새로운 오브젝트가 포함된 레벨에 대한 모델의 성능을 기존의 방법과 비교한다.

2. 이론적 배경



그림 1. 상용 Match-3 게임의 플레이 화면 및 오브젝트 예시

¹ 본 연구는 문화체육관광부 및 한국콘텐츠진흥원의 2022년도 문화기술 연구개발 사업으로 수행되었음 (과제명 : 온라인 게임 콘텐츠 제작 지원을 위한 인공지능 기반 게임 시뮬레이션 기술 개발, 과제번호 : R2022020070) 이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(2021R1A4A1030075). *교신 저자

Match-3 게임은 인접한 두 블록의 스왑 (swap)을 통해 3개의 동일한 색상의 블록을 맞추어 블록을 파괴하는 게임이다. 블록이 파괴되어 생기는 빈공간은 무작위 색상의 블록으로 채워지는데, 이러한 이유로 게임에 랜덤성이 유발된다. 각 레벨에는 플레이어가 해결해야하는 미션과 제한된 이동 횟수가 설정되어 있다. 제한된 횟수안에 미션을 해결하지 못할 경우 레벨 클리어에 실패하기 때문에 게임 기믹 (Gimmick)을 이해하고 전략을 설계함으로써 이동 횟수를 줄여야 한다. 게임 회사들은 재미를 높이기 위해 여러가지 특성을 가지는 오브젝트들을 계속해서 추가하며, 이는 플레이어가 게임에 대한 흥미를 잃지 않게 하고 도전적인 플레이를 만들어낸다. 일반적으로, 적은 이동 횟수를 사용하여 게임을 클리어하는 플레이어를 높은 수준을 가졌다고 평가한다. 그렇기 때문에 플레이 테스트에는 퍼즐 게임 에이전트가 최소한의 이동 횟수로 클리어하는 것이 중요하다.

게임 에이전트를 학습하기 위한 상태정보는 9 × 9 보드판 형태로 표현되는데, 이산적인 게임 오브젝트들의 종류들을 표현하기 위해 주로 One-hot 인코딩 방식을 이용하여 이진 등의 형태로 표현한다. Match-3 게임에서도 Gudmundsson et al. [1]가 이 방법을 차용하였으며, Shin et al. [2]은 상태정보를 0 부터 1 사이로 정규화시켜 나타냈다. 게임이 가지고 있는 블록 종류를 모두 표현하기 위해서는 블록의 개수만큼의 차원이 추가되어야 하고, 만들어지는 차원의 개수는 (높이, 너비, 오브젝트의 개수) 이다.

3. 특성 기반 상태 표현 방법 (Characteristic feature)

본 논문에서는 게임 오브젝트의 특성에 따라 특징 맵을 생성하는 상태 표현방법 (Characteristic feature)을 제안한다. 이는 보드판에 등장하는 오브젝트들을 특성별로 매핑한 특징 맵을 학습에 사용하는 방식이다. 예를 들어, 그림 2의 경우 해당 칸에 기본 블록과 새장 오브젝트가 존재하기 때문에 플레이어가 직접 조작할 수 없다. 따라서 제어 불가능하다는 특성을 갖게 된다. 제안한 방식은 추가된 오브젝트가 기존에 정의된 특성으로 구성된다면 학습된 신경망에 별다른 변경없이 바로 적용할 수 있다는 장점이 있다. 이 방식이 장점을 가지는 이유는 일반적으로 게임이 가지고 있는 특징 때문인데, 새로운 오브젝트가 등장할 때, 기존의 오브젝트들이 가지고 있는 특성의 변형과 조합으로 이루어지는 경우가 많기 때문이다.

표 1. 게임에서 나타날 수 있는 특성의 종류.

Characteristic	Description
Color (C)	색깔 구분 여부
Special Block (SB)	특수 블록 존재 여부
Direction (D)	방향성 여부
Fixed (F)	플레이어에 의해 블록 제어가 불가능한가
Matchable (M)	해당 칸이 매치에 포함될 수 있는가
Inner Durability (ID)	해당 칸이 매치에 포함된다면 내구도가 존재하여 단번에 파괴되지 않는가
Outer Durability (OD)	해당 칸 인접위치가 매치에 포함된다면 내구도가 존재하여 단번에 파괴되지 않는가
Auto Move (AM)	매 스텝마다 해당 블록이 이동하는가

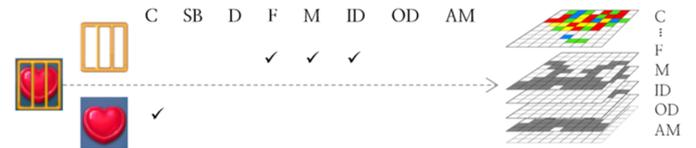


그림 2. Characteristic feature 개요도

표 2. 게임 오브젝트가 대응하는 특성들을 매칭한 표. 음영 처리된 오브젝트들은 모든 레벨에 필수적으로 사용됨.

Object	Characteristic							
	C	SB	D	F	M	ID	OD	AM
기본 블록	✓							
띠 블록	✓	✓	✓					
폭탄 블록	✓							
무지개 블록		✓						
교차 블록	✓	✓						
로켓 블록	✓	✓						
도넛							✓	
설탕					✓	✓		
보스				✓			✓	✓
쿠키					✓			
마시멜로								

중략 (와플, 새장, 젤리, 두더지, 식탁보)

오브젝트들의 특성을 범주화 하기 위해, 게임에 사용되는 오브젝트들이 가지는 특성들을 정의하였고, 표 1에 특성과 그에 따른 설명을 정리하였다. 또한, 표 2에서는 게임 오브젝트와 매칭되는 특성들을 표 1에서 정의한 특성을 이용해 매핑 해주었다.

4. 실험

4.1 강화 학습 모델

실험에서는 상태 (State) 표현 방법에 따른 성능 차이를 조사하기 위해, 이전 연구에서 제안한 One-hot feature (OH)와 본 연구에서 제안하는 Characteristic feature (CH)의 두 상태 표현 방식을 각 모델의 입력으로 설정하였다. 또한, 가능한 행동 중 임의의 행동을 취하는 랜덤 (RD) 에이전트를 추가한 총 3가지 모델에 대한 성능 평가를 진행하였다.

행동 (Action)은 9 × 9 크기 보드에서 각 칸마다 인접한 블록과 스왑 (swap) 가능하다. 각 행마다 가로방향으로 서로 인접한 블록 세트가 8개씩 있으므로 8 × 9 = 72개이고, 마찬가지로 세로방향은 9 × 8 = 72개이다. 따라서 총 144개의 행동 공간이 존재한다. 보상 (Reward)은 매 스텝마다 -1, 게임 클리어 시 +1을 받도록 설정하였다. 강화학습 알고리즘은 PPO를 사용하였으며, 사용한 학습 파라미터는 batch-size=64, gamma=0.9, total-timestep=500K이다 [4]. 게임의 랜덤성 때문에, 100판을 플레이 한 결과의 평균값을 사용하여 평가하였다. 성능 지표는 보드판을 클리어하는데 사용된 이동 횟수로 측정하였으며, 적은 이동횟수로 클리어 할수록 높은 성능을 가진다. 시뮬레이터는 상용 게임으로, 퍼즐윈스튜디오에서 개발한 Lollipop & Marshmallow Match3 게임을 Unity ML-Agents [3] 프레임워크를 통해 강화학습 환경으로 개발하여 사용하였다.

4.2 Train/Test Level

표 3. 레벨 별 포함된 게임 오브젝트를 나타낸 표. 음영 처리된 부분은 새로 추가된 오브젝트이다.

Object	Train					Test 1				Test 2			
	1	2	3	4	5	6	7	8	9	10	11	12	13
도넛	✓						✓						
설탕		✓				✓				✓			✓
보스		✓				✓						✓	
와플			✓									✓	
새장				✓			✓	✓			✓		✓
젤리					✓			✓		✓	✓		
두더지												✓	✓
식탁보												✓	
쿠키												✓	
마시멜로										✓			

상태 표현 방법에 따른 강화학습 모델의 일반화 성능을 측정하기 위해, 게임에 포함된 레벨 13종을 학습 및 평가 세트로 구성하였다. 표 3은 3개로 나누어진 레벨 세트와 각 레벨에 포함된 게임 오브젝트를 나타낸다. Train 레벨은 강화학습 모델을 학습하는 데에 사용되었고, 동일한 게임 오브젝트로 새로 디자인된 레벨을 Test 1으로 구성하였다. 또한, 4가지 새로운 게임 오브젝트가 개발된 것을 가정하여, 이를 포함하는 레벨을 Test 2로 구성하였다. Test 1과 2에서는 Train에서 학습된 모델에서 추가적인 학습없이 모델의 재사용성을 평가하였다.

5. 결과 및 논의

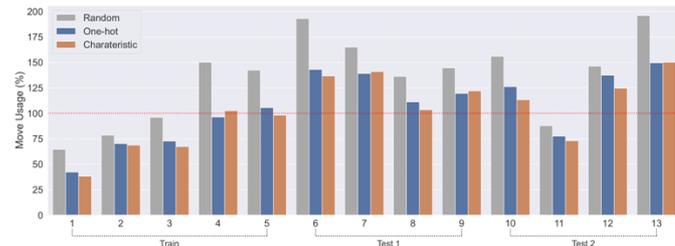


그림 3. 각 레벨 클리어에 허용된 최대 이동 횟수에 비례하여 에이전트들이 사용한 이동 횟수를 나타낸 그림.

표 4. 각 Test 레벨로 Characteristic feature 방식을 One-hot 인코딩 방식과 비교하여 이동 횟수의 개선율을 나타낸 표.

Feature Type	Test1				Test 2			
	6	7	8	9	10	11	12	13
RD	48.26	47.84	49.04	50.62	31.20	26.35	38.04	58.79
OH	35.76	40.34	40.03	41.83	25.23	23.27	35.72	44.88
CH	34.17	40.88	37.28	42.71	22.66	21.91	32.41	45.03
Improvement (1 - CH/OH) %	+4.45	-1.34	+6.87	-2.10	+10.19	+5.84	+9.27	-0.33
Mean	+1.97%				+6.24%			

그림 3에서는 에이전트들이 클리어하는데 사용한 이동 횟수를 레벨 별 허용되는 최대 이동 횟수에 비례하여 나타냈다. 표 4는 학습된 모델이 각 레벨 세트에서 플레이한 결과를 나타낸다. Improvement는 기존의 방법 (OH)과 비교하여 제안하는 방법 (CH)이 어느 정도의 성능 향상이 있는지 나타낸다. 동일한 게임 오브젝트를 가지고 있는 Train과 Test 1에서 각각 3.94%, 1.97%의 성능 향상을 보이며, 따라서 기존의 One-hot 인코딩 방식으로 학습한

모델과 비교하여도 우수한 성능을 보여준다. 또한, 새로운 오브젝트가 등장하는 Test 2에서는 6.24%의 성능 향상을 보였다.

같은 게임 오브젝트로 구성되어 있는 Train 과 Test 1에서 OH 보다 CH가 더 높은 성능은 보이는 이유는 게임 상태를 표현하는 차원이 달라진 것으로 설명할 수 있다. One-hot 인코딩을 수행하면 모든 오브젝트들을 각각 하나의 층으로 표현하게 되어 차원이 상대적으로 커지게 된다. 그렇기 때문에, 모델의 파라미터 수와 상태 복잡도가 적은 CH 방식이 더 좋은 성능을 보인다. 새로운 오브젝트가 추가된 Test 2에서 One-hot 인코딩으로 학습한 모델의 낮은 성능의 원인은 다음과 같다. One-hot 방식의 경우 새로운 오브젝트가 추가되면, 이를 위한 가중치들이 학습되어 있지 않기 때문에 모델이 해당 오브젝트를 고려한 플레이를 하기 어렵다. 반면에, 제안하는 방법으로 학습한 모델은 처음 보는 오브젝트라도 이미 학습된 특성들로 새로운 오브젝트를 표현할 수 있기 때문에 이를 고려한 의사결정이 가능하다.

6. 결 론

본 논문에서는 게임의 오브젝트를 특성별로 카테고리화 하여 매핑한 새로운 상태 표현 방식인 Characteristic feature을 제안한다. 이를 기존에 사용하던 One-hot 인코딩 방식과 학습한 모델과 비교하였으며, 게임 제작과정에서 새로운 컨텐츠가 추가됨을 고려한 평가를 진행하였다. 기존에 널리 쓰이던 One-hot 상태 표현 방식과 비교하였을 때 전반적인 성능 향상이 있었고, 특히 새로운 오브젝트가 추가된 레벨에 대해서는 재학습 없이도 크게 향상된 성능을 보였다.

하지만, 본 논문의 한계는 게임과 관련된 사전지식으로 특성을 이진으로 표현한 것이다. 향후 연구에서는, 제안한 방법을 개선하기 위해서 게임 오브젝트의 특성을 연속적인 값으로 표현할 수 있는 임베딩 방식을 개발할 계획이다. 이 임베딩 방식은 특성 값을 정의하기 위한 사전지식이 필요하지 않을 뿐만 아니라, 이진 표현보다 더 정확한 특성 표현이 가능하다. 결론적으로, 게임의 복잡한 상태 표현이 가능하고, 다른 게임으로도 적용 가능한 일반적인 특성을 학습할 수 있을 것으로 기대한다.

7. 참고 문헌

[1] Gudmundsson, Stefan Freyr, et al. "Human-like playtesting with deep learning." 2018 IEEE Conference on Computational Intelligence and Games (CIG). IEEE, 2018.

[2] Shin, Yuchul, et al. "Playtesting in match 3 game using strategic plays via reinforcement learning." IEEE Access 8 (2020): 51593-51600.

[3] Juliani, Arthur, et al. "Unity: A general platform for intelligent agents." arXiv preprint arXiv:1809.02627 (2018).

[4] Raffin, Antonin, et al. "Stable-baselines3: Reliable reinforcement learning implementations." Journal of Machine Learning Research (2021).