

하스스톤 플레이 인공지능

황금환
광주과학기술원
ghkdrmaghks@gm.gist.ac.kr

Playing AI in hearthstone

Geumhwan Hwang
Gwangju Institute of Science and Technology

요 약

본 보고서에서는 고정된 덱을 이용하여 하스스톤의 클래식 환경에서 높은 성능을 보이는 인공지능의 개발을 목적으로 했다. 덱은 클래식 환경에서 가장 강한 덱인 콤보 드루이드 덱을 사용하였다. 멀리건과 킬 캐치 능력은 따로 Rule-Base로 진행되도록 만들었고, 다른 플레이들에 대해 랜덤으로 플레이를 한 후 그 플레이에 대해 점수를 부여한다. 점수는 자신과 상대의 체력과 필드 상황, 마나와 드로우 같은 추가적인 요소들에 의해 결정되어 그중 가장 높은 점수를 받은 플레이를 채택한다. 그 결과, 인공지능의 플레이 횟수가 높아질수록 높은 성능을 보였다.

1. 서론

하스스톤은 가장 유명한 카드 게임 중 하나로, 1억 이상의 플레이어와 1조 원 이상의 매출을 낸 전설적인 게임이다. 하스스톤의 AI 연구는, 상대방이 덱 빌딩, 또는 덱 빌딩을 포함하는 방향으로 발전해 왔다. 카드 게임에서 덱 빌딩이란, 카드의 쓰임새를 정확하게 이해하고 이를 바탕으로 되어야 한다 생각하여, 덱 빌딩 AI를 만들기 위해서는 우선 이해도 높은 플레이를 하는 AI가 필요하다. 그렇기에, 메타가 가장 고착화되어 좋은 덱을 쉽게 알 수 있는 클래식 모드에서 게임을 플레이하는 AI를 개발했다.

2. 관련 연구

많은 하스스톤 AI 들은 덱 빌딩 과정이 포함된다. 플레이 중심의 AI는 Sabberstone 환경과 이 환경에서의 대회가 가장 활발하게 일어났다. 2020년까지 진행된 대회에서는 고정된 덱을 활용해 플레이를 진행하는 부문이 있었고, 이 대회에서 AI가 많이 개발되었다[2][3]. 다만, 대회가 중단된 2021년 이후로는 플레이 AI 개발이 상당히 느려졌다.

최근의 연구 중 가장 뛰어난 성능을 보인 것이 있다 [1]. 여러 제약사항이 있었지만, 프로그래머 수준의 플레이어와도 밀리지 않는 모습을 보였다. 하지만 역시 플레이 중심의 AI가 아닌 덱 빌딩을 겸하고 있으며, 여러 제약사항이 있었다.

3. 환경 설정

게임 모드는 클래식 모드를 활용한다. 클래식 모드는 오리진얼 환경(2014년 6월)의 카드들만으로 플레이하는 모드다. 카드 풀이 가장 적고, 가장 고착화된 메타를 가지고 있고, 발견과 같은 변수 키워드가 없어 AI를 만들기 적절하다.

덱은 당시 메타 통계[4] 기준 가장 강한 덱 중 하나인 콤보 드루이드를 활용했다. 이 덱은 많은 마나와 이를 통한 강력한 필드, 일명 “자군야포” 콤보를 이용한 강력한 피니 시가 장점인 덱이다. 당시 덱의 파워 기준 가장 강한 덱은 미라클 도적, 컨트롤 전사 등이 있는데 이들은 각각 많은 카드를 활용한 콤보와 필드 컨트롤의 측면에서 AI가 플레이하기 어려운 부분이 있다. 콤보 드루이드 역시 쉬운 덱은 아니지만, 가장 강한 덱 중 하나이고 승리 방법이 하스스톤에서 가장 표준적인 방법이라 선택했다.



Fig 1. 사용한 덱 리스트

개발 환경은 Rosettastone 환경을 사용했다[5] Rosettastone은 하스스톤 강화학습을 하게 해주는 환경으로, 대부분의 카드, 특히 모든 클래식 모드의 카드가 구현되어 있으므로 AI 개발에 적합하다.

4. 알고리즘

ㄱ) 멀리건

멀리건이란 게임을 시작한 직후에, 패를 받고 나서 한번만 필요 없는 카드들을 텍에 넣고 그만큼 다시 드로우할 수 있는 기회를 주는 것이다. 이 게임은 기본적으로 상대방과 상호작용이 필요하지만, 멀리건을 할 때는 유일하게 상대방이 어떤 것을 할지에 상관없이 플레이를 할 수 있다. 또한, 멀리건을 통해 뽑은 카드가 어떤 승률의 영향을 미칠지는 바로 알 수 없으므로, 통계를 활용하여 멀리건 할 카드 조합을 Rule-Base로 정했다[6]. 특히, 특정 카드들의 조합에 따라 승률이 갈리는 경우도 많으므로 최대한 많은 경우의 수에 대해 고려를 할수록 성능이 좋아진다.

ㄴ) 킬 캐치

일반적으로, 영웅의 체력의 가치는 그렇게 높지 않다. 하지만, 이 게임은 결국 상대의 체력을 0으로 만들어야 한다. 그렇기에 상대를 죽일 수 있는지 확인하는 과정은 따로 프로세스를 만들었으며, 이 과정에서는 상대의 체력을 0으로 만들 수 있는지 계속해서 확인해본다.



Fig 2. 킬 캐치의 예시. 상대를 죽일 수 없다면, 이러한 플레이는 대체로 비효율적이다.

이 이후 단계 ㄷ~ㅊ의 과정은, 각 턴당 일정 횟수(=Depth)만큼 반복한다. 시도마다 다른 플레이를 하되, 드로우와 같은 플레이에서의 랜덤 요소는 고정한다, 각 플레이 결과의 점수를 점수 계산 단계에서 계산해, 가장 점수가 높은 플레이를 채택한다.

ㄷ) 필드 세팅

패에서 랜덤하게 한 카드를 선택하여, 그 카드를 플레이할 수 있으면 플레이하고, 없으면 필드 세팅 과정을 끝낸다. 이때, 점수에 영향을 줄 수 있는 플레이들은 미리 고려한다. 예를 들어, “급속 성장” 카드의 빈 마나 수정을 획득하는 효과는 필드나 핸드 외적으로 이득을 주는 카드이므로, 이 카드를 쓴 경우 미리 점수에 이득의 평가치만큼을 더해주는 방식이다. 대상을 설정하는 카드들을 플레이할 때는 랜덤으로 설정한다.

ㄹ) 공격

공격은 공격할 수 있는 캐릭터 중 랜덤하게 진행된다. 그 대상은, 우선순위에 따라 공격할 대상을 설정한다. 먼저, 도발 캐릭터가 있으면 도발 캐릭터를 우선 공격한다. 다음, 이득 교환 (내 몬스터를 살리면서 상대 몬스터를 죽일 수 있는 경우) 이 가능하면 교환한다. 아니라면, 랜덤한 대상을 공격한다.

ㅊ) 점수 계산

우선, 체력에 대한 점수를 더한다. 자신의 체력이 높고, 상대의 체력이 낮을수록 높은 점수를 얻는다(HP 변수). 체력 점수에 대한 비중은 그다지 높지 않지만, 상대의 체력이 낮아져 이로 인한 플레이를 강요하는 경우 추가 점수를 준다. 이 체력은 자신의 “자군야포” 콤보 적을 죽일 수 있는지도(Safe 변수). 똑같이 자신의 체력에 대해서도 지나치게 낮은 경우 점수를 잃는다.

$$HealthScore = 2MyHP - 2OpHP + 50MySafe - 50OpSafe$$

다음, 필드에 대한 점수를 더한다. 기본적으로 전체 공격력의 합+전체 체력의 합이며, 이로온 효과(죽음의 메아리) 등에 대해서는 추가로 보정해준다.

$$FieldScore = \sum_{i \in Myfield} (10ATK_i + 10Health_i + Additional_i) - \sum_{i \in Opfield} (10ATK_i + 10Health_i + Additional_i)$$

마지막, 필드 외적인 점수를 더한다. 카드의 드로우, 마나 수정의 획득 등 필드 외적으로 영향을 주는 효과들이다. 정신 자극과 같이 추가적인 이득을 보는 것을 전제로 하는 경우, 카드의 사용 여부에 따라 요구되는 점수만큼 빼준다.

이 과정들이 끝나고, 점수가 지금까지 나온 점수보다 높다면 결과를 갱신해준다. Depth만큼 반복한 뒤, 최종적으로 가장 높은 점수를 얻은 결과를 플레이한다.

$$Score = HealthScore + FieldScore + AdditionalScore$$

5. 결과

	1	10	100	1,000	3,000	10,000
1	500					
10	993	500				
100	998	783	500			
1,000	1000	823	575	500		
3,000	998	825	560	516	500	
10,000	998	815	560	509	501	500

Table 1. 자신/상대의 Depth와 그에 따른 승리 횟수. 각 1,000회씩 진행하였다.

표는 자신과 상대의 Depth에 따라 1,000회씩 게임을 플레이할 때 승리 횟수에 대한 표이다. Depth가 높은 AI가 낮은 AI와 플레이할 때 승률이 높아지는 것을 볼 수 있다. 특히, Random에 가까운 Depth 1의 경우 거의 완벽하게 승리하는 모습을 보여준다.

또한, Depth가 높아질수록 그 승률 증가량은 적어지는 것을 볼 수 있다. 이에 따라 성능과 계산 시간의 균형을 잘 맞춰야 한다. Depth가 10,00을 넘어갈수록, 그 성능은 많이 증가하지 않고, Depth 3,000과 10,000은 거의 동일한 성능을 보였다.

시간이 늘어날수록 계산에는 오랜 시간이 걸리기 때문에, 그 균형을 잘 맞춰야 한다. 플레이하는 시간은 Depth에 대체로 비례했고, 플레이가 10,00회가 될 때의 플레이 시간은 한 턴당 평균 3초 정도 걸렸다. 이 이상이 된다면 성능의 증가에 비해 플레이어의 시간이 비효율적으로 증가할 것이므로, Depth가 10,000회 정도일 때가 가장 효율적인 성능을 보인다고 할 수 있다.

<https://github.com/GeumHwanHwang/HSClassicPlay> 에서 코드를 다운받을 수 있으며, RosettaStone[5]의 다운로드 및 설치가 선행되어야 한다.



Fig 3. 플레이 화면 예시.

6. Future Work

알고리즘의 개선이 가능할 것이다. 현재는 완전한 랜덤 플레이를 기반으로 하는데, 이 부분에서의 개선이 가능할 것으로 기대된다. 예를 들어, LLM 등의 활용이 가능할 것이다.

더 다양한 텍으로의 확장이 가능할 것이다. 현재는

콤보 드루이드만 이용했지만, 나아가 위니 흑마, 미드레인지 주술사 등의 인기 있고 쉬운 텍은 물론, 미라클도적, 컨트롤 전사와 같은 AI가 숙달하기 어려운 텍들까지 연구해볼 수 있을 것이다. 더 나아가 어떤 텍을 주더라도 그 텍에 맞는 전략을 알아서 실행하는 AI까지 발전할 수 있을 것이다.

나아가, 클래식 모드 뿐만 아니라, 정규전, 야생전 등 다양한 카드 풀에서의 AI도 개발할 수 있다. 이 모드들에서는 단순 카드의 수가 많아지는 것뿐만 아니라, 발견과 같은 새로운 키워드의 추가, 그리고 이런 키워드를 어떻게 활용하고, 어떻게 사용할지 등 더욱 많은 요소를 고려해야 한다.

이러한 완벽한 게임 플레이 AI가 완성된다면, 이를 통해 텍 빌딩 AI, 카드 생성 AI 등을 만든다면 이전의 결과들보다 더욱 개선된 결과를 기대할 수 있다.

7. 결론

본 연구를 통해, 하스스톤 클래식 환경에서의 플레이 중심의 인공지능을 만들 수 있었다. 콤보 드루이드 텍을 이용했으며, 상대의 플레이에 상관없이 진행되어야 할 멀리건과 킬 캐치 부분은 따로 Rule-Base로 진행되었으며, 나머지 플레이어는 랜덤하게 플레이한 점수를 매겨 가장 높은 점수를 받은 플레이를 채택했다.

결과적으로 반복하는 횟수가 많아질수록 성능이 좋아지는 AI를 제작했다.

8. 출처

- [1] Changan Xiao, Yongxin Zhang, Xuefeng Huang, Qinhan Huang, Jie Chen, Peng Sun (2023): Mastering Strategy Card Game (Hearthstone) with Improved Techniques
- [2] Świechowski M, Tajmayer T, Janusz A (2018) Improving Hearthstone AI by combining MCTS and supervised learning algorithms.
- [3] Dockhorn A, Frick M, Akkaya Ü, Kruse R (2018) Predicting opponent moves for improving Hearthstone AI. In: Proceedings of the 18th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems. Springer, pp 621-632
- [4] "[Classic] VS Data Reaper Report #2", Vicious Syndicate, June 6, 2021. URL: <https://www.vicioussyndicate.com/classic-vs-data-reaper-report-2/>
- [5] "RosettaStone", utilForever, URL: <https://github.com/utilForever/RosettaStone>
- [6] "HSReplay", URL: hsreplay.net
- [7] Santos A, Santos PA, Melo FS (2017) Monte carlo tree search experiments in Hearthstone. In: Proceedings of the 2017 IEEE Conference on Computational Intelligence and Games (CIG). IEEE, pp 272-279
- [8] Eiji Sakurai, Koji Hasebe (2023) Decision-Making in Hearthstone Based on Evolutionary Algorithm. In: ICAART