

Analysis and simulator implementation of Mighty, an advanced imperfect information game

Jeongwon Lee*, Kwihoon Kim**, Seung-Hyun Kim***

*COO, Pebblous Inc., Daejeon, Korea

**Principal Researcher, Electronics and Telecommunication Research Institute, Daejeon, Korea

***Professor, Dept. of AI Convergence Education, Korea National University of Education, Chung-buk, Korea

***Professor, Dept. of Computer Education, Korea National University of Education, Chung-buk, Korea

[Abstract]

Mighty is an imperfect information game, similar to the internationally popular four-player card game Bridge, but more complex in terms of game rules and operation. An environment for exploring and analyzing the strategy of the Mighty Game is required, but compared to the development of many simulators for strategy analysis of other card games such as Bridge, there is no analysis tool for the Mighty Game. Even the definition and understanding of the Mighty game at the academic level is lacking. To solve these problems, this paper systematically defined the procedures and rules of the Mighty Game. And based on this definition, we implemented a simulator that can learn Mighty game and analyze various strategies. For the usability and accessibility of the service, the simulator was developed with JavaScript, and various analysis functions are provided in the web environment. Lastly, comparative analysis with other trick-taking games dealt with in the related research domain showed that the Mighty game has its value as an incomplete information game and that there are many characteristics that make it easy to apply AI-based learning methods.

▶ **Key words:** Mighty, Bridge, Game simulator, Trick-taking game, Imperfect information game

[요 약]

마이티는 불완전 정보 게임으로, 국제적으로 대중적인 4인용 카드 게임 브리지와 유사하지만 게임 규칙 및 운영 측면에서 더욱 복잡한 특성을 가지고 있다. 마이티 게임의 전략을 탐구하고 분석하기 위한 환경이 필요하지만, 브리지 등 타 카드 게임의 전략 분석을 위한 시뮬레이터가 다수 개발된 것에 비해 마이티 게임의 분석 도구는 존재하지 않는다. 심지어 마이티 게임에 대한 학문 차원에서의 정의 및 이해가 부족한 상황이다. 이러한 문제를 해결하기 위해, 본 논문은 마이티 게임의 절차 및 규칙을 체계적으로 정의하였다. 그리고 이를 기반으로 마이티 게임을 학습하고 전략을 분석할 수 있는 시뮬레이터를 구현하였다. 시뮬레이터는 서비스의 활용성과 접근성을 고려하여 자바스크립트로 개발되었으며, 다양한 분석 기능을 PC/모바일 웹 환경에서 제공한다. 마지막으로, 관련 분야에서 연구 주제로 다루고 있는 다른 트릭테이킹 게임과의 비교 분석을 통해, 마이티 게임이 불완전 정보 게임으로서 연구 가치가 있으며 AI학습이 용이한 게임 특성이 존재함을 보였다.

▶ **주제어:** 마이티, 브리지, 게임 시뮬레이터, 트릭테이킹 게임, 불완전 정보 게임

-
- First Author: Jeongwon Lee, Corresponding Author: Seung-Hyun Kim
 - *Jeongwon Lee (jeongwon@etri.re.kr), Pebblous Inc., Electronics and Telecommunication Research Institute
 - **Kwihoon Kim (kimkh@knu.ac.kr), Dept. of AI Convergence Education, Korea National University of Education
 - ***Seung-Hyun Kim (kimsh@knu.ac.kr), Dept. of Computer Education, Korea National University of Education
 - Received: 2021. 07. 19, Revised: 2022. 01. 13, Accepted: 2022. 01. 13.

I. Introduction

마이트는 플레이 카드(playing card)로 하는 트릭테이킹(trick-taking) 방식의 불완전 정보 게임이다. 게임은 완전 정보 게임과 불완전 정보 게임으로 나눌 수 있는데, 완전 정보 게임은 바둑, 체스와 같이 게임이 진행되는 과정의 모든 정보가 공개된다. 완전 정보 게임의 전략 연구는 알파고[1]를 통해 정점에 이르렀다. 반면에 죄수의 딜레마, 포커[2], 스타크래프트[3]와 같이 일부 정보가 가려져 있는 게임을 불완전 정보 게임이라 한다. 대부분의 카드 게임은 상대방이 손에 든 카드를 볼 수 없으므로 불완전 정보 게임에 해당한다. 불완전 정보 게임의 어려움은 내가 선택한 전략의 결과가 확률적으로 주어진다라는 것이다. 상대방이 손에 든 카드나 딜러가 가진 덱의 카드를 모르기 때문에, 전략에 대한 결과를 확률적으로 기대할 수밖에 없다. 다양한 경우의 수를 확률적으로 추정할 수밖에 없으므로 전략 연구와 평가가 어렵다.

트릭테이킹은 카드 게임에서 주로 쓰이는 게임 방식인데, 트릭(trick)을 가져오는(taking) 게임이라 풀이할 수 있다. 여기서 트릭은 모든 플레이어가 돌아가며 카드를 테이블에 내려놓는 한 번의 턴 혹은 라운드를 의미하고, 한 번의 트릭에서 가장 높은 카드를 낸 플레이어가 테이블에 나온 모든 점수카드를 가져가는 방식이 반복되는 것이 트릭테이킹 게임이다. 마이트는 브리지와 게임 방식이 유사한데, 브리지가 4인용 게임이고 팀이 고정적인 반면에 마이트는 5인용 게임이고 팀이 매번 바뀌며 팀원의 역할이 게임 진행 도중에 드러난다는 점에서 전략적인 난이도가 높다고 평가할 수 있다.

관련 분야에서 브리지 게임을 대상으로 한 다수의 연구들이 존재하는데 비해, 마이트 게임을 다루는 연구는 극히 일부에 불과하다. 이는 마이트 게임이 한국을 중심으로 일부 매니아 층에서 전승되는 특성으로 판단된다. 온라인용으로 구현된 마이트 게임이 일부 발견되지만, 마이트 게임 자체에 대해 전략을 탐구하고 분석하기 위한 환경은 존재하지 않는다. 심지어 마이트 게임에 대한 학문 차원에서의 정의 및 이해가 부족한 상황이다.

본 논문은 (1)국제적으로 통용되는 트릭테이킹 카드 게임 규칙과 용어를 활용하여 마이트 게임의 절차 및 규칙을 체계적으로 정의하였다. 그리고 이를 기반으로 (2)마이트 게임을 학습하고 전략을 분석할 수 있는 시뮬레이터를 구현하였다. 시뮬레이터를 통해 다양한 카드 조합을 임의로 혹은 무작위로 만들어 전략을 검토하거나 숨겨진 정보들을 보면서 분석하고, 모범 사례의 저장 및 재현이 가능하

다. 또한 (3)관련 분야에서 연구 주제로 다루고 있는 다른 트릭테이킹 게임과 비교 분석하여, 마이트 게임이 불완전 정보 게임으로서 연구 가치가 있으며 AI 학습이 용이한 게임 특성이 존재함을 보였다.

II. Related Works

1. Card game simulation

게임 분야에서 시뮬레이션은 여러 가지 의미로 사용될 수 있다. 첫 번째로, 가장 흔히 쓰이는 개념은 상황 재현을 체험하기 위한 시뮬레이션 게임이다. 실제 상황을 비디오 게임으로 구현하여 화면으로 즐기도록 한 것이다. 예를 들어, 윈도우 운영체제에서 기본으로 제공했던 카드 게임들은 모두 첫 번째 분류에 해당하는 시뮬레이션 게임이다. 두 번째 의미의 시뮬레이션은 인간 플레이어 대신 컴퓨터 알고리즘으로 자동화된 플레이를 수행하여 게임에서 발생 가능한 경우의 수를 탐색하는 것이다. 이 방식은 자동화 플레이 시뮬레이션이라고 할 수 있다. 세 번째는 컴퓨터 환경에서 사용자가 직접 게임 플레이 시뮬레이션을 진행하는 것으로, 위 두 가지 시뮬레이션 범주를 통합한 것이다. 세 번째 분류의 플레이 시뮬레이션은 게임의 밸런스 와 통계적 분석을 자동화하고, 게임 교육과 전략 연구에 활용이 용이하다. 본 논문에서 언급하는 시뮬레이터는 세 번째 분류의 시뮬레이터를 의미한다.

2. Bridge game simulator

마이트와 가장 유사하면서도 대중적인 카드 게임은 브리지(Bridge)이다. 브리지 게임에 대한 분석 및 최선의 전략을 연구하는 다양한 연구가 최근에도 활발히 진행되고 있다[4-7]. 특히 브리지 전략 분석을 위한 시뮬레이터는 몇 가지 사례[8]가 있는데, 대표적으로는 BridgeBuff[9]와 DealMaster[10]가 있으며, 그 외에도 Deal[11], Ted's Bridge World[12], Peter Cheung's Bridge Website[13] 등이 있다.

대표적인 브리지 시뮬레이터 BridgeBuff(Fig.1)는 1992년부터 개발되었으며, 3.4.2절에서 설명할 비딩 평가에 특화되어 있다. BridgeBuff는 특히 시뮬레이션 비딩이라는 개념을 제시하는데, 사용자가 자신에게 주어진 카드 조합을 보고 비딩을 하면 나머지 플레이어의 카드를 무작위로 할당하는 자동화 시뮬레이션을 통해 비딩의 적합성을 평가한다.

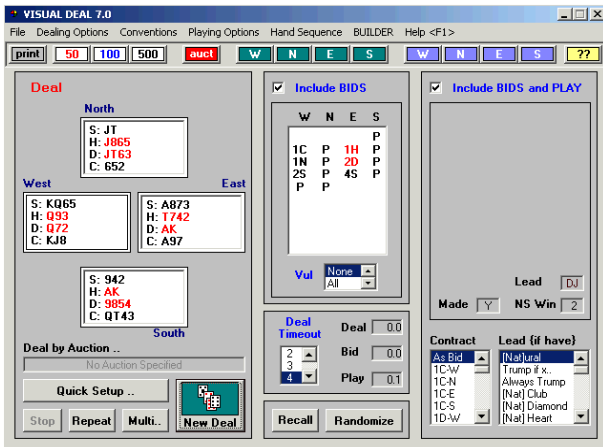


Fig. 1. BridgeBuff[9]

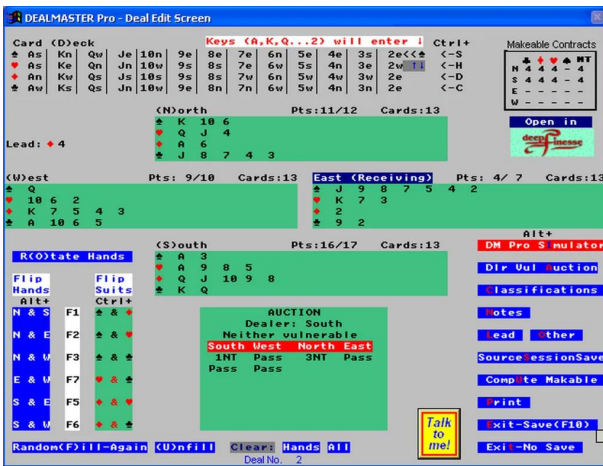


Fig. 2. DealMaster[10]

또 다른 브리지 시뮬레이터 DealMaster(Fig.2)는 1995년부터 개발되었으며, 마찬가지로 비딩 분석에 특화되어 있다. DealMaster는 카드 조합과 조건에 따라 적절한 비딩을 제안하는 식으로 분석 결과를 제공하는데, 사용자가 OR 또는 NOT 키워드를 결합하여 최대 20개의 카드 분배 조건을 제시하면 해당 내용을 대상으로 동시에 분석할 수 있다.

3. Mighty network game

마이티 게임을 위한 네트워크 프로그램 혹은 온라인 서비스는 몇 가지 사례가 있다. 첫 번째 사례는 1999년 서울대학교 컴퓨터공학과 장문성이 개발한 실행 프로그램 '마이티 네트워크 3.2'이다. 서울대학교 컴퓨터공학과 동아리 UPnL에서 이를 수정하고 개선하여 2010년에 '마이티 네트워크 4.0'을 배포하였다[14]. 이 방식은 2.1에서 정의한 첫 번째 의미의 시뮬레이션에 해당한다.

2006년 고려대학교 컴퓨터교육과 동아리 위드갯은 'mightyplayer'를 개발했으며[15], 마이티 전략을 로직으

로 구현하여 여러 에이전트를 만들고 서로 경쟁하도록 하여 가장 우수한 전략을 선발하기도 했다. 이 방식은 2.1에서 정의한 두 번째 의미의 시뮬레이션에 해당한다. 실행 파일 형태로 배포가 되었으나, 2010년 이후로는 업데이트가 이루어지지 않았다.

온라인 서비스도 몇 번의 시도가 있었으나 지금은 모두 서비스가 중단된 상황이다. 엠제이월드라는 온라인 마이티 게임 서비스 마이티클럽을 2002년부터 유료화했지만, 현재는 서비스가 중단되었다. 스타일릭스 등의 사이트에서도 마이티 게임이 가능했지만, 현재는 모두 서비스가 중단되었다. 2013년 안드로이드 앱으로 개발된 'MyMighty'가 0.42 버전까지 업데이트 되었지만, 2018년에 앱이 종료되었다. 2014년에 숙명여자대학교 학생들이 학과 프로젝트로 아이폰 앱을 개발한 적이 있으나 배포 단계에 이르지 못한 것으로 보인다. 가장 최근에는 2018년에 개발된 Python 코드가 있지만, 사이트는 접속이 되지 않는다. 이 상에서 소개한 마이티 네트워크 게임은 2.1에서 정의한 첫 번째 의미의 시뮬레이션에 해당한다. 오프라인으로 즐기던 게임을 5인의 플레이어가 네트워크로 접속해서 게임을 즐길 수 있다. 하지만 실전처럼 게임 하는 시뮬레이션만으로는 전략적인 난이도가 높은 마이티를 교육하거나 전략 분석에 활용하기는 어렵다.

III. Mighty Game Rule

마이티 게임의 규칙은 국가나 지역에 따라 다양한 버전이 있기 때문에, 일관되고 체계적인 연구를 수행하기 위해서는 마이티 게임의 세부 절차를 명확히 정의할 필요가 있다. 본 논문에서는 마이티 게임을 세부 단계로 분류하고, 각 단계에 적용되는 일반화된 규칙을 정의한다. 마이티 규칙의 한글 설명은 위키백과, 나무위키, 위키책의 마이티 항목을 참고했다. 영문 규칙은 Pagat[16] 사이트를 참고했고, 국제적으로 통용되는 트릭테이킹 카드 게임 규칙과 용어를 활용하기 위해 Jack Jack Knave[17], Denexa[18] 페이지를 참고했다.

1. Overview of Mighty game

마이티는 5인이 두 팀으로 나누어 승부를 겨루는 게임이며, 팀 멤버의 구성은 매번 바뀐다. 카드 10장을 받고, 게임에 어떤 역할로 참여할지 비딩을 통해 결정한다. 비딩에서 높은 공약을 제시한 플레이어가 주공이 되어 게임을 주도한다. 주공은 높은 계급의 특정 카드를 가진 플레이어

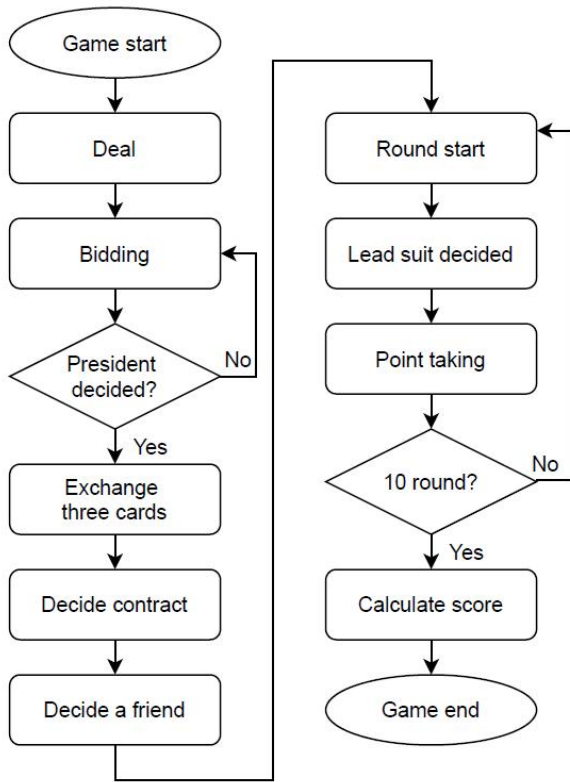


Fig. 3. Flow of the Mighty Game

한 명을 자신의 팀으로 지명할 수 있다. 이를 통해 주공과 프렌드가 한 팀, 나머지 3명이 한 팀이 되어 승부를 겨룬다. 한 게임은 10라운드로 이루어지는데, 각 라운드마다 플레이어는 손에 든 카드를 한 장씩 차례로 내려놓는다. 매 라운드 가장 높은 계급의 카드를 낸 플레이어가 해당 라운드의 점수카드를 모두 획득하며, 10라운드 종료 후 주공 팀의 점수카드를 합하여 주공이 공약으로 정한 수보다 같거나 많은 점수카드를 확보하면 주공 팀의 승리하고 그렇지 않으면 주공 팀이 패배한다. 게임 결과에 따라 주공은 프렌드에 비해 두 배의 보수를 얻거나 잃는다. 마이티는 제로섬 게임이므로, 승리한 팀이 획득한 보수의 합은 패배한 팀이 잃은 보수의 합과 같다. 게임 한 판의 전체적인 진행을 정리하면 다음 순서와 같고, Fig. 3은 전체 진행 흐름을 보인다.

1. 딜 (카드 배분)
2. 비딩 (주공 결정)
3. 주공 카드 3장 교환 후, 공약 확정
4. 프렌드 지명
5. 10회의 라운드 진행
6. 스코어 계산 및 보수 분배

2. Team

마이티는 5명의 플레이어가 두 팀으로 나누어 승부를 겨루는데, 팀이 고정되어 있지 않고 매번 달라진다. 게임 시작할 때 비딩을 통해서 주공으로 결정된 플레이어가 프렌드를 지명하여 팀을 이루게 되는데, 이 팀을 여당이라 부른다. 나머지 세 명은 자동으로 반대편이 되는데, 이 팀을 야당이라 부른다.

3. Cards

마이티는 플레이 카드 네 가지 슈트(♠,♦,♥,♣) 각 13장과 한 장의 조커, 총 53장의 카드로 한다. ♠A를 마이티라 부르며, 마이티는 언제 어떤 상황에서도 가장 높은 계급을 가진다. 그다음 높은 계급은 조커인데, 조커는 3.5.2 절에서 설명할 특정 상황에서 예외적으로 가장 낮은 계급을 가진다. 각 슈트에서는 A, K, Q, J, 10, ... , 2의 순서로 계급이 정해진다. 즉, 카드 계급 순위는 다음과 같다.

1. 마이티 카드
2. 무력화 되지 않은 조커
3. 트럼프 슈트 카드 (A, K, ... , 2의 차례로)
4. 리드 슈트 카드 (A, K, ... , 2의 차례로)
5. 그 외 카드와 무력화된 조커

한편, 각 슈트의 A, K, Q, J, 10 총 20장은 동일하게 점수카드로 취급된다.

4. Process

4.1 Deal

플레이어들에게 카드를 분배하는 절차를 딜(deal)이라 한다. 마이티는 5인용 게임이고 플레이 카드는 조커 포함 53장이 있으므로, 각 플레이어에게 10장의 카드를 분배하면 3장이 남는다. 남은 카드 3장은 보이지 않도록 테이블 위에 뒤집어 놓는다. 딜러는 직전 게임에서 프렌드였던 플레이어가 담당한다.

4.2 Bidding

매 게임 시작할 때 비딩(bidding)을 통해 주공을 정한다. 비딩은 딜이 완료된 후에 한다. 딜러부터 순서대로 주공이 되고자 하는 플레이어는 자신에게 유리한 트럼프 슈트와 획득 가능한 점수카드의 개수를 제시하여 비딩에 참여한다. 비딩에 참가하려면 다른 플레이어보다 높은 공약을 제시해야 하며, 가장 높은 공약을 제시한 플레이어가 주공이 된다. 비딩의 최소 공약은 점수카드 13장이 일반적

이며, 14장부터 시작하는 규칙도 가능하다. 트럼프 슈트를 지정하지 않는 노-트럼프(No-Trump) 비딩은 동일한 숫자의 트럼프 비딩보다 높은 것으로 보기도 한다[17].

4.3 Contract confirmation and friend nomination

주공은 뒤집어 놓은 3장의 카드를 가져와서 자신의 카드와 교환할 수 있다. 원하는 카드 조합 10장을 만들고 남은 3장은 다시 테이블에 뒤집어 놓는다.

3장 교환이 끝나면 주공이 비딩에 참여할 때와 손에 든 카드 조합이 달라졌기 때문에, 주공에게 트럼프 슈트를 변경할 수 있는 기회를 준다. 이때 트럼프 슈트를 변경하면 공약으로 제시한 점수카드의 숫자를 두 개 올려야 한다는 것이 일반적인 규칙이다. 예를 들어, 주공이 비딩에서 부른 공약이 (스페이드, 15) 였고, 카드 3장을 교환한 이후 트럼프 슈트를 다이아몬드로 바꾸고 싶다면 (다이아몬드, 17)로 변경해야 한다. 노-트럼프로 변경할 때에는 점수카드 숫자를 한 개만 올려도 된다는 규칙도 가능하다[17].

공약을 확정하고 나면, 주공은 같은 팀이 되는 프렌드를 지명할 수 있다. 특정 카드를 선택하여 공지하면 해당 카드를 가진 플레이어가 프렌드가 되는데, 이때 해당 카드를 손에 들고 있는 프렌드 자신을 제외하면 주공조차도 프렌드가 누군지 모르는 상태에서 게임을 시작하게 된다. 프렌드는 게임 진행 중 프렌드 카드를 테이블에 내려놓는 순간, 주공을 포함하여 모든 플레이어에게 자신의 정체가 명시적으로 밝혀진다. 프렌드는 주공에게 없는 카드 중 가장 등급이 높은 카드를 지명하는 것이 일반적이지만, 운영 전략과 함께 고려해야 한다.

주공은 프렌드 없이 단독으로 플레이할 수도 있다. 주공이 프렌드를 지정하지 않을 때에는, 전략적으로 자신이 손에 든 카드를 프렌드 카드로 부르는 방식을 많이 사용하는데, 물론 프렌드를 지정하지 않겠다는 사실을 명시적으로 선언(노 프렌드)할 수도 있다. 주공이 프렌드를 지정하지 않는 경우, 게임은 1명의 여당과 4명의 야당으로 팀이 구성된다. 주공은 프렌드 없이 공약에서 제시한 점수카드를 획득해야 게임에서 승리할 수 있으며, 프렌드가 있을 때와 비교하면 게임의 결과에 따라 두 배의 보수를 얻거나 잃게 된다.

4.4 Rounding

마이티의 라운드 진행은 트릭테이킹 방식을 따른다. 모든 트릭테이킹 게임이 공유하는 규칙은 한 트릭에 순서대로 한 장의 카드를 내고 가장 높은 등급의 카드를 낸 플레이어가 해당 트릭의 점수를 가져간다는 것이다. 휘스트, 브리지, 마이티와 같은 전통적인 트릭테이킹 게임은 트럼프

슈트와 리드 슈트를 가진다. 트럼프 슈트는 게임 한 판을 통틀어 가장 등급이 높은 슈트이다. 리드 슈트는 한 트릭의 첫 번째 순서 플레이어가 낸 카드의 슈트로 결정되는데, 다른 플레이어들은 해당 슈트가 있다면 반드시 내야하고, 해당 슈트가 없다면 아무 카드나 낼 수 있다. 리드 슈트는 해당 트릭에 한해서 트럼프 슈트 다음의 등급을 가진다.

마이티 게임은 총 10번의 트릭으로 진행되는데, 본 논문에서는 각각의 트릭을 라운드로 칭한다. 즉, 게임은 총 10라운드로 진행된다. 첫 라운드는 주공이 시작하는데, 주공은 1라운드에서 트럼프 슈트를 낼 수 없다. 또한 조커와 조커 콜러도 낼 수 없다. 주공이 첫 카드를 내면 1라운드의 리드 슈트가 결정되며, 주공에서부터 시계 방향으로 플레이어들이 카드를 한 장 낸다. 이때 각 플레이어는 리드 슈트가 손에 있을 경우 리드 슈트에 해당하는 카드를 반드시 내야 하는데, 마이티와 조커는 예외적으로 언제든지 낼 수 있다. 리드 슈트가 없을 경우에는 아무 카드나 낼 수 있다.

모든 플레이어가 순서대로 카드를 한 장씩 내고 나면 한 라운드, 즉 트릭이 끝난다. 각 라운드가 끝나면, 테이블에 내려놓은 다섯 장의 카드 중 등급이 가장 높은 카드를 낸 플레이어가 테이블의 점수카드를 모두 가져간다. 카드의 등급은 마이티, 무력화 되지 않은 조커, 트럼프 슈트, 리드 슈트 순이며, 같은 슈트 안에서는 숫자가 높은 카드가 등급이 높다. 이때 획득한 점수카드는 자신의 테이블에 카드가 보이도록 펼쳐 놓는다.

두 번째 라운드부터는 직전의 라운드에서 가장 높은 등급의 카드를 낸 플레이어가 손에 있는 카드를 한 장 내면서 시작한다. 이때 2라운드 리드 플레이어는 손에 있는 카드 중 아무 카드나 내려놓을 수 있으며, 해당 카드 무늬가 리드 슈트가 된다. 시계 방향으로 다섯 명의 플레이어가 모두 카드를 내려놓으면 한 트릭이 끝나고, 가장 높은 등급의 카드를 낸 플레이어가 해당 트릭의 점수카드를 모두 가져간다. 이후 10라운드까지 동일하게 진행되지만, 2라운드부터 9라운드까지는 조커가 마이티 다음으로 등급이 높고 10라운드에서는 조커가 무력화된다는 점이 다르다.

4.5 Scoring

10라운드가 끝나면 여당이 획득한 점수카드의 개수와 공약을 비교하여 승패를 결정하는데, 여당의 점수카드가 공약으로 제시한 수보다 같거나 많으면 여당의 승, 여당의 점수카드가 공약으로 제시한 수보다 적으면 여당의 패가 된다. 게임 시작 전 주공이 카드 3장을 교환할 때 주공이 버린 점수카드는 여당이 획득한 것으로 본다.

게임의 스코어는 두 가지 변수를 고려하여 계산된다. 주

공이 공약으로 제시한 점수카드의 개수가 첫 번째 변수인데, 공약이 높을수록 여당이 승리하기 어려운 게임 조건임을 의미하므로 그에 따른 위험수당을 반영한다. 두 번째 변수는 주공이 공약으로 제시한 점수카드 개수와 여당이 실제 획득한 점수카드 개수의 차이이다. 차이가 크면 크게 승리하거나 크게 패배했다는 것을 의미하므로 그에 따른 인센티브(카드수당)를 반영한다. 스코어 계산식은 게임 전략에 중요한 영향을 주는 요인으로, 본 논문에서는 다음 공식을 따른다.

- 여당 승리점수 = (공약 장수 - 기본 장수 + 1) × 위험수당 + (여당 획득 장수 - 공약 장수 + 1) × 카드수당
- 여당 패배점수 = -1 × (공약 장수 - 기본 장수 + 1) × 위험수당 + (공약 장수 - 여당 획득 장수) × 카드수당

< 게임 조건 예시 >

- # 기본 장수 = 13
- # 위험수당 = 300
- # 카드수당 = 200

마이티는 제로섬 게임이기 때문에 승리 팀이 패배 팀의 보수를 가져간다. 게임 결과에 따라, 주공은 두 명의 야당과 보수를 교환하고 프렌드는 한 명의 야당과 보수를 교환한다.

5. Special cards

마이티 게임에는 세 장의 특수 카드가 있다. 가장 등급이 높고 언제나 낼 수 있는 마이티 카드, 두 번째로 등급이 높지만 특정 상황에서 무력화되는 조커, 그리고 조커를 무력화하는 조커콜러이다.

5.1 Mighty card

마이티는 게임 이름이기도 하지만, 특정 카드를 의미하기도 한다. 보통 마이티 카드는 ♠A를 가리킨다. 하지만 주공이 공약으로 제시한 트럼프 슈트가 스페이드일 경우에는 해당 게임에 한정하여 마이티 카드를 ♦A로 한다.

마이티는 가장 등급이 높은 카드이며 게임 중 어느 때라도 낼 수 있다. 조커가 조커콜러에 의해 무력화되는 상황에서도 조커와 마이티를 둘 다 들고 있다면 조커 대신 마이티를 낼 수 있다. 한편, 마이티의 슈트(♠ 혹은 ♦)가 리드 슈트인 라운드에서 플레이어의 손에 든 카드 중 마지막 남은 리드 슈트 카드가 마이티라면, 규칙에 따라 마이티를 내야 한다.

5.2 Joker card

조커는 마이티 다음의 등급을 가진다. 하지만 1라운드와 10라운드에서는 위력이 없으며, 조커콜러 카드에 의해서도 무력화된다.

주공은 1라운드의 리드 카드로 조커를 낼 수 없다. 첫라운드에서 주공이 아닌 다른 플레이어는 조커 카드를 낼 수 있지만, 조커로서의 위력은 없다. 2라운드부터는 리드 플레이어가 조커를 낼 수 있는데, 이때 Fig. 4와 같이 리드 플레이어가 원하는 무늬로 리드 슈트를 선택할 수 있다.

5.3 Joker caller card

조커콜러(Joker caller)는 조커를 무력화시킨다. 조커콜러 카드는 ♣3이며, 트럼프 슈트가 클럽이면 해당 게임에 한정하여 ♥3(일부 지역에서는 ♠3)이 조커콜러 카드가 된다.

조커콜러는 각 라운드의 리드 플레이어가 낼 때에만 조커 무력화 효력을 사용할 수 있다. 주공은 1라운드에서 조커콜러 카드를 사용할 수 없으며, 2라운드 이후에는 리드 플레이어가 조커콜러 카드를 내면서 조커 무력화 여부를 선택할 수 있다. 즉, 조커콜러 카드를 내면서도 조커 무력화를 선택하지 않을 수 있다. 또한, 자신이 조커를 들고 있으면서도 조커콜러 카드를 내면서 조커콜을 부를 수 있고, 이로써 자신이 조커를 가지고 있다는 사실을 다른 플레이어에게 알릴 수 있다.

조커콜이 선언되면 조커를 가진 플레이어는 해당 라운드에서 반드시 조커를 내야 하는데 이때 조커는 가장 낮은 등급이 된다. 하지만 이때 조커를 가진 플레이어에게 마이티가 있다면, 마이티를 내고 조커 무력화 상황을 모면할 수 있다.

IV. Web-based Mighty Simulator

1. Purpose of Mighty simulator

본 논문에서 개발한 마이티 시뮬레이터는 마이티 게임을 분석하기 위한 목적으로 설계하고 구현하였다. 개발 환경은 Intel CPU 3.6GHz, RAM 8GB, 윈도우 64비트 운영체제이다. 시뮬레이터의 UI는 html/css를 사용하여 구현했고, 모든 로직은 JavaScript 언어를 사용하여 구현했다. JavaScript 코드는 일반 텍스트에디터로 구현했고, 브라우저의 개발자 기능을 활용하여 구동 로그와 에러를 확인하였다. Google chrome 브라우저를 타겟 환경으로 설정하였으나, 타 PC 브라우저(Microsoft Edge, Naver Whale)와 모바일 브라우저(Android, BlackBerry, iOS, Opera,

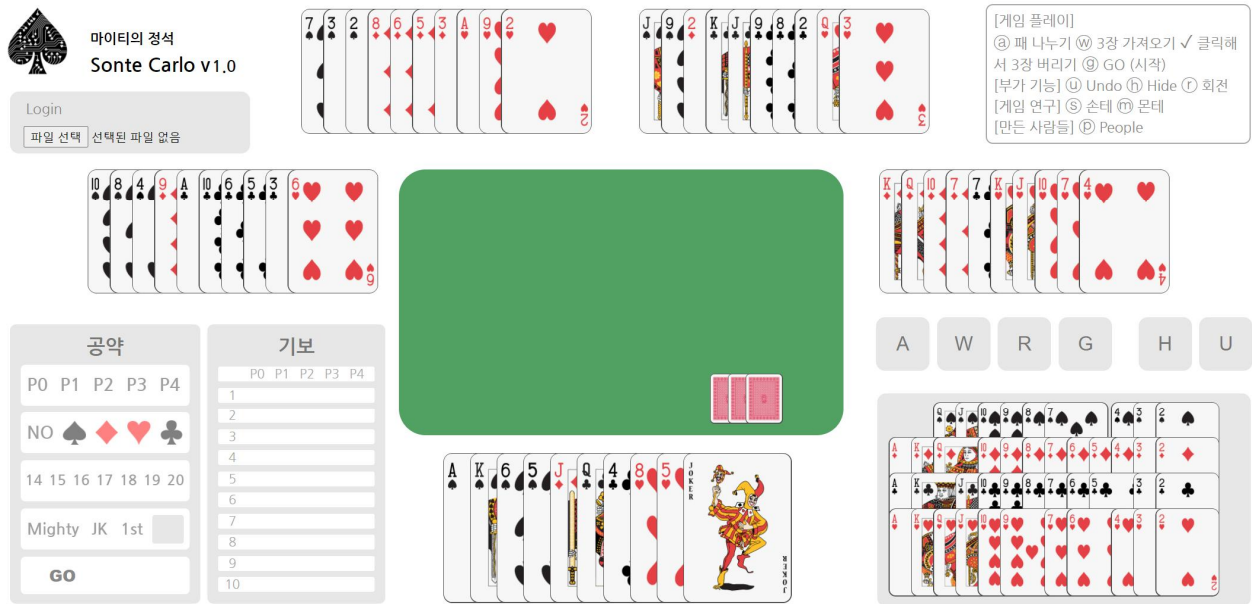


Fig. 4. Entire UI of Mighty Simulator

Windows)에서의 호환성 또한 검증하였다. 개발된 마이트 시뮬레이터는 기존의 마이트 네트워크 게임 프로그램들이 사용자들이 게임을 함께 즐길 수 있도록 네트워크 접속을 지원하거나 온라인 서비스를 제공하는 목적과는 차이를 가진다. 게임 분석을 위한 마이트 시뮬레이터의 기능은 크게 세 가지이다. 공약 연습, 오픈 카드 시뮬레이션, 리와인드 플레이 기능을 제공함으로써 게임 이해 및 전략 연구에 특화하였다.

공약 연습을 위해서는 딜 과정에서 플레이어가 받을 10장의 카드를 무작위 혹은 임의로 조합할 수 있도록 했다. 알고리즘이 추천하는 공약과 비교할 수도 있다. 특히 임의 조합 기능으로 연습 문제를 쉽게 만들 수 있고, 연습과 교육에 활용하기 좋다. 오픈 시뮬레이션을 위해서는 상대방 플레이어가 손에 든 카드를 모두 볼 수 있도록 했으며, 카드 카운팅 기능을 제공한다. 한편, 기보를 저장하여 게임 이해를 위한 통계 분석에 활용하거나, 저장된 기보를 불러와서 특정 시점으로 되돌려 리와인드 플레이를 할 수 있도록 했다.

2. Design and implementation of Mighty simulator

4.1에서 언급한 목적을 달성하기 위해, 마이트 시뮬레이터의 아키텍처를 Fig. 5와 같이 설계하였다. 아키텍처는 크게 공약 연습, 오픈 카드 게임 시뮬레이션, 리와인드 플레이를 제공하는 3개의 큰 모듈로 구성되어 있다. 각 기능들은 하나 이상의 함수들로 구현되었고, 화면 출력을 위한

함수들도 별도로 존재한다. 마이트 게임의 중간 상태를 저장하고 재현할 수 있기 위해, 모든 상황을 기록하는 변수들(GAME, CARD, TRICK)이 존재한다. 이들 변수는 모두 웹서버의 저장소에 저장되고, 리와인드 플레이를 위해 로드되어 특정 시점의 게임 상황을 완벽히 재현할 수 있다.

마이트 시뮬레이터는 Fig. 4에서 보이는 것처럼 웹 기반으로 구현되었다. 마이트 게임 로직 알고리즘과 기능은 JavaScript로 구현하고, 유저 인터페이스는 html과 css를 사용하였다.¹⁾

2.1 Contract practice

마이트의 전략은 비딩에서부터 시작되며 플레이어의 성향에 따라서도 비딩이 달라지기 때문에, 10장의 카드만으로도 비딩의 연습과 교육, 게임의 설계와 전략 분석이 가능하다.

마이트 시뮬레이터는 10장의 카드를 임의로 조합하는 기능을 제공하여, 비딩과 프렌드 선택 및 전략 분석을 위한 예제를 쉽게 만들 수 있도록 했다. 주공 카드를 초기화하고 우측 하단의 카드덱에서 카드를 클릭하여 임의로 주공 카드를 조합할 수 있다. 또한, 주공의 카드 10장을 제외한 나머지 카드들을 다른 플레이어들에게 무작위로 재분배하는 기능을 제공한다. 카드 조합의 다양한 경우의 수를 살펴보면서 공약이 적절했는지 평가하고 운영을 어떻게 할지 검토할 수 있다.

1) <http://sontecarlo.com> 에서 서비스를 제공함

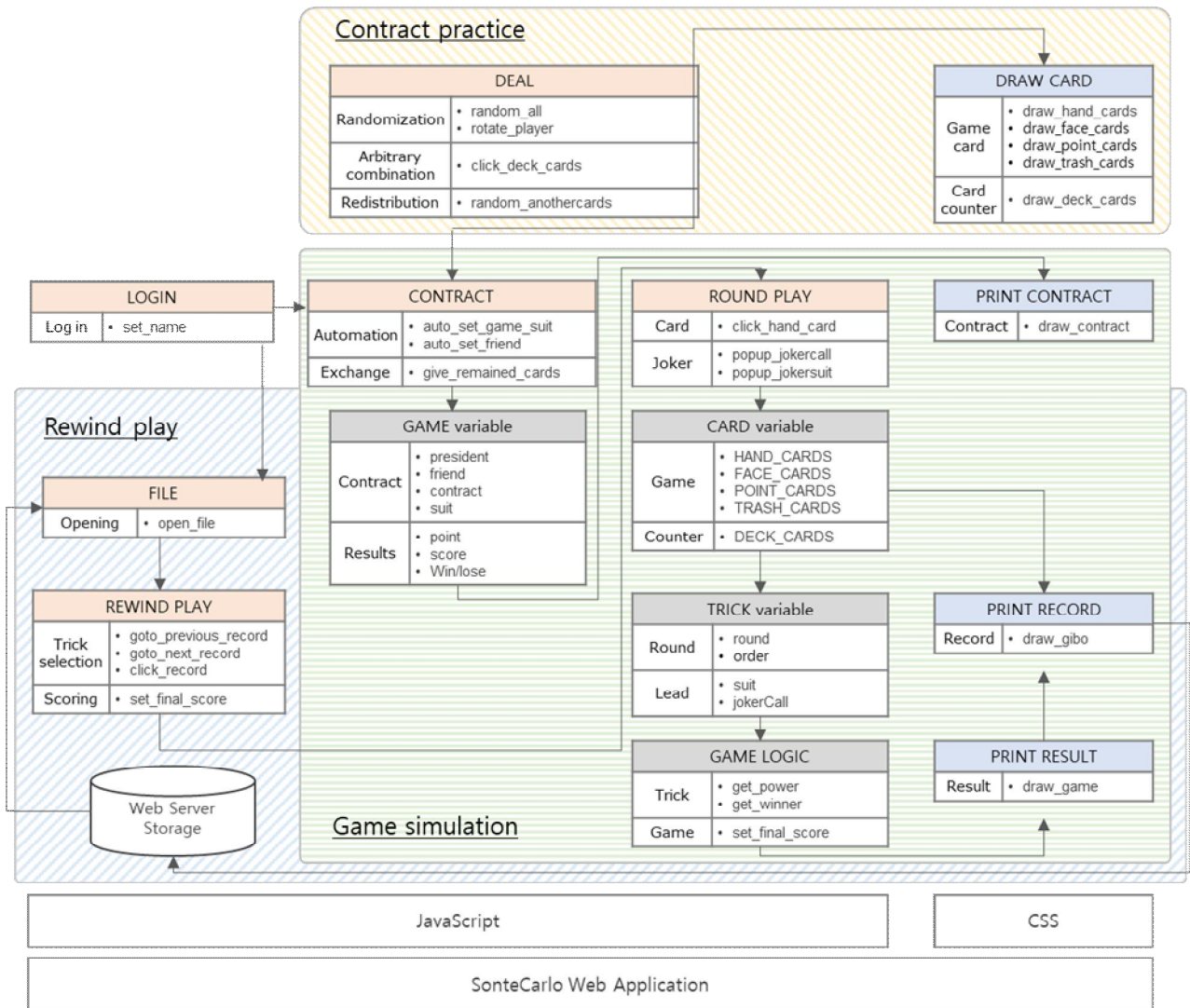


Fig. 5. Architecture of the Mighty Simulator

2.2 Open card and single player simulation

마이티 게임의 교육 및 연습을 위해서, 모든 카드를 오픈한 상태에서 1인 플레이 시뮬레이션을 할 수 있도록 구현했다. 특히 마이티 게임에서는 카드 카운팅(남아 있는 트럼프 슈트 카드의 개수를 세는 것)이 전략 수립에 중요하기 때문에, Fig. 4의 오른쪽 하단에 보이는 카드 카운터 기능을 제공한다.

마이티 시뮬레이터를 이용한 1인 플레이 시뮬레이션의 절차를 순서도로 표현하면 Fig. 6과 같다. 공약 연습을 위해서는 자동 공약 설정 단계까지 진행하면 되고, 게임을 시뮬레이션 하려면 이후 단계까지 진행한다. 마이티 게임 규칙에 익숙한 사용자의 경우, 게임 시뮬레이션에 2분가량 소요된다.

각 플레이어에게 카드 10장이 분배되고, 남은 3장은 테이블 위에 보이지 않게 배치된다. 하단의 플레이어부터 시계 방향으로 P0, P1, P2, P3, P4로 지칭하며, 편의상 사용

자 전면에 위치한 P0를 주공으로 한다. 다른 위치에 있는 플레이어의 카드를 주공으로 시뮬레이션 하려면, 플레이어의 카드를 시계 반대 방향으로 옮겨 P0의 위치로 가져올 수 있다.

무작위로 분배된 카드 조합을 보고 주공을 선택하면 시뮬레이션을 시작한다. 테이블 위에 놓인 3장을 가져오면 주공은 카드를 13장 가지게 되는데, 버리고 싶은 카드를 선택하여 다시 10장을 만들어야 한다. 카드를 잘못 버렸을 경우 언제든지 되돌리기를 할 수 있다.

공약 테이블은 Fig. 4의 왼쪽 하단과 같이 구현했는데, 가장 윗줄에는 주공 플레이어가 표시된다. 두 번째 줄은 주공이 설정한 트럼프 슈트, 세 번째 줄은 주공이 승리 조건으로 선언한 점수카드의 개수이다. 네 번째 줄은 주공이 프렌드로 선택한 카드인데, 마이티 카드(Mighty), 조커(JK), 첫 라운드에서 승리한 플레이어(1st)를 기본으로 배치했고, 텍스트 입력창에 특정 카드를 입력하여 선택할 수

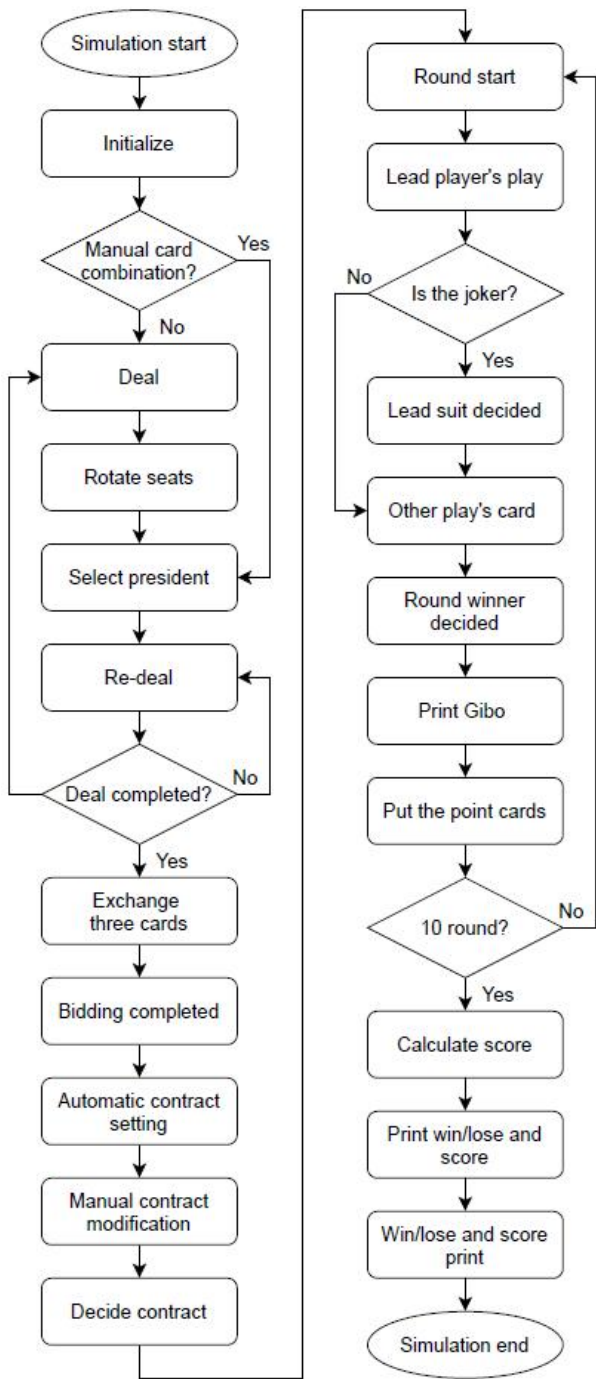


Fig. 6. Flow of the Mighty Simulator

있도록 했다. 트럼프 슈트와 게임 조건, 프렌드 카드는 자동 설정 이후에 변경할 수 있다.

주공이 공약을 확정하면 라운드를 시작하면 된다. 주공의 카드 10장 중 하나를 클릭하여 테이블에 내고, 이어서 Fig. 7과 같이 시계 방향으로 각 플레이어의 카드 하나를 클릭하여 차례로 테이블에 낸다. 한 라운드가 끝나면 가장 높은 계급의 카드를 낸 플레이어가 라운드 워너가 되고, Fig 8과 같이 테이블에 나온 다섯 장의 카드 중 점수카드

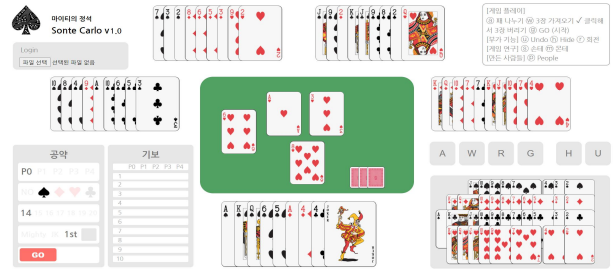


Fig. 7. President led the first round

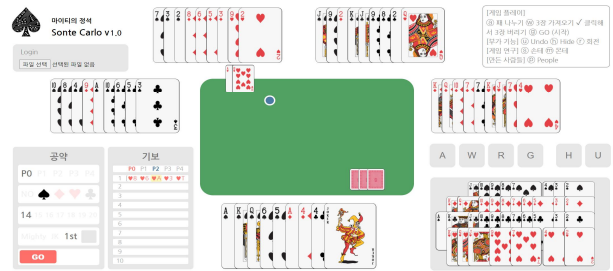


Fig. 8. End of the first round

는 모두 라운드 워너 앞에 펼쳐진 채로 놓인다. 라운드 워너는 다음 라운드를 시작해야 하므로 테이블에 표식을 두었다. 라운드 워너 표식의 파란색은 프렌드를 의미하며, 라운드 워너가 주공이면 빨간색, 야당이면 흰색으로 표시했다. 그리고 이때 마이티 시뮬레이터 좌측 하단의 기보에 한 줄이 추가되는데, 기보에는 각 플레이어가 해당 라운드에서 낸 카드가 기록된다.

이어서 1라운드 워너인 P2 플레이어가 2라운드를 시작하고, P2 플레이어가 선택한 리드 슈트에 따라 다른 플레이어들도 카드를 제출한다. 총 10라운드가 진행되어 플레이어 손에 든 카드가 남지 않으면 게임이 종료된다.

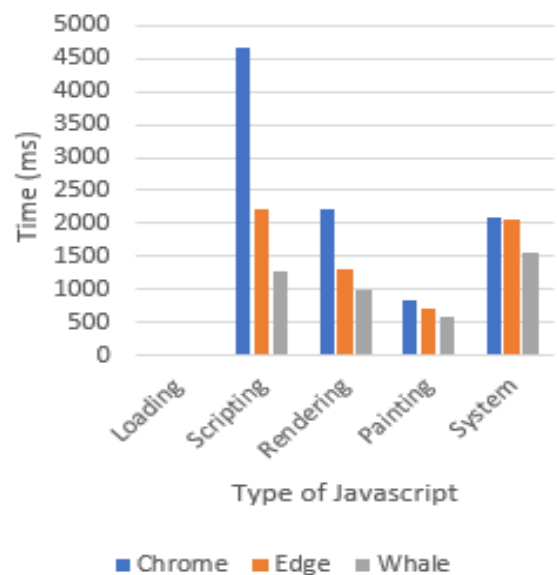


Fig. 9. Executing time of JavaScript per Mighty game

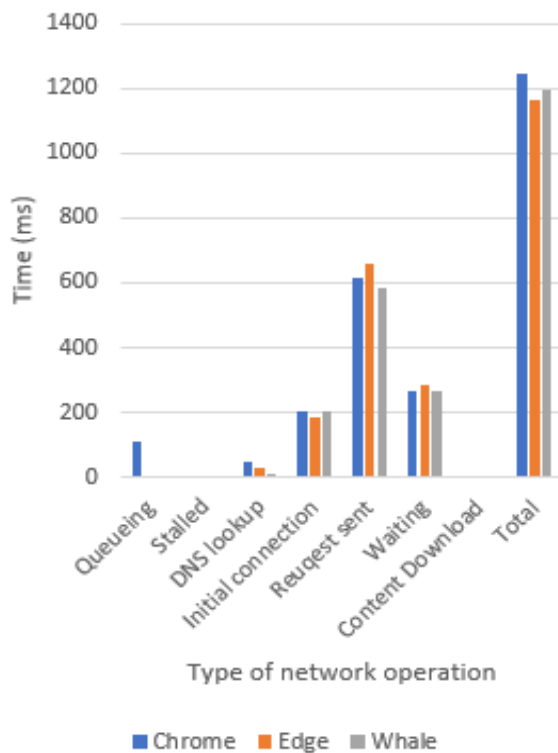


Fig. 10. Record saving time of Mighty game

2.3 Gibo save and rewind play

기보는 Fig. 4의 왼쪽 하단과 같이 열 번의 라운드에서 각 플레이어가 낸 카드를 기록한다. 기보 첫째 줄은 플레이어의 역할을 색깔로 표시했는데, 주공 플레이어는 게임 시작부터 빨간색으로 표시되고, 프렌드는 명시적으로 드러나는 순간부터 파란색으로 표시된다. 각 라운드에서 리드 플레이어가 낸 카드는 ‘(’ 기호로 표시하고, 라운드 위너는 노란색 배경으로 표시하였다. 각 슈트는 ♠, ♦, ♥, ♣으로 표시하고, 숫자는 A, K, Q, J, T, 9, ... , 2로 표시했다. 숫자 10을 T로 표시한 것을 제외하면 플레이 카드의 숫자와 동일하다. 조커는 ☆로 표시하였고, 라운드의 첫 카드로나온 조커는 리드 슈트를 지정해야 하므로 ☆♥와 같이 조커와 리드 슈트를 함께 표시했다.

기보는 전체 라운드에 걸쳐 각 플레이 별로 카드 상태 및 모든 환경 변수들을 json 파일 형식으로 자동 저장된다. 리와인드 플레이 기능을 통해, 기보 파일을 불러와서 플레이를 계속 진행하거나 특정 라운드로 되돌려 새로운 전략으로 플레이하여 게임의 결과를 볼 수 있다.

3. Performance analysis of mighty simulator

구현된 웹 기반 마이티 시뮬레이터는 자바스크립트로 구현되어 있기 때문에, 시뮬레이터가 제공하는 기능에 따른 소요 시간을 측정하여 성능을 분석할 수 있다. 사용자

의 조작에 따른 게임 처리속도는 마이티 게임 규칙의 숙지 여부와 시뮬레이터의 사용 목적에 의존하기 때문에 성능을 측정하기에는 부적합하다. 하지만 사용자가 시뮬레이터를 통해 일반적으로 한 게임을 수행하고 보상을 계산한 뒤 기보를 저장하는 연산은, 동일한 자바스크립트 함수를 동일 횟수로 수행하기 때문에 시뮬레이터의 평균 성능을 측정할 수 있다. 또한 로컬에서 한 게임을 완료하고 게임 정보가 기록된 기보를 웹서버에 업로드하는 단계에서의 성능을 측정할 수 있다. 테스트 환경은 Intel CPU 3.6GHz, RAM 8GB, 윈도우 64비트 운영체제이다.

첫째로, 게임 당 자바스크립트의 구동 시간을 측정하였다. 시뮬레이터를 구성하는 자바스크립트 함수를 5가지 타입>Loading, Scripting, Rendering, Painting, System)으로 구분하였고 게임을 10번 진행하여 평균 시간을 측정하였다. 또한 자바스크립트의 처리 성능은 웹브라우저마다 다르기 때문에 대표적인 웹브라우저 3종(Google Chrome, Microsoft Edge, Naver Whale)을 대상으로 동일한 절차로 평균 시간을 측정하였다. Fig. 9에 따르면, 자바스크립트의 로딩은 평균 14ms로 매우 빠른 반면, 스크립트 처리와 렌더링은 각각 2723ms와 1505ms의 처리속도를 보인다. 시스템 함수 호출은 1903ms이고, 화면 출력은 713ms를 기록했다. 웹브라우저 중에서는 Chrome이 가장 느리고, Whale이 가장 빠른 처리 속도를 보였다.

둘째로, 게임 기보의 서버 저장 시간을 측정하였다. 웹브라우저와 웹서버 사이의 네트워크 연산을 7단계(Queuing, Stalled, DNS lookup, Initial connection, Request sent, Waiting, Content Download)로 게임을 10번 진행하여 평균 시간을 측정하였다. 마찬가지로 웹브라우저 3종(Google Chrome, Microsoft Edge, Naver Whale)을 대상으로 동일한 절차로 평균 시간을 측정하였다. Fig. 10에 따르면, 기보 저장에는 평균 1201ms의 시간이 소요되었고 실제 기보 파일을 전송하는 Request sent 단계가 평균 619ms로 가장 많은 시간을 소모하였다. 웹브라우저 종류에 따른 차이는 거의 존재하지 않았지만, 상대적으로 Chrome이 가장 느린 성능을 기록하였다.

V. Results

본 장에서는 기존의 트릭테이킹 방식 게임인 브리지[9], 휘스트[19], 하트[20,21], 스페이드[22]와 마이티를 비교 분석하여 마이티 게임이 불완전 정보 게임으로서 연구 가치가 있는 환경임을 보인다. Table. 1을 통해 기존 카드게

Table 1. Comparison of characteristics with existing card games

Title	Natl	# of Player	Teamply	Team selection	Special rule	Round	Winning Condition
Mighty	KOR	5	2 : 3	Specific card	Joker call	10	Satisfying contract
Whist	UK	4	2 : 2	Position	X	13	2 nd out of 3 rounds
Bridge	USA	4	2 : 2	Position	Bonus/Penalty	13	100 points below baseline
Heart	USA	4	X	X	Heart breaking, Shoot the moon	13	Accumulated scores
Spade	USA	4	2 : 2	Position	Sack penalty, Double Neal, Neal	13	Your score of 500 or higher or your opponent's score of -200 or lower

임과 마이티 게임의 특징을 비교 고찰할 수 있다.

첫 번째로 마이티 게임은 기존 카드게임에 비해 정보가 가장 적게 제시된다. 가장 많은 인원인 5명의 플레이어가 참여하고, 각 플레이어는 자신에게 분배된 10장의 카드와 주공의 공약, 각 라운드의 기보 정보만으로 게임 상황을 분석해야 한다. 매 게임마다 팀 구성이 변동되기 때문에, 프렌드를 제외하면 팀원을 알지 못하며, 타 플레이어의 플레이만으로 팀원 여부를 예측해야 한다. 따라서 마이티 게임은 피아식별이 어려운 상황에서의 불완전 정보 게임을 연구하기에 좋은 환경을 제공한다.

두 번째로 마이티 게임은 기존 카드게임에 비해 가장 수준 높은 협동 플레이가 필요하다. 프렌드를 제외하면 주공마저도 팀원을 모르는 상태에서 게임을 진행해야 하고, 주어진 상황에서 팀 승리에 도움이 되는 플레이를 수행해야 한다. 플레이 패턴에 기반하여 플레이어들의 역할을 예측해야 하는데, 이를 악용하여 타팀인 것처럼 행동하면서 신임을 얻다가 마지막에 정체를 드러내는 식의 고도화된 협동 전략도 가능하다. 따라서 마이티 게임은 고도화된 협동 플레이를 연구하기에 좋은 환경이다.

세 번째로, 마이티 게임은 기존 카드게임에 비해 명확한 승리 목표와 보상 체계를 가지고 있다. 게임 시작 시점에 주공이 공약한 조건을 만족하는가에 따라서 승패가 결정된다. 보상 조건은 팀원이 함께 획득한 점수카드 개수에 따라 산정되며, 점수카드가 많을수록 보상이 증가한다. 따라서 기존 게임들이 단순히 승패 여부에만 집중하여 소극적인 전략을 추구하는데 반해, 안정적이면서도 최대한의 보상을 위하여 적극적인 전략을 추구하는 특성을 가진다. 따라서 불완전 정보 게임에서 제로섬 조건의 문제를 연구하는데 좋은 환경이다.

네 번째로, 마이티 게임은 낮은 학습 변동성을 가지고 있다. 마이티 게임은 10라운드로 빠르게 게임이 종료되는데 반해, 기존 카드게임은 한 판에 13라운드가 필요하고 여러 판을 통해 승패 조건을 만족해야 한다. 또한 마이티 게임에서 가능한 특수 규칙은 기존 카드게임에 비해 상대적으로 적기 때문에 변칙적인 전략의 영향을 적게 받는다. 따라서

마이티 게임은 짧은 실행 크기와 낮은 변동성을 가지는 특성으로 인해, AI 학습 연구가 용이한 환경이다. 특히, 멀티 에이전트 환경에서 적용 가능한 강화학습 기법을 사용하여 최적의 플레이를 예측할 수 있다. 또한, 마이티 게임 플레이 도중에 얻을 수 있는 정보(예. 주공, 카드, 공약, 기보 등)를 시뮬레이터 동작 과정에서 변수들(4.2절의 GAME, CARD, TRICK)에 저장하고 특징맵 구축에 활용함으로써 순환신경망(RNN) 등 인공신경망 모델을 학습할 수 있다.

VI. Conclusions & Limitation

마이티는 불완전 정보 게임 중에서도 협동 전략의 수립이 어려운 게임이지만, 규칙이 간단하고 변동성이 낮아서 AI 학습을 연구하기에 효과적인 환경이다. 하지만 국제적으로 대중화된 브리지 게임과 비교하면 비딩 및 게임 전략 연구가 부재하며, 심지어 마이티 게임에 대한 학문 차원에서의 정의 및 이해가 부족한 상황이다. 본 연구에서 정의한 마이티 게임의 세부 내용 및 마이티 시뮬레이터 구현을 통해 마이티 게임의 전략 분석 및 평가에 활용할 수 있다.

하지만 본 연구의 단점으로 정량적 측면과 정성적 측면의 한계를 들 수 있다. 정량적 측면에서는 본 시스템의 체계적인 성능 분석이 부족하다. 시스템의 가용성, 안정성, 안전성 측면에서 심도 있는 검토 및 보완이 필요하다. 정량적 측면에서는 본 시스템의 기능 수준에 대한 검증이 부족하다. 전문가 집단의 검증을 통해 각 기능의 만족도를 살펴보고, 개선 방안 및 추가 기능을 제시하여 활용도를 높이는 연구가 필요하다.

향후에는 본 시스템을 운용하면서 적용 사례를 다수 확보하고, 인공지능 플레이어 개발의 위한 AI 학습에 적용할 계획이다. 구체적으로, 시뮬레이션 결과인 게임 기보를 다양한 학습자들로부터 수집하여 기존 카드게임과 같이 향후 AI 학습을 위한 데이터셋으로 활용[23]할 것이다. 한편, 마이티 게임의 전략 분석에 있어서 가장 중요한 정보는 다른 플레이어들이 가질 수 있는 카드 조합의 확률인데, 카

드 분배 확률은 수학적으로 계산하기에는 변수가 많아서 몬테카를로 시뮬레이션으로 추정하는 것이 적절하다. 향후에는 마이티 시뮬레이터에 몬테카를로 시뮬레이션 기능을 추가하여 카드 분배 확률을 통계적으로 추정하고 그에 따라 플레이 전략을 분석하는 연구를 할 계획이다. 궁극적으로 마이티 게임의 분석 결과를 바탕으로 AI 학습을 수행하여, 불완전 정보 게임의 문제들을 해결할 수 있는 방안을 연구할 계획이다.

ACKNOWLEDGEMENT

The authors would like to express our gratitude to Youngdong Kwon(WonikIPS), Min-Soo Kim(Korea Electrotechnology Research Institute), Byoung-Jik Kim(Korea Institute of Nuclear Safety), Sehoon Kim(Agency for Defense Development), Jongjoo Kim(Korea Institute of Science and Technology), Joonhyoun Kim(The Board of Audit and Inspection of Korea), Hyochang Kim(Agency for Defense Development), Sang-Kil Son(Deutsches Elektronen-Synchrotron), Inseok Yang(Korea Research Institute of Standards and Science), Min-Wook Oh(Hanbat National University), Seungpil Jung(SK earthon), and Young Tae Cho(Changwon National University) for their support and feedback in the process of developing Mighty Simulator.

REFERENCES

- [1] S. David, et al. "Mastering the game of go without human knowledge," *Nature*, Vol. 550.7676, pp. 354-359, 2017. <https://doi.org/10.15368/theses.2018.47>
- [2] H. Johannes, S. David, "Deep reinforcement learning from self-play in imperfect-information games," *arXiv preprint arXiv:1603.01121*, 2016.
- [3] A. Kai, C. Antoine, T. Julian, "Alphastar: An evolutionary computation perspective," *Proceedings of the genetic and evolutionary computation conference companion*, pp. 314-315, 2019. <https://doi.org/10.1145/3319619.3321894>
- [4] G. Qucheng, J. Yu, T. Yuandong, "Simple is better: Training an end-to-end contract bridge bidding agent without human knowledge", *Real-world Sequential Decision Making workshop at ICML 2019*, 2019.
- [5] S. Stefano, "Equilibrium Computation Techniques for Bridge," PhD Thesis. University of Illinois at Chicago, 2020.
- [6] Y. Chih-Kuan, H. Cheng-Yu, L. Hsuan-Tien, "Automatic bridge bidding using deep reinforcement learning," *IEEE Transactions on Games*, Vol. 10.4, pp. 365-377, 2018. <https://doi.org/10.1109/tg.2018.2866036>
- [7] Z. Ruiyong, et al. "Design and Development of Bridge AI bid Program based on Double-hand Solver," *IEEE Chinese Control And Decision Conference (CCDC)*, pp. 6281-6286, 2019. <https://doi.org/10.1109/CCDC.2019.8833178>
- [8] F. Ian, "Search and Planning under Incomplete Information: a study using Bridge card play," *Springer Science & Business Media*, 2012. [https://doi.org/10.1016/S0004-3702\(97\)00082-9](https://doi.org/10.1016/S0004-3702(97)00082-9)
- [9] BridgeBuff, <http://www.bridgebuff.com>
- [10] Dealmaster Pro, <http://dealmaster.com>
- [11] Deal 3.1, <http://bridge.thomasoandrews.com/deal>
- [12] Ted's Bridge World, <https://tedmuller.us/Bridge/Simulations.htm>
- [13] Peter Cheung's Bridge Website, <http://crystalwebsite.tripod.com>
- [14] Mighty Network 4.0, Celicath, <https://tail41.snucse.org>
- [15] MightyPlayer by WithCat, <https://code.google.com/archive/p/mightyplayer>
- [16] Y. Cho, J. Chulmin, M. Kim, J. Pannell and D. Roth, "Mighty-card game rules," <https://www.pagat.com/picture/mighty.html>
- [17] Jack Jack Knave, <https://jackjackknave.com/game/mighty>
- [18] Denexa Games, "Mighty", <https://www.denexa.com/blog/mighty>
- [19] O. Tyler A. *The Game of Whist*. 2013.
- [20] S. Nathan R, W. Adam M, "Feature construction for reinforcement learning in hearts," *International Conference on Computers and Games*. Springer, pp. 122-134, 2006. https://doi.org/10.1007/978-3-540-75538-8_11
- [21] W. Maxiém, "Learning to play the Game of Hearts using Reinforcement Learning and a Multi-Layer Perceptron," PhD Thesis. University of Groningen. 2017.
- [22] W. Daniel, et al. "Integrating monte carlo tree search with knowledge-based methods to create engaging play in a commercial mobile game," *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2013. <https://dl.acm.org/doi/10.5555/3014712.3014728>
- [23] R. Douglas, S. Christopher, B. Michael, "Learning policies from human data for skat," *IEEE Conference on Games (CoG)*. pp. 1-8, 2019. <https://doi.org/10.1109/CIG.2019.8848125>

Authors



Jeongwon Lee received the B.S. in Electric Engineering from Seoul National University, M.S. in Department of Biomedical Engineering from Seoul National University, and Ph.D. degrees in Department of Bio and

Brain Engineering from KAIST. Dr. Lee joined the Electronics and Telecommunications Research Institute, Daejeon, Korea, in 2002. He is currently the COO in PEBBLOUS Inc.



Kwihoon Kim received the B.S, M.S. and Ph.D. degrees from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea in 1998, 2000 and 2019, respectively.

Kwihoon Kim is currently a professor in the Department of Artificial Intelligence Convergence Education, Korea National University of Education (KNUE), South Korea. He worked in LG DACOM 2000~2005. From 2005 to 2020, he was a Principle Researcher with Electronics and Telecommunications Research Institute (ETRI). He is interested in AI convergence education, intelligent edge computing, reinforcement learning and knowledge-converged intelligent service.



Seung-Hyun Kim received the B.S. degree in computer engineering from the Kumoh National University of Technology, South Korea, the M.S. degree in computer engineering from POSTECH, South Korea,

and the Ph.D. degree in School of Computing from the Korea Advanced Institute of Science and Technology (KAIST), South Korea. He was a principle researcher of the Information Security Research Division, Electronics and Telecommunications Research Institute(ETRI), South Korea, from 2004 to 2021. He is currently an assistant professor in the Department of Computer Education, Korea National University of Education, South Korea. His main research areas are blockchain, mobile privacy, and computer-based education.