

사례 기반 추론법을 이용한 오텔로 게임 개발에 관한 연구

송은지*

요약

인공지능(AI: Artificial Intelligence)은 지능을 만들 수 있는 방법론이나 실현 가능성 등을 연구하는 컴퓨터 공학 및 정보기술과학 분야이다. 오텔로(Othello) 게임은 다른 게임에 비해 규칙이 간단하며 8 x 8 인 제한적인 공간에서 이루어지기 때문에 AI로 제작되는 사례가 많다. 기존의 알고리즘은 추후에 발생하는 모든 경우의 수를 탐색하거나 룰을 이용하여 처리하기 때문에 처리시간이 오래 걸리며 새로운 상황에 대처하는데 효율적이지 않다.

본 연구에서는 이런 단점을 보완하고자 오텔로 게임에 AI의 한 분야인 사례기반추론(CBR: Case-based Reasoning)알고리즘을 도입한다. CBR알고리즘이란 주어진 문제를 해결하기 위해 과거에 있었던 유사한 문제를 검색하여 상황에 맞는 해결방법을 제시하는 방식을 의미한다. 지금까지 오텔로 게임에 여러가지 AI기술을 이용하였으나 CBR알고리즘을 적용한 사례가 없었다. 본 연구에서는 CBR알고리즘을 오텔로 게임에 적용하여 보다 빠른 연산속도로 다음 작업을 처리할 수 있으며 기존의 사례가 충분할 때는 새로운 상황에 효율적으로 대처할 수 있을 뿐 아니라 사용자로 하여금 보다 어려운 오텔로 게임을 만들 수 있는 시스템을 제안한다.

A Study on the Image Search System using Mobile Internet

Eunjee Song*

Abstract

AI(Artificial Intelligence) refers to the area of computer engineering and IT technology that focuses on the methodology and creation of intelligent agents. The Othello game is often produced with AI, since it is played with relatively simple rules on a board and on a limited space of 8 rows and 8 columns. Previous algorithms take longer time than desirable and often fail to face new circumstances, as they search for all the possible cases and rules. In order to solve this crucial weakness, we propose that a CBR algorithm be applied to Othello. Case-Based Reasoning(CBR), is the process of solving new problems based on the solutions of the past similar problems. We can apply this process to Othello and expedite the process of computer reasoning for a solution to new cases based on the data from accumulated past cases. Then, these new solutions are dynamically added to the set of past cases so that it becomes harder for players(users) to be able to read the pattern. The proposed system in which a CBR algorithm is applied to the Othello game makes the computation process faster and the game harder to play.

key words: Artificial intelligence , Othello game, Cased-based reasoning

1. 서론

인간은 때때로 현 문제를 해결하기 위해서 과거의 경험을 바탕으로 해결방법을 추론한다. 과거의 경험에 근거한 추론방법은 저장되어 있

는 사례들 중 가장 연관되어 있는 사례들을 조회한 후 이를 현재의 문제에 적합하게 변형시켜 결론을 도출하는 방법이라 할 수 있다. 이러한 방법은 인공지능이라는 분야에서 활용되고 있다. 인공지능(AI: Artificial Intelligence)이란 철학적으로 인간이나 지성을 갖춘 존재, 혹은 시스템에 의해 만들어진 지능을 뜻하며 일반적으로 범용 컴퓨터에 적용한다고 가정한다. 또한 그와 같은 지능을 만들 수 있는 방법론이나 실현 가능성 등을 연구하는 컴퓨터 공학 및 정보

※ 제일저자(First Author): 송은지
접수일:2011년 06월 07일, 완료일:2011년 06월 21일
* 남서울대학교 컴퓨터학과
sej@nsu.ac.kr
■ 이 논문은 2011년도 남서울대학교 학술연구비 지원에 의하여 연구 되었음

기술과학 분야를 지칭 한다[1].

AI는 로보틱스, 음성인식 등 여러 분야에 응용되고 있는데 특히 게임에 많이 적용하여 사용되고 있다. 그 예로서 CBR(Case-based Reasoning)알고리즘을 사용하는 블랙&화이트, RBR(Rule-Based Reasoning)알고리즘을 이용하는 심즈와 어쌔신 크리드, 확률 기반 행동 결정 시스템을 이용하는 프리스타일 등이 있다 [2].

본 논문에서 이용한 사례 기반 추론 기법(CBR, Case-Based Reasoning)이란 주어진 새로운 문제에 대해 과거 유사한 문제의 해답을 채택하거나 과거의 사례를 이용하여 새로운 상황을 설명 및 이해하고 새 해답을 평가하기 위해 선례로부터 추정하는 것을 의미 한다. 최근 까지 대부분의 전문가 시스템에서는 규칙기반 추론(RBR: Rule-Based Reasoning)을 주로 사용하였다. RBR에서는 문제영역의 규칙을 인간 전문가로부터 모두 추출한 다음 그것을 정리하여 규칙베이스(Rule-base)를 구현하며, 이를 추론함으로써 해를 얻는다 [3].

그러나 RBR은 잘못 추론한 똑같은 문제를 다시 줄 경우 계속 잘못된 추론을 한다는 치명적인 단점과, 지식의 적용 영역을 벗어나는 문제에 대해서는 전혀 도움을 줄 수가 없다. 이러한 문제의 해결책으로써 CBR이 나왔고 현재의 의료분야, 고객관리, 산업분야 등 많은 분야에 쓰이고 있다.

사례기반 추론기법(CBR, Case-Based Reasoning)은 특별상황의 사례들을 사용한 추론으로서 학습능력이 제한되지 않아 Original Scope이외의 문제를 해결 할 수 있다. 또한, 지식의 추가는 사례의 추가로 가능하다. 이와는 대조적으로 규칙기반 추론(RBR, Rule-Based Reasoning)은 어떠한 문제를 해결하는 데에 있어 필요한 자원이 적지만 기존에 특별한 자료 없이 알고리즘만 있으면 바로 적용이 가능하다는 좋은 점도 있다.[3]

본 논문에서는 오델로라는 게임에 CBR알고리즘을 적용한 시스템을 제안한다. 지금까지 오델로 게임에 여러 알고리즘을 적용하여 왔으나 CBR알고리즘을 적용한 사례는 없었다. CBR알고리즘에 관한 연구를 통해서 이 알고리즘이 일

반 게임 AI제작에도 좋은 조건을 갖추고 있다는 것을 알게 되었다. 예를 들어 승패가 갈라지는 게임에서 과거에 패했던 상황을 저장해 놓는다면 그 상황이 다시 한 번 재현될 때 똑같은 우를 범하지 않을 수 있게 되어 보다 높은 승률을 유지할 수 있게 해줄 것 이며, 그와 더불어 개발자가 일일이 AI를 업데이트 할 필요 없이 자기 스스로 업데이트 해나갈 수 있다는 것이다.

본 연구에서는 AI의 한 분야인 사례기반추론(CBR)알고리즘을 오델로 게임에 적용하여 처리속도가 보다 빠르고 새로운 문제에 효율적으로 대처할 수 있으며 사용자 하여금 보다 어려운 오델로 게임을 만들 수 있는 시스템을 제안한다.

2. 적용모델 정의

2.1 오델로의 정의

오델로란 서양에서 리버시라고도 불리기도 하는데 흑과 백, 이렇게 두 가지 색을 가지고 즐기는 보드 게임이며, 초록색이면서 8x8 크기로 이루어진 오델로판에서 게임을 하게 된다. 오델로는 OPENING BOOK을 이용하는데, 이는 바둑에서 특정 상황에 대한 풀이로 정석적인 것이라고 표현하고, 서양에선 정석을 오프닝이라고 표현한다. 오델로에서도 정석이 존재한다. 바둑 등에서의 정석은 자신이 유리한 상황으로 이끌고 가는데 비해 오델로의 오프닝은 흑과 백이 비슷한 상황에 초점을 맞춘다. 기본적으로 20수 전을 오프닝이라고 표현하고, 이후에는 MidGame 후반부를 EndGame이라고 표현한다. 보통 오프닝은 F5로부터 시작표기가 되어있다. 기본적으로 게임을 시작할 때 둘 수 있는 수는 4개이므로 꼭 첫수가 F5가 아니라고 해도 판을 돌리면 적용이 가능하여 표기하는 것이다. 따라서 컴퓨터의 초반 전략 자료로 Opening Book이 사용되고 있다. 데이터는 그대로 F5로 기록한다고 해도, 실제 게임의 시작위치에 따라 계산해서 넣는 건 좀 까다로운 작업이 될 수 있다. 참고로 바둑에 비해 둘 수 있는 수가 작다고 하지만 모든 수를 저장하고 계산하기엔 오델로도 경

우의 수가 많다.

두 대국자에게는 한쪽 면은 검은색, 반대 면은 흰색인 돌(밑의 사진 참조) 64개가 주어지고 처음엔 각각 전체 돌수의 반인 32개씩의 돌을 가지고 시작한다. 하지만 게임 도중 패스등의 발생으로 상대의 돌이 먼저 다 떨어지게 되면 자기가 가진 돌을 상대방에게 건네 주게 된다. 어느 색이 더 많은가에 따라 승패가 갈린다. 규칙은 다음과 같다.

- (1) 보드 위에 엇갈리게 사각형으로 배치된 돌 4개로 시작한다.
- (2) 돌은 상대방 돌을 양쪽에서 포위하여 잡을 수 있는 곳에 둔다.
- (3) 놓을 곳이 없는 경우에는 차례가 상대방에게 넘어 간다.
- (4) 더 이상 돌을 놓을 수 없게 되면 게임이 종료 된다.
- (5) 게임이 끝났을 때 돌이 많이 있는 플레이어가 승자 된다.
- (6) 돌의 개수가 같다면 무승부가 된다.

2.2 기존의 적용 사례 연구

오델로에 사용 되는 알고리즘은 현 경로상의 노드들만 기억하면 되므로 저장 공간의 수요가 적은 깊이 우선 방식(DFS : depth-first search)과 출발 노드에서 목표노드까지의 최단길이 경로를 보장하는 넓이 우선 방식(BFS : breadth-first search) 그리고 시간 낭비를 줄여주는 알파베타 가지치기(Alpha-Beta Pruning)와 최대 위협의 수를 최소로 결정 하는 미니맥스 알고리즘(Minimax Algorithm) 등을 들 수 있다. MINIMAX 알고리즘이란 상대나 자신이나 모두 최적의 수를 둔다는 전제하에 값을 도출하는 탐색법이다. 본 논문에서는 알파베타 가지치기와 미니맥스 알고리즘을 사용하는 Wzebra와 Logistello를 예로 들었다. Wzebra는 오델로를 하는 사람이라면 누구나 습득해야하는 프로그램이다. 여기에는 오델로 기보라는 것이 존재하여 대진하였던 무수히 많은 기록들이 저장되어 있다. 이 Wzebra는 거너 앤더슨이 위치에 앞서 많은 움직임을 검색하여 최고의 이동을 결정한다. 이 알고리즘은 프로그램이 필요로 하는 모든 위치를 고려하기 위해서 최선의 이동 위치를 찾을 수 있는 장점을 가지고 있다. 반면에, 검색 시 탐색 범위가

광범위 하게 되어 처리 속도가 늦어질 수 있다는 단점이 있다. 그러나 차후에 선택적 검색을 실시하여 중복되는 검색과정을 생략하면 처리 속도를 높일 수 있다.[7]

Logistello는 오델로를 통한 승패를 저장하여 사용자 이를 한눈에 알아 볼 수 있는 수치로 나타내주는 장점을 가지고 있다. Logistello는 마이클 뷰로가 효율적인 구현을 실행하는 하드웨어 트리 프로그램의 여러 가지 새로운 접근 방식에 대한 평가 기능의 건설을 목표로 개발하였다. 게임 아스키로 이동함에 따라 블랙보기에서 마지막 경기 결과에 주석을 추가한 시퀀스에 저장된다. 하지만, 다른 게임을 파일 형식으로 굉장히 많은 데이터를 처리하기 위한 도구, 통계 컴퓨팅, 물류 및 선형 회귀 시스템, 도서 학습을 해결하는 과정등으로 인해 처리 속도가 느다. 이것은 새로운 평가 함수를 일반 선형 회귀를 기반으로 처리 속도를 높일 수 있다.[7]

본 연구에서는 이 2개의 프로그램의 장점을 이용하고 사례기반추론 (CBR: Case-Based Reasoning) 알고리즘을 오델로 게임에 적용하였다. 기존의 추론방법은 일반화한 룰간의 관계를 통하여 최종 결론에 도달하는 방법이다. 이와는 달리 사례기반추론은 저장된 사례들 중 가장 연관된 사례들을 조회하여 이를 변형(Adapt)시켜서 결론을 도출하는 방법이다. 이러한 사례기반추론은 일반화한 룰에 의해 해결하기 어려운 복잡한 문제를 과거의 경험을 토대로 적용함으로써 보다 현실적으로 문제를 해결할 수 있으며 변화에 유연하게 대처할 수 있는 장점을 갖고 있다. 사례기반추론을 위해서는 과거사례의 표현, 유사한 사례의 검색 및 재활용, 새로운 문제에 대해 논리적으로 타당한 해법을 추론하는 기술이 필요하다. 특히 실제 과거사례가 많은 경우 이를 근거로 하여 추론을 하는데 많은 자원이 소요되므로 효율적으로 해결하기 위한 연구가 다양한 방면에서 수행되고 있다. 사례기반추론(CBR: Case-based Reasoning)알고리즘은 재료를 구울 때 어떤 배치가 효과적인지, 발생한 오류에 대해 과거에 있었던 문제점과 비교하여 같거나 유사한 오류의 해결방법을 사용자에게 제시해줄 때, 각 종류마다 다른 문제점이나 해결방법들을 과거 비슷한 자료들을 제공해주는

로써 효율적으로 문제를 해결할 수 있다. 따라서 여러 산업분야에도 많이 사용된다.

3. 시스템 설계 및 구현

3.1 구성

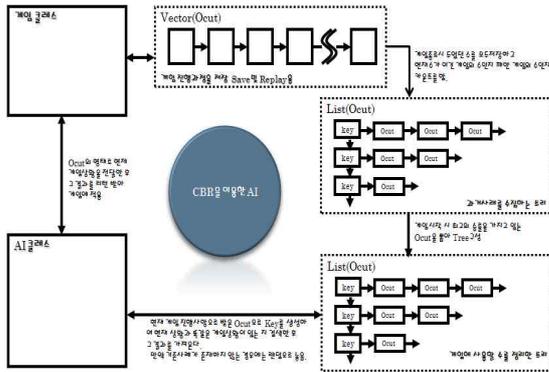


그림 1. CBR알고리즘을 이용한 AI 구성도

그림 1은 CBR알고리즘을 이용한 AI를 사용하는 오텔로의 전체적인 처리 과정인 게임 블록 구성도를 표현한 것이다. 구현하려는 오텔로의 AI에 CBR알고리즘을 적용하는 과정은 다음과 같다.

- (1) 사례 수집을 통해 현재 상황을 저장한다.
- (2) 사례정보의 단순화를 거쳐 수집한 사례에서 추론에 필요한 정보만 추려내어 저장한다.
- (3) 사례정보의 리스트화를 통해 정리한 사례 정보를 검색하기 용이하도록 정리한다.
- (4) 목적달성에 최적화된 과거 사례 검색을 이용하여 진행 중 컴퓨터의 상황을 키워드로 사용하여 최적의 수를 찾아낸다.
- (5) 추론해낸 결과를 게임에 반영한다.

3.2 설계 및 구현

시스템의 분석에 따라 다음과 같이 설계한다. 첫째, 사례 수집을 통해 현재의 게임 상황을

추후 추론에 반영하기 위하여 저장하여야 한다. 그림2처럼 현재의 게임 상황을 저장하는 모습을 나타낸 것이다.

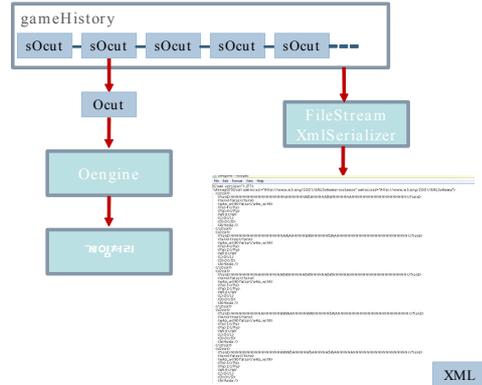


그림 2. 사례수집 구성도

그림 2와 같이 사례 저장과정은 '컴퓨터 대 사용자' 혹은 '사용자 대 사용자'의 게임이 진행될 때 그 게임 과정을 연속된 필름처럼 한 턱씩 저장을 해준다. 그렇게 저장된 정보가 게임이 종료될 때 일괄적으로 다음 게임에 사용하기 위하여 해당내용을 과거사례 저장하는 공간에 저장한다. 중복되는 사례가 있다면 승패 정보만 기록해주고 만약 중복되는 사례가 없다면 새로 생성해준다. 이때 저장하는 정보는 턴, 보드 판의 상황, 돌을 놓는 위치, 해당 게임이 종료될 시의 승패가 있다.

또한 과거사례를 저장할 때 사례정보 이용 키 값을 생성하여 추후 검색할시 용이하게 만든다.

둘째, 사례정보의 단순화 작업을 거쳐 수집한 사례에서 추론에 필요한 정보만 추려내어 저장한다. 모든 데이터를 저장하게 될 경우에는 그림 3과 같이 28가지의 노드가 필요하게 되지만 필요한 정보만을 수집하게 될 경우에는 이와는 다르게 그림 4에서처럼 75%의 노드를 절약할 수 있다.

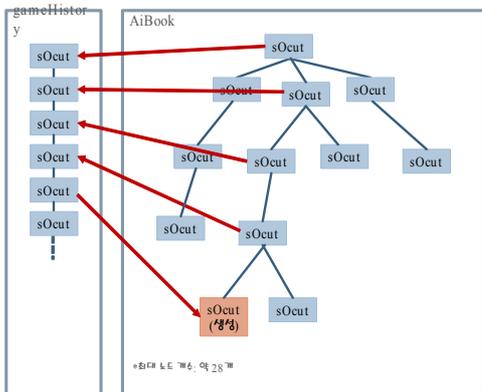


그림 3. 알파베타 가지치기를 사용하기 전

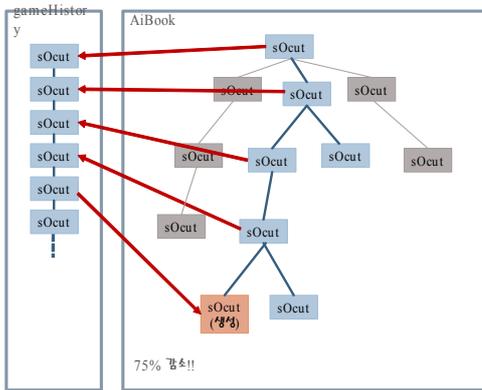


그림 4. 알파베타 가지치기를 사용한 후

셋째, 사례정보의 리스트화를 통해 정리한 사례정보를 추후 검색하기 용이하도록 정리한다.

사례 추출과정을 통해 컴퓨터와의 게임이 시작될 시 과거에 저장했던 사례들 중 가장 승률이 높았던 사례와 승률이 저조한(10%미만)의 사례를 리스트화 시켜 게임에 반영한다. 이렇게 미리 과거 사례를 추출하는 이유는 게임도중에 가장 최적의 수를 추론하는데 소모되는 시간을 줄이기 위해서다. 승률이 저조한 사례도 같이 저장하는 이유는 해당 수를 피하기 위해서이다. 여기서, CBR알고리즘을 이용할 때 가장 좋은 case 즉, 가장 효율적인 사례를 찾는 알고리즘으로 코호넨 네트워크의 경쟁학습 개념을 사용한다. 코호넨 네트워크의 경쟁학습이란 각각의 경쟁대상이 있는 경쟁층과 입력층의 연결강

도벡터와 입력 벡터 값을 비교한 후 가장 가까운 뉴런이 승리하도록 하여 채택하는 방식이다. 코호넨 네트워크의 학습 철학의 가장 큰 특징은 승자독점(winner take all)이다. 코호넨 네트워크와 마찬가지로 CBR에서 수많은 case중 가장 좋은 case를 선택할 때 각각의 case 승률을 카운트하여 승률이 가장 좋은 case를 선택한다.

```
int alphabeta(int ply, int alpha, int beta)
{
    if (ply == 0 || game_over()) return evaluate_current_board();

    for (Move *m = first_available_move(); m != NULL; m = next_available_move())
    {
        make_move(m);
        int new_value = -alphabeta(ply - 1, -beta, -alpha);
        unmake_move(m);
        if (new_value > beta) return new_value // prune
        if (new_value < alpha) alpha = new_value // update our "best so far"
    }
    return alpha;
}
```

그림 5. 알파베타 가지치기 사용 코드

넷째, 목적달성에 최적화된 과거 사례 검색을 이용하여 게임 진행 중 컴퓨터의 차례가 왔을 시 현재의 상황을 키워드로 사용하여 최적의 수를 찾아낸다.

사례 적용과정은 컴퓨터와의 게임이 시작되면 현재 게임의 상황을 검색어로 검색한 후 해당 결과가 있으면 게임에 반영한다. 전 과정에서 정리한 수이기 때문에 바로 사용가능하다. 여기서 과거 사례가 없는 경우 문제가 발생 할 수 있는데 그 문제에 대해서는 설명하기로 한다. 발생하는 문제점 및 해결방법은 먼저 과거 사례가 없을 경우는 MINIMAX 알고리즘을 이용하여 최적의 수를 찾아낸다. 이때 자신 혹은 상대의 최적의 수를 구할시 과거사례를 모아놓은 군집에 해당 사항이 있다면 그것을 활용하여 수행한다. 그리고 과거 사례의 수집이 사용자의 플레이로만 이루어진다는 점을 이용한다. 과거 사례를 모아놓은 군집에서 2개의 컴퓨터 AI군집을 생성, 서로 시합을 하게 함으로써 스스로 과거 사례를 생성하도록 만든다. 이때, 코호넨에

의해 개발된 자기조직화 형상지도 (Self-organizing Feature Maps) 알고리즘을 응용한다. 스스로 게임을 플레이하면서 새로운 뉴런(사례)를 삽입하고, 매 게임이 종료될 때 마다 그 연결 강도(승률)을 재조정 한다.

```

int maximize(int ply)
{
    if (ply == 0 || game_over()) return evaluate_current_board();
    int best = -infinity;

    for (Move *n = first_available_move(); n != NULL; n = next_available_move())
    {
        make_move(n);
        int new_value = minimize(ply - 1);
        unmake_move(n);
        if (new_value > best) best = new_value
    }
    return best;
}

Move *which_move_shall_i_take(int ply)
{
    Move* best_move
    int best_value = -infinity;

    for (Move *n = first_available_move(); n != NULL; n = next_available_move())
    {
        make_move(n);
        int new_value = maximize(ply);
        unmake_move(n);
        if (new_value > best_value)
        {
            best_value = new_value
            best_move = n;
        }
    }
    return best_move
}
}
    
```

그림 6. 사용된 미니맥스 알고리즘 코드

3.3 구현 결과 분석



그림7. 구동 화면

위의 그림 7은 본 시스템을 설계 구현하여 프로그램을 작동 시킨 화면이다. 플레이어가 컴퓨터와 대결을 계속하여 그 결과를 저장하고 그 저장한 내용을 가지고 다시 대결하였을 경우 패하지 않도록 더 효율적인 방법을 제시한다.

4. 결론 및 향후과제

인공지능(AI: Artificial Intelligence) 이란 마치 인간처럼 지각, 추론, 학습, 의사교환, 복잡한 환경에서 행동하는 능력을 보유한 컴퓨터 시스템을 개발하는 분야라고 정의할 수 있다. 인공지능은 로봇, 문자나 음성인식 등 그 응용분야가 매우 넓다. 그 중에서 게임에 이용되는 경우가 많은데 본 연구에서는 AI 의 한 분야인 사례기반추론(CBR: Case-based Reasoning) 알고리즘을 오텔로 게임에 적용한 시스템을 제안하였다. 지금까지 오텔로 게임에 여러 분야의 AI를 적용하였으나 CBR알고리즘을 적용한 사례는 없었다. 최근까지 대부분의 전문가 시스템에서는 규칙기반 추론(RBR: Rule-Based Reasoning)을 주로 사용하였다. RBR에서는 문제영역의 규칙을 인간 전문가로부터 모두 추출한 다음 그것을 정리하여 규칙베이스를 구현하며, 이를 추론함으로써 해를 얻는다. 그러나 RBR은 잘못 추론한 똑같은 문제를 다시 줄 경우 계속 잘못된 추론을 한다는 단점과 지식의

적용 영역을 벗어나는 문제에 대해서는 전혀 도움을 줄 수가 없다.

이러한 단점을 보완하기 위해 오델로게임에 CBR알고리즘을 적용한 시스템을 제안하였는데 CBR알고리즘은 일반 게임 AI제작에 좋은 조건을 갖추고 있다. 예를 들어 승패가 갈라지는 게임에서 과거에 패했던 상황을 저장해 놓는다면 그 상황이 다시 한 번 재현될 때 똑같은 우를 범하지 않을 수 있게 되어 보다 높은 승률을 유지할 수 있게 해줄 것이며 그와 더불어 개발자가 일일이 AI를 업데이트 할 필요 없이 자기 스스로 업데이트 해나갈 수 있다. 본 연구에서는 사례기반추론(CBR)알고리즘을 오델로 게임에 적용하여 처리속도가 보다 빠르고 새로운 문제에 효율적으로 대처할 수 있으며 사용자로 하여금 보다 어려운 오델로 게임을 만들 수 있는 시스템을 제안하였다.

AI정보는 수시로 바뀌지 않고 장기간에 걸친 작업으로 변하지만 향후 관리자가 축적된 사례를 알아 볼 수 있는 서비스를 제공한다면 보다 효율적인 알고리즘 구현이 가능할 것이라 사료된다.

참 고 문 헌

[1]이광형,조충호,“인공지능개론”, 흥릉과학출판사, 2000.
 [2] 조병현,박창준, “게임 인공지능 연구동향”, 전자통신동향분석 제23권 제4호, 2008.
 [3]홍태호, 이희정, 서보밀, “클러스터링 기반 사례기반추론을 이용한 웹 개인화 추천시스템”, 한국지능정보시스템학회논문지 제 11권 제1호, 2005.
 [4]Jacobs, Scott (EDT), “Game Programming Gems 7”, Charles River Media, 2007.
 [5]Chad Carter|이승현, 김상우 역, “Microsoft[®] XNATM(Xbox 360과 윈도우즈를 위한 그래픽과 게임 프로그래밍,Microsoft[®] XNATM Unleashed)”, 지앤선(지&선), 2008.
 [6] 이만재, “게임에서의 인공지능 기술”, 한국정보처리학회 논문지, 제9권 제3호, 2002.
 [7] 이동훈, 우종우, “강화학습에 기반한 오델로 게임의 설계 및 구현”, 한국정보과학회 2005 가을 학술발표 논문집 ,제32권 제2호, 2005.
 [8]권기덕, 김인철, “컴퓨터 게임에서의 인공지능” 인터넷정보학회 논문지, 제8권 제4호, 2007.

송 은 지



1984년 : 숙명여자대학교 수학과 (이학사)
 1988년 : 일본 나고야(名古屋) 국립대학 정보공학과(공학석사)
 1991년 : 일본 나고야(名古屋) 국립대학 정보공학과 (공학박사)
 1991년~1992년 : 일본 나고야(名古屋)국립대학 정보공학과 객원 연구원
 2007년 : 오克兰드대학교 컴퓨터학과 교환교수
 1996년~현 재 : 남서울대학교 컴퓨터학과 교수
 멀티미디어 기술사
 관심분야: 디지털 콘텐츠, 컴퓨터게임, 수치해석, 암호학 등