

## 거리를 고려한 줄고누게임의 학습속도 개선\*

신용우<sup>o</sup>, 정태충\*

동아방송예술대학 게임애니메이션계열<sup>o</sup>, 경희대학교 컴퓨터공학과\*  
littlespmk@hanmail.net, tcchung@khu.ac.kr

### An Improvement of the Learning Speed through Considered Distance on Jul-Gonu Game

Yong-Woo Shin<sup>o</sup>, Tae-Choong Chung\*

Dept. of Game Animation, Dong-Ah Institute of Media and Arts<sup>o</sup>,  
Dept of Computer, Kyung Hee University\*

#### 요 약

보드게임은 많은 수의 말들과 상태공간을 갖고 있다. 그래서 많은 시간동안 학습을 하여야 한다. 또한 상대방과의 대결이 1 대 1로 이루어지지 않고, 여러 말 대 여러 말로 이루어지므로 전략적인 사고가 필요하다. 그러므로 최적의 학습을 적용하여야 한다.

학습 도중에 동일한 최선 값이 있을 때, 줄고누의 문제 영역 지식을 활용한 휴리스틱을 사용해 학습의 속도 향상을 시도하였다. 강화학습을 이용한 말과 제안한 개선된 강화학습을 이용한 말을 비교하기 위해 줄고누게임을 제작하였다. 그래서 일방적으로 공격을 하는 말과 승부를 하게 하였다. 개선된 말은 거리를 고려하여 상대방 말을 공격하였다. 실험결과 개선된 강화학습을 이용한 말의 성능이 학습속도 측면에서 향상됨을 알 수 있었다.

#### ABSTRACT

It takes quite amount of time to study a board game because there are many game characters and different stages are exist for board games. Also, the opponent is not just a single character that means it is not one on one game, but group vs. group. That is why strategy is needed, and therefore applying optimum learning is a must.

If there were equal result that both are considered to be best ones during the course of learning stage, Heuristic which utilizes learning of problem area of Jul-Gonu was used to improve the speed of learning. To compare a normal character to an improved one, a jul-gonu game was created, and then they fought against each other. Improved character considered distance and attacked other one. As a result, improved character's ability was improved on learning speed.

접수일자 : 2009년 11월 11일

일차수정 : 2010년 01월 11일

심사완료 : 2010년 01월 26일

\* 교신저자 : 정태충

※ 이 논문은 2008년도 경희대학교 지원에 의한 결과임

## 1. 서론

2007년에 세계 온라인게임 시장규모는 70억 달러를 기록했으며, 이는 전년 대비 24.7%의 성장을 달성하였고, 이후에도 지속적으로 성장하여 2010년에는 132억 달러 규모로 성장할 것이다[1]. 게임 산업이 날로 발전하고 대규모로 제작되므로, 게임 프로그래밍도 인공지능과 같은 분야에 투자하여 부가가치를 만들어야 할 것이다. 과거에는 엔진과 게임프로그램으로 나누어 개발하던 시점에서 이제는 좀 더 세분화하여 분야별 전문프로그래머가 필요하게 되었다[2].

인공지능 분야에서 현재까지 캐릭터의 이동처리를 위해서는 패턴(Pattern), A\* 알고리즘이나 FSM(Finite State Machine), 퍼지(Fuzzy), FuSM(Fuzzy State Machine)을 이용하여 캐릭터의 자동화를 하고 있다. 패턴이란 미리 주어진 방향으로 캐릭터가 이동하게 하는 단순한 논리이다. FSM은 캐릭터의 여러 가지 행위를 상황에 따라 적용시키는 알고리즘이다. 또한 캐릭터의 움직임을 다양화시키는 퍼지나 FuSM 그리고 길 찾기에 사용되는 A\* 알고리즘 등은 캐릭터가 주어진 방향이나 목적지를 찾는 데에는 유용하게 사용할 수 있으나 캐릭터의 근본적인 지능을 높여주지는 못한다. 패턴이나 FSM을 이용한 캐릭터의 이동방법은 경우의 수가 작게 제한되어 단순하고 보다 많은 경우의 수를 가지는 캐릭터의 전투상황에서 효율적인 전투를 벌이지는 못한다. 그러므로 학습 알고리즘을 적용하여야 한다.

강화학습은 온라인 학습기법으로 캐릭터를 학습시킨다. 강화학습을 이용하여 특정목적을 달성하였을 때마다 보상을 하는 작업을 반복하다보면 캐릭터가 학습하게 된다[3].

강화학습 분야에는 상태공간의 문제, 캐릭터의 지능화 등의 문제가 있는데, 상태공간의 효율적인 사용 등의 문제는 여러 논문에서 다루었다. 그러므로 본 논문에서는 캐릭터의 자동화부분을 다루고자 한다. 기존의 캐릭터의 자동화를 다룬 논문들은 서

양보드게임인 오델로, 틱택토 등을 다루었으나[4,5] 본 논문에서는 줄고누 보드게임을 다루고자 한다.

두 가지 면에서 그 중요성을 설명할 수 있다. 첫째, 한국적인 보드게임으로서 아직 구현되지 않았다는 점이다. 둘째, 오델로 등의 게임은 말의 위치가 정해지면 더 이상 움직이지 않지만, 줄고누보드 게임의 경우 말의 위치가 계속 바뀐다는 점에서 전략적인 움직임이 필요하다.

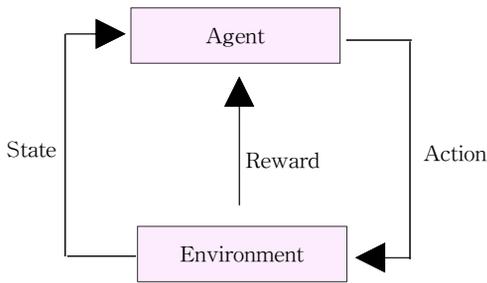
본 논문에서는 우리말과 상대방 말과의 대결상황에서 두 가지의 경우로 보드게임을 구현하여 비교하였다. 첫째, 강화학습을 이용하여 보드게임을 단순 구현하였다. 둘째, 강화학습에서의 최선 값을 산출하는 부분에서 동일한 값이 나올 때, 거리정보를 고려하여 유리한 값을 선택하도록 하였다. 두 가지 경우를 일정한 패턴으로 움직이는 말과 대국하게 하였다. 실험 결과 강화학습을 적용한 말보다 개선된 강화학습을 적용한 말이 우수한 경기를 벌이는 것을 알 수 있었다.

본 논문의 구성은 1장 서론에 이어 2장에서는 관련연구에 대해 살펴보고 3장에서는 지능형 보드게임의 구현에 대해 알아보고 4장에서는 실험 및 결과에 대해 알아본다. 마지막으로 5장에서는 최종 결론을 맺도록 한다.

## 2. 관련연구

### 2.1 강화학습

강화학습이란 많은 상태들의 집합인 환경에서 목적달성을 위한 행위를 수행하여 보상을 받음으로 필요한 행위를 학습하는 것을 말한다.



[그림 1] 강화학습의 모델

환경에는 목적달성에 필요하거나, 필요하지 않은 많은 상태들이 존재한다. 에이전트는 각각의 상태를 경험하여 목적달성의 경우 환경으로부터 보상을 받게 된다. 목적달성을 할 수 없는 상태인 경우 보상을 받을 수 없다. 그러므로 많은 시행착오를 겪을 수 있으며, 모든 상태를 경험해 보아야 한다.

강화학습에는 TD, Q-learning, SARSA 알고리즘 등이 있다. TD (Temporal Difference) 알고리즘은 상태 전이와 보상 값을 바탕으로 반복적으로 가치 함수를 학습한다. SARSA 는 결정된 전략에 따라  $a_{t+1}$ 을 선택한다. Q-learning은 사전에 모델을 설정하거나, 학습할 필요가 없으며, 상태공간을 충분히 경험한다면, 최적의 전략을 만들 수 있어 다양하게 활용되고 있다.

강화학습알고리즘으로서 일반적으로 많이 사용되는 Q-learning 에 대해 알아보자.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t))$$

[식 1]

특정상태  $s$  에서의 행위  $a$  에 대한 보상  $r$  을 얻을 수 있도록 [식 1]이 주어져 있다[3].  $t$  는 이전,  $t+1$  은 이후시간단위를 말한다.  $\max$  란 최대 값을 구함이고,  $\alpha$  는 학습속도 매개변수로서 0 에서 1 사이의 범위이다.  $\gamma$  는 할인계수로서 0 에서 1 사이의 범위이다.

모든 가능한 상태공간에서의 특정상태  $s$  에서의 행위 중 가장 좋은 보상  $r$  값을 저장하고 산출한다. 처음에는 모든 상태에 대하여 보상을 얻기 위

해 무작위로 행위를 하여 경험을 쌓게 된다. 그러나 시간이 지나 경험한 상태가 많아질수록 보상을 토대로 불필요한 상태의 경험을 필요치 않는다. 거의 모든 상태를 경험함으로써 학습이 완료되어 지능적으로 동작하게 된다.

## 2.2 보드게임 (Board Game)

오늘날에는 다양한 장르의 게임이 존재하지만 보드게임은 오랜 역사를 가지며, 단순한 규칙과 사용법으로 초보자를 포함한 많은 사람들에게 사랑받는 게임이다.

보드게임이란 상대방 캐릭터를 제압하는 형태의 게임으로 바둑, 장기, 체스, 오목, 오텔로등이 여기에 해당한다.

한게임을 포함한 여러 게임포털에서 다양한 종류로 서비스하며, 남녀노소 다양한 계층의 사람들이 많이 즐기는 게임장르라 할 수 있다.

본 논문에서 구현한 보드게임은 전통놀이 중 하나인 줄고누 이다.

## 2.3 고누게임 (Gonu Game)

고누놀이의 역사는 매우 깊으나 그 유래를 전하는 기록은 찾아 볼 수 없다. 다만 장기와 바둑의 원초적 형태를 띄고 있어서 고대 중국의 초나라와 한나라 때 생긴 장기놀이가 우리나라에 들어와 재창작된 것으로 짐작 된다[7].

고누놀이는 서민층에서 널리 보급되고 즐겨졌던 놀이인데, 조선시대 사실주의 화가 단원 김홍도의 풍속도에서 고누 두는 모습을 볼 수 있다.

고누놀이는 놀이방법이 단순 소박하여 아무 때 어느 곳에서나 쉽사리 벌일 수 있다.

그 놀이방법이 단순하여 누구나 쉽게 익힐 수 있는 고누는 땅 바닥이나 종이 혹은 널빤지에 말판을 그리고 돌멩이나 나무토막 등으로 말을 삼아, 두 편으로 나뉘어 벌이어 놓은 말을 서로 많이 따먹거나 잡아 가둠으로써 승부를 겨루는 놀이이다.

고누의 종류는 지방에 따라 여러 가지 특징이

있으나 대개 우물고누, 줄고누, 밭고누, 끈질고누, 참고누, 자동차고누, 호박고누, 패랭이고누, 팔팔고누, 포위고누, 장수고누, 왕고누로 구분해 볼 수 있다[8].

이러한 고누의 종류와 이름은 대개 말판의 모양에 따라서 그 명칭이 붙여진 것이다.

줄고누게임의 기본적인 규칙은 다음과 같다. 캐릭터들은 크게 우리말과 상대방 말이 있다. 우리말은 상대방 말을 공격하거나 방어하게 된다. 공격은 상대방 말 옆에 우리말 둘이 존재하면 상대방 말을 포획하게 된다. 또한 실수로 상대방 말의 일직선상에 들어갔을 때는 상대방 말에게 포획 당하도록 되어 있다. 게임 진행방법은 한 번에 한 칸씩 직선으로만 이동하게 되어있다.

### 3. 지능형 보드게임의 구현

본 논문에서는 상대방 말과의 대결을 통해 스스로 학습하는 우리말을 구현한다.

#### 3.1 캐릭터의 설계

캐릭터들은 크게 우리말과 상대방 말이 있다. 우리말이 둘 존재한다. 상대방 말이 둘 존재한다. [그림 2]에서 상단의 두말은 우리말이며, 하단의 두말은 상대방 말이다.

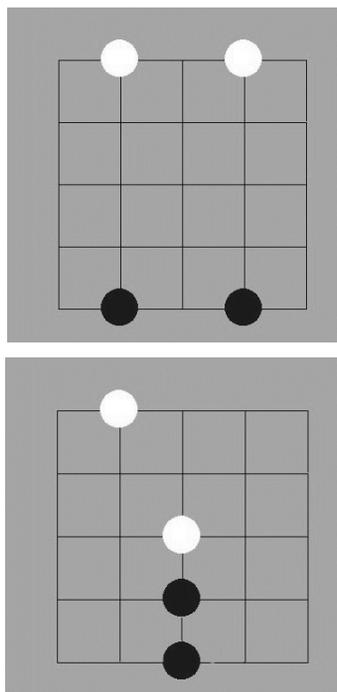
우리말은 상대방 말을 공격하거나 방어하게 된다.

공격은 우리말 둘이 일직선으로 나열되었을 때 그 끝에 상대방 말이 존재하면 상대방 말을 포획하게 된다. [그림 2]에서는 상대방 말이 우리말을 포획한 상태이다. 방어란 일직선상에 놓이지 않도록 피하면 된다.

상대방말의 공격은 우리말 쪽으로 자동적으로 움직이며, 한 수를 두어서 우리말을 포획할 수 있는 상황일 때는 정확히 우리말을 포획할 수 있도록 프로그램 되어 있다. 우리말 하나가 상대방 말에 인접하였을 때는 상대방 말이 포획할 수 있도록, 일직선이 되기 위한 목표지점으로 정확히 이동

하도록 되어있다.

또한 실수로 상대방 말의 일직선상에 들어갔을 때는 상대방 말에게 포획 당하도록 되어 있다. 상대방 말의 경우에도 실수로 우리말의 일직선상에 들어오면 포획 당하게 되어있다.



[그림 2] 우리말과 상대방 말의 초기상태와 포획한 상태

각각의 우리말과 상대방 말의 위치는 메모리에 각각 상황별로 저장되며 각각의 상황에서의 학습점수가 저장된다. 우리말이 움직일 때에는 각 상황에서의 선택할 수 있는 점수 중 유리한 것을 선택하게 된다. 그러나 학습의 초창기에는 저장된 데이터가 많지 않아 좋은 선택을 못하게 된다.

우리말이나 상대방 말은 공격 또는 방어를 하게 되고, 서로 누가 먼저 상대방 말을 포획하는지가 승부의 관건이 된다. 상대방 말은 우리말 방향으로 움직여 우리말을 공략하게 되고, 우리말은 [식 1]의 결과로 주어지는, 학습된 보상 점수에 의해 가장 좋은 방향을 선택하여 이동한다.

[표 1] 상대방 간 포획의 경우의 보상점수

공격주체	대 상	보상점수
상대방말 1, 2	우리 말1	-100
상대방말 1, 2	우리 말2	-100
우리 말 1, 2	상대방 말1	100
우리 말 1, 2	상대방 말2	100

우리말이 하나의 상대방 말을 포획한 경우에는 100의 보상이 주어진다. 상대방 말이 하나의 우리 말을 포획한 경우에는 -100의 보상이 주어진다.

상대방 말의 경우, 처음에는 학습이 이루어지지 않은 우리말을 쉽게 포획하게 된다. 우리말의 학습이 이루어지면서 부터는 우리말도 상대방 말을 포획하게 되고, 최종적으로는 상대방 말이 전혀 우리 말을 포획 할 수 없게 된다.

우리말의 경우, 처음에는 학습이 이루어지지 않았기 때문에 우리말이 상대방 말을 포획하기는 어렵다. 그러나 학습이 이루어진 후에는 상대방 말을 항상 쉽게 포획하여 승리하게 된다.

```

procedure Q_Learning
{
  Find Max from QTable
  Select player character action from QTable
  generate random
  Move player character
  Move enemy character from e_action()
  if (catch)
    Generate plus reward
  if (be captured)
    Generate minus reward

  Update QTable
}
    
```

[그림 3] Q-learning 알고리즘

[식 1]에서 주어진 공식을 이용하여 우리말의 최적의 행동을 찾는다. 4방향으로의 행동을 위한 값이 동일하다면, 난수를 발생시켜 임의의 방향을 선택한다. 주어진 방향으로 우리말을 이동한다. 상대방 말이 우리말을 공격하기 위해 이동한다.

우리말이 상대방 말을 포획하였을 경우 양의 보상점수를 할당한다. 상대방 말이 우리말을 포획하

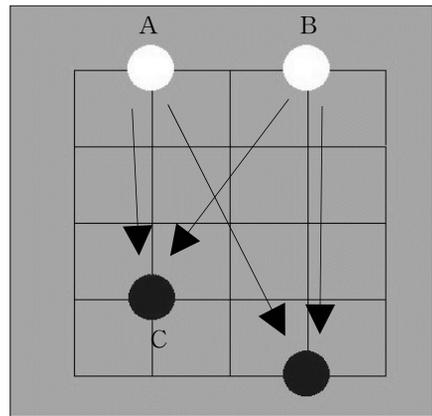
였을 경우 음의 보상점수를 발생시킨다. 보상점수를 [식 1]을 이용하여 저장한다.

### 3.2 제안하는 알고리즘

강화학습에서의 상태정보는 계속되는 게임의 결과에 따라 보상 값들이 상태정보에 누적되게 되어 있다. 게임이 어느 정도 진행되면 [식 1]로 인해 누적된 보상 값에 의해 우리말에 가장 유리한 값을 선택할 수 있도록 되어 있다.

게임의 초창기에는 0으로 초기화하여 모두 같은 값을 갖고 있다. 그리고 게임이 어느 정도 진행되지 않은 초창기에는 아직 보상 값이 많이 누적되지 않아 동일한 값들을 많이 가지고 있다. 이 때 보통 무작위로 그중 하나의 값을 선택하여 게임을 진행하게 된다. 이 때 적절한 위치로 이동하지 못하여 상대방 말에게 포획되거나 엉뚱한 방향으로 움직여 학습이 잘 이루어지지 않는다.

본 논문에서는 이러한 단점을 없애고자, 동일한 값들이 추출될 때, 거리정보를 고려하여 유리한 방향으로 움직이도록 Q-learning 알고리즘을 개선하였다.



[그림 4] 거리를 고려한 개선된 방법

[그림 4]에서와 같은 상황이 있다고 하자. 현재 게임의 초창기라고 가정하여 보상 값이 많이 누적되지 않아 동일한 0의 값들을 갖고 있다. [그림 3]

에서와 같이 QTable에 저장된 보상 값 중 최적의 값을 선택하여야 한다. 그러나 현재는 같다고 가정한다. 이런 경우 [그림 3]에서는 난수(Random Number)를 발생시켜 나아갈 길을 선택한다. 그러나 [그림 5]에서는 아래와 같은 방법으로 최적의 선택을 하게 된다. 선택된 곳으로 갔을 때에 승패가 발생하게 되어 보상 값이 주어진다. 새로운 보상 값을 감안하여 [그림3,5]와 같이 QTable을 갱신한다.

A와 B는 우리말이며, C와 D는 상대방 말이다.

본 논문에서는 다음과 같은 방법으로 거리를 고려한 상대방말 공략 방법을 제안하였다. 첫째, C와 D중 공략할 말을 선택하기 위해 A와 B에서 가까운 말을 선택한다. 그러므로 C가 선택된다. 둘째, A와 B에서 C로 가는 거리가 먼 B를 선택한다. 셋째, B에서 C로 가는 길 중에서 가로와 세로에서 거리가 짧은 가로방향을 선택한다.

우리말인 A와 B의 이동방향을 선택 할 때, 상대방말에 무조건 가까이 가는 것이 능사가 아니다. 본게임의 경우 두개의 말이 상대방을 동시에 공격하는 구조이므로 무조건 먼저 접근할 경우 오히려 상대방말의 공격을 당할 수 있다. 그러므로 전체적으로는 가까이 가면서 하나의 단위말의 입장에서는 거리를 두면서 접근하는 것이 효율적이다.

```

procedure Q_Learning
{
  Find Max from QTable
  Select player character action from QTable
  if (all player action value is Equal)
    if (total value is shorter than)
      if (one of two characters is longer than)
        if (one of two sides is shorter than)
          move to opponent

  Move player character
  Move enemy character from e_action()
  if (catch)
    Generate plus reward
  if (be captured)
    Generate minus reward
  Update QTable
}
    
```

[그림 5] 제안하는 알고리즘

[식 1]에서 주어진 공식을 이용하여 우리말의 최적의 행동을 찾는다. 주어진 방향으로 우리말을 이동한다. 상대방 말이 우리말을 공격하기 위해 이동한다.

우리말이 상대방 말을 포획하였을 경우 양의 보상점수를 할당한다. 상대방 말이 우리말을 포획하였을 경우 음의 보상점수를 발생시킨다. 보상점수를 [식 1]을 이용하여 저장한다.

이 방법을 사용함으로써 생기는 알고리즘 사용시간은 5500 개의 학습 횟수를 기준으로 하였을 때, 기존 방법과의 알고리즘 수행시의 시간차는 20분의 1초 정도로 미미함으로, 큰 오버헤드는 없다고 본다.

#### 4. 실험 및 결과

실험은 우리말이 둘, 상대방 말이 둘일 때로 구분하여 실험하였다.

하나의 상태공간은  $5 \times 5$  이므로 25개의 격자셀로 구성된다. 우리말의 관점에서 볼 때 고려하여야 할 요소는 다음과 같다.

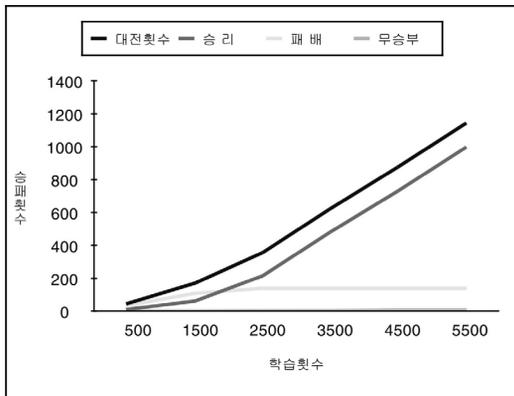
- ㉞ 우리말들의 위치
- ㉟ 상대방 말들의 위치
- ㊱ 우리말들이 공격 또는 방어를 위해 이동 가능한 공간

우리말의 상태공간은  $25 \times 25$  이고 상대방말의 경우도 같다. 나머지 고려하여야 할 요소는 8방향이 된다. 그러므로 상태공간의 크기는  $25 \times 25 \times 25 \times 25 \times 8$ 로 3,125,000 이 된다.

우리말은 강화학습 되었으므로 현재의 상황에서 최상의 지점으로 이동하도록 설계되었으며, 상대방 말은 무조건적인 공격을 하도록 프로그램 되어 있다.

[표 2] 강화학습에서의 실험결과

학습횟수 \ 승패횟수	대전횟수	승리	패배	무승부
500	48	11	37	0
1,500	171	62	109	0
2,500	356	215	139	2
3,500	622	480	139	3
4,500	875	729	139	7
5,500	1138	992	139	7



[그림 6] 강화학습에서의 실험결과

[표 2]에서 보면 알 수 있듯이 강화학습을 하는 말과 일방적으로 공격위주로 움직이는 말은 차이가 있음을 알 수 있다.

강화학습을 하는 경우에 학습속도측면에서 본다면 아군 말은 적군 말의 공격에 2500회에서 139회 이상 포획당한 후에는 더 이상 포획당하지 않는다는 것을 알 수 있다.

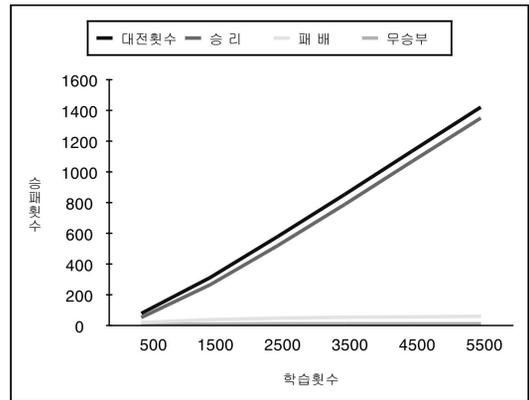
그 이후로는 표와 같이 포획횟수는 그대로이므로, 단 한 번도 더 이상 포획이 추가되지 않음을 알 수 있다.

공격위주의 말과 강화학습 된 말과의 포획횟수를 비교해 보면 초반 500 회전에서는 공격위주의 말이 서너 배로 포획하는 것을 볼 수 있다. 1500회에서는 차이가 줄어들고, 2500회에서는 강화 학습된 말의 포획횟수가 상대방 말보다 앞질러나가는 것을 볼 수 있으며, 5500회에서는 큰 차이를 벌리는 것을 알 수 있다.

이상에서 알 수 있듯이, 중요한 점은 계속 실험을 할 경우에, 상대방 말의 포획횟수는 정지된 채, 강화 학습된 말의 포획횟수가 차이를 벌리리라 유추된다.

[표 3] 개선된 강화학습에서의 실험결과

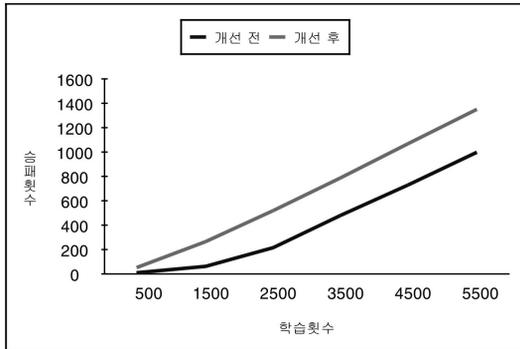
학습횟수 \ 승패횟수	대전횟수	승리	패배	무승부
500	84	57	21	6
1,500	312	265	38	9
2,500	578	519	48	11
3,500	853	788	54	11
4,500	1135	1068	56	11
5,500	1415	1344	60	11



[그림 7] 개선된 강화학습에서의 실험결과

강화학습을 하는 경우에 학습속도측면에서 본다면 우리말은 500회에서부터 상대방말을 승리횟수에서 앞서는 것을 알 수 있다. 그 이후로도 [표 3]과 같이 승리횟수는 큰 차이를 보이며 늘어나는 것을 알 수 있다.

공격위주의 말과 학습된 말은 움직이는 패턴 자체가 다르다. 강화학습된 말은 모든 상대공간에서의 학습이 완료되었고, 일방공격형 말은 처음이나 마지막이나 똑같은 방식의 공격을 수행한다.



[그림 8] 강화학습과 개선된 강화학습에서의 비교결과

[그림 8]에서 개선하기 전의 강화학습과 개선된 후의 강화학습의 실험결과를 직접적으로 비교해 보았다. 개선전의 실험결과가 처음에 완만한 곡선을 그리는 반면에, 개선후의 결과는 빠르게 학습하는 것을 볼 수 있다.

## 5. 결 론

3D 게임과 온라인게임이 완전히 자리 잡은 요즘, 게임시나리오가 게임의 흥미를 이끌어내지만 캐릭터의 자동화문제는 여전히 남은 숙제라고 할 수 있다. 캐릭터의 자동화로 인해 게임의 재미가 더할 수 있고, 그것은 게임프로그램머의 몫이다.

그 동안 여러 가지 인공지능 알고리즘이 연구되고, 사용되어왔지만 강화학습은 보드게임분야에서 많이 연구되지 않았다.

본 논문에서는 강화학습 알고리즘을 이용하여 우리말과 상대방 말이 대국하는 상황에서 상대방 말이 우리말을 공격할 때 음의 보상 값을 부여하고 우리말이 상대방 말을 공격할 때 양의 보상 값을 부여하여, 우리말이 학습하게 하여 지능적으로 움직이게 하였다.

일반적인 강화학습에서는 일정기간 보상 값이 쌓이기 전에는 정확한 값을 산출하기가 어렵다. 그러므로 일정기간 동안은 동일한 값이 산출되었을 때 무작위 난수 값을 발생시켜 평균적으로 균등하

게 순서를 부여한다. 그러나 본 논문에서는 이러한 단점을 개선하기 위하여 동일 값이 산출될 때, 우리말과 상대방 말의 거리를 고려한 줄고누 문제 영역 지식을 활용한 휴리스틱을 사용해 학습의 속도향상을 시도하였다. 그 결과 학습속도가 향상됨을 알 수 있었다.

구현된 우리말이 지능적으로 잘 움직이는지 확인하기 위해, 보드게임을 제작하여 강화학습으로 움직이는 우리말과 개선된 강화학습을 적용한 우리말을 비교하였다. 실험결과 강화학습한 말보다 개선된 말이 더 많은 학습속도 향상을 보임을 알 수 있었다.

## 참고문헌

- [1] 한국게임산업진흥원, “2008 대한민국 게임백서”, 2009
- [2] 신용우, “강화학습을 이용한 지능형 게임캐릭터의 제어”, 인터넷정보학회논문지, 제8권 5호, pp 91-97, 2007
- [3] Richard Sutton, Andrew G. Barto, “Reinforcement Learning : An Introduction”, MIT Press, Cambridge, MA, 1998.
- [4] Imran Ghory, “Reinforcement learning in board games.”, available at <http://www.cs.bris.ac.uk/Publications/Papers/2000100.pdf>, 2004.
- [5] Nee Jan van Eck, Michiel van Wezel., “Reinforcement Learning and its Application to Othello”, available at <http://www.few.eur.nl/few/people/mvanwezel/rl.othello.ejor.pdf>, 2004
- [6] Steve Woodcock, “Game AI : The State of the Industry”, Game Developer Magazine, 2000.
- [7] 심우성감수, “전통놀이 50선”, 농협, 1996
- [8] 심우성, “우리나라 민속놀이”, 동문선, 1996



신 용 우 (Yong-Woo Shin)

2004년 경희대학교 지능시스템전공 (박사수료)  
1990년-2000년 프리랜서게임프로그래머, LG데이콤  
2000년-현재 동아방송예술대학 게임애니메이션계열 교수  
2010년-현재 DIMA웹스 대표

관심분야 : 게임제작, 인공지능, 디지털영상

---



정 태 충  
(Tae-Choong Chung)

1980년 서울대학교 전자공학과 (학사)  
1982년 한국과학기술원 전자공학전공 (공학석사)  
1987년 한국과학기술원 전자공학전공 (공학박사)  
1987년-1988년 KIST 시스템공학센터 선임연구원  
1988년-현재 경희대학교 컴퓨터공학과 교수

관심분야 : 기계학습, 보안, 최적화, 에이전트

---