강화학습을 이용한 줄고누게임의 인공엔진개발

Artificial Engine Development through Reinforcement Learning on Jul-Gonu Game

신 용 우*

Yong-Woo Shin

요 약

게임프로그램 제작이 단순히 3D 또는 온라인게임 등으로 분류하여 엔진과 게임프로그래밍을 하던 시기를 지나 이제는 게임프로그래밍의 종류를 세분화하여 인공지능 게임프로그래머의 역할이 게임을 좀 더 재미있게 할 수 있는 시점이라 하겠다.

본 논문에서는 강화학습 알고리즘을 이용하여 보상 값을 받아 줄고누 보드게임 말이 학습하게 하여 지능적으로 움직이게 하였다. 구현된 게임 말이 지능적으로 잘 움직이는지 확인하기위해, 보드게임을 제작하여 상대방 말과 승부를 하게 하였다. 실험결과 일정횟수 학습한 이후, 임의로 움직이는 말보다 성능이 월등히 향상됨을 알 수 있었다.

Abstract

Game program manufacture had been classed by 3D or on-line game etc. simply. But, atomized game programmer's kind now. So, Artificial Intelligence game programmer's role is important.

This paper used reinforcement learning algorithm for Jul_Gonu board characters to learn, and so they can move intelligently. To compare a learned character to an random one, a board game was created, and then they fought against each other. As a result, learned character's ability was far more improved.

☞ keyword : Reinforcement Learning, Artificial Intelligence, Board Game, Q Learning

1. Introduction

온라인게임의 보급이 많아지고 장르별로도 다양하게 롤플레잉이나 시뮬레이션 장르뿐이 아니라, 캐쥬얼게임과 스포츠게임등 다양한 장르의 게임이 보급되고 있다.또한 3D 게임의 제작 또한 많아지고 있다. 과거에는 엔진과 게임프로그램으로 나누어 개발하던 시점에서 이제는 좀 더 세분화하여 분야별 전문프로그래머가 필요하게 되었다.

인공지능 분야에서도 현재까지 캐릭터의 이동처리를 위해서는 패턴(Pattern), A* 알고리즘이나 FSM, 퍼지(Fuzzy), FuSM 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 다양화 시키는 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 FSM 을 이용한 캐릭터의 이동방법은 경우의 수가 작게 제한되어 단순하고 보다

* 정 회 원 : 동아 방송예술대학 게임애니메이션계열 교수
ywshin@dima.ac.kr
[2008/07/17 투고 - 2008/07/24 심사 - 2008/10/16 심사완료]

많은 경우의 수를 가지는 캐릭터의 전투상황에서 효율적인 전투를 벌이지는 못한다. 그러므로 학습 알고리즘을 적용하여야 한다.

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

강화학습 분야에는 상태공간의 문제, 캐릭터의 지능화 등의 문제가 있는데, 상태공간의 효율적인 사용 등의 문제는 여러 논문에서 다루었다. 그러므로 본 논문에서는 캐릭터의 자동화 부분을 다루고자 한다. 기존의 캐릭터의 자동화를 다룬 논문들은 서양보드게임인 오델로, 틱택토 등을 다루었으나[2][3]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, 줄고누보드게임의 경우 말의 위치가 계속 바뀐다는 점에서 전략적인 움직임이 필요하다.

본 논문에서는 우리말과 상대방 말과의 대결상황에서 우리말이 상대방말을 효과적으로 공격할 수 있도록 학습시킨다. 보드게임을 제작하여 일정한 패턴으로 움직이는 상대방 말을 상대로 학습을 하여, 일정한 횟수의 학습에 다다랐을 때 스스로 상대방 말을 지능적으로 상대하게 된다. 임의의 방향으로 움직이는 우리말 비교할 때 월등하게 좋은 학습효과를 알 수 있다.

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 보드게임 (Board Game)

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

보드게임이란 상대방 캐릭터를 제압하는 형태의 게임으로 바둑, 장기, 체스, 오목, 오델로 등이 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.

2.2 게임 인공지능의 개요

사람과 컴퓨터가 상호작용하여 플레이하는 게임에는 어디에나 인공지능이 존재하고 있다.

우리나라에서 아주 유명했던 스타크래프트라는 전략게임에서 캐릭터들의 길을 찾기 위해 사용되었던 A* 길 찾기 알고리즘은 장애물을 피해 원하는 목적지로 효율적인 방식으로 캐릭터들을 이동시키는 길을 생성하였다.

비교적 단순한 게임에도 인공지능은 사용되고 있다. 상대캐릭터의 이동이나 공격, 방어 등의 행위에 따라 아군캐릭터의 행위를 제어하는 유한상태기계(FSM)는 모든 게임에서 광범위하게 사용되고 쉽게 구현가능하다.

인간의 일상생활을 게임화 하였던 심즈는 인공생명 알고리즘을 게임에 사용하였다.

많은 몬스터들을 같은 방향으로 움직이는 경우에 플로킹(Flocking) 알고리즘을 사용한다.

게임의 재미를 배가하기 위해서 좀 더 다양한 값들을 선택할 수 있는 퍼지(Fuzzy) 나 퍼지와 유한상태기계의 장점을 결합한 퍼지상태기

계(FuSM) 를 사용할 수도 있다.

이외에도 캐릭터 자체의 능력을 향상시킬 수 있도록 학습시키는 신경망이 있다. 그러나 신경망은 학습시키는 데에 많은 시간이 필요하다는 단점이 있다.

강화학습은 온라인 학습으로 캐릭터의 능력을 학습시키는 데에 적합하다. 또한 아직까지 강화학습을 게임에 적용한 논문은 많지 않다.

본 논문에서는 강화학습을 이용하여 보드게임에서 캐릭터의 지능이 향상되도록 하였다.

3. 지능형 보드게임의 구현

본 논문에서는 상대방 말과의 대결을 통해 스스로 학습하는 우리말을 구현한다.

3.1 줄고누 게임 알고리즘

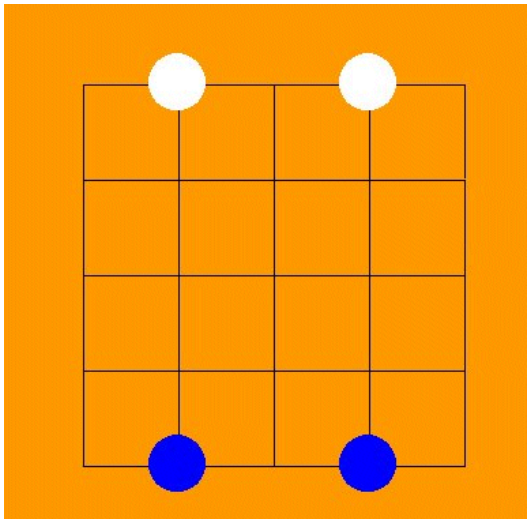
캐릭터들은 크게 우리말과 상대방 말이 있다. 우리말이 둘 존재한다. 상대방 말이 둘 존재한다.

우리말은 상대방 말을 공격하거나 방어하게 된다.

공격은 상대방 말 옆에 우리말 둘이 존재하면 상대방 말을 포획하게 된다. 방어란 상대방 말 둘 옆에 놓이지 않도록 하면 된다. 게임 진행방법은 한 번에 한 칸씩 직선으로만 이동하게 되어있다.

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 이동하도록 되어있다.

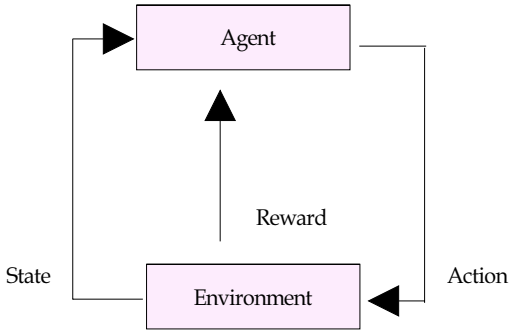
또한 실수로 상대방 말의 일직선상에 들어갔을 때는 상대방말에게 포획 당하도록 되어 있다. 상대방 말의 경우에도 실수로 아군말의 일직선상에 들어오면 포획 당하게 되어있다.



(그림. 1) (아군캐릭터와 적군캐릭터의 초기상태)

3.2 강화학습

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



(그림 2) (강화학습의 모델)

환경에는 목적달성에 필요하거나, 필요하지 않은 많은 상태들이 존재한다. 에이전트는 각각의 상태를 경험하여 목적달성의 경우 환경으로

부터 보상을 받게 된다. 목적달성을 할 수 없는 상태인 경우 보상을 받을 수 없다. 그러므로 많은 시행착오를 겪을 수 있으며, 모든 상태를 경험해 보아야 한다.

강화학습알고리즘으로서 일반적으로 많이 사용되는 Q-learning 에 대해 알아보자.

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

특정상태 s 에서의 행위 a 에 대한 보상 r 을 얻을 수 있도록 식이 주어져 있다. t 는 이전, $t+1$ 은 이후시간단위를 말한다. $\max$란 최대값을 구함이고, α는 학습속도매개변수로서 0에서 1 사이의 범위이다. γ는 할인계수로서 0에서 1 사이의 범위이다.

모든 가능한 상태공간에서의 특정상태 s 에서의 행위 중 가장 좋은 보상 r 값을 저장하고 산출한다. 처음에는 모든 상태에 대하여 보상을 얻기 위해 무작위로 행위를 하여 경험을 쌓게 된다. 그러나 시간이 지나 경험한 상태가 많아질수록 보상을 토대로 불필요한 상태의 경험을 필요치 않는다. 거의 모든 상태를 경험함으로서 학습이 완료되어 지능적으로 동작하게 된다.

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 학습된 보상점수에 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 100 의 보상이 주어진다. 상대방 말이 하나의 우리말을 포획한 경우에는 -100 의 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 우리말을 포획 할 수 없게 된다.

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

4. 실험 및 결과

실험은 우리말이 둘, 상대방 말 이 둘일 때로 구분하여 실험하였다.

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

㉮ 우리말들의 위치
㉯ 상대방 말들의 위치
㉰ 우리말들이 공격 또는 방어를 위해 이동 가능한 공간

우리말의 상태공간은 25 × 25 이고 상대방말의 경우도 같다. 나머지 고려하여야 할 요소는 8방향이 된다. 그러므로 상태공간의 크기는 25

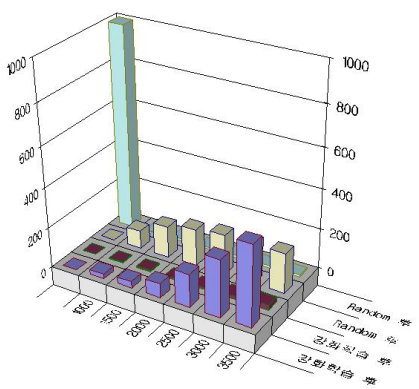
× 25 × 25 × 25 × 8 로 3,125,000 이 된다.

우리말은 강화학습 되었으므로 현재의 상황에서 최상의 지점으로 이동하도록 설계되었으며, 상대방 말은 무조건적인 공격을 하도록 프로그램 되어 있다. 두 가지 경우로 실험하였다. 첫째, 상대방 말이 공격할 수 있는 경우의 수 중 임의로 하나를 택하여 공격하는 경우이다. 둘째, 상대방 말이 공격할 수 있는 경우의 수 중 최단거리를 가지는 말이 공격하는 경우이다. e_action() 은 우리말을 공격하도록 상대방 말을 움직이는 알고리즘이다. Q_table 은 보상 값을 담고 있는 상태공간을 저장한 기억장소를 의미한다.

```
procedure Q_Learning
{
        Find Max from Q_Table with Previous State.
        Select player character action from QTable
        Move player character
        Move enemy character from e_action()

        if (catch)
            Generate plus reward
        if (be captured)
            Generate minus reward
        Find Max from Q_Table with Current State.
        Select player character action from QTable
        Multiply gamma and max and add reward.
        Update QTable with Previous State.
}
```

(그림3) (제안하는 알고리즘)



(그림 4) (우리말과 상대방 말의 실험결과)

(표 2) 우리말과 상대방 말의 실험결과

승패 횟수 / 학습 횟수	강화학습 후 (우리 말)		Random 후 (상대방 말)	
	승	승	패	패
1,000	25	6	91	3
1,500	33	7	155	10
2,000	71	8	184	10
2,500	166	10	200	10
3,000	284	16	200	10
3,500	404	25	200	10

(표 3) (우리말과 최단거리 상대방 말의 실험결과)

승패 횟수 / 학습 횟수	강화학습 후 (우리 말)		최단거리 후 (상대방 말)	
	승	승	패	패
1,000	275	5	14	0
1,500	441	5	14	0
2,000	608	5	14	0
2,500	775	5	14	0
3,000	941	5	14	0
3,500	1108	5	14	0

차트와 그림에서 보면 알 수 있듯이 강화학습을 하는 말과 일방적으로 공격위주로 움직이는 말이 차이가 있음을 알 수 있다.

강화학습이나 일방공격형 말에서 각각 두 가지의 숫자가 나온 것은 직접 포획한 경우와 스스로 포획된 경우의 차이이다. (표 2)에서 강화학습을 하는 경우에 승률을 본다면 아군 말은 적군 말의 공격에 2500회에서 200회 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 1,000 회전에서는 공격위주의 말이 서너 배로 포획하는 것을 볼 수 있

다. 2000회에서는 두 배 정도로 차이가 줄어들고, 2500회를 기점으로 강화 학습된 말이 더 이상 적에게 포획당하지 않는 것을 알 수 있다. 3000회에서는 강화 학습된 말의 포획횟수가 상대방 말보다 앞질러나가는 것을 볼 수 있으며, 3500회에서는 두 배 이상 차이를 벌리는 것을 알 수 있다.

(표 3) 에서는 상대방 말이 공격하는 경우에, 최단거리를 유지하는 말이 공격하는 경우를 실험하였다. (표2) 에서의 경우보다 아군에게는 유리하게 작용하고, 상대방에게는 불리하게 작용하는 것을 알 수 있다. 이런 점으로 볼 때, 학습 없이 무조건 근거리에서 공격하는 것이 능사가 아님을 알 수 있다.

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

5. 결 론

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

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

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

구현된 우리말이 지능적으로 잘 움직이는지 확인하기위해, 보드게임을 제작하여 강화학습으로 움직이는 우리말과 일방 공격형 상대방 말이 대결을 하게 하였다.

실험결과 일방적인 공격형 말보다 강화 학습한 우리말이 상대방 말을 더 많이 포획하는 것을 알 수 있었다.

또한 일정한 수의 포획횟수가 일어난 후, 더 이상 상대방 말은 우리말을 포획할 수 없었다. 우리말의 일방적인 승리가 이어진다.

참 고 문 헌

[1] Richard Sutton, Andrew G. Barto, "Reinforcement Learning :An Introduction", MIT Press, Cambridge, MA, 1998.

[2] Imran Ghory, "Reinforcement learning in board games.", available at http://www.cs.bris.ac.uk/Publications/Papers/2000100.pdf, 2004.

[3] Nee Jan van Eck, Michiel van Wezel., "Reinforcement Learning and itsApplication to Othello", available at http://www.few.eur.nl/few/people/mvanwezel/rl.othello.ejor.pdf, 2004

[4] Steve Woodcock, "Game AI : The State of the Industry", Game Developer Magazine, 2000.

[5] Steve Rabin, AI Game Programming Wisdom 2, Charles River Media, 2003

[6] Steve Rabin, AI Game Programming Wisdom, Charles River Media, 2002

[7] 신용우, 게임프로그래밍 길잡이, 대림출판, 2002

[8] 신용우, 인공지능 게임프로그래밍, 대림출판, 2004

[9] Andrew Kirmse, Game Programming Gems 4, Delmar Thomson Learning, 2004.

[10] Mark Deloura, Game Programming Gems 3, Charles River Media, 2002.

[11] Mark Deloura, Game Programming Gems 2, Charles River Media, 2001.

◐ 저 자 소 개 ◑

신 용 우(Yong-Woo Shin)
2004년 경희대학교 컴퓨터공학과 지능시스템전공 (박사수료)
1990년-1993년 (주)진도 전산부
1994년-1995년 프리랜서 게임프로그래머
1996년-2000년 LG데이콤 인터넷사업단
2000년-현재 동아방송예술대학 게임애니메이션계열 교수
현재 한국게임학회 이사
현재 게임물등급위원회 자문위원
관심분야 : 게임프로그래밍, 데이터마이닝, 인공지능
E-mail : ywshin@dima.ac.kr