개선된 강화학습을 이용한 줄고누게임의 학습속도개선

An improvement of the learning speed through Improved Reinforcement Learning on Jul-Gonu Game

신 용 우* 정 태 충**
Yong-Woo Shin Tae-Choong Chung

요 약

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

본 논문에서는 강화학습 알고리즘을 이용하였다. 보상 값을 받아 보드게임 말이 학습하게 하여 지능적으로 움직이게 하였다. 학습 도중에 동일한 최선 값이 있을 때, 줄고누 문제 영역 지식을 활용한 휴리스틱을 사용해 학습의 속도 향상을 시도하였다. 단순 구현된 말과 개선 구현된 말을 비교하기 위해 보드게임을 제작하였다. 그래서 일방공격형 말과 승부를 하게 하였다. 실험결과 개선 구현된 말의 성능이 학습속도 측면에서 월등히 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.

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

☞ keyword : Reinforcement Learning, Jul-Gonu, Gonu, Learning Speed, Game 강화학습, 줄고누, 고누, 학습속도, 게임

1. 서 론

온라인게임의 보급이 많아지고 장르별로도 다양하게 롤플레잉이나 시뮬레이션 장르뿐이 아니라, 캐쥬얼게임 과 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. FSM 은 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.

* 정 회 원 : 동아방송예술대학 게임애니메이션계열 교수
ywshin@dima.ac.kr
** 정 회 원 : 경희대학교 컴퓨터공학과 교수 (교신저자)
tcchung@khu.ac.kr
[2008/09/09 투고 - 2008/09/22 심사 - 2008/11/26 심사완료]

패턴이나 FSM 을 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 유사알고리즘보다 비교적 짧은 시간에 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 자동화부분을 다루고자 한다. 기존의 캐릭터의 자동화를 다룬 논문들은 서양보드게임인 오델로, 틱택토 등을 다루었으나 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, 줄고누 보드게임의 경우 말의 위치가 계속 바뀐다는 점에서 전략적인 움직임이 필요하다.

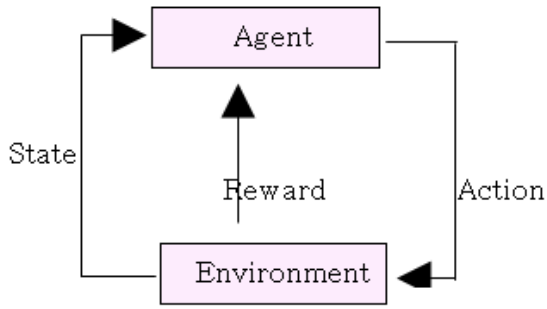
본 논문에서는 우리말과 상대방 말과의 대결상황에서 두 가지의 경우로 보드게임을 구현하여 비교하였다. 첫째, 강화학습을 이용하여 보드게임을 단순 구현하였다. 둘째, 강화학습에서의 최선값을 산출하는 부분에서 동일한 값이 나올 때 좀 더 유리한 값을 선택하도록 하였다. 두 가지 경우를 일정한 패턴으로 움직이는 말과 대국하게 하였다. 실험 결과 단순 강화학습을 적용한 말보다 개선된 강화학습을 적용한 말이 우수한 경기를 벌이는 것을 알 수 있었다.

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.

강화학습알고리즘으로서 일반적으로 많이 사용되는 Q-learning 에 대해 알아보자.

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

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, α 는 학습속도매개변수로서 0에서 1 사이의 범위이다. γ 는 할인계수로서 0에서 1 사이의 범위이다.

모든 가능한 상태공간에서의 특정상태 s 에서의 행위 중 가장 좋은 보상 r 값을 저장하고 산출

한다. 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.

2.2 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, 오델로등이 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 [5].

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 따 먹거나 잡아 가둠으로써 승부를 겨루는 놀이이다.

고누의 종류는 지방에 따라 여러 가지 특징이 있으나 대개 우물고누, 줄고누, 밭고누, 곤질고누, 참고누, 자동차고누, 호박고누, 패랭이고누, 팔팔고누, 포위고누, 장수고누, 왕고누로 구분해 볼 수 있다 [6].

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

3. 지능형 보드게임의 구현

본 논문에서는 상대방 말과의 대결을 통해 스스로 학습하는 우리말을 구현한다.

3.1 캐릭터의 설계

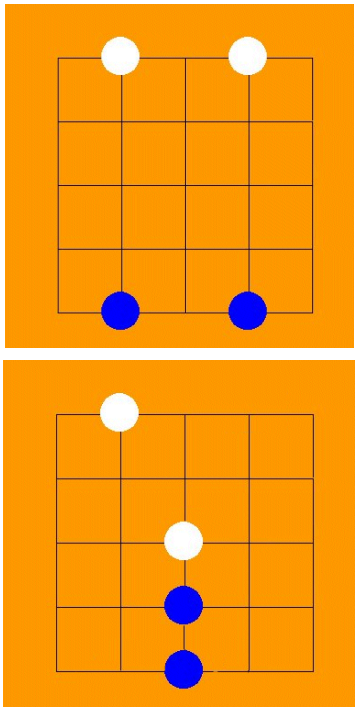
캐릭터들은 크게 우리말과 상대방 말이 있다. 우리말이 둘 존재한다. 상대방 말이 둘 존재한다.

우리말은 상대방 말을 공격하거나 방어하게 된다.

공격은 우리말 둘이 일직선으로 나열되었을 때 그 끝에 상대방 말이 존재하면 상대방 말을 포획하게 된다. 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 이동하도록 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 종료상태

각각의 우리말과 상대방 말의 위치는 메모리에 각각 상황별로 저장되며 각각의 상황에서의 학습 점수가 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.

```
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) 단순 Q-learning 알고리즘

3.2 제안하는 알고리즘

강화학습에서의 상태정보는 계속되는 게임의 결과에 따라 보상 값들이 상태정보에 누적되게 되어있다. 게임이 어느 정도 진행되면 누적된 값에 의해 우리말에 가장 유리한 값을 선택할 수 있도록 되어있다.

게임의 초창기에는 0으로 초기화하여 모두 같은 값을 갖고 있다. 그리고 게임이 어느 정도 진행되지 않은 초창기에는 아직 보상 값이 많이 누

적되지 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 우리말이 근접해 있을 때 우리말들이 일렬로 모여 공격태세를 취할 수 있고 엉뚱한 방향으로 움직이지 않도록 Q 러닝 알고리즘을 개선하였다.

```
procedure Q_Learning
{
        Find Max from QTable
        Select player character action from
        QTable
        if (all player action is Equal)
          if (two character is each other near)

            if (White_y[0] > White_y[1])
                        move to up
            if (White_y[0] < White_y[1])
                        move to down
            if (White_x[0] > White_x[1])
                        move to left
            if (White_x[0] < White_x[1])
                        move to right
            if (no action)
                        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 4) 제안하는 알고리즘

4. 실험 및 결과

실험은 우리말이 둘, 상대방 말 이 둘일 때로 구분하여 실험하였다.

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

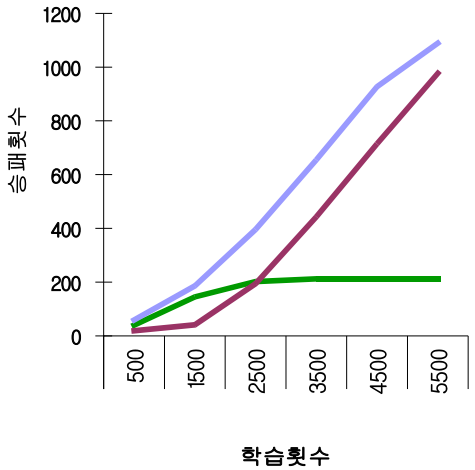
㉮ 우리말들의 위치
㉯ 상대방 말들의 위치
㉰ 우리말들이 공격 또는 방어를 위해 이동 가능한 공간

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 프로그램되어 있다.

(표 2) 단순강화학습에서의 실험결과

승패 횟수 / 학습 횟수	대전 횟수	승 리	패 배
500	59	19	40
1,500	186	41	145
2,500	396	194	202
3,500	655	443	212
4,500	927	715	212
5,500	1089	977	212



(그림 5) 단순강화학습에서의 실험결과

(표2) 에서 보면 알 수 있듯이 강화학습을 하는 말과 일방적으로 공격위주로 움직이는 말은 차이

가 있음을 알 수 있다.

강화학습을 하는 경우에 학습속도측면 에서 본다면 아군 말은 적군 말의 공격에 3500회에서 212회 이상 포획당한 후에는 더 이상 포획당하지 않는다는 것을 알 수 있다.

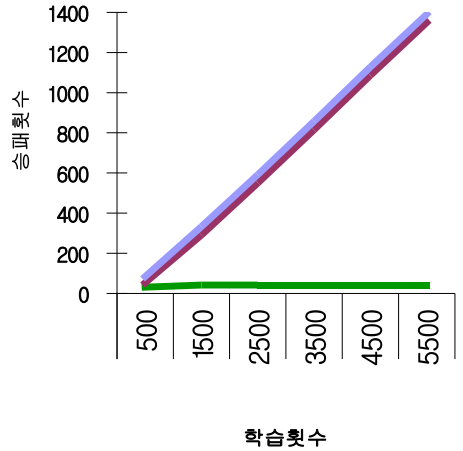
그 이후로는 표와 같이 포획횟수는 그대로이므로, 단 한 번도 더 이상 포획이 추가되지 않음을 알 수 있다.

공격위주의 말과 강화학습 된 말과의 포획횟수를 비교해 보면 초반 1,000 회전에서는 공격위주의 말이 서너 배로 포획하는 것을 볼 수 있다. 2500회에서는 차이가 줄어들고, 3500회에서는 강화 학습된 말의 포획횟수가 상대방 말보다 앞질러나가는 것을 볼 수 있으며, 4500회에서는 세 배 이상 차이를 벌리는 것을 알 수 있다.

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

(표 3) 개선된 강화학습에서의 실험결과

승패 횟수 / 학습 횟수	대전 횟수	승 리	패 배
500	84	53	31
1,500	333	291	42
2,500	592	550	42
3,500	857	815	42
4,500	1128	1086	42
5,500	1391	1349	42



(그림 6) 개선된 강화학습에서의 실험결과

(표3) 에서 보면 알 수 있듯이 강화학습을 하는 말과 일방적으로 공격위주로 움직이는 말은 차이가 있음을 알 수 있다.

강화학습을 하는 경우에 학습속도측면 에서 본다면 아군 말은 적군 말의 공격에 1500회에서 42회 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 학습된 말은 움직이는 패턴자체가 다르다. 강화학습된 말은 모든 상태공간에서의 학습이 완료되었고, 승 아니면 패로 구분 짓는 상황에서 일정횟수 이상에서 항상 이기는 것은 당연하다.

공격위주의 말과 강화학습 된 말과의 포획횟수를 비교해 보면 초반 500 회전부터 개선된 강화학습의 말이 약간 더 많이 포획하는 것을 볼 수 있다. 그러나 곧 일방공격형 말의 승리는 정지되고 2500회에서는 10배 이상 차이가 나고 회를 거듭할수록 계속적으로 차이가 남을 알 수 있다.

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 보드게임분야에서 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.

일반적인 강화학습에서는 일정기간 보상 값이

쌓이기 전에는 정확한 값을 산출하기가 어렵다. 그러므로 일정기간 동안은 동일한 값이 산출되었을 때 무작위 난수 값을 발생시켜 평균적으로 균등하게 순서를 부여한다. 그러나 본 논문에서는 이러한 단점을 개선하기 위하여 동일 값이 산출될 때, 줄고누 문제 영역 지식을 활용한 휴리스틱을 사용해 학습의 속도향상을 시도하였다. 그 결과 학습속도가 향상됨을 알 수 있었다.

구현된 우리말이 지능적으로 잘 움직이는지 확인하기위해, 보드게임을 제작하여 단순강화학습으로 움직이는 우리말과 개선된 강화학습을 적용한 우리말을 비교하였다.

실험결과 단순 강화학습한 말보다 개선된 말이 더 많은 학습속도 향상을 보임을 알 수 있었다.

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 its Application 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] 심우성감수, "전통놀이 50선", 농협, 1996

[6] 심우성, "우리나라 민속놀이", 동문선, 1996

[7] 신용우, "지능형 고누게임에 관한 연구", 석사학위논문, 경희대학교, 1998

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

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

◐ 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

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
E-mail : tcchung@khu.ac.kr