

유전 알고리즘과 신경망을 이용한 RPG 게임 캐릭터의 제어

권오광, 박종구⁰

성균관대학교 정보통신공학부

fresian@ece.skku.ac.kr, pjik@yurim.skku.ac.kr⁰

Control of RPG Game Characters using Genetic Algorithm and Neural Network

O-Kyang Kwun, Jong-Koo Park⁰

School of Info. and Communication Engineering, Sungkyunkwan Univ

요약

게임의 발전에 따라 게임에 등장하는 NPC(Non-Player Character)들의 지능 또한 중요성을 더해 가고 있다. 단순히 이동하고 플레이어를 공격하기만 하는 수준을 넘어서 NPC들 역시 다양한 기술과 전술을 사용하는 것이 최근의 MMORPG 게임의 추세이다. 본 논문에서는 신경망과 유전자 알고리즘을 이용하여 볼플레이 게임에 사용되는 캐릭터에게 학습 및 적응 능력을 부여하는 방법을 제안한다.

제안된 지능 캐릭터가 얼마나 게임의 규칙과 전술을 잘 학습하고 적응하는지를 살펴보기 위하여 본 논문에서는 간단한 게임 모델을 제작하여 실험하였다. 캐릭터는 탱커(Tanker), 딜러(Dealer), 힐러(Healer)의 3가지 종류가 있으며, 지능 캐릭터 집단은 신경망과 유전 알고리즘으로 학습되고 FSM으로 움직이는 적 캐릭터 집단과의 전투를 통해 학습한다. 실험 결과 지능 캐릭터가 전투를 통해 자신과 적의 능력에 따른 적절한 전투 방식을 스스로 학습하고, 게임 규칙의 변화에 적응하는 것을 볼 수 있었다.

Abstract

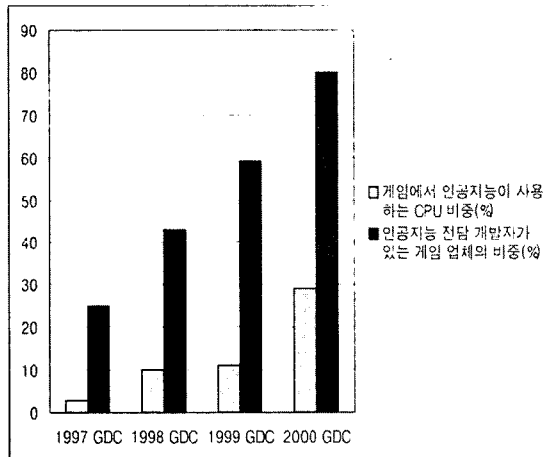
As the development of games continues, the intelligence of NPC is becoming more and more important. Nowadays, the NPCs of MMORPGs are not only capable of simple actions like moving and attacking players, but also utilizing variety of skills and tactics as human-players do. This study suggests a method that grants characters used in RPG(Role-Playing Game) an ability of training and adaptation using Neural network and Genetic Algorithm. In this study, a simple game-play model is constructed to test how suggested intellect characters could train and adapt themselves to game rules and tactics. In the game-play model, three types of characters(Tanker, Dealer, Healer) are used. Intellect character group constructed by NN and GA, and trained by combats against enemy character group constructed by FSM. As the result of test, the proposed intellect characters group acquire an appropriate combat tactics by themselves according to their abilities and those of enemies, and adapt change of game rule.

Keyword : Genetic Algorithm, Netrual Network, Computer Game, MMORPG

1. 서론

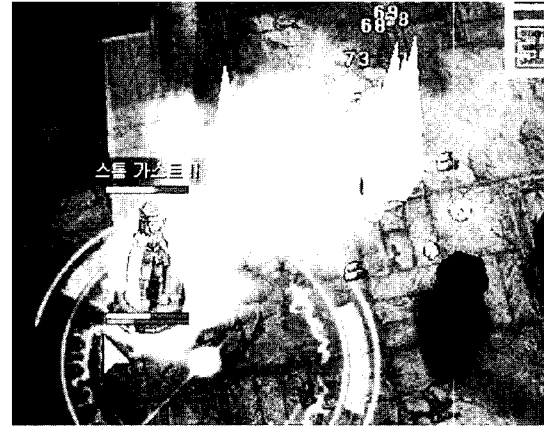
1999년 GDC(Game Developers Conference)이후 게임 캐릭터들을 좀 더 사실적이고 생명체와 비슷한 행동을 하도록 만들기 위해 인공지능이 게임 내에 널리 사용되었다. 그래픽이나 사운드 기능이 일정 수준에 도달하자 게이머들은 보다 자연스럽고 재미있는 게임을 요구하였으며, 인공지능이 이의 돌파구가 되고 있다. 1997년의 경우 게임에 있어 인공지능에 사용되는 CPU의 비율은 5% 이하였으나 2000년의 경우 25%의 CPU가 인공지능 처리에 사용되고 있다. 또한 1997년의 경우 인공지능 담당 프로그래머가 있는 게임 프로젝트는 25%에 불과하였으나 2000년의 경우 80%의 프로젝트에서 인공지능을 담당하는 전담 프로그래머가 있다는 결과를 보인다.[1][표 1]은 게임에서의 인공지능 비중 변화를 보여주는 그래프이다.

최근 들어서는 리니지 2나 월드 오브 워크래프트와 같은 대형 온라인 게임들이 활성화되면서 단순한 순발력에 의거한 조작이나 반복적인 플레이보다는 복잡한 환경에서 다양한 전략과 전술을 구사해야 하는 경향이 강해지고 있다. 이에 따라 게임에 등장하는 NPC(Non-Player Character)에게 지능을 부여할 필요성이 더욱 높아지게 되었다.[2] 그러나 기존의 온라인 게임에서의 캐릭터는 프로그래머가 미리 정해진 패턴대로 움직이는 기능만을 갖추고 있기 때문에 복잡한 게임 환경에 적용할 수 있는 능력이 없다. 또한 고정된 패턴을 가진 캐릭터의 행동은 쉽게 예측가능하기 플레이어를 금방 질리게 만든다.



[표 1] 게임에서의 인공지능 비중 추이

기존 온라인 게임에서 플레이어와 전투를 벌이는 NPC의 인공지능은 대부분 두 가지뿐이다. 플레이어를 보고 가만히 있는가, 공격하는가. 여기서 더 나아가서 캐스팅을 인식하는가라던가 동족 NPC를 공격하는 데 반응한다던가 하는 등의 변형이 나오긴 했지만, 아직까지 대부분 온라인 게임에서 NPC의 행동패턴은 단순한 편이다.



[그림 1] 단순한 행동패턴을 가진 NPC

[그림 1]은 한 국내 온라인 게임에서 플레이어가 기술(skill)을 사용해 만들어 낸, 접근하면 피해를 받는 영역을 인식하지 못하고 이동하는 NPC의 모습이다. 플레이어들은 이러한 NPC들과 싸우는 일에 끝 싫증을 내게 되며, NPC와의 전투를 경험치를 얻기 위한 지루한 작업처럼 생각하게 된다.

게임 내의 NPC가 지능적이고 다양한 행동 패턴을 가지게 되면 게임 내에서의 전투는 플레이어로 하여금 좀 더 긴장감 있고 다양한 즐거움을 맛볼 수 있게 해 줄 것이다. 본 논문에서는 사용자가 NPC와의 전투에서 긴장감과 다양한 즐거움을 느낄 수 있게 하기 위해 게임 내의 NPC가 지능적으로 행동하고 상황의 변화에 스스로 적용할 수 있는 방법을 제안한다.

2. 관련 연구

2.1 온라인 RPG 게임

RPG 게임의 원형은 보드 게임으로, 몇 명의 장교 후보생들이 방의 가운데에 놓인 지도 위에 탱크, 군인 등의 모형들을 놓고 규칙에 따라 움직이던 워 게임(war game)에서 그 유래를 찾을 수 있다. RPG는 Role-Playing Game의 약자로, 플레이어

는 가상의 세계 내에서 등장하는 인물이 되어 그 인물의 성격과 행동을 연기하며 게임을 풀어나가게 된다.

온라인 RPG 게임은 이러한 RPG 게임을 다수의 플레이어가 네트워크를 통해 접속, 한 세계 내에서 서로 의사소통하며 게임을 즐길 수 있게 만든 것으로, 그래픽이 포함된 최초의 온라인 RPG 게임은 넥슨의 '바람의 나라' 이다. 이후 국내에서는 온라인 RPG 붐이 일어났고, NC 소프트의 '리니지'와 그라비티의 '라그나로크' 등 많은 온라인 RPG가 나타났다.

2.2 FSM과 FuSM

FSM(Finite State Machine: 유한 상태 기계)은 유한 수의 상태를 가지고, 그 자신의 상태와 외부로부터의 입력에 근거한 일단의 규칙에 따라 자신의 상태를 변화시키는 규칙 기반 시스템으로 if-else나 switch-case 문장만으로 구현할 수 있기 때문에 구현하거나 이해하기 쉬워 널리 사용된다. FSM과 FuSM은 그동안의 연구에서 컴퓨터 게임에서의 인공지능 기법으로 주로 사용되어 왔다.[3,4]

2.3 신경망

신경망(Neural Network)은 인간의 신경 구조를 모사한 것으로서, 일반적으로 병렬로 수행되는 수많은 간단한 처리 장치로 구성되는 시스템으로 정의한다. 신경망의 가장 큰 특징은 학습 능력으로, 규칙을 정해주지 않아도 신경망은 학습을 통해 지속적으로 더 나은 행동을 찾아낼 수 있다. 이는 게임에 있어서 큰 장점이다.

신경망을 게임에 적용하는 연구는 많이 시도되어 왔으며, 대부분이 오목이나 Tic-Tac-Toe 등의 보드 게임이었지만 최근에는 대전 액션 게임이나 인공생명 등 다양한 분야에 사용하는 연구가 이루어지고 있다.[5,6]

2.4 유전 알고리즘

유전 알고리즘(Genetic Algorithm, GA)은 자연 생태계의 진화 과정을 모사한 탐색 알고리즘이다. 유전 알고리즘은 수많은 해들 중 우수한 것들끼리의 교배를 통해 세대가 진행될수록 더 나은 해를 찾아간다. 이 과정에서 지역 최적해에 빠지는 것을 막기 위해 돌연변이 등의 기법이 사용된다.

유전 알고리즘은 탐색공간이 매우 크거나 일반적인 해를

찾기 힘든 문제(TSP문제 등)에서 뛰어난 성능을 발휘하며, 반드시 최적해가 필요한 경우가 아닐 때 유용하다.[7] 이는 GA가 빠르게 최적해 근처까지 수렴하지만, 최적해 주위에서 정확한 답을 찾기까지의 성능이 좋지 않기 때문이다. 게임에서는, 대전 액션 게임에 유전 알고리즘을 적용하려는 연구가 시도된 적이 있다.[8] 이 연구에서는 염색체가 캐릭터간의 거리와 행동에 따라 어떤 행동을 할 것인지에 대한 정보를 포함하고 있다.

3. 지능 캐릭터 집단의 설계

3.1 게임 구성

본 논문에서는 지능 캐릭터의 학습과 검증을 위해 간단한 게임 모델을 만들었다. 이 모델은 RPG 게임의 전투를 단순화시킨 것으로, 각각 4개의 캐릭터로 이루어진 두 팀이 제한된 2차원 공간 내에서 전투를 벌여 그 결과에 따라 점수를 획득한다.

두 팀 중 한 쪽은 FSM으로 행동을 결정하며, 다른 한 쪽은 신경망으로 행동을 결정하게 된다. 신경망으로 구성된 캐릭터(지능 캐릭터로 통칭) 집단이 FSM으로 구성된 캐릭터(상대 캐릭터로 통칭) 집단과의 전투에서 얻은 점수는 유전 알고리즘에서 적합도로 사용된다.

3.1.1 캐릭터 성향과 직업

지능 캐릭터는 세 가지의 성향을 지닌다. 이는 현재 상용화되어 있는 온라인 RPG 게임에서의 일반적인 분류이다. 실제 MMORPG에서 캐릭터의 역할은 더 세부적으로 분화되지만, 본 논문에서는 가장 큰 3종류의 분류만 사용했다.

탱커(Tanker): 탱커는 맨 전열에 나서서 팀의 다른 캐릭터들이 공격을 받지 않게 막아주며 적과 싸우는 성향의 캐릭터를 뜻한다. 탱커 캐릭터에게 가장 중요한 능력치는 체력이다.

딜러(Dealer): 딜러는 적에게 최대한의 데미지를 입히는 성향의 캐릭터를 뜻한다. 딜러 캐릭터에게 가장 중요한 능력치는 공격력이다.

힐러(Healer): 힐러는 아군의 체력을 회복해주는 성향을

지닌 캐릭터이다. 오직 치유 능력이 있는 캐릭터만이 힐러 성향을 가질 수 있다.

이러한 파라미터가 필요한 이유는, 캐릭터의 지능을 게임 자체와 분리시키기 위해서이다. 실제 게임에서는 게임의 종류에 따라 다양한 클래스의 캐릭터가 등장할 수 있으며, 이들 모두에 대해 따로 학습을 시키는 것은 비효율적이다. 그렇기 때문에 캐릭터의 움직임을 제어하는 신경망은 캐릭터의 세부적인 클래스나 능력에 신경쓰지 않고, 그 캐릭터의 성향만을 기준으로 행동을 지시할 수 있다.

게임 모델에서 사용되는 캐릭터의 클래스는 다음과 같다.

- 전사(Fighter) : 전사는 높은 체력과 중간 정도의 공격력을 지니며, 탱커의 역할을 맡는 클래스이다.
- 도적(Thief) : 도적은 중간 정도의 체력과 높은 공격력을 지니며, 딜러의 역할을 맡는 클래스이다.
- 마법사(Magician) : 마법사는 낮은 체력과 가장 높은 공격력을 지니며, 딜러의 역할을 맡는다.
- 사제(Priest) : 사제는 낮은 체력과 낮은 공격력을 지니며, 유일하게 힐러의 역할을 맡을 수 있는 캐릭터이다.

본 논문의 게임 모델에서는, 각 팀은 전사 1명, 도적 1명, 마법사 1명, 사제 1명으로 구성된다.

캐릭터는 공격력, 체력, 공격속도, 회복력의 3가지 파라미터를 가지고 있다. 게임 환경을 단순하게 하기 위해 시야 제한은 넣지 않았으며, 모든 캐릭터는 필드 전체를 볼 수 있다.

각 클래스의 파라미터는 다음과 같다.

- 전사: 공격력 2, 체력 100, 공격속도 2
- 도적: 공격력 4, 체력 75, 공격속도 2
- 마법사: 공격력 10, 체력 50, 공격속도 4
- 사제: 공격력 3, 체력 50, 공격속도 4, 치유량 15

캐릭터는 공격시 상대방의 체력을 공격력만큼 감소시킨다. 공격속도는 공격에 걸리는 시간(턴 단위)이며 공격 명령이 내려진 후 공격속도만큼의 턴이 지나야 공격이 이루어진다. 치유 기술 사용시 대상의 체력을 치유력만큼 회복시킨다. 치유 기술 사용에는 공격시와 같은 시간이

필요하다. 치유가 무한으로 계속되어 전투가 끝나지 않는 상황을 막기 위해 사제 캐릭터의 치유는 사용 횟수 제한을 두었다.

실험에서는 모든 파티의 구성을 동일하게 하며, 전사 1명, 도적 1명, 마법사 1명, 사제 1명으로 구성된다.

3.1.2 행동 설계

캐릭터는 대기, 이동, 공격, 치유, 도주의 5가지 상태(state)를 지닌다.

대기: 캐릭터는 아무 일도 하지 않는다. 이 상태는 공격이나 치유에 걸리는 시간동안 캐릭터를 정지시키는 데만 사용된다.

이동: 캐릭터는 아군 전체의 좌표의 평균 지점 방향으로 이동한다.

공격: 캐릭터는 지정된 적을 공격한다. 지능 캐릭터는 신경망에 의해 지정된 적을, 상대 캐릭터는 적대 수치가 가장 높은 적을 공격한다.

후퇴: 캐릭터는 자신을 공격하는 첫 번째 적의 반대 방향으로 이동하게 한다. 해당 캐릭터를 공격하고 있는 적이 없다면, 캐릭터는 대기 상태로 전이한다.

치유: 캐릭터는 지정된 아군을 치료한다. 지능 캐릭터는 신경망에 의해 지정된 아군을, 상대 캐릭터는 현재 체력이 가장 낮은 아군을 치료한다. 해당 캐릭터에게 치유 능력이 없다면, 이 명령은 무시된다.

공격의 방식에는 근거리 공격과 원거리 공격이 있으며, 전사/도적/사제 캐릭터는 근거리 공격을, 마법사 캐릭터는 원거리 공격을 한다. 근거리 공격의 거리는 1칸이며, 마법사의 원거리 공격 거리는 10칸이다.

3.1.3 실험 환경

게임은 40*40 크기의 맵에서 진행되며, 각 캐릭터는 맵에서 1칸을 차지한다. 시작시 지능 캐릭터 팀과 상대방 캐릭터 팀은 맵의 오른쪽 끝과 왼쪽 끝에서 나타나며, 곧바로 상대방을 인식하고 전투를 시작하게 된다. 전투에 사용되는 파라미터와 행동은 앞에서 설명한 대로이며, 체력이 0 이하가 된 캐릭터는 사망한다. 전투는 지능 캐릭터와 상대 캐릭터 집단 중 한 쪽이 전멸하거나 300턴이 지날 때까지 계속

된다. 전투 결과의 평가 규칙은 다음과 같다.

- 제거한 상대 캐릭터 하나당 200점.
- 전투 후 살아남은 지능 캐릭터 하나당 200점.
- 상대 캐릭터들에게 입힌 데미지 총량만큼의 점수.
- 아군을 치료한 치유량 만큼의 점수.
- 300턴에서 1턴 일찍 전투가 끝날 때마다 1점씩 추가.
- 잘못된 행동(이미 제거된 적을 공격하거나 제거된 아군을 치료하거나, 치료 능력이 없는 캐릭터에게 치료 명령이 내려질 경우)마다 1점씩 감점.
- 총점이 0이하이면 0점으로 처리.

전투가 끝나면 평가 점수를 적합도로 교배 및 돌연변이가 일어나며, 다음 세대의 지능 캐릭터 집단이 생성되고 전투가 반복된다.

3.1.4 상대 캐릭터의 설계

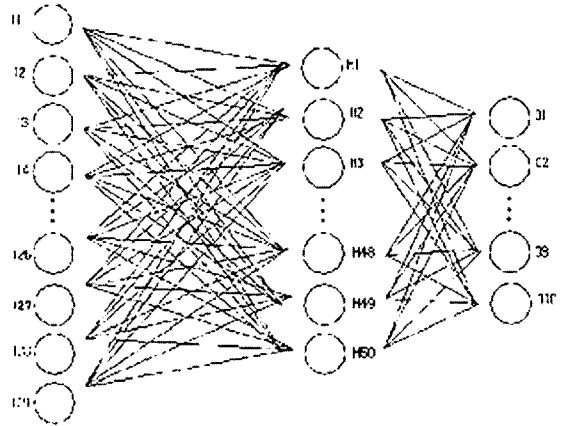
상대 캐릭터는 FSM으로 제작되었으며, 아래의 조건에 따라 자신의 상태를 선택한다.

1. 적을 발견하면 자신에게 가장 많은 적대(Aggro)수치를 지닌 적을 공격한다. 적대 수치는 캐릭터가 특정 상대로부터 받은 데미지에 비례한다. 전투가 시작됨과 동시에 각 캐릭터는 상대의 위치를 알 수 있으므로, 캐릭터는 전투 시작과 동시에 공격 상태로 전이한다.
2. 치유가 가능한 캐릭터는 피해를 입은 아군이 있다면 가장 체력이 낮은 아군을 치료한다.
3. 체력이 일정 이하인 캐릭터는 도주한다.

현재 상용화되어 있는 대부분의 MMORPG에서 NPC는 적대 수치가 가장 높은 적을 공격한다. 상용화된 게임에서는 이와 관련된 다양한 규칙이나 기술들이 존재하지만, 본 논문에서는 게임 모델을 최대한 단순화시키기 위해 그러한 특징들을 사용하지 않았다.

3.2 신경망 구조

본 논문에서 사용한 신경망의 구조는 [그림 2]와 같다. 입력 노드는 I1~I29, 은닉 노드는 H1~H50, 출력 노드는 O1~O10이다.



[그림 2] 신경망 구조

지능 캐릭터 집단을 조종하는 신경망은 상대 캐릭터 집단과의 대결을 통해 경험을 얻고, 그것을 바탕으로 스스로를 진화시킨다. 상대 캐릭터 집단은 인공지능 없이 개개인으로만 움직이는 캐릭터나 FSM과 같은 기존의 방법으로 구현된 캐릭터, 온라인 게임에서 사람이 조작하는 캐릭터 모두가 가능하다. 여기서는 FSM으로 구현된 캐릭터 집단을 상대 캐릭터 집단으로 사용했다.

게임 상황에서 지능 캐릭터 N의 행동을 결정하기 위해서는 몇 가지 정보가 필요하다. 우선, 상대방 캐릭터의 존재 여부와 위치, 그리고 현재의 행동을 알아야 한다. 상대방이 접근해 있는지 혹은 멀어져 있는지, 공격중인지 후퇴중인지, 체력은 얼마나 남아 있는지를 살피고, 그에 따라서 행동해야 한다. 이는 최대 4명의 적 모두에 대해 적용되어야 한다. 또한 신경망은 아군의 위치와 상태에 대해서도 알고 있어야 한다. 아군 캐릭터가 어디에 있는지, 그리고 남은 체력은 얼마인지에 따라 그 캐릭터를 공격시킬 것인지 후퇴시킬 것인지, 공격한다면 누구를 공격하는 것이 가장 유리한지를 판단할 수 있다.

신경망은 상대방 캐릭터의 위치와 행동, 남은 체력과 아군 캐릭터의 위치와 남은 체력, 현재 행동 명령을 내릴 캐릭터의 성향 및 특성을 입력으로 받아들인다. 신경망에 입력하기 위해서는 각 입력값이 숫자로 매핑되어야 한다. 각 캐릭터의 위치는 자신의 위치와의 상대좌표로 입력되며, 남은 체력은 퍼센테이지로 표현된다. 캐릭터의 행동의 종류는 유한하므로 역시 정수로 매핑 가능하다. 아군과 적군의 수는 각각 4명이기 때문에 결과적으로 입력노드의 개수는 $4*4+3*4+1=29$ 개

가 된다. 각 입력 노드의 기능은 [표 2]와 같다.

출력의 개수는 지능 캐릭터가 취할 수 있는 행동의 개수와 같다. 단 공격과 치유 명령은 대상을 지정해야 하기 때문에 각각의 대상에 대한 명령마다 다른 노드를 사용한다. 결과적으로 신경망의 출력 노드 수는 $1+4(\text{각각의 적에 대한 공격 명령})+1+4(\text{각각의 아군에 대한 치유 명령})=10$ 개가 된다.

노드	역할	노드	역할
1	적 1의 X거리	16	적 4의 상태
2	적 1의 Y거리	17	아군 1의 X거리
3	적 1의 체력	18	아군 1의 Y거리
4	적 1의 상태	19	아군 1의 체력
5	적 2의 X거리	20	아군 2의 X거리
6	적 2의 Y거리	21	아군 2의 Y거리
7	적 2의 체력	22	아군 2의 체력
8	적 2의 상태	23	아군 3의 X거리
9	적 3의 X거리	24	아군 3의 Y거리
10	적 3의 Y거리	25	아군 3의 체력
11	적 3의 체력	26	아군 4의 X거리
12	적 3의 상태	27	아군 4의 Y거리
13	적 4의 X거리	28	아군 4의 체력
14	적 4의 Y거리	29	캐릭터의 성향
15	적 4의 체력		

[표 2] 입력 노드 매핑

은닉 노드는 50-100개 사이로 설정해 본 결과 50개로도 결과에 큰 차이가 없었으며, 본 실험에서는 50개의 은닉 노드를 사용했다.

캐릭터는 진행중인 동작을 중간에 멈출 수 없다. 그렇기 때문에 캐릭터가 공격과 같이 1턴 이상이 걸리는 행동을 하고 있을 경우는 그 동작이 끝날 때까지 지휘자 캐릭터가 명령을 내릴 필요가 없으므로, 다음 명령은 그 행동이 끝난 후에 이루어지게 된다. 학습이 진행됨에 따라 신경망은 특정한 상황(적과 아군의 위치 및 상태)에 따른 최적의 명령이 무엇인지를 학습하게 된다.

3.3 진화 방식

각 지능 캐릭터 집단은 자신의 유전 정보를 가지고 있고, 이 유전 정보를 이용하여 신경망을 구성한다. 각 캐릭터의 행동은 신경망에 의해 결정된다.

유전자는 신경망의 각 노드들의 가중치 리스트를 나타내는 실수값들로 구성되어 있다. 3.2 절에서 구현한 신경망의 구조는 입력 노드 29개, 은닉 노드 50개, 출력 노드 10개이므로 총 노드 수는 $29 \times 50 + 50 \times 10 = 1950$ 개가 된다. 유전자 코

드의 길이가 길기 때문에 교배 방식은 다점 교배를 사용했으며, 교차점은 10개로 설정했다.

유전 알고리즘에서 해를 찾는 시간에 큰 영향을 주는 것은 선택압(selection pressure)과 돌연변이율이다. 선택압은 우수한 해들과 열등한 해들 사이의 적합도 차이 비율을 뜻한다. 선택압 k 는 가장 좋은 해의 적합도가 가장 나쁜 해의 적합도의 k 배가 되도록 조정하는 것을 의미한다. 선택압이 높을수록 수렴은 빠르나 설익은 수렴 (premature convergence)의 가능성이 높아진다. 반면 선택압이 너무 낮으면 해집단의 평균 품질이 좋아지지 않을 가능성이 많다.[9] 선택압이 높을 경우 유전 알고리즘은 깊이 우선 탐색(depth-first search)에 가까운 형태를 띠고, 낮을 경우는 너비 우선 탐색(breadth-first search)에 가까운 형태가 된다. 선택압은 일반적으로 3~4사이에서 정하며, 본 실험에서는 3~8 사이의 값을 실험한 결과 가장 우수한 결과를 보이는 4를 선택했다. 또한 전 세대의 가장 우수한 개체 1개를 보존하여 다음 세대로 넘기는 엘리트주의를 채택했다.

돌연변이는 세대가 진행됨에 따라 변이의 강도를 점점 줄여나가는 비균등 변이(non-uniform mutation)를 사용했다. 사용된 변이 공식은 (1)과 같으며, 이는 일반적으로 사용되는 비균등 변이 공식을 단순화한 것이다.

$$v = \begin{cases} v + \Delta(t, UB - v), & \text{if } r = 0 \\ v - \Delta(t, v - LB), & \text{if } r = 1 \end{cases} \quad (1)$$

여기서

$$\Delta(t, y) = y \cdot r_2 \cdot \left(1 - \left(\frac{t}{T}\right)^2\right), \quad (2)$$

r : [0, 1] 범위의 난수. r_2 : 제어 상수.

UB: v 의 최대값. LB: v 의 최소값.

t : 현재 세대수. T : 최대 세대수(끝나는 시간).

이다. r_2 값은 1을 사용했다.

각 세대의 개체수는 100개로, 학습은 총 1000 세대동안 진행되었다.

4. 실험 결과

지능 캐릭터의 평가는 상대방 캐릭터와 300턴간 전투를 벌인 후 '상대방 캐릭터의 남은 체력/아군 캐릭터의 남은 체력'으로

평가했다. 여기에 죽인 상대방 캐릭터의 숫자마다 추가 점수를 주고, 아군 캐릭터의 사망시 점수를 깎도록 했다. 이러한 방식으로 게임 규칙의 변화에 따른 적응 능력을 확인하기 위해 학습 도중 게임 규칙을 변화시켜서 지능 캐릭터 집단이 이에 적응하는지를 확인했다. [표 3]은 기본 게임 규칙을 이용해 100세대까지 학습시켰을 때의 세대별 점수 표이다.

1 st fitness average:	88, best: 582
100 th fitness average:	370, best: 1054
200 th fitness average:	391, best: 1054
300 th fitness average:	404, best: 1054
400 th fitness average:	388, best: 1054
500 th fitness average:	395, best: 1054
600 th fitness average:	404, best: 1169
700 th fitness average:	398, best: 1905
800 th fitness average:	491, best: 1933
900 th fitness average:	697, best: 1969
1000 th fitness average:	1023, best: 2017

[표 3] 세대별 평균 점수와 최고 점수(1)

학습 결과 적합도는 완만하게 상승하다가 특정 시점을 기준으로 급격히 상승하고, 이후 다시 완만한 상승이나 정체를 반복했다. 전 세대의 가장 우수한 개체가 그대로 다음 세대로 넘어오기 때문에 평균 적합도는 상승하지만 각 세대의 최우수 개체는 계속 유지되다가, 그보다 더 뛰어난 개체가 등장하면 바뀌는 경향을 보였다. 지능 캐릭터 집단은 100세대가 지나기 전에 모든 캐릭터들이 한 명의 적을 집중적으로 공격하는 것이 유리하다는 것을 학습했다. 하지만 여기에서 사제 캐릭터는 적당한 시점에 아군을 치료하는 쪽이 더 유리하다는 것을 학습하는 데는 상당한 시간이 걸렸다. 600세대 이후 치유에 대해 학습한 지능 캐릭터들의 적합도는 빠르게 상승했다.

1000세대 이후, 게임 규칙을 다음과 같이 바꾼 상태에서 학습을 계속 진행했다.

마법사 캐릭터의 공격력: 8, 사제 캐릭터의 치유력: 17

게임 규칙을 바꾼 후의 학습 결과는 [표 4]와 같았다.

1 st fitness average:	271, best: 411
50 th fitness average:	268, best: 872
100 th fitness average:	264, best: 1024
150 th fitness average:	273, best: 1024
200 th fitness average:	276, best: 1024
250 th fitness average:	267, best: 1024
300 th fitness average:	267, best: 1216
350 th fitness average:	326, best: 1912
400 th fitness average:	1021, best: 1970

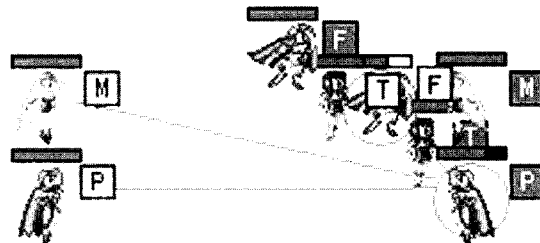
[표 4] 세대별 평균 점수와 최고 점수(2)

첫 번째 규칙에서, 사제 캐릭터는 치유 능력을 사용하지 않는 쪽으로 학습한다. 상대방 캐릭터의 공격력에 비해 치유량이 부족해서, 치유 능력을 사용해도 집중적으로 공격받는 캐릭터를 살릴 수 없기 때문이다. 그러나 규칙을 바꾸게 되면 치유량이 공격력을 상회하기 때문에, 사제 캐릭터는 치유 능력 사용 횟수를 모두 사용할때까지 캐릭터를 살릴 수 있다. 그렇기 때문에 두 번째 규칙에서 사제 캐릭터는 치유 능력을 사용하는 쪽이 유리하다.

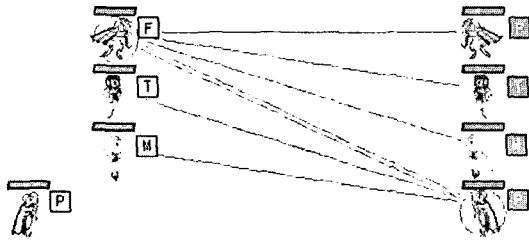
변경된 규칙으로 학습한 후, 이 사실을 학습하지 못한 지능 캐릭터 집단의 적합도는 급격히 떨어졌지만 학습이 계속됨에 따라 이전 수준과 비슷한 적합도를 획득했다.

이것은 지능 캐릭터 집단이 변화한 게임 규칙에 적응했다는 것을 뜻한다. [그림 3]과 [그림 4]는 각각 규칙 변경 이전과 이후의 모습이다. 붉은 선은 공격, 회색 선은 치유를 나타내며, 전사, 도적, 마법사, 사제 캐릭터에는 각각 F, T, M, P의 기호가 표시되어 있다.

[그림 3]에서는 사제 캐릭터가 치유 능력을 사용하지 않고 다른 캐릭터와 함께 적을 공격한다. 그러나 규칙이 바뀐 후 실험한 결과인 [그림 4]에서는 사제 캐릭터가 아군을 치유하는 것을 볼 수 있다.



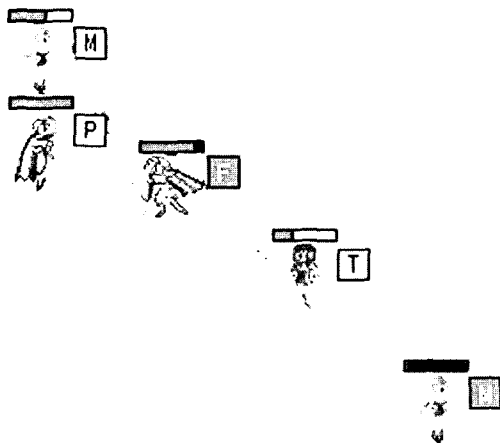
[그림 3] 규칙 변경 전 학습된 사제 캐릭터의 행동



[그림 4] 규칙 변경 후 학습된 사제 캐릭터의 행동

또한 실험에서 지능 캐릭터 집단은 FSM으로 설계된 상대 캐릭터 집단에 비해 우수한 성능을 보여 주었다. 700세대 이후의 각 세대의 가장 우수한 개체들은 상대 캐릭터 집단을 전멸시키고 4명중 2명, 혹은 3명이 생존했으며, 상대 캐릭터 집단이 보여주지 못한 효율적인 전투 방식을 찾아내기도 했다. 예를 들어 3.1.4 장에서 설명했듯이 상대 캐릭터는 일정 이하로 체력이 떨어지면 도주하도록 되어 있는데, 한번 도주한 캐릭터는 체력을 회복하기 전에는 다시 전투에 참가하지 않으므로 위협이 되지 않는다.

[그림 5]에서 상대 마법사 캐릭터(M으로 표시)는 체력이 떨어진 상태이기 때문에 도주한다. 이 캐릭터는 체력이 회복되지 않는 한 다시 공격하지 않으므로 지능 캐릭터들은 도망치는 적을 추적하는 대신 남아 있는 적에게 공격을 집중하고 있다.



[그림 5] 위협이 되는 적부터 공격하는 지능 캐릭터

5. 결론 및 향후 연구방향

본 실험에서 사용된 상대 캐릭터 집단은 실제 게임에서 처

럼 사용하는 기술(skill)은 없지만, 기본적인 행동은 국내에서 서비스되고 있는 MMORPG의 NPC들과 거의 유사하다. 이는 실제 게임에서도 이러한 학습 방법이 효과를 발휘할 수 있음을 의미한다. 제안된 캐릭터 학습 방법은 게임 내의 NPC들이 좀 더 다양하고 인간이 조작하는 것에 가까운 행동을 보임으로서 사용자들에게 긴장감과 즐거움을 줄 수 있으며, 게임의 규칙이나 환경이 변화해도 스스로 학습함으로써 개발자의 부담을 줄일 수 있다.

그러나 MMORPG의 NPC 학습에 신경망을 적용하기 위해서는 아직 기술적인 문제점들이 남아 있다. 본 게임 모델에서는 각 집단의 캐릭터 수를 4명씩으로 고정했지만 실제 게임에서는 훨씬 많은 수의 적 또는 아군 캐릭터가 존재할 수 있다. 이 경우 모든 입력값들을 고려하기 위해서는 신경망의 크기가 기하급수적으로 늘어나게 되며, 이는 NPC 서버에 부담을 주게 될 가능성이 크다.

이러한 문제를 해결하기 위한 방법 중 하나는, 지능 캐릭터 집단의 제어를 계층적으로 처리하는 것이다. 지능 캐릭터 집단 4개를 제어하는 신경망을 학습시키면, 5개의 신경망으로 총 16개의 캐릭터를 제어할 수 있게 된다. 이에 대한 연구는 본 논문의 범위를 벗어나므로 상세하게 다루지 않는다.

참고문헌

- [1] 이만재, “게임에서의 인공지능 기술,” 한국정보처리학회지, Vol. 9, No. 3, pp. 69-76, May, 2002.
- [2] Steve Woodcock, “Game AI : The State of the Industry 2002,” Game Developer Magazine, 2002.
- [3] Mark Deloura, Game Programming Gems 2, Charles River Media, 2001.
- [4] Daniel Johnson and Janet Wiles, “Computer Games With Intelligence,” IEEE International Fuzzy Systems Conferences, Vol. 3, pp. 1355-1358, 2001.
- [5] 조병헌, 정성훈, 성영락, 오하령, “인공지능: 신경망을 이용한 지능형 게임 캐릭터의 구현,” 정보처리학회논문지B, Vol. 11, No. 7, pp. 831-840, 2004.
- [6] 성백균, “게임 캐릭터의 전략 구현을 위한 인공생명 시뮬레이션,” 충주대학교 논문지, Vol. 35 No. 2, pp.

- 357-366, 2000.
- [7] Thang Nguyen Bui and Byung Ro Moon, "On multi-dimensional encoding/crossover," International Conference on Genetic Algorithms, pp. 49-56, 1995.
- [8] 이면섭, 조병현, 성영락, 정성훈, 오하령, "유전자 알고리즘을 이용한 대전형 액션게임의 지능캐릭터 구현," 정보처리학회논문지B, Vol. 12, No. 3, pp. 329-336, 2005.
- [9] 문병로, 유전알고리즘, pp. 41, 두양사, 2003.



권오광

2005년 성균관대학교 전기전자 및 컴퓨터공학부 졸업(학사)
2005년 ~ 현재 성균관대학교 컴퓨터공학 대학원(석사과정)
관심분야: 게임 인공지능, 온라인게임 서버 프로그래밍 등



박종구

1987년 서울대학교 제어계측공학과 졸업(학사)
1989년 서울대학교 제어계측공학과(공학석사)
1993년 서울대학교 제어계측공학과(공학박사)
1995년 ~ 현재 성균관대학교 정보통신공학부 교수
2002년 ~ 현재 산업자원부지정 성균관대학교
게임기술개발지원센터 센터장
관심분야: 가상현실, 게임 기반기술, 제어이론 및 응용 등

논문투고일 - 2006년 3월 14일
심사완료일 - 2006년 6월 7일