유전자 알고리즘을 이용한 게임 NPC 행동 경로 분석 및 흥미 향상

하 준*, 김재웅**

요약

대부분의 게임제작자가 용, 로봇, 혹은 설치류와 같이 인간이 아닌 생물 등을 제어하기 위한 다양한 시도를 하고는 있으나 현재까지 게임 진행에 관계없는 등장요소(NPC:Non Player Character)들에게 인공 지능을 부여하는 연구는 부족한 실정이다. 게임에 인공지능을 적용하기 위해서는 복잡한 문제를 해결하기 위한 많은 연구가 필요하지만 NPC에게 다른 인격, 감정 또는 두려움이나 동기유발성의 경향을 갖은 외형을 부여하는 시도를 위해 인공지능 기술을 사용한다면 콘텐츠의 사실적 흥미유발 효과는 증가될 수 있다. 지금까지는 기존 하드웨어 자원의 제약으로 인하여 상대적으로 게임 내의 인공지능에 대한 관심도가 낮았으나 최근 게임에는 개발자, 제작자 및 관리자들에게 게임 인공지능의 중요성이 확고하게 자리를 잡아가고 있다. 본 연구는 게임 사용자의 높은 관심을 유지하기 위하여 유전학적 기반의 인공지능 기법을 적용하여 게임 NPC 행동경로를 분석한 후 게임 흥미성을 향상시키는 방법을 제시한다. 또한 설계된 게임 NPC 행동경로가 게임 맵에서 고르게 사용되어 짐을 보인다.

Analysis of Game NPC Behavior Routine and Improvement of Interesting using Genetic Algorithm

Jun He*, Jae-Woong Kim**

ABSTRACT

In games game player are not always interested in giving non-player characters human-level intellect. Game producers write codes to control nonhuman creatures such as dragons, robots, or even rodents. Game AI still needs more study to solve fairly complex problems. However, if NPC is used for attempt to have different personalities and unusual appearances with emotions, fear, and disposition, the effect of entertainment and variety of the game content is expected to be much stronger. Up to now AI interest level in the game industry is low relatively owing to the restrictions of the existing , hardware resources , but in recent game developers , producers and managers think the game AI of as an important factor. This study analyses the behavioral course of NPC applied genetic AI Method to maintain the user's interest and provides the way how to enhance the user's interest. It is shown that Designed and realized game NPC behavioral course is used properly in the game map.

Key Words : Artificial Intelligence, Fuzzy State Machine, Genetic Algorithm, Game

* 공주대학교 컴퓨터공학과(✉ictree@kongju.ac.kr)
** 공주대학교 컴퓨터공학부
· 제1저자(First Author) : 하 준 · 교신저자(Correspondent Author) : 김재웅
· 접수일(2012년 3월 21일), 수정일(1차 : 2012년 4월 16일), 게재확정일(2012년 4월 19일)

Ⅰ. 서론

1.1 현재 일반적인 게임 AI수준

현재 게임에서 가장 많이 사용되는 AI 기술은 속임수이다. 예를 들면, 전쟁 가상 게임에서 컴퓨터는 정찰병을 파견하지 않고도 상대방의 모든 정보(기지의 위치, 유형, 개수 및 장비들의 위치 등)에 접근 할 수 있다. 이렇듯 일반적으로 사용하는 컴퓨터에게 인간상대에 대항할 예리한 정보를 제공 한다. 그러나 속임수는 예상외의 결과를 초래 할 수 있는데 만일 컴퓨터가 속임수를 쓰고 있다는 것이 게임 사용자에게 노출될 경우에, 게임 사용자는 그의 노력이 쓸모없다고 여길 것이고 게임에 대한 관심을 잃게 될 것이다. 더구나 속임수 사용능력이 컴퓨터에 과다하게 부여될 경우 게임 사용자가 컴퓨터를 상대하는 것이 불가능해 진다. 따라서 속임수는 상대방의 게임에 대한 관심과 즐거움을 유지시키기 위해 반드시 일정한 균형을 맞추어야 한다.

1.2 게임인공지능

1.2.1 게임AI기술과 논쟁대상

게임인공지능기술은 일반적으로 결정론 과 비결정론 두 가지로 구분된다. 결정론적인 인공지능기술은 게임AI의 빵과 버터의 관계로 비유할 수 있다. 이 기술은 예측이 가능하고, 빠르며 실행가능하고, 이해하기 쉽고, 시험가능하고, 결함을 제거하기 쉬운 장점이 있다.

그러나 현재 많이 사용되는 결정론적 방법은 모든 예측 가능한 시나리오를 만들고 프로그래밍 해야 하는 어려움이 뒷따른다. 또한 게임 습성이 몇 번의 게임 수행을 통해 쉽게 예견되어지므로 게임전체 주기를 통해 학습과 진화의 촉진이 어렵다. 반면에 비결정론적 방법은 학습과 예측불허의 게임진행을 촉진하며, 개발자들이 모든 가능한 시나리오의 예상되는 행동을 명확하게 프로그램 할 필요는 없다. 또한 비결정론적 방법은 스스로 학습과 추정을 할 수 있고 명확한 명령 없이 돌출행동에 대한 대처 행동을 진행한다.[1]

1.2.1.1 결정론

결정론적 행동 혹은 동작은 구체적이고 예측할 수 있으며, 불확실성은 존재하지 않는다. 결정론적 행동의 간단한 예는 추적 알고리즘이다. 이는 특정한 등장요소의 x, y 좌표값이 목표지점에 일치할 때까지 어떤 목표점을 향해서 움직이도록 NPC(Non Player Character)를 명확히 프로그램 한다.

1.2.1.2 비결정론

비결정론적 행동은 결정론적 행동과 확실성에 대한 반대 행동이다. 행동은 불확실성의 등급을 갖고 예측할 수 없으며(불확실성의 등급은 적용된 인공지능 방법과 얼마나 잘 그 방법이 이해되었느냐에 따라 달라진다), 비결정적 행동의 한 예는 전투전술을 적용하기 위해 NPC가 학습하는 것이다. 그러한 학습은 신경망(neural network), 베이시안 기술(bayesian technique), 유전적 연산방법(genetic algorithm)등으로 사용한다.[2]

1.2.1.3 유전학

임의 변이(Random mutations)은 새로운 것을 시도하는 자연적 방법들이다. 만일 임의 변이가 "종"을 개선한다면, 변이적 미래세대로 진행될 것이고, 그렇지 못하다면 반대의 경우가 될 것이다. 부모 세대의 우성 유전자로부터 전해온 염색체는 임의적 변이와 재결합함으로서, 그들의 환경에 적응하는 자식세대를 생성한다. 이 개념은 게임에서도 똑같이 적용된다. 생물학적 세상과 마찬가지로 게임세상의 요소들은 변화된 상황에 적응하고 진화해 나가도록 만들어진다. [3]

II.게임인공지능 디자인 구상

2.1 게임인공지능

사용자를 위한 도전적 게임환경을 구축하는 것은 게임디자이너의 과제이다. 사실상 게임산업의 커다란 부분이 게임세상의 균형에 관여하고 있다. 게임환경은 사용자들에게 충분히 도전적이어야 하며, 그렇지 못하면 게임 자체가 너무 쉬워 짧은 시간내에 흥미를 잃게 된다.[4] 반대로 만일 게임이 너무 어려우면 사용자는 좌절을 경험하게 되어 일부 사용자들은 빠르게 포기하게 될 것이다. 게임 디자인의문제점은 사용자들의 게임기술이 매우 다르다는 사실에도 있다. 그러므로 균형적이고 도전적인 게임을 구축한다는 것은 굉장히 어려운 문제이다. 게임인공지능의 유전적 연산방식은 이런 문제를 해결 하는데 도움이 된다.

2.2 게임인공지능의 유전적 개념의 사용

게임에서 유전적 알고리즘은 4단계로 분류할 수 있다. 그림 1은 게임의 유전적 연산방식의 4단계 과정이다.

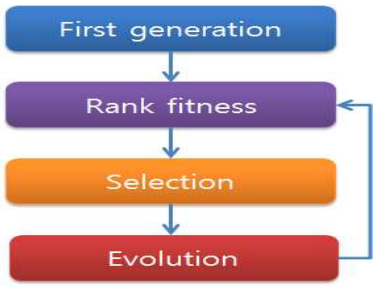

그림 1. 게임의 유전적 연산방식
Fig. 1 Genetic Algorithm of Game

첫 번째 단계는 생성이며, 이는 시작특징을 지정함으로 시작된다. 만일 그 환경에 상호작용하기 시작하면, 두 번째 단계인 각 객체를 등급화하는 방법인 등급적합화 단계가 수행된다. 이 과정은 어떤 구성원들이 가장 성공적인지 말해주며, 이후 세 번째 단계인 선택과정에서 다음 세대를 생성하기 위한 특정 구성원을 선택한다. 다음 세대를 생성하기 위해서 이전 세대의 가장 성공적인 특징을 사용한다. 마지막 단계인 진화는 발전적인 세대를 한하기 위해 이 특징들을 결합하는 과정이며, 그 결과 두 번째 세대가 생성되고 이 과정을 계속 반복한다.[5]

유전적 연산방식은 본질적으로 특징의 가장 알맞은 설정을 찾는 최적화과정이다. 이 과정은 특정 문제의 최적화된 해결법을 찾는다.

2.3 유전적 연산방식의 장점과 단점

게임에서 유전적 연산방식은 어떤 문제를 해결하는 최적화된 해결법을 찾아가는 방법이다. 물론, 게임인공지능에는 많은 문제들이 산재해 있고 그리고 그들 모두가 유전적 연산방식으로 해결 할 수 있는 것은 아니다. 한 예로 통로찾기 게임은 유전적 연산방식으로 찾아질 수는 있지만 A* 알고리즘을 이용할 경우 더 적합하다. 유전적 연산방식은 문제의 요소들이 부분적으로 예측 불가능할 때 적합하다[6]. 게임 디자이너는 예측할 수 없는 상황에 게임인공지능을 적용한다. 게임 디자인에 있어 게임 환경에서 예측하기 가장 어려운 요소가 바로 사용자이며, 어떤 게임수준에서는 게임디자이너들이 도전 하는 상대들을 창조하기 위해 사용자의 행동을 예측할 수 있어야 한다. 불행히도 그것을 예측하거나 모든 가능한 사용자들의 행동을 설명하는 일은 쉽지 않은 일이다[7].

III.유전학적 게임개발을 이용한 인공지능

3.1 진화과정

진화과정은 다음과 같은 단계로 진행된다.

(1) 문제풀이를 따라 암호화 한다.

(2) 현 집합 함수F(n)에서 집합F를 임의로 초기화 한다.

(3) 집합내의 매개 개체Fi의 적합함수G(fi)를 계산 한다. 적합도 함수는 매개체의 성능이 좋고 나쁨을 표시한다.

(4) 현 집합으로부터 선택된 두 구성원이 선택받을 확률은 비례적이다. 즉, 적응성이 더 높거나 선택되어질 가능성이 더 큰 적응 염색체에 선택될 확률이 비례적으로 높다.

(5) 예정된 교차확률 (crossover rate)에 의해 선택된 매개 염색체의 임의의 위치에서 교차교배를 한다.

(6) 예정된 변이확률 (mutation rate)에 의해 선택된 염색체내 유전체의 위치를 통해 상대되는 유전체를 교체 한다.

(7) 만일 조건을 만족하지 못하면 (3)으로 돌아가 계속 반복한다[8].

그림 2는 진화과정의 7단계를 표현한다.

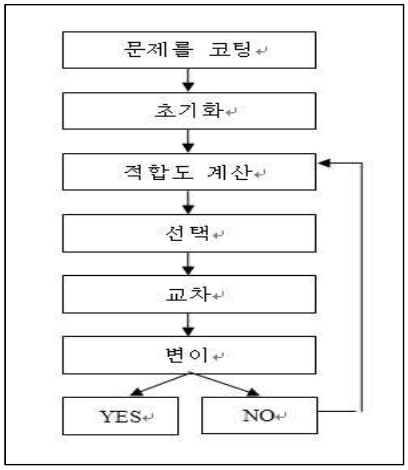

그림 2. 진화 과정
Fig. 2 Evolution Processing

3.2 유전적 연산방식 실현의 경로와 방법

3.2.1 문제풀이

그림 3은 지도를 탐색할 입구와 출구가 있으며 몇 가지의 장애물이 있는 지도탐험경로이다. 정수의 2차원배열 지도는 Scene[][]으로 나타내며 0으로 통과할 수 있는 공간을 표시하고, 7은 벽 혹은 장애물을 표시하며 8은 입구, 9는 출구를 표시한다.

우선적으로 어떤 장애물에도 걸리지 않고 나갈 수 있는 출구를 찾아야 한다. 이런 지도설계 방법은 GameScene류에 속하며 상수배열의 형식으로 지도의 출발점, 결승점을 저장한다. 지도보존 외에 이 클래스에는 GameScene MonsterWay[][]를 요구하며, NPC가 지도에서 지나간 경로를 기록하는 역할을 한다.

1(UP), 2(DOWN), 3(LEFT) 그리고 4(RIGHT)로 구성된 동작방향으로 NPC가 얼마나 갔는지 측정하고 NPC가 갈 수 있는 제일 먼 거리를 계산해 적합성함수를 얻는데 이것는 NPC의 최종위치에서 출구까지의 거리와 정비례 한다. NPC와 EXP 위치가 NPC에 근접하면 더 높은 적용점수를 얻게 된다. NPC가 사실상 출구에 도착하면 9로부터 나가게 되고, 순환은 자동적으로 끝나 하나의 집합을 얻게 된다.

0	0	0	7	0	9	0
0	0	0	0	0	0	0
0	7	0	0	7	0	0
0	0	0	0	7	0	7
0	7	0	7	7	0	0
7	0	0	0	7	0	0
0	8	7	0	0	0	7

그림 3. 지도탐험경로
Fig. 3 Routine of Exploration on Map

3.2.2 초기화

7단계에 기반한 유전학 연산방식은 첫 번째 단계 염색체 코딩을 한다. 각 염색체는 NPC방향을 따라 코드로 움직이며, NPC는 상, 하 ,좌, 우의 네가지 운동방향이 있는데 Go[] 배열의 4가지 방향을 뜻하는 염색체 움직임으로 암호화된다. 먼저 프로그램을 통해 상수인 임의 수의 1234열로 만들어진 구성은 방향에 따라 염색체 {3,2,1,2,4,3,2,1……} 등과 같은 NPC 행동을 얻게 된다. 두 번째 단계는 NPC가 지도 입구에 위치하도록 하고, NPC가 Go[]배열에 따라 명령대로 한발씩 다가간다. 만약 하나의 방향이 NPC로 하여금 벽 혹은 장애물과 부딪히면 이 명령을 무시하고 다음 명령을 실행한다. 이런 방식으로 모든 방향을 수행하거나, NPC가 출구로 나갈 때까지 진행한다. 임의로 생성된 대량의 염색체 과정에서는 몇 개의 개체가 NPC가 통로를 찾을 수 있도록 출구에 도달할 수 있다.

유전 알고리즘은 정수의 배열을 임의 숫자의 원초 집합으로 정하고 그들 각각이 NPC로부터 얼마나 멀어졌는지 시험하고 측정해 그 중에 제일 좋은 종자가 다음세대를 생성하여 그들의 자손이 출구와 가깝게 한다. 이런 방식으로 집합이 나올 때까지 반복한다. 그러므로 염색체를 포함하거나 적합성함분수와 관련이 있는 염색체 구조를 정의하여야 하며 그 구조는 아래와 같다.

```
public class Monster{
  int Go[];
  double FitnessScore;
  public void Monster() {};
  public void Monster(int num) {
        for(int i=0; i<num; ++i) {
                }
        }
}
```

염색체 대상을 만들 때 하나의 정수를 매개 변수로 하여 구조함수에 전달하면 자동으로 정수 형태 길이의 임의 배열조합이 산출되는데 이때, 적합성을 0으로 초기화하여 유전자의 설정을 완성한다.

3.2.3 적합도 계산

세 번째 단계는 유전 알고리즘에서 제일 핵심 단계인 염색체 집합중 개체의 적합성분수를 측정하는 것이다. 함수 OrderWay()는 상, 하, 좌, 우의 네개 방향을 대표하는 Go[]에 의해 NPC에서 출구와의 거리를 계산하여 하나의 적합성분수를 찾는다. 아래는 적합성분수의 계산법이다.

```
int Way_X = Math.abs(In_X - Out_X);
int Way_Y = Math.abs(In_Y - Out_Y);
return 1/(Way_X+Way_Y+1);
```

Way_X 와 Way_Y는 NPC에서 출구와의 수평편차거리와 수직편차거리이다. 만약NPC가 출구에 도착하였다면 Way_X + Way_Y = 0이다. OrderWay()는 각세대의 제일 높은 점수의 적합성을 유지하고 하나의 새로운 유전자 집합이 만들어질 때마다 모두 보존할 필요가 있다.

3.2.4 선택

네 번째 단계는 현재 집합 중에서 2개의 개체를 선택해 교차(hybridization)를 준비한다. 균등비례 룰렛휠 선택은 교차의 일반적인 방법이다. 선택될 확률은 그들의 적합성점수와 정비례하며 적합성점수가 높은 염색체일수록 선택될 확률이 높다. 하지만 이는 적합성이 높은 원소가 무조건 다음세대로 선택된다는 말이 아니라 선택될 확률이 제일 높다는 뜻이다.

```
while (Monster.fatherID< v_num)
{
  Monster Father1 = new Monster(TPick());
  Monster Father2 = new Monster(TPick());
}
```

3.2.5 교차

매번 반복되는 과정 중에 자식 염색체와 최고 점수를 갖는 부모로 될 두개의 염색체가 필요한데, 균등비례 룰렛 휠 방법은 적합성점수가 가장 높은 부모염색체를 선택한다.

```
Monster son1 = new Monster();
Monster son2 = new Monster();
public void Change(Father1.Go, Father2.Go,
 son1.Go, son2.Go);
```

3.2.6 변이

두개의 새로운 염색체를 만들어 선택된 부모 염색체와 함께 교차함수 Change()에 전달한다. 이 함수는 교차를 실행하여 새로운 염색체의 비트 문자열을 son1과son2에 보관한다.

```
public void Aliens(son1.Go){}
public void Aliens(son2.Go){}
```

3.2.7 변이판단

두 개의 새로운 비트문 라열 son1, son2가 추가 되므로서 염색체 변이된 자손 염색체를 생성한다. 이 과정을 반복 하여 염색체가 결과에 도달 하는 동안 원래의 집합은 새로운 집합으로 교체된다.

경우에 따라서 염색체가 변이된 자손은 출구를 찾을 수 없는 경우도 있다. 이 자손집합은 그림 2의 작합도 계산 단계로 돌이가 다시 계산한다. 그림 4는 Up Down, Left Right 값이 반복되면서 이동을 하여 찾지 못하는 경우 이다.

0	0	0	7	0	9	0
0	0	0	0	0	0	0
0	7	0	0	7	0	0
0	0	0	0	7	0	7
0	7	0	7	7	0	0
7	0	0	0	7	0	0
0	8	7	0	0	0	7

그림 4. 출구를 찾지 못하는 경우의 예
Fig. 4 Example as not finding exit

Ⅳ. 결론

본 연구는 유전학적 기반의 인공지능 기법을 적용한 게임 NPC 행동 연구를 통해 다양한 행동경로를 제공하도록 설계하였다.

일방적인 게임 NPC 행동경로는 개발 초기에 결정되므로 게임의 진행이 미리 정의된 경로로만 단순화되는 문제가 있어 본 논문은 문제해결을 위해 유전학적 기법을 통해서 게임 NPC의 기본 행동경로를 정의하였다. 이러한 결과 게임 NPC의 행동경로는 전체 게임 맵에 표현되었으며, 경로 다양화를 통해 게임 사용자에게 반응되어지는 행동경로가 달라짐에 따라 게임 형태요구에 보다 높은 만족도를 제공할 것으로 보인다.

향후 유전자 알고리즘에 대한 시험과 행동경로 및 사용자 만족도와의 상관관계에 관한 연구가 필요하다.

참고문헌

[1] Judea Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, *Morgan Kaufmann Publishers, Inc.* 2004.

[2] Steve Rabin, *AI Game Programming Wisdom*, Charles River Media, 2008.

[3] Mat Buckland, *AI Techniques for Game Programming*, Premier Press, 2009.

[4] M. Tim Jones, *AI Application Programming*, Charles River Media, 2007.

[5] Stephen Cass, *Mind Games: to Beat The Competition Video Games are Getting Smarter*, IEEE Spectrum Online, 2002.

[6] Daniel Johnson and Janet Wiles, *Computer Games With Intelligence*, IEEE International Fuzzy Systems Conference, 2001.

[7] Mark DeLoura, *Game Programming Gems 2*, Charles River Media, 2001.

[8] Fuzhaohui and Dingmeng, *AI Techniques and Its Application in Game Programming*, HuNanXuBao, 2005

저자소개

何俊(he jun)

2005년 한남대학교 디자인학과 학사
2008년 공주대학교 게임디자인학과 (공학석사)
2009년~현재 공주대학교 컴퓨터공학부 박사과정
※ 관심분야 : 게임인공지능

김재웅(Jae-Woong Kim)

1988년 중앙대학교 대학원 컴퓨터공학과 (이학석사)
2000년 대전대학교 대학원 컴퓨터공학과 (공학박사)
1992년~현재 공주대학교 컴퓨터 공학부 교수
※ 관심분야 : 소프트웨어 공학, 게임 소프트웨어 개발 방법론