

EIC(Evolutional Intelligent Character) 모델을 이용한 지능적인 실시간 게임 캐릭터의 구현

강성관**, 안태홍**, 김국송**, 김종혁**, 김흥기*
**전남과학대학 게임 제작과, *동신대학교 컴퓨터학부
**Dept. of Game Product Chunnam Techno College

Abstract

In the majority of today's computer games, the behaviour of characters are controlled by pre-defined game logic or pre-generated motion. As game developers strive for richer and more interactive games, they often encounter limitations with this approach.

This paper attempts to construct a game model using Genetic Algorithms (GAs) in order to produce more intelligent and compelling computer games. Based on learning ability, the use of GAs will enable the characters to continually evolve, providing a changing and dynamic game environment. A real-time game was implemented to investigate the performance and limitations of the system.

요 약

오늘날 컴퓨터 게임의 대부분은 캐릭터의 행동 유형을 게임 로직에서 미리 정의된 게임 로직이나 초기에 설정된 움직임에 대한 정보를 이용하여 제어한다. 게임 개발자들은 더 풍부하고 다양한 내용을 담고 있고 뛰어난 환경 판단 및 대처 능력 등을 갖는 게임을 개발하기를 원하기 때문에 이러한 방식은 한계에 직면하게 된다.

본 논문에서는 지능적이고 게임 플레이어로 하여금 다양한 흥미를 유발할 수 있는 컴퓨터 게임의 개발을 위해서 유전자 알고리즘을 사용한 게임 모델을 구성하였다. 학습 능력에 기초한 유전자 알고리즘을 사용함으로써 변화 가능성과 동적인 게임 환경을 고려하면서 계속적으로 진화하는 캐릭터를 만들 수 있을 것이다. 실시간 게임은 제한한 시스템의 실행과 한계를 연구하기 위해 실행되었다.

1. 소개

오늘날 게임 개발자들이 직면해 있는 주요 관심사는 게임 플레이어를 위해 더 풍부하고 다양한 내용을 담고 있고 뛰어난 환경 판단 능력 및 대처 능력 등을 갖는 인공지능형 게임을 더욱 많이 만들어 내는 것이다. 오늘날 컴퓨터 게임의 대부분에서 캐릭터의 움직임은 룰[1]에 의해 미리 정의되어 지거나 애니메이션 시스템이나 움직임 캡처에 기초한 키 프레임틀 사용하여 미리 설정되어진다. 그러나 게임 개발자들은 환경에 대한 대처 능력을 갖는 지능적인 게임을 개

발하기를 원하기 때문에 그러한 시도는 제한적인 상호작용 능력, 제어하기 어려운 복잡도 문제, 인력, 자금, 시간의 소요 등에서 몇몇 심각한 한계에 부딪힌다.

본 논문에서는 컴퓨터 게임에서의 인공지능 기술에 대한 현재의 접근방식들을 검토할 것이다. 유전자 알고리즘을 사용한 지능적인 게임 모델에 대해 자세하게 설명하고 이를 구성하게 될 것이다. 실험적으로 제작된 실시간 게임에서 유전자 알고리즘의 적용을 통해서 수행과 모델의 제한성을 관찰하고 분석할 것이다.

II. 컴퓨터 게임에서의 인공지능

일반적으로 게임 플레이어들은 실제적인 경험을 할 수 있고, 실제와 비슷한 모의실험이 가능하며 다양한 흥미를 제공할 수 있는 게임을 즐긴다[2]. 달리 말하면, 그들은 실제와 유사한 환경에서 지능적인 적과 플레이하기를 원한다. 그러나 미리 정의되어진 게임 알고리즘은 자주 게임 캐릭터들과 게임 플레이어나 게임 캐릭터들과 게임 환경 사이의 상호 연관성을 제한한다. 현재 게임 인공지능의 두 가지 주요한 접근방식은 룰 기반 인공지능과 행동 양식이다.

2.1 룰 기반 게임 인공지능

룰 기반 인공지능은 대부분의 게임들에서 적의 지능에 대한 기초가 된다. 보통 룰은 만약 적에 의해 추적을 당하게 된다면 달아나게 되는 것처럼 조건(자극)과 행동(응답)으로 구성된다.

2.2 미리 정의된 행동 유형

미리 정의된 행동 유형들은 컴퓨터 게임에서 지능을 지원 하는 곳에 자주 사용된다. 유형은 임무를 수행하기 위한 단계들의 순서이고 일정한 조건이 발생할 때 유발될 수 있다. 이것은 행동 유형이고 게임의 인공지능에서 겉보기에 복잡한 생각을 처리하는 좋은 방법이다. 사실, 오늘날 많은 게임은 아직 게임 로직의 대부분이 그러한 양식들을 사용한다 [3].

그러나 룰 기반의 인공지능과 행동 기반의 인공지능 모두 유연성과 다양성 면에서 분명한 한계성들을 갖는다. 모든 가능한 룰과 행동 유형의 준비는 실제 게임에서 많은 한계를 가진다. 많은 행동 양식들과 룰들이 준비 되었을 때라도 그런 알고리즘을 가지고 아주 많은 가능성의 조합을 제어 하는 것은 어렵다

III. 컴퓨터게임들에서 유전자알고리즘

많은 게임 개발자들은 그들 게임들에서 유전자 알고리즘의 사용을 도입하기 시작하였다. 오늘날 컴퓨터 게임에서 유전자 알고리즘의 주요 응용들은 실시간 게임보다는 체스 [4], 카드[5], 정치 게임[6]과 경제 전략 게임과 같은 전략 게임들에 집중되어 있다. 현재 이러한 유형의 게임들에서 유

전자 알고리즘이 적용된 게임들로는 Creatures와 Cloak, Dragger과 DNA(CDDNA)을 예로 들 수 있다.

Cyberlife의 Creatures는 화학적 기반의 유전자 알고리즘과 Nom으로 불리는 가상 애원동물을 제어하기 위한 신경망의 결합을 사용한다. 새로운 특성을 만들어내는 능력은 게임에서 가장 인기있고 흥미있는 부분의 하나가 될 것이다. 그러나 이 게임은 게임의 필수적인 특성들인 분명한 목적을 포함하지 못하고 경쟁 측면이 부족하기 때문에 실제의 컴퓨터 게임보다 시뮬레이션으로 분류된다.

Oidian System의 Cloak, Dagger와 DNA (CDDNA)는 전략 게임으로 유전자 알고리즘을 사용하여 적 캐릭터를 더욱 지능적으로 만들었다. 플레이어는 그들의 게임 일부를 실제 움직이지 않지만, 게임의 일부를 움직이거나, 건설하거나, 해산하기 위한 명령의 목록을 제출한다. 컴퓨터 플레이어는 동일한 종류의 명령들을 만드는 유전자와 같은 특별한 프로그램을 동작시킴으로써 게임을 플레이한다.

유전자 알고리즘[7]은 자연적인 진화의 과정으로부터 행동 유형을 얻는 기계 학습의 접근과 같은 많은 인공지능의 응용을 위해 사용되어 진다.

인공지능의 기법에는 신경망, 퍼지, 유전자 알고리즘 등을 사용한 것들이 있다. 일반적으로 신경망을 학습 및 데이터 검색 등에서 많이 이용되며 퍼지 시스템은 불확실한 데이터로부터 해답을 얻고자 하는 곳에서 이용된다. 이 중에서 유전자 알고리즘은 수많은 데이터들 중에서 원하는 데이터를 검색하거나 local minimum에 빠진 상태를 벗어나기 위한 방법의 하나로써 사용되어지고 있다.

이 연구의 목적은 더욱 지능적인 게임을 만들기 위한 목적의 실시간 게임에서 캐릭터의 행동을 위한 유전자 알고리즘을 적용하는 것이다. 유전자 진화를 사용한 진화 설계 처리 시스템인 EDGE 시스템[8]은 새로운 시스템에서 진화 엔진으로 사용되고 변형되었다.

IV. 지능적인 진화 게임

컴퓨터 게임에서 유전자 알고리즘의 특징에 기초하여 진화하는 지능적인 게임(EIG) 모델을 구성하였다. 유전자 알고리즘의 학습 능력[4]을 사용한 이 모델은 생존하기 위해 그들 환경으로부터 학습이 가능한 캐릭터의 계속적인 진화를 지원한다.

4.1 진화하는 지능적인 캐릭터(EIC)의 시스템 모델

EIC 모델은 유전자 알고리즘과 객체 지향 데이터 구조를 기반으로 구성되었다. 전체 시스템은 그림 1과 같이 게임 요소 데이터베이스(GEDb), EIC 엔진과 실시간 게임 플레이 필드(RPF)의 3가지의 주요한 모듈로 구성되었다. 이 엔진은 모델에서 문제에 독립적인 모듈이다. 그러므로 다른 형태의 게임들에 적용이 가능하다. 이 모듈은 EDGE 시스템으로부터 진화 처리와 개체의 행동에 대한 유전자 표현 등의 모두의 연구 결과로 이용된다[9].

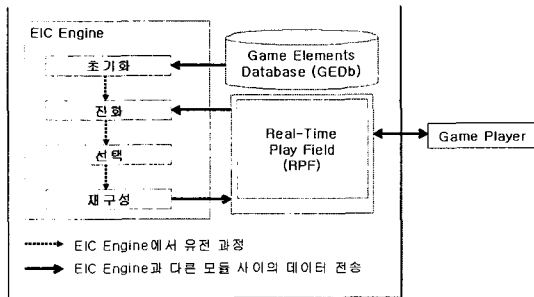


그림 1. EIC 처리 모델

4.1.1 게임 요소 데이터베이스 (GEDb)

게임 플레이를 시작했을 때, 게임 프로그램이나 게임 플레이어는 그림 1처럼 GEDb로부터 캐릭터들을 포함하는 게임의 요소들의 집합을 선택하고 검색하여 그것을 EIC 엔진으로 전송한다. 그들 요소들은 게임 개발 과정동안에 게임 개발자들에 의해 준비되어지고 객체로써 저장되어 진다. 검색된 요소들 중에서 진화가 필요한 행동들은 유전자 구조에 의해 정의되어진 유전자형으로 부호화되어진다[9]. GEDb는 숙련된 플레이어들과 유전자 알고리즘이 적용되지 않은 다른 알고리즘들을 위해 미리 진화하거나 미리 훈련된 캐릭터들을 포함할 수 있다. 객체 지향 데이터 구조를 사용함으로써 게임 개발자에게 효과적으로 게임 개발에 관한 개념, 시각화, 게임요소를 프로그램하는 것에 도움을 줄 수 있다.

4.1.2 EIC 엔진

그림 1에서 초기화 처리는 게임을 시작하거나 재설정할 때 요구되어진다. 이 처리는 다음의 처리에서 캐릭터들의 개별적인 집단을 지원한다.

평가 처리는 개별적인 행동의 적응도를 계산한다. 이러한 처리의 결과는 선택 처리에서 기초 자료로 사용된다.

선택 처리는 그들이 적응도에 기초한 이전의 집단으로부터 개별적인 새로운 집단을 선택한다. 그리고 선택된 캐릭터들은 재구성 처리로 보내진다.

재구성 처리는 교차와 돌연변이 연산을 통해서 개별적인 행동 유형이 변경된다. 변환된 해들은 RPF로 보내진다.

새로 생성된 캐릭터들은 RPF에서 그들의 행동 유형에 따라 행동한다. 정해진 시간이나 스테이지 후에 캐릭터들은 그들의 적응도 평가를 위해 다시 평가 처리로 보내진다. 이러한 반복적인 처리를 통해서 캐릭터들의 행동 유형은 진화하고 세대가 계속된다.

4.1.3 실시간 플레이 필드 (RPF)

RPF는 게임 플레이어가 게임 객체를 지켜보고 게임을 플레이하는 곳이다. 교차와 돌연변이에 의해 캐릭터 처리의 행동양식이 변화되고 게임 플레이어나 다른 캐릭터들에 대응하여 동작하고 감응한다.

4.2 특징 정보의 표현

캐릭터 유전자 구조(CGs)[10]은 이러한 모델을 위해 설계되고 사용된다. 이 구조는 유전자형에서 캐릭터의 행동 정보를 표현하는 체계이다. CGs는 유전자 알고리즘에 의해 변화되고 진화된 행동만을 포함한다. 유전자는 CGs의 인스턴스이다.

유전자형이 유전자 집합이고 일반적으로 이진 문자열로 구성된 유전자 코드에서 객체의 전체 구조를 표현한다. 그것의 물리적인 정보를 의미하는 유전자형의 변환을 통해서 다양한 해들의 생성과 빠른 변환 등과 같은 많은 이득을 얻는다[9].

표현형은 유전자형의 물리적 표현이다. 게임 환경에 대한 캐릭터의 행동 양식의 적응도를 평가할 수 있다. 평가 처리는 이 레벨에서 수행된다. 표현형의 그래픽 표현이 화면상에서 플레이어를 위해 표시된다.

그림 2는 유전자 알고리즘으로 먼저 각 캐릭터들에 대한 정보를 문자열로 코드화를 하고 이를 하나의 집단으로 설정한다. 다음으로 문자열에 대한 적응도를 평가하고 평가된 적응도에 따라 적응도가 낮은 객체는 도태시키고 높은 객체만을 선택하여 새로운 객체로 선택하게 된다.

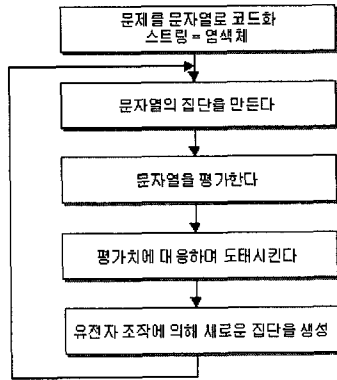


그림 2. 유전자 알고리즘

V. 수행

EIC 모델이 유전자 알고리즘에 기반한 게임 엔진의 수행을 시뮬레이션하기 위해서 간단한 2차원 실시간 게임에서 수행된다.

그림 3와 같은 게임은 두 종류의 캐릭터를 포함하고 있다. 그것은 게임에서는 약하지만 좋은 캐릭터인 10개의 Bleu Bean(BB)와 BB를 사냥하는 2개의 Ghost로 구성되어 있다. BB들을 추적하는 정해진 알고리즘은 Ghost를 제어한다. BB들은 추적하는 Ghost로부터 달아나기 위해 유전자 알고리즘 기반의 인공지능을 사용한다. 그러므로 게임의 목적은 가능한 BB를 공격하는 Ghost로부터 구조하는 것이다. 플레이어가 이 게임을 실행했을 때, EIC 엔진은 외면상 2개의 Ghost인 컴퓨터 캐릭터에 적용할 수 있다.

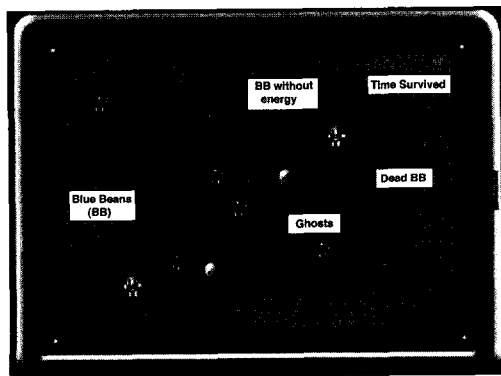


그림 3. 모델에 기반한 예제 게임. 2개의 Ghost가 10개의 BB를 추적하고 있다. BB는 에너지가 떨어지면 희미해진다

5.1 BB(Blue Bean)의 진화

각 BB 객체는 동작을 시작했을 때 결정되는 것(Ghost로부터의 거리를 고려하여), 8개의 방향과 속도를 변경하는 것의 3가지의 스스로 제어 가능한 행동 양식을 가지고 있다. 그들은 3개의 행동양식을 제외한 어떠한 지식도 가지고 있지 않는다. 각 BB는 제한된 에너지를 제공하고(제어가 불가능한) 전체의 움직임은 속도의 동작 거리에 비례하여 많은 양의 에너지를 소모한다. 그러므로 모든 BB는 움직임을 효과적으로 제어할 필요가 있다. BB는 이전 스테이지의 어떠한 전략에 대한 지식도 알지 못한다. 게임의 진행 중에 그들은 경험으로부터 학습을 시작한다. 진화 처리는 개별적인 캐릭터가 변경되어지는 유전자형을 유지하고 더 구조화되고 지능적인 유전자형을 만든다.

5.2 Ghost 캐릭터

Ghost 캐릭터는 FSM에 기초하여 행동하거나 움직인다. RPF에서 Ghost는 BB를 찾고, 따라다니고, 죽이고 나면 다른 BB를 찾는다. Ghost에서 자극, 내부 상태와 응답의 기본적인 구조는 아래와 같다.

- 준비되었으면 BB를 찾는다
- BB가 발견되면 BB를 추적한다
- BB가 잡히면 BB를 죽인다
- BB가 죽으면 대기 상태가 된다

Ghost는 정해진 추적 알고리즘에 기초하여 BB를 추적한다. 현재의 제작된 게임에서 추적 알고리즘은 입력으로 BB(Bx, By)와 Ghost (Gx, Gy)의 공간 좌표를 사용하고 아래와 같이 BB의 이동에 따라 Ghost를 움직이게 한다.

```

if (Bx < Gx) Gx++; else if (Bx < Gx) Gx--;
if (By < Gy) Gy++; else if (By < Gy) Gy--;
    
```

5.3 검색과 추적에의 유전자 알고리즘 사용

유전자 알고리즘을 사용함으로써 Ghost가 BB를 최대한 빠른 시간 내에 검색하여 쫓아가는 최단거리 알고리즘을 구현하였고, 반대로 BB가 Ghost로부터 벗어날 수 있는 최단거리 알고리즘을 구현함으로써 게임상의 캐릭터들의 실시간으로 움직일 수 있도록 하였다.

5.4 학습

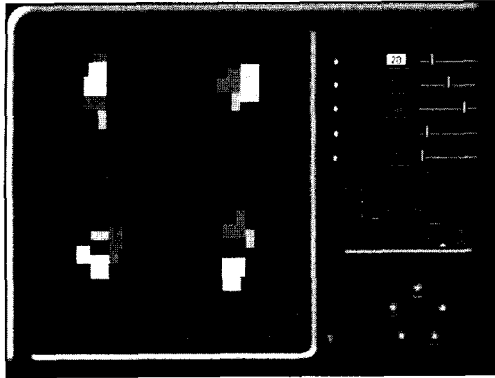


그림 4. 유전자 알고리즘의 설정

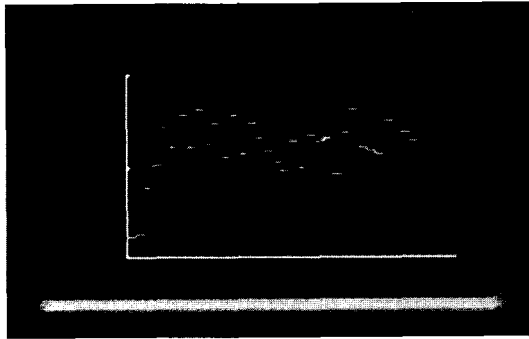


그림 5. 유전자 알고리즘의 수행 그래프

제작된 게임은 그림 4처럼 유전자 알고리즘의 파라미터들을 설정하고 시작한다.

그림 4에서 파라미터의 설정은 처리가 50개의 세대나 스테이지가 있는 10개의 BB를 가지고 시작함을 보여준다. 모든 채구성 처리에서 각 캐릭터는 다른 하나와 약 90%의 교배 기회를 가질 수 있다. 그리고 각 유전자를 위한 위치에서 가져오는 돌연변이의 기회는 대략 3%이다.

진화가 시작되었을 때, 캐릭터의 평균 적응도나, 지능은 매우 낮다(38초의 구조시간). 그림 3에서와 같이 진화 처리가 수행되면 지능은 극적으로 상승된다(15세대 동안 340초의 구조시간). 50세대를 통해 가장 좋은 적합도의 값은 461초이다. 우수한 종의 유전자형은 GEDb에 저장되고 추후에 잘 훈련되거나 지능적인 캐릭터로 사용될 수 있다. 개선은 처리가 수렴점에 도달했을 때 중지된다. 그러나 수렴 문제는 유전자 알고리즘의 응용에 대한 평가 문제들이 매우 흔히 발생한다. 이러한 방해물은 좋은 집단을 보존하는 것으

로 극복될 수 있다.

VI. 결론

본 논문에서는 더 지능적이고 효과적인 게임 인공지능을 구현하기 위해 실시간 게임엔진에 유전자 알고리즘을 도입하였다. 일반적인 유전자 알고리즘의 처리와 표현이 EIC 엔진에서 적용되었다. 모델은 테스트 게임을 사용하여 수행되었고 그것의 수행을 분석하였다.

연구의 결과와 수행은 게임 인공지능에서 유전자 알고리즘의 사용은 매우 희망적인 결과와 새로운 모델에 대한 이득적인 측면을 보여준다.

향후 연구과제로는 퍼지나 신경망을 이용한 자동 룰 생성 알고리즘의 적용 및 다양한 게임 알고리즘에의 유전자 알고리즘을 적용한 연구가 필요할 것이다.

VII. 참고문헌

1. Badler, N.I., Barsky, B.A. and Zeltzer, D. (ed.): Making them move: mechanics, control, and animation of articulated figures, Morgan Kaufmann, San Mateo (1991)
2. Duntemann, J.: Breathing life into your arcade game sprites, PC Techniques, 5(2) (1994) 89
3. Holland, J.H.: Adaptation in Natural and Artificial Systems, MIT Press (1975)
4. Reif, J.H.: The complexity of two-player games of incomplete information, Journal on Computers and System Science, 29(2) (1984) 274-301
5. Tzeng, C.H. and Purdom, P.W.: A Theory of Game Trees. AAAI 1983: (1983) 416-419
6. Vincke, S.: The ORC problem, Game Developer (1998)
7. LaMothe, A.: Building Brains into Your Games, Game Developer Magazine, August (1995)
8. Grefenstette, J.J.: Optimisation of control parameters for genetic algorithms, IEEE Trans SMC, 16 (1986) 122-128
9. Jo, J.H. and Gero, J.S.: Space layout planning using an

evolutionary approach, Journal of Artificial Intelligence in Engineering, 12(3) (1998) 149-162

10. Jo, J.H.: Character Gene Schema for Evolutionary Intelligent Game Engine, working paper, Griffith University (2000)



강성관

1997년 2월 조선대학교 전자공학과 (공학사)
 1999년 2월 조선대학교 대학원 (공학석사)
 1999년 3월 ~ 조선대학교 대학원(박사과정)
 1999년 3월~2000년 2월 조선이공대학 겸임교수
 2001년 3월 ~ 전남과학대학 겸임교수
 관심분야: 이미지프로세싱, 인공지능, 패턴인식, 게임엔진, 가상현실, 애니메이션 등



김국송

1997년 창원대학교 토목공학과(공학사)
 2001년 8월 아주대학교 대학원 컴퓨터공학과(공학 석사)
 2000년 ~ 현재 웹도 프로그램 팀장
 2002년 ~ 현재 전남과학대학 게임제작과 겸임교수

김종혁

2002년 현재 (주)한호홍업 게임사업본부 개발이사
 단국대학교 방송영상학과 강사
 전남과학대학 게임제작과 겸임교수



안태홍

1987년 2월 조선대학교 전자공학과 (공학사)
 1992년 8월 조선대학교 대학원(공학석사)
 1998년 2월 조선대학교 대학원 (공학박사)
 1993년 ~ 전남과학대학 게임제작과 학과장
 관심분야: 이미지프로세싱, 인공지능, 패턴인식, 게임엔진, 가상현실, 애니메이션 등



김홍기

1984년 전남대학교 계산통계학과 (이학사)
 1986년 전남대학교 대학원 (이학석사)
 1996년 전남대학교 대학원 (이학박사)
 1991년 ~ 동신대학교 컴퓨터학과 부교수
 관심분야: 공간데이터베이스, 컴퓨터그래픽스, 멀티미디어시스템