

논문 2005-42TE-3-6

상대캐릭터의 행동패턴에 적응하는 지능캐릭터의 구현

(Implementation of Intelligent Characters adapting to Action Patterns of Opponent Characters)

이 면 섭*, 조 병 현**, 정 성 훈***, 성 영 락****, 오 하 령****

(Lee Myun-Sub, Cho Byeong-heon, Jung Sung-hoon, Seong Yeong-rak, and Oh ha-ryoung)

요 약

본 논문에서는 유전자 알고리즘을 이용하여 특정 행동 패턴을 보이는 상대 캐릭터에 적절히 적응할 수 있는 대전 액션 게임용 지능캐릭터의 구현 방법을 제안한다. 이를 위하여 현재 행동과 단계뿐 만 아니라 과거행동까지 고려하여 학습한다. 제안한 방법이 얼마나 효과가 있는지 알아보기 위하여 현재의 행동과 단계만을 적용한 실험(실험-1)과 과거 행동을 추가한 행동패턴을 적용한 실험(실험-2) 결과를 비교 분석하였다. 실험 평가는 두 캐릭터가 획득한 점수를 측정하여 그 비로써 평가하였다. 실험 결과 초기에는 실험-1에서 높은 점수비로 시작하지만 일정 세대 이후부터는 실험-2의 점수비가 좋아지며, 실험-2에서는 모두 최적해를 찾을 수 있었다. 또한 실험-2에서 지능 캐릭터는 행동이 완료되는 시점에서 최대의 점수를 얻기 위해 이동(전진, 후진)이나 시간 지연 동작을 하여 스스로 진화하면서 게임 규칙을 학습함을 보였다.

Abstract

This paper proposes an implementation method of intelligent characters that can properly adapt to action patterns of opponent characters in fighting games by using genetic algorithm. For this intelligent characters, past actions patterns of opponent characters should be included in the learning process. To verify the effectiveness of the proposed method, two types of experiments are performed and their results are compared. In first experiment(exp-1), intelligent characters consider current action and its step of an opponent character. In second experiment (exp-2), on the other hands, they take past actions of an opponent characters into account additionally. As a performance index, the ratio of score obtained by an intelligent character to that of an opponent character is adopted. Experimental results shows that even if the performance index of exp-1 is better than that of exp-2 at the beginning of stages, but the performance index of exp-2 outperforms that of exp-1 as stages go on. Moreover, optimum solutions are always found in all experimental cases in exp-2. Futhermore, intelligent characters in exp-2 could learn moving actions (forward and backward) and waiting actions for getting more scores through self evolution.

Keywords : genetic algorithm, artificial intelligent, intelligent character, game, actions pattern

I. 서 론

인공지능을 이용한 대표적인 게임으로는 Half Life, Age of Kings, Black & White, The Sims 등이 있다. Half Life^[1,2]는 일련의 행동을 스케줄로 묶어서 처리한

방식으로 FSM(Finite State Machine)을 개념에 따라 계층으로 나누어 처리하였고, Age of Kings^[3]는 룰 베이스를 적용한 전략 게임이다. Black & White^[4,5]는 인공 생명 기법을 적용하였으며 Decision Tree 방식을 적용하여 코드를 바꾸지 않고도 외부에서 변경 가능한 구조로 되어 있다. The Sims^[6,7]는 인공 생명과 FSM을 적용하였으며, 행동에 대한 정보들을 각각의 객체가 가지고 있으면서 캐릭터에게 여러 행동에 대한 이벤트를 발생시키는 객체지향 방식을 사용하였다. 그러나 이러한 방법들은 게임에서 사용자가 조작하지 않는 NPC(Non Player Character)가 스스로 학습하고 진화하는 지능 캐릭터와는 거리가 있다.

논문 [8,9]에서는 보다 진보된 지능형 게임을 개발하기 위하여 개체가 스스로 게임의 규칙을 학습하고 또한

* 정회원, 인천전문대학 컴퓨터정보과
(Department of Computer & Information Science, Incheon City College)

** 학생회원, 전자통신연구원 디지털콘텐츠연구단
(Digital Content Research Division, ETRI)

*** 정회원, 한성대학교 정보공학부
(Department of Information and Communication Engineering, Hansung Univ.)

**** 정회원, 국민대학교 전자공학과
(School of Electrical Engineering, Kookmin Univ.)

접수일자: 2005년3월14일, 수정완료일: 2005년9월7일

주어진 환경에 스스로 적응하며 진화하는 새로운 종류의 지능 캐릭터를 제안하였다. 제안한 지능 캐릭터는 게임의 규칙을 모르는 상황에서도 상대 캐릭터와 게임을 진행해 나가면서 획득한 점수로 스스로 자신의 지식을 향상시킬 수 있었다. 이를 위하여 유전자 알고리즘으로 게임 규칙에 대한 지식을 학습하였다. 그러나 기존 연구[8,9]에서는 상대 캐릭터의 현재의 행동과 단계만을 이용하여 학습하기 때문에 특정 행동패턴을 보이는 상대 캐릭터에 최적에 가깝게 대응하지 못하는 문제가 있다. 예를 들어 게이머나 상대 캐릭터가 특정한 행동을 반복하거나 아니면 특정한 거리에서는 특정한 행동을 반복하는 경우가 있다. 이러한 경우 문제를 해결하기 위하여 본 논문에서는 상대 캐릭터의 행동과 단계뿐만 아니라 과거의 행동까지 고려하여 학습하게 함으로서 행동패턴에 대한 대응이 가능한 지능 캐릭터를 구현하였다.

제안한 방법을 검증하기 위하여 일정한 행동패턴을 가지는 상대캐릭터와 게임을 하고 지능캐릭터가 획득하는 점수와 상대 캐릭터가 획득하는 점수를 기록하여 점수비로서 지능캐릭터의 대응 능력을 측정하였다. 기존 방법^[8,9]과의 비교를 위하여 실험-1에서는 상대 캐릭터의 현재 행동과 거리 그리고 단계를 기준으로 지능 캐릭터가 학습하고, 실험-2에서는 상대 캐릭터의 현재 행동과 거리 및 단계뿐만 아니라 과거 행동까지 고려하여 학습하였다. 그리고 점수비를 이용하여 실험-1과 실험-2를 비교 분석하였다. 실험결과 초기에는 실험-1의 점수비가 좋으나 일정 세대 이후에는 실험-2의 점수비가 더욱 좋아 지는 것을 관찰할 수 있었다. 또한 실험-1에서는 상대 캐릭터의 행동패턴에 대한 최적의 대응을 찾지 못하였으나 실험-2에서는 행동 패턴 5가지에 대해서 모두 최적의 해를 찾았다. 이러한 결과는 학습시 상대 캐릭터의 과거 행동까지 고려함으로써 일종의 예측적인 대응이 가능한 것에 기인한다. 본 논문에서 제안한 방법이 행동 패턴이 없는 경우에도 적용되는 지를 알아보기 위하여 무작위적으로 행동하는 캐릭터와의 게임도 실험하였다. 실험결과는 무작위적으로 행동하는 상대 캐릭터에서도 이전의 방법보다 더욱 좋은 점수비를 보였다. 이러한 결과는 본 논문에서 제안한 방법이 상대 캐릭터가 특정 행동 패턴을 보이는 경우나 그렇지 않은 경우 모두에 유용한 방법으로 이전 방법보다 진일보한 방법임을 확인해 준다.

본 논문의 구성은 다음과 같다. II장에서는 행동패턴을 학습하는 지능 캐릭터에 대하여 설명하고, III장에서는 게임 및 지능 캐릭터의 구현에 대해 기술한다. IV장에서는 실험 결과를 분석하며 V장에서 결론을 맺는다.

II. 행동패턴을 학습하는 지능 캐릭터

본 논문에서 제안한 지능 캐릭터는 상대 캐릭터의 현재의 행동 뿐 만 아니라 과거의 행동도 학습함으로써 상대 캐릭터가 일정한 행동패턴을 보일 때 효과적으로 대응할 수 있도록 하였다. 본 절에서는 상대 캐릭터의 현재 행동과 더불어 과거 행동을 학습하는 것이 어떠한 장점을 가져다주는 지를 설명한다.

연구 [8,9]에서는 게임을 단순화하기 위하여 각 캐릭터의 행동을 1클록으로 종료하도록 하였으며 [10,11]에서는 보다 실제 게임과 유사하게 하기 위하여 캐릭터의 행동을 여러 단계로 세분화하여 해당 행동의 정해진 단계가 모두 지나야 공격이 완료되게 하였다. 그러므로 연구 [8,9]에서는 상대 캐릭터의 행동만을 가지고 지능 캐릭터를 학습시켰으며, 연구 [10,11]에서는 상대 캐릭터의 행동만을 고려하는 것이 아니라 현재 진행되고 있는 상대 캐릭터의 행동이 몇 단계에 해당되는지도 파악을 하여 상대 캐릭터의 행동 종료시점보다 먼저 종료되는 공격을 해야 점수를 얻을 수가 있었다. 그러나 논문 [8,9,10,11] 모두, 상대 캐릭터의 과거 행동은 고려하지 않았다. 지능 캐릭터가 상대 캐릭터의 현재행동과 더불어 과거 행동을 추가하여 학습하게 되면 그렇지 않은 경우 보다 다음과 같은 상황에서 더욱 효과적인 대응을 할 수 있다. 다음은 과거 행동을 추가로 고려할 경우 지능 캐릭터가 더 좋은 대응 행동을 선택할 수 있는 이유이다.

지능 캐릭터나 상대 캐릭터가 모두 공격행동을 하고 있을 때는 먼저 공격의 마지막 단계에 이른 공격을 한 캐릭터가 점수를 획득하며 그렇지 않은 캐릭터의 공격은 무효화 되어 점수를 얻지 못한다. 만약에 두 캐릭터의 공격이 동시에 마지막 단계에 이르면 두 캐릭터는 각각 자신의 공격 점수를 획득한다. 물론 모든 공격은 유효거리를 따져서 유효거리 밖에서 이루어진 공격은 점수를 인정하지 않는다. 이와 같이 두 캐릭터가 서로 공격을 하고 있는 경우에는 지능 캐릭터나 상대 캐릭터의 점수를 결정하는 것은 오로지 두 캐릭터의 현재 공격이 무엇인지, 거리가 어떤지, 어떤 캐릭터의 공격 단계가 몇 단계에 있는지에만 의존하며 두 캐릭터의 과거 행동이 무엇이었던지는 관련이 없다.

* 이후로 실험-1과 같이 학습된 지능캐릭터를 단순지능캐릭터로 실험-2와 같이 학습된 지능캐릭터를 행동패턴을 학습하는 지능 캐릭터로 부른다.

그러나 상대 캐릭터가 공격을 하지 않고 이동행동을 하는 경우에는 상황이 달라진다. 이 경우에는 상대 캐릭터의 행동은 이동으로서 지능 캐릭터는 상대 캐릭터의 이동행동 및 거리만으로 자신의 행동을 결정해야 한다. 과거의 행동을 고려하지 않고 오로지 상대 캐릭터의 현재 행동만으로 지능 캐릭터의 행동을 결정하는 경우에는 상대 캐릭터의 이동행동 및 거리만으로는 효과적인 지능캐릭터의 행동 결정을 할 수 없다. 왜냐하면 이동행동은 상대 캐릭터가 어떻게 공격해 올지에 대한 정보를 갖고 있지 않아서 지능 캐릭터가 이에 대처할 수 있는 많은 정보를 제공하지 못하기 때문이다. 그러나 상대 캐릭터가 이동을 포함한 공격에서 어떤 패턴을 보일 경우 지능 캐릭터가 현재 이동행동과 더불어 과거 행동을 고려함으로써 상대 캐릭터의 이동행동 다음에 행동을 어느 정도 파악할 수 있고 이에 따라서 적절히 대응할 수 있다. 실제 게임을 하는 사람의 경우 그 사람만의 독특한 행동방식이 있는 경우가 있다. 예를 들어 특정한 행동을 반복하거나 아니면 특정한 거리에서는 특정한 행동을 반복하는 경우이다. 이런 경우에 아래와 같이 두 캐릭터 간의 거리가 5일 때 상대 캐릭터가 다음과 같은 순서대로 행동한다고 가정하자.

후진, 아래발, 전진, 아래발, 후진, 아래주먹, 전진, 위주먹*

이 패턴에 대한 단순 지능 캐릭터와 행동패턴을 학습하는 지능 캐릭터의 단계별 행동을 그림 1과 같이 표시하였다.

그림 1의 (a)는 단순 지능 캐릭터의 행동이고, (b)는 행동패턴을 학습하는 지능 캐릭터의 행동이다. 그림(a)에서 상대 캐릭터는 후진과 전진을 교대로 2번 수행하기 때문에 지능 캐릭터가 이동하지 않는다면 두 캐릭터 간의 거리를 계속 5로 유지한다.

상대 캐릭터의 과거 행동을 고려하지 않는 단순지능 캐릭터는 첫 번째 후진(1단계)과 두 번째 후진(3단계)에 대해 상대 캐릭터의 행동과 거리가 같기 때문에 같은 아래발 공격을 하게 된다. 1단계의 상대캐릭터가 후진 후에 아래발 공격을 하면 지능 캐릭터의 아래발 공격이 먼저 종료되므로 상대 캐릭터의 첫 번째 후진 다음의 아래발 공격이 무효화되어 지능 캐릭터가 점수를 획득한다(1단계).

그러나 두 번째 후진 (3단계)에서는 다음의 아래주먹

* 주먹공격 유효거리 0-4, 발공격 유효거리 3-7, 옆차기 유효거리 6-10 이고 아래주먹(2틱 1점), 위주먹(3틱 2점), 아래발(4틱 3점), 위발(5틱 4점), 옆차기(6틱 5점)라고 가정한다.

1단계	거리	5-6	6	6	6	6	
	상대	후진	아래발	아래발	아래발	아래발	
2단계	거리	6-5	5	5	5	5	
	상대	전진	아래발	아래발	아래발	아래발	
3단계	거리	5-6	6	6	6-5		
	상대	후진	아래주먹	아래주먹	전진		아래주먹 유효거리 밖
4단계	거리	5	5	5			
	상대	위주먹	위주먹	위주먹			
실행 결과		총 9점 획득					

(a) 단순 지능 캐릭터의 행동

1단계	거리	5-6	6	6	6	6	
	상대	후진	아래발	아래발	아래발	아래발	
2단계	거리	6-5	5	5	5	5	
	상대	전진	아래발	아래발	아래발	아래발	
3단계	거리	5-6	6	6	6-5	5	5
	상대	후진	아래주먹	아래주먹	전진	위주먹	위주먹
4단계	거리	6-7	7	7	7-6	6	5
	상대	후진	옆차기	옆차기	옆차기	옆차기	옆차기
실행 결과		총 11점 획득					

(b) 행동 패턴을 학습하는 지능 캐릭터 행동

그림 1. 단순지능캐릭터와 행동패턴을 학습하는 지능캐릭터의 행동 비교

Fig. 1. Comparison of actions between a simple intelligent character and an intelligent character that learns action patterns.

공격이 먼저 종료하지만 유효거리 밖이므로 다시 지능 캐릭터의 아래발 공격으로 지능 캐릭터가 점수를 획득한다. 그리고 4단계에서는 두 캐릭터 모두 유효거리 밖이므로 점수를 획득하지 못한다. 반면에 상대 캐릭터의 과거 행동을 고려하는 행동 패턴을 학습하는 지능 캐릭터는 첫 번째 후진과 두 번째 후진에 대해 상대 캐릭터의 행동과 거리가 같아도 과거 행동이 다르기 때문에 구분하여 대응할 수 있다. 결국, 과거 행동을 고려하지 않는 단순 지능 캐릭터는 9점을 획득하고, 과거 행동까지 포함하여 행동하는 행동 패턴을 학습하는 지능 캐릭터는 11점을 획득할 수가 있다.

이와 같이 상대 캐릭터의 공격 패턴에 이동이 포함된 경우 행동패턴을 학습하는 지능캐릭터는 단순지능캐릭터에 비하여 효과적인 반응을 할 수 있다**. 그러나 상대 캐릭터의 공격 패턴이 오로지 공격만으로 이루어진 경우에는 이 절의 앞에서도 언급한 것처럼 상대 캐릭터의 현재 공격에 의해서만 점수가 결정되므로 과거행동을 고려하는 행동패턴을 학습하는 지능캐릭터가 더 좋은 대응을 할 수 없다. 하지만 게임의 규칙에서 점수를 부여할 때 공격행동별로 점수를 부여하는 것에 추가하여 일련의 공격행동에 따라 가산점을 부여한다면 공격 패턴이 공격만으로 이루어져도 과거행동을 고려하는 행동패턴을 학습하는 지능캐릭터가 단순 지능캐릭터보다 더욱 좋은 행동을 결정할 수 있다. 예를 들어, 옆차기와 돌려차기가 연속된 경우 두 번째 돌려차기의 점수를

** 본 논문에서는 이동을 중심으로 설명했으나 공격이 아닌 다른 행동 (예로 방어 혹은 점프등)에서도 똑같은 효과를 준다.

옆차기 없이 돌려차기만 했을 경우의 점수에 배를 준다고 했을 경우에 단순지능캐릭터는 현재의 상대 캐릭터의 행동 돌려차기만을 고려하므로 학습으로 선택된 어떤 행동 하나만을 반복할 것이다. 그러나 행동패턴을 학습하는 지능캐릭터는 옆차기-돌려차기의 연속공격과 어떤 행동-돌려차기를 구분할 수 있고 학습을 통하여 각 경우에 가장 효과적인 행동을 선택할 수 있다. 이와 같이 행동패턴을 학습하는 지능캐릭터는 기존의 단순지능캐릭터를 보다 일반화 지능화한 캐릭터로서 여러 경우에 더욱 효과적이다.

행동패턴을 학습하는 지능캐릭터가 행동패턴이 없는 상대와 대전하는 경우에도 단순 지능 캐릭터보다 우수한지 알기 위하여 두 캐릭터를 무작위적으로 행동하는 상대 캐릭터와 대결시켜 보았다. 실험결과 IV장에서 보여주듯이 무작위적으로 행동하는 상대 캐릭터와의 대결에서도 행동패턴을 학습하는 지능캐릭터가 단순지능캐릭터보다 더욱 효과적인 대응을 하여 보다 많은 점수를 획득함을 알 수 있다. 이를 통하여 본 논문에서 제안한 행동패턴을 학습하는 지능캐릭터가 기존의 단순지능캐릭터보다 우수함을 알 수 있다.

III. 게임 및 지능 캐릭터의 구현

본 장에서는 제안한 행동 패턴을 학습하는 지능 캐릭터를 학습시키기 위해 적용된 유전자 알고리즘의 여러 파라미터 및 염색체의 구조에 대하여 설명하고, 게임 규칙에 대하여 설명한다.

1. 대전 액션 게임 구현

본 논문에서 제안한 행동 패턴을 학습하는 지능캐릭터를 학습시키기 위하여 다음과 같은 게임을 구현하였다. 게임에서 캐릭터의 행동은 모두 14가지이며 이동이나 방어 행동은 6가지, 공격 행동은 8가지로 하였다. 캐릭터의 이동 범위는 일차원 상의 거리 0부터 거리 13까지 구간 14를 이동할 수 있도록 하였으며 한 번의 이동 명령으로 이동할 수 있는 범위는 1로 제한하였다. 본 연구에서 지능 캐릭터와 상대 캐릭터의 각 행동은 표1과 같이 각 행동별로 소요시간, 공격점수, 유효거리를 나타내고 있다. 표 1에서 정지 상태인 ID(정지)와 이동 행동으로는 GO(전진)와 BK(후진)이 있으며 막기 행동으로는 GD(막기)가 있고 공격을 피하기 위한 수단으로는 DN(앞기)와 JP(점프)가 있다. 이와 같이 하나의 클

* 단, 본 논문에서는 연속 공격에 대한 가산점을 이용한 실험은 제외하였다.

표 1. 행동의 종류, 단계 및 점수

Table 1. Types, steps and scores of actions.

	행동종류	유효거리	단계	점수
0	정지(ID)	-	1	0
1	전진(GO)	-	1	0
2	후진(BK)	-	1	0
3	점프(JP)	-	1	0
4	앞기(DN)	-	1	0
5	막기(GD)	-	1	0
6	아래 주먹(DP)	0 ~ 4	2	1
7	윗 주먹(UP)	0 ~ 4	3	2
8	아래발공격(DA)	3 ~ 7	4	3
9	위 발 공격(UA)	3 ~ 7	5	4
10	옆 차기(SK)	6 ~ 10	6	5
11	돌려 차기(RK)	6 ~ 10	7	6
12	들어 찌기(LC)	9 ~ 13	8	7
13	돌려 찌기(RC)	9 ~ 13	9	8

적으로 행동이 종료되는 행동은 획득하는 점수가 없으며 유효거리도 없다. 각 캐릭터는 모든 거리에서 어떠한 공격도 가능하지만 점수로 인정받기 위해서는 제한된 유효거리 내에서 공격이 이루어질 때에만 점수가 인정된다.

상대방의 공격에 대처하는 방법으로는 GO, BK 등으로 상대방과의 거리를 조절하거나 GD, JP, DN 등의 방어 자세를 취할 수도 있다. 거리 조절 방법은 상대방이 현재 취하고 있는 공격의 유효거리 밖으로 이동하는 것으로 상대방과의 거리가 4일 때 UP(위 주먹) 공격을 해올 경우 BK하면 거리가 5가 되어 유효거리 밖이므로 상대방의 UP 공격은 무효가 된다. 그러나 거리가 2일 때, 상대공격이 UP인 경우에는 GO나 BK를 하더라도 유효거리 이내이므로 타격을 받는다. 이런 경우에는 방어 자세를 취하는 방법이 있으며 유효한 방어 방법에 대한 규칙은 다음과 같다.

각각의 공격에 대해 취할 수 있는 방어 방법으로는 DP(아래 주먹공격), UP, DA(아래 발 공격), UA(위 발 공격) 공격에 대해서 GD하면 공격 점수가 인정되지 않으며 DP, DA 공격에 대해서 JP를 하면 공격 점수가 인정되지 않는다. 또, UP, UA 공격에 대해서 DN하면 공격 점수가 인정되지 않는다. 공격 시작 시간이 다르더라도 공격이 동시에 성공 할 경우 점수는 높은 공격 점수에서 낮은 공격 점수를 뺀 점수를 높은 공격을 한 캐릭터에게 부여하였다. 모든 공격은 반드시 유효 거리 이내에서 공격이 이루어질 경우만 점수로 인정한다. 그러나 캐릭터가 수행 중이던 행동은 임의로 취소하거나 공격 도중에 다른 행동으로 전환할 수는 없다.

2. 유전자 알고리즘의 파라미터

유전자 알고리즘의 중요 파라미터는 적합도, 선택, 교배, 돌연변이, 집단의 크기, 교배확률 및 돌연변이 확률이다[12]. 그림2는 본 논문에서 지능 캐릭터가 진화하는 과정이다. 먼저 지능 캐릭터의 집단에서 하나의 개체를 선택하여 상대 캐릭터와 게임을 한 후에 게임 결과의 적합도에 따라 부모 염색체를 선택하고, 교배, 돌연변이를 거쳐 종료 조건이 만족할 때까지 반복하게 된다. 적합도(fitness)는 지능 캐릭터와 상대 캐릭터가 얻은 점수의 차를 이용하였다. 선택 연산자의 종류는 적합도의 적용 방법에 따라 여러 가지가 있으나 본 연구에서는 널리 알려진 룰렛휠 (roulette wheel)선택^[13] 방법을 사용하였다. 룰렛휠 선택은 선택확률(적합도)에 따라 부모의 집단에서 개체를 선택하는 것으로서, 유전자를 변경하지 않으므로 집단 내에 있는 유전자의 형태 변화에는 영향을 미치지 않는다. 그러나 확률적 속성 때문에 선택과정에서 최적 염색체가 반드시 선택된다는 보장은 없다. 빠른 진화를 위하여 본 연구에서는 한 세대 내에서 가장 좋은 적합도를 가지는 염색체 다음 세대에 유전되도록 하였다^[14].

교배(crossover) 연산자는 선택된 염색체들이 교배를 통해 서로의 유전 정보를 교환함으로써 새로운 개체를 생성하는 과정이다. 본 연구에서는 유전자의 형태가 이차원이므로 다차원 교배인 블록 교배(block crossover)^[15]를 적용하였다

돌연변이(mutation) 연산자는 개체의 염색체 중에서 무작위로 선택된 문자열을 임의로 변경하는 것이다. 이는 부모해에 존재하지 않는 속성을 도입하여 탐색 영역을 확대해 준다. 본 연구에서는 염색체 표현이 0~10 사이의 정수이므로 난수를 발생시켜 돌연변이 확률에 해당되면 선택된 유전자를 염색체내의 정보 중에서 임의의 숫자로 대체하였다.

그 밖에 유전자 알고리즘 중에서 고려할 수 있는 구성요소로는 세대수, 집단의 크기, 교배확률, 돌연변이 확률, 종료조건 등이 있다. 본 연구에서는 집단의 크기(P.S.)는 30, 교배확률(Pc)는 1, 돌연변이 확률(Pm)은

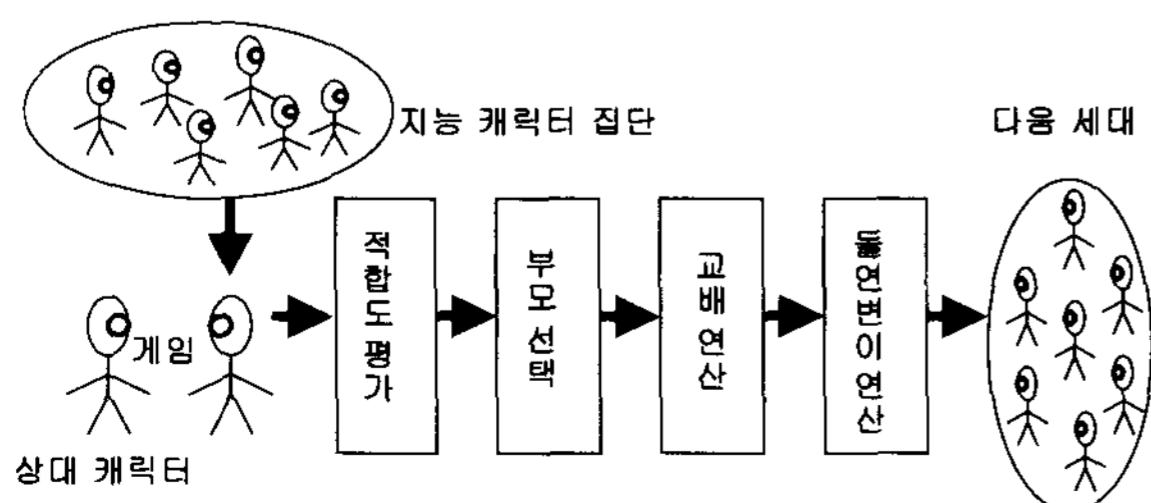


그림 2. 알고리즘의 동작 방식
Fig. 2. Process of algorithm.

0.01로 하였다.

3. 지능 캐릭터의 염색체 표현

유전자 알고리즘을 적용시키기 위해서는 먼저 게임 정보에 대한 염색체 표현이 선행되어야 한다. 이전 논문에서와는 다르게 과거 행동을 학습하기 위해서는 염색체의 표현에도 과거 행동을 구분할 수가 있어야 한다. 이를 위하여 염색체는 그림 3과 같이 캐릭터간의 거리와 현재 진행 중인 행동의 단계, 그리고 과거 행동까지를 포함하도록 하였다. 이에 따라 염색체는 그림 3과 같이 2차원 구조이며 각 셀의 숫자는 상대 캐릭터의 행동을 숫자로 표시한 것이다. 캐릭터의 행동 종류로는 이동이나 방어 행동 6가지와 공격행동 8가지 등 모두 14가지이며, 캐릭터가 이동할 수 있는 거리는 거리 0부터 13까지 모두 14단계이다.

그림 3(a)는 이전 논문에서 적용한 염색체 구조로서, 상대 캐릭터의 현재 공격과 단계, 거리에 따른 지능 캐릭터의 행동을 나타낸다. 예를 들어 상대 캐릭터가 거리 13에서 아래주먹 공격(6, DP) 공격의 시간 1t 일 때 지능 캐릭터는 9(위 발 공격)을 하고 있음을 나타낸다. 본 논문에서 제안하는 지능 캐릭터는 상대 캐릭터의 과거 행동을 구분해야 하기 때문에 (a)에서 지능 캐릭터의 행동이 (b)와 같이 과거 행동에 따른 지능 캐릭터의 행동을 저장하는 구조로 확장된다. 예를 들어 상대 캐릭터가 거리 13에서 아래주먹 공격(6, DP) 공격의 시간 1t 라고 가정하자. 만약 상대 캐릭터의 과거 행동이 ID 라면 지능 캐릭터는 아래주먹 공격(6, DP)을 하고, 과거 행동이 GO라면 위 발공격(9, UA)를 하게 된다. 단순 지능 캐릭터에서의 염색체 크기는 거리×모든 행동 단계의 합(14×50=700) 이지만 과거 행동을 포함하기 위해서 과거행동(0~13)을 더 기억하고 있어야 한다. 그러므로 과거 행동을 학습하는 경우 염색체의 크기는 거리

행동종류	ID	GO	BK	JP	DN	GD	주먹 공격					...			
							DP	UP	DA	UA	SK		RK	LC	RC
시간	1t	1t	1t	1t	1t	1t	1t	2t	1t	2t	3t	...			
거리	0	9	6	2	6	3	11	0	7	4	2	13	...		
(a) 현재 행동만 고려	1	7	3	5	9	10	2	4	9	2	1	13	...		
2	6	9	2	1	10	8	2	6	9	3	1	...			
...	:	:	:	:	:	:	:	:	:	:	:	:	:		
12	3	13	7	4	9	11	2	0	10	4	7	...			
13	7	4	9	3	2	4	9	12	0	4	9	...			
(b) 과거 행동 고려	과거행동종류	ID	GO	BK	JP	DN	GD	DP	UP	DA	UA	SK	RK	LC	RC
지능캐릭터의 행동		6	9	2	1	10	8	2	6	9	3	1	11	5	9

그림 3. 초기 염색체 구조의 일부
Fig. 3. The part of the initial chromosome.

×모든 행동의 단계 합×과거행동이므로 14×50×14=9800이 된다. 검색체의 크기가 크다는 것은 해를 찾는 시간이 더 길어진다는 단점이 있지만 검색체 크기만 으로 비교하기에는 무리가 있다. 왜냐하면 게임 규칙이 다르고 검색체의 크기가 다르더라도 조건에 따라 고려하지 않는 경우의 수도 많이 발생하기 때문이다.

IV. 실험 결과 및 분석

지능 캐릭터의 성능을 검증하기 위하여 표 2와 같이 5개의 행동패턴을 갖는 상대캐릭터를 사용하였다. 여기서 사용한 5개의 행동 패턴은 같은 조건일 때 지능 캐릭터의 행동을 알아보기 위한 것으로 전진과 후진을 반복하는 패턴을 사용하였다. 표 2에서 실험-1은 단순 지능 캐릭터와 게임을 한 후의 결과이며 실험-2는 행동패턴을 학습하는 지능 캐릭터와 게임을 한 후의 결과이다. 결과에서 보는 바와 같이 행동패턴을 학습하는 지능캐릭터는 최적의 대응방법을 찾음을 보여 준다. 표 2의 실험-1에서 굵은 숫자는 유효거리 범위를 벗어나 공격이 이루어진 경우로 점수로 인정 되지 않는 것을 표시한 것이다. 실험-2의 진한 숫자(1,2)는 지능 캐릭터가 더 높은 점수를 얻기 위하여 이동 행동을 한 후 공격하는 것을 보여 주고 있다. 표 2에서 단순지능 캐릭터는 최적값을 찾지 못하였으므로 최대값에 도달한 세대수를 표시하였고, 실험2에서는 최적값이 나온 세대수이며 그 이후로는 표시하지 않았다. 실험2에서 찾은 최적해는 유일한 최적해는 아니며 파라미터 값을 조정하여 실험을 한다면 다른 해가 존재할 가능성도 있다.

그림 4, 그림 5, 그림 6은 패턴별로 단순지능 캐릭터의 실험(실험-1)과 행동패턴을 학습하는 지능 캐릭터의 실험(실험-2) 결과로서 세대수가 증가하면서 두 캐릭터

표 2. 패턴별 최적 행동과 점수
Table 2. The optimal actions and their scores against each pattern.

패턴 종류	초기 거리	상대 캐릭터의 행동 패턴	최적 점수	실험1 단순지능 캐릭터	최대값의 세대수	실험2 행동 패턴을 학습하는 지능캐릭터	최적값의 세대수
패턴 1	7	2,8,1,8,2,13,0	12	8-8-8-5 (9점)	12	8-8-2-11 (13점)	49
패턴 2	5	9,1,11,2,10,1,6	14	8-3-11-1-3-9-12 (7점)	15	8-5-2-11-1-9-6 (14점)	97
패턴 3	4	2,8,1,8,6,2,6,13,1,7	18	8-8-2-8-11-1-0 (15점)	53	8-8-2-8-2-11-8 (18점)	143
패턴 4	9	11,2,9,1,10,2,11,1,9	23	10-3-2-13-13-2 (21점)	1689	10-10-10-2-13 (23점)	912
패턴 5	8	10,2,8,6,7,6,7,6	18	9-5-13-13 (15점)	110	1-8-13-2-12 (18점)	4195

간의 점수차가 증가하는 정도를 보여 주고 있다. 그림 4부터 그림 6에서 세대수의 마지막 좌표 값은 최적값(최대값)일 때의 세대수로서 실험-1에서는 최적값이 없으므로 최대값을 의미하며 실험2에서는 최적값이 나올 때의 세대수이다. 최대값이나 최적값이 나온 이후로는 세대수가 증가하더라도 더 이상의 값이 존재하지 않으므로 표시하지 않았다. 실험-1보다 실험-2에서 세대수가 증가한 후에 최적값을 찾는 것은 과거행동까지 포함하여 학습하는 캐릭터의 검색체 수가 단순지능 캐릭터 보다 커서 초기에 적응하는데 시간이 걸리기 때문이다.

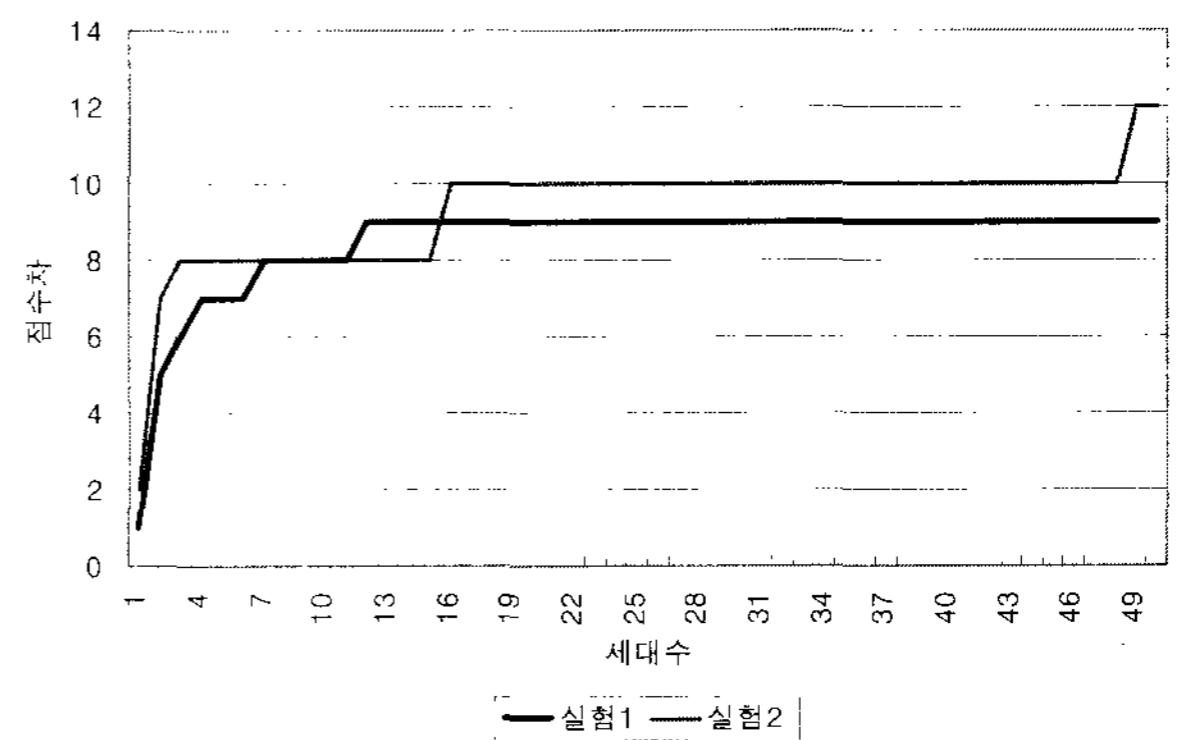


그림 4. 행동 패턴1의 결과 비교
Fig. 4. Experimental results for action pattern 1.

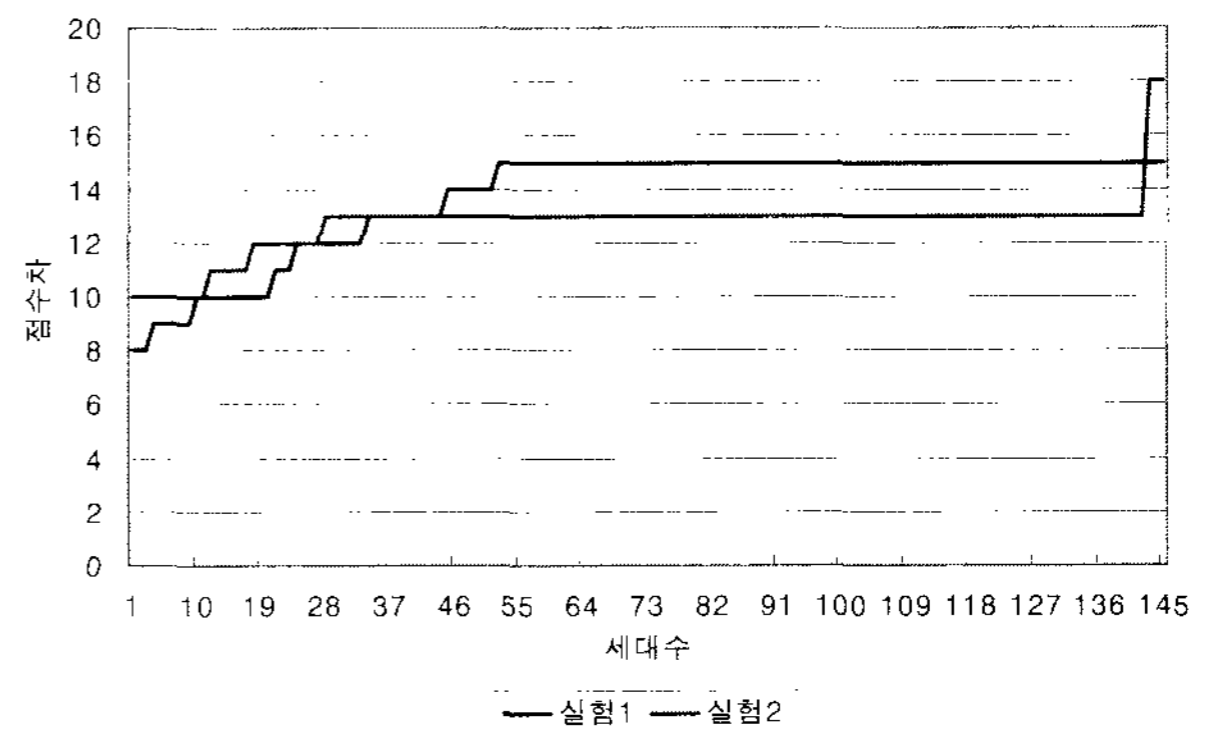


그림 5. 행동 패턴3의 결과 비교
Fig. 5. Experimental results for action pattern 3.

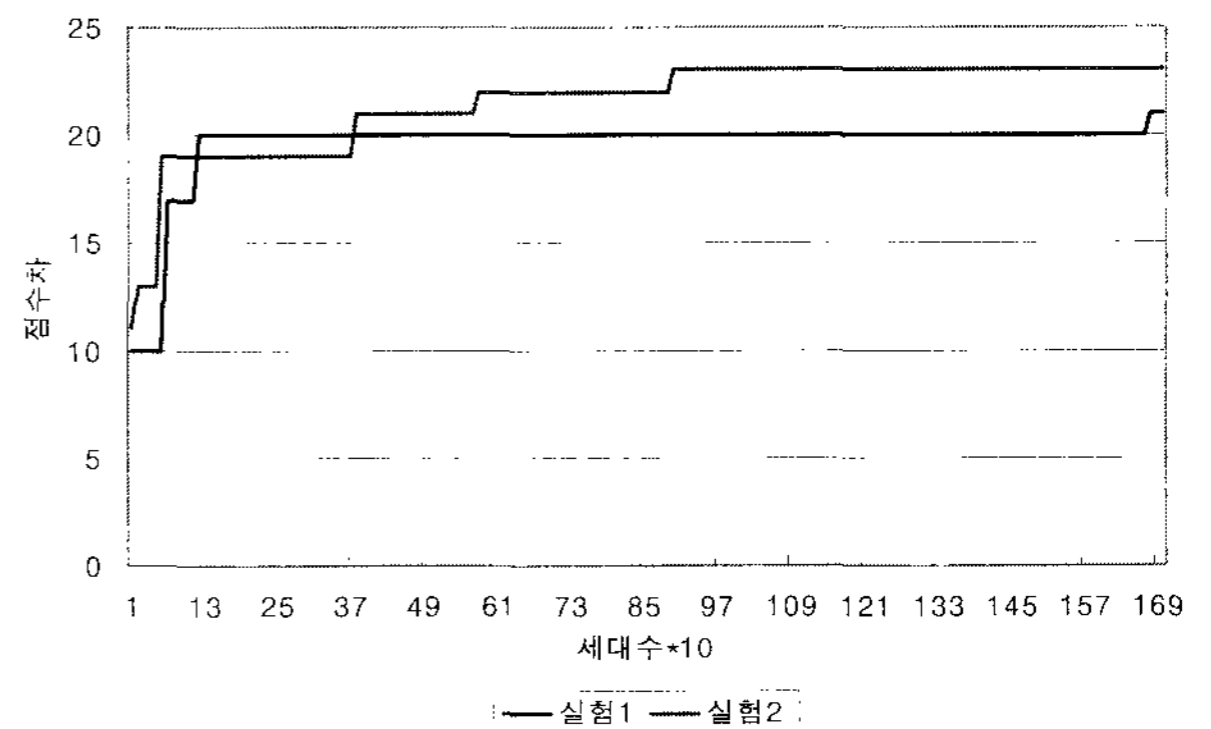


그림 6. 행동 패턴4의 결과 비교
Fig. 6. Experimental results for action pattern 4.

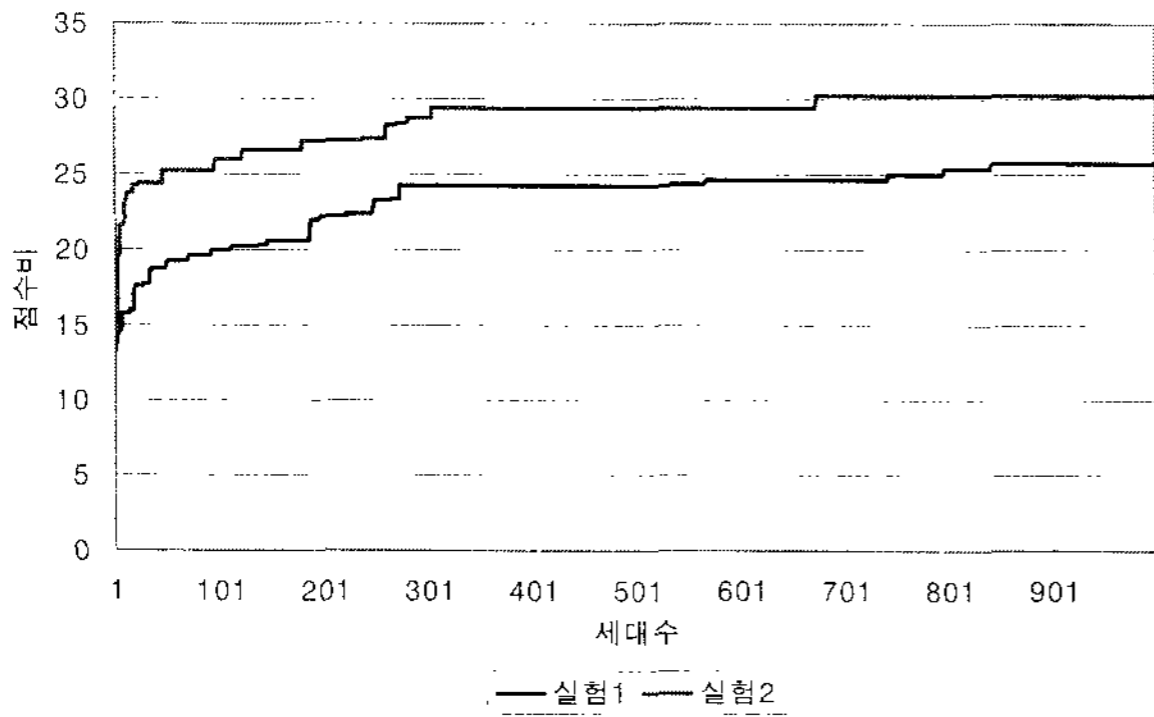


그림 7. 무작위 행동일 때의 결과 비교
 Fig. 7. Comparison of experimental results against randomly acting opponents.

그림 7은 행동패턴으로 행동하지 않고 무작위로 행동하는 상대 캐릭터에 대해서 지능 캐릭터의 성능을 평가하기 위하여 다양한 SEED값으로 5번의 실험결과에 대한 점수비를 평균한 값이다. 실험-1은 현재행동만을 학습하는 지능 캐릭터의 점수비이고 실험-2는 과거 행동까지 학습하는 지능 캐릭터와 게임을 한 결과를 점수비로 나타내었다. 세대수를 1000세대까지 실험한 결과 실험-2가 더 높은 점수비를 보였으며, 본 논문에서 제안한 지능 캐릭터는 상대방 캐릭터가 행동 패턴을 가지는 경우는 물론 무작위로 행동할 때에도 보다 나은 행동을 학습하는 것을 알 수 있다.

이와 같이 행동 패턴을 학습하는 지능 캐릭터는 주어진 조건에서 더 높은 점수를 획득하기 위해서 이동(전진과 후진) 행동은 물론이고 지능 캐릭터의 행동이 끝나는 단계를 조정하여(패턴2) 최적을 찾을 수가 있었다. 또한 실험-1에서는 과거 행동의 결과에 영향을 받지 않아서 거리와 행동 단계가 같으면 늘 같은 행동을 하게 되지만 실험-2에서는 과거의 행동이 현재의 행동에 영향을 미치므로 과거 행동에 따라 효과적인 현재행동을 학습할 수 있다. 이러한 결과는 행동패턴을 학습하는 캐릭터가 상대 캐릭터의 행동 패턴을 학습함을 입증하며 무작위로 행동하는 상대 캐릭터에 대해서도 좋은 결과를 나타내는 것으로 보아 일반적인 경우에도 단순 지능캐릭터보다 우수함을 보여준다. 이는 본 논문에서 제안한 지능캐릭터가 상대 캐릭터가 행동패턴을 보이는 경우나 그렇지 않은 경우 두 경우 모두에서 기존의 방법보다 효과적임을 보여주는 결과이다.

V. 결 론

본 논문에서는 유전자 알고리즘을 이용하여 대전 액션 게임용 지능 캐릭터를 구현할 때 상대 캐릭터의 행

동 패턴에 대해서도 적절히 대응할 수 있는 방법을 제안하였다. 이를 위하여 상대 캐릭터의 현재 행동뿐만 아니라 과거의 행동도 학습에 포함 시켰다. 지능 캐릭터의 성능을 평가하기 위하여 5개의 행동패턴을 보이는 상대 캐릭터를 이용하여 현재의 행동과 단계만을 고려하여 행동하는 단순지능 캐릭터(실험-1)와 현재의 행동, 단계 그리고 과거 행동까지 포함하여 패턴행동을 학습하는 지능 캐릭터(실험-2)를 실험하였다.

본 연구에서 제안한 지능 캐릭터의 실험2의 결과에서는 모든 패턴에 대해 최적을 찾았으며 주어진 조건에서 최적을 찾기 위해서 이동(전진과 후진) 행동은 물론, 지능 캐릭터의 행동이 끝나는 단계를 조정하여(패턴2) 최적을 찾을 수가 있었다. 또한 실험-1에서는 과거 행동의 결과에 영향을 받지 않으므로 거리와 행동 단계가 같으면 늘 같은 행동을 하게 되지만, 실험-2에서는 과거의 행동을 고려하여 과거 행동에 따라 보다 효율적인 현재의 행동을 선택할 수 있다. 그러므로 실험-2에서의 지능 캐릭터는 같은 조건이라도 더 높은 공격 점수를 얻기 위해 이동행동을 하는 것을 확인하였으며 행동 패턴을 보이는 상대 캐릭터에 대해서 최적을 찾는데 실험-1보다 좋은 결과를 보여주었다. 그리고 특정 패턴이 아닌 무작위로 동작하는 상대 캐릭터에 대해서도 더 좋은 결과를 보여 주고 있다. 따라서 본 논문에서 제안한 방법이 상대캐릭터가 행동패턴을 갖는 경우는 물론 그렇지 않은 경우에도 우수함을 알 수 있다. 그러나 행동 패턴일 경우에는 점수 차로 비교하였고 무작위 경우에는 점수비로 결과를 비교하였으나 추후 연구에서는 최적해에 수렴하는 속도등을 비교해 볼 필요도 있다고 생각된다. 추후 연구로서 에너지 개념을 추가하고 2차원 게임으로 확장하는 등 현재 유행하는 실제 게임과 유사한 게임에 적용하기 위한 노력도 필요할 것이다.

참 고 문 헌

- [1] D.C Pottinger and J.E.Laird, "Game AI : The State of the Industry Part 2," Gameasutra Magazin Nov 2000, Vol. 08
- [2] J.E.Laird, "Using a computer game to develop advanced AI," 2001 IEEE Computer, July 2001.
- [3] D.Carmel and S.Markovitch, "Learning Models of Opponent's Strategy in Game Playing,"
- [4] Mark DeLoura, Game Programming Gems 2, Charles River Media, 2001.
- [5] Micheal van Lent and john Laird, "Developing an artificial Intelligence Engine," Proc. of the Game Developers Conference, 1999, pp. 577~588

[6] John E. Laird and John C. Duchi, "Creating Human-like Synthetic Characters with Multiple Skill Levels" Symposium AAAI 2000.

[7] John E. Laird and M.van Lent, "Human-level AI's Killer Application : Interactive Computer Games," Proc. AAAI 2000, pp. 1171~1178

[8] 이면섭, 조병헌, 정성훈, 성영락, 오하령, "유전자 알고리즘을 이용한 대전형 액션게임의 지능 캐릭터", 정보과학회, 2005 submitted.

[9] 조병헌, 정성훈, 성영락, 오하령, "신경망을 이용한 지능형 게임 캐릭터의 구현", 정보처리학회, 제 11-B권 제7호, 2004.

[10] 이면섭, 조병헌, 정성훈, 성영락, 오하령, "대전 액션게임에서의 유전자 알고리즘을 이용한 지능 캐릭터", 전자공학회 논문지 제 41권 제4호

[11] 조병헌, 정성훈, 성영락, 오하령, "대전 액션 게임을 위한 신경망 지능 캐릭터의 구현", 한국 퍼지 및 지능시스템학회 논문지 Vol. 14, No. 4, pp.383~389, 2004.

[12] D. Whitley, R. Beveridge, C. Guerra and C. Graves. "Messy Genetic Algorithms for Subset Feature Selection", International Conference on Genetic Algorithms 1997.

[13] Darrell Whitley, "An Overview of Evolutionary Algorithms". Journal of Information and Software Technology 43:817~831, 200.

[14] Darrell Whitley, "A Genetic Algorithm Tutorial", Statistics and Computing (4):65~85, 1994.

저 자 소 개



이 면 섭(정회원)
 1985년 국민대학교 전자공학과 (공학사)
 1987년 인하대학교 전자공학과 (공학석사)
 2005년 국민대학교 전자공학과 (공학박사)

1990년~현재 인천전문대학 컴퓨터정보과
 전임강사, 조교수, 부교수
 <주관심분야 : 유전자 알고리즘, 게임, 컴퓨터 그래픽>



조 병 헌(학생회원)
 1997년 국민대학교 전자공학과(공학사)
 1999년 국민대학교 전자공학과(공학석사)
 2005년 국민대학교 전자공학과(공학박사)

2005년~현재 전자통신연구원 디지털콘텐츠 연구단
 게임기술개발센터 네트워크 가상환경
 연구팀 연구원
 <주관심분야 : 게임인공지능, 유전자알고리즘, 신경망>



정 성 훈(정회원)
 1988년 한양대학교 전자공학과 (공학사)
 1991년 한국과학기술원 전기및 전자공학과(공학석사)
 1995년 한국과학기술원 전기및 전자공학과(공학박사)

1995년~1996년 한국과학기술원 전기및전자공학과(위촉연구원)
 1996년~1998년 한성대학교 정보전산학부 정보통신공학전공 전임강사
 1998년~2002년 한성대학교 정보전산학부 정보통신공학전공 조교수
 2002년~현재 한성대학교 정보통신공학과 부교수



성 영 락(정회원)
 1989년 한양대학교 전자공학과 (공학사)
 1991년 한국과학기술원 전기 및 전자공학과(공학석사)
 1995년 한국과학기술원 전기 및 전자공학과(공학박사)

1995년~1996년 한국과학기술원 위촉연구원
 1996년~1998년 국민대학교 전자공학부 전임강사
 1998년~2002 국민대학교 전자공학부 조교수
 2002년~현재 국민대학교 전자정보통신공학부 부교수
 <주관심분야 : 시뮬레이션, 고장감내, 내장형 시스템>



오 하 령(정회원)
 1983년 서울대학교 전기공학과 (공학사)
 1983년~1986년 삼성전자 종합연구소
 1988년 한국과학기술원 전기전자과 컴퓨터공학전공(공학석사)
 1992년 한국과학기술원 전기전자과 컴퓨터공학전공(공학박사)
 1992년~1996년 국민대학교 전자공학부 조교수
 1996년~2001 국민대학교 전자공학부 부교수

2001년~현재 국민대학교 전자정보통신공학부 교수
 <주관심분야 : 병렬처리, 내장형 시스템, 고장감내>