

다사용자 온라인 게임을 위한 지능형 MOB 에이전트 설계

김진수, 방용찬
 건양대학교 전산게임학과
 e-mail : jinskim@konyang.ac.kr

Design of Intellectual MOB Agent for Multi-player Online Game

Jin-Soo Kim, Yong-Chan Bang
 Dept. of Computer Science & Game, Konyang University

요 약

기존의 다사용자 온라인 게임에서 구현되어 있는 MOB(Mobile Character)들은 ‘대기’와 ‘공격’의 2 가지 상태를 가지며 사용자의 ‘공격’이라는 이벤트에만 반응하도록 설계되어 있는 수동적인 에이전트들이다. 본 논문에서는 기존의 ‘대기’와 ‘공격’ 상태에 ‘회피’ 상태를 추가하고 3 가지 각각의 행동 전이에 따른 행동 패턴을 행동 특성 곡선으로 표현하며 ‘공격’과 ‘접근’ 자극을 스트레스 모형에 적용하여 스트레스에 따른 MOB 에이전트의 행동 패턴 변화를 설명하고 주변의 다른 에이전트들과의 협동을 도모할 수 있는 지능적인 NPC 에이전트를 설계한다.

1. 서 론

최근 몇 년간 일반 가정에도 고속 인터넷의 빠른 보급과 디지털 콘텐츠의 불법 복제가 늘어남에 따라 많은 게임들이 패키지게임보다 온라인게임으로 개발되고 있는 추세이다. 또한 많은 온라인 게임들이 여러 사용자들을 지원하는 다사용자 온라인 게임으로 개발됨에 따라 동일한 게임 내에서 많은 사용자들이 동시에 게임을 진행하게 된다. 이러한 게임들은 실사용자들 간에 행동으로만 진행되기도 하지만 게임의 진행을 위하여 실사용자가 조종하지 않는 NPC (None Play Character) 에이전트들도 존재하고 있다. 이러한 NPC 에이전트들은 게임의 진행만을 위하여 존재하기도 하지만 실사용자가 사냥하는 NPC 도 존재하며 이런 NPC 들을 MOB 라고 한다. 기존의 다사용자 온라인 게임에서는 이러한 MOB 의 행동패턴은 ‘공격’과 ‘대기’의 수동적인 상태만을 가지며 단순하고 반복적인 경향이 있다[1].

다음 <표 1>은 기존의 게임에서 공격과 대기 이외의 MOB 행동 패턴을 정리한 것이다. <표 1>에서와 같이 공격과 대기 이외에 각 MOB 에 부여된 특정한 행동패턴이라는 것이 하나가 추가 된 것 일뿐 지능적인 MOB 에이전트는 없다고 볼 수 있다.

따라서 본 논문에서는 이러한 단순한 행동 패턴을 가진 MOB 들을 스트레스 모형을 이용하여 능동적이고 지능적인 MOB 들로 설계한다.

<표 1> 기존 게임에서의 MOB 행동 비교

Game	MOB	Pattern
B 사에 W 게임	랩터	랩터가 공격 당하면 주위에 있는 다른 랩터가 와서 공격을 한다.
	인간족	일정한 데미지를 입으면 도망간다.
G 사에 R 게임	지오그래퍼	주위에 MOB 가 데미지를 입으면 그 MOB 를 치료 해준다.
	페코페코	페코페코가 공격 당하면 주위에 있는 다른 페코페코가 와서 공격을 한다.
N 사에 R 게임	골렘	처음에는 돌처럼 있지만 일정한 시야범위에 들어오면 공격한다. 단, 깨어 있는 상태에서는 비 공격성을 띈다.

2. 지능형 MOB 설계

본 논문에서 설계하는 지능형 MOB 에이전트는 단순한 행동 패턴을 가지고 있던 기존의 MOB 에이전트들에게 패턴의 다양화와 지능화를 부여하기 위하여 스트레스 모형을 사용하고 있다. 스트레스 모형은 건축학, 의학 등에서 널리 사용되고 있으며 측정 값이 특정 임계 값을 넘을 때 물체의 파괴 또는 병의 발병과 같은 특정 이벤트가 발생하는 것을 모형화 하며 하나의 측정값에 대한 임계 값이 하나의 곡선으로 표

현된다. 따라서 측정값이 임계 값을 넘어설 때 이벤트가 발생한다[2].

이러한 스트레스 모형은 단순한 모형이기 때문에 다양한 행동 패턴을 나타내기 위하여 6 개의 컨트롤 점을 갖는 2 차 베지어 곡선(Bézier curves)으로 표현된 행동특성 곡선을 사용한다[3][4][5].

2.1 스트레스 모형

본 논문에서는 MOB 가 받는 자극을 스트레스 S 라고 정의하고 스트레스는 공격 자극에 의한 스트레스 A_s 와 공간 자극에 의한 스트레스 D_s 로 정의한다. 공격 자극의 스트레스 A_s 는 다음의 (식 3)에서와 같이 공격을 받는 시점에서 큰 폭으로 상승하다가 최대값이 되면 더 이상 증가하지 않는 특징이 있다. 공간 자극의 스트레스 D_s 는 다음의 (식 4)에서와 같이 MOB 와 사용자 캐릭터와의 거리 D 에 비례하여 시간에 관계없이 지속되는 모습으로 정의한다. D_s 에서의 거리 D 는 다음 (식 1)과 같다.

$$D = Ms - U \quad \text{---- (식 1)}$$

(식 1)에서 M_s 는 MOB 의 시야범위를 나타내고 U 는 사용자 캐릭터와의 유클리드 거리를 나타내며 시야범위 내에 여러 사용자 캐릭터가 있을 경우 각 사용자 캐릭터로부터 받는 공간 자극을 계산하여 총합을 D_s 로 계산한다. 여기에서 U 는 다음의 (식 2)와 같다

$$U = \sqrt{(Mobx - Playx)^2 + (Moby - Playy)^2} \quad \text{--- (식 2)}$$

(식 2)에서 $Mobx$ 와 $Moby$ 는 MOB 의 좌표 값이고 $Playx$ 와 $Playy$ 는 사용자의 좌표 값이 된다.

$$A_s = \alpha t \quad \text{---- (식 3)}$$

(단, $A_s \geq Max$ 이면 $A_s = Max$ 이다)

(식 3)에서 α 와 Max 는 상수로 각 MOB 의 수치를 나타낸다. 공격 자극은 일정치 이상 받더라도 자극의 크기는 같아진다. 즉, Max 값을 넘어서도 그 값은 Max 값 만큼만 받게 된다.

$$D_s = \frac{2^D}{D} \quad \text{---- (식 4)}$$

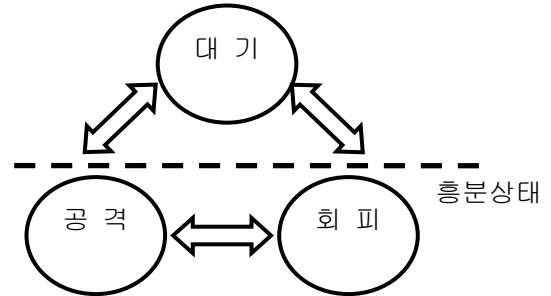
MOB 가 받는 스트레스 S 는 다음 (식 5)와 같이 공격 자극에 의한 스트레스 A_s 와 공간 자극에 의한 스트레스 D_s 의 합으로 나타내어지며 시간의 흐름에 따라 계속 누적된다.

$$S = A_s + D_s \quad \text{---- (식 5)}$$

MOB 의 스트레스 S 는 초기상태의 결정과 행동 특성 곡선에서 사용된다.

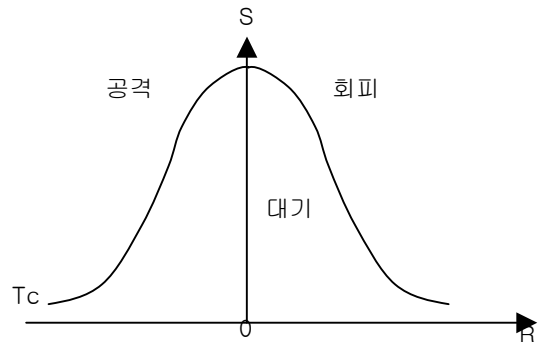
2.2 초기 상태의 결정

본 논문에서 제안하는 MOB 의 상태는 ‘대기’, ‘공격’, ‘회피’ 의 3 가지의 상태이다. 이 상태들은 다음 (그림 1)과 같이 한 상태에서 다른 상태로 전이 될 수 있다. MOB 의 3 가지의 상태 중 ‘공격’ 과 ‘회피’ 상태를 흥분상태로 정의하며 흥분 상태일 경우 MOB 가 받는 스트레스 S 를 이용하여 행동 특성 곡선에 의해 상태 전이를 결정하고 ‘대기’ 상태에서 흥분 상태로의 정의는 초기 상태 결정에서 이루어진다.



(그림 1) MOB 의 상태 전이도

MOB 는 평상시에는 ‘대기’ 상태로 외부의 자극을 기다린다. 외부의 자극은 사용자 캐릭터의 행동에 의해 주어지며 이 자극이 임계 값을 넘을 때 흥분 상태로 전이되며 ‘공격’ 또는 ‘회피’ 상태를 결정할 때 다음 (그림 2)의 모델을 이용하게 된다. 이러한 상태는 기존의 MOB 들이 사용자 캐릭터가 다가 공격할 경우 반격을 한다던가 또는 가장 가까운 사용자 캐릭터를 공격한다던가 하는 단순한 패턴의 행동이 아닌 사용자 캐릭터가 다가가서 서있는 다거나 공격할 경우에 사용자 캐릭터에 따라 공격이나 회피를 선택하여 좀더 다양한 행동 패턴을 만들기 위해 고안되었다.



(그림 2) 초기상태의 결정

(그림 2)에서 S 는 MOB 가 받는 스트레스이고 수치가 높을수록 공격적인 MOB 가 되는 A_c 는 MOB 마다 부여되는 공격성향 상수이고 T_c 는 MOB 의 행동 활성화 임계값이다. T_c 는 항상 A_c 보다 커야 한다. T_c 는 식. 6 과 같다.

$$T_c = \alpha \cos x + 2\alpha \quad \text{(단, } -\alpha \leq x \leq \alpha \text{)} \quad \text{--- (식 6)}$$

여기에서 α 는 상수로 각 MOB 의 수치이다.

각각의 상태는 다음 <표 2>와 같다.

<표 2> 각각의 상태

	R > 0	R < 0
S+Ac < Tc	대기	대기
S+Ac > Tc	회피	공격

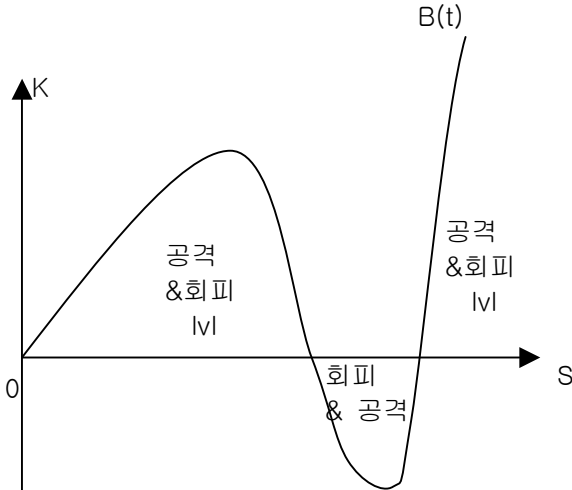
$$R = (Play.att - Mob.att) - Mob.att \quad \text{--- (식 7)}$$

(식 7)에서 Play.att 는 사용자 캐릭터의 공격력으로 계산되고 Mob.att 는 MOB 의 공격력으로 계산된다. MOB 의 상태는 스트레스 S 의 누적과 공격 성향 상수 Ac 를 더한 값이 Tc 를 넘어설 때 흥분상태로 전이되며 상태의 결정은 상대적인 공격력 R 에 따라 달라진다. R 이 음수일 경우는 공격상태를 R 이 양수일 경우에는 회피 상태로 전이되며 R 의 크기가 클수록 Tc 에 미치지 못한 경우에도 상태의 전이가 일어난다. 이는 MOB 와 사용자 캐릭터간의 공격력의 차이가 클 때 좀더 빠른 상태전이가 일어나도록 하기 위함이다.

초기 상태의 결정이 이루어지면 MOB 는 ‘ 공격 ’ 상태가 되어 사용자 캐릭터에게 다가가 공격 행동을 하거나 ‘ 회피 ’ 상태가 되어 사용자 캐릭터와 가장 먼 방향으로 도망치는 행동을 하게 된다. 이러한 행동을 하는 중에서도 MOB 는 사용자 캐릭터로부터 계속적인 스트레스를 받게 되고 그 스트레스에 따라 상태의 유지 또는 전이를 MOB 마다 부여된 행동 특성 곡선에 의해 결정하게 된다.

2.3 행동 특성 곡선

행동 특성 곡선은 MOB 마다 다른 행동 패턴을 부여하기 위한 방법으로 (그림 3)과 같은 모습이다.



(그림 3) MOB 의 행동 특성 곡선

$$B(t) = \sum_{c=0}^N P_c \frac{M!}{d(N-c)!} t^c (1-t)^{N-c} \quad (0 \leq t \leq 1) \quad \text{--- (식 8)}$$

(식 8)에서 N 은 조절 점의 개수이고 t = 0 이면 시작점을 의미 하고 t = 1 이면 마지막 점을 의미 한다. P_c 는 x, y 좌표 값을 갖는 하나의 구조체를 의미한다.

즉, $P_c = \begin{pmatrix} x \\ y \end{pmatrix}$ 를 의미 한다.

행동특성 곡선에 적용하면 (식 9)가 된다.

$$B(t) = (1-t)^5 P_0 + 5t(1-t)^4 P_1 + 10t^2(1-t)^3 P_2 + 10t^3(1-t)^2 P_3 + 5t^4(1-t) P_4 + t^5 P_5 \quad \text{--- (식 9)}$$

행동 특성 곡선은 (그림 3)과 같이 6 개의 컨트롤 점을 갖는 2 차 베지어 곡선으로 표현한다. S 의 값이 증가 함에 따라 K 가 음과 양으로 변화한다. 초기 상태의 결정에서 ‘ 공격 ’ 이 나올 경우 행동 특성 곡선에서 K 값이 0 보다 클 때는 ‘ 공격 ’, 0 보다 작을 때는 ‘ 회피 ’ 가 된다. 초기상태에서 ‘ 회피 ’ 가 나올 경우 행동 특성 곡선에서 K 값이 0 보다 클 때는 ‘ 회피 ’, 0 보다 작을 때는 ‘ 공격 ’ 이 부여되어 각 MOB 마다 개성 있는 행동을 나타내게 할 수 있게 된다.

(그림 3)의 공격적인 행동을 나타내는 곡선은 스트레스의 증가에 따라 처음에는 ‘ 공격 ’, 이후 ‘ 회피 ’ 그리고 다시 ‘ 공격 ’ 순으로 변화 하고 회피적인 행동을 나타내는 곡선은 스트레스의 증가에 따라 처음에는 ‘ 회피 ’, 이후 ‘ 공격 ’ 그리고 다시 ‘ 회피 ’ 순으로 변화한다.

2.4 다른 에이전트와의 협동

기존의 게임에서 나타난 MOB 에이전트들의 협동은 시야 범위 내에 아군이 공격 당할 경우 무조건 사용자 캐릭터를 공격하는 방식이었다. 본 논문에서는 다른 에이전트와 좀더 역동적인 협동을 위하여 스트레스의 전파를 이용한다. 스트레스 전파 메시지는 스트레스를 받는 MOB 가 발신하고 발신 MOB 의 시야 범위 내에서 전파된다. 스트레스 전파 메시지는 다음 <표 3>과 같다.

<표 3> 스트레스 전파 메시지

ID	발신 MOB 의 고유 ID
S	발신자의 스트레스
Loc{x, y}	발신자의 좌표
Hob	흡 제한을 위한 카운트

발신자 ID 는 중복된 전파를 피하기 위한 것이고 스트레스 S 는 전파될 스트레스를 나타낸다. 그리고 발신자의 위치는 거리에 따라 스트레스 강도를 계산할 때 필요하며 흡 카운트는 메시지의 전파 범위를 제한하기 위한 것으로 메시지가 한번 전파될 때 마다 1 씩 감소한다. 메시지를 받은 MOB 의 스트레스 S'는 다음 (식 10)과 같이 계산한다.

$$S' = \left(1 - \frac{\sqrt{(Locx-x)^2 + (Locy-y)^2}}{Sr}\right) \times S \quad \text{--- (식 10)}$$

(식 10)의 Loc.x 와 Loc.y 는 발신자의 위치이고 x 와 y 는 메시지를 전파 받은 MOB 의 위치가 된다. 그리고 Sr 은 발신자의 시야거리이고 S 는 발신자의 스트레스이다.

스트레스를 전파 받은 MOB 는 전파 받은 스트레스 S' 를 이용하여 자신의 스트레스를 계산하고 초기 상태 결정과 행동 특성 곡선을 이용하여 자신의 상태를 결정하고 행동하게 된다. 따라서 MOB 들은 집단 공격 또는 집단 회피 등의 행동을 할 수 있다. 사용자 캐릭터가 하나의 MOB 를 죽이고 주변의 다른 MOB 를 사냥 할 경우 해당 MOB 는 이미 스트레스를 받고 있었기 때문에 좀 더 빠른 상태 변화를 하게 될 것이며 주변의 다른 MOB 들 역시 누적된 스트레스로 집단 행동을 하게 된다.

2.5 스트레스의 감소

어떠한 자극도 없을 경우 스트레스의 감소가 발생한다. 감소하는 값은 (식 11)과 같다.

$$S' = \frac{S}{t} \quad \text{---- (식 11)}$$

(식 11)의 S' 는 시간과 반비례하며 스트레스를 받지 않는 시간 동안 줄어 들게 된다. 스트레스가 0 이 되면 As 와 Ds 도 0 이 되고 초기 상태 결정도 대기로 바뀌며 행동 특성 곡선도 초기화가 된다. 따라서 MOB 는 다양한 행동들을 할 수 있게 된다.

3. 적용 결과

제안된 방법들을 이용할 경우 기존의 방식에 비하여 MOB 의 생존시간이 늘어나고 사용자 캐릭터가 받는 공격의 합이 늘어남에 따라 좀 더 역동적인 게임의 진행이 가능할 것이다. 앞에서 제시한 <표 1>의 MOB 들에게 본 논문의 방법을 적용한 결과는 다음 <표 4>와 같다.

<표 4> 기존 게임에 적용결과

Game	MOB	Pattern
B 사에 W 게임	랩터	스트레스에 따라 사용자 캐릭터에게 주위에 있는 MOB 들과 함께 집단 공격을 하거나 단체로 도망을 간다.
	인간족	스트레스에 따라 사용자 캐릭터에게 주위에 있는 MOB 들과 함께 집단 공격을 하고 일정 데미지를 입은 MOB 는 도망을 간다.
G 사에 R 게임	지오그래퍼	스트레스에 따라 사용자 캐릭터에게 주위에 있는 MOB 들과 함께 집단 공격을 하거나 단체로 도망을 간다. 그리고 주위에 MOB 가 데미지를 입으면 그 MOB 를 치료 해준다.
	페코페코	스트레스에 따라 사용자 캐릭터에게 주위에 있는 MOB 들과 함께 집단 공격을 하거나 단체로 도망을 간다.
N 사에 R 게임	골렘	스트레스에 따라 사용자 캐릭터에게 주위에 있는 MOB 들과 함께 집단 공격을 하고 스트레스가 없는 경우 돌처럼 움직이지 않는다.

<표 4>에서처럼 MOB 가 스트레스를 받는 양에 따라 MOB 의 움직임이 달라지고 스트레스의 감소로 행동특성 곡선의 초기화로 인하여 같은 MOB 라도 각기 다른 사용자 캐릭터에 서로 다르게 반응이 나타날 수 있다. <표 5>는 사용자 캐릭터에 따른 MOB 의 패턴 변화를 정리해 본 것이다.

<표 5> 사용자 캐릭터에 따른 MOB 의 반응

MOB	사용자캐릭터	스트레스의 양	Pattern
MOB[Lv20, Att 40, 공격적]	[Lv 10, Att 20]	30%	사용자 캐릭터를 공격함
	[Lv 30, Att 60]	30%	사용자 캐릭터 반대 방향으로 도망을 감
MOB[Lv20, Att 40, 공격적]	[Lv 10, Att 20]	70%	사용자 캐릭터 반대 방향으로 도망을 감
	[Lv 30, Att 60]	70%	사용자 캐릭터를 공격함

<표 5>에서 MOB 의 Lv 는 20 이고 공격력은 40 이고 공격적인 성향을 가지고 있는 MOB 가 Lv 가 10 이고 공격력이 20 인 사용자 캐릭터에게는 공격하였는데, Lv 가 30 이고 공격력이 60 인 사용자 캐릭터에게는 공격하지 않고 회피하였다. 이와 같이 동일한 MOB 라도 사용자 캐릭터와 스트레스의 양에 따라 움직임이 달라짐을 알 수 있다.

4. 결론

본 논문에서는 기존의 다사용자 온라인 게임의 단순하고 수동적인 MOB 에이전트에 스트레스 모형과 행동 특성 곡선 그리고 스트레스의 전파 방법을 적용하여 지능적이고 다양한 패턴을 갖는 능동적인 MOB 에이전트를 설계하였다. 적용 결과에서 나온 자료에서와 같이 현재 출시되고 있는 다사용자 온라인 게임에도 쉽게 적용할 수 있다는 장점이 있다. 또한 실사용자는 MOB 의 움직임을 쉽게 예측할 수 없게 되어 게임에서 더욱 박진감을 느낄 것이다.

향후 연구 과제는 초기 상태의 결정에 퍼지와 같은 유연한 방법의 도입이 필요하다.

참고문헌

- [1] 이만제 “ 게임에서의 인공지능 기술”, 정보처리, 제 9 권, 제 4 호, 2002.
- [2] Khamis I.H. "An alternative to the Weibull step-stress model" International Journal of Quality & Reliability Management, Vol.16, No.2, pp.158-165, Feb. 1999.
- [3] Paul Bourke “ <http://astronomy.swin.edu.au/~pbourke/curves/bezier/>” 1996.
- [4] 류광 역 『 3D 게임 프로그래밍 & 컴퓨터 그래픽을 위한 수학 제 2 판』 pp.530-536, 2004.
- [5] Chen, Q. "A class of Bezier-like curves", Computer aided geometric design, Vol.20, No.1, 2003.