온라인 게임을 위한 향상된 지능형 MOB 에이전트 설계

김진수, 방용찬
건양대학교 전산게임학과
e-mail : jinskim@konyang.ac.kr, light9898@konyang.ac.kr

Design of Improved Intellectual MOB Agent for Online Game

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

요 약

기존의 온라인 게임에서 구현되어 있는 수동적인 MOB(Mobile Character)에 ‘ 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 MOB 에이전트를 [1]논문에서 설계하였다. 본 논문에서는 [1]논문에서의 모형을 향상시키기 위하여 행동패턴을 구체화하고 수식을 추가하였으며, 또한 스트레스 카운터를 추가하여 보다 현실적인 모형을 설계하였다.

1. 서 론

최근 몇 년간 일반 가정에 고속 인터넷의 빠른 보급과 디지털 컨텐츠의 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 에이전트 라고 한다. 기존의 다사용자 온라인게임에서는 이러한 MOB 에이전트의 행동패턴은 ‘ 공격’ 과 ‘ 대기’ 의 수동적인 상태만을 가지며 단순하고 반복적인 경향이 있다[2]. [1]논문에서 사용된 수식을 변형하고 새로운 알고리즘을 도입하여 더욱 역동적인 MOB 에이전트를 설계하고자 한다.

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 모형화하며 하나의 측정값에 대한 임계 값이 하나의 곡선으로 표현된다. 따라서 측정값이 임계 값을 넘어설 때 이벤트가 발생한다[3].

이러한 스트레스 모형은 측정값이 임계 값을 넘어설 때 이벤트가 발생한다는 단순한 모형이기 때문에 다양한 행동 패턴을 나타내기 위하여 6 개의 컨트롤 점을 갖는 2 차 베지어 곡선(Bézier curves)으로 표현된 행동특성 곡선을 사용한다. [1]

본 논문에서는 사용자 캐릭터의 접근 또는 공격으로 인한 MOB 의 스트레스에 대하여 설명하고 스트레스의 양을 계산하는 알고리즘에 모든 경우에 대한 스트레스 값을 더하므로 정확한 스트레스 값을 계산하기 위한 공식과 패턴 이외에 행동을 추가 함으로서 보다 역동적인 MOB 에이전트를 설계할 수 있다.

마지막으로 스트레스를 받는 양의 차이를 둠으로써 보다 현실적인 모형을 만들수 있다.

2.1 스트레스 모형

본 논문에서는 MOB 에이전트가 받는 자극을 스트

레스 S 라고 정의하고 스트레스는 공격 자극에 의한 스트레스 As 와 공간 자극에 의한 스트레스 Ds 로 정의한다. 공격 자극의 스트레스 As 는 다음의 (식 1)에서와 같이 공격을 받는 시점에서 큰 폭으로 상승하다가 최대값이 되면 더 이상 증가하지 않는 특징이 있다.

$$As = \alpha C \quad \text{---- (식 1)}$$
(단, $As \geq Max$ 이면 $As = Max$ 이다)

(식 1)에서 α 와 Max 는 상수로 각 MOB 의 수치를 나타낸다. α 는 공격받을 때의 자극의 크기를 결정하는 변수이다. α 값이 크면 공격적 성향을 띠는 MOB 이고 α 값이 작으면 비공격적 성향을 띠는 MOB 에이전트이다.

C 는 공격 카운터로 공격 받은 횟수이다. 공격 자극은 일정치 이상 받더라도 자극의 크기는 같아진다. 즉, As 의 최대 값은 Max 값이 된다. 따라서 스트레스는 공격의 자극 보다는 공간의 자극에 더 많은 영향을 받게 된다.

공간 자극의 스트레스 Ds 는 다음의 (식 2)에서와 같이 MOB 와 사용자 캐릭터와의 거리 D 에 비례하여 시간(t)에 따라 변화 하는 모습으로 정의한다.

$$Ds = 2^t D \quad \text{---- (식 2)}$$
(단, t 의 단위는 ms 이다.)

(식 2)에서 t 에 따라 공간자극의 세기가 달라진다. 그리고 D 는 작을수록 MOB 에이전트와 사용자 캐릭터간의 거리가 멀고 D 가 클수록 MOB 에이전트와 사용자 캐릭터간의 거리가 가까워 진다. Ds 에서의 거리 D 는 다음 (식 3)과 같다.

$$D = Ms - U \quad \text{---- (식 3)}$$

(식 3)에서 Ms 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 Ds 로 계산한다. 즉, MOB 에이전트주의에 많은 사용자 캐릭터가 있을 경우 MOB 가 받는 스트레스의 양은 더욱 많아 지게 되어 행동의 변화가 빨라진다. 그리고 D 의 값이 작으면 MOB 에이전트와 사용자캐릭터간의 거리는 멀어지고 크면 MOB 에이전트와 사용자캐릭터간의 거리는 가깝다. (식 3)의 U 는 다음의 (식 4)와 같다

$$U = \sqrt{(MobX - PlayX)^2 + (MobY - PlayY)^2} \quad \text{-- (식 4)}$$

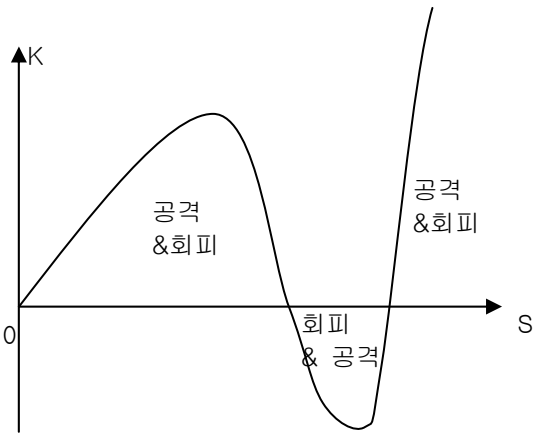
(식 4)에서 MobX 와 MobY 는 MOB 의 좌표 값이고 PlayX 와 PlayY 는 사용자의 좌표 값이 된다. 그리고 U 의 값이 작으면 MOB 에이전트와 사용자캐릭터간의 거리는 가깝고 크면 MOB 에이전트와 사용자캐릭터간의 거리는 멀어지게 된다. MOB 가 받는 스트레스 S 는 다음 (식 5)와 같이 공격 자극에 의한 스트레스 As 와 공간 자극에 의한 스트레스 Ds 의 합으로 나타내어지며 시간의 흐름에 따라 계속 누적된다.

$$S_N = S_{(N-1)} + \sum_{i=0}^{n} As(i) + \sum_{i=0}^{n} Ds(i) \quad \text{---- (식 5)}$$

(식 5)에서 S_N 는 현재 스트레스이고 $S_{(N-1)}$ 는 이전 스트레스의 양이다. i 는 사용자 캐릭터의 수 이다. 만약 MOB 에이전트 시야에 3 명의 사용자캐릭터가 있다고 가정하면 i=0, 1, 2 가 된다. 그리고 각각의 사용자 캐릭터로부터 받는 As 와 Ds 의 합이 MOB 에이전트가 받는 실제 스트레스의 값이 된다. 따라서 많은 사용자 캐릭터가 한 MOB 에이전트 주위에 있거나 공격을 하면 스트레스의 양은 커진다.

2.2 행동 특성 곡선

행동 특성 곡선은 MOB 에이전트 마다 다른 행동 패턴을 부여하기 위한 방법으로 (그림 1)과 같은 모습이다.



(그림 1) MOB 에이전트의 행동 특성 곡선

$$B(t) = \sum_{c=0}^{N} P_c \frac{N!}{c!(N-c)!} t^c (1-t)^{N-c} \quad \text{--- (식 6)}$$
(단, $0 \leq t \leq 1$)

(식 6) $B(t)$ 는 서로 다른 위치에 있는(discrete) N 개의 조절점에 의해 얻어지는 곡선을 구하는 연속함수 있다. 여기서 N 은 조절 점의 개수이고 t = 0 이면 시작 조절점(c = 0)을 의미 하고 t = 1 이면 마지막 조절점 (c = N)을 의미 한다. P_c 는 x, y 좌표 값을

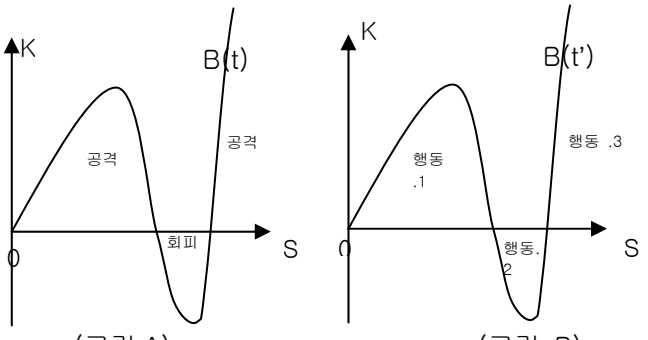
갖는 하나의 순서쌍(x, y 좌표를 가지는 것)를 의미한다. 즉, $P_c = \begin{pmatrix} x \\ y \end{pmatrix}$ 를 의미 한다.

이것을 행동특성 곡선에 적용하면 (식 7)이 된다.

$$B(t) = (1-t)^5 P_0 + 5t(1-t)^4 P_1 + 10t^2(1-t)^3 P_2 + 10t^3(1-t)^2 P_3 + 5t^4(1-t)P_4 + t^5 P_5 \quad \text{(식 7)}$$

행동 특성 곡선은 (그림 1)과 같이 6 개의 컨트롤 점을 갖는 2 차 베지어 곡선으로 표현한다. S 의 값이 증가 함에 따라 K 가 양과 음으로 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 에이전트 마다 개성 있는 행동을 나타나게 할 수 있게 된다.

(그림 1)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. 또한 공격과 회피는 다른 형태로 나타날 수도 있다. 공격과 회피는 본 논문에서는 흥분상태로 설정하였다. 따라서 MOB 에이전트 마다 부여되는 특정한 행동들 또한 흥분 상태로 생각을 할 수 있다. 예로 공격을 하던 중 MOB 에이전트 마다 부여된 특정 스킬을 사용할 수도 있다. 따라서 처음 설정한 행동 특성 곡선(2 차원 베지어 곡선)을 이용하여 처음에는 흥분상태를 결정하고 또 다른 곡선을 사용하여 다음 행동을 결정 할 수 있다는 것이다. 즉, 처음에는 상태를 결정한다. 그리고 그 상태에서 할 수 있는 행동을 결정 지을 수 있다. 본 논문의 공격과 회피는 상태를 의미 할 뿐이다. 따라서 MOB 에이전트는 (그림 2)와 같이 행동의 패턴이 여러 가지 모습으로 나타날 수 있다.



(그림 A) (그림 B)

(그림 2) 상태에서 행동으로 변화 과정

(그림 2)에 (그림 A)의 K 값이 만약 양수가 나와 공격이라는 상태가 되면 (그림 A)의 K 값이 양수 일 동안 (그림 B)와 같이 변화 하여 행동을 변화 시켜 줄 수 있다. 즉, (그림 A)에 $B(t)$ 에 t 와 K 값을 변화 시켜주고 K 와 S 를 작게 하여 새로운 모델 (그림 B)를 만들어 준다. 그러면 (그림 A)에서 K 값이 양수일 때 동안은 (그림 B)의 행동을 변화 시켜 줄 수 있다. (그림 A)는 MOB 에이전트의 상태를 의미하고 (그림 B)는 (그림 A) 상태에서 MOB 에이전트가 할 수 있는 행동을 정의 할 수 있다는 것이다. (그림 B)는 (식 7)에서의 $B(t)$ 을 $B(t')$ 으로 표현한다.

$$B(t') = (1-t')^5 P'_0 + 5t'(1-t')^4 P'_1 + 10t'^2(1-t')^3 P'_2 + 10t'^3(1-t')^2 P'_3 + 5t'^4(1-t')P'_4 + t'^5 P'_5 \quad \text{(식 8)}$$

만약 위와 같이 행동이 3 가지일 경우 나올 수 있는 행동패턴은 총 9 가지이다. 그리고 같은 MOB 에이전트일지라도 스트레스 양에 따라 나올 수 있는 패턴의 모양도 달라질 수 있다.

2.3 스트레스 카운터

보다 현실적인 스트레스 모형을 만들기 위해 여기서 스트레스 카운터를 사용하였다. 스트레스가 없는 상태에서 스트레스를 받게 되면 스트레스를 받는 속도는 스트레스가 있는 상태보다 느릴 것이다. 반면 스트레스가 있는 상태에서 또 다른 자극으로 인하여 스트레스를 받게 된다면 스트레스를 받는 속도는 빨라 질 것이다. 따라서 본 논문에서는 스트레스를 받는 속도에 관련된 변수인 스트레스 카운터를 두어 스트레스를 받는 속도의 변화를 줄 것이다. 만약 스트레스 S 가 있는 경우 MOB 에이전트가 사용자 캐릭터를 죽이면 그 MOB 에이전트의 스트레스 카운터는 0 이 되고 스트레스도 0 이 된다. 그리고 스트레스를 받는 식은 (식 5)’ 이 되어 (식 9)와 같아 진다

$$S_N = \frac{S_{(N-1)} + \sum_{i=0}^{n} As(i) + \sum_{i=0}^{n} Ds(i)}{2} \quad \text{---(식 9)}$$

그리고 스트레스가 0 이 되기 전에 또 다른 자극으로 스트레스에 변화를 받을 경우 그 MOB 에이전트가 받는 스트레스의 양은 (식 10) 과 같아 진다. 그리고 스트레스 카운터는 1 이 된다.

$$S_N = 2(S_{(N-1)} + \sum_{i=0}^{n} As(i) + \sum_{i=0}^{n} Ds(i)) \quad \text{---(식 10)}$$

또한 스트레스 카운터는 초기에는 -1 로 스트레스를 받는 양은 (식 5)와 같다. 즉, 스트레스가 없는 MOB 에이전트는 2 배 느리게 스트레스의 양이 늘어나고 스트레스가 있는 상태의 MOB 에이전트는 스트

레스의 변화가 2 배 빠르게 진행이 된다. 이와 같이 함으로서 스트레스를 받는 속도의 변화를 MOB 마다 다르게 줄 수 있다. 따라서 MOB 에이전트의 행동의 변화는 더욱 역동적이 될 수 있을 것이다.

3. 적용 결과

[1]논문에서 MOB 에이전트는 여러 사용자로부터 스트레스를 받지 못하였다. 본 논문에서는 이와 같은 문제점을 수정을 하였다. 다음 <표 1>은 여러 사용자로부터 스트레스를 받는 양을 비교 한 것이다.

<표 1> 스트레스 양 비교

	플레이어 1	플레이어 n
[1] 논문	S	S
본 논문	S	Sn

또한 스트레스를 받는 속도의 차이를 두었다. 스트레스 카운터를 이용하여 스트레스가 카운터가 -1 인 MOB 에이전트와 스트레스 카운터가 0 인 MOB 에이전트와 스트레스 카운터가 1 인 MOB 에이전트들에 스트레스를 받는 속도를 다르게 하므로 현실적인 스트레스 모형을 제시 한다. 다음 <표 2>는 스트레스 카운터에 따른 스트레스를 받는 양을 비교 한 것이다.

<표 2> 스트레스 카운터에 따른 스트레스 양 비교

	스트레스 카운터 -1	스트레스 카운터 0	스트레스 카운터 1
[1] 논문	S	S	S
본 논문	S	$\frac{S}{2}$	2S

마지막으로 [1]논문에서는 패턴만 제시하였지만 본 논문에서는 행동을 추가하여 패턴 곡선안에 또다른 행동 특성 곡선을 이용하여 행동을 선택 할 수 있게 하였다. 만약 공격에서 행동이 대검 공격, 필살 공격, 마법 공격이 있고 회피에서 도망, 은신, 순간이동이 있다고 가정하자. 다음 <표 3>은 각각의 행동에 대한 비교이다.

<표 3> 공격 및 회피행동 비교

	[1] 논문	본 논문	
공격 행동	일정한 확률을 가지고반복한다.	0~20%	대검 공격
		20~60%	마법 공격
		60~100%	필살 공격
회피 행동	일정한 확률을 가지고반복한다.	0~20%	도망
		20~60%	은신
		60~100%	순간이동

<표 3>에서와 같이 스트레스의 양에 따라 서로 다른 공격과 회피를 하고 있다는 것을 알 수 있다.

4. 결 론

본 논문에서는 [1]논문에서의 문제점을 보완하여 보다 현실적인 스트레스 모형을 설계하였다. 수식ㄱ나 알고리즘을 변경하였고 행동 특성 곡선에 새로운 모습이 그려지게 되도록 변경하였다. 따라서 알고리즘이 보다 유연해지게 되고 패턴 안에서 다시 행동으로 나뉘어 지기 때문에 같은 MOB 에이전트라고 해도 다른 행동을 할 수 있기 때문에 사용자는 보다 역동인 MOB 에이전트를 상대할 수 있게 된다. 따라서 실사용자들에게 보다 역동적인 게임을 제공하여 사용자로 하여금 게임에 몰입을 할 수 있도록 만들어준다.

기존 스트레스 모형에서는 스트레스를 이전의 스트레스에 한 명의 사용자에게만 스트레스를 받게 되어있었지만 본 논문에서 제시한 알고리즘에서는 여러 사용자에게 동시에 스트레스를 받게 설계되어 더욱 빠른 상태 변화가 일어 날수 있게 되었고 스트레스 카운터를 통해 각각 MOB 에이전트의 스트레스를 받는 속도에 변화를 주었다. 그것으로 인해 같은 시야에 있는 MOB 에이전트들이 서로 다른 스트레스를 받게 된다. 또한 패턴 안에 행동을 만들어서 하나의 패턴이 결정되면 그 안에 또 다른 행동 특성곡선이 만들어져서 행동을 결정하게 된다. 따라서 MOB 에이전트는 사용자 캐릭터에게 각기 다른 행동을 보이므로 지루했던 온라인게임의 사냥을 즐겁게 할 수 있을 것이다.

향후 연구 과제는 초기 상태의 결정에 퍼지 이론(fuzzy theory)과 같은 유연한 방법의 도입이 필요하다.

참고문헌

[1] 김진수, 방용찬 " 다사용자 온라인 게임을 위한 지능형 MOB 에이전트 설계", 한국정보처리학회, 제 12 권 제 1 호, 2005.

[2] 이만제 " 게임에서의 인공지능 기술", 정보처리, 제 9 권, 제 4 호, 2002.

[3] 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.

[4] Paul Bourke " http://astronomy.swin.edu.au/~pbourke/curves/bezier/" 1996.

[5] 류광 역 『 3D 게임 프로그래밍 & 컴퓨터 그래픽을 위한 수학 제 2 판』 pp.530-536, 2004.

[6] Chen, Q. "A class of Bezier-like curves", Computer aided geometric design, Vol.20, No.1, 2003.