

FSM 기반의 게임 NPC 인공 지능 평가

이면재

백석대학교 정보통신학부

davidlee@bu.ac.kr

An Artificial Intelligence Evaluation on FSM-Based Game NPC

MyounJae Lee

Division of Information & Communication, BaekSeok University

요약

게임 NPC(Non Player Character)는 게임 플레이어와 대전 또는 협력함으로써 게임의 재미를 증가시키는 중요한 요소이다. 대부분 기존 게임에서 제공되는 NPC 인공지능은 FSM(Finite State Machine)으로 제작되어 행동 패턴이 정해져 있고 능력이 동일한 특징을 갖고 있다. 따라서 이러한 특징을 갖는 NPC들과 대전하는 플레이어는 창조적인 게임 플레이를 진행하는 것이 어려울 수 있다.

본 논문은 이 문제점을 개선하기 위하여 실제 생활에서 늑대들이 먹이를 사냥하는 행동 모델을 게임 NPC의 행동 모델로 제작하고 이를 평가하기 위한 것이다. 이를 위하여 첫째, 실세계에서 늑대들이 먹이를 포획하기 위한 행동 상태들을 조사 연구한다. 둘째, 이 행동 상태들을 Unity3D 엔진을 이용하여 구현한다. 셋째, 구현된 NPC들의 상태 전이 비율과 실세계의 NPC들의 상태 전이 비율, 일반적인 게임 NPC의 상태 전이 비율을 비교한다. 비교 결과, 구현된 NPC들의 상태 전이 비율은 실세계의 상태 전이 비율과 비슷함을 보인다. 이는 구현된 NPC들의 행동 패턴이 실세계의 늑대 사냥 행동 패턴과 유사함을 의미하는데, 이렇게 함으로써 플레이어에게 보다 증가된 사용자 경험을 제공할 수 있다.

ABSTRACT

NPC in game is an important factor to increase the fun of the game by cooperating with player or confrontation with player. NPC's behavior patterns in the previous games are limited. Also, there is not much difference in NPC's ability among the existing games because it's designed to FSM. Therefore, players who have matched with NPCs which have the characteristics may have difficulty to play.

This paper is for improving the problem and production and evaluation of the game NPC behavior model based on wolves hunting model in real life. To achieve it, first, the research surveys and studies behavior states for wolves to capture prey in the real world. Secondly, it is implemented using the Unity3D engine. Third, this paper compares the implemented state transition probability to state transition probability in real world, state transition probability in general game. The comparison shows that the number of state transitions of NPCs increases, proportions of implemented NPC behavior patterns converges to probabilities of state transition in real-world. This means that the aggressive behavior pattern of NPC implemented is similar to the wolf hunting behavior pattern of the real world, and it can thereby provide more player experience.

Keywords : Artificial Intelligence(인공 지능), Behavior Pattern of NPC(NPC 행동 패턴), Hunting Model(사냥 모델), Player Experience(플레이어 경험)

Received: Sept. 06, 2014 Accepted: Oct. 07, 2014

Corresponding Author: MyounJae Lee(BaekSeok University)

E-mail: davidlee@bu.ac.kr

© The Korea Game Society. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

ISSN: 1598-4540 / eISSN: 2287-8211

1. 서 론

컴퓨터 게임이 처음 소개 될 당시에는 소프트웨어나 하드웨어의 제약 때문에 게임 인공 지능 기술을 게임 제작에 활용하는 데에 한계가 있었다. 그러나 최근에는 하드웨어 기술의 발달과 소프트웨어 기술의 발달, 플레이어들의 사용자 경험 향상에 관한 요구들로 인하여 다양한 영역에서 게임 인공 지능 기술이 활용되고 있다[1].

게임에서 사용하는 대표적인 인공 지능 기법들에는 그래프 형태로 유한개의 상태들이 연결되어 있는 규칙 기반 시스템으로서 적이나 NPC들의 상태를 제어하는 FSM(Finite State Machine)[2], NPC 들의 집단 형성[3], 학습을 통해 행동을 결정하는 신경망(Neural Net)[4], 자연 생태계의 진화 이론을 이용하여 캐릭터의 상태와 행동을 결정하는 유전자 알고리즘[5]과 길찾기(Path-Finding)[6] 등이 있다. 이와 같이 다양한 인공지능 기술을 게임에 적용하는 경우 플레이어들은 보다 높은 사용자 경험을 느낄 수 있다[7,8].

특히 이 중에서 NPC는 플레이어들과 협력하거나 대전하는 대상이기 때문에 플레이어들의 주요 관심사이며 플레이어들의 게임 몰입에 중요한 역할을 수행한다. 플레이어에 비해 NPC들의 능력이 너무 약하거나 강한 경우 플레이어들은 다시 NPC들과 대전하고 싶은 마음이 줄어들 수 있기 때문에 플레이어의 능력에 적절한 NPC 능력 설계가 필요하다. 따라서 NPC의 행동 패턴과 능력 등을 실제와 같이 다양하게 설계하는 것은 플레이어들에게 NPC들의 다음 행동 패턴에 대한 예측을 어렵게 하여 주기 때문에 NPC에 대한 인공 지능 연구는 중요하다.

NPC에 대한 인공 지능 연구에는 사람과 같이 행동하는 지능형 기술 개발을 위하여 주로 기존 인공 지능 기술에서와 같이 신경망, 유전자 알고리즘, 사례 기반 추론, 학습과 같은 방법[9]들이 주로 사용되어져 왔으며 NPC 그룹에 대한 행동에 관한 무리짓기 연구[10], 플레이어의 행동을 데이터베이

스에 저장하여 플레이어의 행동에 적합하게 대응하는 적응형 NPC에 관한 연구[15], 확률 FSM의 추론 방식을 이용하여 NPC의 특성을 자동적으로 게임에 반영하는 연구[16] 등이 있다.

NPC 인공 지능은 주로 FSM 방법으로 제작된다. FSM 방법은 상태의 개수가 적은 경우 NPC 행동이 단순하여 플레이어가 쉽게 NPC의 행동을 예측할 수 있으며 많은 경우 구현이 복잡한 특징을 갖고 있다. FSM 방법으로 NPC 인공 지능을 제작하는 경우 게임 상황에 맞추어서 현재 상태에서 다음 상태로 무조건 전이되는 방법[8,13]과 플레이어의 입력과 게임 상황과 관계없이 무조건 확률 분포에 따라 다음 상태로 이동하는 방법[16] 등이 있다. 두 방법 모두 실세계에서 동물이나 사람들의 공격 행동과는 너무 차이가 발생하여 플레이어가 NPC와 대전하는 재미가 상실될 수 있다.

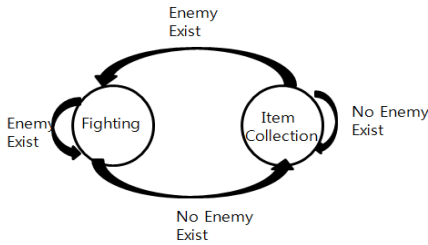
본 논문은 플레이어와 대전하는 경우 실세계의 동물 공격 행동과 유사한 NPC들의 인공 지능을 구현하고 구현된 NPC 인공 지능을 상태 전이 확률로 비교 평가한다. 이를 위하여 본 논문은 첫째 실제 늑대 사냥 행동 패턴을 관찰하고 분석한 연구 논문[11,12]을 Unity3D 엔진을 이용한다. 늑대 사냥 행동 패턴은 검색, 접근, 공격, 먹이 포획 상태로 구성되며 현 상태에서 다른 상태들로 전이되기 위한 확률 값을 갖고 있다. 이 확률 값을 실제 늑대 행동 패턴을 추적하여 관찰한 연구[11,12]의 결과 값이다. 둘째, 구현된 늑대 사냥 행동 패턴의 상태 전이 확률 값과 실세계의 늑대 사냥 행동 패턴의 상태 전이 확률 값과 일반적으로 사용되는 NPC 인공 지능의 상태 전이 확률 값을 비교한다. 본 논문에서는 이 값을 비교함으로써 FSM 기반의 게임 NPC 인공 지능을 평가하는데, 실세계의 상태 변화 확률 값에 근사할수록 NPC 인공 지능은 우수한 것이다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구를 설명하고 3장에서 구현 결과를 기술한다. 4장에서 결론 및 추후 연구 방향을 논한다.

2. 관련 연구

2.1 FSM 인공 지능

FSM은 유한 개의 상태와 조건으로 구성되는 것으로 각 상태와 상태의 전이를 위해 필요한 외부 조건 즉 이벤트를 정의하여 상태 전이를 구현하는 방법이다. FSM을 NPC 인공 지능에 적용하는 경우 먼저 NPC의 행동 패턴을 여러 개의 상태로 나누고 상태에 따른 외부 조건을 정의한다. 이후에 현재 상태와 현 상태에 입력될 수 있는 이벤트, 그리고 해당 이벤트에 따른 다음 상태를 정의의한다. [Fig. 1][8,13]의 FSM은 전투 상태와 아이템 수집 상태로 구성된다. 전투(아이템 수집) 상태인 경우 적이 없으면(있으면) 아이템 수집(전투) 상태로 이동한다.

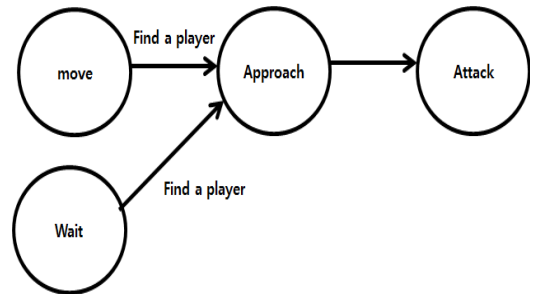


[Fig. 1] An Example of FSM

이 방법은 NPC의 행동 개수가 상태의 개수에 따라 제한을 받기 때문에 플레이어들에게 NPC 행동이 예측될 수 있다. 또한 적의 유무에 따라 상태가 무조건 결정되기 때문에 다음 상태를 쉽게 예측할 수 있다.

Half-life 2의 경우 플레이어가 이동 중 여러 곳에 배치된 NPC와 전투를 벌이다가, 보스 NPC를 만나서 제거하면 스테이지가 클리어 되는 게임이다. 이 게임 또한 NPC들이 FSM 방법으로 구현되었는데 NPC 상태들은 지정된 경로를 따라 이동하거나 제자리에서 기다리는 대기 상태, 플레이어를 발견하게 되면 플레이어에게 접근하는 상태, 플레이어를 공격하는 상태로 구성된다. 이 게임에서도

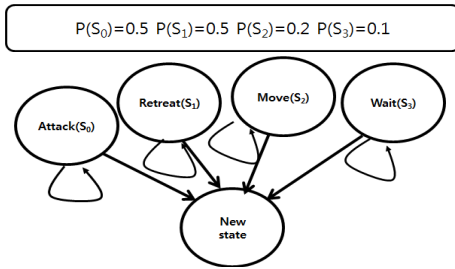
특정 조건만 맞으면 현재 상태에서 다음 상태로 무조건 전이된다. [Fig. 2]에서는 Half-life 2 게임에서의 NPC들의 FSM을 보여준다. [Fig. 2]와 [Fig. 3]에서와 같은 FSM으로 NPC 인공 지능이 제작되는 경우 임의의 상태에 있을 경우 이벤트가 들어오면 다음 상태는 항상 동일한 상태가 된다.



[Fig. 2] FSM of Half-life

이를 개선한 확률적 FSM 방법은 현재 상태에서 다음 상태를 결정할 때 확률에 따라 다음 상태를 결정한다. [Fig. 3]은 이를 보여준다. 이 방법에서는 플레이어의 입력을 고려하지 않고 다음 상태를 결정하기 때문에 NPC가 다소 엉뚱한 행동을 할 수 있다. 또한 확률 값에 따라서 다음 상태가 결정되기 때문에 사용자 경험에 적합한 값을 설정해야 하는 어려움이 있다.

[Fig. 1]과 [Fig. 2], 그리고 [Fig. 3] 방법으로 제작된 NPC 인공 지능의 경우 모두 현재 상태에서 다음 상태로 순차적으로 전이한다. 즉 외부 상황과 조건을 고려하지 않고 현재 상태에서 다음 상태로 전이한다. 예를 들어 [Fig. 2]의 경우 실세계에서는 공격(Attack) 상태에서 다시 접근(Approach) 상태로 가거나 접근 상태에서 이동(Move) 상태로 가는 경우가 발생할 수 있는데 이 경우가 고려되지 않은 문제점을 갖고 있다. 따라서, 실세계에서의 동물들의 상태 전환을 NPC 인공 지능에 적용하는 것이 필요하다.



[Fig. 3] Stochastic FSM

2.2 늑대 사냥 모델

NPC 인공지능이 [Fig. 1]과 [Fig. 2]와 같이 제작된 경우 이벤트에 따라서 무조건 다음 상태로 전이되었기 때문에 플레이어가 NPC의 다음 상태를 쉽게 알 수 있다. 또한, [Fig. 3]과 같이 확률적 FSM으로 제작된 NPC 인공지능의 경우 플레이어의 입력 즉 이벤트를 고려하지 않고 확률에 따라 무조건 다음 상태로 전이되기 때문에 NPC들은 플레이어의 행동 반응에 적합하게 대응하지 못하는 문제점을 갖고 있다.

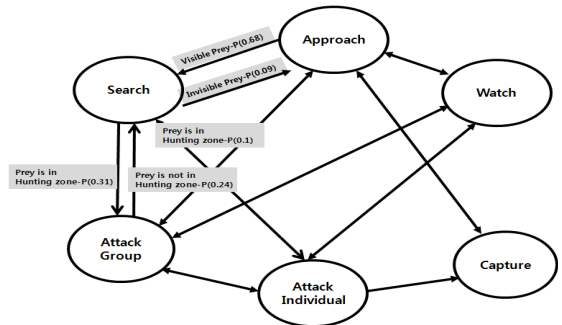
본 연구는 이러한 문제점을 개선하기 위하여 실세계에서의 늑대 사냥 패턴들을 NPC 인공지능으로 제작하기 위한 시도이다. 늑대들의 사냥 패턴은 여러 상태로 구성되어 있고 각 상태들은 이벤트와 확률 값에 따라서 다음 상태로 전이된다. 즉, [Fig. 1]과 [Fig. 2]의 FSM과 [Fig. 3]의 FSM을 결합하되 실세계의 상태 전이 확률 값을 사용함으로써 플레이어들에게 실제감을 제공한다.

늑대들의 사냥 행동 패턴에 관한 연구는 1995년 도부터 2000년까지 Yellowstone National Park에서 진행되어졌다. 늑대 51마리를 풀어놓은 후 이 늑대들의 행동에 대해 2명의 관찰자가 정리하였다 [14]. 늑대는 51마리로 시작 되었지만 해마다 30%-50%정도의 새끼를 낳아서 110마리까지 관찰되었다. 늑대들은 2-7개의 그룹을 형성하였고 각 그룹은 2마리부터 27마리로 구성되었다.

[Fig. 4]는 늑대 사냥 행동의 FSM으로 실세계에서의 다음 상태로 전이되기 위한 이벤트와 Search 상태에서의 상태 전이 확률 값의 예를 보

여준다. [Fig. 4]의 FSM에서 현재 상태에서 다음 상태로 전이되기 위한 확률 값[11,12]은 [Table 1]이다. 기존 FSM 방법인 [Fig. 1], [Fig. 2]와 [Fig. 3]와 다르게 이벤트와 상태 전이 확률 값이 모두 표시되어 있다. Search 상태는 먹이를 검색하는 상태이며 Approach 상태는 먹이를 발견하고 접근하는 상태이고 Attack Group 상태와 Attack Individual 상태는 먹이를 포획하기 위해 개인 또는 그룹이 공격하는 상태이다. Capture 상태는 먹이를 포획한 상태이다.

(eq. 1)은 늑대 이동 크기[14]를 나타낸다. r 은 늑대부터 먹이까지의 거리를 나타내고, C 영역은 늑대의 “controlled” zone으로써 늑대가 먹이를 사냥하기 위해 느리게 이동하는 영역이다. 늑대에서 먹이까지의 거리가 C 영역보다 작은 경우에 늑대는 먹이와의 거리를 일정하게 유지하면서 공격이나 포획 기회를 잡기 위해 기다린다. r 이 D 영역보다 작은 경우 늑대의 이동 크기는 무한대 즉 먹이를 포획하기 위해 매우 빠르게 이동한다. D 영역은 먹이의 “Death” zone, 먹이가 늑대에게 포획될 반경을 나타낸다. 먹이가 D 영역 안에 있는 경우 늑대는 먹이를 포획할 수 있다.



[Fig. 4] FSM of wolves behavior

[Table 1] Probabilities of Transitions Between States: Search, A:Approach, W:Watch, AG: Attack Group, AI:Attack Individual, C:Capture)

Preceding State	Following State					
	S	A	W	AG	AI	C
S	.00	.68	.00	.31	.01	.00
A	.09	.00	.12	.69	.09	.01
W	.32	.35	.00	.27	.06	.00
AG	.24	.09	.03	.13	.51	.00
AI	.16	.06	.02	.16	.08	.52

$$\text{magnitude} = \begin{cases} 0, & r > C \\ \frac{r - D}{C - D}, & D < r \leq C \\ \infty, & r \leq D \end{cases} \quad (\text{eq. 1})$$

3. 구현

3.1 구현 방법

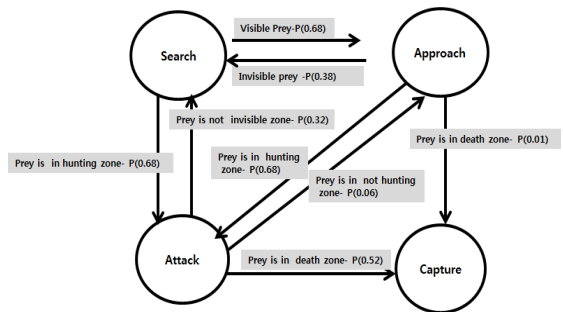
본 논문의 개발 환경은 [Table 2]와 같다.

[Table 2] Implementation environment

Platform	Specification
CPU	Intel(R) 4x, 3400 MHz
RAM	(4GB DDR3 PC3-12800) x2
HDD	Seagate 2TB
VIDEO	GeForce GTX 660
OS	Window 7
Program	Unity 4

[Fig. 5]는 [Fig. 4]의 늑대 사냥 FSM 모델을 구현을 위해 간략화 한 것이다. Watch 상태는 Capture 상태로 전이되는 확률이 0이므로 Search 상태에 포함시키고[11,12,14], 늑대들이 그룹으로 사냥하여 먹이를 포획하는 확률이 0이므로 Attack_Group 상태를 이전 상태에서 제거하되 다음 상태의 Attack Group과 Attack Individual 상태를 모두 Attack 상태로 통합시켰다. 예를 들어, Approach 상태에서 Attack 상태로 전이되는 확률

값은 0.78인데, 이는 [Table 1]에서의 Attack_Group 확률 값(0.69)과 Attack_Individual 확률 값(0.09)을 더한 값으로 계산했기 때문이다. 실제 세계에서 Attack 상태에서 Capture 상태로의 상태 전이 확률 값이 0.52인데 이는 Attack_Individual에서 Capture 상태로 전이되는 확률 값(0.52)와 Attack_Group에서 Capture 상태로 전이되는 확률 값(0.00)을 더한 값이다.



[Fig. 5] FSM of wolves behavior for Implementation

[Table 3]은 구현에 사용된 상태 전이 확률값을 보여준다.

[Table 3] Probabilities of Transitions Between States for Implementation

Preceding State	Following State			
	Search	Approach	Attack	Capture
Search	.00	.68	.32	.00
Approach	.21	.00	.78	.01
Attack	.18	.06	.24	.52

[Table 4]는 Approach 상태에서의 상태 변화를 위한 의사코드(Pseudo code)를 보여준다. 단계 (1)에서 각 늑대들과 먹이의 거리를 계산한다. 이때 늑대들 각각은 객체로 제작되었기 때문에 먹이와의 거리는 독립적으로 계산된다. 단계 (2)에서 control zone, 즉 늑대가 먹이를 사냥할 수 있는 거리에 있는 경우에는 단계 (3)에서 (10)까지가 진행된다. 늑대와 먹이의 거리가 control zone보다 큰 경우에는 Search 상태가 지속된다. 단계 (3)에서는 랜덤 함

수를 실행하는데, 이때 숫자 범위는 1부터 100까지로 한정한다. 이때 랜덤 함수를 이용하여 생성된 값을 Prob에 저장한다. Prob가 79보다 작은 경우에는 다음 상태로 Attack 상태가 된다. 79보다 작은 경우는 1부터 78까지로 이 범위에 포함된 숫자가 생성될 확률은 0.78이 된다. 0.78값은 [Table 3]의 Attack 상태로 전이될 확률 값과 동일하다. 단계 (6)에서는 랜덤으로 생성된 숫자가 79보다 같거나 크고 98보다 같거나 작은 경우 다음 상태를 Search 상태로 변화시킨다. State는 다음 상태를 저장하는 변수이다.

[Table 4] Pseudo code for state transition of wolf from Approach state

```

(01) Distance=(wolf,prey)
(02) if (Distance<C_zone) {
(03)     Prob= Random.range(1,100)
(04)     if (Prob<79)
(05)         State=Attack
(06)     else if(79<= Prob <=99)
(07)         State=Search
(08)     else
(09)         State=Capture
(10) }
    
```

3.2 구현 결과

본 절에서는 2장에서 기술된 늑대 행동 패턴의 상태 변화를 구현한 경우의 상태 변이 확률 값과 실세계에서의 상태 변이 확률 값, 즉 [Table 3]에서의 값과 일반적인 게임에서 상태 변이 확률 값을 비교한다. 일반적인 게임에서의 상태 변이 확률 값은 외부 조건만 만족되는 경우 무조건 다음 상태로 전이 되므로 1으로 설정한다. 이때 늑대의 마리 수는 1이며 매 프레임마다 이동하는데 비하여 먹이는 1개이고 고정적인 위치를 갖는다. (eq. 1)의 C와 D 값은 각각 8과 4로 설정하였다. 늑대와 먹이의 초기 위치는 고정적이며 늑대가 먹이를 먹으면 먹이 위치는 랜덤 함수로 결정된다.

[Fig. 6]은 Search 상태에서 Approach 상태로의 상태 전이 확률 값을 비교한 결과이다. X축은

Search 상태에서 발생하는 총 상태 변이 횟수이며 Y축은 그 중에서 Approach 상태로 변이되는 비율을 나타낸다. 이 상태는 늑대가 먹이를 찾기 위해 배회하다가 먹이를 발견하고 접근하는 상태로 변이되는 과정이다. 구현된 결과 값이 상태 전이 횟수가 많아질수록 실세계에서의 상태 변이 확률 값인 0.68에 수렴됨을 보여준다. 일반적인 NPC 인공 지능의 경우 무조건 Search 상태에서 Approach 상태로 전이되므로 1이 된다.

[Fig. 7]은 Approach 상태에서 Attack 상태로의 상태 전이 확률 값을 비교한 결과이다. 이 상태는 늑대가 먹이를 찾기 위해 접근하다가 먹이를 포획하기 위해 공격하는 과정으로 NPC가 플레이어를 공격하기 위해 접근하다가 공격을 시도하는 경우에 적용될 수 있다. 구현된 결과 값이 [Fig. 6]에서와 같이 상태 전이 횟수가 많아질수록 실세계에서의 상태 변이 확률 값에 수렴됨을 보여준다.

[Fig. 8]은 Attack 상태에서 Capture 상태로의 상태 전이 확률 값을 비교한 결과이다. 이 상태는 늑대가 먹이를 포획하기 위해 공격하여 포획에 성공한 경우로서 NPC가 플레이어를 공격하여 생명치를 감소시킨 경우에 적용될 수 있다. [Fig. 5]와 [Fig. 6]과 동일하게 구현된 경우의 상태 전이 확률 값은 상태 전이 횟수가 많아질수록 실세계에서의 상태 변이 확률 값에 수렴됨을 보여준다.

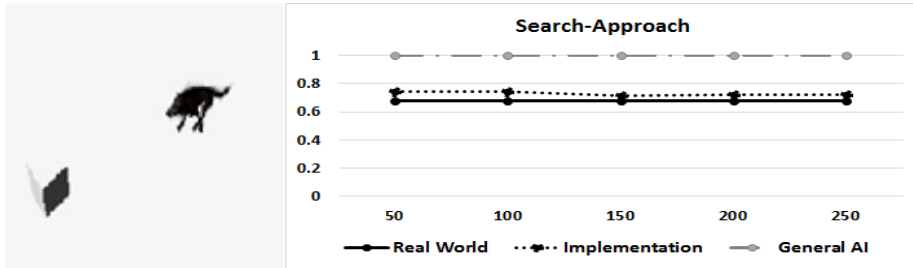
[Table 5]는 [Fig. 6]부터 [Fig. 8]까지의 상태 전이 확률의 평균을 비교한 것이다. 구현된 경우의 상태 전이 확률은 실세계의 상태 전이 확률과 큰 차이가 없음을 알 수 있다.

[Table 5] Mean Probabilities of Transitions Between States

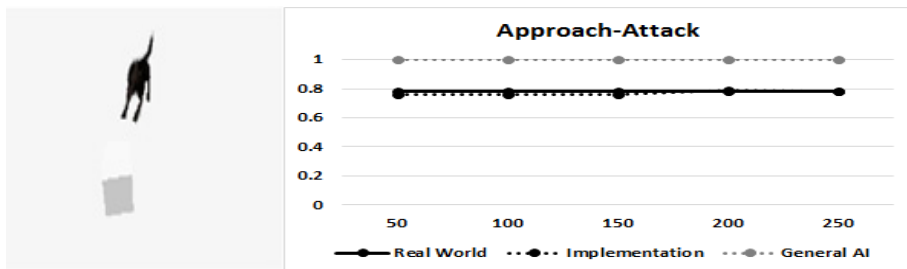
Transition		AI		
Proceeding State	Following State	Real World	Implementation	General
Search	Approach	0.68	0.726	1
Approach	Attack	0.78	0.77	1
Attack	Capture	0.52	0.556	1

결론적으로 구현된 경우의 NPC 행동 패턴의 상태 전이 확률은 상태 전이 횟수가 많아질수록 실세계의 상태 전이 확률에 수렴됨을 보인다. 이는

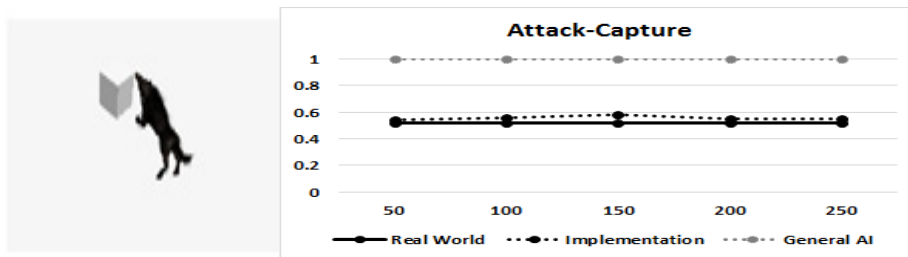
구현된 NPC 행동 모델이 플레이어에게 실세계에서의 늑대가 플레이어를 공격하는 듯한 느낌을 제공할 수 있다는 것을 의미한다.



[Fig. 6] Comparison of transition from Search state to Approach state



[Fig. 7] Comparison of transition from Approach state to Attack state



[Fig. 8] Comparison of transition from Attack state to Capture state

4. 결론 및 추후 연구 방향

본 논문은 단순하고 반복적인 행동 패턴으로 플레이어를 공격하는 NPC들의 행동 패턴의 문제점을 개선하고 개선된 NPC들의 행동 패턴을 다른

NPC들의 행동 패턴과 비교 평가하기 위한 연구이다. 이를 위하여 실세계의 늑대 사냥 행동 모델을 게임 NPC 인공 지능에 접목하기 위하여 늑대 사냥 행동의 상태 즉 Search, Approach, Attack, Capture 상태를 Unity3D 엔진을 이용하여 구현하고 실세계의 늑대 사냥 행동의 상태 전이 확률 값과 구현된 경우의 상태 전이 확률 값, 그리고 일반

적인 게임에서의 NPC 상태 전이 확률 값을 비교하였다. 비교 결과, 상태 전이 횟수가 많아질수록 구현된 경우의 상태 전이 확률 값은 실세계의 상태 전이 확률 값에 수렴함을 알 수 있었다. 이에 비교하여 일반적인 게임에서의 NPC 상태 전이 확률은 실세계에서의 상태 전이 확률과 많은 차이가 발생하였다.

본 논문에서 제안된 늑대 사냥 행동 모델을 실제 NPC 인공 지능에 적용하는 경우 플레이어의 게임 몰입에 도움을 줄 것이라고 판단된다. 추후에는 본 논문의 늑대 사냥 행동 모델을 실제 게임 NPC에 적용하여 게임 몰입을 비교할 예정이다.

ACKNOWLEDGMENTS

이 논문은 2014도 백석대학교 대학 연구비에 의하여 수행된 것임(This research is supported by 2014 BaekSeok University Fund).

REFERENCES

- [1] Lee Eun-Hee, Park Choong-Shik, Cho Sung-Hyun, "A Study on the Intelligent NPC in MMORPG", Proceedings of Korea Contents Society, Vol.4, No.1, 2006.
- [2] Manjae Lee, "Artificial Intelligence in Game", Korea Information Processing Society, Vol.9, No.3, pp.69-76, 2002.5
- [3] N.D Cho, B.G Sung, K.T Kim, "Artificial life simulation game characters through the implementation of the strategy", Spring Conference of KISSE, Vol.27, No.1, pp.241-243, 2000.
- [4] David B. Fogel, "Using Evolutionary Programming to Create Neural Networks That are Capable of Playing Tic-Tac-Toe", Int'l Joint Conf. Neural Networks, New York, pp.875-880, 1993.
- [5] Marc Ponsen, IMPROVING ADAPTIVE GAME AI WITH EVOLUTIONARY LEARNING, a thesis submitted in fulfillment of the requirements for the degree of Master of Science, Delft, 2004.
- [6] Ross Graham, Hugh McCabe and Stephen Sheridan, "Path finding in Computer Games", ITB J., Vol. 9, pp.223-230, 2004.
- [7] Incheol Kim, "Utilizing Computer Games for Effective AI Education, Korean Society For Computer Game", Vol. 26, No. 3, pp.109-118, 2014.
- [8] MyounJae Lee, "Implementation of NPC Artificial Intelligence Using Agonistic Behavior of Animals", Journal of digital convergence, Vol.12, No.1, 2014.
- [9] Myoung-hee Cha, "A Study for Autonomous Learning capacity for plan-based NPC", Korean Society For Computer Game, Volume.25, No.3, pp.149-156, 2012.
- [10] Jae Moon Lee, "An Efficient Flocking Behaviors for Large Flocks by Using Representative Boid", Journal of Game Society, Vol.8, no. 3, pp.87-95, 2008.
- [11] J. D. Madden, R.C. Arkin, D. MacNulty, "Multi-robot System Based on Model of Wolf Hunting Behavior to Emulate Wolf and Elk Interactions", Proc. IEEE International Conference on Robotics and Biomimetics (ROBIO 2010), Tianjin, China, Dec. 2010.
- [12] D.R. MacNulty, L.D. Mech, D.W. Smith, "A Proposed Ethogram of Large-carnivore Predatory Behavior, Exemplified by the Wolf", Journal of Mammalogy, Vol. 88, No. 3, pp.595-605, June 2007.
- [13] H.S Park, K.J Kim, "Current research of artificial intelligence games", KISSE, 2013.7
- [14] A. Sánchez and A. Weitzenfeld, "Multi-Agent Formations in a Herd of Wolves Hunting Model", Technical report, Instituto Tecnológico Autónomo de México, Mexico City, Mexico 2004.
- [15] Nara Kim, Kyhyun Um, Kyungeun Cho, "An Action Information Management Method for Creating Adaptive NPC", Journal of Game Society, Vol.8, No.3, 2008.3.
- [16] Kyungeun Cho, Hyungje Cho, "Generating various NPCs Behavior using Inference of Stochastic Finite Automata", Journal of Game Society, Vol.2, No.2, 2002.11.



이 면 재 (MyounJae Lee)

2009.3-현재 백석대학교 정보통신학부 멀티미디어전공
부교수

관심분야 : 3D 게임 프로그램, 인공지능, 기능성 게임
