



게임에 생명을 불어넣기 위한 첨단 기법

글 | 한국전자통신연구원(ETRI) 김현빈 외 가상현실연구부

연재 순서

1. 3D 게임엔진에 대하여
2. 렌더링 엔진
3. 애니메이션 엔진
4. 사운드 엔진
5. 서버 엔진
6. 게임 인공지능(이번호)
7. 맵 에디터
8. 게임진행 모듈

초기의 컴퓨터 게임은 하드웨어, 그래픽, 사운드 등의 제약 때문에 인공지능에 있어서는 거의 전부가 게임 프로그래머의 단순한 하드코딩에 의존해 단순하게 처리됐다. 그러나 그래픽이나 사운드의 기능이 일정한 수준에 도달하자 게이머들은 보다 자연스럽게 재미 있는 게임을 요구하게 됐으며, 이로 인해 1990년대 후반부터 인공지능 기술이 게임에서 중요한 역할을 하기 시작했다. 게임에서의 CPU 점유율을 예로 들면, 1997년도에는 5% 이하였으나, 2000년도에 들어서면서 거의 30%에 가까운 점유율로 증가했다. 또한 인공지능 전담 프로그래머의 비율도 1997년에는 개발사의 25%에 불과하던 것이, 2000년도에 들어서 80%로 급격히 증가하게 됐다.

'게임 인공지능'이란 고전적인 의미에서는 게임 내에서 컴퓨터에 의해 제어되는 캐릭터나 에이전트로 정의하기도 하지만 최근에는 좀더 구체적으로 스스로 생각할 수 있고 주변 환경이나 과거의 경험 등에 따라서 지능적으로 행동할 수 있는 자율성을 가진 캐릭터나 에이전트라고 정의하기도 한다. 그러면 인공지능이 게임에서 담당할 수 있는 역할은 무엇일까? 우선 게임에서의 인공지능은 등장 캐릭터의 지능적인 행동을 구현함으로써 게이머가 조작하지 않는 NPC(Non-Player

Character)들의 움직임을 자연스럽게 제어하거나 게이머의 상대 역할 또는 보조자 역할을 한다.

이 경우 게임의 핵심은 인공지능과의 대결이다. 인공지능은 게이머에게 무조건 이기는 것이 목적이 아니라 유사한 수준의 상대 역할을 수행해줄 수 있어야 한다. 또한 RPG와 같은 경우에 주인공의 보조자 역할을 해주거나 게임의 초보자를 이끌어 주는 역할을 하는 캐릭터가 될 수도 있다. 그리고 인공지능은 게임에서의 애니메이션 동작 제어를 담당하거나, 캐릭터가 현재의 위치에서 목적지까지 갈 수 있도록 이동경로를 찾아주는 역할을 하기도 한다.

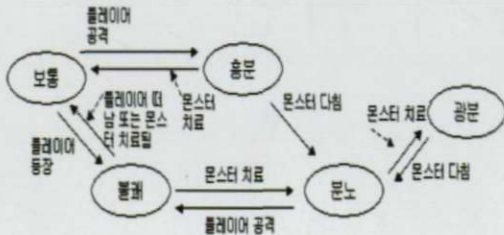
게임에서의 인공지능은 한두 가지의 알고리즘만으로 구현하기는 힘들다. 게임의 수준과 장르에 따라서 다르겠지만 여러 가지의 인공지능 기법을 조합해 하나의 게임에 필요한 인공지능을 적용할 수 있는 것이다. 그러면 게임에서 적용되고 있는 몇 가지의 대표적인 방법들에 대해 알아보기로 한다.

FSM(Finite State Machine)

FSM은 현재 가장 널리 사용되는 인공지능 처리 방식 가운데 하나이다. FSM이란 유한



한 개수의 상태(sate)를 이용해 NPC의 행동 양식을 표현하거나 게임 세계를 관리하는 방법이다. 상태란 행동처리를 위한 기본 단위가 되며, 각 상태는 주어지는 조건에 따라서 다른 상태로 전이될 수가 있다. <그림1>은 한 몬스터의 행동 양식을 FSM으로 표현한 예이다.



<그림1> 몬스터의 FSM

<그림1>의 예에서 볼 수 있듯이 몬스터의 행동 양식은 여러 개의 상태로 나누어지며 현재의 상태와 조건에 따라서 외부에 대처하는 방법이 결정된다. FSM은 이해하기 쉽고 구현도 어렵지 않아 특별히 뛰어난 인공지능을 필요로 하지 않는 대부분의 게임에서 사용된다. 그러나 복잡한 게임의 경우 상태의 수가 많아지게 되고 그에 따라 상태 다이어그램을 정리하기 어려워지며, 상태 변화를 가능하게 하는 외부 입력(조건) 루틴이 급속도로 복잡해진다. 또한 FSM을 게임의 상대방이나 몬스터로 적용한 경우 게임 진행 중 유사한 경험을 몇 번 겪고 나면 인공지능의 행동 패턴을 예측할 수 있게 되므로 게임의 재미가 반감될 수도 있다. 이런 문제점들을 해결하기 위해 상태의 수가 많아질 경우 상태들을 몇 개의 그룹으로 묶어 계층적으로 구성하는 계층적 FSM를 사용하기도 한다.

이런 계층적 FSM은 하프 라이프(Half Life)라는 게임에서 사용됐다. 그리고 퍼지 FSM은 퍼지(fuzzy) 이론을 접목해 상태의 입력과 출력에 퍼지 함수를 적용해 동일한 외부 상황에도 다른 출력을 얻도록 함으로써 상대방이나 몬스터의 행동을 예측하기가 어렵게 돼 보다 현실적인 게임을 즐길 수 있도록 해준다.

길찾기(Path Finding)

게임에서 가장 자주 등장하는 현실적인 문제 가운데 하나는 현재 위치에서 목적지 또는 목표물까지 가는 경로를 찾는 것이다. 예를 들어 전략 게임에서 목적지를 마우스로 알려주면 가장 빠른 지름길을 찾아 해당하는 장소로 이동해야 하는 문제로 전략 게임뿐만 아니라 거의 모든 게임에서 등장하는 문제이다. 이를 위해 가장 널리 사용되는 방법은 에이스타(A*) 알고리즘

<그림2>을 이용하는 방법이다.

이 방법은 예상비용(Estimated cost)을 이용해 경로에 대한 탐색범위를 효율적으로 제한하는 방식이다. 맵(map)의 특성에 따라 다양하게 휴리스틱(heuristic) 가중치를 적용시킬 수 있는 장점이 있어 길 찾기에서 다양하게 응용되고 있으나, 목표물까지 가는 도중에 길이 차단됐거나 폭파돼 끊겼을 경우처럼 지형이 일시적으로 바뀌었을 때는 에이스타 알고리즘만으로 모든 길찾기 문제를 해결할 수 없으며 추가적인 방법이 요구된다. 길찾기 문제는 인공지능 문제 중에서 비교적 많은 연구가 진행된 분야이며, 대부분 에이스타 알고리즘과 다른 방법을 함께 이용해 해결한다.

방대한 지형에서의 길찾기일 경우에는 출발지점과 도착지점 사이의 거리가 멀어질 수 있다. 이런 경우에는 탐색해야 할 공간이 폭발적으로 증가하게 되므로 메모리의 낭비 등으로 탐색 효율이 떨어지게 된다. 따라서 출발지점과 도착지점 사이의 중간 경유지를 생성해 길찾기를 단계적으로 수행하는 계층적 길찾기 방법을 이용해 해결할 수도 있다.

플로킹(Flocking)

플로킹이란 새나 벌, 어류 등과 같은 수많은 개체들이 무리를 지어 집단적으로 움직이는 모습을 흉내내어 묘사하는 방법을 일컫는다. 플로킹의 대상이 되는 새나 벌, 물고기 등과 같은 개체를 보이드(boid)라고 부른다. 일반적으로 보이드들의 집단 행동을 자연스럽게 묘사하기 위해서는 다섯 가지의 원칙을 적절하게 적용해야 하는데, 이러한 원칙들에는 다음과 같은 것들이 있다.

분리(seperation) | 보이드가 주변의 다른 보이드들과 적당한 거리를 유지하도록 해 보이드 간의 충돌이 일어나지 않게 하는 것이 목적이다. 무리의 각 보이드마다 주변 보이드들과의 거리를 판단하고 적당한 거리를 유지할 수 있도록 조정한다.

정렬(alignment) | 보이드가 주변의 다른 개체들과 속도 벡터(속력과 방향)가 동일하도록 유지시켜서 전체 무리의 움직임이 일관될 수 있도록 하는 것이 목적이다. 보이드의 방향과 속력을 주변 보이드의 평균적인 방향과 속력에 맞추어 해결한다.

응집(cohesion) | 보이드들이 분산되지 않고 하나의 무리로서 모일 수 있게 하는 역할을 한다. 주변 보이드들과의 평균

위치(분포에 있어서의 무게 중심)로 움직일 수 있도록 방향 및 속력을 조정해 준다.

회피(avoidance) | 보이드가 장애물이나 다른 보이드에 부딪히지 않도록 하기 위해 일정 거리를 내다보고 장애물의 존재 유무를 파악해 그것을 피할 수 있도록 방향을 조정함으로써 구현할 수 있다. 장애물뿐만 아니라 적들에 해당하는 개체들에 대해서도 적용될 수 있다.

이동(migration) | 미리 정의해놓은 목적지를 향해 이동할 수 있도록 하는 원칙이다. 물론 보이드 무리가 어디로 가야하는지에 대해 전혀 알지 못하고, 장애물이나 적들을 회피하며 다른 보이드들과 보조를 맞추어 임기응변적으로 이동할 수도 있다.

이와 같은 원칙들을 적절하게 적용함으로써 플로킹 기법은 개체 유닛들의 집단 이동을 자연스럽게 구현할 수 있고, 게임을 하는 사용자에게 보다 현실감있는 환경을 제공하는 수단이 될 수 있다.

그 외의 기법들

팀 인공지능 | 최근 게임은 온라인 기능이 강조돼 그 결과로 참여하는 인원간의 팀워크를 중요시하게 됐다. 이에 기반해 인공지능 기술에도 팀 인공지능이 중요한 이슈로 등장했다. 1명의 지휘자와 다수의 팀원으로 이루어진 집단의 인공지능을 처리하는 방법으로 팀 배치, 지형정보를 이용한 전략적 이동, 개개인의 역할 분배 등이 중요한 요소로 이를 위해 3가지 레벨로 나누어 처리할 수 있다. 우선 전략 레벨에서는 팀이 달성해야 할 목표에 의해 통제되고, 팀 레벨에서는 각 목표를 달성하기 위한 계획에 의해 통제되며, 개인 레벨에서는 각 개개인의 행동 규칙에 의해 통제된다.

LOD AI | LOD AI(Level Of Detail AI)란 현재 스크린에 보이는 캐릭터의 인공지능은 구체적인 알고리즘을 적용하며, 보이지 않는 캐릭터의 인공지능은 보다 단순한 알고리즘을 사용하는 것을 말한다.

인공생명 | 살아있는 생명체의 행동이나 행위를 흉내내어 캐릭터 등에 적용하는 기술을 말한다.

인공지능 기법을 사용한 게임들

게임에서의 인공지능을 구현하기 위해서는 한두 가지 방법만을 사용하기보다는 게임의 수준이나 장르에 따라 몇 가지 방법을 조합해 구현하는 것이 일반적이다. 블랙 & 화이트(Black & White)에서는 인공생명이라는 기술이 적용됐는데, 캐릭터 자신의 욕구와 상태에 따라 사용자 행동에 대한 반응이 다이내믹하게 나타날 수 있도록 했다. 또한 사용자가 캐릭터에게 어떤 행동을 반복해서 보여주게 되면 그 행동을 똑같이 따라 하고 상이나 벌을 제시함으로써 선과 악을 구분하고, 교육이 가능하도록 했다. EA사의 심시티 시리즈로 유명한 심즈(The Sims)는 윌라이트(Will Write)가 개발한 것으로 일상생활을 게임으로 옮겨놓은 것으로, 각 캐릭터의 욕구가 있어서 그것들을 잘 충족시켜줘야 순조롭게 플레이를 할 수 있다. 심즈에서의 가장 특징적인 인공지능 요소는 캐릭터의 욕구에 기반한 인공지능 기법이 사용된 것과 어떤 행동에 대한 정보를 각각의 오브젝트가 갖고 있다는 것이다. 또한 인공생명 기법과 퍼지 기법을 조합해 적용한 게임으로도 알려져 있다. 하프라이프에서는 몬스터나 적들의 일련의 행동을 스케줄로 묶어 처리함으로써 플레이를 할 때면 정말 만만한 상대가 아니라는 느낌을 받을 수 있도록 처리했다. 또한 플로킹 방법을 적용해 무리의 움직임을 자연스럽게 구현한 게임이기도 하다.



〈그림2〉 인공생명 기법이 적용된 심즈



〈그림3〉 인공지능 기능이 향상된 언리얼 토너먼트

이상과 같이 게임에서의 인공지능은 점점 더 발전하고 있고, 그 필요성도 중요시되고 있다. 이전에는 부수적인 요소로 인식됐으나 하드웨어 및 그래픽 기술 등이 발전하면서 인공지능의 가능성과 중요성이 점차 커져가고 있다. 