레이싱 게임에서의 물리엔진과 네비게이션 메쉬

송승호*, 김혜영*
*홍익대학교 게임학부 게임소프트웨어전공
e-mail : illusion0926@naver.com
hykim@hongik.ac.kr

Physics Engine with the Navigation Mesh from Racing Game

Seung-Ho Song*
Hye-Young Kim*
*Dept. of Game Software, Hon-Ik University

요 약

레이싱 게임은 자동차의 물리 지형과의 마찰 그리고 오브젝트간의 충돌을 구성하여 시뮬레이션 되어야 한다. 그러나 NPC 의 이동 시에 NPC 도 PC 와 같이 일반적인 자동차 물리를 사용한다면 많은 부하가 걸리게 된다. 따라서 본 논문에서는 부하를 줄이고 NPC 의 이동을 구성하며 인공지능을 포함하여 스스로 길을 찾아가는 효과를 연출하기 위해 물리엔진과 네비게이션 메쉬를 사용하였다.

1. 서론

최근 개발되는 게임, 시뮬레이션에서는 많은 수의 NPC 가 등장한다. 그 중 차량 NPC 는 레이싱 뿐 아니라 도시 시뮬레이션 등에서도 많이 사용되고 있다. 이러한 시뮬레이션상의 차량 NPC 는 몇몇 곳의 시작 지점과 중간지점, 도착지점을 설정하여 정해진 위치를 순환하거나 NPC 를 생성하고 도착지점에 도착한 후에 시작 지점으로 위치를 바로 수정하거나 NPC 를 삭제한 후 새로 생성하는 방법을 사용하고 있다.[1]

이러한 방법들은 가상세계를 구축함에 있어서 플레이어 눈앞에서 차량 NPC 가 사라지거나 플레이어가 갈 수 없는 곳으로 들어가 보이지 않는 곳에서 사라지는 등 플레이어가 시뮬레이션의 가상세계의 어색함을 느끼게 된다.

구현 시에 이러한 순환의 어색함을 줄이기 위하여 네비게이션 메쉬와 A* 알고리즘을 사용하여 무작위의 위치로 도착지점을 설정하고 도착 후에 다시 무작위로 도착지점을 설정하여 순환 경로와 다른 경로로 차량 NPC 가 움직이는 것을 설명한다.

상용 엔진인 Bullet Physic 엔진과 공간 표현 기법의 대표적인 네비게이션 메쉬를 바탕으로 하여 차량 NPC 의 경로 탐색과 플레이어와의 상호 작용을 적용하였다.

본 논문의 구성은 다음과 같다 2 절은 네비게이션 메쉬를 바탕으로 기본적인 경로탐색을 구현하여 기존 순환 경로와의 차이점을 보인다. 3 절은 Bullet Physics 의 NPC 데이터에 네비게이션 메쉬를 사용여 탐색한 경로를 설정하여 NPC 차량의 이동부분을 기술하였다. 4 절에는 결과화면 보인다. 5 절에서는 결론과 추후과제를 기술한다.

2. 네비게이션 메쉬 경로탐색

네비게이션 메쉬는 기본적으로 Poly 를 List 로 연결하여 연결된 리스트 내에서의 경로를 탐색하게 된다. 이는 3D 환경에서도 2D 환경과 같은 경로탐색 알고리즘을 사용할 수 있게 한다. 네비게이션 메쉬는 3D 메쉬를 사용하여 경로 탐색을 진행하기 때문에 여러 제약 조건을 설정할 수 있다. 기본적으로 사용 가능한 제약이 정점의 색을 이용하여 현재 지점에서의 이동 불가능 지역 설정이다.[2]

네비게이션 메쉬는 이미 지정된 경로를 따라가는 경로 설정이 아닌 기본 구성된 영역 내에서 2D 경로 탐색 기법을 사용하여 실시간으로 목표지점을 설정하고 해당 경로를 탐색하여 경로를 설정할 수 있는 것이 장점이다.[3]

표 1 네비게이션 메쉬 경로탐색과 순환경로의 장 단점

	장점	단점
순환경로	• 연산의 양이 적음	• 경로 재설정이 불가능 • 여러 경로를 직접 지정해야 함
네비게이션메쉬	• 경로를 실시간으로 탐색하여 재설정 가능 • 3D 환경에서 2D 환경의 경로 탐색 알고리즘 사용	• 연산의 양이 많음 • 제약여부가 없을 시 네비게이션 메쉬내 모든 경로 이동가능

3. Bullet Physics 설정

네비게이션 메쉬의 경로탐색이 끝나고 목표지점으로의 이동을 위해 차량 NPC 의 위치 값을 수정한다. 위치 값의 수정은 월드행렬을 수정하는 것인데 물리엔진에서는 월드행렬의 수정으로 위치를 실시간으로 이동 시킬 시에는 다른 오브젝트와의 상호관계 적용이 되지 않는다. 실시간으로 위치 값을 강제 수정하는 것이기 때문에 물리 연산이 반영된 후에 다시 강제 수정된 값으로 위치 되기 때문이다.

이 점을 수정하기 위하여 차량 NPC 는 충돌형태를 구성하는 것 외의 설정을 해주어야 한다.

NPC 의 형태를 Rigidbody 외에 CharacterContloller 를 이용하여 물리엔진의 독립적 물리 가상세계에 추가한다. Rigidbody 만을 사용할 시에는 오브젝트의 속도 등을 변화 시켜 충돌을 반영하기 때문에 위치 이동을 하는 네비게이션 매쉬를 사용할 시 요구하는 위치까지 이동할 수 없기 때문이다.

```
m_Parent->ResolveMotionOnMesh(m_Position,
m_CurrentCell, NextPosition, &NextCell);

if (NextPosition.x - m_Position.x > 0)
    walkDirection += strafeDir;
else
    walkDirection -= strafeDir;

if (NextPosition.z - m_Position.z > 0)
    walkDirection += forwardDir;
else
     walkDirection -= forwardDir;

m_character>
setWalkDirection(walkDirection*walkSpeed);
```

4. 구현 결과

네비게이션 메쉬를 바탕으로 경로를 탐색하고 차량 NPC 가 이동할 수 있었고 이동경로는 Bullet Physics 에 적용 되어 다른 오브젝트와 상호작용이 가능하였다.

4.1 이동경로 탐색 결과

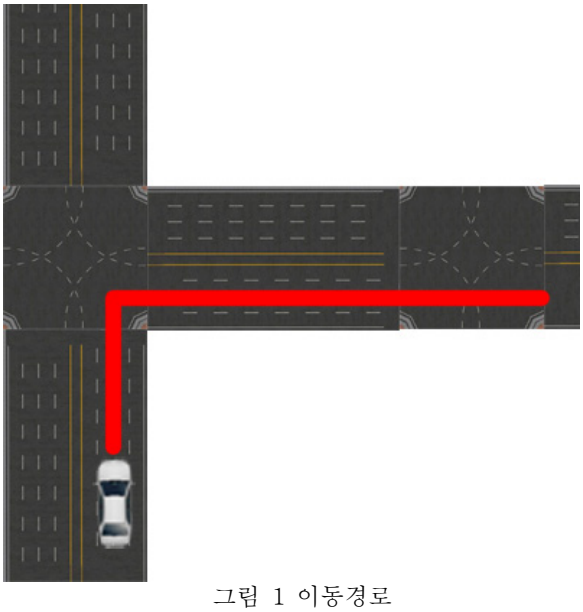
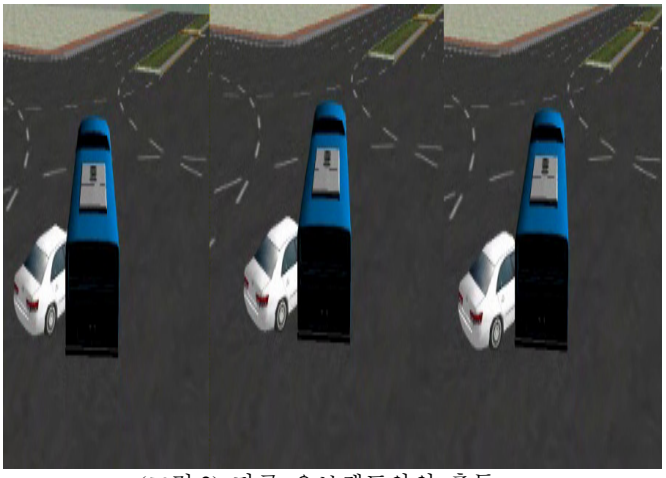


그림 1 이동경로

네비게이션 메쉬를 사용하여 목표지점을 임의로도 설정이 가능하고 직접적으로 설정하여 경로를 만드는 것도 가능하였다. 정점의 색을 사용하여 Poly 의 색을 구별하여 다른 차선으로 이동을 금지하게 설정하였다.

4.2 충돌설정 결과



(그림 2) 다른 오브젝트와의 충돌

Bullet Physics 를 사용하여 NPC 의 충돌객체를 설정하고 플레이어와의 충돌이 가능하였고 충돌 이후에도 차량 NPC 는 경로를 재 탐색하여 목표지점에 무사히 도착할 수 있었다. 이는 기전 시뮬레이션에서 충돌 이후 삭제되거나 위치가 재설정되는 NPC 와의 차이를 보였다.

5. 결론 및 향후 연구방향

네비게이션 메쉬와 Bullet Physic 를 사용하여 레이싱 게임을 구현하였다. NPC 차량이 실시간으로 경로탐색, 목표 재설정을 반복하며 자연스러운 도로 환경을 구현하였다. 차량 NPC 는 충돌 이후에도 다시 목표지점을 향한 경로탐색을 실행하여 이동할 수 있었다..

정점좌표 색의 검사로 반대차선 침입을 막았으나 같은 차선에서 역 주행을 하였고 방향성 지표에 따라 경로탐색 알고리즘을 수정할 필요가 있어 차선에 맞는 방향에 따라 이동 할 수 있게 구현할 계획이다.

참고문헌

[1] 이 승주, “인접한 블록의 움직임 벡터의 방향성을 고속 움직임 추정 알고리즘”, 혜전대학, vol. 23, 2005

[2] “3D 임에서의 이동 장애물을 고려한 동적 경로 탐색기법”,, 한국게임학회 논문지, 제 6 권, 제 3 호, 2006 년 9 월

[3] “내접원을 이용한 3D 게임에서의 이동 경로 곡선화”, 한국 정보과학회 학술지, 논문지, 2003 년 추계학술발표