

3차원게임을 위한 물리엔진에서의 관절체 구조 연구

허원*, 손민우, 신동일, 신동규
세종대학교 대학원 컴퓨터공학과 분산멀티미디어 연구실

e-mail:{heowon, minwoo15, dshin, shindk}@gce.sejong.ac.kr

A Research on Joint Types in Physics Engines for 3D Games

Won Heo*, Minwoo Son, Dongil Shin, Dongkyoo Shin
Dept. of Computer Engineering, Sejong University

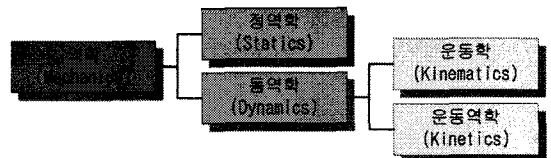
요 약

초기의 3D 게임 엔진들은 3차원 이미지 렌더링에 초점을 맞춰 개발이 되었기 때문에 현실세계에서와 마찬가지로 물체들 간에 발생하는 선형운동 및 회전운동 등의 고전 물리학의 운동법칙과 운동에 의해 발생하는 물리적 현상에 대한 고려가 미비하였다. 3차원 게임 개발자들에게 게임을 구성하는 객체 및 요소를 현실적으로 인식하기 위한 물리학의 중요성이 부각되었고, 이후에 많은 개발자 및 개발회사들이 현실 세계의 물리현상을 게임에 적용시키고 있다. 이에 본 논문에서는 3차원 게임을 위한 물리엔진에서 물체들의 사실적인 표현과 움직임에 도움이 되는 관절체의 종류와 구조에 대해 연구하였다.

1. 서론

초창기의 3차원 게임 엔진들은 3차원 이미지 렌더링에 초점을 맞춰 개발이 되었기 때문에 현실세계에서와 마찬가지로 물체들 간에 혹은 물체 자체에서 발생하는 선형운동 및 회전운동 등의 고전 물리학의 운동법칙과 운동에 의해 발생하는 물리적 현상에 대한 고려가 미비하였다. 90년 대 중반 이후에 3차원 게임 콘텐츠를 위한 현실성 있는 물리 효과의 표현이 게임 개발자들에 의해 요구되기 시작했고, 이후에 많은 개발자들 및 개발 회사들로 하여금 현실 세계의 물리현상을 게임에 적용시키는 촉매제가 되었으며 3차원 게임엔진 개발에서 물리현상을 처리할 수 있는 기능을 고려하게 되었다. 이에 본 논문에서는 3차원 게임에서 물체들의 사실적인 표현과 움직임에 도움이 되는 관절체의 종류와 구조에 대해 연구하였다.

3차원 게임엔진에 사용되는 물리학은 고전역학에 속해있는 뉴턴역학[1]에 대한 내용들을 주를 이룬다. 뉴턴역학은 뉴턴의 제 2운동법칙인 속도-가속도 운동법칙을 중심으로 하여 제 1운동법칙인 관성 운동의 법칙을 기반으로 하며 제 3운동법칙인 작용-반작용 법칙을 이용하여 운동에 대한 결과에 변위로 적용시키는 것으로 실세계에서 물체 간에 작용하는 힘과 운동의 관계를 통해 물리적 현상을 이해하는 것을 목적으로 한다. 뉴턴역학은 물체 간에 작용하는 힘과 운동의 상태에 따라 [그림 1]과 같이 정역학과 동역학으로 분류된다.



[그림 1] 힘과 운동에 따른 역학 분류

게임을 위한 물리학의 내용은 캐릭터나 물체들의 상태 및 행동을 시뮬레이션하기 위해 사용된다. 따라서 실제상황을 재현하는 시뮬레이션 장르의 게임

2. 관련연구

2.1 물리엔진

2.1.1 게임엔진에 적용되는 물리학

에서 사용되며 동역학의 이론들이 주로 사용된다

2.1.2 물리 엔진관련 기술 동향

게임엔진 기술에 물리학이 집중적으로 사용되는 장르는 레이싱, 비행, 당구, 그리고 골프와 같은 시뮬레이션 게임들이다. 게임엔진 기술에서 물리학이 사용되는 부분은 기본적으로는 캐릭터 혹은 물체들의 이동, 충돌반응, 발사체의 움직임을 시뮬레이션하는 부분이다. 특히 물체가 비행기, 배, 호버크래프트, 혹은 자동차와 같이 특화된 것이라면 자체적인 고유한 움직임의 특징들을 갖고 있으며 이런 특화된 물체들을 위해서는 고급화된 물리학의 이론들이 사용된다[2].

2.1.3 물리 엔진의 기술적 특징

물리엔진은 게임에 등장하는 물체에 대한 물리시뮬레이션을 위한 것이다. 즉 게임에 등장하는 캐릭터를 포함한 장비나 아이템들의 동작을 현실에서처럼 표현하기 위해 물리학의 이론을 사용하는 기술이기 때문에 다음과 같은 특징들이 있다.

- ◆ 물체의 외부 힘이나 움직임에 따른 동작을 처리하기 때문에 뉴턴의 운동법칙과 강체(rigid body)에 대한 동역학이론들을 주로 사용한다.
- ◆ 물리엔진기술은 컴퓨터그래픽스 기술과 밀접한 관련이 있다.
- ◆ 물리엔진기술은 게임인공지능기술과 밀접한 연관이 있다.
- ◆ 많은 수의 액션을 실시간으로 처리해야 하기 때문에 물체의 물리적 모델이 복잡한 경우, 최적화기법들을 사용된다.

2.2 Body

Body는 동역학 시뮬레이션[3]에서 운동체를 구성하는 요소이다. 이러한 운동체들은 Body와 Joint로 구성이 되며 1개의 Body에는 적어도 1개 이상의 Joint가 할당이 되며 Joint들의 형태에 따라 Body의 동작이 결정이 된다. Body들은 실제 물리적인 특성요소로서 질량과 위치좌표 및 각 운동의 연산에 필요한 값들을 갖는다. 이들은 물체를 구성함에 있어서 chain구조를 갖는다. 이러한 chain구조는 크게 open chain구조와 closed chain 구조로 나눌 수 있으며 강체 시뮬레이션 시스템은 closed chain구조를 통해 운동체를 구성한다.

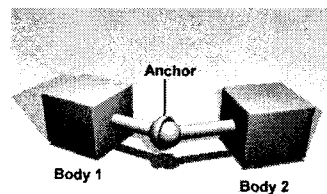
2.3 Joints and constraints

Joint는 앞서 언급한 Body와 함께 운동체를 구성하는 요소로서 이 Joint의 형태에 따라 Body들의 움직임이 제어된다. Joint들을 통해 Body가 제어되는 원리는 각 Joint들은 DOF(Degree Of Freedom)[4]이라는 자유도의 수치를 통해 움직일 수 있는 영역을 정의하며 이 영역 안에서 Body들이 동작하는 것이다. Joint들이 갖는 또 다른 의미는 처음 Joint를 통해 Body 및 또 다른 Joint의 존재 여부 및 위치를 추적할 수 있고 이를 통해 좀 더 유연한 동작을 얻을 수 있다. 이러한 Joint들의 움직임을 제어하고 실제 운동하는 물체에 속한 Joint들의 운동량을 계산하기 위해서는 equation들이 필요하며 이러한 equation들을 Constraint equation[3]이라고 한다. 또한 Joint들의 동작에서 발생하는 에러는 Body들의 동작을 부정확하게 만드는데 이러한 에러들을 보정할 수 있는 모듈로서 CFM(Constraint Force Mixing)과 ERP(Error Reduction Parameter)를 사용한다.

3. 물리엔진에서의 Joint 구현

앞서 언급하였지만 Joint들의 형태에 따라 Body들의 동작이 결정되기 때문에 강체 시뮬레이션 시스템에서는 다양한 형태의 Joint들에 대한 연구가 이루어지고 있다. 본 논문에서는 6가지 형태의 Joint들을 연구하였으며 이에 대한 내용은 다음과 같다.

3.1 Ball Joint



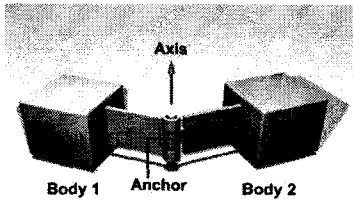
[그림 2] Ball Joint

이것은 한 body의 ball 부분과 다른 body의 socket 부분이 한 위치에 존재하며 이루는 Joint이다. 다음은 Ball Joint를 구성하는 mxJointBall class의 변수들을 나타낸 표이다.

Variable	Description
anchor1	첫번째 Body의 고정점
anchor2	두번째 Body의 고정점

[표 1] mxJointBall class variable table

3.2 Hinge Joint



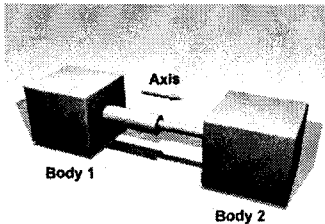
[그림 3] Hinge Joint

이것은 두 Body에 연결되어 있는 각각의 hinge 들이 하나의 고정장치에 같은 축을 이루며 연결되어 있는 Joint이다. 다음은 Hinge joint를 구성하는 mxJointHinge class의 변수들을 나타낸 표이다.

Variable	Description
anchor1	첫 번째 Body의 고정점
anchor2	두 번째 Body의 고정점
axis1	첫 번째 Body의 축
axis2	두 번째 Body의 축
qrel	Body1에서 Body2로의 회전에 대한 초기화
limot	영역과 운동에 대한 정보

[표 2] mxJointHinge class variable table

3.3 Slider Joint



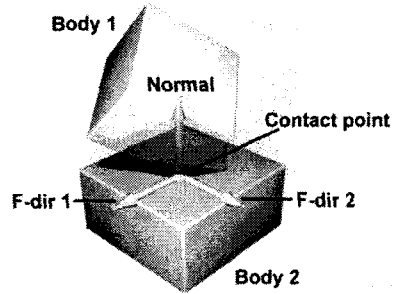
[그림 4] Slider Joint

이것은 piston과 socket이 같은 축을 이루며 축 방향으로만 움직일 수 있도록 연결되어있는 Slider Joint이다. 이것을 사용하면 두 Body들이 같은 방향을 갖도록 제한된다. 다음은 Slider joint를 구성하는 mxJointSlider class의 변수들을 나타낸 표이다.

Variable	Description
axis1	첫 번째 Body에 대한 축
qrel	Body1에서 Body2에 대한 회전행렬
offset	Body1에 연결된 Body의 지점
limot	운동/영역 정보

[표 3] mxJointSlider class variable table

3.4 Contact Joint



[그림 5] Contact Joint

Contact Joint는 두 Body들이 하나의 contact point에 일시적으로 연결되어 있는 Joint이다. 이것은 두 Body들에게 속성값을 적용시킴으로써 contact point에서 서로 관통하는 것을 막아준다. 이 Joint는 한 사이클을 주기로 생성, 소멸된다. 즉 이 Joint에 충돌현상이 일어나면 충돌에 대한 반응을 연산해 준 후 Joint가 소멸된다. 다음은 Contact Joint를 구성하는 mxJointBall class와 body들의 속성을 나타내는 mSurfaceParameter, mContactGeom class의 변수들을 나타낸 표이다.

Variable	Description
surface	동작을 구성하는 파라미터들
geom	mContactGeom의 인스턴스
fdir1	최초의 마찰방향

[표 4] mContact class variable table

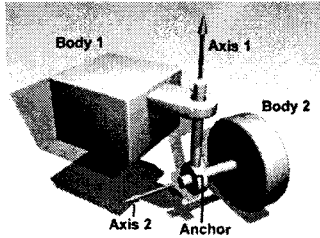
Variable	Description
mode	Contact 플래그를 설정
mu	마찰방향을 설정하는 값
bounce	접촉면에서의 두 물체 간 떨어진 값
bounce_vel	접촉면에서의 속도
soft_erp	접촉 normal 벡터에서의 ERP값
soft_cfm	접촉 normal 벡터에서의 CFM값
motion1	마찰방향1에 대한 표면속도
motion2	마찰방향2에 대한 표면속도
slip1,slip2	마찰방향에서의 미끄러짐

[표 5] mxJointHinge class variable table

Variable	Description
pos	접촉 위치
normal	단위 벡터
depth	접촉 깊이
g1,g2	Body1,2의 기하정보

[표 6] mContact class variable table

3.5 Hinge2 Joint



[그림 6] Hinge2 Joint

이것은 두개의 hinge들이 서로 다른 축을 가지고 연속해서 연결되어 있는 Joint이다. 위 그림에서 한 축은 방향을, 다른 하나는 회전을 담당하는 것처럼 두개의 축이 서로 다른 운동값을 가질 수 있다. 다음은 Hinge2 Joint를 구성하는 mxJointHinge2 class의 변수들을 나타낸 표이다.

Variable	Description
anchor1	첫 번째 Body의 고정점
anchor2	두 번째 Body의 고정점
axis1	첫 번째 Body의 축
axis2	두 번째 Body의 축
c0	축1과 축2에 대한 코사인 값
s0	축1과 축2에 대한 사인 값
v1	첫 번째 Body에 대한 각속도
v2	두 번째 Body에 대한 각속도
limot1	축1에 대한 운동과 영역
limot2	축2에 대한 운동과 영역
susp_erp	정지 파라미터 ERP
susp_cfm	정지 파라미터 CFM

[표 7] mxJointHinge2 class variable table

3.6 Fixed Joint

이것은 두 Body들이나 Body와 정적인 환경에 대한 고정된 상대위치와 방향을 유지시켜준다. 이 Joint는 실제 사용하기에는 적합하지 않고 디버깅용으로만 사용된다. 다음은 Fixed joint를 구성하는 mxJointFixed class의 변수들을 나타낸 표이다.

Variable	Description
offset	Body들 사이의 관련된 offset

[표8] mxJointFixed class variable table

3.7 JointGroup

이것은 운동체를 구성하는 Joint들의 그룹을 저장하는 클래스이다. 이것은 Joint들을 관리하는 매니저의 역할을 한다. 다음은 JointGroup을 구성하는

mxJointGroup class의 변수들을 나타낸 표이다.

Variable	Description
num	스택에 저장된 Joint의 수
stack	Joint들이 저장된 스택

[표 9]mxJointGroup class variable table

4. 결론 및 향후 전망

3차원 게임은 기존의 2차원의 게임 콘텐츠와는 달리 표현에 필요한 많은 데이터와 이로 인한 양질의 콘텐츠를 게임에 제공하고 있는 것이다. 이제 이러한 콘텐츠의 내용은 3차원 게임을 단순한 시뮬레이션의 차원에서 한 단계 높은 차원의 요소들을 적용하는 기술로 발전하고 있다. 특히 이러한 요소 중에서 물리적인 환경 요소는 게임 사용자들이 직접 체험으로 느낄 수 있는 것이기 때문에 이를 위한 물리 엔진 및 관련 기술의 개발은 가속될 것이다.

현재 이 논문에서 연구된 관절체 구조들을 사용하여 물리엔진 시스템을 구현 중에 있으며 이번 연구되지 않는 관절체들에 대한 연구도 계속 진행될 예정이다.

참고문헌

- [1] David M. Bourg, "Physics for Game Developers", p.1, O'Reilly, 2001
- [2] Multi Media eXtension, Intel, 1995, <http://www.intel.com>
- [3] Mark W. Spong, M. Vidyasagar, "Robot Dynamics and Control", p.5, Wiley, 1989
- [4] Eric Lengyel, "Mathematics for 3d game programming and computer graphics", p.295, Charles River Media, 2001