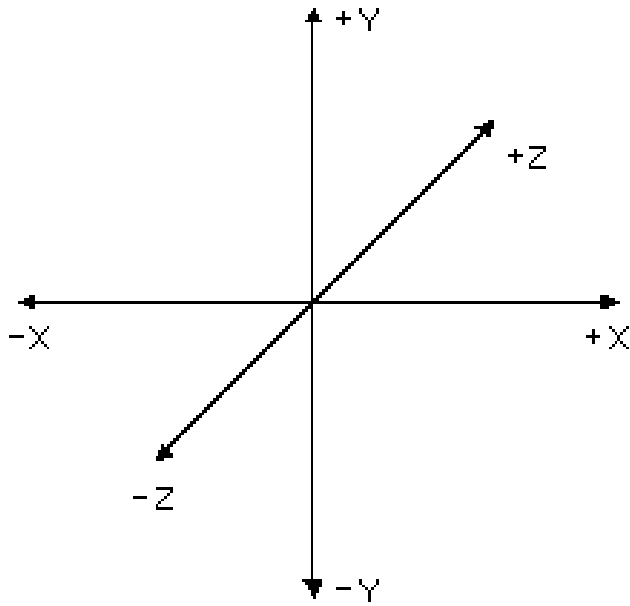


Particle Dynamics

[3D Coordinate System]

- 3차원 좌표계

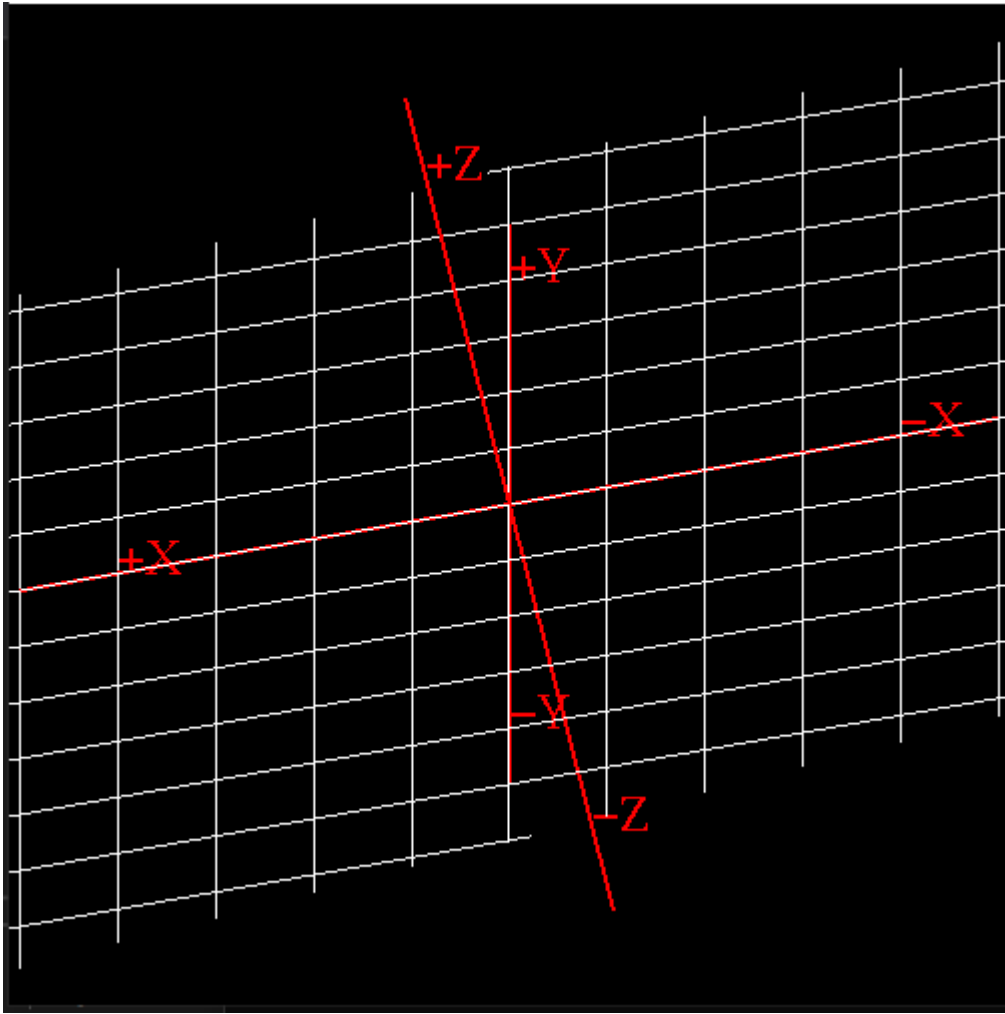


- > x축 : 오른쪽이 양수
- > y축 : 위쪽이 양수
- > z축 : 사용자에게 가까운 쪽이 음수
- > 모든 x,y,z축의 범위는 (-1,1)이다.
 - >> 주어진 범위는 (-1,1)이지만
 - >> 투영의 효과를 다르게 해야한다

Particle Dynamics

[3D Coordinate System]

- 3차원 좌표계



[문제점]

- Spring들과 좌표가 달라서 공간의 역할 X
- 좌표계 시점(카메라) 이동이 제한적
> (Y축,Z축 회전만 가능) - 키보드로
- 입체라는 느낌이 거의 없음

[해야 할 것]

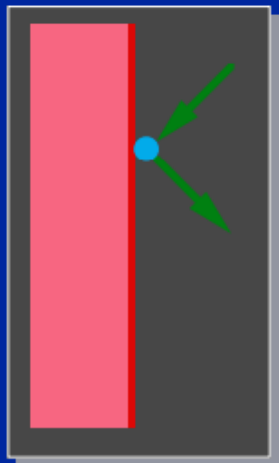
- Particle들과 좌표 동일
- 모든 축에 대한 회전, 이동, 확대 및 축소 구현
(키보드 or 마우스)
- X-Y 평면과 다른 공간들의 색 구분 해보기

[코드]

https://github.com/Winteradio/_OPENGL_make_3D_Coordinate_System

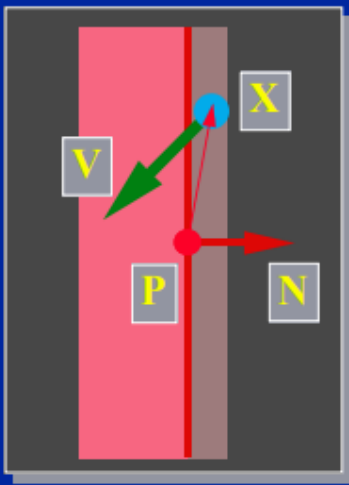
Particle Dynamics

[Particle Dynamics] – Collision and Contact



- Later: rigid body collision and contact.
- For now, just simple point-plane collisions.
- Add-ons for a particle simulator.

- > 반응의 결과
 - >> 충돌
 - >> 접촉



$$(X - P) \cdot N < \epsilon$$
$$N \cdot V < 0$$

- Within ϵ of the wall.
- Heading in.

- 충돌과 접촉의 구현방식

[기존]

- > 정해진 구역의 모든 면에 대하여 접촉 및 충돌 여부 판단
- > 속도가 엄청 빠를 시, 점들의 위치를 재조정

[변경]

- > 바닥면에 대해서만 접촉 및 충돌 판단
- > 속도가 엄청 빠를 시, 시간 간격을 조정
 - > 점이 접촉 및 충돌 판단 구역 내 들어올 수 있게
 - > Timer함수를 재귀함수로 사용하여 생각하기

- [코드] – h,cpp 및 클래스 구분만 제대로 구현

https://github.com/Winteradio/_OPENGL_make_3D_Spring