선형대수학 - 중간

□ 선형대수학이란

○ 선형대수학 이론

* 선형대수학 : 선형으로 이루어진 대(대신할 대) 수학
  - 일대일 대응이 된다.
  - 비선형을 선형으로 만드는 작업
    -> 아주 짧게 잘라 선으로 만듦 :: 미분(순간의 기울기)
* 좌표계 : 기준을 가지고 그 기준에서 어느 축으로 얼마나 떨어져있는지 측정
* 행렬 : 선형방정식을 쉽게 풀기위해 사용한 도구
* 스칼라
  - 크기만 가지는 값
  - 벡터와는 다르게 기준이 없어도 됨
* 벡터 : 크기와 방향을 가지는 양
* 위치벡터 : 원점을 시점으로 하고 점 P를 종점으로 하는 벡터를 점 P의 위치벡터(Position vector)라고 함
* 차원 : 한 성분을 다른 성분으로 대체할 수 없다. (독립)

□ 벡터

○ 벡터

* 어느 시점에서 보느냐에 따라 -인지 +인지 관점에서 봄
* 시작 위치와 상대 위치가 반드시 존재해야 함
* 음수 양수를 없애기 위해 제곱을 하고 루트를 씌워줌
  :: sqrt(sqr(Obj.x - x) + sqr(Obj.y - y)); // 이 방법을 이용하면 연산량이 많기에 thrust를 이용(가속화)
* 방향이 있기 때문에 기준점이 필요함 (기준에 따라 +, -로 기준을 나눔)
* 튜플 : 성분의 쌍(여러 가지 요소들의 한 묶음)

* 벡터의 덧셈
  - a = S-b
  - a + b = S

* 벡터의 뺄셈
  - 벡터는 방향성을 가지기 때문에 거리는 동일하며, 부호만 바뀐다.

* 벡터의 크기
  - k == 1 : 기존 크기
  - 0 < k < 1 : 축소
  - 1 < k : 확대

* 벡터의 노움과 단위벡터
  - 물체가 빛을 받을 때 방향에 따라 빛을 받는 양이 다름
    // 노말 벡터와 빛의 방향의 각도 = 램버트 앵글러

* 단위벡터
  - U = X / |X| 는 X 방향의 단위 벡터 = 정규화(Normalize)

* 벡터 법칙
  - 교환 법칙 : X+Y = Y+X
  - (X+Y)+Z = X+(Y+Z)
  - X+0 = 0+X
  - X+(-X) = X
  - k(m)X = (km)X
  - 분배 법칙 : k(X+Y) = kX+kY
  - (k+m)X = kX+mX
  - 1X = X
  - ||X+Y|| = ||Y+X||
  - X+Y = Y+X
  - ||P|| >= 0
  - P < 0,0,0, - - - , 0

○ 내적과 외적

* 내적 : 충동을 감지하기 위해 사용
* 외적 :

□ 좌표계

○ 좌표계

* 1차원 좌표계(직선)
* 2차원 좌표계(면)
  - 그림자 좌표계(평면 사교 좌표계)

  - 회전 좌표계(극 좌표계)

* 3차원 좌표계(도형)
  - 구면 좌표계(환경 매핑)

□ 변환

* 서로 기준점이 다른 좌표계에서 서로 기준점을 맞추는 것

○ 선형 변환

* 선형 변환 : 선형의 관점에서 변환

- 일대일 대응이 되어야함 :: 하나의 정의역에 하나의 치역

- 회전 행렬은 가역 행렬(역행렬이 존재)

- 직교 좌표계는 텍스쳐에 많이 쓰인다 :: 텍스쳐는 2D이기 때문

  -> 붙이는 방법을 맵핑이라고 함

○ 회전 변환

* 회전 변환

- Polar 좌표계 : 회전 좌표계

□ 연산자

○ 반전 연산자

* 거울 효과

○ 사영 연산자

* 그림자 투영

○ 회전 연산자 (오일러 회전)

* 회전에서 사용

○ 확대와 축소 연산자

* scale

○ 동차 변환

* 회전, 전단, 반전, 확대, 축소, 이동, 원근, 크기를 행렬의 곱으로 한 번에 표현한다.

○ 선형 변환의 합성 (함수의 합성)

* 함수와 함수를 합성한다.

○ 복소수

* 복소수 : 실수 + 허수 (z = a + bi)

○ 사원수 (쿼터니언)

* 복소수를 확장한 것 :: 실수와 허수를 한 공간에 두기 위함
* 사원수(쿼터니언) : 복소수를 3차원으로 끌어올린 것
  - 회전 및 곡면을 보간 할 때 사용
  - 점을 허수 공간, 길이를 실수 공간에 둔다.
  - 일반 회전 변환 시 짐벌락(gimbal lock) 현상을 방지하고, 연산속도를 올리기 위해 사용 => 회전행렬이 0,0,0이 됨
  - 회전을 예로 들면 x축, y축, z축 회전이 아닌 w축 회전을 새로 만들어서 회전을 시킴

* = (S, i, j, k) // 실수 부분 : 스칼라(S), 허수 부분 : 벡터(i, j, k)
  = Sq + qx i + qy j + qz k // 크기를 나타내는 qx, qy, qz가 있음
  = Sq + Vq

  => 각 i j k는 수직을 이루고 있음
  // i*j = k, i*k = j, j*k = i

○ 기하 변환

* 한 차원을 올려서 변환을 함 (쿼터니언처럼 w축을 추가)

□ 충돌

* 내적 : 각도를 측정하는 도구
* 2D 외적 : 좌 / 우를 체크하는 데 사용 :: 수치
* 3D 외적 : 삼각 폴리곤의 방향을 계산 :: 법선벡터

○ 경사면에 세우기

* 해당 경사면에 접하는 점의 z값의 차이를 구하여 그 값만큼 z값을 이동시킨다.

○ 점과 점의 충돌

* 등식이 아닌 부등식을 사용하여 일정 범위 안에 들어오면 충돌로 본다.

○ 점과 선분의 충돌

* P1의 점 / P0->P2의 선분
* P0->P1를 잇는 벡터를 그릴 수 있다.
* V1 과 V를 외적한 값에서 sin시타가 0이 되면 V의 선 안에 점이 존재한다.

* 1) 3분할로 분다 => P2 < P0 , P0 <= P2 <= P1 , P2 > P1
* 2) P2의 충돌 범위 반지름을 구하고, 위치의 길이가 반지름 보다 작으면 충돌로 본다
* P0을 기준 v1의 각도가 +고, P1을 기준 v2의 cos각도가 -일 때 두 각도를 곱하여 -면 P0<?<P1 의 범위 안에 있다.

○ 선분과 선분의 충돌

* 점과 선분의 충돌에서 |V1| <= |V| 이어야 한다.

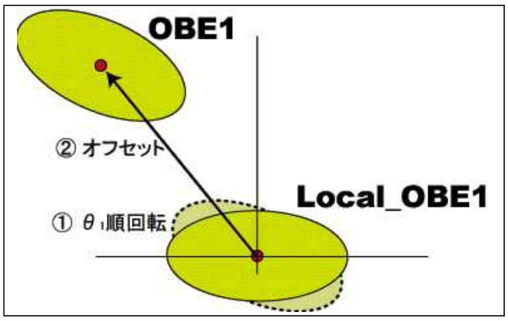
○ 점에서 평면까지의 거리

○ 다각형의 충돌

* 바운딩 박스를 이용하여 충돌을 시킨다.
* 바운딩 박스를 여러 개 만들어 세부적으로 충돌을 처리할 수 있다.

○ 타원과 타원의 충돌

* C1 타원을 0,0 점으로 이동변환
* C1 장축을 단축으로 줄임
* C1은 원이 되는데, 이 원의 반지름을 빼주어 점으로 만들어줌
* 변환된 C2의 원과 C1의 점을 충돌 체크



○

*