Transformations

그려질 대상을 매 프레임 변화시키는 좋은 방법은 매트릭스를 사용하는 것이고, 이번 챕터에서는 그에 필요한 수학적인 내용을 같이 볼 예정이다.

Vectors

보통 Direction과 Magnitude(a.k.a strength or length)를 가진다고 정의함.

보통 벡터는 방향을 나타내므로 원점이 다르더라도 방향과 크기가 같으면 같은 벡터로 판정.

방향 벡터와 위치 벡터의 구분이 어려운데, 위치 벡터의 경우 (0,0,0) 좌표에서 특정 위치를 가리키는 벡터를 의미함.

Scalar vector operations

스칼라 : 1차원 숫자를 말함. 스칼라와 벡터의 덧셈/뺄셈/곱셈/나눗셈은 스칼라를 벡터와 같은 차원의 벡터로 만들어 같은 위치의 원소와 연산.

Vector negation

벡터 부정은 각 요소에 빼기 기호를 추가한 것. 크기는 같고 방향은 반대가 됨. v \* -1과 같음.

Addition and subtraction

벡터의 덧셈과 뺄셈은 같은 위치의 원소의 값을 더하고 뺀 새로운 벡터를 만드는 것. 기하적 의미는 책에 소개된것과 같음.

뺄셈은 두번째 벡터의 부정과 첫째 벡터를 더하는것과 동일함.

Length

벡터의 길이는 피타고라스 정리를 통해서 구할수 있음.

유닛 벡터는 길이가 1인 벡터.

노말라이즈는 벡터의 길이로 벡터를 나눠서 유닛 벡터로 만드는 것.

Vector-vector multiplication

Dot product

결과물은 벡터가 아닌 스칼라로 나오기 때문에 스칼라 곱이라고도 부름.

v\*k = |v|\*|k|\*cos(theta)

이때 theta는 두 벡터사이의 각. 그리고 v와 k가 단위 벡터라면

v\*k = 1\*1\*cos(theta)

여기에 acos를 통해서 theta값을 구할수 있음.

cos(theta) 값이 0인지 1인지에 따라서 두 벡터가 평행(parallel)한지 직교(orthogonal)하는지 알수 있음.

Cross product

결과물이 벡터로 나오기 때문에 벡터곱 이라고도 부름.

오직 3차원에서만 정의되고, 두 벡터와 직교하는 벡터가 나옴.

iXj = |i|\*|j|\*sin(theta)\*k;

Matrices

매트릭스는 직사각형으로 숫자를 늘어놓은 것.

텍스트, 시계이(가) 표시된 사진

자동 생성된 설명이거는 2x3 행렬

(i,j) 와 같은 식으로 원소를 인덱싱 할 수 있음. 위 행렬에서 (2,1)은 4가 되는 식.

Addition and subtraction

행렬의 덧셈과 뺄셈은 벡터와 마찬가지로 같은 위치에 있는 원소를 더하고 빼면 됨. 결과는 행렬.

텍스트, 시계, 게이지이(가) 표시된 사진

자동 생성된 설명 : 덧셈

텍스트, 시계, 게이지, 장치이(가) 표시된 사진

자동 생성된 설명 : 뺄셈

Matrix-scalar products

행렬과 스칼라의 곱은 각 요소에 스칼라 값을 곱해주면 됨.



Matrix-matrix multiplication

행렬과 행렬의 곱셈은 다음 두가지 제한을 가짐.

1. 오직 왼쪽 행렬의 column의 개수와 오른쪽 행렬의 row의 개수가 동일한 경우에만 곱할수 있음.
2. 교환법칙이 성립하지 않음. A\*B != B\*A

 2x2행렬곱

Matrix-vector multiplication

N차원 벡터가 있다면, MxN행렬과 곱할수 있다. N차원 벡터를 Nx1 행렬로 생각하면 곱셈연산을 할수 있다.

Identity matrix

다른 벡터 혹은 행렬과 곱해져도 곱해진 대상에 변화가 없는 행렬.

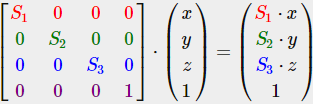
텍스트, 시계이(가) 표시된 사진

자동 생성된 설명

Scaling

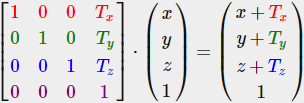
특정 좌표(x,y,z)기준으로 벡터의 크기를 늘이고 줄이는 방법.

단위 행렬의 값을 원하는 값으로 변경한 행렬로 벡터를 스케일링 할 수 있음.



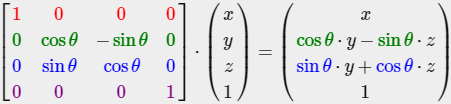
Translation

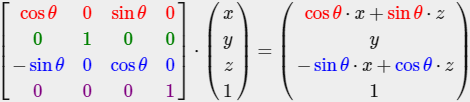
위치벡터의 각 원소에 특정한 값을 더해서 원하는 위치로 이동시키는 방법.

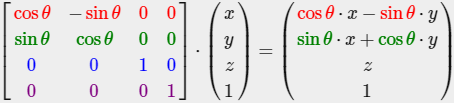


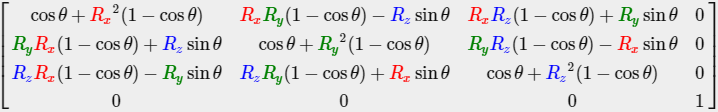
Rotation

회전 연산은 회전시키길 원하는 벡터와 회전시킬 각도, 회전의 축이되는 벡터 세가지가 필요하다.

 축이 x-axis 인경우.

 축이 y-axis 인 경우.

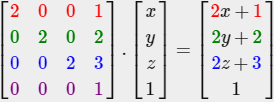
 축이 z-axis 인 경우.

 축이 (Rx, Ry, Rz)인 경우.

Combining matrices

서로 다른 특성을 가진 행렬을 곱해서 둘의 특성을 모두 가진 새로운 행렬을 만들수 있음.





In practice

GLM(openGL Mathematics)

OpenGL에서 수학을 위해서 사용하는 header only library