

변환 (Transformation) - 이동, 회전, 스케일

3차원 모델(Model) - 3차원 형상을 가상의 레이어로 표현한 것
삼각형 여러개를 2인 모양이 그 모양 같아 메쉬(Mesh)라고 한다

2차원의 이동(Translation)은 vec3을 사용하고 회전(Rotation)은

각각 174번 가지고 정의가 가능해서 float을 사용한다.

크기(Scale)은 각각 float인 scalar를 사용한다

scale이 0이면 화면에 아무것도 그려지지 않고 음수(-)가
되면 뒤집어서 그려지지 않는다. (두 번 뒤집으면 다시 앞면이나 그려진다)

이런 Transformation 계속 변하니 Update에서 구현된다.

이동(Translation) - vec3

$$\text{vertexBuffer} = \text{object.vertex} + \frac{\text{vec3(translation)}}{\text{이동하는 방향 벡터}}$$

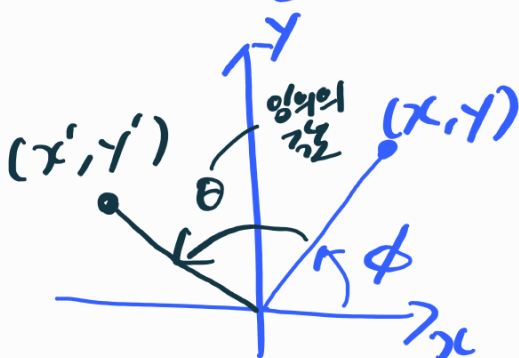
벡터와 스칼라의 식적연산은 벡터의 모든 값에 적용된다

$$\text{Vector} \pm \text{Scalar} = (x \pm \text{Scalar}, y \pm \text{Scalar}, z \pm \text{Scalar})$$

object의 vertex를 직접 그리지 않고 vertexBuffer에 참조해
그리는 이유는 원본 데이터를 보관하기 위해서이다

회전(Rotation) - 2차원에선 어떤 점에 대해 회전해볼지가 중요하다!
(3차원은 회전축)

어떤 점을 원점에 대해 회전시키는 경우



$$x = r \cdot \cos(\phi)$$
$$y = r \cdot \sin(\phi)$$

$$x' = r \cdot \cos(\phi + \theta)$$
$$y' = r \cdot \sin(\phi + \theta)$$

삼각함수의 덧셈정리

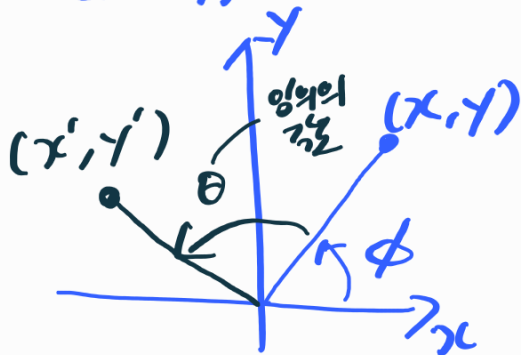
$$\cos(x+y) = \cos x \cos y - \sin x \sin y$$

$$\cos(x-y) = \cos x \cos y + \sin x \sin y$$

$$\sin(x+y) = \sin x \cos y + \cos x \sin y$$

$$\sin(x-y) = \sin x \cos y - \cos x \sin y$$

Tip! 나중에 x 각, y 각 회전할 때
각각의 x 값을 y 값을 고정하고
나머지 y, z or x, z 값을
사용하여 계산하면 된다.
계산식은 동일하다.



$$x = r \cdot \cos(\phi)$$

$$y = r \cdot \sin(\phi)$$

$$x' = r \cdot \cos(\phi + \theta)$$

$$y' = r \cdot \sin(\phi + \theta)$$

$$x' = r \cdot \cos(\phi + \theta)$$

$$= r \cdot \cos(\phi) \cos(\theta) - r \cdot \sin(\phi) \sin(\theta)$$

$$x' = x \cos(\theta) - y \sin(\theta)$$

$$y' = r \cdot \sin(\phi + \theta)$$

$$= r \cdot \sin(\phi) \cos(\theta) + r \cdot \cos(\phi) \sin(\theta)$$

$$y' = y \cos(\theta) + x \sin(\theta) = x \sin(\theta) + y \cos(\theta)$$

원점에 대해 회전하는 사각형을 보면 x', y' 를 구하는 수 있다
결국 3차원에서는 각도를 기준으로 회전을 짓고 같다

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \text{ 행렬로 표현하는 법}$$

스케일(scale, 크기)

$$\text{VertexBuffer} = \text{Object.vertex} \times \text{vec3}(\text{scale})$$

회전, 이동, 스케일을 모두 적용하려면 임시 벡터 vec3 temp 를 만들어
모든 계산을 다 한 후 모든 transformation을 계산해 결과를 vertexbuffer에 넣는다.
Transformation을 적용하는 순서가 중요하다. 보통은 Scale \rightarrow Rotation \rightarrow Translation 순으로
계산한다. 하지만 필요에 따라 계산 순서를 바꿀 수도 있다. (ex: 행성의 공전과 자전)