OAGLE 문서

**물리 엔진**

**기본 모델**

**(VAO)**

**외부 모델**

**GL**

**초기화**

**셰이더**

**OpenGL 코어**

**GLAD**

**씬/카메라**

**개체**

**GLFW**

**수학**

수학 모듈

**\* 수학 모듈은 다음을 포함하여 사용할 수 있습니다.**

#include “oaglem.h”

**\* 개요**

수학 모듈은 그래픽 개체 상태를 표현함에 있어 필요한 2~4차원 벡터, 행렬, 사원수를 정의합니다.

**\* 목차**

1. [벡터](#수학벡터)

2. [행렬](#수학행렬)

3. [사원수](#수학사원수)

4. [전역 요소](#수학전역)

**\* 본문**

1. 벡터

**벡터는 아래 클래스로 표현됩니다.**

nvec<D, T>

D는 unsigned int 타입이며 차원을 나타냅니다. T는 벡터 성분의 타입입니다.

T=2~4, D=int/unsigned/float/double에 따라 다음과 같이 미리 클래스들이 정의되어 있습니다. 일반적으로 위치/크기/방향 등을 나타냅니다.

ivec(2~4), uvec, vec, dvec

**\* 멤버 변수**

T entry[D]: 배열 형식으로 된 성분입니다.

T x, y, z, w | T r, g, b, a | T s, t, p, q : 각각 1~4번째 성분과 데이터 공간을 공유합니다. 4차원 미만인 벡터의 경우에도 사용 가능하지만 제공되는 함수에서는 사용되지 않습니다.

**\* 멤버 함수**

nvec()

영벡터를 생성합니다.

nvec(T a)

모든 성분이 a인 영벡터를 생성합니다.

nvec(T x, T y, T z=0, T w=0)

벡터의 첫 4개 성분을 초기화합니다. 5차원 이상의 벡터를 만들어 사용할 경우 그 뒤 성분은 0으로 초기화되지 않으니 주의하세요.

nvec(const nvec&)

복사 생성합니다. 다른 차원도 사용 가능하며 둘 중 더 작은 차원을 기준으로 초기화됩니다.

void set(a)

모든 성분을 하나의 값으로 초기화합니다. =과 동일합니다.

void set(const nvec&)

다른 벡터의 값을 복사해 옵니다. =과 동일합니다.

nvec& operator=(T)

모든 성분을 하나의 값으로 초기화합니다.

nvec& operator+=, -= , \*=, /=(const nvec&)

다른 벡터와 성분별 연산을 하여 이 벡터에 저장합니다. 다른 차원의 경우 둘 중 더 작은 차원을 기준으로 계산됩니다.

nvec operator+, -, \*, /(const nvec&)

다른 벡터와 성분별 연산을 하여 반환합니다. 다른 차원의 경우 둘 중 더 작은 차원을 기준으로 계산됩니다.

operator+=, -=, \*=, /=, +, -, \*, /(T a)

벡터의 모든 성분에 대하여 주어진 값과 연산합니다.

bool operator==, !=

벡터의 모든 성분이 동일하면 참을 반환합니다. 정밀도 오차에 대한 고려를 하지 않습니다.

T& operator[](ptrdiff\_t i)

벡터의 i번째 성분에 접근합니다.

operator T\*()

T형 배열로 사용할 수 있도록 포인터를 반환합니다. const 버전도 따로 있습니다.

operator nvec<D, T2>()

다른 타입이나 명시적 캐스팅이 가능한 성분 간 직접 캐스팅을 합니다. ivec에 나눗셈 연산을 할 때 소수점 단위가 필요한 경우 등이 필요할 때 코드 길이를 줄일 수 있습니다.

nvec normalize()

이 벡터와 동일한 방향이며 크기가 1인 벡터를 반환합니다.

float length2()

이 벡터의 유클리드 길이 제곱을 반환합니다. 반환값은 항상 float입니다.

float length()

이 벡터의 유클리드 길이를 반환합니다. 반환값은 항상 float입니다.

float distance2(const nvec&)

다른 벡터와의 거리의 제곱을 반환합니다. 반환값은 항상 float입니다.

float distance(const nvec&)

다른 벡터와의 거리를 반환합니다. 반환값은 항상 float입니다.

T dot(const nvec&)

다른 벡터와의 내적값을 반환합니다.

2. 행렬

**행렬은 아래 클래스로 표현됩니다.**

mat2, mat3, mat4

mat2는 2차원 이미지의 회전변환을 하는 데 사용할 수 있지만 성능이 극도로 중요한 2D 게임이 아닌 이상 사용할 일은 없을 것입니다.

mat3는 3차원 이미지의 회전변환 및 2차원 이미지의 아핀변환을 하는 데 사용할 수 있습니다. 마찬가지로 2차원을 위한 아핀변환으로는 사실상 사용할 일이 없습니다.

mat4는 3차원 이미지의 아핀변환을 하는 데 사용됩니다.

**\* 멤버 변수**

행렬의 멤버 변수는 각각 접근이 가능하지만 일반적으로 그럴 일은 없을 것입니다.

float a[]: 성분의 배열입니다. 인덱스는 행을 따라 증가합니다.

float \_11, \_12, ...: 1행 1열, 1행 2열, ...의 성분을 뜻합니다. 배열 a와 메모리를 공유합니다.

double 및 long double 성분은 직접 지원하지 않습니다.

**\* 멤버 함수**

mat()

각각의 단위 행렬을 생성합니다.

mat(float \_11, float \_12, ...)

각각의 성분을 지정하여 행렬을 생성합니다.

float& operator[] (ptrdiff\_t i)

a[i]와 동일한 연산입니다.

mat(const mat&)

행렬을 복사 생성합니다.

void toI()

단위행렬로 변경합니다.

mat& operator+=, -=(const mat&), mat operator +, - (const mat&)

다른 행렬과 성분별로 더하거나 뺍니다.

mat operator\*(const mat&), mat operator\*=(const mat&)

행렬곱을 수행합니다.

mat& operator\*=, /=(float), mat operator\*, /(float)

행렬에 실수배를 합니다.

float det()

행렬식을 반환합니다.  
mat inverse()

역행렬을 반환합니다.

mat transpose()

전치 행렬을 반환합니다.

operator float\*()

행 우선 순서로 된 배열을 반환합니다. openGL에서는 전달할 때 transpose를 참으로 하여 전달해야 합니다.

static mat3 translate(const vec2&), (float, float)

2차원 병진 아핀 변환 행렬을 반환합니다.

static mat3 scale(const vec2&), (float, float)

2차원 크기 아핀 변환 행렬을 반환합니다.

static mat3 rotate(float)

Z축 기준의 2차원 회전을 반환합니다. (라디안 입력)

static mat3 rotate(float x, float y, float z)

3차원 오일러 회전 행렬을 반환합니다.

mat3::operator mat2()

좌측 상단 2x2 행렬로 캐스트합니다.

mat4::operator mat3()

좌측 상단 3x3 행렬로 캐스트합니다.

static mat4 translate(const vec3&), (float, float, float)

3차원 병진 아핀 변환 행렬을 반환합니다.

static mat4 scale(const vec3&), (float, float, float)

3차원 크기 아핀 변환 행렬을 반환합니다.

static mat4 rotate(const vec3& axis, float angle)

axis를 축으로 angle(라디안)만큼 회전하는 3차원 회전 행렬을 반환합니다.

static mat4 rotate(float roll, float pitch, float yaw)

3차원 오일러 회전 행렬을 반환합니다. (라디안 입력)

static mat4 lookAt(const vec3& eye, const vec3& at, const vec3& up)

카메라의 위치와 방향을 정하는 [lookAt 행렬](https://www.songho.ca/opengl/gl_camera.html)을 반환합니다.

static mat4 TRS(const vec3& t, const Quaternion& r, const vec3& s)

병진(t), 회전(r), 크기(s)를 모두 적용한 아핀 변환을 반환합니다. 각각 구한 후 곱하는 것보다 빠릅니다.

static mat4 perspective(float fovy, float aspect, float dnear, float dfar)

[표준 뷰 볼륨 직육면체에 들어올 대상 뿔대(절두체)를 조절하는 행렬](https://www.songho.ca/opengl/gl_projectionmatrix.html)을 계산합니다.

3. 사원수

**사원수는 아래 클래스로 표현됩니다.**

Quaternion

3차원 회전을 표현하는 데 주로 사용됩니다.

**\* 멤버 변수**

float c1, ci, cj, ck

각각 실수부분/i부분/j부분/k부분입니다. 행렬과 마찬가지로 접근은 가능하나 개별 멤버에 접근하거나 조절할 상황은 사실상 없습니다.

**\* 멤버 함수**

모두 접근은 가능하나 static 멤버와 toMat4()를 제외하면 회전을 위해 사용될 일은 없다고 봐도 무방합니다.

float abs2()

사원수 크기의 제곱을 반환합니다.

float abs()

사원수 크기를 반환합니다.

Quaternion conjugate()

켤레(공액)사원수를 반환합니다.

Quaternion inverse()

사원수의 “우측”에 곱하면 1이 되는 값을 반환합니다.

Quaternion operator+, -, \*, /(const Quaternion&), Quaternion& operator+=,-=,\*=,/=(const Quaternion&)

사원수끼리 연산한 값을 반환합니다. \*=, /= 모두 오른쪽에 곱합니다.

void compound(const Quaternion&), void compound(const vec3&, float)

현재 사원수에 주어진 회전이 추가로 가해집니다.

static Quaternion rotation(const vec3& axis, float angle)

axis를 축으로 angle(라디안)만큼 회전하는 3차원 회전 사원수를 반환합니다.

static Quaternion euler(float yaw, float pitch, float roll)

오일러 회전에 해당하는 사원수를 생성합니다.

4. 전역 요소

float PI

원주율 값입니다.

float deg(float rad), float rad(float deg)

라디안을 도로, 도를 라디안으로 변경합니다.

vec3 cross(const vec3&, const vec3&)

두 3차원 벡터의 외적값을 반환합니다.

void print(프린트할 대상, const char\* tag=””, char end='\n')

벡터, 4x4 행렬, 사원수를 출력합니다. 비교적 편리한 디버그를 위한 함수입니다. nvec의 경우 2~4차원만 지원됩니다.

nvec lerp(const nvec&, const nvec&, const nvec&)

2개의 벡터의 선형 보간을 반환합니다. 각 차원별로 보간됩니다.

Quaternion lerp(const Quaternion&, const Quaternion&, float t)

사원수의 선형 보간을 반환합니다. t=0에 가까울수록 첫 번째 인수에 가까워집니다.

Quaternion slerp(const Quaternion&, const Quaternion&, float t)

사원수의 구면 선형 보간을 반환합니다. 이는 회전에 대한 보간을 의미합니다. t=0에 가까울수록 첫 번째 인수에 가까워집니다.

GL 커버 모듈

**\* GL 커버 모듈은 다음에서 사용되고 있습니다.**

main.cpp

**\* 개요**

GL 커버 모듈은 그래픽을 표시하기 위한 정점, 셰이더, 텍스처 등에 대한 등록을 정의하며 초기 세팅을 적당한 값으로 맞춥니다.

**\* 목차**

1. 초기 세팅

2. 정점

3. 셰이더

4. 텍스처

**\* 본문**

1. 초기 세팅

2. 정점

3. 셰이더

4. 텍스처