# Homework: 정육면체 만들기

컴퓨터공학과 1924073 최유현

구현 설명

기존의 drawcomponent.cpp코드를 참고하여

const auto로 정점을 8개 생성하였고, 그 점들을 add\_triangle로 이어서 하나의 정육면체를 만들었습니다. Vertex, edges, faces가 화면에 뜨도록 해놓았고 아래는 이에 대한 실행화면과 pmp write를 이용하여 생성한 obj파일을 meshlab으로 열어본 서식파일 입니다..  
  
텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명소프트웨어, 스크린샷, 멀티미디어 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명

obj파일도 잘 생성되었습니다.

2. 코드  
#include "DrawComponent.h"

#include "gl/freeglut.h"

#include "pmp/io/write\_obj.h"

#include "pmp/io/io.h"

#include "pmp/algorithms/normals.h"

void DrawComponent::Init()

{

InitMyMesh();

// compute normals

pmp::vertex\_normals(mesh);

pmp::write(mesh, "hm11.obj");

}

void DrawComponent::InitLoadFile()

{

pmp::read(mesh, "obj\\kitten.obj");

pmp::vertex\_normals(mesh);

}

void DrawComponent::InitMyMesh()

{

// Add 8 vertices for a cube

const auto v0 = mesh.add\_vertex(pmp::Point(0, 0, 0));

const auto v1 = mesh.add\_vertex(pmp::Point(1, 0, 0));

const auto v2 = mesh.add\_vertex(pmp::Point(1, 1, 0));

const auto v3 = mesh.add\_vertex(pmp::Point(0, 1, 0));

const auto v4 = mesh.add\_vertex(pmp::Point(0, 0, 1));

const auto v5 = mesh.add\_vertex(pmp::Point(1, 0, 1));

const auto v6 = mesh.add\_vertex(pmp::Point(1, 1, 1));

const auto v7 = mesh.add\_vertex(pmp::Point(0, 1, 1));

// Add 6 faces for the cube

mesh.add\_triangle(v0, v1, v2);

mesh.add\_triangle(v0, v2, v3);

mesh.add\_triangle(v1, v5, v6);

mesh.add\_triangle(v1, v6, v2);

mesh.add\_triangle(v5, v4, v7);

mesh.add\_triangle(v5, v7, v6);

mesh.add\_triangle(v4, v0, v3);

mesh.add\_triangle(v4, v3, v7);

mesh.add\_triangle(v3, v2, v6);

mesh.add\_triangle(v3, v6, v7);

mesh.add\_triangle(v4, v5, v1);

mesh.add\_triangle(v4, v1, v0);

std::cout << "vertices: " << mesh.n\_vertices() << std::endl;

std::cout << "edges: " << mesh.n\_edges() << std::endl;

std::cout << "faces: " << mesh.n\_faces() << std::endl;

}

void DrawComponent::Draw()

{

auto n = mesh.get\_vertex\_property<pmp::Normal>("v:normal");

for (auto f : mesh.faces()) {

glBegin(GL\_TRIANGLES);

for (auto v : mesh.vertices(f)) {

auto p = mesh.position(v);

glNormal3dv(n[v].data());

glVertex3dv(p.data());

}

glEnd();

}

}