



컴퓨터 그래픽스 입문



학번 : 2016110056

학과 : 불교학부

이름 : 박승원

날짜 : 2017년 4월 13일



Lab 6. Modeling

Programming Practice

0.1 Transform your ironman model to make it look correctly. (5pt)

Listing 1: obj파일 리더 헤더파일

```
#include<vector>
#include"matrix.h"

class ObjReader
{
public:
    ObjReader(std :: string filename);
    std :: vector<Matrix<float>> vertexes;
    std :: vector<unsigned> indices;
    float range[6]; // minx, miny, minz, maxx, maxy, maxz
};
```

Listing 2: obj파일 리더 구현부

```
#include<fstream>
#include<limits>
#include"readobj.h"
using namespace std;

ObjReader::ObjReader(string file )
{
    string s;
    ifstream f( file );
    while(getline(f, s)) {
        stringstream ss;
        ss << s;
        ss >> s;
        if(s == "v") {
            float x, y, z;
            ss >> x >> y >> z;
            vertexes.push_back(Matrix<float>{x,y,z});
        } else if(s == "f") {
            unsigned a, b;
            char c;
            while(ss >> a >> c >> c >> b) indices.push_back(a-1);
        }
    }
```

```

}
range[0] = range[1] = range[2] = numeric_limits<float>::max();
range[3] = range[4] = range[5] = numeric_limits<float>::min();
for(auto& a : vertexes) {
    for(int i=0; i<3; i++) if(a[1][i+1] < range[i]) range[i] = a[1][i+1];
    for(int i=0; i<3; i++) if(a[1][i+1] > range[i+3]) range[i+3] = a[1][i+1];
}
for(auto a : range) cout << a << ' ';
cout << endl << vertexes.size() << ' ' << indices.size() << endl;
}

```

Listing 3: 실행파일

```

#include<chrono>
#include<thread>
#include<iostream>
#include"glutil.h"
#include"readobj.h"
using namespace std;
extern Matrix<float> KeyBindMatrix;

int main(int ac, char** av)
{
    if (! glfwInit () ) return -1;

    GLFWwindow* window = glfwCreateWindow(640, 480, "Color Cube", NULL, NULL);
    if (! glinit (window)) return -1;

    Matrix<float> mm{4,4};

    ObjReader obj3d(av[1]);
    vector<Matrix<float>> color{obj3d.vertexes.size(), Matrix<float>{1,0,0}};
    unsigned fv = gl_transfer_data (obj3d.vertexes);
    unsigned fc = gl_transfer_data (color);
    unsigned fe = gl_transfer_index (obj3d.indices);
    float mid[3] = {obj3d.range[3] - obj3d.range[0],
                   obj3d.range[4] - obj3d.range[1],
                   obj3d.range[5] - obj3d.range[2]};
    Matrix<float> scale {4,4}, translate {4,4};
    scale . glscale (1.0/ mid[0], 1.0/ mid[1], 1.0/ mid[2]);
    translate . gltranslate (mid[0]/2, mid[1]/2, mid[2]/2);
    mm = scale;
    // compile shaders

```

```

unsigned shader_program =
    make_shader_program("src/vertex_shader . glsl ", "src/fragment_shader . glsl ",
        "a_pos", "a_color");
if (!shader_program) return 0;
while (!glfwWindowShouldClose(window)) {
    glClear (GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

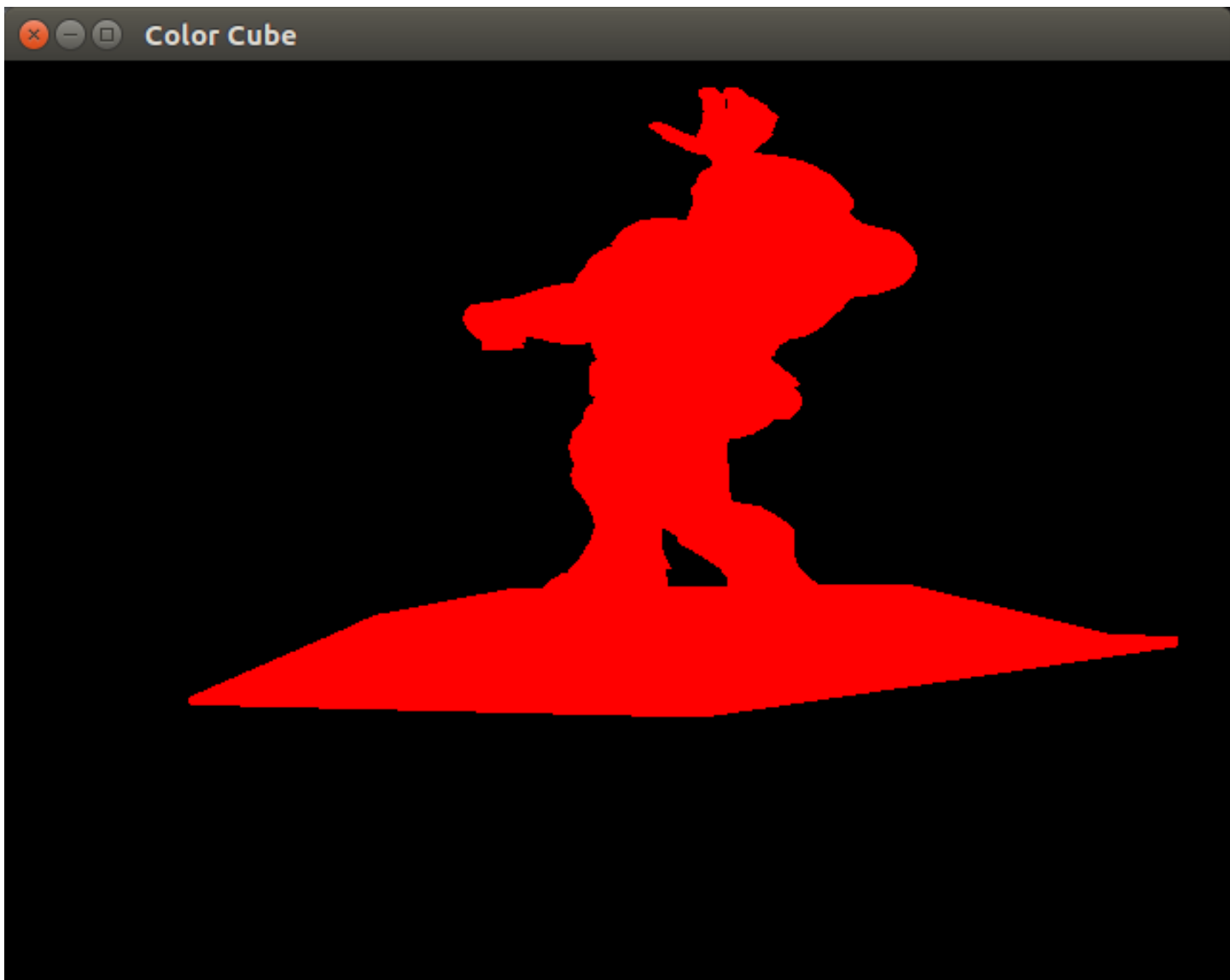
    glUseProgram(shader_program);
    auto tm = KeyBindMatrix * scale ;
    transfer_matrix (shader_program, tm, "KeyBindMatrix");

    gl_bind_data (fv, fc, fe);
    glDrawElements(GL_TRIANGLES, obj3d.indices.size(), GL_UNSIGNED_INT, 0);

    glfwSwapBuffers(window);
    glfwPollEvents ();
    this_thread :: sleep_for (chrono :: milliseconds (50) );
}
glfwTerminate();
}

```

대부분의 모듈은 이미 지난 시간에 제출한 것을 재활용해서 다시 올리지 않습니다.



파일명을 인자로 해서 obj 파일을 읽어들이는 방식으로 했다.
scale만을 조절해서 아이언맨을 표시할 수 있었다.

0.2 Use Blender (or other modeling software) to model a simple (and probably cute) 3D smiley. Export it to an OBJ file and read it in YOUR program. (5pt)

Listing 4: 위의 실행 파일에서 눈을 검게 하고 구를 노란색으로 하기 위해 한 줄만을 첨가했다.

```
#include<chrono>
#include<thread>
#include<iostream>
#include"glutil.h"
#include"readobj.h"
using namespace std;
extern Matrix<float> KeyBindMatrix;

int main(int ac, char** av)
{
    if (! glfwInit () ) return -1;
```

```

GLFWwindow* window = glfwCreateWindow(640, 480, "Color Cube", NULL, NULL);
if (! glinit (window)) return -1;

Matrix<float> mm{4,4};

ObjReader obj3d(av[1]);
vector<Matrix<float>> color{obj3d.vertexes.size(), Matrix<float>{1,1,0}};
for(int i=0; i<64; i++) color[i] = Matrix<float>{0,0,0};
unsigned fv = gl_transfer_data (obj3d.vertexes);
unsigned fc = gl_transfer_data (color);
unsigned fe = gl_transfer_index (obj3d.indices);
float mid[3] = {obj3d.range[3] - obj3d.range[0],
                obj3d.range[4] - obj3d.range[1],
                obj3d.range[5] - obj3d.range[2]};

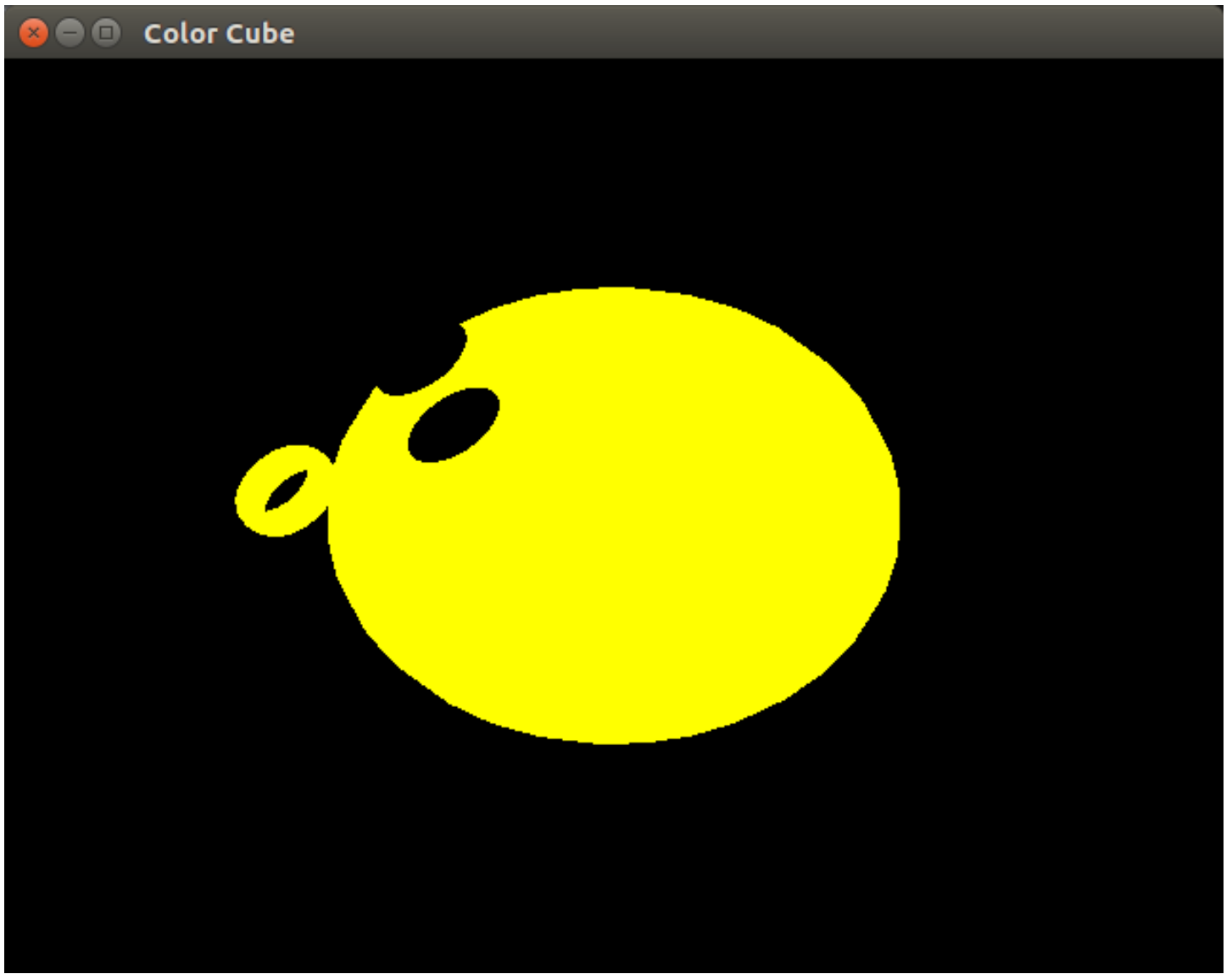
Matrix<float> scale {4,4}, translate {4,4};
scale . glscale (1.0/ mid[0], 1.0/ mid[1], 1.0/ mid[2]);
translate . gltranslate (mid[0]/2, mid[1]/2, mid[2]/2);
mm = scale;
///compile shaders
unsigned shader_program =
    make_shader_program("src/vertex_shader . glsl ", "src/fragment_shader . glsl ",
        "a_pos", "a_color");
if (! shader_program) return 0;
while (! glfwWindowShouldClose(window)) {
    glClear (GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    glUseProgram(shader_program);
    auto tm = KeyBindMatrix * scale ;
    transfer_matrix (shader_program, tm, "KeyBindMatrix");

    gl_bind_data (fv, fc, fe);
    glDrawElements(GL_TRIANGLES, obj3d.indices.size(), GL_UNSIGNED_INT, 0);

    glfwSwapBuffers(window);
    glfwPollEvents ();
    this_thread :: sleep_for (chrono:: milliseconds (50));
}
glfwTerminate();
}

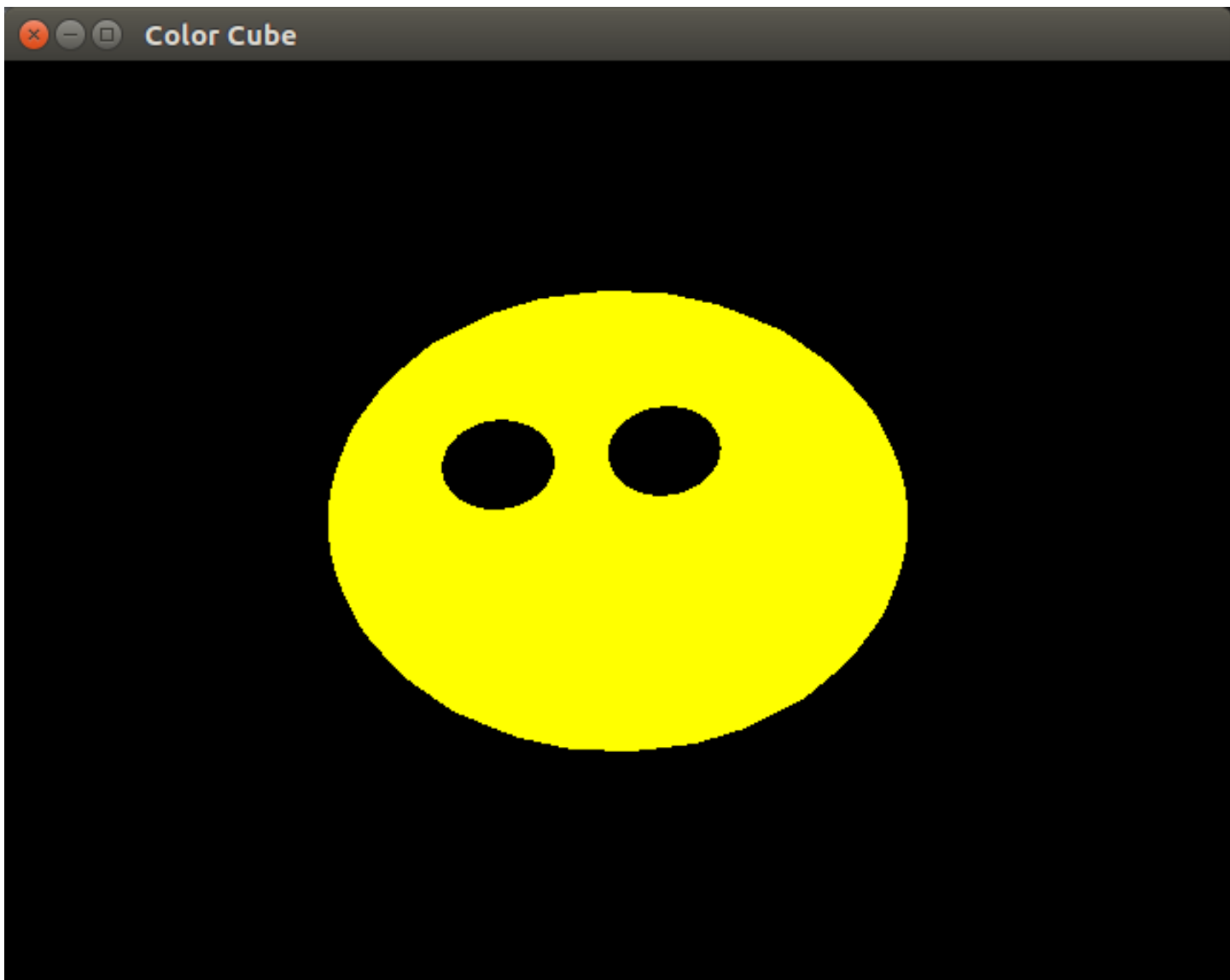
```



입이 있다는 것을 표현하기 위해 옆으로 돌린 모습.

블렌더가 익숙하지 않아 예쁜 모양은 나오지 않았지만, 얼굴을 구로 눈을 디스크로 입을 토러스로 표현해보았다.

입의 색깔이 같은 색이라 정면에서는 보이지 않아 옆으로 돌려보았다.



정면에서 본 모습
삼각형으로 만들어 obj를 구현하는 데에는 성공했다.
Ugly!! 죄송합니다...