컴퓨터 그래픽스 입문

학번 : 2016110056

학과 : 불교학부

이름 : 박승원

날짜 : 2017년 3월 30일

dongguk
UNIVERSITY

제 1 절　Geometric Transformation

Programming Practice

(Write down what you did and how you made the result. Otherwise I don't know what you did.)

1. Change the size of the box using scale.(3pt)

    2. Translate the position of the box.(3pt)

    3. Revolve the box around the origin.(2pt)

    4. Add the box that is revolved along previous box. (1pt)

    5. The added box revolves and rotates like earth. (1pt)
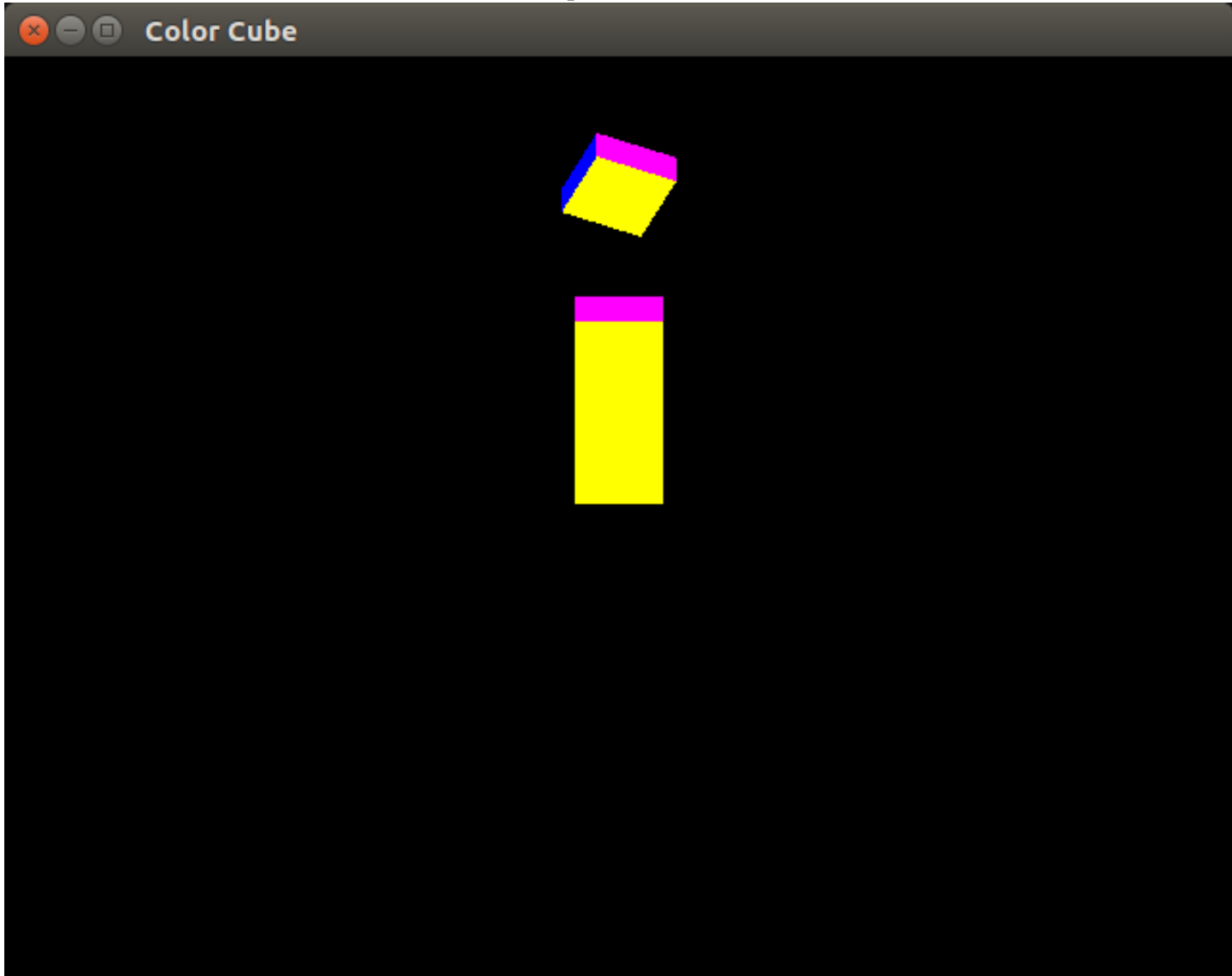

```cpp
#include<chrono>
#include<thread>
#include" glutil .h"
using namespace std;
extern Matrix<float> KeyBindMatrix;

int main()
{
    Matrix<float> translate {4,4};
     translate . gltranslate  (0,0, sqrt (2) ) ;
```

```cpp
    if (! glfwInit ()) return −1;
    GLFWwindow* window = glfwCreateWindow(640, 480, "Color Cube", NULL, NULL);
    if (! glinit (window)) return −1;
    glortho (10) ;

    auto pl = polygon(4) ;
    vector<Matrix<float>> pl2;
    pl2. insert (pl2.end(), begin(pl), end(pl)) ;
    pl = translate * pl; // z+1
    pl2. insert (pl2.end(), begin(pl), end(pl)) ; // append elevated 4 vertex
    // for (auto& a : pl2) a = grotate * a ;// rotate a little to have a good view
    valarray<float> color(72) ; // {} does not make 72 size −  initialize_list   construct
    color [ slice (0,12,3) ] = 1;
    color [ slice (26,12,3) ] = 1;
    color [ slice (13,4,3) ] = 1;
    color [ slice (49,8,3) ] = 1;

    int idx [24] = {0,1,2,3,   4,5,6,7,   0,1,5,4,   1,2,6,5,   2,3,7,6,   0,3,7,4};
    vector<Matrix<float>> v;
    for(auto& a : idx) v.push_back(pl2[a]) ;

    auto fc = gl_transfer_data ( color ) ;
    auto fv = gl_transfer_data (v) ;
    Matrix<float> con{4,4}, m{4,4};
    const float r = 0.7;

    vector<Matrix<float>> v2{v}, v3{v};
    con = con. gltranslate (0,3,0) * con. glscale (1,3,1) * con. glrotateZ (M_PI/4);
    for(auto& a : v) a = con * a;
    con = con. gltranslate (0,8,0) * con. glrotateX (0.1) ;
    auto fv2 = gl_transfer_data (v2) ;
//  for (auto& a : v2) a = con * a;
//  con.E() ;
//  con = con. glrotateX (0.1) ;
    float k = 0.1;
    while (! glfwWindowShouldClose(window)) {
        glClear (GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

        // 공전하는  긴  막대
        for(auto& a : v) a = m.glrotateX (0.1) * a;
        gl_transfer_data (v, fv) ;
        glBindBuffer (GL_ARRAY_BUFFER, fc);
```

```
            glEnableClientState (GL_COLOR_ARRAY);
            glColorPointer (3, GL_FLOAT, 0, nullptr);

            glBindBuffer (GL_ARRAY_BUFFER, fv);
            glEnableClientState (GL_VERTEX_ARRAY);
            glVertexPointer (3, GL_FLOAT, 0, nullptr); //3 float is 1 vertex stride 0,
            glDrawArrays(GL_QUADS, 0, 24);//mode, first, count

            glDisableClientState (GL_COLOR_ARRAY);
            glDisableClientState (GL_VERTEX_ARRAY);

            // 스스로    자전하는    큐브
            int i=0;
            for(auto& a : v2) a = m.glrotateZ (0.1) * a;
            v3 = v2; // 자체적인    회전을    위한    변환
            for(auto& a : v3) a = m.glrotateX(k) * m. gltranslate (0,8,0) * a; // 이동
            k+=0.1; //  회전각도를    증가
            gl_transfer_data (v3, fv2);
            glBindBuffer (GL_ARRAY_BUFFER, fc);
            glEnableClientState (GL_COLOR_ARRAY);
            glColorPointer (3, GL_FLOAT, 0, nullptr);

            glBindBuffer (GL_ARRAY_BUFFER, fv2);
            glEnableClientState (GL_VERTEX_ARRAY);
            glVertexPointer (3, GL_FLOAT, 0, nullptr); //3 float is 1 vertex stride 0,
            glDrawArrays(GL_QUADS, 0, 24);//mode, first, count

            glDisableClientState (GL_COLOR_ARRAY);
            glDisableClientState (GL_VERTEX_ARRAY);

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

소스 코드에 주석을 참조바랍니다.