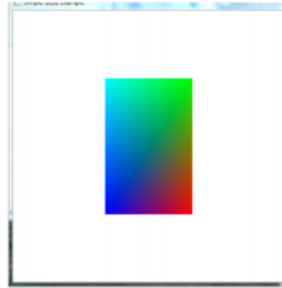


**SORU :**

Aşağıdaki adımları takip ediniz:

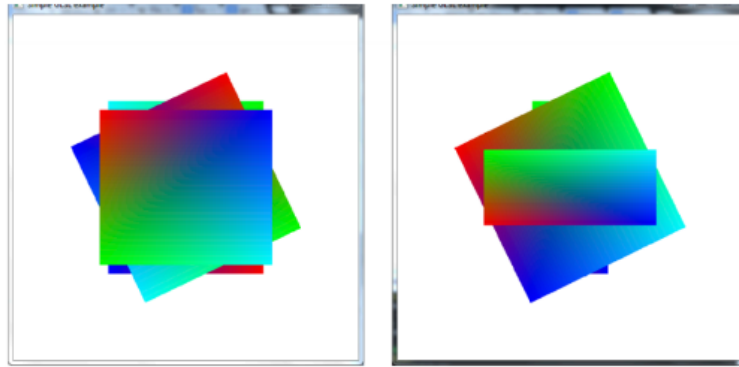
Sahne içine sadece y-düzlemine göre dönebilen bir renkli dikdörtgen yerleştiriniz (bkz. Şekil 1).



**Şekil 1**

Sadece z-düzlemine göre dönebilen ikinci bir renkli dikdörtgen yerleştiriniz.

Sadece x-düzlemine göre dönebilen üçüncü bir renkli dikdörtgen yerleştiriniz (bkz. Şekil 2).



**Şekil 2**

Bu üç dikdörtgen birbirilerine yakın konumda olacak şekilde sahne içine yerleştirilmelidir.

Her üç dikdörtgen de belirtilen eksen etrafında dönmelidir.

Dikdörtgenler genList ile oluşturulmalıdır.

Dönme esnasında düzlemler birbiri içine geçmelidir.

Programın ilk aşamasında derinlik testi yapmadan sonuç elde edilmeli (sonuç Şekil 2 'deki gibi olmalıdır).

Programın ikinci aşamasında derinlik testi uygulanmalıdır (sonuç Şekil 3 'deki gibi olmalıdır)

**YANIT :**

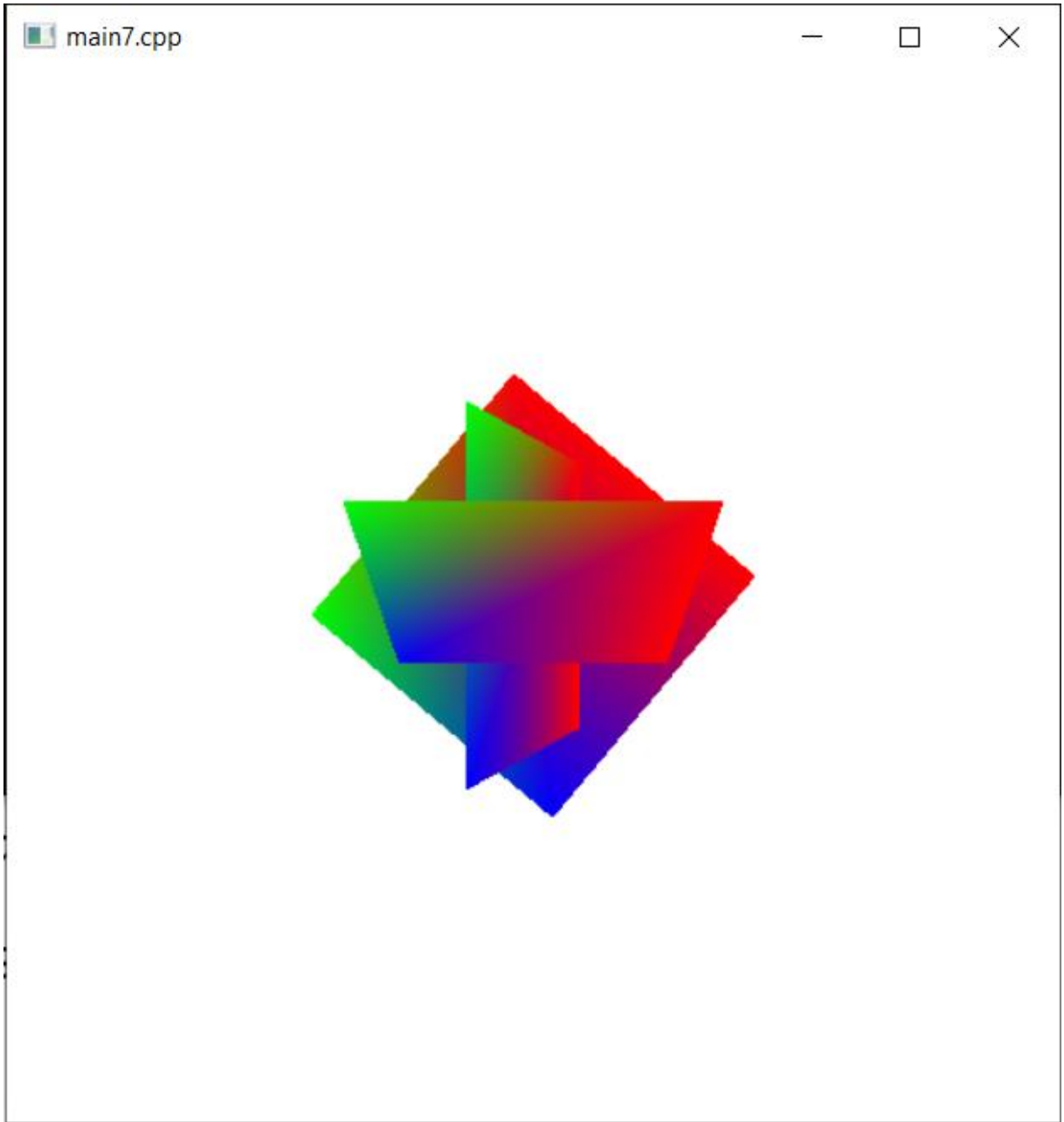
Tüm dikdörtgenler GL\_QUADS ile çizildi.

glTranslatef fonksiyonu ile yerleri X,Y 0 da Z değeri -5 te olacak şekilde iç içe çizildi.

X,Y,Z eksenlerinde döndürme işlemini glRotatef fonksiyonu ile yapıldı.

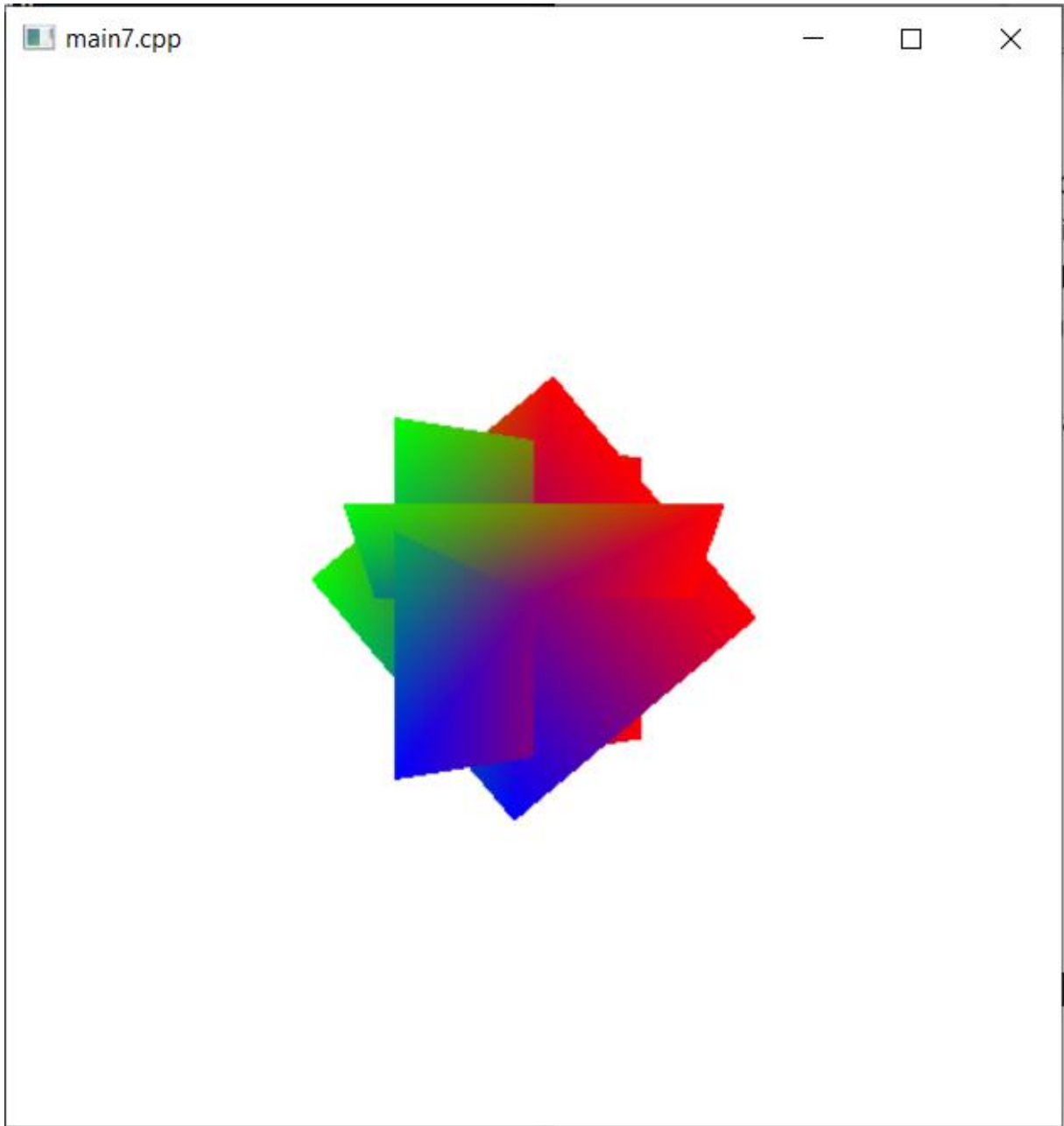
Küçük harf x, y ve z tuşları ile basılan tuş eksenindeki dikdörtgen döndürüldü.

Depth buffersız ekran ıktısı



Depth buffer olmadan i ie Őekilde derinliđi yapamadım.

Depth buffer eklendikten sonraki ekran ıktısı



Depth buffer ile iç içe geçme işlemini sorunsuz yapmakta.

Program kodlarında depth buffersız program ile depth buffer eklenmiş hali arasındaki farklılıklar yeşil ile gösterilmiştir.

### Program kodları

```
#include <iostream>

#include <GL/freeglut.h>

static float AngleX = 0.0; //X açısı değişkeni
static float AngleY = 0.0; //Y açısı değişkeni
static float AngleZ = 0.0; //Z açısı değişkeni

void drawScene(void)
{
    glClear(GL_COLOR_BUFFER_BIT);



---


    glEnable(GL_DEPTH_TEST); //depth buffer eklentisi
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); //depth
buffer eklentisi


---


//Z ekseninde hareket edecek dikdörtgen

    glLoadIdentity();

    glTranslatef(0.0, 0.0, -5.0); //Z ekseninde 5 birim geride
    glRotatef(AngleZ, 0.0, 0.0, 1.0); //Z açısı ve ekseninde
hareket ettirilerek çizdiriliyor.

    glColor3f(0.0, 0.0, 0.0);

    glBegin( GL_QUADS );

    glColor3f( 1.0f, 0.0f, 0.0f );
    glVertex3f( 1.0f, 1.0f, 0.0f );
    glColor3f( 0.0f, 1.0f, 0.0f );
    glVertex3f( -1.0f, 1.0f, 0.0f );
    glColor3f( 0.0f, 0.0f, 1.0f );
    glVertex3f( -1.0f, -1.0f, 0.0f );
    glColor3f( 1.0f, 0.0f, 0.0f );
    glVertex3f( 1.0f, -1.0f, 0.0f );

    glEnd();
```

**//Y ekseninde hareket edecek dikdörtgen**

```
glLoadIdentity();
glTranslatef(0.0, 0.0, -5.0); //Z ekseninde 5 birim geride
glRotatef(AngleY, 0.0, 1.0, 0.0); //Y açısı ve ekseninde
hareket ettirilerek çizdiriliyor.
glColor3f(0.0, 0.0, 0.0);
glBegin( GL_QUADS );
glColor3f( 1.0f, 0.0f, 0.0f );
glVertex3f( 1.0f, 1.0f, 0.0f );
glColor3f( 0.0f, 1.0f, 0.0f );
glVertex3f( -1.0f, 1.0f, 0.0f );
glColor3f( 0.0f, 0.0f, 1.0f );
glVertex3f( -1.0f, -1.0f, 0.0f );
glColor3f( 1.0f, 0.0f, 0.0f );
glVertex3f( 1.0f, -1.0f, 0.0f );
glEnd();
```

**//X ekseninde hareket edecek dikdörtgen**

```
glLoadIdentity();
glTranslatef(0.0, 0.0, -5.0); //Z ekseninde 5 birim geride
glRotatef(AngleX, 1.0, 0.0, 0.0); //X açısı ve ekseninde
hareket ettirilerek çizdiriliyor.
glColor3f(0.0, 0.0, 0.0);
glBegin( GL_QUADS );
glColor3f( 1.0f, 0.0f, 0.0f );
glVertex3f( 1.0f, 1.0f, 0.0f );
glColor3f( 0.0f, 1.0f, 0.0f );
glVertex3f( -1.0f, 1.0f, 0.0f );
glColor3f( 0.0f, 0.0f, 1.0f );
glVertex3f( -1.0f, -1.0f, 0.0f );
glColor3f( 1.0f, 0.0f, 0.0f );
glVertex3f( 1.0f, -1.0f, 0.0f );
```

```

glEnd();
glutSwapBuffers();
}

```

---

```

void resize(int w, int h) //Hareket ettiği durumda tekrar boyutlandırıyor.
{
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glFrustum(-1.0, 1.0, -1.0, 1.0, 1.5, 20.0); //bakış açısını ayarlıyor.
/*glFrustum(left, right, bottom, top, near, far);
    glMatrixMode(GL_MODELVIEW);
}

void keyInput(unsigned char key, int x, int y)
{
    switch (key)
    {
        case 'x': //küçük x tuşu basıldı mı?
            AngleX += 10.0; //x açısını 10 arttırıyor.
            glutPostRedisplay(); //tekrar çizdirme fonksiyonu
            break;
        case 'y': //küçük y tuşu basıldı mı?
            AngleY += 10.0; //y açısını 10 arttırıyor.
            glutPostRedisplay();//tekrar çizdirme fonksiyonu
            break;
        case 'z': //küçük z tuşu basıldı mı?
            AngleZ += 10.0; //z açısını 10 arttırıyor.
            glutPostRedisplay();//tekrar çizdirme fonksiyonu
            break;
        default:break;}}

```

```
int main(int argc, char **argv)
{
    glutInit(&argc, argv);

    glutInitContextVersion(4, 3);
    glutInitContextProfile(GLUT_COMPATIBILITY_PROFILE);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("main7.cpp");
    glutDisplayFunc(drawScene);
    glutReshapeFunc(resize);
    glutKeyboardFunc(keyInput); //klavyeden tuş basıldığında
tetiklenecek fonksiyon.
    glClearColor(1.0, 1.0, 1.0, 0.0);
    glutMainLoop();
}
```

Çağın AĞIRDEMİR