



BİLGİSAYAR MÜHENDİSLİĞİ

CSE 461

Bilgisayar Grafikleri

ÖDEV#1 RAPORU

Nevra Gürses

161044071

I. GÖREV TANIMI

OPENGL ve GLUT kütüphaneleri kullanarak 3 boyutlu bir ev yapılacaktır. Bu ev için etkileşimli olarak öteleme, döndürme ve ölçekleme dönüşümleri uygulanacaktır. Dönüştürme işlemleri için klavye tuşları kullanılacaktır.

II. YÖNTEM VE SONUÇ

İlk olarak projede kullanılacak olan GLUT kütüphanesi **glutInit** fonksiyonu kullanarak başlatıldı. Ardından, **glutInitDisplayMode** fonksiyonu ile görüntüleme modu ayarlandı, **glutInitWindowSize** fonksiyonu ile pencere boyutunu ayarlandı ve üst düzey pencere **glutCreateWindow** fonksiyonu ile oluşturuldu. Sunucu tarafında GL yeteneklerini etkinleştirmeyi veya devre dışı bırakmayı sağlayan **glEnable** fonksiyonu "GL_COLOR_MATERIAL" ve "GL_DEPTH_TEST" değerlerini aktif hale getirmek için kullanıldı.

Yapılan bu işlemler main fonksiyonu içerisinde uygulanmaktadır.

Geçerli pencere için ekran geri aramasını (callback) ayarlayan **glutDisplayFunc** fonksiyonu, geçerli pencere için klavye geri aramasını(callback) ayarlayan **glutKeyboardFunc** fonksiyonu ve geçerli pencere için yeniden şekillendirme geri aramasını(callback) ayarlayan **glutReshapeFunc** fonksiyonu sırasıyla main fonksiyonu içerisinde çağırıldı. Aşağıda bu işlemlerin ekran görüntüsü gösterilmektedir.

```
glutDisplayFunc(constructHome);  
glutKeyboardFunc(keyboardKeys);  
glutReshapeFunc(reshapeWindow);
```

Bu fonksiyonların parametrelerinde kullanılan fonksiyonların içinde yapılanlar ayrıntılı olarak anlatılacaktır.

3D Evin Oluşturulması

Ev oluşturulmadan önce gerekli matris ayarlamaları, kamera pozisyonları, açı gibi önemli faktörler oluşturulmuştur. Bu kısım gösterilecek olunursa:

```
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);  
glMatrixMode(GL_MODELVIEW);  
glLoadIdentity();  
glRotatef(camera_angle, 0.0f, 1.0f, 0.0f);  
glTranslatef(-1.0f, -1.5f, -2.0f);  
glPopMatrix();  
glPushMatrix();  
glTranslatef(1.0f, 1.0f, 0.0f);
```

Dönme, Öteleme ,Ölçekleme işlemleri:

Evin dönme, öteleme, ölçekleme gibi hareketleri yapabilmesi için **glRotatef**, **glTranslatef** **glScalef** fonksiyonları kullanıldı. glRotatef , mevcut matrisi bir rotasyon matrisiyle çarpmayı; glTranslatef, mevcut matrisi bir çeviri matrisiyle çarpmayı; glScalef mevcut matrisi genel bir ölçekleme matrisiyle çarpmayı sağlamaktadır.

Dönme işlemi, dönme açısı ve x,y,z koordinatlarına göre ayrı ayrı ayarlanmıştır. Bahsedilen bu işlemler gösterilecek olunursa:

```
/*Translation,rotation and scaling operations.*/
glTranslatef(angle_translateX,angle_translateY,angle_translateZ);
glRotatef(angle_rotateZ, 0.0f, 0.0f, 1.0f);
glRotatef(angle_rotateX, 1.0f, 0.0f, 0.0f);
glRotatef(angle_rotateY, 0.0f, 1.0f, 0.0f);
glScalef(angle_scaleX, angle_scaleY, angle_scaleZ);
```

Dönme, çevirme ve ölçekleme açılarının ilk değerleri global olarak tanımlanmıştır.

```
GLfloat angle_rotateX = 0.0f;
GLfloat angle_rotateY = 300.0f;
GLfloat angle_rotateZ = 0.0f;
GLfloat angle_translateX = 0.0f;
GLfloat angle_translateY = 0.0f;
GLfloat angle_translateZ =0.0f;
GLfloat angle_scaleX = 1.0f;
GLfloat angle_scaleY = 1.0f;
GLfloat angle_scaleZ = 1.0f;
```

angle_rotateY açısı ev ilk gözüktüğünde 3 boyutlu olduğunun net olarak belli olması için 300.0f olarak ayarlanmıştır.

Ardından, ev poligonlar ve üçgenler kullanılarak oluşturulmuştur. Renklendirmeler **glColor3f** fonksiyonu; x ,y, z koordinatları ile nokta(vertex) belirlenmesi **glVertex3f** fonksiyonu ile sağlanmıştır. Poligon ve üçgenler kullanarak ev oluşturma kısmı gösterilecek olunursa:

```

/* Right (according to us) rectangle*/
glBegin(GL_POLYGON);
glColor3f (1.0, 0.11, .68);
glVertex3f (0, 0, 0.0);
glVertex3f (0.5, 0, 0.0);
glVertex3f (0.5, 0.5, 0.0);
glVertex3f (0.0, 0.5, 0.0);
glEnd();

/* Left (according to us) rectangle*/
glBegin(GL_POLYGON);
glColor3f (1.0, 0.11, .68);
glVertex3f (0, 0, 0.5);
glVertex3f (0.5, 0.0, 0.5);
glVertex3f (0.5, 0.5, 0.5);
glVertex3f (0, 0.5, 0.5);
glEnd();

/* Front rectangle */
glBegin(GL_POLYGON);
glColor3f (0.73, 0.16, .96);
glVertex3f (0.5, 0, 0.5);
glVertex3f (0.5, 0, 0.0);
glVertex3f (0.5, 0.5, 0.0);
glVertex3f (0.5, 0.5, 0.5);
glEnd();

/* Back rectangle */
glBegin(GL_POLYGON);
glColor3f (0.73, 0.16, .96);
glVertex3f (0, 0, 0.5);
glVertex3f (0, 0, 0.0);
glVertex3f (0, 0.5, 0.0);
glVertex3f (0, 0.5, 0.5);
glEnd();

/* Bottom rectangle */
glBegin(GL_POLYGON);
glColor3f (0.73, 0.16, .96);
glVertex3f (0, 0, 0.0);
glVertex3f (0, 0, 0.5);
glVertex3f (0.5, 0, 0.5);
glVertex3f (0.5, 0, 0.0);
glEnd();

/* Front triangle roof*/
glBegin(GL_TRIANGLES);
glColor3f (.55, 0.2, .2);
glVertex3f (0.5, 0.5, 0.0);
glVertex3f (0.5, 0.5, 0.5);
glVertex3f (0.5, 0.8, 0.25);
glEnd();

/* Back triangle roof */
glBegin(GL_TRIANGLES);
glColor3f (.55, 0.2, .2);
glVertex3f (0, 0.5, 0.0);
glVertex3f (0, 0.5, 0.5);
glVertex3f (0, 0.8, 0.25);
glEnd();

/* Right (according to us) roof*/
glBegin(GL_POLYGON);
glColor3f (.45, 0.2, .2);
glVertex3f (0, 0.5, 0.0);
glVertex3f (0.5, 0.5, 0.0);
glVertex3f (0.5, 0.8, 0.25);
glVertex3f (0, 0.8, 0.25);
glEnd();

/* Left(according to us) roof*/
glBegin(GL_POLYGON);
glColor3f (.45, 0.2, .2);
glVertex3f (0, 0.5, 0.5);
glVertex3f (0.5, 0.5, 0.5);
glVertex3f (0.5, 0.8, 0.25);
glVertex3f (0, 0.8, 0.25);
glEnd();

/* Left(according to us) window*/
glBegin(GL_POLYGON);
glColor3f (.15, 0.5, .5);
glVertex3f (0.51, 0.33, 0.45);
glVertex3f (0.51, 0.33, 0.32);
glVertex3f (0.51, 0.48, 0.32);
glVertex3f (0.51, 0.48, 0.45);
glEnd();

/* Right (according to us) window*/
glBegin(GL_POLYGON);
glColor3f (.15, 0.5, .5);
glVertex3f (0.51, 0.33, 0.05);
glVertex3f (0.51, 0.33, 0.18);
glVertex3f (0.51, 0.48, 0.18);
glVertex3f (0.51, 0.48, 0.05);
glEnd();

/* Door */
glBegin(GL_POLYGON);
glColor3f (.4, 0.2, .1);
glVertex3f (0.51, 0, 0.18);
glVertex3f (0.51, 0, 0.32);
glVertex3f (0.51, 0.3, 0.32);
glVertex3f (0.51, 0.3, 0.18);
glEnd();

```

Yapılan işlemlerden sonra oluşan 3 boyutlu ev aşağıda gösterilmektedir:



reshapeWindow(int width, int height) fonksiyonu:

glutReshapeFunc fonksiyonunun geçerli pencere için yeniden şekillendirme geri aramasını ayarladığından bahsedilmiştir. Bu fonksiyonun parametresinde kullanılan reshapeWindow fonksiyonu içerisinde görüntü alanını ayarlayan **glViewport** fonksiyonu, matris modunu ayarlayan **glMatrixMode** fonksiyonu, mevcut matrisi kimlik matrisi ile değiştiren **glLoadIdentity** fonksiyonu ve bir perspektif izdüşüm matrisi oluşturulan **gluPerspective** fonksiyonu kullanılmıştır.

keyboardKeys(unsigned char key, int x, int y) fonksiyonu:

glutKeyboardFunc fonksiyonunun geçerli pencere için klavye geri aramasını ayarladığından bahsedilmiştir. Bu fonksiyonun parametresinde kullanılan keyboardKeys fonksiyonu içerisinde klavye tuşlarına basınca gerçekleştirilecek işlemler belirlenmiştir.

Öncelikle hangi klavye tuşunun hangi işlemi sağlayacağı #define yönergesi kullanılarak belirlenmiştir:

```
#define ROTATE_XPLUS_KEY 'u'
#define ROTATE_XMINUS_KEY 'n'
#define ROTATE_YPLUS_KEY 'k'
#define ROTATE_YMINUS_KEY 'h'
#define ROTATE_ZPLUS_KEY 'o'
#define ROTATE_ZMINUS_KEY 'l'
#define TRANSLATE_YPLUS_KEY 'w'
#define TRANSLATE_YMINUS_KEY 'z'
#define TRANSLATE_XPLUS_KEY 'd'
#define TRANSLATE_XMINUS_KEY 'a'
#define TRANSLATE_ZPLUS_KEY 'r'
#define TRANSLATE_ZMINUS_KEY 'f'
#define SCALE_PLUS '+'
#define SCALE_MINUS '-'
```

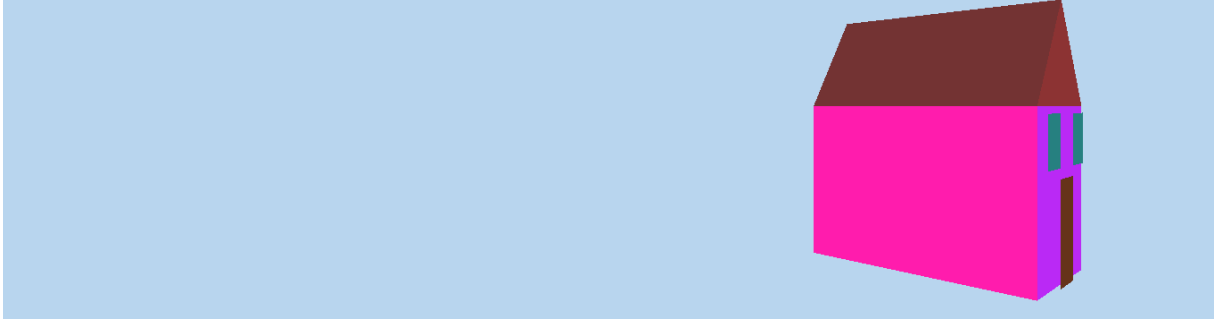
Dönme işlemleri için k,h,u,n,o,l klavye tuşları, öteleme işlemleri için a,d,w,z,r,f klavye tuşları ve ölçeklendirme işlemleri için + ve – tuşları kullanılmaktadır.

Basılan klavye tuşuna göre ilgili açı (dönme, öteleme veya ölçeklendirme) pozitif veya negatif olarak değişmektedir ve böylece hareket sağlanabilmektedir. If-else koşulları ile klavye tuşu kontrolleri ve faaliyetleri sağlanmıştır. Bu kısımdan örnek olarak küçük bir ekran görüntüsü verilecek olursa:

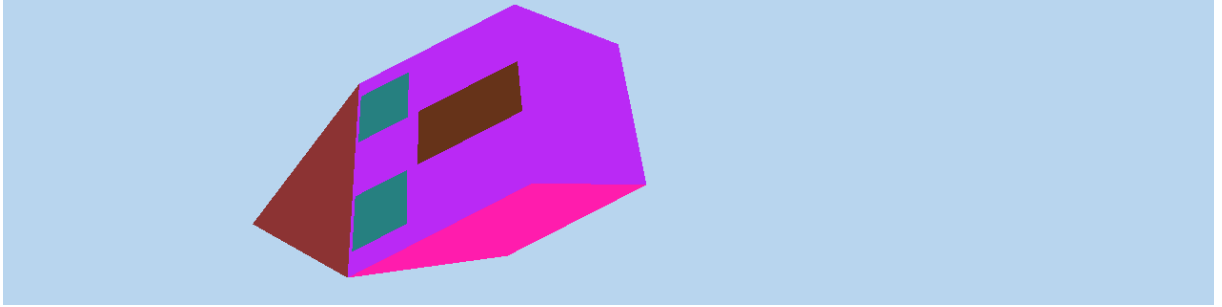
```
/*This function gets keyboard keys and changes angles for rotating,translation and scaling*/
void keyboardKeys(unsigned char key, int x, int y) {

    if(key == ROTATE_XPLUS_KEY){
        angle_rotateX += 1.0;
    }
    else if(key == ROTATE_XMINUS_KEY){
        angle_rotateX -= 1.0;
    }
    else if(key == ROTATE_YPLUS_KEY){
        angle_rotateY += 1.0;
    }
    else if(key == ROTATE_YMINUS_KEY){
        angle_rotateY -= 1.0;
    }
}
```

Tüm yapılan işlemler sonucunda öteleme, dönme ve ölçekleme işlemlerinden bazı ekran görüntüleri gösterilecek olunursa:



Öteleme Örneği



Dönme Örneği



Ölçekleme Örneği

ÖDEV VİDEO LİNKİ:

<https://www.youtube.com/watch?v=fKHvzFkqaR8>