CSE 461 – ÖDEV 1

Fatih Selim YAKAR -161044054

BİLGİSAYAR GRAFİKLERİ ÖDEV 1 DOKÜMANTASYONU

1. Sahnenin Ve Pencerenin Oluşturulması

Öncelikle glutInit(…) fonksiyonu kullanılarak GLUT kütüphanesi başlatıldı. Ardından glutInitDisplayMode(..) fonksiyonu kullanılarak başlangıç modu GLUT_DOUBLE, GLUT_RGB, GLUT_DEPTH bayrakları ile ilklendirildi. Daha sonra glutInitWindowSize(..) fonksiyonu ile program başlayınca oluşturulacak pencerenin boyutları belirlendi ve glutCreateWindow(…) ile pencere oluşturuldu.

```
int main(int argc, char * * argv) {
    glutInit( & argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(1280, 720);
    glutCreateWindow("Computer Graphics HW1 - 3d Home");
```

2. Aydınlatma Modelinin İlklendirilmesi

glColorMaterial(…) ile malzeme renginin mevcut rengi izlemesi sağlandı. Ardından glMaterialfv(…) fonksiyonları ile aydınlatma modeli için malzeme parametreleri belirtildi. Son olarak ise glEnable ile ışıklandırma yetenekleri devreye alındı.

```
void initialize() {
    glColorMaterial(GL_FRONT, GL_AMBIENT_AND_DIFFUSE);
    glEnable(GL_COLOR_MATERIAL);
    GLfloat specular[] = { 0.3, 0.3, 0.3, 0.3  };
    GLfloat shininess[] = { 90.0 };
    GLfloat ambient[] = { 0.1, 0.1, 0.1, 0.1 };
    glMaterialfv(GL_FRONT, GL_AMBIENT, ambient);
    glMaterialfv(GL_FRONT, GL_SPECULAR, specular);
    glMaterialfv(GL_FRONT, GL_SHININESS, shininess);
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    glEnable(GL_DEPTH_TEST);
}
```

3. Evin Oluşturulması

Evin oluşturulması için makeScene() fonksiyonu kullanıldı. Bu fonksiyonda öncelikle kameranın konumu belirlendi. Ardından glTranslatef, glRotatef ve glScalef fonksiyonları ile evin çeşitli hareketleri yapabilme yeteneği eklendi.

```
void makeScene() {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glMatrixMode(GL_MODELVIEW);

    glLoadIdentity();
    glRotatef(0.0f, 0.0f, 1.0f, 0.0f);
    glTranslatef(-1.0f, -1.5f, -2.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(1.0f, 1.0f, 0.0f);

    glTranslatef
        (_translate_angle_x,_translate_angle_y,_translate_angle_z);
    glRotatef(_rotate_angle_z, 0.0f, 0.0f, 1.0f);
    glRotatef(_rotate_angle_y, 0.0f, 1.0f, 0.0f);
    glRotatef(_rotate_angle_x, 1.0f, 0.0f, 0.0f);
    glScalef(_scale_angle_x, _scale_angle_y, _scale_angle_z);

    glColor3f(1.0, 0.25, 1.0);
    glColor3f (.5, 0.5, .25);
    glClear (GL_COLOR_BUFFER_BIT);
```

Hareket yapabilme yeteneğinin ardından her parça için glColor3f fonksiyonu ile renk, glNormal3f fonksiyonu ile doğrultu, glVertex3f fonksiyonu ile kenar ölçüleri ayarlanarak evin tüm parçaları çizildi.

```
/* Front Rectangle */
glBegin(GL_POLYGON);
glColor3f (.2, .75, .55);
glNormal3f(0.0f, 0.0f, -1.0f);
glVertex3f (0.1, 0.1, 0.0);
glVertex3f (1, 0.1, 0.0);
glVertex3f (1, 0.6, 0.0);
glVertex3f (0.1, 0.6, 0.0);
glEnd();


/* Rear Rectangle */
glBegin(GL_POLYGON);
glColor3f (.2, .75, .55);
glNormal3f(0.0f, 0.0f, 1.0f);
glVertex3f (0.1, 0.1, 0.5);
glVertex3f (1, 0.1, 0.5);
glVertex3f (1, 0.6, 0.5);
glVertex3f (0.1, 0.6, 0.5);
glEnd();
```

.

.

Son olarak glutDisplayFunc(…) fonksiyonuna parametre olarak makeScene(…) fonksiyon göstericisi verilerek geçerli pencerede evin çizilmesi sağlandı.

4. Klavye Girdilerinin Alınması ve Görüntünün Güncellenmesi

Klavye girdilerinin alınması ve evin durumunun güncellenmesi amacıyla updateWithKeypress(…) fonksiyonu yazıldı. Bu fonksiyon içinde w,a,s,d,q,a harfleri evi döndürmek için kullanıldı. t,f,g,h,r,v tuşları çevirmek için kullanıldı. 1 ve 2 tuşları ise ölçeklendirme için kullanıldı. Fonksiyon girdi olarak gelen klavye sinyaline göre global _rotate_angle_x/y/z, _scale_angle_x/y/z ve _translate_x/y/z değişkenleri üzerinde ekleme veya çıkarma yaparak manipülasyon sağladı. Sağlanan değişim ise glutPostRedisplay() fonksiyonu ile güncellendi.

```
void updateWithKeypress(unsigned char key, int x, int y) {
    if(key=='d'){
        _rotate_angle_y += 4.0;
    }
    else if(key=='a'){
        _rotate_angle_y -= 4.0;
    }
    else if(key=='s'){
        _rotate_angle_x += 4.0;
    }
    else if(key=='w'){
        _rotate_angle_x -= 4.0;
    }
    else if(key=='z'){
        _rotate_angle_z += 4.0;
    }
    else if(key=='q'){
        _rotate_angle_z -= 4.0;
    }
    else if(key=='1'){
        _scale_angle_x += 0.1;
        _scale_angle_y += 0.1;
        _scale_angle_z += 0.1;
    }
    else if(key=='2'){
        _scale_angle_x -= 0.1;
        _scale_angle_y -= 0.1;
        _scale_angle_z -= 0.1;
    }
    else if(key=='h'){
        _translate_angle_x += 0.1f;
    }
    else if(key=='f'){
        _translate_angle_x -= 0.1f;
    }
    else if(key=='t'){
        _translate_angle_y += 0.1f;
    }
    else if(key=='g'){
        _translate_angle_y -= 0.1f;
    }
    else if(key=='r'){
        _translate_angle_z += 0.1f;
    }
    else if(key=='v'){
        _translate_angle_z -= 0.1f;
    }
    else if(key==27){
        exit(0);
    }
    glutPostRedisplay();
}
```

Show the Debug area

Yukarıda bahsedilen updateWithKeypress(...) fonksiyonu glutKeyboardFunc(…) içinde çalıştırılarak sağlanan girdiler updateWithKeypress(…) fonksiyonu ile işlendi.

5. Görüntünün Tekrar Boyutlandırılması ve Perspektifi

Görüntünün pencerenin geçerli boyutuna göre tekrar boyutlandırılması için resizeFunction(…) fonksiyonu yazıldı. Bu fonksiyonun içinde glViewport(…) fonksiyonu kullanılarak görüntü alanı ayarlandı. Ardından glMatrixMode(…) içindeki GL_PROJECTION bayrağı ile geçerli matris modu belirtilip, glLoadIdentity(…) fonksiyonu ile mevcut matris kimlik matrisi ile değiştirildi. En son ise gluPerspective(…) fonksiyonu ile perspektif projeksiyon matrisi oluşturuldu.

```
void resizeFunction(int w, int h) {
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(45.0, (double) w / (double) h, 1.0, 200.0);
}
```

Yukarıda bahsedilen resizeFunction(…) fonksiyonu glutReshapeFunc(..) fonksiyonu içinde çalıştırılarak tekrar boyutlandırma sağlandı.

6. Olay İşleme Döngüsü ve Hepsinin Birleştirilmesi

Yukarıda belirtilen tüm işlemleri yapabilmek amacıyla ana fonksiyonda tüm diğer fonksiyonlar çalıştırıldı. En sonda ise olay işleme fonksiyonun başlaması için glutMainLoop() fonksiyonu çalıştırıldı.

```
int main(int argc, char * * argv) {
    glutInit( & argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(1280, 720);
    glutCreateWindow("Computer Graphics HW1 - 3d Home");
    initialize();
    glutDisplayFunc(makeScene);
    glutKeyboardFunc(updateWithKeypress);
    glutReshapeFunc(resizeFunction);
    glutMainLoop();
    return 0;
}
```