

CSE 461

ASSIGNMENT 1

RAPOR

Video linki:

https://www.youtube.com/watch?v=m3_jPosaDG4

Miray Yıldız

161044023

Bu ödevde, OpenGL kütüphanesiyle 3D bir ev oluşturdum ve bu evin üzerinde ödevde belirtildiği gibi translation, rotatin and scaling transformations'ları uyguladım. Ödevde kullandığım fonksiyonları ve aşamaları açıklamam gerekirse;

```
void drawScene() {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glClearColor(1.0,1.0,0.7,1.0);          /* Background color */

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}
```

Yukarıda görünen kod kesitinde, drawScene fonksiyonuyla evimi ekrana çiziyorum. Burada bazı özel fonksiyonlar kullandım, şöyle ki:

- glClear ile renk bufferını temizledi.
- glClearColor ile arka plan rengini ayarladım. Renk kodlarını şu site yardımıyla buldum : <https://www.tug.org/pracjourn/2007-4/walden/color.pdf>
- glMatrixMode fonksiyonunu son matrisi seçmek için kullandım , planlanan dönüşümün belirlenmesine başlamak için GL_PROJECTION parametresini çağırdım.
- glLoadIdentity() fonksiyonuyla, projeksiyon moduna her girdiğimde, matrisin identity matrisine sıfırlanmasını ve böylece yeni görüntüleme parametrelerinin bir öncekiyle birleştirilmemesini sağladım.

```
glTranslatef(1.0f, 1.0f, 0.0f);
glRotatef(angle, 0.0f, 1.0f, 0.0f);
glRotatef(angle2, 1.0f, 0.0f, 0.0f);
glRotatef(angle3, 0.0f, 0.0f, 1.0f);
```

```
glScalef(0.7f, 0.7f, 0.7f);
```

- glTranslatef() fonksiyonu ile öteleme olayını yönettim.
- glRotatef() ile döndürme işlemlerini yönettim.
- glScalef() ile büyütme ve küçültme işlemlerini yönettim.

```
/* front of the house*/

glClear (GL_COLOR_BUFFER_BIT);

glColor3f (0.8, 0.7, 1.0);
glBegin(GL_POLYGON);
glVertex3f (0.1, 0.1, 0.0);
glVertex3f (1.0, 0.1, 0.0);
glVertex3f (1.0, 0.575, 0.0);
glVertex3f (0.1, 0.575, 0.0);
glEnd();

/* Back of the house */
```

Evi çizme aşamasında, glClear ile renk bufferını temizledim ve glColor3f ile rengimi belirledim. glBegin fonksiyonu içinde olan GL_POLYGON parametresiyle bize bir obje çiziminin başladığını ifade eder. Ben de burada bir obje çiziminin başladığını

söyleyerek GL_POLYGON parametresiyle verilen noktaları doğrularla birleştir ve içlerini renklendir komutunu vermiş oldum. glVertex3f fonksiyonları ile yüzeyin dört köşe noktası için koordinat değerlerini belirttim.

```
320     glEnd();
321     glFlush ();
```

Tüm parçaları oluşturduktan sonra, yukarıdaki resimde de görüldüğü gibi glBegin() ile başlayan çizim işlemini glEnd() ile bitirdim. glFlush() ile ise ekrana yazdırdım.

Main fonksiyonunda,

```
327 /* Main fuction */
328 int main(int argc, char * * argv) {
329     glutInit( & argc, argv);
330     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
331     glutInitWindowSize(1000, 800);
332
333     glutCreateWindow("My House");
334
335     initRendering();
336
337     glutDisplayFunc(drawScene);
338
339     glutKeyboardFunc(Choices);
340
341     glutReshapeFunc(handleResize);
342
343     glutMainLoop();
344
345     return 0;
346 }
```

- glutInit() fonksiyonu ile GLUT'ı ilkledim. Tüm GLUT işlemlerinden önce GLUT ilklenmelidir.
- glutInitDisplayMode da bir başlatım modudur.
- glutInitWindowSize ile açılacak olan pencerenin boyutlarını belirledim.
- glutCreateWindow ile penceremi isimlendirdim.
- Pencerenin içinde gösterilecek olan şeyleri, glutDisplayFunc ile çağırdım. Bu fonksiyonun parametresi ekrana çizdirme yaptığım fonksiyon.
- glutKeyboardFunc fonksiyonunun parametresi switch-case ile tuşları yönettiğim fonksiyon. Klavyeden bir tuşa basıldığında bu fonksiyon çağırılır.
- glutReshapeFunc ile yeni yükseklik ve genişlik için olanak sağladım.
- glutMainLoop sayesinde programım kendisine gelecek eventleri dinlemeye başladı ve eğer tanımlıysa gelen olaya göre tanımlanmış bir callback'i çalıştırdı.