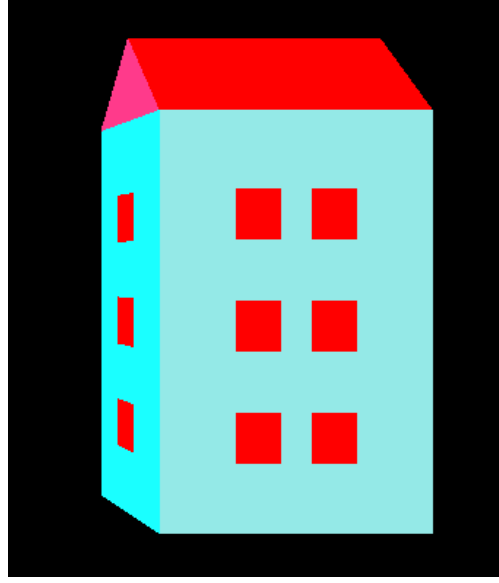


CSE 461 ASSIGNMENT #01

Gökhan Has – 161044067

Ödev için basit bir şekilde aşağıdaki ev yapısı yapılmıştır. Birbiri ile simetrik 6 tane dörtgen (2 adet kenar dörtgenleri, 2 adet geniş dörtgenler ve 2 adet çatıda bulunan dörtgen) ile, 2 tane üçgen (çatıda) kullanılmıştır.



Ödevde istenildiği üzere OpenGL ve GLUT kütüphaneleri kullanılmıştır. İlk olarak glutInitWindowSize ile pencere boyutu oluşturulmuştur. glutCreateWindow fonksiyonu yardımıyla pencere üzerindeki yazı metni verilmiştir. Pencere hazırlama fonksiyonları çağırıldıktan sonra evin çizimin yapıldığı draw3DHouse fonksiyonu, glutDisplayFunc'a parametre olarak verilir. Bu işlemten sonra keyboardActions fonksiyonu, glutKeyboardFunc'a parametre olarak verilir. Bu fonksiyon klavyede tuşlara basıldığında oluşacak aksiyonların bulunduğu fonksiyondur. Pencere boyutu değiştiğinde veya güncellendiğinde aksiyonların bulunduğu fonksiyon resize fonksiyonudur ve glutReshapeFunc'a parametre olarak verilmiştir. Sonsuz döngüde uygulamanın çalışması ve aynı olayların tekrarlanması glutMainLoop fonksiyonu çağırıldığında olmaktadır.

Rotate, Scale ve Translate işlemleri için gerekli olan toplamda 9 değişken vardır. Bu değişkenlerin tipi GLfloat tipindedir ve X, Y, Z eksen olmak üzere her işlem için 3 adet değişken mevcuttur.

```
GLfloat x_rotate = 0.0f;  
GLfloat y_rotate = 0.0f;  
GLfloat z_rotate = 0.0f;  
  
GLfloat x_scale = 1.0f;  
GLfloat y_scale = 1.0f;  
GLfloat z_scale = 1.0f;  
  
GLfloat x_translate = 0.0f;  
GLfloat y_translate = 0.0f;  
GLfloat z_translate = 0.0f;
```

keyboardActions fonksiyonunu parametreleri GLUT kütüphanesinde önceden tanımlanmıştır. Fonksiyonun ismi değişse bile parametrelerini sayısı ve tipi değişemez. glutKeyboardFunc, bu parametrelili fonksiyon alacak şekilde yazılmıştır. keyboardActions fonksiyonda klavyeden değerler basıldığında transform, scale ve rotate işlemleri yapılır. **BU TUŞLARA CAPS LOCK AÇIK İKEN BASILDIĞINDA ÇALIŞMAZ, TEPKİ VERMEZ. CAPS LOCK KAPALI OLMALIDIR.** Bu tuşlar ve işlemleri sırasıyla aşağıda tanımlanmıştır :

w	: x ekseninde negatif 1.0'lık bir dönme (rotate) yapar
s	: x ekseninde pozitif 1.0'lık bir dönme yapar
a	: y ekseninde negatif 1.0'lık bir dönme yapar
d	: y ekseninde pozitif 1.0'lık bir dönme yapar
q	: z ekseninde pozitif 1.0'lık bir dönme yapar
e	: z ekseninde negatif 1.0'lık bir dönme yapar
z	: x ekseninde pozitif 0.1'lik bir öteleme (translate) yapar (sağa öteler)
x	: x ekseninde negatif 0.1'lik bir öteleme yapar (sola öteler)
c	: y ekseninde pozitif 0.1'lik bir öteleme yapar (yukarı öteler)
v	: y ekseninde negatif 0.1'lik bir öteleme yapar (aşağı öteler)
b	: z ekseninde pozitif 0.1'lik bir öteleme yapar (kullanıcıya yakın öteler)
n	: z ekseninde negatif 0.1'lik bir öteleme yapar (kullanıcıya uzak öteler)
f	: Tüm eksenlerde pozitif 0.1'lik bir ölçeklendirme (scale) yapar (büyültür)
g	: Tüm eksenlerde negatif 0.1'lik bir ölçeklendirme yapar (küçültür)
esc	: Programı kapatır

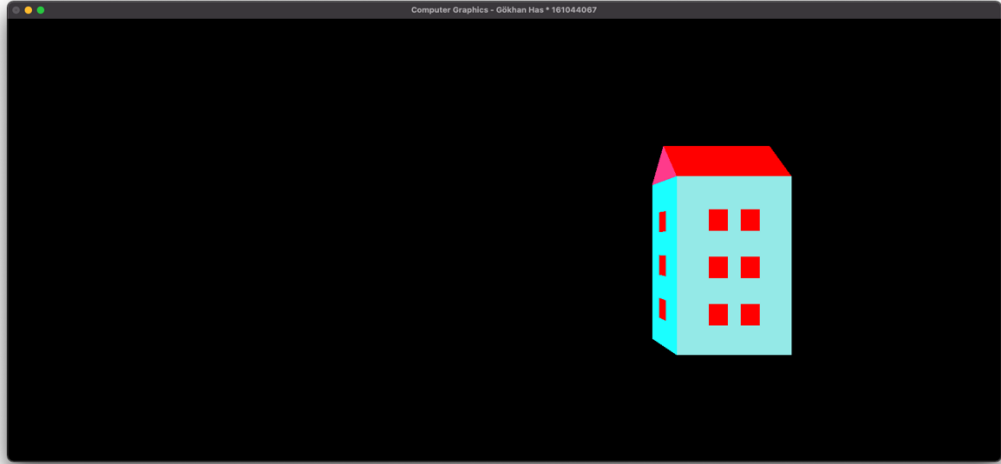
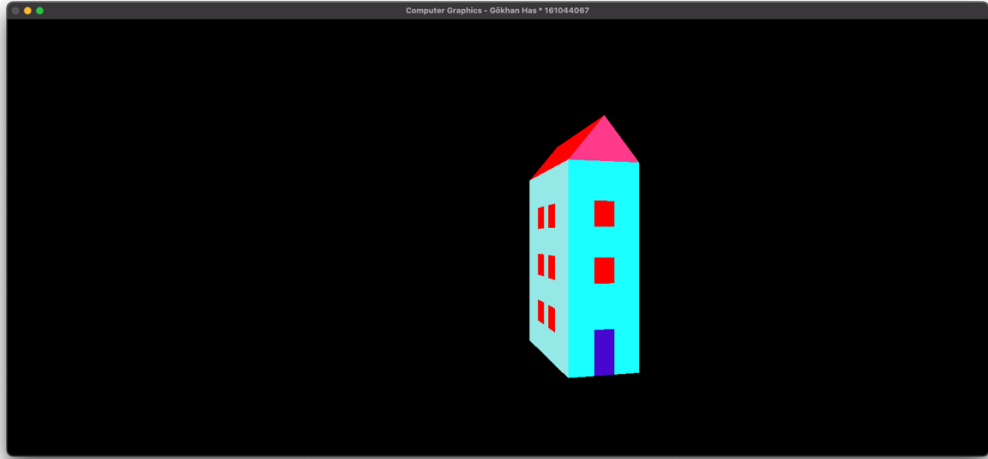
Bu tuşlarla evin nasıl değiştiği **daha detaylı** bir biçimde videoda anlatılmıştır. Daha sonra glutPostRedisplay fonksiyonu çağrılır. İlgili değişkenlerin değerleri güncellendikten sonra tekrarda çizimlerin güncellenmesi için bu işlem yapılır.

resize fonksiyonu pencere boyutu değiştiğinde veya güncellendiğinde evin boyutunda aynı orana göre değişmesini sağlar. İçerisinde ilk olarak glViewport fonksiyonu çağrılır. Bu fonksiyonun ilk iki parametresi x ve y'dir. Bu koordinatları pencere cinsinden with ve height'a göre dönüştürür. Görüntü alanı genişliği ve yüksekliği, uygulamaya bağlı olan bir aralığa sabitlenir. Sonra glMatrixMode fonksiyonu ile sonraki matriks işlemlerini projeksiyon matriksi yığınınına uygular. glLoadIdentity ile ise Özdeşlik matrisini, varsayılan durumuna geri döndürülür. Başlangıç noktasından yeniden başlamak istenildiğinde kullanılır. Son olarak gluPerspective fonksiyonu ile işlem tamamlanır. Bu fonksiyon kullanıcının evi nasıl görmesini ayarlayan fonksiyondur. Yeniden yükleme işlemi yapıldığında aynı oranda kullanıcının görmesini sağlar. Örneğin ilk parametresi 90 olsaydı ev daha uzakta olacaktı. İkinci parametres, orandır. Son iki parametre ise z ekseninde yakınlaştırma, uzaklaştırma ve yakınlaştırıp uzaklaştığında görüntünün terse doğru gitmesi ile ilgilidir.

draw3DHouse fonksiyonu ev bileşenlerinin çizildiği fonksiyondur. Her çizimden sonra glClear fonksiyonu çağrılmaz ise ekran siyah veya renklerin arasında siyah kısımlarla kalmaktadır. Önceki pikseller temizlenmelidir. Sonraki matriks işlemlerini modelview matriks yığınınına uygular (glMatrixMode). glLoadIdentity ile Özdeşlik matrisini, varsayılan durumuna geri döndürür. Başlangıç noktasından yeniden başlamak istenildiğinde kullanılır. İlk glTranslatef fonksiyonu evin pencerede ortalarında gözükmesi için sahip olması gereken eksen değerleridir. 3 işlem için tanımlanan 9 değişkenin hangi özellikler için kullanılacağı bildirilir. Daha sonra ev çizimi GL_POLYGON ve GL_TRIANGLES kullanılarak yapılır. Bu tiplerde sadece

koordinat ve bazen renk deęiřtirilerek çizme işlemi yapılır. glColor3f ile renk verilir. glBegin ile çizilecek objenin tipi belirlenir (polygon, triangle). glVertex3f ile noktalar verilir. Son olarak glEnd fonksiyonu ile o andaki çizilecek objenin çizimi bitirilir.

Objelerin çizim işlemleri bitirildikten sonra glFlush ile Arabelleęin (buffer) hepsi boşaltılır ve verilen tüm komutların gerçek işleme motoru tarafından kabul edildięi zamana kadar hızlı yürütölmesi saęlanır. glPopMatrix ile geçerli matriks yığınının bir ařaęı iter ve geçerli matrisi çoęaltır. Yani, bir glPushMatrix çağrısından sonra yığının üstündeki matris, altındaki matris ile aynıdır. Son olarak glutSwapBuffers fonksiyonu kullanılır. Bilgisayara geçerli çerçeve ile işin bittięini bildiren fonksiyondur. Bu çerçevenin görüntölenebilmesi ve bir sonraki üzerinde çalışmaya başlayabilmek için arabelleklerin deęiřtirilmesi gerekir.



Gökhan Has - 161044067