

DÖNÜŞÜMLERİ PROGRAMLAMA

Dönüşümler iki farklı yöntem ile belirlenebilir:

❑ Matrisleri belirleyerek

`glLoadMatrix, glMultMatrix`

❑ İşlemi belirleyerek

`glRotate, glOrtho`

❑ Programcı gerçek matrisin ne olduğunu bilmek zorunda değildir.

❑ Kameranın pozisyonu ve cisimlerin 3D geometrisi ile ilgili çizme, görüntüleme, konumlandırma ve yönlendirme işlemlerinden önce matrislerin hangi işlem için düzenleneceği belirtilmelidir:

`glMatrixMode(GL_PROJECTION)`

❑ Dönüşümler ilgili sırada bir araya getirilir. Artarda yazılan dönüşümler son yazılandan ilk yazılana doğru gerçekleştirilir.

Matris İşlemleri

❑ Matris yığınının (stack) belirleme

`glMatrixMode(GL_MODELVIEW veya GL_PROJECTION)`

❑ Diğer Matris veya Yığın İşlemleri

`glLoadIdentity() glPushMatrix()`

`glPopMatrix()`

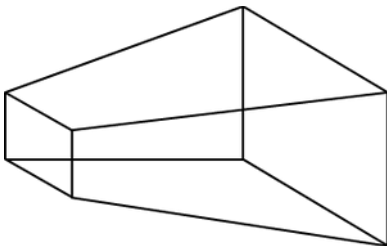
❑ Viewport

❑ Genellikle pencere genişliği ile aynıdır

❑ Eğer viewport genişlik/yükseklik oranı projeksiyondakinden farklı ise nesneler olduklarından farklı biçimde görünebilirler (daha geniş, daha yüksek, ...)

`glViewport(x, y, genişlik, yükseklik)`

Projeksiyon Dönüşümü



❑ Perspektif izdüşümde (projeksiyon) görünüm uzayı frustum adı verilen kesik piramit gibidir.

❑ Perspektif izdüşüm

`gluPerspective(fovy, aspect, zNear, zFar)`

`glFrustum(left, right, bottom, top, zNear, zFar)`

Orthografik paralel izdüşüm

`glOrtho(left, right, bottom, top, zNear, zFar)`

`gluOrtho2D(left, right, bottom, top)`

İzdüşüm Dönüşümlerini Uygulamak

Orthografik izdüşüm için tipik kullanım örneği:

`glMatrixMode(GL_PROJECTION);`

`glLoadIdentity();`

`glOrtho(left, right, bottom, top, zNear, zFar);`

Dönüşümleri Görmek

Kamerayı konumlandırma

Kamera ayağını aç ve kamerayı yerleştir

Bir sahnede hareket etmek için

Görüntüleme dönüşümünü değiştir ve sahneyi tekrar çiz

`gluLookAt(eyex, eyey, eyez,`

`aimx, aimy, aimz,`

`upx, upy, upz)`

Yukarı (up) vektör kameranın üstünün neresi olduğunu belirler.

?

İzdüşüm Uygulaması



Konumlandırma Dönüşümlerini Uygulamak

❓ Nesneyi taşımak:

```
glTranslate{fd}( x, y, z )
```

❓ Nesneyi bir eksen etrafında döndürmek:

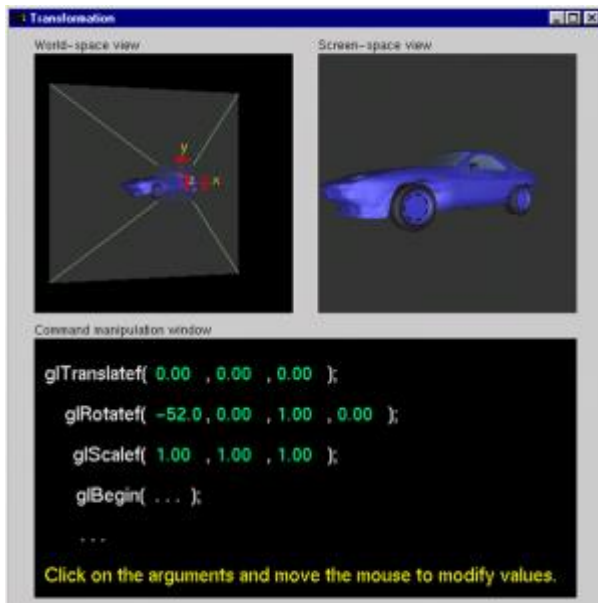
```
glRotate{fd}( açı, x, y, z )
```

❓ Açı derece cinsinden girilir

❓ Nesneyi büyötmek/küçöltmek, genişletmek/daraltmak ve yansımasını oluşturmak:

```
glScale{fd}( x, y, z )
```

Dönüşüm Uygulaması



Kamera yeri ve konumlandırma arasındaki ilişki

❓Kamerayı belirli bir doğrultuda taşıma ile görüntölenen dünyadaki tüm nesneleri tam tersi olan doğrultuda taşıma arasında bir fark yoktur.

❓Görüntüleme dönüşümleri birçok konumlandırma dönüşümüne eşittir.

❓gluLookAt() komutunun yaptığı işi birçok glTranslate ve glRotate kombinasyonu ile yapabilirsiniz.

Animasyon ve Derinlik Tamponu

Animasyon :

❑Çift renk tamponu kullanımı isteği yapılır

```
glutInitDisplayMode( GLUT_RGB | GLUT_DOUBLE );
```

❑Renk tamponu temizlenir

```
glClear( GL_COLOR_BUFFER_BIT );
```

❑Sahne çizimi yapılır

❑Ön ve arka tamponlar yer değiştirilir

```
glutSwapBuffers();
```

❑2, 3 ve 4. adımlar tekrar edilerek animasyon sağlanır

```
glutIdleFunc() callback kullanılır
```

Derinlik Tamponu:

❑Derinlik tamponu kullanımı isteği yapılır

```
glutInitDisplayMode( GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH );
```

❑Özellik aktif hale getirilir

```
glEnable( GL_DEPTH_TEST );
```

❑Renk ve derinlik tamponları temizlenir

```
glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );
```

❑Sahne çizilir

❑Renk tamponları yer değiştirilir

Kaynak: <http://altanmesut.trakya.edu.tr/>