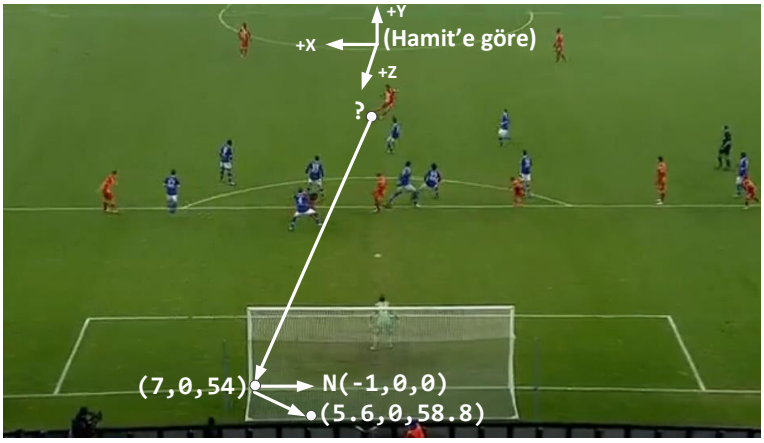
Karadeniz Teknik Üniversitesi
Bilgisayar Mühendisliği Bölümü
Öğr.Gör. Ömer ÇAKIR

BIL320 Bilgisayar Grafikleri I
Bütünleme Sınavı, 18.06.2013, 17:00
Süre : **75** Dakika

CEVAPLAR

1. Şampiyonlar ligi son 16 turu rövanş maçında Galatasaray'ın ilk golünü Selçuk'un pasına uzaktan mükemmel bir vuruşla Hamit atmıştı. Top (Hamit'e göre) sağ kale direğinden sekip ağlarla buluşmuştu. Topun ağlarla buluştuğu nokta **(5.6, 0, 58.8),** kale direğinden sektiği nokta **(7,0,54),** direğin normali **(-1,0,0),** şutu çektiği noktanın kale direğinden sektiği noktaya uzaklığı da **25** metre olsun. Hamit'in şutu hangi noktadan çektiğini <u>ışın izleme ile</u> hesaplayınız. **(20P)**



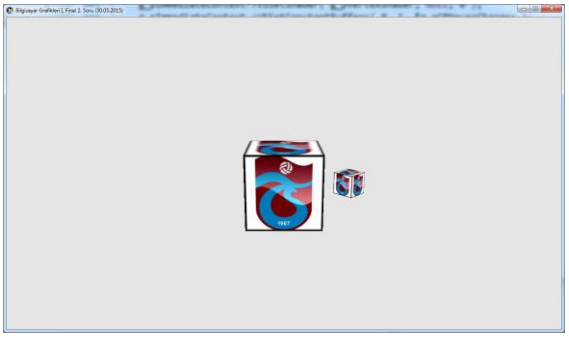
R_d = normalize((7,0,54)-(5.6,0,58.8))
= (0.28,0,-0.96)

$R_{reflected}$ = (-0.28, 0, -0.96)

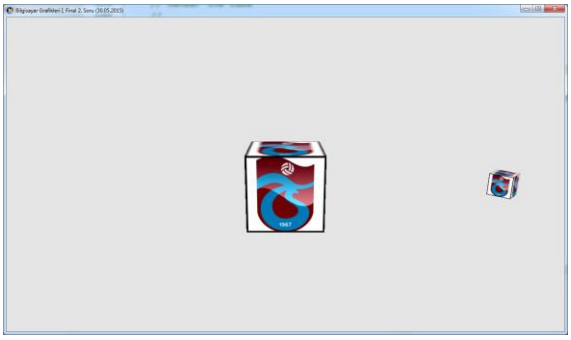
$Hamit_{ayak}$ = (7,0,54) + 25*(-0.28,0,-0.96)
= (0,0,30)

```
mRotate = XMMatrixRotationY(XM_PIDIV4); //45° CW
mTranslate = XMMatrixTranslation(6.0f,0.0f,0.0f);
mScale = XMMatrixScaling(0.3f,0.3f,0.3f);
```

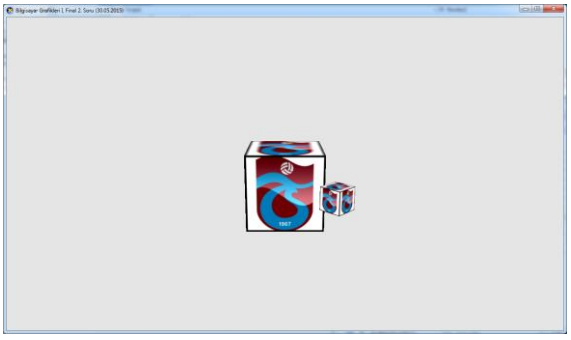
2. Aşağıda ilgili şeklin altına, büyük küpe ait **g_World** matrisi yukarıdaki matrisler ile setlenerek çizilen küçük küp için **g_World** matrislerini yazınız. **(20P)**



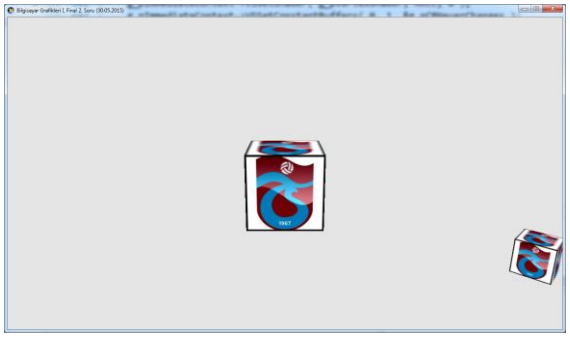
```
g_World = mRotate * mTranslate * mScale ;
```



```
g_World = mRotate * mScale * mTranslate ;
g_World = mScale * mRotate * mTranslate ;
```

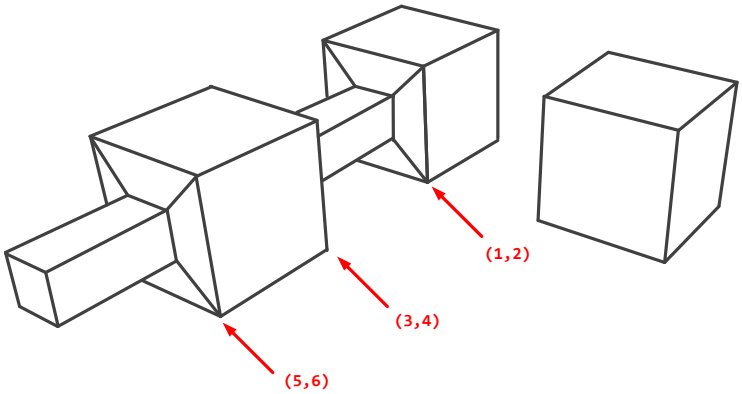


```
g_World = mTranslate * mScale * mRotate ;
g_World = mTranslate * mRotate * mScale ;
```



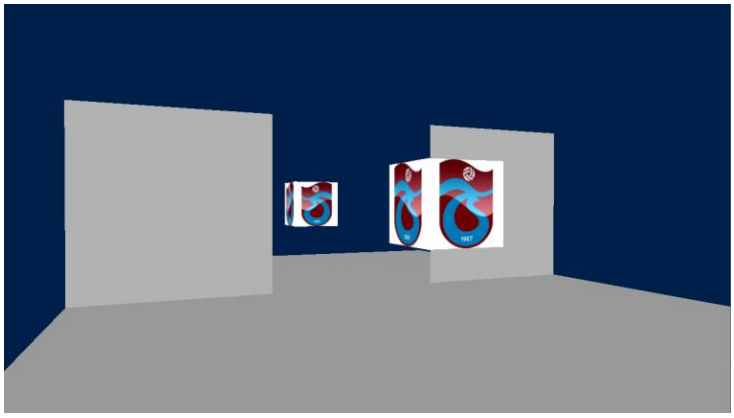
```
g_World = mScale * mTranslate * mRotate ;
```

Arka sayfaya geçiniz →



3. Yandaki şekilde, sağdaki küpün MAYA ortamında Extrude, Move, Scale ve Rotate toolları ile soldaki hale getirilmesi için kaç kez Extrude yapmak gerekir? Extrude yapılan yüzeyleri oklarla soldaki şekil üzerinde gösteriniz. **(30P)**

6 kez Extrude yapmak gerekir.



4. DirectX'te Stencil buffer ile aynasallığın nasıl gerçeklendiğine dair aşağıda verilen kod ile yandaki şekil render edilmiştir. Boş satırlara gerekli kodları yazınız. **(30P)**

Not → Aşağıdaki kodun sayfaya sığması için orjinalindeki bazı satırlar silinmiştir.

```
D3D11_DEPTH_STENCIL_DESC mirrorDesc;
mirrorDesc.FrontFace.StencilPassOp                = D3D11_STENCIL_OP_REPLACE;

mirrorDesc.FrontFace.StencilFunc                  = D3D11_COMPARISON_ALWAYS;

D3D11_DEPTH_STENCIL_DESC drawReflectionDesc;
drawReflectionDesc.FrontFace.StencilPassOp        = D3D11_STENCIL_OP_KEEP;

drawReflectionDesc.FrontFace.StencilFunc          = D3D11_COMPARISON_EQUAL;

void Render(float dt)
{
      // Render Box to Back Buffer
      g_pImmediateContext->DrawIndexed( 36, 0, 0 );

      // Render Ground to Back Buffer
      g_pImmediateContext->Draw( 6, 0 );

      // Render Wall on the Left Side of the Mirror to Back Buffer
      g_pImmediateContext->Draw( 6, 6 );

      // Render Wall on the Right Side of the Mirror to Back Buffer
      g_pImmediateContext->Draw( 6, 18 );

      // Render Mirror to Stencil Buffer and Back Buffer
      g_pImmediateContext->OMSetDepthStencilState(MarkMirrorDSS, 61);
      g_pImmediateContext->Draw( 6, 12 );

      // Calculate Reflection of the Box
      XMVECTOR mirrorPlane = XMVectorSet(0.0f, 0.0f, 1.0f, -6.0f); // z = 6 plane
      XMMATRIX R = XMMatrixReflect(mirrorPlane);
      g_World_Box = g_World_Box * R;

      // Render Reflection of Box
      g_pImmediateContext->OMSetDepthStencilState(DrawReflectionDSS, 61);
      g_pImmediateContext->DrawIndexed( 36, 0, 0 );
}
```