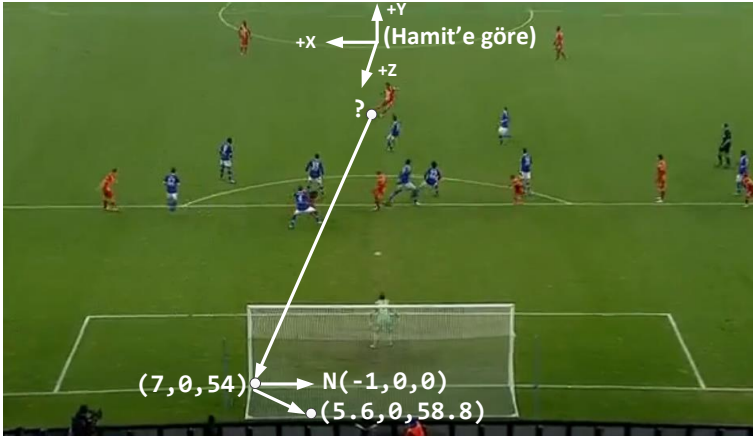




1. Cep telefonlarının, hesap makinesi, saate bakmak gibi herhangi bir amaçla kullanılması yasaktır. Telefon **kapalı** ve **cepte** olmalıdır.
2. **Sınavın başında** öğrenciler anlamadıkları noktaları **sesli olarak sorup**, Hoca cevapladıktan sonra **sınav boyunca soru sormak yasaktır**.

NUMARA :	AD SOYAD :ÖĞRETİM	DEĞERLENDİRME :
----------	------------	----------------	-----------------

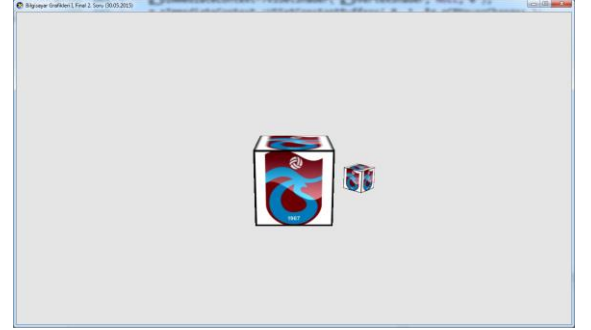
1. Şampiyonlar ligi son 16 turu rövanş maçında Galatasaray'ın ilk golünü Selçuk'un pasına uzaktan mükemmel bir vuruşla Hamit atmıştı. Top (Hamit'e göre) sağ kale direğinden sekip ağlarla buluşmuştu. Topun ağlarla buluştuğu nokta $(5.6, 0, 58.8)$, kale direğinden sektiği nokta $(7, 0, 54)$, direğin normali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



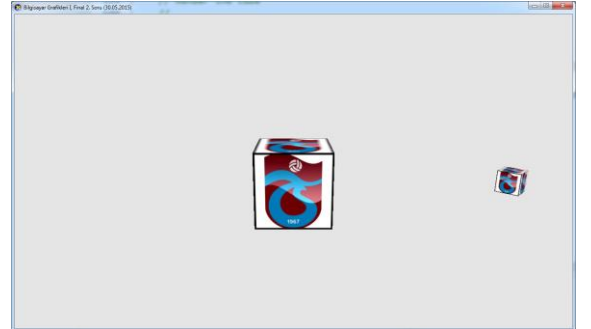
$mRotate = XMMatrixRotationY(XM_PIDIV4); //45^\circ CW$
 $mTranslate = XMMatrixTranslation(6.0f, 0.0f, 0.0f);$
 $mScale = XMMatrixScaling(0.3f, 0.3f, 0.3f);$

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

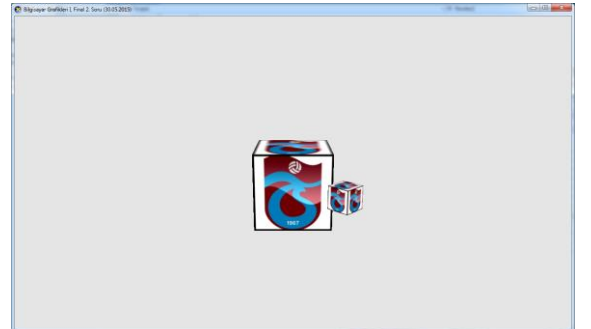
Not → Bakış noktası $(0, 3, -10)$ 'dadır. (20P)
Büyük küpün merkezi $(0, 0, 0)$ 'dadır, köşe noktaları $-1, +1$ değerleri ile setlenmiştir.
 $45^\circ CW$: Saat yönünde 45 derece dönme işlemi.



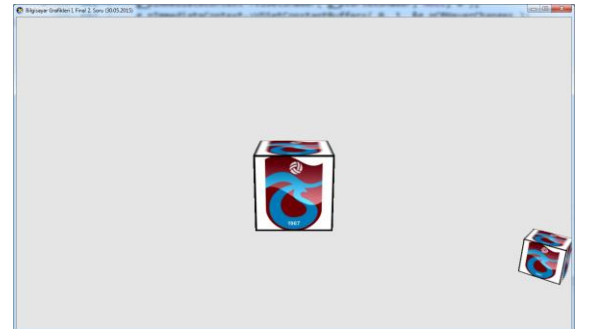
$g_World =$



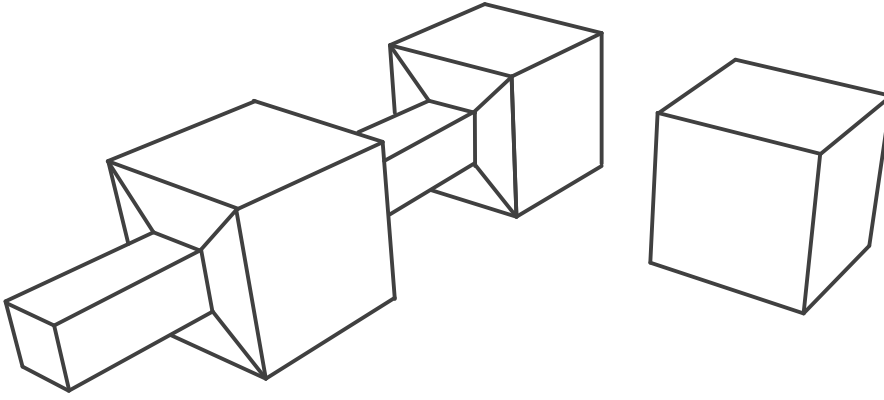
$g_World =$



$g_World =$

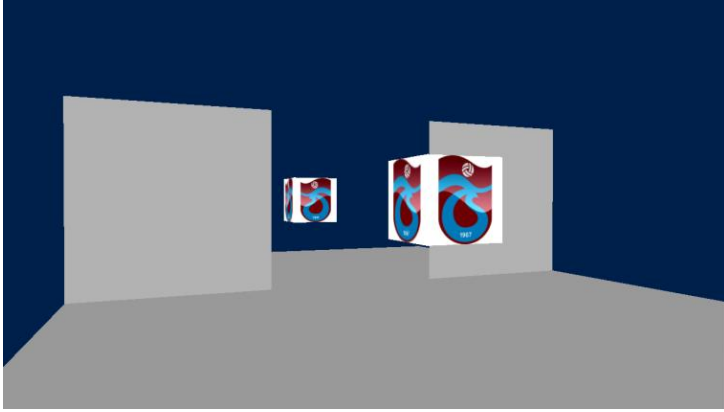


$g_World =$



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)

..... kez Extrude yapmak gerekir.



4. DirectX'te Stencil buffer ile aynasallığın nasıl gerçekleştirildiğine 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_.....;

mirrorDesc.FrontFace.StencilFunc             = D3D11_COMPARISON_.....;

D3D11_DEPTH_STENCIL_DESC drawReflectionDesc;
drawReflectionDesc.FrontFace.StencilPassOp   = D3D11_STENCIL_OP_KEEP;

drawReflectionDesc.FrontFace.StencilFunc     = D3D11_COMPARISON_.....;

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(..... , ..... );
    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(..... , ..... );
    g_pImmediateContext->DrawIndexed( 36, 0, 0 );
}

```