



CEVAPLAR

```
D3DXMatrixRotationY( &mRotateY, -t );  
D3DXMatrixTranslation( &mTranslate, -4, 3, 0 );  
D3DXMatrixScaling( &mScale, 0.2f, 0.2f, 0.2f );  
g_World = mScale * mRotateY * mTranslate * mRotateY;
```

1.

a) (10P)

g_World matrisi, (0.2f, 0.2f, 0.2f) oranında ölçeklenmiş cisme hem kendi etrafında hem de (0, 3, 0) noktası etrafında Y ekseninde 4 birim yarıçaplı dönme işlemi yaptırır.

b) (10P)

```
D3DXMatrixTranslation(&mTranslate, 0, 3, 4 );  
D3DXMatrixTranslation(&mTranslate, 0, 3, -4 );
```

2. (20P)

```
SimpleVertex vertices[] =  
{  
    { D3DXVECTOR3( 0, 2, 1), D3DXVECTOR2(1, 0) },  
    { D3DXVECTOR3( 1, 0, 1), D3DXVECTOR2(1, 1) },  
    { D3DXVECTOR3( 1, 0, -1), D3DXVECTOR2(0, 1) },  
    { D3DXVECTOR3( 0, 2, -1), D3DXVECTOR2(0, 0) },  
  
    { D3DXVECTOR3( 0, -2, 1), D3DXVECTOR2(1, 1) },  
    { D3DXVECTOR3( 0, -2, -1), D3DXVECTOR2(0, 1) },  
    { D3DXVECTOR3( 1, 0, -1), D3DXVECTOR2(0, 0) },  
    { D3DXVECTOR3( 1, 0, 1), D3DXVECTOR2(1, 0) }  
};  
  
DWORD indices[] =  
{  
    0,1,2,  
    0,2,3,  
  
    4,5,6,  
    4,6,7  
};
```

Alternatif 3 farklı çözüm daha vardır. Kaynak koddan bakabilirsiniz.

3. (20P)

```
for(int i=0; i<3; i++)  
{  
    toLight = normalize( vLightPos[i] - input.PosW );  
  
    dotEyeNorm = dot( toLight, input.Norm );  
    if( dotEyeNorm > 0.0F )  
        diffuseColor = dotEyeNorm * vLightColor[i];  
    else  
        diffuseColor = float3( 0.0F, 0.0F, 0.0F );  
  
    toEye = normalize( EyePos - input.PosW );  
    reflected = toLight - 2 *  
        dot(toLight, input.Norm) * input.Norm;  
  
    dotEyereflected = dot( toEye, reflected );  
    if( dotEyereflected > 0.0F )  
        specularColor = pow( dotEyereflected, 64.0F )  
            * vLightColor[i];  
    else  
        specularColor = float3( 0.0F, 0.0F, 0.0F );  
  
    finalColor.rgb += saturate( 0.5F * diffuseColor  
        + 0.5F * specularColor );  
}
```

Specular bileşen hesaplanırken toLight değil fromLight kullanılmalıdır. fromLight olarak -toLight alınabilir veya normalize(input.PosW-vLightPos[i]) şeklinde de tanımlanabilir.

4. (20P)

```
HRESULT InitDevice()  
{  
    g_pWorldVariable = g_pEffect->  
        GetVariableByName( "World" )->AsMatrix();  
}  
  
void Render()  
{  
    g_pWorldVariable->SetMatrix((float*)&g_World);  
}
```

5. (20P)

g_pd3DDevice, Draw() veya DrawIndexed() emri ile backbuffera çizim yapar. g_pSwapChain, Present() emri ile backbufferın içeriğini ekranda görüntüler.