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

BIL3008 Bilgisayar Grafikleri I
Final, 06.06.2017, 10:00, D1-2
Süre : **90** Dakika

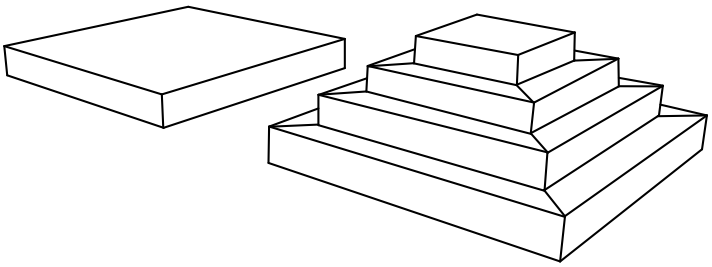
CEVAPLAR

1. Bakış noktası **(0,-4,91)** ve **(0,-12,97)** olduğunda **N(0,-0.8,0.6)** normaline sahip aşağıdaki üçgenin, arkayüz (backface) olup/olmadığını belirleyiniz. **(30P)**

U_0(0,60,180) U_1(-60,0,100) U_2(60,0,100)

$(0,-4,91)-(0,60,180)=(0,-64,-89)_{toEye\_1}$
$(0,-64,-89)*(0,-0.8,0.6)=-2.2$ (Backface)

$(0,-12,97)-(0,60,180)=(0,-72,-83)_{toEye\_2}$
$(0,-72,-83)*(0,-0.8,0.6)=+7.8$ (notBackface)



2. Yukarıda soldaki dikdörtgen prizmayı MAYA'da sağdaki piramide dönüştürmek için kaç kez Extrude yapmak gerekir? **(20P)**

6 kez Extrude yapmak gerekir.

3. DirectX12'de 1'den fazla PSO (pipeline state object) tanımlamayı gerektiren bir örnek veriniz. **(20P)**

Phong, Textured, Solid gibi 1'den fazla Pixel Shader olduğunda PSO'nun .PS (Pixel Shader) değişkeni ilgili Pixel Shader'a setlenerek farklı PSO'lar tanımlanır. Render() fonksiyonunda o anda çizilecek cisim Phong, Textured, Solid modlarından hangisinde çizilecekse o moda ait PSO SetPipelineState() ile setlenir.

Stencil buffer kullanılarak aynasal yansımada PSO'nun .DepthStencilState değişkeni farklı setlemeler içeren D3D12_DEPTH_STENCIL_DESC türden mirrorDSS ve reflectionsDSS değişkenlere setlenerek farklı PSO'lar tanımlanır.
mirrorDSS cismin back buffera çizildiği pixellerin stencil bufferdaki eşleniklerine 0-255 arası int değerin yazılmasını sağlayan setlemeleri içerir.
reflectionsDSS cismin yansımasının yalnızca stencil bufferda ayna için setlenen pixellere çizilmesini (back bufferda) sağlayan setlemeleri içerir.

```
mRotate30  = XMMatrixRotationY(XM_PI / 6);          // 30° CW
mRotate45  = XMMatrixRotationY(XM_PI / 4);          // 45° CW
mRotate60  = XMMatrixRotationY(XM_PI / 3);          // 60° CW
mTranslate = XMMatrixTranslation(4.0f, 0.0f, 0.0f);
mScale     = XMMatrixScaling(0.5f, 0.5f, 0.5f);

g_World = mRotate30 * mTranslate * mRotate60  * mScale; // 01
g_World = mRotate45 * mTranslate * mRotate45  * mScale; // 02
g_World = mRotate60 * mTranslate * mRotate30  * mScale; // 03
g_World = mRotate30 * mTranslate * mScale * mRotate60;  // 04
g_World = mRotate45 * mTranslate * mScale * mRotate45;  // 05
g_World = mRotate60 * mTranslate * mScale * mRotate30;  // 06
g_World = mRotate30 * mScale * mTranslate * mRotate60;  // 07
g_World = mRotate45 * mScale * mTranslate * mRotate45;  // 08
g_World = mRotate60 * mScale * mTranslate * mRotate30;  // 09
g_World = mScale * mRotate30 * mTranslate * mRotate60;  // 10
g_World = mScale * mRotate45 * mTranslate * mRotate45;  // 11
g_World = mScale * mRotate60 * mTranslate * mRotate30;  // 12
```

4. Küçük küpün **g_World** matris setlemelerinden eşdeğer olanların satır numarlarını kutucuklara yazınız. **(30P)**

Not: Bakış noktası **(0,4,-9)**'dadır. Büyük küpün merkezi **(0,0,0)** noktasındadır ve köşe noktaları **-1,+1** değerleri ile setlenmiştir. Dönme işlemleri saat yönündedir. (**C**lock**W**ise).

İpucu: Eşdeğer matrislerin hepsi ikişerli gruplar halindedir.

1	4	2	5	3	6
7	10	8	11	9	12