



CEVAPLAR

1. DirectX 12'de stencil buffer ile cisimlerin aynasal yansımaları çizilirken stencil bufferda **aynaya**
- ait piksellerin belli bir **int** değere setlenmesi
 - yansımaların yalnızca bu piksellere çizilmesi
- için sırasıyla hangi **StencilPassOp** ve **StencilFunc** setlemeleri yapılır? (20P)

Yanlış cevaptan 5P kırılacaktır!

(A)	StencilPassOp = D3D12_STENCIL_OP_REPLACE; StencilFunc = D3D12_COMPARISON_FUNC_EQUAL;
(B)	StencilPassOp = D3D12_STENCIL_OP_KEEP; StencilFunc = D3D12_COMPARISON_FUNC_ALWAYS;
(C)	StencilPassOp = D3D12_STENCIL_OP_REPLACE; StencilFunc = D3D12_COMPARISON_FUNC_ALWAYS;
(D)	StencilPassOp = D3D12_STENCIL_OP_KEEP; StencilFunc = D3D12_COMPARISON_FUNC_EQUAL;

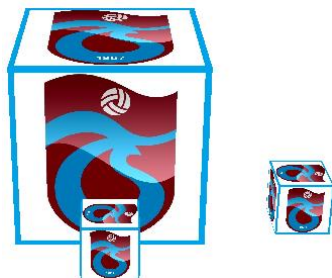
```
Rot45 = XMMatrixRotationY(XM_PI / 4); // 45 CW  
Trans = XMMatrixTranslation(5.0f, 0.0f, 0.0f);  
Scale = XMMatrixScaling(0.5f, 0.5f, 0.5f);
```

```
g_World_1 = Scale * Rot45 * Trans * Scale * Rot45;  
g_World_2 = Rot45 * Scale * Trans * Rot45 * Scale;  
g_World_3 = Scale * Rot45 * Trans * Rot45 * Scale;  
g_World_4 = Rot45 * Scale * Trans * Scale * Rot45;  
g_World_5 = Scale * Trans * Rot45 * Scale * Rot45;
```

2. Yukarıdaki transformasyon matrislerinden 4'ü eşdeğer, 1'i farklıdır. Farklı olan hangisidir? (20P)

Yanlış cevaptan 5P kırılacaktır!

- (A) g_World_1
(B) g_World_2
(C) g_World_3
(D) g_World_4
(E) g_World_5



```
struct Vertex  
{
```

```
    XMFLOAT3 position;  
    XMFLOAT4 color;  
};
```

```
Vertex vertices[] = { . . . }  
DWORD indices[] = { . . . }
```

3. Üçgenlerden oluşan bir küp yalnızca **Vertex** türünden köşe noktaları içeren vertex buffer ile çizildiğinde bufferın byte cinsinden boyu **X** olsun.

Hem vertex hem de index buffer kullanılarak çizildiğinde vertex ve index bufferların toplam boyu **Y** olsun. **X** ile **Y** arasındaki fark nedir? (20P)

Not → **FLOAT**: 4byte, **DWORD**: 4byte alınız.

İpucu → **X** ve **Y**'deki vertex buffer içerikleri farklıdır.

Yanlış cevaptan 5P kırılacaktır!

- A) 80 B) 160 C) 320 D) 640 E) 1280

$$X = 36 * 7 * 4 = 1008$$

$$Y = 8 * 7 * 4 + 36 * 4 = 368$$

$$1008 - 368 = 640$$

```
v -0.5 0.0 0.5
```

```
v 0.5 0.0 0.5
```

```
v -0.5 0.0 -0.5
```

```
v 0.5 0.0 -0.5
```

```
vt 0.0 0.0
```

```
vt 1.0 0.0
```

```
vt 0.0 1.0
```

```
vt 1.0 1.0
```

```
vn 0.0 1.0 0.0
```

```
vn 0.0 1.0 0.0
```

```
vn 0.0 1.0 0.0
```

```
vn 0.0 1.0 0.0
```

```
f 1/1/1 2/2/2 3/3/3
```

```
f 3/3/3 2/2/2 4/4/4
```

4. Yukarıdaki OBJ dosyasının STL hali nedir? (20P)

```
solid  
facet normal 0.0 1.0 0.0  
outer loop  
vertex -0.5 0.0 0.5  
vertex 0.5 0.0 0.5  
vertex -0.5 0.0 -0.5  
endloop  
endfacet  
facet normal 0.0 1.0 0.0  
outer loop  
vertex -0.5 0.0 -0.5  
vertex 0.5 0.0 0.5  
vertex 0.5 0.0 -0.5  
endloop  
endfacet  
endsolid
```

P0(-10, 10, 40)

P1(10, 10, 40)

P2(10, -10, 40)

P3(-10, -10, 40)

P4(10, 10, 60)

P5(-10, 10, 60)

P6(-10, -10, 60)

P7(10, -10, 60)

CW Rotation around Y-axis

$$\begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix}$$

5. Köşe noktalarının konumları yukarıda verilen bir küp (0,0,1) doğrultusu boyunca 40 birim ilerledikten sonra y ekseninde saat yönünde 90 derece dönüyor ve bu sefer (1,0,0) doğrultusu boyunca 60 birim ilerliyor. Köşe noktalarının yeni konumları ne olur? (20P)

(0,0,1) Boyunca 40br İlerleyince:

P0(-10, 10, 80)

P1(10, 10, 80)

P2(10, -10, 80)

P3(-10, -10, 80)

P4(10, 10, 100)

P5(-10, 10, 100)

P6(-10, -10, 100)

P7(10, -10, 100)

Center = (0,0,90)

P-Center'lar Hesaplandığında:

P0(-10, 10, -10)

P1(10, 10, -10)

P2(10, -10, -10)

P3(-10, -10, -10)

P4(10, 10, 10)

P5(-10, 10, 10)

P6(-10, -10, 10)

P7(10, -10, 10)

P'ler Rotasyon Matrisi ile Çarpıldığında:

P0(-10, 10, 10)

P1(-10, 10, -10)

P2(-10, -10, -10)

P3(-10, -10, 10)

P4(10, 10, -10)

P5(10, 10, 10)

P6(10, -10, 10)

P7(10, -10, -10)

P+Center'lar Hesaplandığında:

P0(-10, 10, 100)

P1(-10, 10, 80)

P2(-10, -10, 80)

P3(-10, -10, 100)

P4(10, 10, 80)

P5(10, 10, 100)

P6(10, -10, 100)

P7(10, -10, 80)

(1,0,0) Boyunca 60br İlerleyince:

P0(50, 10, 100)

P1(50, 10, 80)

P2(50, -10, 80)

P3(50, -10, 100)

P4(70, 10, 80)

P5(70, 10, 100)

P6(70, -10, 100)

P7(70, -10, 80)