

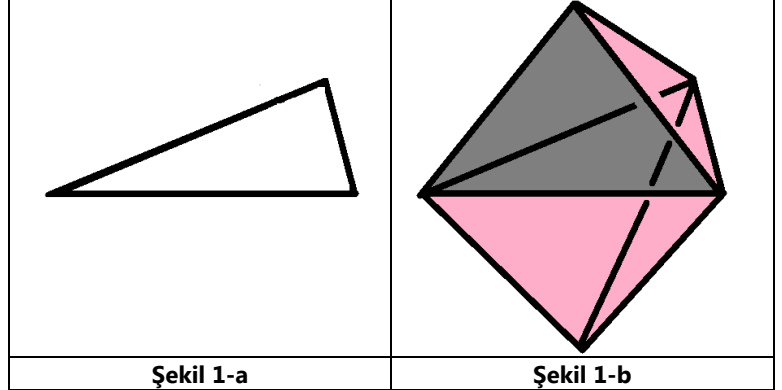


Sınavda Uyulması Gereken Kurallar

1. Cep telefonlarının, hesap makinesi, saate bakmak gibi herhangi bir amaçla kullanılması **yasaktır**. Telefon kapalı ve cepte olmalıdır.
2. Sorularda anlaşılmayan bir nokta varsa ilk **20 dakika** sorulabilir. Her öğrencinin **1** kez soru sorma hakkı vardır.

1. D3DXVECTOR3 vertices[] =
{
D3DXVECTOR3(-1.0f, 0.0f, 1.0f),
D3DXVECTOR3(1.0f, 0.0f, 1.0f),
D3DXVECTOR3(0.0f, 0.0f, -1.0f),
D3DXVECTOR3(0.0f, 1.0f, 0.0f),
D3DXVECTOR3(0.0f, -1.0f, 0.0f)
};

DWORD indices[] =
{
...
}



Yukarıdaki vertices[] isimli vertex bufferı kullanarak **Şekil 1-b** 'deki gibi bir düzgün altıyüzlü (hexahedron) oluşturmak için gerekli **6** üçgeni indices[] isimli index bufferda **saat yönünde** tanımlayınız. **(25 PUAN)**

Not→ **Şekil 1-b** 'de **7** üçgen görünüyor. Bunlardan tanımlayacağınız **6** üçgenden biri **Şekil 1-a** 'daki **olmamalıdır**.

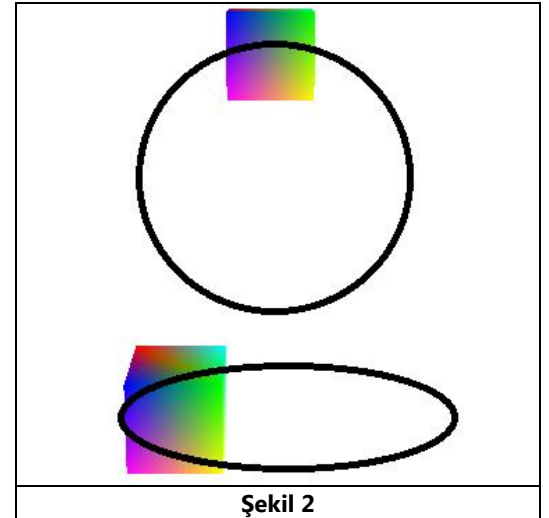
2. D3DXMatrixRotationY(&mRotateY, t);
D3DXMatrixRotationZ(&mRotateZ, t);
D3DXMatrixTranslation(&mTranslate1, ?, ?, ?);
D3DXMatrixTranslation(&mTranslate2, ?, ?, ?);
D3DXMatrixTranslation(&mTranslate3, ?, ?, ?);
D3DXMatrixTranslation(&mTranslate4, ?, ?, ?);

g_World1 = mTranslate1 * mRotateY * mTranslate2 ;
g_pWorldVariable->SetMatrix((float*)&g_World1);

g_pTechnique->GetDesc(&techDesc);
g_pTechnique->GetPassByIndex(0)->Apply(0);
g_pd3dDevice->DrawIndexed(36, 0, 0);

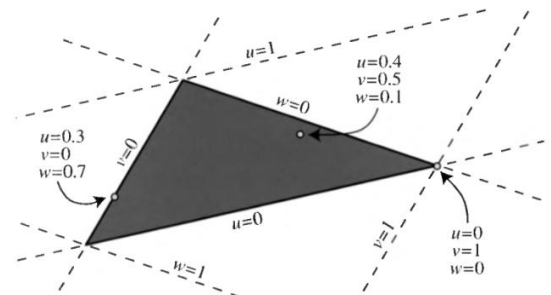
g_World2 = mTranslate3 * mRotateZ * mTranslate4 ;
g_pWorldVariable->SetMatrix((float*)&g_World2);

g_pTechnique->GetPassByIndex(0)->Apply(0);
g_pd3dDevice->DrawIndexed(36, 0, 0);



Yukarıdaki programın çıktısı **Şekil 2**'de gösterilmiştir. Küplerden biri g_World1 ile (0.0f, -5.0f, 0.0f) noktasında Y eksenini etrafında 2 birim; diğeri de g_World2 ile (0.0f, 0.0f, 4.0f) noktasında Z eksenini etrafında 3 birim yarıçaplı daire çizecek şekilde programda ?, ?, ? ile gösterilen yerlere gerekli öteleme koordinatlarını yazınız. **(25 PUAN)**

3. Bilindiği gibi (u,v) barisentrik koordinatlar kullanılarak (P₀,P₁,P₂) üçgeni içindeki herhangi bir nokta P₀+u(P₁-P₀)+v(P₂-P₀) ifadesi ile hesaplanabilir. İfade yeniden düzenlenirse (1-u-v)P₀+uP₁+vP₂ halini alır. (1-u-v), w ile temsil edilirse ifade wP₀+uP₁+vP₂ olur. Bu son ifadeye göre aşağıda köşe noktalarının koordinatları (P₀,P₁,P₂) olarak verilmiş üçgen içindeki (80,55,0) noktası için (w,u,v) koordinatlarını bulunuz. **(25 PUAN)**



(Real-Time Rendering III, p:764)

P₀=(20, 10, 0) P₁=(60, 80, 0) P₂=(100, 50, 0)

4. a) Index buffer, vertex bufferın boyu ile ilgili hangi avantajı sağlar? Küp için açıklayınız. **(15 PUAN)**
b) Vertex shader ve pixel shaderın görevleri nelerdir? **(10 PUAN)**