



CEVAPLAR

1. Şampiyonlar ligi çeyrek final rövanş maçında Galatasaray'ın 3. golünü Amrabat'tan gelen topa şık bir topuk vuruşuyla Drogba atmıştı. Topun Drogba'nın topuğundan aynasal sekip gol olduğu varsayalım. Topun kale çizgisini geçtiği nokta $(-4,0,67)$, Drogba'nın topuğundan sektiği nokta $(-4,0,57)$ ve sektiği noktanın normali $(0.8,0,0.6)$ 'dır. Drogba'nın Amrabat'a uzaklığı 25 metredir. Amrabat'ın pası hangi noktadan verdiğini hesaplayınız. (30P)



Backward Ray Tracing (Geri çekim :) ...

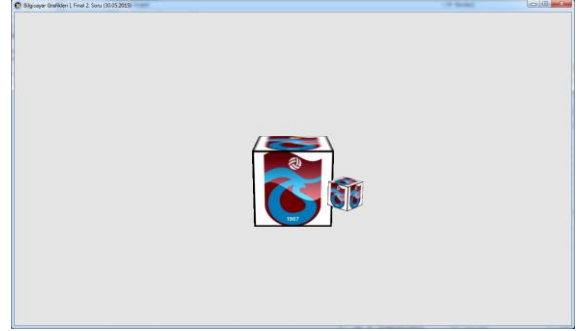
**Geri çekimde kale çizgisinden Drogba'ya gelen topun doğrultusu :
gericekimTopRd = (0, 0, -1)**

**Drogba'nın topuğundan yansıma doğrultusu :
fromDrogba = (I-2(I*N)*N) = (0,96, 0, -0,28)**

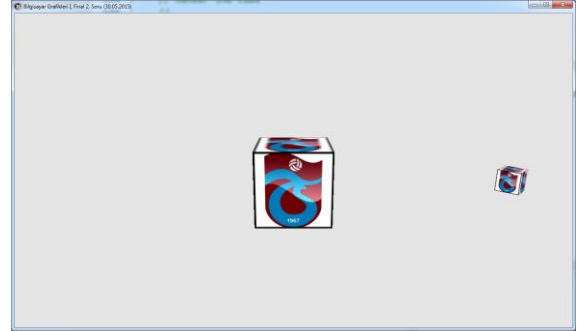
**Amrabat'ın topa vurduğu nokta :
ayakAmrabat = (20, 0, 50)**

```
mRotate = XMMatrixRotationY(XM_PIDIV4); //45° CW  
mTranslate = XMMatrixTranslation(6.0f,0.0f,0.0f);  
mScale = XMMatrixScaling(0.3f,0.3f,0.3f);
```

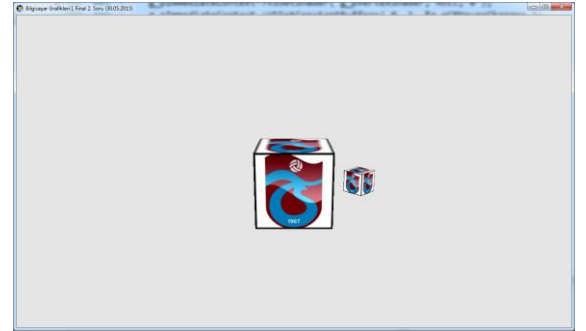
2. Aşağıda ilgili şeklin altına, büyük küpe ait **g_World** matrisi yukarıdaki matrisler ile setlenerek çizilen küçük küp için **g_World** matrislerini yazınız. (20P)



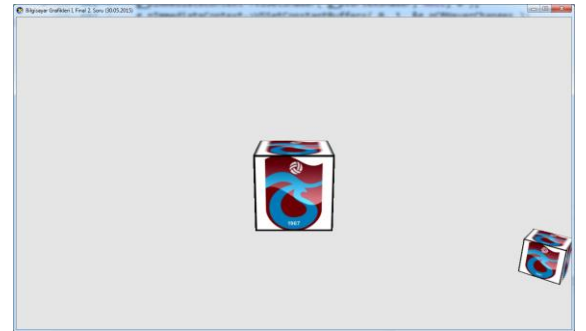
```
g_World = mTranslate * mScale * mRotate ;  
g_World = mTranslate * mRotate * mScale ;
```



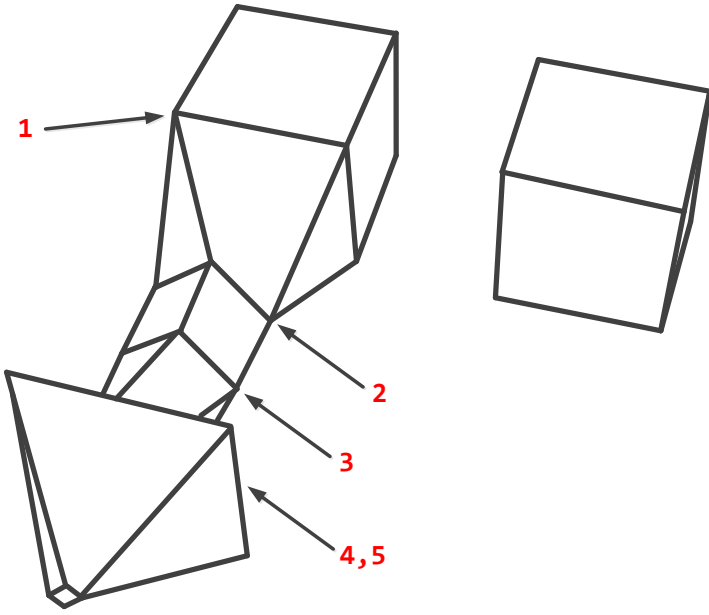
```
g_World = mRotate * mScale * mTranslate ;  
g_World = mScale * mRotate * mTranslate ;
```



```
g_World = mRotate * mTranslate * mScale ;
```



```
g_World = mScale * mTranslate * mRotate ;
```



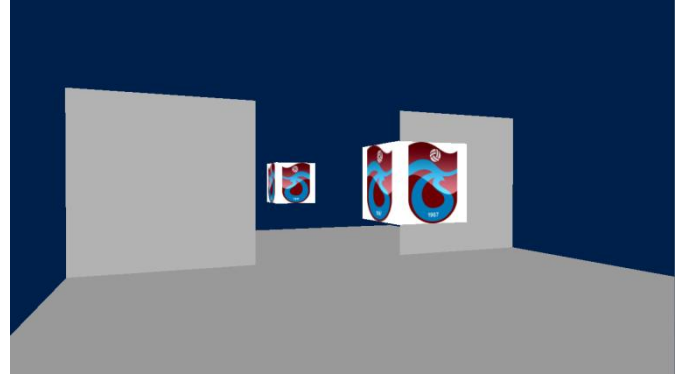
3. Yukarıda sağdaki küpün MAYA ortamında Extrude, Move, Scale ve Rotate toolları ile soldaki hale getirilmesi için kaç kez Extrude yapmak gerekir? Extrude yapılan yüzeyleri soldaki şekil üzerinde gösteriniz. (10P)

**5 kez Extrude yapmak gerekir.
İlan edilen videoyu oynatınız.**

$$C = C_{src} \otimes F_{src} \boxplus C_{dst} \otimes F_{dst}$$

4. Yukarıdaki Blending eşitliğini değişkenlerin alpha-blending yaparken aldıkları değerleri de vererek kısaca açıklayınız. (20P)

Blending eşitliğinde, back buffera ikinci yazılan C_{src} source pikselin rengi F_{src} isminde; back buffera ilk yazılan destination rengi C_{dst} de F_{dst} isimli bir blending katsayısı (factor) ile çarpılıp \boxplus ile temsil edilen bir işlem (blending operation) ile son renk değeri elde edilmektedir. Blending işlemi alpha-blendingde ADD işlemidir. Blending katsayısı olarak da $F_{src} = SRC_ALPHA$ ve $F_{dst} = INV_SRC_ALPHA$ (yani $1 - SRC_ALPHA$) alınır.



5. DirectX'te Stencil buffer ile aynasallığın nasıl gerçekleştirildiğini adım adım maddeler halinde açıklayınız. (20P)

1. Stencil buffer 0'lanır.
2. Ayna hariç cisimler back buffera çizilir.
3. Stencil buffera 0'dan farklı bir değer örneğin 61 yazılacak şekilde DSS (depth stencil state) setlemesi yapılır.
4. Ayna back buffera çizilirken stencil bufferda ona ait piksellere 61 yazılır.
5. Aynaya yansımaları çizilecek cismin yansımaları hesaplanır.
6. Stencil bufferda yalnızca 61 olan piksellere back bufferda çizim yapmak üzere DSS setlenir.
7. Stencil bufferda 61 olan pikseller için aynanın yansımaları back buffera çizilir.