

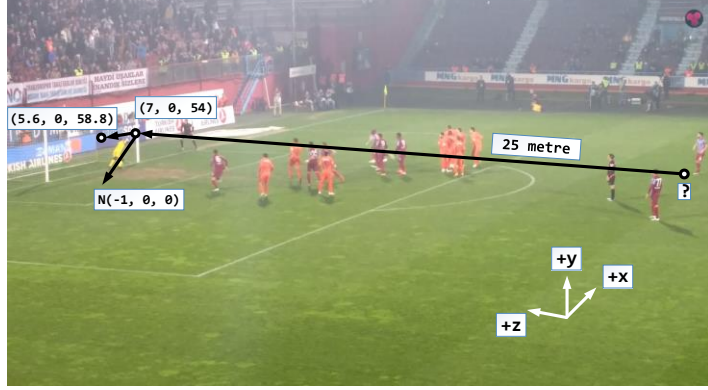


**Sınavda Uyulması Gereken Kurallar**

1. Cep telefonlarının, hesap makinesi, saate bakmak gibi herhangi bir amaçla kullanılması yasaktır. Telefon **kapalı** ve **cepte** olmalıdır.
2. **Sınavın başında** öğrenciler anlamadıkları noktaları **sesli olarak sorup**, Hoca cevapladıktan sonra **sınav boyunca soru sormak yasaktır**.

NUMARA :	AD SOYAD :	İMZA :	DEĞERLENDİRME :
----------	------------	--------	-----------------

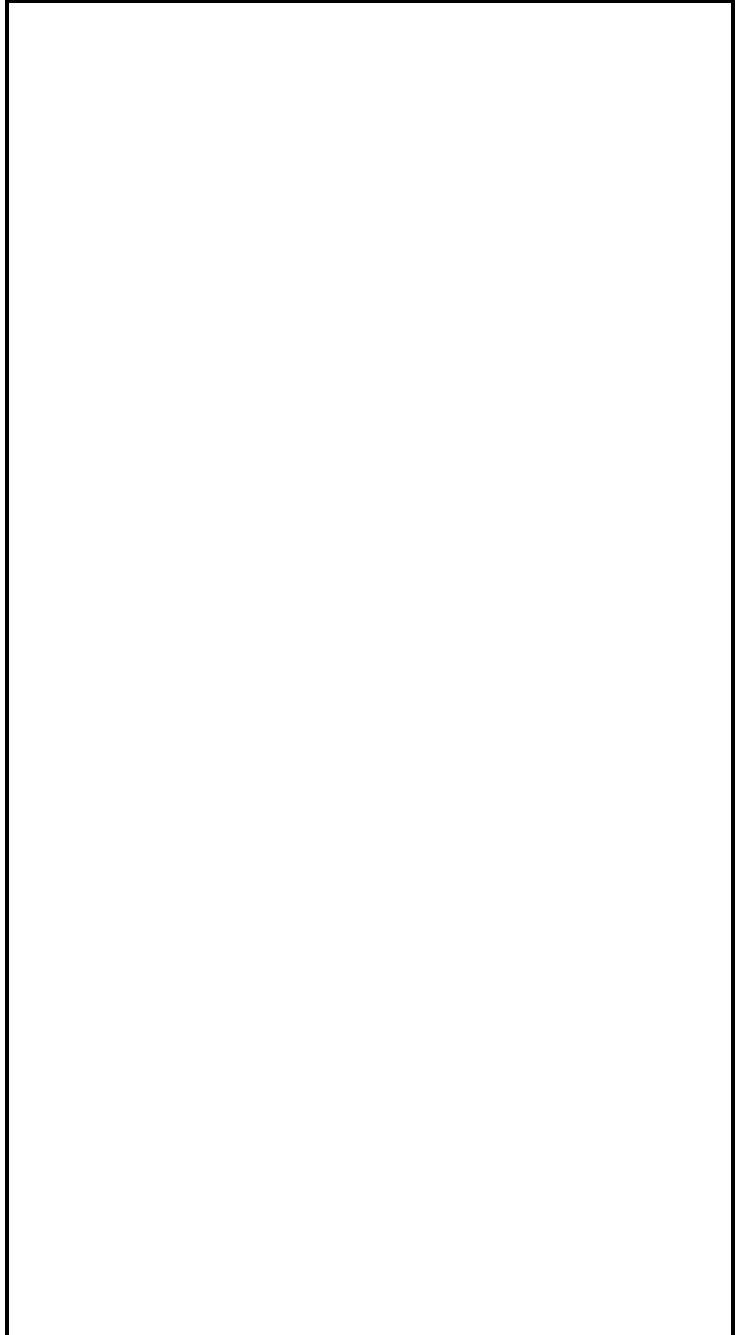
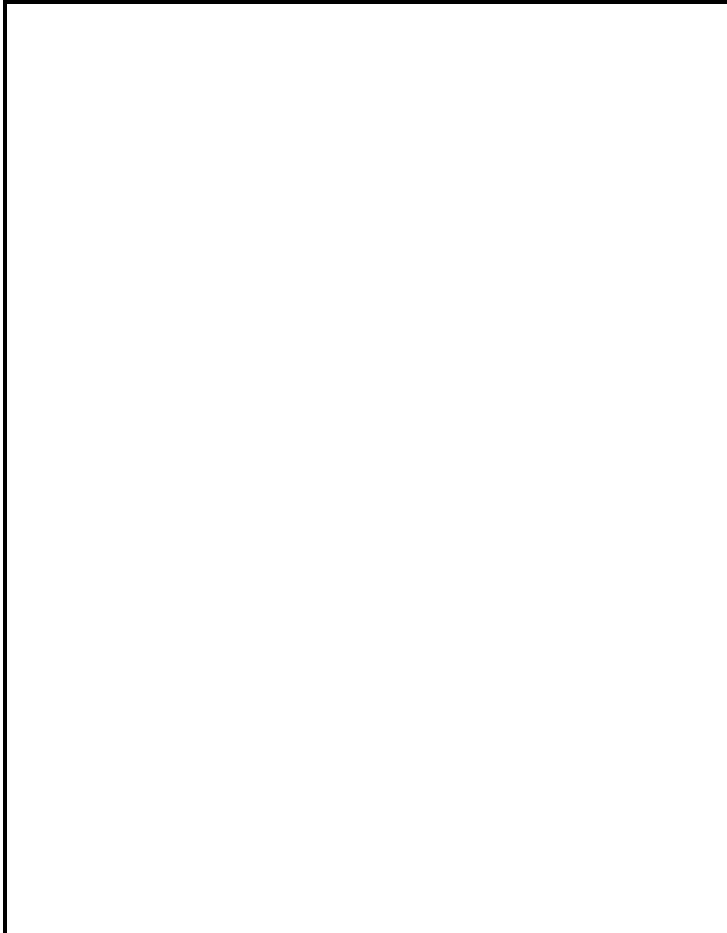
1. UEFA'nın twitter hesabında "[free-kick master](#)" olarak övülen Mehmet EKİCİ, bu sezon serbest vuruş gollerinden birini Başakşehir takımına [atmıştı](#). Top sağ kale direğinden aynasal yansıma göre sekip ağlarla buluşmuştu. Topun ağlarla buluştuğu nokta  $(5.6, 0, 58.8)$ , kale direğinden sektiği nokta  $(7, 0, 54)$ , direğin normali  $(-1, 0, 0)$ , şutu çektığı noktanın kale direğinden sektiği noktaya uzaklığı da **25 metre** olsun. Mehmet'in şutu hangi noktadan çektiğini ışın izleme ile hesaplayınız. (30P)



$U_0(0, 30, 160)$   $U_1(30, -30, 80)$   $U_2(-30, -30, 80)$   
 $V_0(0, -30, 160)$   $V_1(30, 30, 80)$   $V_2(-30, 30, 80)$

2. Yukarıda köşe noktaları verilen U ve V üçgenlerinin arkayüz (backface) olup olmadıklarını belirleyiniz. Dilediğiniz yöntemi kullanabilirsiniz. Bakış noktası  $(0, 0, 0)$ 'dadır. Görüntü düzleminin bakış noktasına uzaklığı 5 birimdir. Normalleri normalize etmeniz gerekmiyor. (30P)

$$R_1 \times R_2 = (R_{1y}R_{2z} - R_{1z}R_{2y}, R_{1z}R_{2x} - R_{1x}R_{2z}, R_{1x}R_{2y} - R_{1y}R_{2x})$$



```
Rotate = XMMatrixRotationY( XM_PIDIV4 ); //45° CW
Translate = XMMatrixTranslation(4.0f, 0.0f, 0.0f);
Scale = XMMatrixScaling( 0.5f, 0.5f, 0.5f );
```

```
g_World = Rotate * Translate * Rotate * Scale; //1
g_World = Rotate * Scale * Rotate * Translate; //2
g_World = Translate * Rotate * Scale * Rotate; //3
g_World = Scale * Rotate * Translate * Rotate; //4
g_World = Rotate * Translate * Scale * Rotate; //5
g_World = Rotate * Scale * Translate * Rotate; //6
```

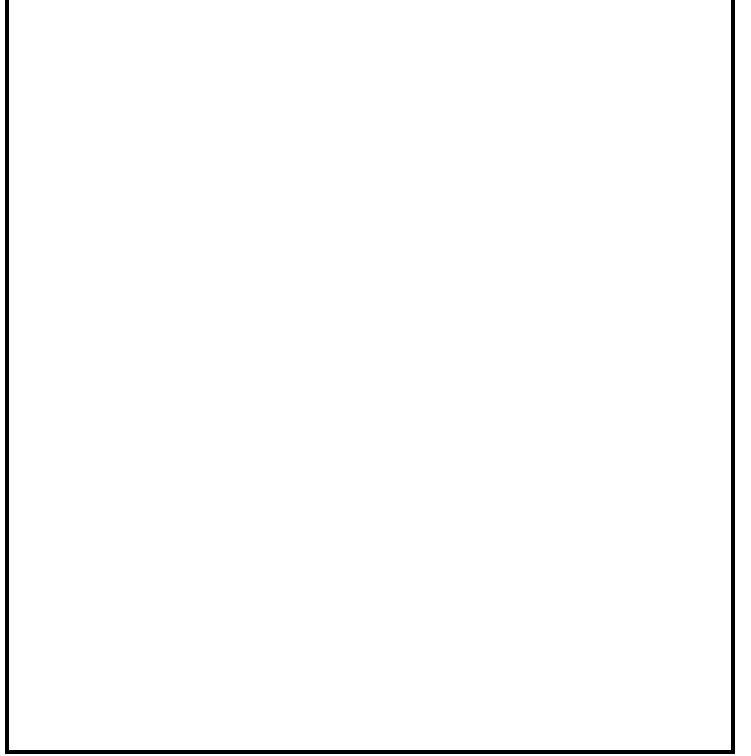
3. Küçük küpe ait 1-6 arası rakamlarla temsil edilen yukarıdaki **g\_World** matrisi setlemelerinin **I, II, III** ve **IV** ile temsil edilen aşağıdaki ekran görüntüleri ile doğru eşleştirildiği şıkkı işaretleyiniz? (25P)

Not → Bakış noktası **(0,4,-8)**'dedir.  
 Büyük küpün merkezi **(0,0,0)**'dadır, köşe noktaları **-1,+1** değerleri ile setlenmiştir.  
**45° CW** : Saat yönünde 45 derece dönme işlemi.

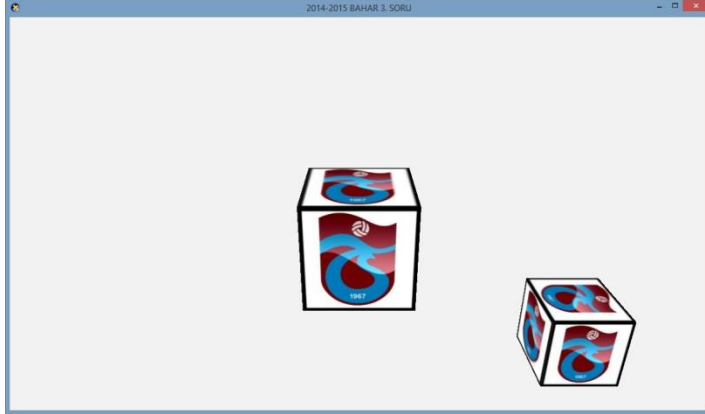
*Yanlış cevaptan 5P kırılacaktır!*

- |     |       |      |       |        |
|-----|-------|------|-------|--------|
| (A) | I-1,5 | II-3 | III-2 | IV-4,6 |
| (B) | I-4,6 | II-3 | III-2 | IV-1,5 |
| (C) | I-1,4 | II-5 | III-2 | IV-3,6 |
| (D) | I-1,6 | II-5 | III-2 | IV-3,4 |
| (E) | I-1,6 | II-3 | III-2 | IV-4,5 |

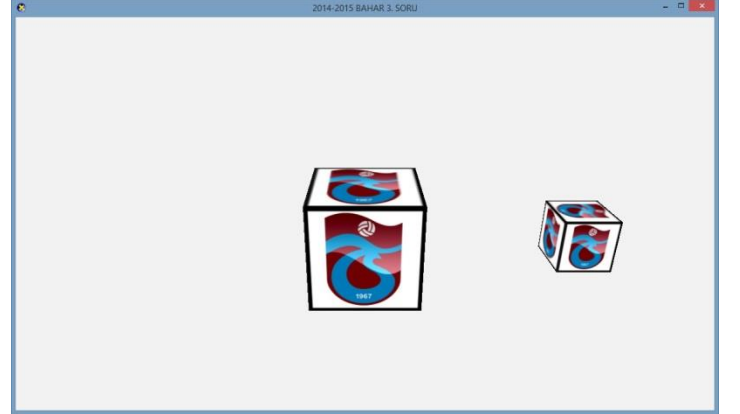
4. 3. soruda büyük küp ve küçük küp aynı vertex/index bufferları kullandığı halde farklı **World** matrisleri kullanılarak farklı büyüklükte ve konumda çizilebilmektedir. Bu iki küpün sanki farklı vertex/index bufferlar kullanıyorlarmış gibi aynı anda ekranda görüntülenmesini sağlayan nesnenin adı nedir? Bunu nasıl ve hangi fonksiyonu kullanarak yapmaktadır? (15P)



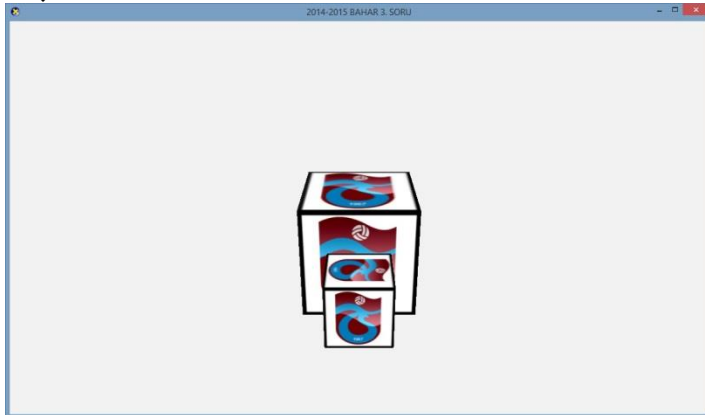
I)



III)



II)



IV)

