



Sınavda Uyulması Gereken Kurallar

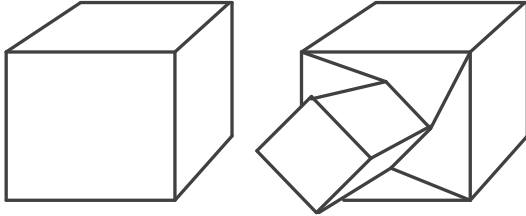
1. Cep telefonlarının, hesap makinesi, saate bakmak gibi herhangi bir amaçla kullanılması yasaktır. Telefon **kapalı** ve **cepte** olmalıdır.
2. **Sınavın başında** sorular kısaca açıklanacaktır. Öğrencilerin soruları cevaplandıktan sonra sınav boyunca soru sormak yasaktır.
3. Soru kağıdına numaranızı ve isminizi yazıp imzalayınız. Soru kağıdı **Sizde kalacaktır**.

NUMARA :

AD SOYAD :

İMZA :

Aşağıdaki sorulardan yalnızca dilediğiniz 5 tanesini cevaplayınız.



1. MAYA ortamında yukarıda soldaki küpün sağdaki hale gelebilmesi için gerekli işlemleri maddeler halinde yazınız. (20P)

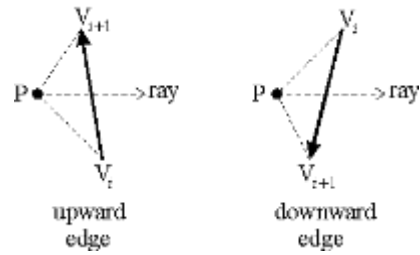
2. MAYA'da animasyon yaparken hareket edecek cismin hızının nasıl değişeceğini ayarlamak için Graph Editördeki eğriler nasıl güncellenir? Maddeler halinde yazınız. (20P)

3. a) Paralel izdüşüm tekniklerinden aksonometrik izdüşüm kendi içinde kaç sınıfa ayrılır? Açıklayınız. (6P)

b) Aşağıda ters perspektif dönüşüm mantığı ile oluşturulmaya çalışılan silindir görüntüsü doğru oluşturulabilmiş midir? Oluşturulmadıysa nasıl bir hata yapılmış olabilir? Oluşturulduysa bu görüntü (silindir için) nasıl doğru bir görüntü olabilir? (14P)



6. a) Işın izlemede kullanılan kesişim testi yöntemlerinden biri de **winding number** yöntemidir. Bu yöntem **crossings** testine benzer olmakla beraber farklı bir yöntemdir. Yukarıya çıkan (upward) kenarlar için +1, aşağıya inen (downward) kenarlar için de -1 değeri verilerek toplamlarından winding sayısı elde edilir. Bu sayı ile kesişim olup olmadığı belirlenir. Aşağıda verilen crossings testinde güncellemeler yaparak winding number yönteminin yalnızca kodunu elde ediniz. (6P)



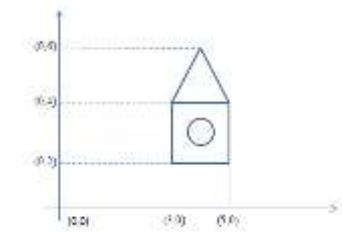
```

bool PointInPolygon(t, P)
returns ({TRUE, FALSE});
1: bool inside = FALSE
2: e0 = v_{n-1}
3: e1 = v_0
4: bool y0 = (e0_y >= t_y)
5: for i = 1 to n
6:   bool y1 = (e1_y >= t_y)
7:   if (y0 != y1)
8:     if (((e1_y - t_y)(e0_x - e1_x) >= (e1_x - t_x)(e0_y - e1_y)) == y1)
9:       inside = !inside
10:  y0 = y1
11:  e0 = e1
12:  e1 = v_i
13: return inside;

```

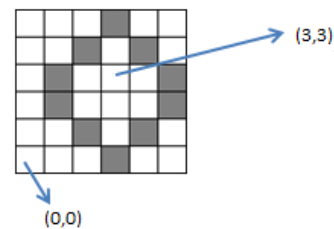
b) Bakış noktası $(0,0,0)$ dan çıkan ve $R_d=(0,0,1)$ doğrultusu boyunca giden R ışını **kırmızı** renkli T veya **mavi** renkli U veya **yeşil** renkli V üçgeniyle kesişmektedir. Üçgenler aynı normallere sahip olup normaler $N_{T,U,V}=(0, 0.6, -0.8)$ dir. Kesişim noktası için görüntü düzleminde ışının geçtiği pikselde T, U veya V üçgenlerinden birinin rengi görülmektedir. Işın izleme yöntemiyle görünmeyen yüzeyleri kaldırarak bu üçgenin hangi üçgen olduğunu bulunuz. Işın ile üçgenlerin tanımladığı yüzey arasındaki kesişim noktaları aynı zamanda üçgenlerin içindedir. Üçgenlerin köşe noktalarının koordinatları aşağıda verilmiştir. (14P)

Polygon (-30, 40, 85, 30, 40, 85, 0, -40, 25) // T üçgeni
 Polygon (-30, 40, 90, 30, 40, 90, 0, -40, 30) // U üçgeni
 Polygon (-30, 40, 80, 30, 40, 80, 0, -40, 20) // V üçgeni



4. Yanda koordinatları verilen şekil **OpenGL** kodları kullanarak çizilmek isteniyor. Bunun için OpenGL iskelet programının **gosterim()** fonksiyonu kullanılacaktır. Elinizde kare, üçgen ve daire şekillerini belirtilen ölçüde $(0,0)$

koordinatını merkez olarak alarak çizen **daire_ciz()**, **ucgen_ciz()**, **dortgen_ciz()** fonksiyonları hazır bulunmaktadır. Bu hazır fonksiyonları kullanarak yanda verilen şekli çizen **gosterim()** fonksiyonunu yazınız. (20P)



5. Yandaki şekilde verilen kapalı poligon **çekirdek doldurma yöntemi** ile doldurulmak isteniyor. Başlangıç noktası olarak $(3,3)$ koordinatı seçiliyor ve yığına itiliyor. Kapalı poligonun çekirdek doldurma yöntemiyle doldurulmasını ve bu esnada kullanılan yığının her bir adımıdaki durumunu ayrıntılı şekilde gösteriniz. (20P)