

# **BU ÇALIŞMA KAĞIDINDA VERİLEN KONULAR ÖZELLİKLE ÇALIŞMASI GEREKMEKLE BERABER DERSTE GÖRÜLEN TÜM KONULAR SINAV İÇERİĞİDİR:**

## **Giriş ve Tarih:**

Bilgisayar grafiği nedir? Bilgisayar Grafiği oluşturmanın temel elamanları nedir?

Bilgisayar grafiğinin tarihsel gelişimini kısaca anlatınız.

Görüntüleme cihazları hakkında bilgi veriniz? CRT nedir? Çalışmasını açıklayınız. Tek renkli(monochrome) ve renkli monitör yapılarını açıklayınız.

Taramalı(Rastering) ve Vektörel imge oluşturma sistemlerini açıklayınız? Avantaj-dezavantajlarını yazınız.

Frame Buffer(Çerçeve tamponu) nedir açıklayınız. Buffer büyüklüğünün çözünürlükle olan ilişkisini açıklayınız.

Grafik kartlarının yapısını ve mikroişlemcili sistemlerle ilişkisini açıklayınız.

## **Görüntüleme(İmge oluşturma) Sistemleri:**

2-Boyutlu (2-Dimensional Viewing-Transformation Pipeline)Görüntüleme-Dönüştürme İşhattı 'nın blok diyagramını çizerek her bir ünitenin giriş ve çıkışını belirterek fonksiyonlarını açıklayınız.(GENİŞ OLARAK AÇIKLANMASI YAPILMALI; Modelling Coordinate(Modelleme Kordinatları):Her bir nesneye ait ayrı ayrı modelin oluşturulduğu kordinatlar(3-boyutlu cisimin ölçüleri) , World-Coordinate(Dünya Kordinatları): Modelleri oluşturulan cisimlerin ortak bir kordinatta birleştirilmesi , Viewing Coordinate(Görüntüleme Kordinatları): Tüm resim içinden istenilen Pencereye(Window) ait kısmın seçilmesi(Clipping), Normalized Viewing Coordinate: Seçilmiş imgenin bütün kordinat değerlerinin belirli sınırlar arsına getirilmesi(Örn:OpenGL [-1,1]), Device Coordinates :Aygıt Kordinatı: Normalize edilen resimin görüntü aygıtında gösterileceği(Viewport) yere aktarılması. (Örn:Frame Buffer'da uygun hafıza noktalarına aktarılması)

Clipping, Window, Viewing, Viewport terimlerini açıklayınız.

Projeksiyon Nedir?

Perspektif ve Orthographic projeksiyon nedir? açıklayınız.

Viewing Volume nedir?

## **Geometri ve Transformasyon Matrisleri:**

2-boyutlu imgeler için Translational(Öteleme), Rotational(Döndürme) ve Scaling(Ölçeklendirme) Matrisleri nasıl oluşturulur. Temel geometriyi vererek açıklayınız.

Homojen(Homogeneous) kordinat nedir? Ne amaçla kullanılır?

İstenilen bir noktaya göre döndürme yapılacak imge için hangi işlemler gereklidir?

Yansıma:(Reflection)X eksenine ve Y eksenine göre döndürmede M matrisi nasıl olmalıdır?

Makaslama, Kaykılma(Shear): Makaslama matrisini şekiller çizerek açıklayınız.

## OPENGL

OpenGL hakkında bildiklerinizi yazınız.

OpenGL kütüphanelerini açıklayınız.

GL\_MODELVIEW'de hangi işlemler yapılabilir.(Bu moda girdikten sonra kullanılan komutlara bakarak çıkartabilirsiniz.)

GL\_PROJECTION'de hangi işlemler yapılabilir.

### Matrisler Hakkında:

OpenGL'deki 3 matris modu:

- GL\_MODELVIEW
- GL\_PROJECTION
- GL\_TEXTURE

Matrisler glMatrixMode(...); ile seçilir.

```
glMatrixMode(GL_MODELVIEW);  
glLoadIdentity();  
glTranslatef(10.5, 0, 0);  
glRotatef(45, 0, 0, 1);
```

```
-glFrustrum (...);  
-gluPerspective (fovy,aspect, near, far);  
-glOrtho (...);  
-gluLookAt (...);  
-glViewport(0, 0, 400, 400)
```

### Aşağıdaki fonksiyonları açıklayınız:

```
glutInit(&argc, argv);  
glutInitDisplayMode (GLUT_DEPTH | GLUT_DOUBLE |GLUT_RGBA);  
glutSwapBuffers();  
glShadeModel(GL_FLAT); //GL_SMOOTH
```

```
//CALLBACK fonksiyonları  
glutDisplayFunc( Display );  
glutKeyboardFunc( Keyboard );  
glutReshapeFunc( Reshape );  
glutMouseFunc( Mouse );  
glutPassiveMotionFunc( PassiveFunc );  
glutMotionFunc( MouseDraggedFunc );  
glutIdleFunc( Idle );  
glutMainLoop();
```

```
glVertex3f (2.0, 4.1, 6.0);  
glVertex2i (4, 5);
```

```
glColor3f(0.0, 0.0, 0.0);
```

**Nasıl Kullanılır:**

- \_ GL\_POINTS individual points
- \_ GL\_LINES pairs of vertices interpreted as individual line segments
- \_ GL\_LINE\_STRIP series of connected line segments
- \_ GL\_LINE\_LOOP same as above, with a segment added between last and first vertices
- \_ GL\_TRIANGLES triples of vertices interpreted as triangles
- \_ GL\_TRIANGLE\_STRIP linked strip of triangles
- \_ GL\_TRIANGLE\_FAN linked fan of triangles
- \_ GL\_QUADS quadruples of vertices interpreted as four-sided polygons
- \_ GL\_QUAD\_STRIP linked strip of quadrilaterals
- \_ GL\_POLYGON boundary of a simple, convex polygon