

## **OpenGL TANIMI Uygulama**

1. Giriş Günümüzde yazılım ve donanımın gelişmesi ile birlikte bilgisayar grafikleri alanında oldukça önemli gelişmeler kaydedilmektedir. Bu gelişmelere paralel olarak yazılım geliştirme kütüphaneleri de ortaya çıkmaktadır. OpenGL (Open Graphic Library) de gelişmiş grafik uygulamaları için geliştirilen bir kütüphanedir. OpenGL grafik kütüphanesi, grafik donanımına bir yazılım arayüzüdür. 2D ya da 3D nesnelerden oluşan hareket eden görüntüler üreten, interaktif, çok çeşitli bilgisayar platformlarında çalışabilecek programlar yaratmamıza olanak sağlar. OpenGL, donanım-bağımsız bir arayüzdür. Görüntüde bulunan nesneleri tanımlamak ve bu nesneler üzerinde gerek duyulan işlemleri gerçekleştirmek için yaklaşık 700 farklı komut içerir. (Bu komutlardan yaklaşık 650 tanesi saf OpenGL komutu ve yaklaşık 50 tanesi ise OpenGL Utility Library de bulunmaktadır [1]. OpenGL komut yorumlama modeli, istemci/sunucu (client/server) modelidir. Uygulama kodu (istemci), komut çağrılarını yapar. Bu komutlar, OpenGL (sunucu) tarafından yorumlanır ve işlenir. OpenGL, grafik programını koştan ve yarattığımız grafiği görüntüleyen bilgisayarlar farklı olduğunda bile etkili çalışacak şekilde tasarlanmıştır. Bu durum, istemci ve sunucunun farklı tip bilgisayarlar olabileceği bir bilgisayar ağı ortamında geçerlidir. OpenGL 'in donanım-bağımsız olmasının nedeni, pencere işlemlerini (windowing task-ekranda bir pencere oluşturmak gibi) yapan ya da kullanıcıdan girdi alan herhangi bir komutunun bulunmamasıdır. Belirtilen bu işleri gerçekleştirmek için işletim sisteminin mevcut özellikleri kullanılır. Ancak işletim sisteminde pencere işlemlerini gerçekleştirmek karmaşık işlemler içerdiğinden tüm bu işlevleri barındıran ve işletim sistemine özel olarak yazılmış GLUT (Graphic Library Utility) kütüphaneleri bulunmaktadır. OpenGL, 3D nesneleri tanımlamak için yüksek-seviye komutlar içermez. Bunun yerine; nokta, doğru ve poligon gibi alt-seviye geometrik primitifleri

kullanarak nesneleri tanımlamamıza olanak sağlar. Primitifler, bir ya da daha fazla vertex ile tanımlanır. Bir nokta, bir doğrunun başlangıç ve bitiş noktaları, bir poligonun iki kenarının kesiştiği köşe noktası, vertex 'ler yardımıyla belirlenir.

## **2. OpenGL**

### **2.1.OpenGL'in Avantajları**

OpenGL kullanarak grafikler oluşturma'nın avantajları aşağıda sıralanmıştır. • Tüm OpenGL uygulamaları, işletim sistemi ne olursa olsun, OpenGL API uyumlu donanımlar üzerinde mükemmel görsel sonuçlar üretebilir. • Grafik donanımlarının yeni gelişmiş özellikleri, OpenGL tarafından, geliştirme mekanizması (extension mechanism) sayesinde kullanılabilir. Yani OpenGL, donanıma özel, gelişmiş özellikleri kullanmak için API fonksiyonları içerebilir. , • OpenGL temelli grafik uygulamaları, çok çeşitli sistemler üzerinde çalışabilir. (tüketici elektroniği – consumer electronics, PC, iş istasyonu – workstation, süper bilgisayarlar gibi) • OpenGL grafik kütüphanesi kullanılarak, çok daha az bir kod satırıyla daha yüksek performansa sahip uygulamalar geliştirmek mümkündür. • OpenGL grafik kütüphanesine dair teknik bilgi içeren birçok kaynak mevcuttur.(internet, kitaplar, vs.) • Birçok programlama dili (C, C++, Fortran, Ada, Java gibi) OpenGL tabanlı uygulama geliştirmemize olanak sağlar. 2.2.Open GL Utility (GLUT) OpenGL platformdan bağımsız olduğu için bazı işlemler bu kitaplık ile yapılamaz. Örneğin kullanıcıdan veri almak, bir pencere çizdirmek gibi işler hep kullanılan pencere yöneticisi ve işletim sistemine bağlıdır. Bu yüzden bir an için OpenGL'in platform bağımlı olduğu düşünülebilir. Çünkü çalışma penceresini her pencere yöneticisinde (her ortamda) farklı çizdirecek bir canlandırma programı yazmak demek her bilgisayarda çalışacak ayrı pencere açma kodu yazmak demektir. Bu ise OpenGL'in doğasına aykırıdır. Bu

gibi sorunları aşmak için OpenGL Araç Kiti (GLUT - OpenGL Utility Toolkit) kullanılmaktadır. GLUT, birçok işletim sistemine aktarılmış bir kitaplıktır. Amacı OpenGL programlarının pencerelerini oluşturmak, klavye ve fareden veri almak gibi ihtiyaçları karşılamaktır. GLUT olmadan da OpenGL programlama yapılabilir, örneğin Linux'ta kullanılan X-Window sistemin kendi işlevleri kullanılarak pencere çizdirilebilir fakat bu kod sadece X-Window'da çalışır. Kod Windows'a götürülüp derlendiğinde çalışmaz, çünkü Windows'da X-Window işlevleri yoktur. Benzer şekilde Windows tabanlı işletim sistemlerinin de kendilerine has pencere oluşturma işlevleri vardır. Bu yüzden bu deneyde GLUT kitaplığı kullanılarak klavye ve fare için işletim sisteminden bağımsız giriş/çıkış işlemleri yapılması sağlanmıştır.

### 2.3. OpenGL Söz Dizimi

OpenGL komutları, `gl` öneki ile başlarlar. (örnek; `glClearColor()`). Benzer şekilde OpenGL tarafından tanımlı sabitler de `GL_` öneki ile başlarlar ve kelimeler birbirinden `_` ile ayrılacak şekilde büyük harfle yazılırlar. (örnek; `GL_COLOR_BUFFER_BIT`). `glColor3f` komutundaki 3 sayısı da 3 parametre alacağı anlamına gelmektedir. `f` ise verilen parametrelerin float olacağı anlamına gelmektedir. `glVertex2i(1,3)`; ya da `glVertex2f(1.0,3.0)`; gibi