

# Bilgisayar Grafiği

## HAFTA 7

## OpenGL

**Arş. Gör. Dr. Gülüzar ÇİT**

Bilgisayar ve Bilişim Bilimleri Fakültesi

Bilgisayar Mühendisliği Bölümü

[gulizar@sakarya.edu.tr](mailto:gulizar@sakarya.edu.tr)

# Konu & İçerik

- OpenGL
  - OpenGL Nedir?
  - OpenGL'in Tarihçesi
  - OpenGL'nin Özellikleri
  - OpenGL Durum Makinesi
  - Uygulama Alanları
  - OpenGL Programının Genel Yapısı
  - GLUT
  - OpenGL Komut Biçimi
- Kaynaklar



# OpenGL Nedir?

- OpenGL
  - Open Graphics Library - Açık Grafik Kütüphanesi
  - Açık standarttır, yani pek çok şirket OpenGL'in gelişimine katkıda bulunabilir
  - Açık kaynak kodlu değildir.
- Gelişmiş donanım desteğini kullanarak hem iki hem de üç boyutlu grafikleri ekrana çizmek için kullanılan ücretsiz bir grafik uygulama geliştirme aracıdır.
- İki veya üç boyutlu grafik çizdirmek için kullanılan bir API (Application Programming Interface)'dir, bir başka deyişle kitaplıktır. Bu kitaplık yazılım geliştiricilere grafik donanımını yönetme şansı verir.
- Programlama dillerindeki önemli bir eksikliği giderir.
- Grafik donanımlarının gücünden yararlanır

# OpenGL'in Tarihçesi

- 1992 yılında SGI (Silicon Graphics) tarafından çok amaçlı platform bağımsız bir grafik API (Application Programming Interface) olarak geliştirilmiştir.
- 1992'den beri OpenGL'in gelişimi OpenGL ARB tarafından denetlenmektedir
  - OpenGL ARB (Architecture Review Board)
    - OpenGL'in geleceğini yönetmek için 1992'de kurulmuş bağımsız bir uluslar arası ticaret birliği – konsorsiyumdur
    - 3DLabs, ATI, DELL, Hewlet-Packard, IBM, Intel, NVIDIA, Sun Microsystems ve Silicon Graphics gibi büyük grafik satıcıları ve endüstri liderlerinden oluşur
- Günümüzde **4.6** sürümüne ulaşmıştır ve OpenGL sahip olduğunuz grafik donanımı tarafından sunulan tüm özelliklere erişim sağlayan yüzlerce fonksiyona sahiptir.

# OpenGL'in Özellikleri

## ➤ **Taşınabilirlik**

- Bu kitaplık işletim sisteminden ve işletim sisteminin çalıştığı platformdan bağımsızdır. Nasıl ki ekrana yazı yazmak kullanıcıdan veri almak ANSI C'de printf() ve scanf() gibi işlevlerle standartlaştırılmış ve hangi işletim sistemine giderseniz gidin bu iki işlev aynı işi yapıyorsa, OpenGL kitaplığı da ekrana grafik çizmeyi standartlaştırmıştır. OpenGL sayesinde grafik kartının modeli veya işlemcinin mimarisi gibi donanımsal etkenlerden bağımsız programlama yapılır. Ayrıca aynı zamanda işletim sisteminden de bağımsız programlama yapılır. Kolay kullanım ve bu "taşınabilirlik" özellikleri yüzünden OpenGL popüler bir araç olmuştur.

# OpenGL'in Özellikleri ...

## ➤ İşletim sisteminden bağımsızlık

- OpenGL kullanan bir programı işletim sisteminizde çalıştırmanız için öncelikle işletim sisteminizde programın çalışırken kullanacağı işlevleri içeren kitaplığın (genel adı "runtime-library" – çalışma anı kitaplığı) bulunması gerekir
- OpenGL çalışma anı kitaplığı Linux, Unix, Mac OS, OS/2, Windows işletim sistemlerinde vardır. Windows ailesinde standart olarak gelir. Yani OpenGL kullanan bir programı çalıştırmak için bir çaba harcamazsınız.

# OpenGL'in Özellikleri

## ➤ **Pencere yöneticisinden bağımsızdır**

➤ OpenGL kullanılarak yazılmış programlar, Win32, MacOS ve X-Window pencere yöneticilerinde sorunsuz çalışırlar.

## ➤ **Birçok programlama dilinden kullanılabilir**

➤ Ada, C, C++, C#, Fortran, Python, Perl ve Java programlama dilleri kullanılarak OpenGL kitaplığından faydalanılabilir.

# OpenGL Durum Makinesi

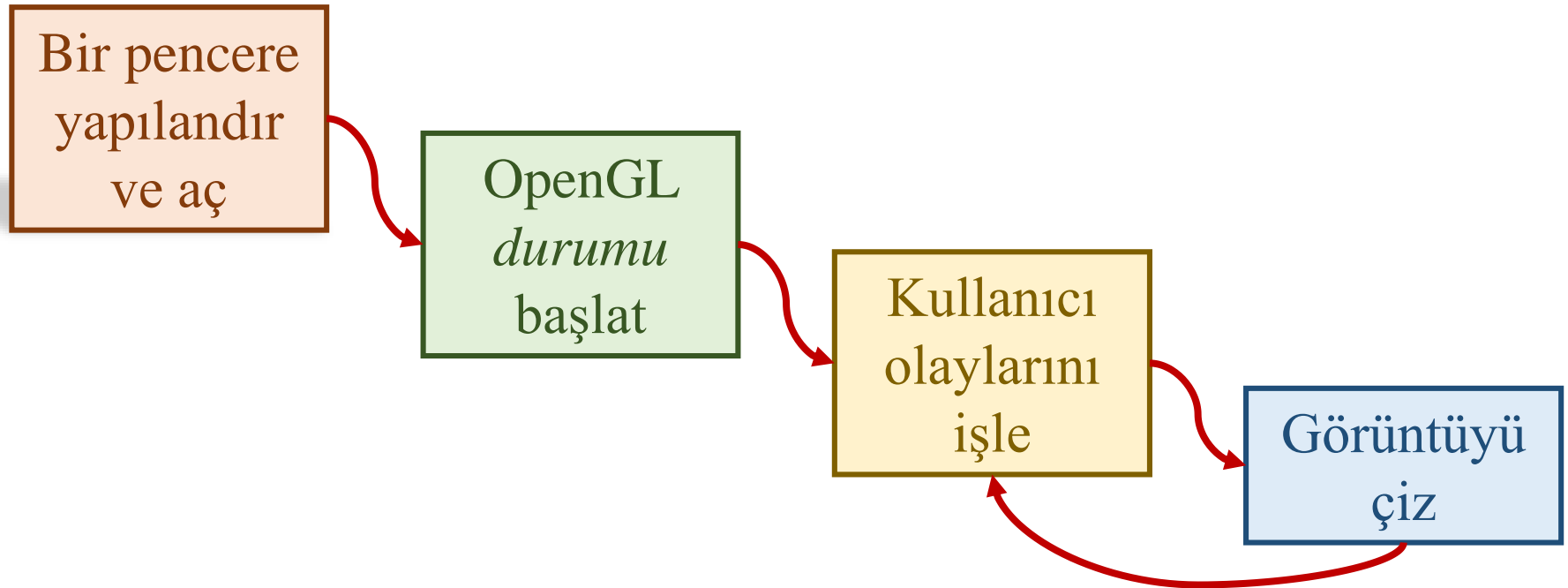
- OpenGL bir “Durum makinesi” (state machine) gibi çalışır
  - Yani her an verilen bir konumdadır ve sonraki komutlar bu konumda olduğu göz önüne alınarak çalışır.
    - Örneğin “ışıklandırma açık” ya da “kapalı” gibi.
    - Bir pencere oluşturduğumuzda pencereye atadığımız renk kırmızı, yeşil, mavi vs.. tamponlar(buffer) bu renk değerleri ile temizlenir ve bizim bir dahaki müdahalemize kadar penceremizin rengi belirlediğimiz durumda kalır.
  - OpenGL’de bir çok durum değişimi söz konusudur .
    - Örneğin OpenGL’de nokta oluşturulduğu zaman noktanın boyutunun sabit değeri 1’dir. Bir pencere oluşturup sadece bir nokta oluşturduğunuz zaman bu nokta görünmeyecek kadar ufak oluşmaktadır. Bu durumu değiştirmek için glPointSize fonksiyonu ile noktanın belirlenmiş durumunu değiştirebiliriz bu değişim diğer OpenGL durumlarında olduğu gibi bir dahaki değişime kadar sabit kalacaktır. Bu durum değişimlerinde bazı öğeleri etkin ve pasif (glEnable ve glDisable) olarak belirlememiz gerekmektedir.



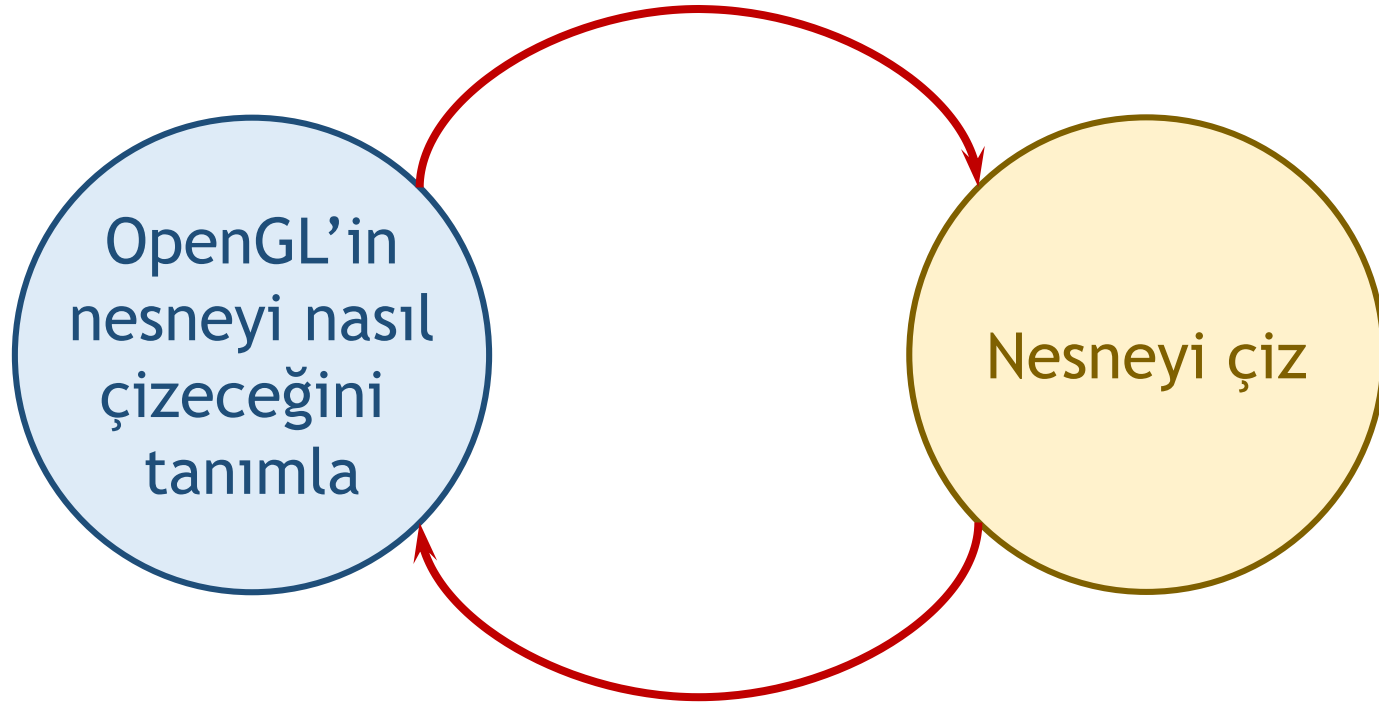
# OpenGL Uygulama Alanları

- Grafik uygulamaları
- Bilgisayar Oyunları – Doom3
- CAD modelleme
- Sanal Gerçeklik
- ...

# OpenGL Programının Genel Yapısı



# OpenGL Nasıl Çalışır?



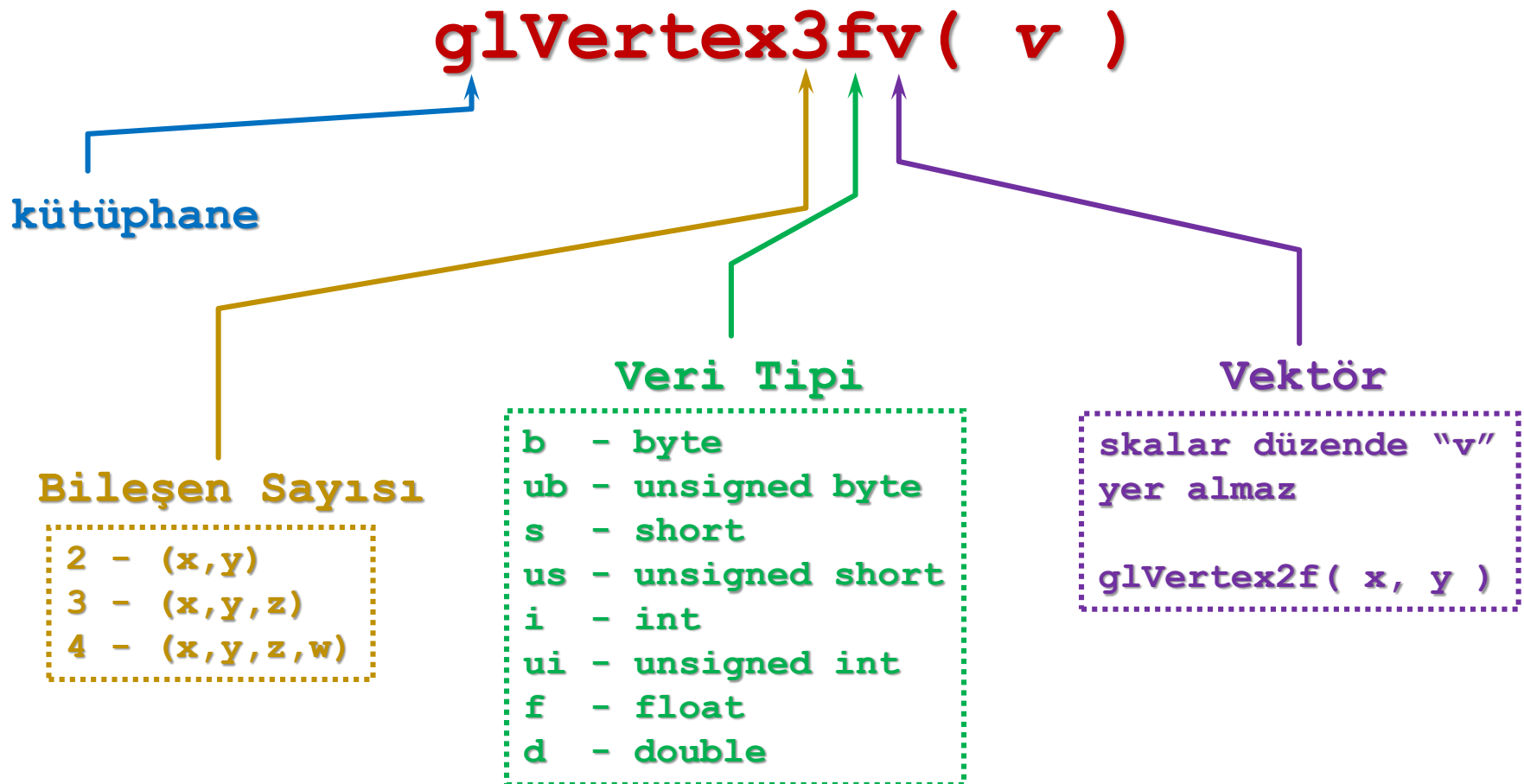
# OpenGL Kütüphaneleri ...

- OpenGL'e destek olmak ve fonksiyonelliğini artırmak için pek çok kütüphanesi bulunur.
  - GLU – OpenGL Utility Library
    - Eğriler, yüzeyler, yüzey döşeme görüntüleri, ... içerir.
  - AGL, GLX, WGL
    - OpenGL ile pencere sistemleri arasındaki katmandır
  - SDL – Simple Direct Media Layer
    - Cross-platform multimedya kütüphanesi
    - Ses, 2B ve 3B grafik, vb. desteği sağlar
    - 3B grafik desteğinden dolayı cross-platform oyun programlama için popüler bir tercihtir

# OpenGL Kütüphaneleri

- GLUT – OpenGL Utility Toolkit
  - Daha yalın bir pencere sistemi bütünleştirmesi sunar
  - Resmi anlamda OpenGL'in bir parçası değildir
  - freeGLUT, openGLUT gibi açık kaynak versiyonları mevcut
  - Son güncel versiyonu  $\Rightarrow$  3.7.6
- ...

# OpenGL Komut Biçimi



# OpenGL Programlarında Gerekenler

## ➤ Başlık Dosyaları

```
#include <GL/gl.h>
```

```
#include <GL/glu.h>
```

```
#include <GL/glut.h>
```

## ➤ Kütüphaneler

➤ `opengl32.lib`, `glut.lib`, `glut32.lib`

## ➤ Veri Tipleri (uyumluluk için)

➤ `GLfloat`, `GLint`, `GLenum`, ...

# GLUT

- Amacı OpenGL programlarının pencerelerini oluşturmak, klavye ve fareden veri almak gibi ihtiyaçlarını karşılamak olan GLUT birçok işletim sistemine aktarılmış bir OpenGL kitaplığıdır.
- OpenGL platformdan bağımsız olduğu için bazı işlemler bu kitaplık ile yapılamaz. Örneğin kullanıcıdan veri almak, bir pencere çizdirmek gibi işler kullanılan pencere yöneticisi ve işletim sistemine bağlıdır. Bu yüzden bir an için OpenGL'in bu durumlarda platforma bağımlı olduğunu düşünebiliriz. Çünkü penceresini her pencere yöneticisinde farklı çizdirecek bir canlandırma programı yazmak demek her bilgisayarda çalışacak ayrı pencere açma kodu yazmak demektir. Bu ise OpenGL'in doğasına aykırıdır. GLUT, yazılım geliştiricileri bu tip bir platforma bağılıktan tamamen kurtarıyor.



# GLUT

- 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!
- GLUT kitaplığı kullanılarak klavye ve fare için işletim sisteminden bağımsız giriş/çıkış işlemleri yapılması sağlanır.

# GLUT Callback Fonksiyonları

- Bir olay gerçekleştiğinde çağrılacak rutin
  - Pencere büyüklüğü değişimi
  - Kullanıcı girişi
  - Animasyon
- GLUT ile callback'lerin kaydedilmesi

```
glutDisplayFunc( goruntule );  
glutIdleFunc( idle );  
glutKeyboardFunc( klavye );
```

# GLUT Callback Fonksiyonları...

## ➤ **glutInit**

- Uygulamanın komut satırından bilgi almasını ve sistemi başlatmasını sağlar

## ➤ **glutInitDisplayMode**

- Pencere için çizim özelliklerini belirler.
  - RGB renk biçimi, tekli tampon biçimi

## ➤ **glutWindowSize**

- Piksel olarak pencere boyutu

## ➤ **glutWindowPosition**

- Ekranın sol üst köşesinden uzaklığı

## ➤ **glutCreateWindow**

- “xxx” adındaki pencerenin oluşturulması

## ➤ **glutDisplayFunc**

- Görüntüleme geri bildirim fonksiyonu

## ➤ **glutMainLoop**

- Sonsuz olay döngüsüne giriş

# Basit Bir OpenGL Programı

```
#include <glut.h>

void init();

void myDisplay();

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(0, 0);
    glutCreateWindow("ilk pencere");
    glutDisplayFunc(myDisplay);

    init();

    glutMainLoop();
}
```

gl.h ve glu.h da  
glut.h ile eklenir

Pencere özellikleri tanımlandı

Görüntüleme callback fonksiyonu

OpenGL durumu tanımlandı

Olay döngüsüne giriş

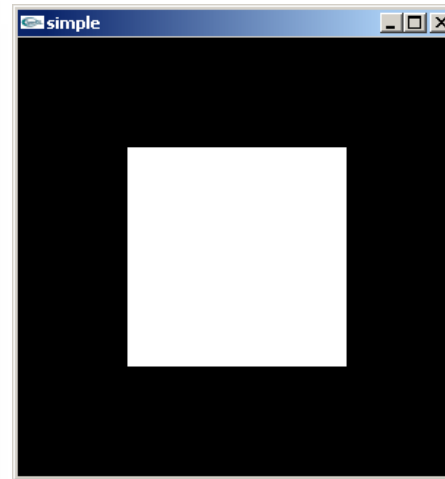
# Basit Bir OpenGL Programı

- Siyah bir arka plan üzerine beyaz bir kare çizer.

```
#include <glut.h>

void mydisplay() {
    glClear(GL_COLOR_BUFFER_BIT);
    glBegin(GL_POLYGON);
    glVertex2f(-0.5, -0.5);
    glVertex2f(-0.5, 0.5);
    glVertex2f(0.5, 0.5);
    glVertex2f(0.5, -0.5);
    glEnd();
    glFlush();
}

int main(int argc, char** argv) {
    glutCreateWindow("simple");
    glutDisplayFunc(mydisplay);
    glutMainLoop();
}
```



# Kaynaklar

- <https://www.opengl.org/>
- <https://learnopengl.com/Getting-started/OpenGL>