Android İşletim Sisteminde OpenGL ile Grafik Programları Geliştirme

Caner Ulutürk[1], Osman Gökalp[2], Aybars Uğur[1]

[1]Ege Üniversitesi, Bilgisayar Mühendisliği Bölümü, İzmir

[2]Yaşar Üniversitesi, Yazılım Mühendisliği Bölümü, İzmir

canerulu1@yahoo.com, osman.gokalp@yasar.edu.tr, aybars.ugur@ege.edu.tr

Özet: Mobil cihazlar günümüzde çok hızlı bir şekilde gelişmekte ve kullanımı yaygınlaşmaktadır. Mobil işletim sistemleri de buna paralel olarak büyük önem kazanmıştır. Yüksek bir ivmeyle yaygınlaşmakta olan Android işletim sistemi, açık kaynak kodlu olup akıllı telefonlar ve tablet bilgisayarlar gibi mobil cihazlar için geliştirilmiştir. Yaygınlaşmasındaki en önemli etken, kullanıcıların çeşitli ihtiyaçlarına cevap veren uygulamalara kolaylıkla ulaşabilmeleridir. Alışveriş, finans, haberleşme, eğitim ve daha birçok kategori içerisinde en çok ilgi çekenlerden bazıları oyun ve eğlence amaçlı olan uygulamalardır. Android işletim sisteminde grafik yazılımları geliştirmek için en uygun teknolojilerden birisi, OpenGL' in mobil cihazlara yönelik sürümü olan OpenGL ES'dir. Bu çalışma kapsamında, OpenGL ES tabanlı 3 boyutlu bir oyun tasarlanarak gerçekleştirimi yapılmıştır. Ayrıca OpenGL ES arayüzünün grafik programlamadaki avantajları ve önemi üzerinde durularak, farklı sürümleri karşılaştırılmıştır.

Anahtar Sözcükler: Android, OpenGL ES, Bilgisayar Grafikleri, Mobil Oyunlar

Abstract: Today, mobile devices are developing very fast and their usage is becoming widespread. Correspondingly, mobile operating systems came into prominence. Android operating system which is becoming widespread by leaps and bounds is open source and developed for mobile devices like smart phones and tablet computers. The biggest factor of its proliferation is easy accessibility of users to applications which are satisfying their needs. Some of the most featured applications within shopping, finance, communication, education and many other categories are game and entertainment applications. In Android operating system, one of the most convenient technologies for developing graphics software is OpenGL ES which is version of OpenGL for mobile devices. Within the scope of this study, a 3D game based on OpenGL ES is designed and implemented. In addition, advantages and importance of OpenGL ES interface on graphics programming are emphasized and its different versions are compared.

1. Giriş

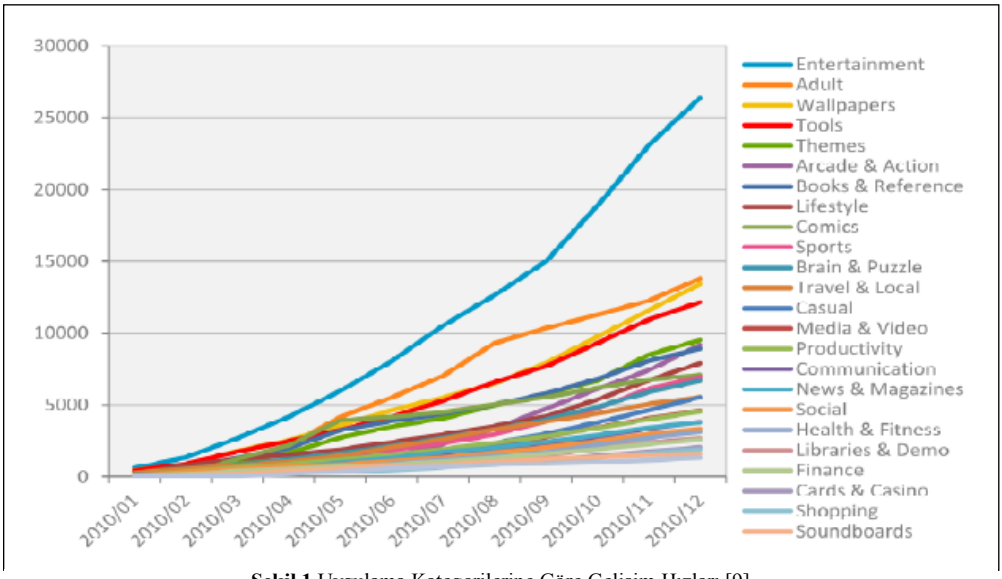
Günümüzde mobil cihazların gelişmesi ve yaygınlaşması ile birlikte mobil işletim sistemleri de büyük önem kazanmıştır. Android, akıllı telefonlar ve tablet bilgisayarlar gibi mobil cihazlar üzerinde çalışabilmesi için geliştirilmiş olan açık kaynak kodlu bir işletim sistemidir ve hızla büyüyen bir kullanım alanına sahiptir [2]. Yaygınlaşmasındaki en önemli etken, kullanıcıların çeşitli ihtiyaçlarına cevap veren uygulamalara kolaylıkla erişebilmeleridir. Alışveriş, finans, haberleşme, eğitim, grafik ve daha birçok kategoride geliştirilmiş çok sayıda uygulama bulunmaktadır.

Android, Inc. İlk olarak 2003 yılında Andy Rubin ve ortakları tarafından, Amerika Birleşik Devletleri' nin California eyaletinde küçük bir şirket olarak kuruldu. Daha sonra Google tarafından 2005 yılında satın alındı. 2007 yılında ise Google tarafından da desteklenen "Open Handset Allience" [1] kuruldu ve 2008 yılında ilk ticari sürümü olan Android 1.0 ortaya çıkarıldı. Bu tarihten sonra birçok sürüm geliştirildi ve son olarak Ekim 2011 tarihinde Android 4.0 piyasaya sürüldü [4][5].

Android uygulama geliştirme ortamında programlama dili olarak Java kullanılmaktadır. Android işletim sisteminin çekirdeği Linux tabanlı olacak şekilde tasarlanmıştır. Linux çekirdeği, Android mimarisinin en alt katmanında yer almaktadır. Bir üst katmanda ise kütüphaneler ve Dalvik Sanal Makinesi bulunmaktadır. Bu sanal makine ile Java derleyicisi tarafından derlenmiş sınıflar ".dex" biçimine çevrilerek Linux çekirdeği üzerinde çalıştırılabilmektedir. Üçüncü seviyede Uygulama Çerçevesi katmanı yer almaktadır. Çeşitli alanlarda geliştirilen uygulamalar ise en üst katmandadır [3].

Bilgisayar grafikleri, özel bir takım yazılım ve donanım araçlarıyla, bilgisayarlar tarafından görüntü verilerinin temsil edilmesi ve üzerinde manipülasyon yapılabilmesi olarak tanımlanabilir [6]. Mobil cihazlar standart masaüstü bilgisayarlara göre daha düşük işlemci gücüne ve bellek kapasitesine sahip olup güçlerini taşıdıkları bir bataryadan sağlamak zorundadırlar. Bu kısıtlayıcı durumlara rağmen, son yıllardaki gelişmeler ile mobil cihazların gerek donanımsal, gerekse yazılımsal yeteneklerinin ilerleme kaydetmesiyle birlikte, bilgisayar grafiklerinin bu alanda da kullanılması önem kazanmıştır. AndroidZoom tarafından açıklanan rapora [9] göre 2010 yılı boyunca Google Android Market üzerinde 170,000 uygulama yayınlanmıştır. Bu uygulamalar arasında en büyük paya sahip olanı eğlence amaçlı uygulamalardır ve büyüme hızına bakıldığında da en hızlı büyüyen kategoridir (Şekil 1). Oyunların eğlence amaçlı uygulamalar arasındaki yeri dikkate alındığı zaman mobil grafiklerin önemi bir kez daha ortaya çıkmaktadır.

Android ortamında grafik uygulamaları, kullanış amaçları ve ihtiyaçları doğrultusunda farklı şekillerde geliştirilebilmektedir. İki boyutlu uygulamalar geliştirilecekse "Canvas" sınıfı içinde yer alan hazır çizim metotları kullanılabilmektedir. "Canvas" sınıfının yetersiz kaldığı durumlarda, düşük seviyeli platform bağımsız bir uygulama geliştirme arayüzü olan OpenGL ES kullanılabilir. Opengl ES, OpenGL'in mobil cihazlar için uyarlanmış bir sürümü olup bazı gereksiz ve az kullanılan bileşenler çıkarılmış veya mobil cihazlar için uyarlanmıştır [7].



Şekil 1 Uygulama Kategorilerine Göre Gelişim Hızları [9]

Android ile grafik uygulamaları geliştirmenin bir diğer yolu ise NDK (Native Development Kit) kullanımıdır. NDK ile Java yanında C, C++ gibi diğer programlama dilleri ile yazılmış olan hazır OpenGL ES kaynak kodları doğrudan kullanılabilmektedir. Bunun yanında Android 3.0 (Honeycomb) sürümü ile birlikte gelen RenderScript de, grafik programları geliştirmek için bir diğer alternatiftir. C programlama dili kullanarak donanıma daha yakın kodlama yapmak isteyen geliştiriciler bu teknolojiyi kullanmayı tercih edebilirler.

2. OpenGL ES

2.1 Giriş

OpenGL, Open Graphics Library (Açık Grafik Kütüphanesi) anlamına gelmektedir. OpenGL ES (OpenGL for Embedded Systems – Gömülü Sistemler için OpenGL) ise, OpenGL' in gömülü sistemler için geliştirilmiş olan sürümüdür. Kâr amacı gütmeyen şirketler birliği olan Khronos Group Inc. tarafından yönetilmektedir. Cep telefonları, PDA' lar ve oyun konsolları gibi mobil cihazlarda 2D/3D grafiklerin çizimi için kullanılan OpenGL ES, programlama dili ve platform bağımsız olarak kullanılabilmektedir [10]. OpenGL ES' in, 1.0, 1.1 ve 2.0 olmak üzere üç adet sürümü bulunmaktadır. Bu sürümlerin hepsi Android işletim sistemi tarafından da desteklenmektedir. OpenGL ES 1.0/1.1 sürümleri Android 1.0 sürümünden itibaren desteklenirken, OpenGL ES 2.0 desteği, Android 2.2 sürümünden itibaren verilmeye başlanmıştır [11].

2.2 OpenGL ES 1.0/1.1

OpenGL ES 1.0/1.1, orijinal OpenGL API' sinde bulunan birçok fonksiyon çıkarılarak oluşturulmuştur. OpenGL ES 1.0, OpenGL 1.3 özelliklerine bağımlı olarak tanımlanırken, OpenGL ES 1.1 ise OpenGL 1.5 özelliklerine bağımlı olarak tanımlanmıştır [10]. OpenGL ES 1.0 sürümü ile tam olarak uyumlu olan OpenGL ES 1.1 sürümünde, donanım hızlandırması üzerinde durulmuştur. Geliştirilmiş işlevsellik, görüntü kalitesi ve arttırılmış performans sunmanın yanı sıra hafıza kullanımını azaltarak güç tasarrufu da sağlamaktadır. Sabit fonksiyon ünitesi (fixed function pipeline - FFP) kullanılmaktadır [8]. FFP, bir kaplama ünitesidir (rendering pipeline). FFP, cihaz

tarafından sağlanan algoritmaları kullanarak kaplama işlemini gerçekleştirmektedir. Ancak bu algoritmalar sabittir ve yenileri programcı tarafından eklenememektedir. FFP, günümüzde yerini yavaş yavaş gölgelendirici (shaders) gibi programlanabilir ünitelere bırakmaktadır [13].

2.3 OpenGL ES 2.0

OpenGL ES 2.0, OpenGL 2.0 özelliklerine bağımlı olarak tanımlanmak ile birlikte programlanabilir 3D grafik ünitesi üzerinde durmaktadır. Programlanabilir donanım için geliştirilmiştir [8]. Programlanabilir ünite (programmable pipeline), sahnelemenin nasıl olacağına, programların (shaders) karar verdiği bir kaplama ünitesidir [14]. OpenGL ES Shading Language (gölgeleme dili) kullanılarak köşe (vertex) ve parça (fragment) gelgelendiriciler ile nesneler oluşturulabilmektedir. OpenGL ES 2.0, OpenGL ES 1.0/1.1' de bulunan sabit fonksiyon dönüşümleri ile parça fonksiyonlarını desteklememektedir [8].

2.4 OpenGL ES Sürümleri Karşılaştırması

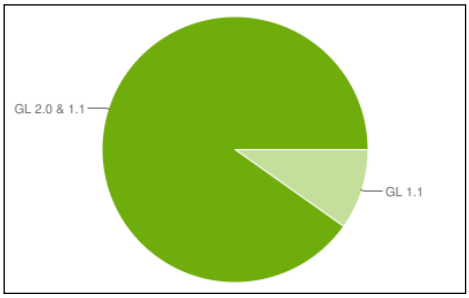
OpenGL ES'in tüm sürümleri ile yüksek performanslı 2D/3D grafikler ve animasyonlar gerçekleştirilebilmektedir. Grafik programlama açısından bakıldığından OpenGL ES 1.0/1.1 sürümleri ile 2.0 sürümü arasında önemli farklar bulunmaktadır. Bu nedenle, geliştiricilerin kullanacakları sürümü seçerken aşağıda belirtildiği gibi, göz önünde bulundurmaları gereken konular vardır [11]:

Performans: OpenGL ES 2.0 sürümü genel olarak 1.0/1.1 sürümüne göre daha yüksek performans sunmaktadır. Ancak bu durum, kullanılan cihaza göre de farklılık gösterebilmektedir.

Cihaz Uyumluluğu: OpenGL ES sürümünün cihazlara ve Android sürümüne uygunluğunun göz önünde bulundurulması gerekmektedir. OpenGL ES 2.0 sürümünü desteklemeyen cihazlar ve Android sürümleri mevcuttur. 3 Kasım 2011' de sona eren 7 günlük periyot boyunca toplanan veriye göre, Android işletim sistemli cihazların yaklaşık %90.2' si OpenGL ES 1.0/1.1 ve 2.0 versiyonlarını desteklerken, %9.8' i sadece OpenGL ES 1.0/1.1 versiyonlarını desteklemektedir (Şekil 2).

Kodlama: OpenGL ES 1.0/1.1 sürümleri ile 2.0 sürümü arasında kodlama açısından büyük farklar bulunmaktadır. OpenGL ES 1.0/1.1 sürümleri, sabit fonksiyon ünitesi ve diğer yardımcı fonksiyonları barındırır. OpenGL ES 2.0 sürümünde ise bu fonksiyonlar bulunmamaktadır. Bunların yerine gölgelendirici kullanılmaktadır. Bu nedenle OpenGL ES ile programlamaya yeni başlayan geliştiriciler, 1.0/1.1 sürümleri ile kodlama yapmayı, 2.0 sürümüne göre daha hızlı ve kolay bulabilirler.

Grafiklerin Kontrolü: OpenGL ES 2.0, gölgelendiricilerin kullanımı ile tam olarak programlanabilir ünite ve buna bağlı olarak grafikler üzerinde daha yüksek kontrol sağlamaktadır. Grafik işleme ünitesi üzerinde direkt kontrol sağlanmasıyla, programcılar, 1.0/1.1 ile gerçekleştirilmesi oldukça güç olan efektleri yaratabilmektedirler.



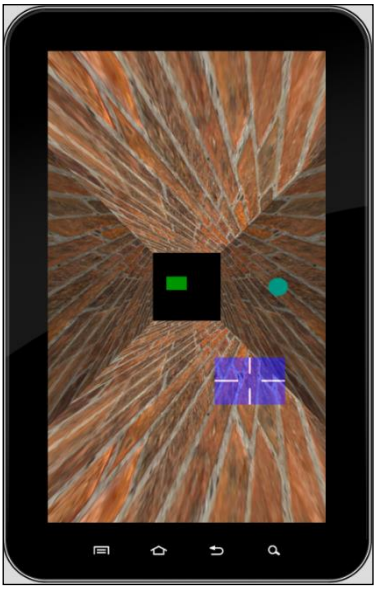
Şekil 2: Android kullanan cihazların OpenGL ES sürümlerini destekleme yüzdeleri [15]

3. Android Üzerinde OpenGL ile Oyun Geliştirme

İlk video oyunlarından olan Pong'un 3 boyutlu bir sürümü Android ortamında geliştirilmiştir (Şekil 3). Bir çeşit tenis oyunu olan Pong, 2 oyuncu tarafından bir adet top ile karşılıklı olarak oynanmaktadır. Oyun ortamı, sınırları duvarlar ile belirlenmiş bir dikdörtgenler prizması olarak betimlenebilir. Bu prizmanın iki ucunda birer adet oyuncu bulunmaktadır. Her bir oyucunun sahip olduğu dikdörtgen şeklinde bir raketi vardır. Oyuncular raketleri hareket ettirerek kendilerine doğru gelen topu karşılamaya çalışırlar. Kullanıcı kontrolü dokunmatik ekran ile sağlanan bu oyunda raketler, sağ-sol, yukarı-aşağı ve istenilen tüm yönlerde hareket ettirilebilmektedir. Rakete çarpan top karşı oyuncuya doğru yol almaya başlar. Topun yönü, rakete ve duvarlara çarpmasıyla birlikte geliş açısına bağlı olarak değişmektedir. Temaslar, çarpışma tespiti yöntemleri ile belirlenerek topun temel yansıma kurallarına uygun olarak yön değiştirmesi sağlanmaktadır.

Oyuncunun amacı, kendine doğru gelen topu karşılamak ve rakibe doğru tekrar yönlendirmektir. Bir oyuncu, kendine gelen topu karşılayamaz ise, rakip oyuncu sayı kazanır. En çok sayı kazanan oyuncu, oyunun galibi olarak belirlenir. Oyunda, kolay, orta ve zor olmak üzere 3 farklı zorluk seviyesi bulunmaktadır.

Grafiklerin gerçekleştiriminde OpenGL ES 1.0 teknolojisi kullanılmıştır. OpenGL' den farklı olarak OpenGL ES' de dikdörtgen, daire, küre gibi hazır şekiller bulunmamaktadır. Bu nedenle dikdörtgen şeklinde olan raketlerin ve duvarların çizimleri 2 adet üçgenin birleştirilmesi ile gerçekleştirilmiştir.



Şekil 3: Geliştirilen Mobil Cihaz tabanlı oyundan bir sahne

Küre şeklinde olan topun çizimi için de trigonometrik fonksiyonlar yardımıyla noktalar belirlenmiş ve bu noktalar birleştirilerek çizim gerçekleştirilmiştir. Daha gerçekçi bir görüntü vermek amacıyla, duvarlara desen kaplama yöntemi uygulanmıştır. Oyuncunun raketi, görüş açısının kısıtlanmaması için yarı saydam olarak tasarlanmıştır. Bunu sağlamak için "alpha blending" kullanılmıştır. "Alpha Blending" yönteminde, yarı-saydam ön plan rengi, arka plan rengi ile birleştirilerek harmanlanmış yeni bir renk oluşturulmaktadır [12].

Rakip oyuncunun hareketi, basit bir yapay zekâ algoritması tarafından belirlenmektedir. Bu algoritma, topun geliş yönüne göre hesaplamalar yaparak uygun hareket yönünü tayin etmektedir. Oyunun zorluk düzeyi arttıkça, topun hızı, yapay zekânın tepki hızı ve başarısı da artmaktadır. Kullanıcı ise, dokunmatik ekran yardımı ile raketini sürükleyerek hareket ettirebilmektedir. Oyuncunun ekran ile etkileşimleri Android işletim sistemi tarafından hazır olarak

sunulan bir arayüz ile belirlenerek geliştiricinin kullanımına sunulmaktadır. Bu konu ile ilgili yapılan işlemler aşağıdaki kod parçası üzerinde incelenebilir (Tablo 1).

4.Sonuç

OpenGL ES, mobil cihazlar üzerinde iki ve üç boyutlu grafikler içeren yazılımlar ve oyunlar geliştirmek için iyi bir altyapı sunmaktadır. Teknolojinin ilerlemesi ile, Mobil Cihazlar üzerinde daha hızlı, gerçekçi ve karmaşık sahnelere sahip grafik yazılımları geliştirilebilecektir. OpenGL ES ve benzeri grafik arayüzlerinin gelişimi ve kullanımının yaygınlaşması ile 3D grafik içeriği, mobil yaşamın da ayrılmaz bir parçası durumuna gelebilecektir. Üç boyutlu etkileşimli grafikler mobil cihazlarda da oyunlar ve eğlence sektörünün yanında, eğitim yazılımlarından tıp ve mühendislik uygulamalarına, simulasyonlardan arttırılmış gerçeklik uygulamalarına kadar birçok alanda kullanım potansiyeline sahiptir ve yazılım kalitesini, anlaşılırlığını artırmaktadır.

OpenGL ES programlamaya başlayan eski OpenGL programcılarının da, yeni ortama uyum sağlamak ve farklılıkları dikkate almak için belirli bir zaman ayırmaları gerekmektedir. OpenGL ES, hazır birtakım iki ve üç boyutlu nesneleri içermemektedir. Yazılım geliştirirken, mobil ortamın getirdiği performans sınırlamaları da dikkate alınmalıdır. Bu alanda sürümler ilerledikçe kullanım kolaylığının ve performansın artırılması, örneklerin çoğalması programcıların işini ve öğrenim sürecini kolaylaştıracaktır.

OpenGL ES ile programlamaya yeni başlayan geliştiriciler 1.0/1.1 sürümlerini kullanarak uygulama geliştirmeyi, 2.0 sürümüne göre daha hızlı ve kolay bulabilirler. OpenGL ES 1.0/1.1' de hazır olarak bulunan bazı çizim ve manipulasyon fonksiyonları OpenGL ES 2.0 da bulunmamaktadır. OpenGL ES 2.0' da bu hazır fonksiyonların işlevleri geliştirici tarafından gölgeleyiciler aracılığı ile yapılmaktadır. Bu durum beraberinde daha fazla kod yazmayı getirmektedir. Ancak gölgeleyicilerin OpenGL ES 2.0' da kullanımı ile, OpenGL ES 1.0/1.1' e oranla daha yüksek performanslı grafik uygulamaları elde etmek mümkündür. Bunun yanında OpenGL ES 2.0 sürümü, nesneler üzerinde daha yüksek kontrol sağlamaktadır. OpenGL ES 2.0 ile, OpenGL ES 1.0/1.1' de gerçekleştirmesi oldukça zor olan efektler yaratılabilir.

İşletim sistemlerinin ve cihazların desteklemesi ile birlikte OpenGL ES 2.0 sürümünün kullanımı son yıllarda artış göstermiştir. Android işletim sisteminin 2.2 sürümünden itibaren OpenGL ES 2.0' ı desteklemeye başlaması ile birlikte, yeni geliştirilen Android işletim sistemli cihazların çok büyük bir kısmı da OpenGL ES 2.0 desteği vermeye başlamıştır. Günümüzde Android işletim sistemine sahip cihazların yaklaşık %90.2' si OpenGL ES 2.0' ı desteklemektedir. Yeni geliştirilen uygulamalarda OpenGL ES 2.0 kullanılması tavsiye edilmektedir[11].

Kaynaklar

[1] Open Handset Alliance, http://www.openhandsetalliance.com/

[2] Wikipedia, http://en.wikipedia.org/wiki/Android_(operating_system)

[3] Android, http://developer.android.com/guide/basics/what-is-android.html

[4] The Android Story, http://www.xcubelabs.com/the-android-story.php

[5] Wikipedia, http://en.wikipedia.org/wiki/Android_version_history

[6] Wikipedia,
http://en.wikipedia.org/wiki/Computer_graphics

[7] Pulli, K., 'The Rise of Mobile Graphics',
http://www.iqmagazineonline.com/magazine/pdf/v_3_1_pdf/iss5(Pg14-15).pdf

[8] Opengl ES,
http://www.khronos.org/opengles/

[9] AndroidZoom,
http://blog.androidzoom.com/docs/2010-AndroidMarketZOOM-byAndroidzoom_ENG.pdf

[10] Wikipedia,
http://en.wikipedia.org/wiki/OpenGL_ES

[11] Android,
http://developer.android.com/guide/topics/graphics/opengl.html

[12] Fastgraph,
http://www.fastgraph.com/help/alpha_blending.html

[13] Tech-Artists,

http://tech-artists.org/wiki/Fixed-Function_Pipeline

[14] Tech-Artists,

http://tech-artists.org/wiki/Programmable_Pipeline

[15] Android,
http://developer.android.com/resources/dashboard/opengl.html