

## CLDC Konfigürasyonuna Sahip Mobil Cihazlarda

### 3B Grafiklerin Görselleştirilmesi

Oğuz Dönmez<sup>1</sup>, Aybars Uğur<sup>2</sup>, Ata Önal<sup>2</sup>

<sup>1</sup> Ege Üniversitesi, Tire Kutsan MYO, Bilgisayar Teknolojisi ve Programlama Programı 35900, İzmir

<sup>2</sup> Ege Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, İzmir

oguz.donmez@ege.edu.tr, aybars.ugur@ege.edu.tr, ata.onal@ege.edu.tr

**Özet:** Bu çalışmada, CLDC konfigürasyonuna sahip mobil cihazlar üzerinde M3G arayüzü ile bir 3B grafik gösterim uygulaması yapılmıştır. İlk olarak konuyla ilgili literatürde bulunan bilgiler verilmiş daha sonra geliştirilen uygulama anlatılmıştır. Geliştirilen uygulama ile, iki değer için iki kategori için 3B sütun türü grafik ile gösterilmesi gerçekleştirilmiştir. Bu uygulama ikiden fazla değer için ikiden fazla kategori için farklı grafik türleriyle gösterilmesi sağlanarak geliştirilebilir.

**Anahtar Kelimeler:** Mobil Teknolojiler, Görselleştirme, J2ME, MIDP, M3G.

#### Visualizing 3B Charts on CLDC-Configured Mobile Devices

**Abstract:** In this study, 3D graphical representation application on cldc-configured mobile devices has been done using M3G API. First related information in literature has been specified then implemented application has been described. By implemented application, two values in two categories has been represented with 3D column chart. This application can be improved by representing more values in more categories with different chart types.

**Keywords:** Mobile Technologies, Visualization, J2ME, MIDP, M3G.

#### 1. Giriş

Cep telefonu ve palm gibi CLDC konfigürasyonuna sahip mobil cihazlar sınırlı kaynaklara ve sınırlı ekran büyüklüğüne sahip olduğu için bu cihazlarda 3B grafiklerin gösterilmesi önemli araştırma konularından biridir.

CLDC konfigürasyonuna sahip mobil cihazlar için Sun'ın ürettiği Java versiyonu olan J2ME, uygulama geliştirmede kullanılabilir. Mobil cihazlar için geliştirilen MIDP uygulamalarında 3B içerik eklemek için M3G API kullanılabilir.

Bu makalenin 2., 3. ve 4. bölümlerinde sırasıyla J2ME, MIDP ve M3G hakkında bilgiler verilmektedir. Bölüm 5'te ise, geliştirilen uygulama kısaca tanıtılmakta ve Bölüm 6'da örnek bir senaryo üzerinde anlatılmaktadır. Bölüm

7'de ise yapılan çalışma sonunda elde edilen sonuçlar verilmiştir.

#### 2. J2ME (JAVA 2 Micro Edition)

Java 2 Micro Edition(J2ME); PDA'lar, cep telefonları ve diğer elektronik ve gömülü cihazlar gibi sınırlı donanım kaynaklarına sahip makineler için Sun'ın ürettiği Java versiyonudur. 128 KB RAM gibi az belleğe sahip ve tipik masaüstü bilgisayarlarda kullanılan işlemcilerden çok az güce sahip makinelerde kullanılmak üzere üretilmiştir. J2ME, profiller kümesi içerir. Her profil belirli tipteki cihaz(cep telefonları, PDA'lar, mikro dalga fırınlar) için tanımlanmıştır. Belirli tipteki cihaz için gerekli minimum sınıf kütüphanesini ve cihazı desteklemek için Java sanal makinesinin bir belirtimini içerir. Herhangi profildeki sanal makine-

nin Java 2 Standard Edition(J2SE) ve Java 2 Enterprise Edition(J2EE)'de kullanılan sanal makineyle aynı olmasına gerek yoktur[1].

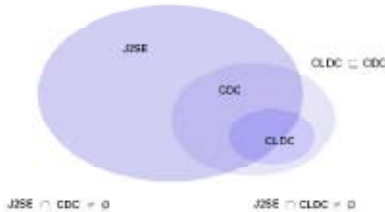
Sun'ın yayınladığı profiller [1]:

- Foundation Profile: Yeni nesil elektronik cihazlar için profil. CDC temelli ve grafik arayüzü olmayan cihazlar içindir.
- Mobile Information Device Profile(MIDP): hücrel telefon ve PDA'lar gibi mobil bilgi cihazları için profil. CLDC temellidir.
- Personal Basis, Personal and RMI Profile: CDC&Foundation Profile temelli cihazlar için temel grafik ve RMI desteği sunar.

Profilin kendisi hiç bir şey yapmaz; sadece belirtimi tanımlar. Profiller konfigürasyonla gerçekleştirilirler Mevcut konfigürasyonların bazıları [1]:

- Connected Device Configuration(CDC): Yeni nesil gömülü ve elektronik cihazlar için Foundation Profile'in gerçekleştirimidir.
- Connected Limited Device Configuration (CLDC): Palm cihazları, cep telefonları gibi küçük, kaynak sınırlı cihazların MIDP gerçekleştirimidir.

Şekil 1, CDC ve CLDC arasındaki ilişkiyi göstermektedir. Aynı zamanda onların J2SE API'si ile ilişkilerini de göstermektedir. CDC, bazı ekstra sınıflar olduğu J2SE'nin alt kümesidir. CLDC, CDC'nin bir alt kümesidir [2].



Şekil 1. J2SE, CDC ve CLDC karşılaştırması

KVM(Kuauı VM), Java sanal makinesinin tamamen yeni gerçekleştirimidir, küçük ci-

hazlarda kullanılmak için optimize edilmiş bir gerçekleştirimdir. KVM, klasik sanal makineyle aynı bytecode kümesini ve aynı sınıf-dosya biçimini kabul eder [3].

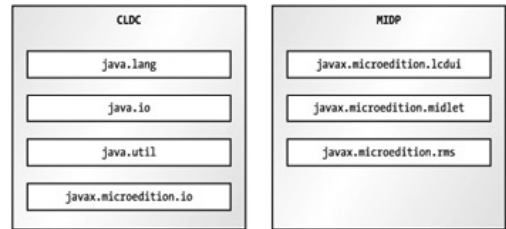
KVM'nin özellikleri[3]:

- Long integer ve floating point veri tipleri seçimlidir.
- Garbage collection'da nesne sonlandırma yoktur.
- JNI desteği yoktur (KVM'nin içindeki native metotlar sanal makinenin içine derlenir, kullanıcı tarafından yüklenemez).
- Cihazı kullanmadan sınıf geçilemez.
- Çok boyutlu diziler seçimlidir.
- Kullanıcı tanımlı sınıf yükleyicileri yoktur (Sadece KVM tarafından desteklenen sistem sınıf yükleyicisi mevcuttur).

### 3. MIDP:

Bir MIDP uygulaması için mevcut API'ler CLDC ve MIDP paketlerinden gelir (Şekil 2). CLDC, J2SE dünyasından alınmış API'nin özünü oluşturur [4].

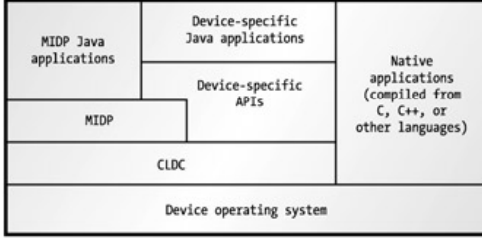
- Dil sınıfları java.lang'ta
- Akım sınıfları java.io'da
- Basit koleksiyonlar java.util'de
- Javax.microedition.io'da genelleştirilmiş network API'si bulunur.



Şekil 2. MIDP API'yi oluşturan paketler

MID(Mobile Information Device)'ler farklı türden ürünleri çalıştırabilir. Şekil 3 olasılıkları gösterir. Her cihaz belirli türden işletim sistemini gerçekleştirir. Native uygulamalar bu katmanın üzerinden direk olarak çalışır. Cihaz

işletim sisteminin üzerinde CLDC ve MIDP API'leri vardır. Cihaza özgü java uygulamaları cihaz satıcıları tarafından sağlanan java API'lerini kullanabilir [4].



Şekil 3. MID'lerin çalıştırıldığı ürünler

MIDP'in avantajları [4]:

Taşınabilirlik: MID uygulama geliştirme için diğer araçları kullanma yerine Java'yı kullanmanın avantajı taşınabilirliktir.

Güvenlik: Java; appletler gibi indirilmiş kodu güvenli bir şekilde çalıştırma yeteneğine sahiptir.

Mobil cihazlar için Java uygulamalarına MIDlet denir. J2ME platformu için uygulama geliştirmek için JDK(Java Development Kit) ve J2ME Wireless Toolkit'in bilgisayara kurulması gerekir. J2ME WTK, MIDlet yaratmak için önemli araçları içerir. MIDlet geliştirme ortamını ve 3D ve Mobile Media uygulamaları gibi seçilebilir kütüphaneler için gerekli seçilebilir paketleri sağlar. Ayrıca MIDlet'leri imzalayarak uzak mobil cihaza kurulmadan önce onların onaylanmasını sağlar [5].

MIDlet Geliştirme aşamaları [2]:

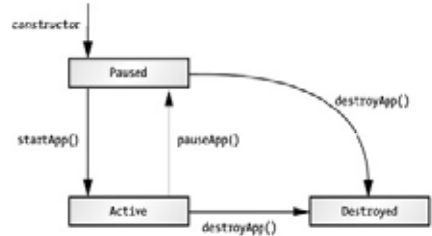
- Java uygulamanı yaz.
- Uygulamayı derle.
- Ön-geçerliliğini(pre-verify) yap.
- JAR dosyasına paketle
- Uygulama tanımlayıcısını yarat.
- MIDlet suitini yayımla.

Bir MIDlet, sınıf dosyaları cihaza taşınarak yüklenir. Sınıf dosyaları, Java Archive(JAR)'a

paketlenirken eşlik eden tanımlayıcı dosya(.jad uzantılı) JAR'ın içeriğini tanımlar [4].

Bir MIDlet yaşam döngüsü boyunca aşağıdaki durumlardan geçer(Şekil 4)[4]:

- MIDlet çalıştırılmak üzereyken, bir örnek yaratılır. MIDlet'in constructor'ı çalıştırılır ve MIDlet *Paused* durumundadır.
- Sonra, uygulama yöneticisi startApp()'ı çağırdıktan sonra MIDlet *Active* durumuna girer.
- MIDlet *Active* iken, uygulama yöneticisi pauseApp()'ı çağırarak çalışmasını durdurabilir. Bu MIDlet'i yeniden *Paused* durumuna koyar. notifyPaused() çağırarak MIDlet kendisini *Paused* durumuna geçirebilir.
- destroyApp() çağırarak, uygulama yöneticisi MIDlet'in çalışmasını sona erdirebilir. MIDlet yok edildiğinde garbage collection beklenir. notifyDestroyed() çağırarak MIDlet kendini yok edebilir.



Şekil 4. MIDlet Yaşam Döngüsü

Her MIDlet, javax.microedition.midlet paketinde bulunan soyut MIDlet sınıfını extend etmelidir. MIDlet'in soyut sınıftaki en az 3 metodu override etmesi gerekir, startApp(), pauseApp() ve destroyApp() [5].

#### 4. M3G:

Mobile 3D Graphics(M3G) API, Java Specification Request(JSR) 184'te tanımlanmıştır. Java programlamayı destekleyen mobil cihazlar için 3D API standardı yaratma çabasıdır. M3G sınıfları javax.microedition.m3g paketinde bulunur [7].

M3G, geometri ve diğer nesnelere hiyerarşi içinde organize eder bu nedenle sahne çizgesi tabanlı denir. Sahne çizgesi tabanlı olmasının nedeni, her düğüme dönüşüm uygulayabilmektir. Bu da hiyerarşik animasyon yapabilmemizi sağlar [8].

M3G, Java 3D değildir. Java 3D standart java platformunun yeteneklerini kullanır. Ve mobil cihazlardan daha fazla bellek ve işleme gücüne sahip PC'ler için tasarlanmıştır. M3G ve Java 3D farklı amaçlarla tasarlanmış ayrı ve uyumsuz API'lerdir. M3G, geliştiricilere 3B grafikleri çizmek için iki yol sağlar: *immediate mode* ve *retained mode* [9].

- Immediate mode'da, grafik komutları direk boru hattına yayımlanır ve kaplama motoru onları hemen çalıştırır. Bu yöntem kullanılırken; geliştirici, kaplama motoruna her animasyon çerçevesinde ne çizeceğini belirtmelidir [9].
- Retained mode, bütün geometrik nesnelere 3B dünyada ağaç yapısında bağlayan sahne çizgesi kullanır. Geometrik yapı, konum ve görünüm gibi her nesne hakkında yüksek seviye bilgi çerçeveden çerçeveye tutulmaktadır [9].

M3G'den faydalanabilecek uygulama alanları [8]:

- Oyunlar
- Harita görselleştirme
- Kullanıcı arayüzleri
- Canlandırılmış Mesajlar
- Ekran koruyuculardır.

Bu ihtiyaçları karşılamak için API hem yüksek seviye hem de düşük seviye grafik özelliklerini destekler [8].

## 5. Geliştirilen Uygulamanın Tanıtılması

Cep telefonuna yüklenecek JAR dosyası iki sınıftan oluşur: DGraphics ve ExampleCanvas. DGraphics, kullanıcının grafiklerle ilgili bilgileri girdiği formların görüntülenmesini ve grafiği

görseltirecek olan *canvas*'ı çağıran sınıftır. ExampleCanvas, girilen bilgilere göre 3B grafiği oluşturup görüntüleyecek sınıftır.

Geliştirilen uygulama MIDP uyumlu ve M3G API'yi destekleyen cep telefonlarına yüklenerek çalıştırılabilir.

## 6. Örnek Senaryo

Uygulama çalıştırıldığında ilk olarak kullanıcının grafiklerle ilgili bilgileri girmesi için kullanılan formlar görüntülenir. İlk veri girişi formu şekil 5'te gösterilmiştir.

Veri Giriş1

Baslik

1.Seri ismi

2.Seri ismi

1.Kategori ismi

2.Kategori ismi

Tamam

Şekil 5. Veri Giriş1 Formu

Veri Giriş2

Kamyon

Veri1

Veri2

Otobüs

Veri1

Veri2

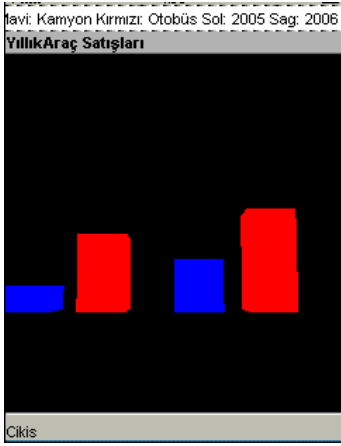
Veri1

Menu

Şekil 6. Veri Giriş2 Formu

Bu forma kullanıcı grafik başlığı, seri isimleri ve kategori isimlerini girer. Bu bilgiler girildikten sonra ilgili kategoriler için ilgili verilerin girildiği ikinci veri giriş formu görüntülenir(Şekil 6).

İkinci veri giriş formuna da ilgili bilgiler girildikten sonra grafiğin bulunduğu *canvas* görüntülenir (Şekil 7). Grafik başlığı dışındaki grafikte ilgili bilgi *ticker*'da görüntülenir.



Şekil 7. Grafiğin gösterildiği *canvas*

Dikdörtgenler prizması çizdirmek için ilk olarak köşelerin koordinatları tanımlanmış, daha sonra köşelerin renkleri tanımlanmıştır. Bu renk ve koordinatlar için köşe dizinleri oluşturulmuş, köşe tamponuna bu renk ve koordinatlar yüklenmiştir. Bu köşe tamponu kullanılarak ağ oluşturulup *group* sınıfından oluşturulmuş nesneye eklenmiştir. Şekil 8, bu işlemleri gerçekleştiren kodu göstermektedir.

```
._VertexData4=new VertexBuffer();
VertexArray koseler4=new VertexArray(renkler2.length/3,3,1);
koseler4.set(0,renkler2.length/3,renkler2);
._VertexData4.setColors(koseler4);
VertexArray koseler4=new VertexArray(koseler4.length/3,3,2);
koseler4.set(0,koseler4.length/3,koseler4);
._VertexData4.setPositions(koseler4,1.0f,null);
._Triangles=new TriangleStripArray(indices,new int[] {indices.length});
seri4=new Mesh(_VertexData4._Triangles,(Appearance));
grup.addChild(seri1);
grup.addChild(seri2);
grup.addChild(seri3);
grup.addChild(seri4);
```

Şekil 8. Dikdörtgenler prizması oluşturup ekleyen kod

## 7. Sonuç

Bu çalışmada, J2ME, MIDP ve M3G kullanılarak CLDC konfigürasyonuna sahip cihazlarda 3B içerik görüntüleme uygulaması gerçekleştirilmiştir.

Bu uygulama ile iki nesnenin verilerinin iki kategoride 3B sütun grafik türünde gösterimi gerçekleştirilmiştir.

Bu uygulama, ikiden fazla nesnenin verilerinin ikiden fazla kategoride farklı grafik türlerinde gösterimi ile geliştirilebilir.

M3G teknolojisinin, sahne çizgesi yapısı ile mobil cihazlar üzerindeki üç boyutlu grafik işlemlerini oldukça kolaylaştırdığı ve hızlandırdığı görülmektedir. Bu tür API'lerdeki gelişmeler, üç boyutu, mobil cihazların da ayrılmaz bir parçası haline getirecektir.

## 8. Kaynaklar

[1] Tauber D. A., "What is J2ME", [www.onjava.com/pub/a/onjava/2001/03/08/J2ME.html](http://www.onjava.com/pub/a/onjava/2001/03/08/J2ME.html), 2001.

[2] "J2ME: Step by step", [www.digilife.be/quickreferences/PT/J2ME%20Step%20by%20step.pdf](http://www.digilife.be/quickreferences/PT/J2ME%20Step%20by%20step.pdf).

[3] Giguere E., "Java 2 Micro Edition: Professional Developer's Guide", John Wiley & Sons, 2000.

[4] Knudsen J., "Wireless Java: Developing with Java 2, Micro Edition", Apress, 2001.

[5] Goyal V., "J2ME Tutorial, Part 1: Creating MIDlets", <http://today.java.net/pub/a/today/2005/02/09/j2me1.html>, 2005.

[6] Day B., "Codecamp: Developing Wireless Applications using MIDP 2.0, WMA and MMA", [developers.sun.com/events/techdays/presentations/seattle/odecampWirelessApplicationsusingMIDP20.pdf](http://developers.sun.com/events/techdays/presentations/seattle/odecampWirelessApplicationsusingMIDP20.pdf), 2004.

[7] Höfele C., “3D Graphics for Java mobile devices, Part1: M3G’s immediate mode”, <http://www-128.ibm.com/developerworks/library/wi-mobile1/index.html>, 2005.

[8] Hasselgren J., “M3G-Overview”, [www.cs.lth.se/EDA075/lectures/L3.pdf](http://www.cs.lth.se/EDA075/lectures/L3.pdf), 2006.

[9] Mobile 3D Graphics API - Wikipedia, the free encyclopedia, [http://en.wikipedia.org/wiki/JSR\\_184](http://en.wikipedia.org/wiki/JSR_184), Wikipedia.

[10] Mahmoud Q. H., “Getting Started With the Mobile 3D Graphics API for J2ME”, <http://developers.sun.com/techtoc/mobility/apis/articles/3dgraphics/>, 2004.