

Akademik Biliřim '07

Dumlupınar Üniversitesi / Kütahya

Üç Boyutlu Binaların Web Üzerinde Otomatik Olarak JOGL ile Modellenmesi

Aybars UĞUR, Eray HANGÜL, Tahir Emre KALAYCI, Dođan AYDIN

Sunan : Eray HANGÜL erayhangul@gmail.com

Ege Üniversitesi, Bilgisayar Mühendisliđi Bölümü, 35100, Bornova/İZMİR

Bu alıřmada biz ne yaptık ?

- Bu alıřmada **N** katlı herhangi bir binanın, *JOGL* ('*Java Open Graphics Library*') ile web tabanlı olarak üç boyutlu (3B) modellenmesini saęlayan bir yazılım geliřtirilmiřtir.
- Bina grselleřtirmeyi otomatikleřtiren bu uygulamadan elde edilen sonular, 3B ierik oluřturmada kullanılan teknikler bazında tartiřılarak irdelenmiřtir.
- Anahtar Kelimeler : **Grselleřtirme, Ü Boyutlu Bilgisayar Grafikleri, Web3D, Modelleme, JOGL**

JOGL Nedir ?

- *JOGL*, Java tabanlı *OpenGL* destekli grafiksel uygulamalar geliřtirmek için, açık kaynak kod projesi řeklinde geliřtirilip ortaya çıkarılmıř bir programlama arayüzüdür.
- *JOGL* kullanılarak oyunlar, etkileřimli eğitim amaçlı uygulamalar, grafiksel modelleme / tasarım yapılabilen editörler vb. geliřtirilmektedir.
- *JOGL*'un çalışma mantığı, *C* programlama dili ile yazılmıř *OpenGL* kütüphanesine ait yordamların, arka planda *JNI* ('*Java Native Interface*') kullanılarak çağırılması çerçevesi üzerine oturtulmuřtur.
- *JOGL*, *OpenGL* çağrılarını belirli bir kaç sınıfın içerdiği metodların prosedürel olarak çağırılması řeklinde kullanımına imkan vermekte; sonuçta bu tarz bir yaklařım daha hızlı bir řekilde grafiksel görüntüleme yapılmasına büyük katkı sağlamaktadır.

Java Tabanlı 3B İçerik Oluşturma Kütüphaneleri

- *Java* programlama dili ile üç boyutlu içerik oluşturmanın temelleri ilk defa 1996 yılında *Intel*, *Silicon Graphics*, *Apple* ve *Sun* firmalarının girişimiyle başlamıştır.
- 1998 yılı sonlarında *Java3D*'nin ortaya çıkışı ile başlayan süreç, 2000 yıllarında *GL4Java*'nın ve 2003 – 2004 döneminde *Java3D*'nin gelişimine ara verildiği sırada *JOGL*'un ortaya çıkışı ile farklı bir boyut kazanmıştır.

Java3D

- *Java*'da, 3D içerik oluşturmada, *JOGL* öncesi kullanılan en yaygın uygulama programlama arayüzü *Java3D*'dir. Günümüzde *Java3D* ile geliştirilmiş bir çok grafiksel uygulama mevcuttur.
- *Java3D*'nin çalışma mantığı, ağaç veri yapısında yer alan düğüm gruplarının genel görüntüleyici tarafından belirlenen hiyerarşik yapıya göre grafiksel içeriğin oluşturulması şeklinde belirlenmiştir. Bu yaklaşım, uygulama geliştiriciler açısından programlama seviyesinde kolaylık getirirken, uygulamaların çalışma hızının düşük olmasına sebep olmaktadır.
- *Java3D*'de grafiksel öğelerin oluşturulmasında *OpenGL*'de olduğu kadar gerçekçilik verilememektedir. *JOGL*, hız ve gerçekçiliğin arttırılması ve uygulamalardaki belirsiz hatalar ile uyumsuzlukların giderilmesi sonucu; *Java3D*'nin belirtilen eksikliklerini kapatarak, 3D içerik oluşturmada daha etkin bir çözüm durumuna gelmiştir.

LWJGL

- *JOGL* 'a alternatif olabilecek belli başlı uygulama programlama arayüzlerinden biri de *LWJGL* ('*Lightweight Java Game Library*')'dir.
- Her iki kütüphane de arka planda yapılacak bir çok işi kendi yapıları içinde çözümlenmektedirler ancak *LWJGL* kendi pencere sistemi üzerinden çalıştığından var olan *Java* uygulamaları ile birleştirilmesi pek de kolay değildir.
- *JOGL*, *Java*'nın *AWT*, *Swing* gibi bileşenleriyle rahatlıkla birleştirilerek kullanılabilir. *LWJGL* grafiksel bir uygulamayı geliştirmek için hazır bir paket gibi kullanılabilirken *JOGL* sadece grafiksel öğelerin görüntülenmesinde kullanılır.
- Örneğin ses ve ses efekti işlemleri için *JOAL*('*Java Open Audio Library*') ve girdi kontrolü (klavye, fare, oyun çubuğu vb.) için *JInput* kütüphaneleri kullanılarak bütünsel sonuca gidilebilir.

JOGL Kurulum

- **İndir** - <https://jogl.dev.java.net>
- **Kopyala** - **.dll* | **.so* - (*JRE*) Java Çalışma Ortamı/bin
- **Kopyala** - “jogl.jar” - (*JRE*)/lib/ext
- **Yaz ve Dene** - *System.loadLibrary(“jogl”);*
- **Kontrol Et** - “*UnsatisfiedLinkException*”

Web Tabanlı 3B Bina Modelleme



- 3B olarak modellenecek bir binanın var olan çizim veya modelleme programlarıyla herhangi bir birey tarafından ortaya çıkarılması çok kolay olmayan bir süreçtir.
- Binaların üç boyutlu olarak otomatik bir şekilde modellenmesinde prosedürel programlama yaklaşımının kullanımı son yıllarda önem kazanmaktadır.

Web Tabanlı 3B Bina Modelleme

– II –

Aşağıdaki soruların tatmin edici cevaplarının elde edilmesi prosedürel yaklaşımın doğruluğunu kanıtlayabilecek nitelikte olacaktır :

- **Bu tarz uygulamalar geliştirmek için hangi şekilde prosedürel programlama yapmak gerekir?**
- **Gerçek zamanlı bina oluşumunu sağlayabilmek için geliştirilen kapsamlı bir uygulama yeterince hızlı sonuç üretebilir mi?**
- **Mimari açıdan ne kadar doğru ve görsel açıdan ne ölçüde inandırıcı sonuçlar elde edebiliriz?**
- **Gerçek zamanlı görüntüleme prosedürel yaklaşım aynı anda bir çok binayı farklı detay seviyesinde görüntüleyebilecek kadar yetenekli olabilir mi?**

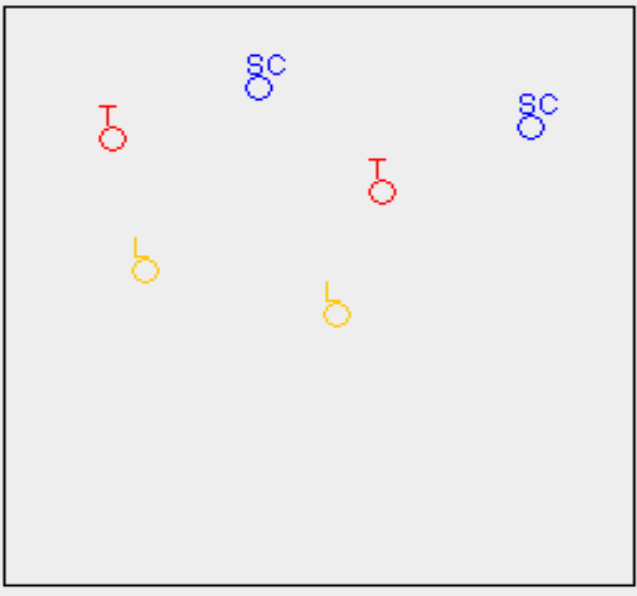
Çok Katlı 3B Bina Modelleme Çalışmamız

- Örneklenen çalışma ile 3B modelleme bilgisi olmayan herhangi bir bireyin; binanın kat sayısı, her kattaki asansör ve merdiven sayısı gibi temel parametreleri kullanıcı dostu bir arayüzden girerek, otomatik olarak 3B binayı oluşturabilmesi sağlanmıştır.
- Fare kullanımı ile binanın herhangi bir açıdan döndürülerek görüntülenebilmesi ve klavyeden yön tuşları kullanılarak binanın pozisyonunun güncellenebilmesi imkanları sunulmuştur.
- Kullanıcı verdiği parametrelerin değişmesine göre ne tip bir 3B bina modeli ortaya çıkabileceğini hızlı ve pratik bir şekilde görebilmektedir.
- Gelistirdiğimiz uygulamada, otomatik bina oluşturma yaklaşımı, önceki çalışmalardan farklı olarak; JOGL ile ilk defa web üzerinde uygulanmıştır.

Uygulama Parametre Girdi Ekranları

Building Parameters	
Floor Count	2
Floor Length	4
Floor Width	4
Floor Height	2

Kullanıcıdan *bina kat sayısı* ve her bir katın *uzunluk, en* ve *yükseklik* değerini belirtmesi beklenmektedir.

Building Parameters	Floor Parameters
	
0. Floor	

Kullanıcının girdiği temel parametrelerin dışında, her bir kat için istenildiği takdirde; belirtilen sayıda, kata ait hedef noktası, merdiven ve asansör yerleştirilebilmektedir.

Örneğin yukarıdaki şekilde binanın 1. katı için 'T' harfi ile kırmızı renkte belirtilen 2 adet eleman hedef noktasını, 'L' harfi ile turuncu renkte belirtilen 2 adet eleman asansörü ve 'SC' ile mavi renkte gösterilenler merdivenleri belirtmektedir.

Swing -> JOGL Normalizasyon

- Parametreler girildikten sonra binanın oluşturulması istendiğinde, öncelikle *Swing JPanel*'inde yer alan parametrelerin koordinat değerleri *JOGL*'un koordinat sistemine göre normalize edilerek pozisyon değerleri güncellenmektedir.

```
// Z eksenini için...
```

```
public float getZCoordinateValue(float floorWidth, float normalizedZ) {  
    float returnValue = -1;  
    returnValue = floorWidth * normalizedZ * 0.5f;  
    return returnValue;  
}
```

```
// X eksenini için...
```

```
public float getXCoordinateValue(float normalizedX, float floorLength) {  
    float returnValue = -1;  
    returnValue = Math.abs((floorLength/2) - (normalizedX * floorLength));  
    return returnValue;  
}
```

Görüntüleme Listesi Kullanımı ile Katların Çizdirilmesi

```
mainDisplayList = gl.glGenLists(1);  
  
gl.glNewList(mainDisplayList, GL.GL_COMPILE);  
  
for (int i = 0; i <= building.getFloorCount() - 1; i++) {  
    drawFloor(building.getFloor(i));  
  
}  
  
gl.glEndList();
```

Bir Katın Çizdirilmesini Örnekleyen Kod Bloğu

```
gl.glColor3d(Math.random(), Math.random(), Math.random());
```

```
// Poligon Çizici sınıfımız kullanılarak kat çizdiriliyor...
```

```
polygonDrawer.drawWireCube(building.getFloorLength(),  
                             building.getFloorHeight(),  
                             building.getFloorWidth());
```

```
gl.glPushMatrix();
```

```
gl.glTranslatef(0, building.getFloorThick()/2 - building.getFloorHeight()/2, 0);
```

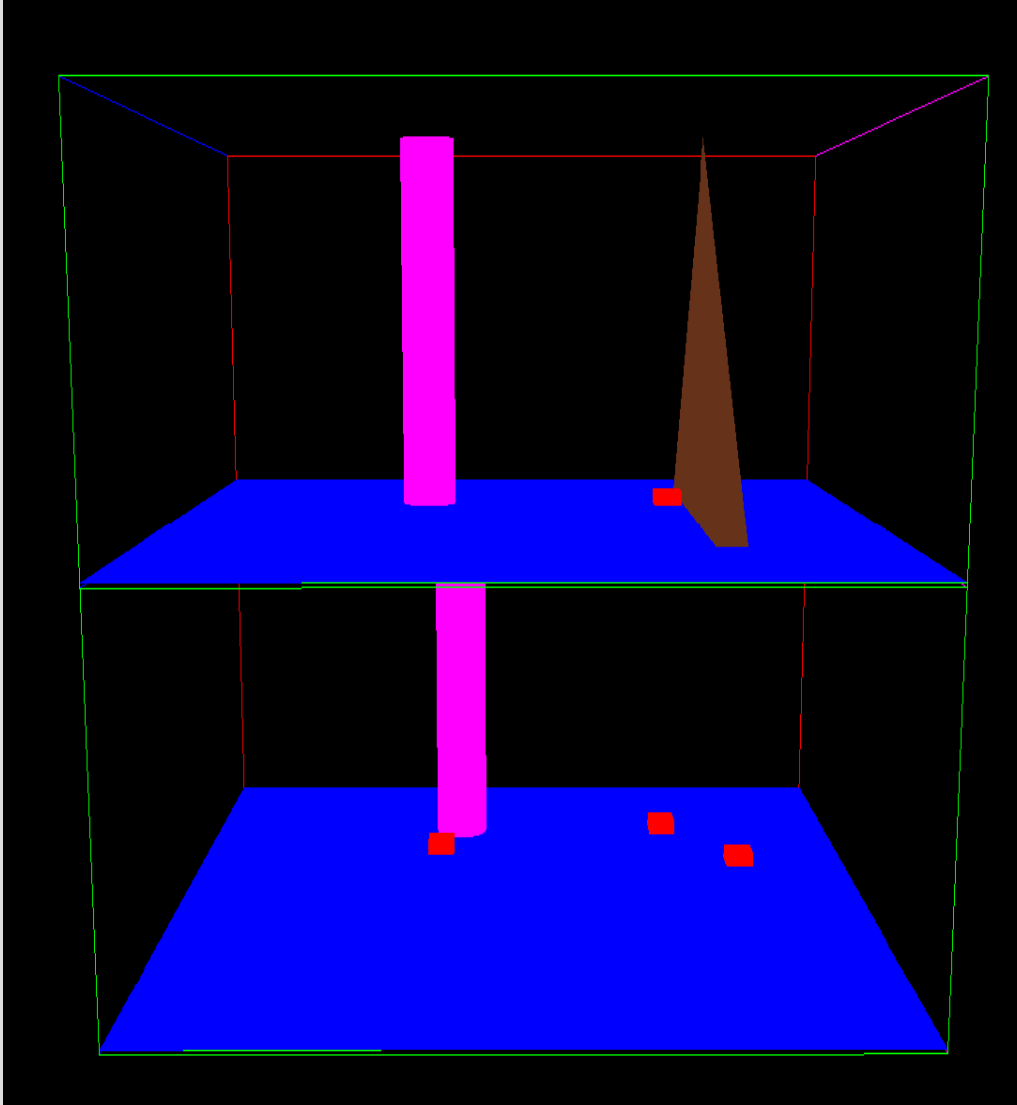
```
gl.glScalef(1, building.getFloorThick(), 1);
```

```
// Poligon Çizici sınıfımız kullanılarak kat tabanı çizdiriliyor...
```

```
polygonDrawer.drawSolidCube(building.getFloorLength(),  
                              building.getFloorThick(),  
                              building.getFloorWidth());
```

```
gl.glPopMatrix();
```

Uygulamanın Ürettiği Farklı Bina Görüntüleri – I –

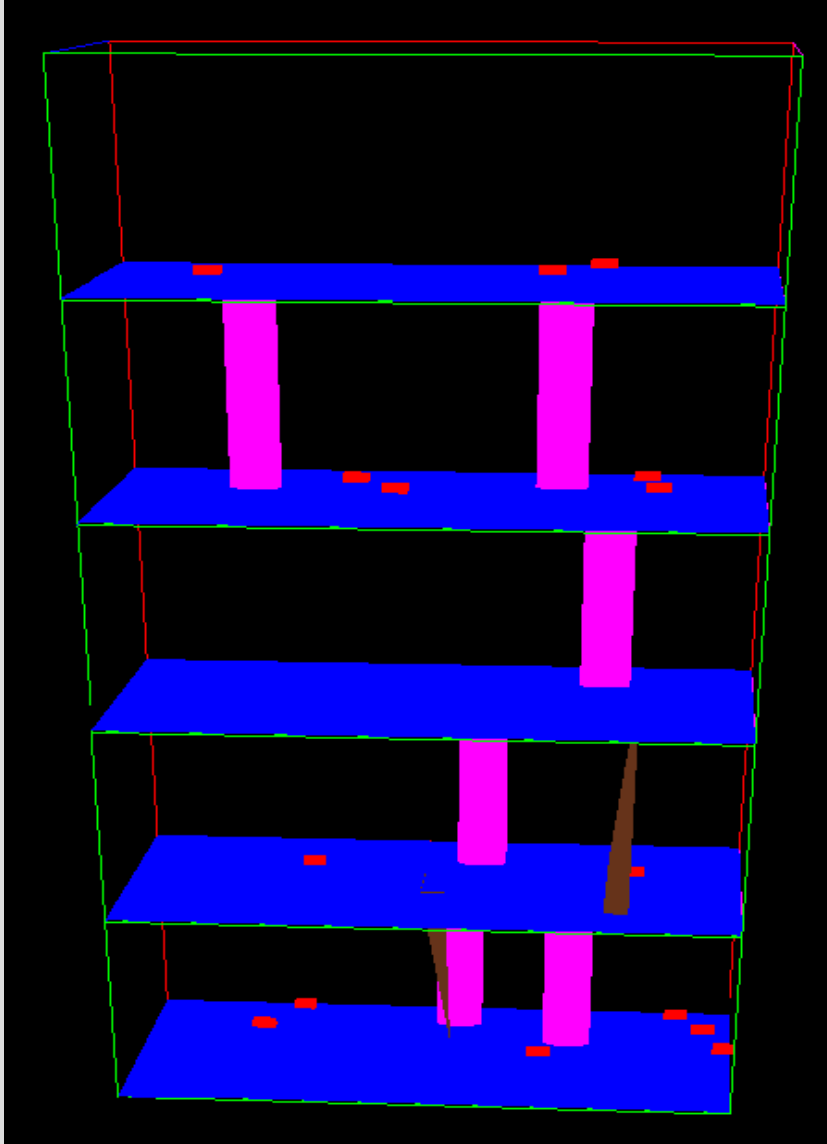


*2 Katlı Bina Model
Örneği*

(1. Kat : 3 adet
hedef noktası, 1
adet **asansör**)

(2. Kat : 1 adet
hedef noktası, 1
adet **asansör**,
1 adet **merdiven**)

Uygulamanın Ürettiği Farklı Bina Görüntüleri – II –



5 Katlı Bina Model Örneği

- (1. Kat : 6 adet **hedef noktası**, 2 adet **asansör**, 1 adet merdiven)
- (2. Kat : 2 adet **hedef noktası**, 1 adet **asansör**, 1 adet merdiven)
- (3. Kat : 1 adet **asansör**)
- (4. Kat : 4 adet **hedef noktası**, 2 adet **asansör**)
- (5. Kat : 3 adet **hedef noktası**)

Sonuçlar – I – JOGL, Java3D, OpenGL Kütüphanelerinin Karşılaştırılması

- Uygulama geliştirilirken *JOGL* dışındaki grafiksel içerik oluşturma kütüphaneleri ve teknikleri gerçekleştirim sırasında kullanılan teknikler bazında irdelenerek karşılaştırılmıştır.

	<i>JOGL</i>	<i>Java3D</i>	<i>OpenGL</i>
Grafiksel İçerik Kalitesi	<i>Çok İyi</i>	<i>İyi</i>	<i>Çok İyi</i>
Görüntüleme Hızı	<i>İyi</i>	<i>Orta</i>	<i>Çok İyi</i>
Programlama Kolaylığı	<i>İyi</i>	<i>Çok İyi</i>	<i>Orta</i>
Platform Uyumluluğu	<i>İyi</i>	<i>İyi</i>	<i>Çok İyi</i>
Web Yayınlama Kolaylığı	<i>Çok İyi</i>	<i>Çok İyi</i>	<i>Az</i>
Farklı Uygulamalarla Ortak Çalışabilme	<i>İyi</i>	<i>İyi</i>	<i>Çok İyi</i>

Sonuçlar – II – Uygulamaya Nasıl Erişebilirim ?

- Geliştirilen uygulama *Java*'nın '*Web Start*' teknolojisi kullanılarak internet üzerinden kullanılabilir hale getirilmiştir.
- <http://yzgrafik.ege.edu.tr/projects/MFVis> adresinden uygulama etkileşimli olarak kullanılabilir. Bu yaklaşım, çözümün daha hızlı ve kolay bir şekilde kullanıcıya sunulması bakımından önemlidir.
- Günümüzde bir çok uygulama web hatta mobil tabanlı çalışabilecek şekilde geliştirilerek bilgiye erişimdeki sınırların kalktığı örneklenmektedir.

Sonuçlar – III – JOGL'un Avantajları

- Gerçekleştirilen bu uygulama sonucunda *JOGL*'un web tabanlı üç boyutlu içerik oluşturmada gayet başarılı olduğu anlaşılmıştır.
- *OpenGL* ve *Java* ortamlarının birleştirilerek web üzerinden kullanılabilmesini sağlaması, bir çok projenin daha çok kişiye; çok daha hızlı bir şekilde ulaşabilmesini sağlayacaktır.
- *Java* geliştiricileri üç boyutlu grafiksel programlamayı rahatlıkla kullanabileceklerdir.

Sonuçlar – IV – Uygulama Avantajları ve Gelecek Çalışmalarımız

- Kullanıcıların, istedikleri bir bina modelini, internet üzerinden kolaylıkla oluşturabildikleri gözlemlenmiştir.
- Yazılım geliştiriciler açısından ise, *OpenGL* altyapısının kullanılması sayesinde çok karmaşık modeller hızlı bir şekilde oluşturularak, üzerlerinde işlemler yapılabildiği anlaşılmıştır.
- Çalışmamızda, desen kaplama (*Texture Mapping*), hazır 3B model yükleme (*3D Object Loading*) ve karmaşık aydınlatma (*Illumination Effects*) gibi modüllerin eklenmesi aşamasına gelinmiştir. Böylece uygulama, mimari ve görsel açıdan daha gerçekçi bina modellerini de destekleyecektir.

Yararlanılan Kaynaklar

[1]. Canroy, K., JOGL : A Beginner's Guide and Tutorial, 2 Eylül 2004.

[2]. Twilleagear, D., Kesselman, J., Goldberg, A., Petersen, D., Soto, C. J., Melissinos, C., Java Technologies for Games.

[3]. Ugur, A., Bilgisayar Grafikleri Ders Notları, EGE Üniversitesi, Bilgisayar Mühendisliği Bölümü, 2002 – 2003.

[4]. Wei, Y., Integrating web 2D and 3D technologies for architectural visualization: applications of SVG and X3D/VRML in environmental behavior simulation, Web3D 2006: 11th International Symposium on 3D Web Technology, 2006.

[5]. Whelan, G., Automatic Building Generation, Institute of Technology Blanchardstown, Graphics and Gaming Group, 2006.

[6]. Xu, Z., Yen, Y., Chen, X. J., OpenGL Programming in Java .

[7]. JOGL : Java Bindings for OpenGL, <https://jogl.dev.java.net>

[8]. Nehe Productions, <http://nehe.gamedev.net>

[9]. OpenGL, <http://www.opengl.org>, *SGI*

[10]. Sun Developer Network, <http://java.sun.com>, *Sun Microsystems*

