

# 3 BOYUTLU GÖRÜNTÜ DOSYALARI PROJESİ

Kubilay Kaplan - 180201123

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

kubilay.kaplan@windowslive.com

## ÖZET

Bu program; proje dizininde bulunan “.nkt” uzantılı dosyalardaki veriyi okuyarak, dosyalarda sıralanan üç boyutlu olarak ifade edilen noktalar ile proje isterlerine bağlı olarak çeşitli işlemler yapar ve kontrollere/işlemlere bağlı çıktıları da “.nkt” uzantılı dosyalara yazar.

## I. GİRİŞ

### 1) Proje Tanımı

Projede, proje dizininde bulunan “.nkt” uzantılı tüm dosyalardan okuma yapılması istenmektedir. Üç boyutlu nokta bilgilerini içeren bu dosyalar, binary ve ascii şeklinde olduğu gibi, noktaların renk bilgisi (RGB) taşıyıp taşımadığına göre de farklı formatta olabilirler. Format bilgilerinin doğruluğunun, ilk satırlarda deklare edilen bilgiler ile nokta bilgileri arasında bir çakışma olup olmadığının, üç boyutlu noktaların eksik veya yanlış formatta olup olmadığının kontrolü beklenmektedir. Kontrol neticesinde hatalı bulunan dosyalar için ileriki adımlardaki işlemlerin yapılmaması, uyumlu dosyalar için bu işlemlerin yapılması ve aynı zamanda da ekrana yazdırılan çıktı bilgilerinin, proje

dizininde bir klasörde (output), gene “.nkt” uzantılı bir çıktı dosyasına yazdırılması istenmiştir.

Kontrol safhasından sonra ise şu isterlerin hesaplanması gerektiği belirtilmiştir:

- Birbirine en yakın nokta ve en uzak noktaların bilgileri ve nokta numaraları
- Tüm noktaları içine alacak en küçük küpün köşe noktalarının bilgileri.
- Merkezinin üç boyutlu koordinatı ve yarıçapı bilgileri kullanıcıdan istenilen bir kürenin içinde kalan noktaların bilgileri ve nokta numaraları
- Her bir noktanın birbirlerine olan uzaklıklarının ortalaması

Ayrıca; tüm bu kontrol ve hesaplama işlemlerinin her input dosyası için aynı anda yapılması, kullanıcıya bir menü vasıtasıyla sunulması, dosya okuması ve kontrolü yapmadan diğer işlemlere geçilmemesi beklenmiştir.

### 2) Planlama

Proje üzerinde çalışmaya başlamadan önce, ana hatlarıyla kurguladığım plana büyük ölçüde bağlı kaldım. Bu planlama şu şekildeydi:

- Switch/case yapısı kullanılarak bir menü hazırlanmalı.
- Mümkün olduğunca çok fonksiyon tanımlanmalı, main() fonksiyonun içi olabildiğince sade kalmalı.
- Ana gövde struct yapısı kullanılarak hazırlanmalı.
- Okuma/yazma ve kontrol işlemleri için file ve string işlemleri kullanılmalı.
- Projede belirtilen isterler için haricen algoritmalar kurulmalı.

## II. PROBLEM ÇÖZÜM VE UYGULAMA

### 1) Yararlanılan Teknolojiler

Uygulama, C programlama dili kullanılarak, Code:Blocks geliştirme ortamında oluşturulmuştur.

### 2) Akış Şeması

Ektedir.

### 3) Probleme Çözüm Yaklaşımı

“.nkt” dosyaları ile daha rahat bağlantı kurabilmek içinde *Dosya* struct’ı oluşturuldu, her satır string olarak değerlendirildi. RGB özelliğine sahip olan ve olmayan üzere iki ayrı nokta struct’ı oluşturuldu. *Dosya* struct’ının içinde dosya ismi, çıkış yolu, satır sayısı, nokta sayısı gibi özelliklerin yanında, *Nokta* tipinden struct’a da yer verildi. 0/1 enum *boolean* tanımlandı ve gene *Dosya* struct’ı içerisinde binary, RGB, geçerli gibi *boolean* özelliklere yer verildi.

Struct ifadeleri tanımlandıktan sonra

main() fonksiyonunda switch/case ve do/while yapıları ile menü yapıldı. Kullanıcı “1” no’lu seçeneği seçmeden, yani dosya okuması yapmadan diğer seçenekler ile işlem yapılamaması sağlandı, ancak bir uyarı mesajına gerek görülmedi. “5” no’lu seçenek olan son işlemten sonra program kendini kapatmaktadır.

Dosya işlemleri ile sadece “.nkt” uzantılı dosyaların okuması sağlandı. String işlemleri, struct yapısı ve fonksiyonlar sayesinde; 3 boyutlu noktaların verildiği dosyaların ascii/binary ve RGB’ye sahip olma/olmama özelliklerinin kontrol edilebileceği şekilde program geliştirildi.

Gelişmiş string işlemleri sayesinde dosya format bilgisi ve noktaların formatlarının kontrolü sonrası, uyumsuz olan dosyalara ileriki işlemleri yapılamaması sağlandı. Ayrıca fonksiyonlar yoluyla, program açıldığında *output* klasörünün temizlenmesi ve “.nkt” uzantılı dosyaların okunmaya başlanmasından sonra, output dosyasının oluşturulması sağlandı.

İsterler için geliştirilen algoritmalara ilişkin bilgiler akış diyagramında verilmiştir. İsterlere yönelik tanımlanan fonksiyonlarda, math.h kütüphanesinden yararlanılmıştır.

### 4) Kod

6 kütüphaneden yararlanılan bu uygulamada, 18 adet fonksiyon

tanımlanmıştır ve program yaklaşık 1020 satır kod içermektedir.

### 5) Uygulamanın Çalıştırılması

Uygulama oldukça basit bir menü ve görünüme sahiptir. Uygulama çalıştırıldığında, kullanıcıyı aşağıdaki menü karşılar:

- “1- Dosya Kontrolü
- 2-En Yakın/Uzak Noktalar
- 3-Kup
- 4-Kure
- 5-Nokta Uzaklıkları”

Kullanıcı klavyeden “1” girişini yapmadığı sürece herhangi bir dosyadan veri okunmaz, dolayısıyla diğer işlemler de yapılamaz. Ancak bu durumda bir uyarı ekranı gözükmez, menü tekrar kullanıcıya sunulur. Sırasıyla tüm işlemler yapılır. Son işlemten sonra program kapanır.

Diğer seçenekler girildiğinde, uygulama istelere uygun işlemler yapmakta, ekrana ve output dosyalarına veriyi yazdırmaktadır.

### 6) Kullanılan Kütüphaneler

- stdio.h
- stdlib.h
- string.h
- ctype.h
- dirent.h
- math.h

## III. DEĞERLENDİRME VE

## YORUM

Proje öncesi süreç düşünüldüğünde; ciddi (1000+ satır kod) sayılabilecek bir uygulama geliştirmek çeşitli güçlükler yaşattığı kadar, sevindirici tecrübeler de yaşattı.

Uygulamanın nispeten büyük olması, *debug etmenin* önemini ortaya koydu. Debug sistemini çok iyi bilmiyor oluşum, genellikle vakit kaybettirdi. Uygulama geliştirmenin çeşitli safhalarında, istediğim değerlerin ne olduğunu görebilmek için *printf* fonksiyonu kullanmak zorunda kaldım.

“.nkt” dosyalarını okuduktan sonra, kontrol işleminin oldukça kompleks olması beni oldukça zorladı. Artık daha derin düşünmememe katlı sağladı.

Uygulamaya gelmek gerekirse; format bilgisi kontrol ve nokta formatı kontrol mekanizmalarının sıkıntısız; istelerle ilgili kurulan algoritmaların doğru ve eksiksiz çalıştığını düşünmekteyim. Sonradan ortaya çıkabilecek bir hata, yanlış algoritma kurmaktan veya herhangi birşeyi gözden kaçırmaktan kaynaklanabilir; fakat ben istelerle ilgili doğru algoritmalar kurduğumu düşünmekteyim.

## IV. KAYNAKÇA

- 1)  
<https://stackoverflow.com/questions/30877110/how-to-determine-directory-or-file-in-linux-in-c>
- 2)  
<https://stackoverflow.com/questions/38976582/reading-text-file-until-eof-using-fgets-in-c>

