PROGRAMLAMA LABORATUVARI 1, PROJE 1

Semanur Bağcı, Ece Peker

### [semanurbagci0@gmail.com](mailto:semanurbagci0@gmail.com) , [eceepeker@gmail.com](mailto:eceepeker@gmail.com)

KOCAELİ ÜNİVERSİTESİ/MÜHENDİSLİK FAKÜLTESİ/BİLGİSAYAR

MÜHENDİSLİĞİ BÖLÜMÜ

**1.Projede bizden istenen:** Farklı türdeki 3 boyutlu görüntü dosyalarından okuma yapılarak verilecek problemlerin çözümü istenmektedir.

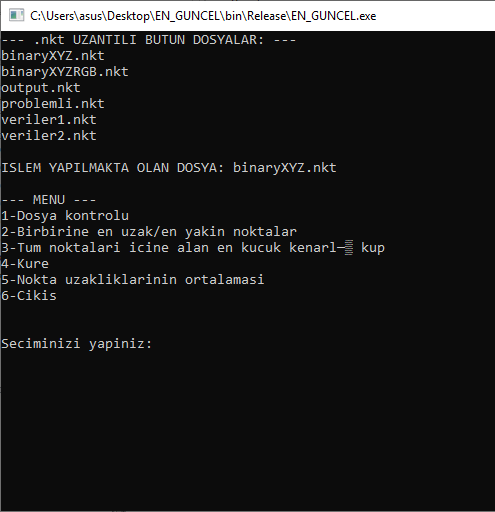
**2.Projenin amacı:** Proje gerçekleştirimi ile beraber öğrencilerin dosyalama ve lineer cebir konularında program yazabilmesi amaçlamaktadır.

**3.Özet:**

Programımız; kodun içinde bulunduğu klasördeki dosyaların okunmasıyla başlar. “.nkt” uzantılı dosyaları bulup bu dosyalar üzerinde aşağıda yer vereceğimiz kontrol ve işlemleri yapar:

Dosyanın data bilgisini, ardından r g b bilgisini kontrol eder ve elde edilen kontrol sonuçlarınca dosyaları kendi format ve bilgilerine göre okur, içlerindeki bilgileri tanımladığımız değişkenler yardımıyla tutar. Tüm bu işlemler sonucunda kullanıcı şöyle bir menüyle karşılaşır ve ilgili dosya üzerinde (ISLEM YAPILMAKTA OLAN DOSYA: … şeklinde çıktısı verilmiştir.)

yapılmasını istediği işlem doğrultusunda seçimini yapar.



Yapılan seçim üzerine de ilgili secimin işlemi gerçekleştirilir, sonuçlar ekrana yazdırılır. Program sonlanır.

Programımız; versiyon bilgisi, “ALANLAR x y z” formatı, nokta sayısı konularında hatalı ya da eksik dosyalar için hata mesajlarını verir, diğer işlemleri yapmaz.

**4.Proje Yalancı Kodu:**

ADIM 1- BAŞLA.

ADIM 2- Klasör içindeki dosyaları oku.

ADIM 3- “.nkt” uzantılı olan dosyaları bir dizide tut ve ekrana yazdır.

ADIM 4- Dosyaların içinde teker teker dolaş.

ADIM 5- if (dosyanın data bilgisi == binary) then dosyanın r g b bilgisini kontrol et.

ADIM 6- if (r g b bilgisi yoksa) then dosyanın içini oku.

ADIM 7- Kullanıcı için menü çıktısını ver. Ve kullanıcıdan seçim bilgisi al.

ADIM 8- switch(secim) yapısıyla; case 1: dosya için versiyon bilgisi, ALANLAR x y z formatı uygunluğu nokta sayisi kontrollerini yap. Kontrol sonuçlarını ekrana yazdır.

ADIM 9- case 2: dosya içindeki noktalardan birbirlerine en yakın olan noktaları ve birbirlerine en uzak olan noktaları bul. Nokta bilgilerini ekrana yazdır.

10-case 3: dosya içindeki tüm noktaları içine alacak küpün köşe noktalarını bul, sonuçları ekrana yazdır.

ADIM 11- case 4: Kullanıcıdan daha önce yarıçap ve merkez nokta koordinatları alınmamış ise bu bilgileri al. Alınmış olan ya da alınan bilgilerle oluşması beklenen kürenin içinde olacak noktaları ve kürenin içinde olacak olan toplam nokta sayısını bul ve ekrana yazdır.

ADIM 12- case 5: Tüm noktaların birbirlerine olan uzaklığını bul.

toplam\_uzaklık+=uzaklık

ort\_uzaklık=toplam uzaklık/uzaklık\_sayisi

Ortalama uzaklığı ekrana yazdır.

ADIM 13- case 6: (varsa) bir sonraki dosyaya geç/programı sonlandır.

ADIM 14- if(r g b bilgisi varsa) then dosyanın içini oku.

ADIM 15- ADIM 7’ye dön.

ADIM 16- if(dosya data bilgisi == “ascii”) ADIM 7’ye dön.

ADIM 17- BİTİR.

**5.Giriş:**

Program C dilinde yazılmıştır. Proje yapılırken derleyici olarak CodeBloks kullanıldı.

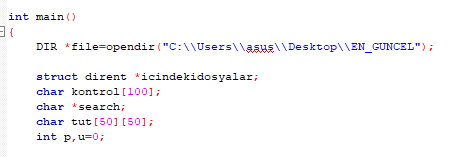
Program dosya okuma, dosyaya yazdırma algoritmalarını, birtakım lineer işlemleri koda dökebilme, algoritma yazabilme becerilerini sınamakta. Program sonuçlarına programın da içinde bulunduğu klasördeki output.nkt dosyasından da ulaşılabilir.

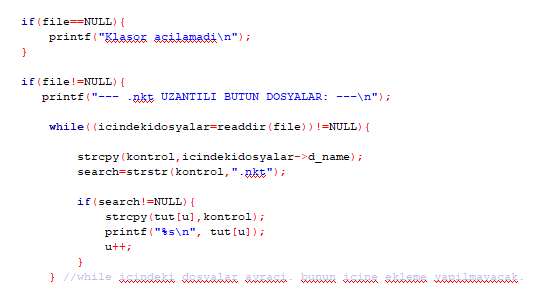
**6.Yöntem:**

Projeye başlamadan önce ve proje süresince dosya işlemleri konusunda çalışma ve araştırmalar yaptık.

Önce dosya okuma işlemlerini, sonrasında ise lineer işlem algoritmasını oluşturmaya çalıştık.

Proje süresince karşımıza çıkan eksik ve yanlışlar doğrultusunda da yeni araştırmalar yaparak yeni kütüphane, fonksiyon ve dosya okuma metotları da öğrendik.





**6.1.Proje Süresince Karşılaşılan Sıkıntılar:**

Proje yapım aşamasında küçük eksik ya da hatalar sonucunda kodumuzun yanlış çalıştığı ya da hiç çalışmadığı durumlar oldu.

Örneğin; dosya okuması yaparken, nokta sayılarını tuttuğumuz sayacı sayma işlemi bittikten sonra sıfırlamadığımız için program ilk dosyayı okuyor ama sonrasında gelen dosyaları okumada sorun çıkarıyordu. Küçük bir “sayac=0;” işlemiyle bu sorunu çözdük.

Yazdığımız kod iç içe geçmiş birçok kontrol ve döngü işlemlerini içerdiğinden parantezler konusunda bazı sorunlar yaşadık. Bu sorunu da parantezlerin yanına yorum satırlarıyla hangi işleme ait olduklarını yazarak çözmeye çalıştık.

r g b bilgisi bulundurmayan binary dosyada okuma yapabiliyor ama okunan verileri struct’ta tutamıyorduk. program olmayan r g b bilgileri için random değer ataması yapıyordu. r g b’siz binary için ayrı bir struct tanımlayarak programın doğru atama yapmasını sağladık.

Case 5’te bulunan ortalama uzaklık hesabı

işlemi, 10’ar tane nokta içeren örnek binary dosyalarında ve bizim programın doğruluğunu kontrol etmek için oluşturduğumuz ve “deneme.nkt” ismini verdiğimiz, içinde binary dosyalarda olduğu gibi az sayıda nokta bildisinin bulunduğu dosyada çalışmasına rağmen, 10000 ve yaklaşık 85000 noktanın bulunduğu ascii dosyalarda çalışmasına rağmen hatalı sonuç veriyor. Biz nokta bir şekilde bellekte bir taşma sonucu olabileceğini düşündük ama denesek de herhangi bir çözüm bulamadık.

**6.2. Kullanılan Kütüphaneler:**

#include <stdio.h>

#include <stdlib.h>

#include <dirent.h>

#include <string.h>

#include <math.h>

**6.3. Kullanılan Bazı Fonksiyonlar:**

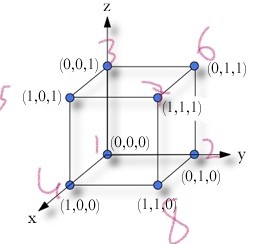
* opendir()
* readdir()
* fopen()
* fclose()
* strcpy()
* strstr()
* fgets()
* sscanf()
* fread()
* printf()
* scanf()
* fprintf()

**6.4. Bazı Denklemler:**

* uzaklik=(sqrt(((rsiz\_bin[i].x)-(rsiz\_bin[j].x))\*((rsiz\_bin[i].x)-(rsiz\_bin[j].x)))+(((rsiz\_bin[i].y)-(rsiz\_bin[j].y))\*((rsiz\_bin[i].y)-(rsiz\_bin[j].y)))+(((rsiz\_bin[i].z)-(rsiz\_bin[j].z))\*((rsiz\_bin[i].z)-(rsiz\_bin[j].z))))
* ort\_uzaklik=toplam\_uzaklik/uzaklik\_sayisi

**7.Kaynakça:**

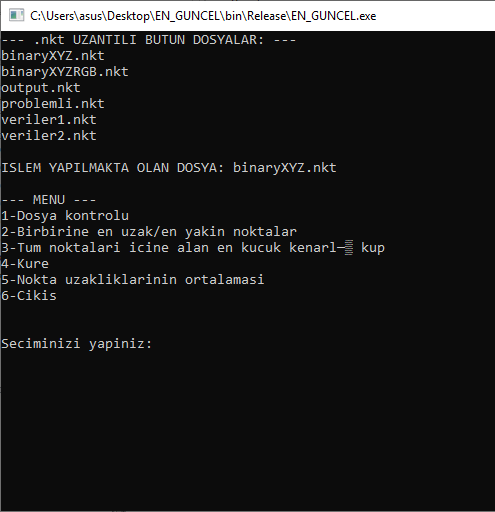
* <https://www.bilaloz.net/tag/sozde-kod-ornek/>
* <https://stackoverflow.com/questions/17598572/read-write-to-binary-files-in-c>
* <https://drive.google.com/drive/folders/1RWlSaFEaXei4_7trdfUlwBnbv18S2zEs> (2018-2019 Programlama dersi slaytları)
* <http://web.firat.edu.tr/iserhatlioglu/BOLUM8.htm>

****

* case 4’te oluşacak küpte köşeleri belirlerken yukarıdaki görselden yararlandık.
* <https://pubs.opengroup.org/onlinepubs/7908799/xsh/dirent.h.html>
* <https://www.youtube.com/channel/UCZNZj3mkdCGJfCoKyl4bSYQ>

**8.Programımızla ilgili bazı görseller:**

* Kullanıcının karşısına çıkacak menü:

****

* Proje hazırlama sırasında bizlere verilen test dosyaları üzerinde gerçekleştirilen bazı işlemler:

