**3 BOYUTLU GÖRÜNTÜ DOSYALARI PROJESİ**

Taha Batuhan TÜRK - İbrahim PAMUK

180202007 – 180202023

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

[tbturkk@gmail.com](mailto:tbturkk@gmail.com) – [ibrahimpamuk@outlook.com](mailto:ibrahimpamuk@outlook.com)

**1-) Projenin Tanımı:**

Bu kısım sadece projenin açıklamasını okuyup edindiğimiz ön bilgiye göre yazılmıştır. Projenin açıklamasını dikkatli bir şekilde okuduğumuzda beş adımlı bir probleme çözüm bulmamızı istemekte olduğunu gördük. Bu adımları izlerken, bize daha önceden verilip, içerisinde başlık bilgileri ve nokta bilgilerini tutan “.nkt” uzantılı dosyaların kullanılacağını anladık. Ve yapılan işlemlerin “output.nkt” ye yazılmasını sağladık.

Birinci adımda dosyaların başlık bilgilerinin doğruluğunun kontrol edileceğini anladık. Eğer başlık bilgilerinde hata yoksa diğer işlemlere geçilir aksi takdirde hata olan dosyalarda işlem gerçekleştirilmez. Ve yapılan işlemler bütünün “output.nkt” ye yazılmasını sağladık.

İkinci adımda hatasız dosyaların içerdiği nokta verilerinde en yakın ve en uzak noktaları bulmamız ve bu noktaların “output.nkt” ye yazılmasını sağladık.

Üçüncü adımda hatasız dosyaların içermiş oldukları tüm nokta verilerini içeren en küçük kübün köşegenlerinin nokta bilgilerini “output.nkt” ye yazılmasını sağladık.

Dördüncü adımda kullanıcıdan alınan küre bilgileri doğrultusunda bu kürenin içinde

kalan noktaların bilgilerini “output.nkt” ye yazılmasını sağladık.

Beşinci adımda hatasız dosyalardan çekilen her bir noktanın birbirlerine olan uzaklıklarının ortalamasının alınması ve bunun “output.nkt” ye yazılmasını sağladık.

Başlık bilgi kontrolü: Dosyadan okuma işlemi sırasında veya okuma işleminden sonra kontrolünü gerçekleştirmemiz gerekmektedir. Dosyanın başlık bilgileri aşağıda verilen formata aykırıysa hata mesajı döndürülür ve sonraki dosyanın kontrolüne geçilir.

1. #Noktalar dosya formatı
2. VERSION 1
3. A) ALANLAR x y z r g b
4. ALANLAR x y z
5. NOKTALAR X (X: nokta sayısı)
6. DATA ascii – DATA binary

**2-) Yapılan Araştırmalar ve Karşılaşılan Sıkıntılar**

Bu kısım proje öncesi ve sonrası araştırmaları ve de projenin yapım aşamasındaki sıkıntıları ve çözümlerini içermektedir.

Klasör listelemeyi klasik dosya işlemleri ile yapamadığımız için araştırıp “dirent.h” kütüphanesini kullanabileceğimizi öğrendik.

Binary dosya ile işlemler yaparken nasıl formatlı bilgi çekeceğimizi ve bilgi yazabileceğimizi bilmiyorduk, kısa bir araştırmayla “fread” ve “fwrite” fonksiyonlarını kullanabileceğimizi öğrendik.

Komut paneli ekranında fazla girdi olunca görsel karışıklık olduğunu gördük ve bunun önüne geçmek için ekran temizleyici fonksiyon olan “system(CLS)” komutunu kullanabileceğimizi öğrendik.

Arayüzü ilk başta tasarlayıp ona göre fonksiyonları yazdığımız için, sonradan istenen arayüz değişikliğine kısa süre içinde birebir adapte edemedik ve çok benzer bira arayüze döndürdük.

Binary rgb dosyadan nokta çekerken rgb verileri 32bit(4byte int) alındığı ve bizden 4bitlik(1byte int) değerleri istendiği için ilk başta bir çözüm bulamadık fakat elde ettiğimiz rgb verilerini 256ya mod aldığımızda bizden istenen 4bitlik değerlere eşit olduğunu fak ettik. Ve çektiğimiz rgb verileri 256ya mod alıp nokta dizisine atadık.

**2.1-) Proje Sırasında Yararlanılan Teknolojiler**

Projeyi C programlama dili kullanarak, CodeBlocks geliştirme ortamında kodladık. Programı kodlarken C programlama dilinin bize sunduğu pointer olanaklarını, dosyalama işlemi fonksiyonlarını kullandık.

Binary dosyaların içeriğini karşılaştırıp test etmek için hexed.it sitesinin altyapısından yararlandık.

**3-) Tasarım**

**3.1) Akış Şeması**

Kısım ektedir. (EK-1)

**4-) Genel Yapı**

**4.1) Kullanıcı Kısmı**

Program çalıştığında ilk olarak projenin bulunduğu klasördeki “.nkt” uzantılı dosyaları listelemektedir. Ardından kullanıcıdan alınan satırdaki “.nkt” uzantılı dosya açılır. Ve işlem listesinin bulunduğu karşılama ekranı gelir. Kullanıcı ilk önce 1. işlemi girmek koşuluyla sırayla istediği işlemi yapabilir. Daha sonra kullanıcı sıradaki nkt dosyasına geçiş yapmak istiyorsa girişiyle belirtir ve yeni “.nkt” uzantılı dosya seçimi yapması istenir. Bu işlemler tüm “.nkt” uzantılı dosyalar üzerinde gerçekleştirilene kadar devam eder. Kullanıcı tüm nkt dosyalarıyla işlemleri yaptıktan sonra programı sıfır tuşlayarak programı sonlandırır.

AKIŞ ŞEMASI

Kullanıcıdan programın bitirilmesi için 0 girişi istendi

Başlık veya nokta bilgisinde hata varsa

Hata mesajı ekrana ve dosyaya yazıldı, dosya kapatıldı.

Başlık veya nokta bilgilerinde hata yoksa

Kullanıcıdan işlem seçimi istendi

Seçilen tüm işlemler sırasıyla yapıldı ve elde edilen sonuçlar ekrana ve dosyaya yazıldı.

Seçilen dosyayla ilgili yapılabilecek işlemler ekrana bastırıldı

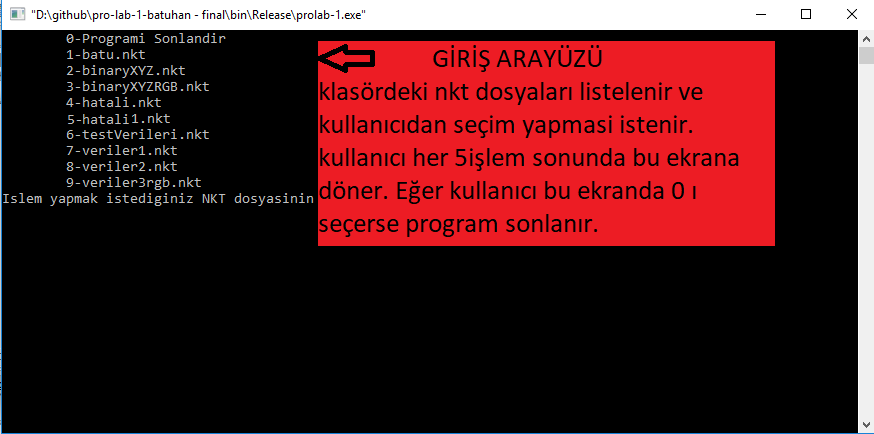
Kullanıcıdan 0-9 arasında bir giriş yapılması istendi

Dosya seçim arayüzü ekrana bastırıldı

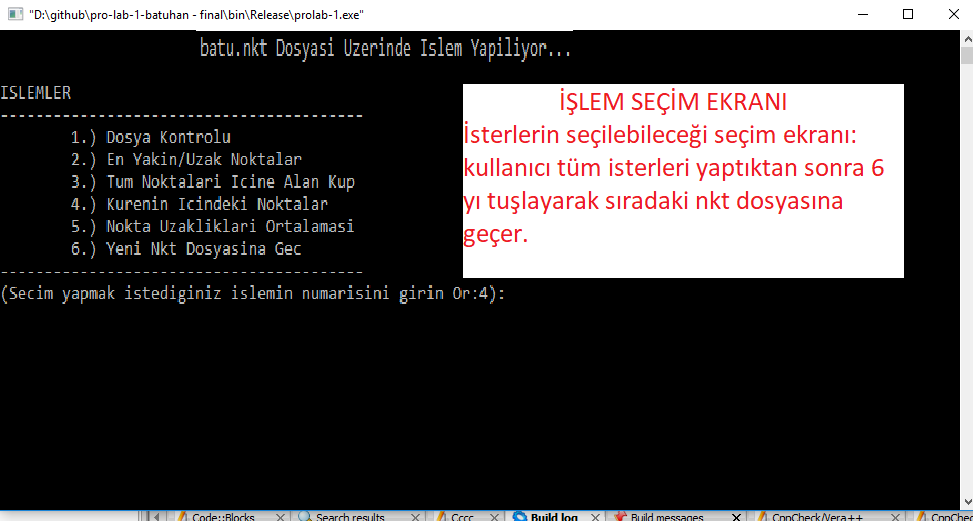
Kullanıcıdan -önce dosya kontrolü olacak şekilde- dosya işlemi alındı

Başlık ve nokta bilgileri dosyadan okundu

EK-1



ŞEKİL1- GİRİŞ ARAYÜZÜ



ŞEKİL2- SEÇİM EKRANI

**4.2) Kod Kısmı**

Kod kısmına baktığımızda ise; önce dosya içindeki nokta bilgilerinin, dosya içerisinde nokta sayılarının, toplam “.nkt” uzantılı dosya sayısının, noktaların data kontrolü, alan kontrolünü ve kürenin sadece bir kere girilmesini sağlayan kureSecim i“global” değişken olarak tanımladık. Daha sonra main fonksiyonumuzda global olarak tanımladığımız output FILE’ını “a”(append) modda açtık. Ardından tüm nkt dosyalarında işlem yapabilmek için üst while döngüsünü başlattık. Üst while’ın içinde bulunan klasörü listele fonksiyonu yeni nkt dosyasında işlem yapabilmek için, klasörde bulunan nkt uzantılı dosyaları satır numaraları ile birlikte ekranda listeler.(bkz. ŞEKİL1) dosyaSecim fonksiyonu kullanıcının seçtiği satırdaki dosyanın ismini alır ve sonra dosyaAc fonksiyonu bu isimdeki dosyayı açar.(global olan FILE pointer’a atar.) Eğer kullanıcı 0.satırı seçerse bu kullanıcının programı sonlandırmak istediğini belirtir ve program sonlandırılır. Eğer dosya açılırsa Şekil2 de gösterilen “Seçim Ekranı” gelir. Daha sonra Alt while döngüsüne girilir ve kullanıcı sıradaki nkt dosyasına geçmek için 6. İşlemi seçene kadar, 1 den başlamak koşuluyla tüm işlemleri yapabilir. Eğer kullanıcı 6 yı seçerse geçerli nkt dosyası kapatılır, bellekten noktalar için ayrılan alan serbest bırakılır.

İŞLEMLER MENÜSÜ

1) Dosya Kontrolü: Açılan dosyanın başlık bilgilerini ve noktaların doğruluğunu kontrol eder. Eğer yanlışsa dosya kapatılır ve yeni nkt dosyasına geçilir. Eğer doğruysa Nokta sayısı kadar globalNoktalar struct dizisi için bellekten yer ayırılır ve dosyadan çekilen noktalar bu struct dizisine atanır.

2) En Yakın/Uzak Noktalar: 2.isteri gerçekleştirir

3) Tüm Noktaları İçine Alan Küp: 3. İsteri gerçekleştirir.

4) Kürenin İçindeki Noktalar: 4. İsteri gerçekleştirir.

5) Nokta Uzaklıkları Ortalaması: 5. İsteri gerçekleştirir

**4.2) Kod Bilgisi**

Bu kısımda kodun fiziksel özellikleri yazmaktadır. Programın tamamı, kendi yazdığımız fonksiyonlar ve main fonksiyonun içindeki kod kısımları da dahil olmak üzere toplam 928 satırdan oluşmaktadır. Programda; nokta bilgilerini tutmak üzere toplam 4 adet struct yapısı, 16 fonksiyon(main fonksiyonu dahil 17) ve 10 tane global değişken kullandık. Programa dahil edilmiş 5 kütüphane bulunmaktadır. (Bkz. 5)

**5-)Kütüphaneler**

Bu kısımda projemize “include” ettiğimiz kütüphanelerimiz bulunmakta:

1-) <stdio.h> : C standart kütüphanesinde bulunan, input, output dosyalarını programımıza eklememizi sağlayan kütüphanedir.

2-)<stdlib.h> : C standart kütüphanesinde bulunan bellek yönetimi, ortamla etkileşim, tamsayı aritmetiği, arama, sıralma, dönüştürme gibi işlevlerin bildirimlerini içinde bulunduran bir başlık dosyasıdır.

3-)<dirent.h> : Bir dizin içindeki dosyaların listelenmesinde kullanmış olduğumuz kütüphanedir.

4-)<string.h> : String’lerde yapacağımız işlemleri daha kısa sürede yapabilmemizi sağlayan, bir çok yararlı fonksiyonu içinde bulunduran String’lere özgü bir kütüphanedir.

5-)<math.h> : C programlama dilinde temel matematiksel işlemleri gerçekleştirmek için tasarlanmış bir standart kütüphane öncü dosyasıdır.

6-)<conio.h> : Ekran üzerinde işlem yapılmasını sağlayan fonksiyonları bünyesinde bulunduran kütüphanedir.

**6-)Referanslar**

<http://pubs.opengroup.org/onlinepubs/7990989775/xsh/readdir.html>

<https://hexed.it/>

<https://www.quora.com/What-is-include-conio-h>

<https://www.tutorialspoint.com/c_standard_library/>

<https://github.com/tronkko/dirent>

<http://tayfunseyran.blogcu.com/ekran-fonksiyonlari/9276738>

<https://c4ctuscode.blogspot.com/2017/07/include-nedir-ne-anlama-gelir.html>

http://www.ahmetsayar.com/