

3 BOYUTLU GÖRÜNTÜ DOSYALARI PROJESİ

Erdem Nayın 180202050 – Enes Malik Tok 180202042

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

erdem_nayin@hotmail.com – etok6181@gmail.com

1-)Özet

Program açıldığında, klasörde bulunan dosyalar taranarak daha önce hesaplama yapılmışsa “output.nkt” dosyası harici “.nkt” uzantılı olanlar ayrılıyor ve bilgileri bellekte tutuluyor.

Kullanıcı dosyaları kontrol ettirdiğinde bu “.nkt” uzantılı dosyaların uygunluğu kontrol ediliyor. Programın ilerleyen safhalarında kullanıcının yapacağı işlemler için uygun olan dosyaların isimleri bellekte tutuluyor. Uygun olmayan dosyalar ise bellekte tutulmuyor ve belirtilen hata mesajları ekranda gösteriliyor.

Kullanıcı yaptırmak istediği işlemlerden birini seçtiğinde bellekte tutulan uygun dosyaların isimleriyle, belirtilen işlemler; her dosyanın kendine özgü algoritmalarla, her dosya için ayrı ayrı hesaplanıyor ve sonuçlar ekranda belirtiliyor.

Sonuçlar klasörde bulunan “output.nkt” isimli dosyada saklanıyor. Program bitiminde buradan, istenilen verilere erişilebilir.

2-)Giriş

Projede yapılması istenen bir klasörün içinde bulunan “.nkt” uzantılı tüm dosyalardan ayrı ayrı 3 boyutlu nokta verilerinin okunması ve gerekli hesaplamaların yapılmasıdır.

Veriler “.nkt” uzantılı dosyalarda standartları daha önceden belirlenmiş şablonlar halinde tutulmaktadır.

Dosyaların ilk kısımları başlık bilgileri olarak adlandırılan ve dosyalarda bulunan 3 boyutlu noktaların özelliklerini belirten bilgilerden oluşmaktadır.

Başlık bilgilerinin ilk satırları yorum satırları, ikinci satırları dosyaların sürüm numaralarını, üçüncü satırları dosyada bulunan 3 boyutlu noktaların hangi bilgileri içerdiği, dördüncü satırları dosyalardaki 3 boyutlu nokta sayısını, beşinci satırları dosyada bulunan bilgilerin saklanış biçimlerini ifade etmektedir.

Üçüncü satırda, dosyada bulunan noktaların bilgilerinin koordinat sisteminde ifade edilişi iki şekilde olabilir:

1. ALANLAR x y z: Noktaların 3 boyutlu koordinat sisteminde bulundukları yerler verilmiştir.
2. ALANLAR x y z r g b: Noktaların koordinat sistemindeki yerleri ve renklerini tanımlayan kırmızı-yeşil-mavi kodlarıyla verilmiştir.

Beşinci satırda dosyada bulunan bilgilerin ASCII veya BINARY

formatlarından hangisinde verildiğini ifade etmektedir.

Projede her dosya için kontroller yapılması gerekmektedir. Dosyaların başlık bilgileri yukarıda belirtilen formatlarda değilse dosyadan bilgi alımı yapılmamalı ve bunu belirten hata mesajının verilmesi gerekmektedir.

Dosyaların başlık bilgileri yukarıda belirtilen formatlara uygunsa, dosyalarda bulunan noktaların sayısı, içerdiği koordinat ve renk bilgileri başlık bilgileriyle uyumluluğunun kontrolünün yapılması gerekmektedir.

Uyumlu değilse bunları belirten hata mesajlarının ekrana verilmesi gerekmektedir.

Örneğin: 156.nokta verisi r g b bilgileri olmadan verilmiştir, A.nkt dosyasındaki nokta sayısı başlıkta belirtilen sayıyla uyuşmamaktadır.

Programda her bir dosya için okunan noktalardan aşağıdaki değerler hesaplanmalıdır:

1. Birbirine en yakın ve en uzak noktaların bilgileri ve nokta numaraları
2. Tüm noktaları içine alacak bir küpün kenar noktalarının bilgileri
3. Merkezinin 3 boyutlu koordinat düzlemindeki yeri ve yarıçapı kullanıcı tarafından girilen bir kürenin içinde kalan noktaların bilgileri ve nokta numaraları
4. Her bir noktanın, birbirlerine olan uzaklıklarının ortalaması

3-)Yöntem

Proje C programlama dili ve Code:Blocks geliştirme ortamı kullanılarak yapıldı.

C dilinde bulunan hazır kütüphanelerden ve fonksiyonlardan yararlanıldı. Bulunmayan fonksiyonlar ekibimiz tarafından yapıldı.

Projede; dosyada bulunan bilgilerin kontrollerinin yapılışında, dosyalardan bilgileri alırken, birçok dosyadan veri alırken, binary dosya türündeki dosyalarda veri alırken, programda bulunan menünün tasarımı ve menünün işleyiş biçiminde, istenilen küpün köşe koordinatları hesaplanırken birçok problemle karşılaşıldı. Problemlerin çözümünde ekip halinde çözüm aranıldı, aynı problem veya benzeri bir problemle karşılaşmış arkadaşlarımızla görüş alışverişi yapıldı, internette daha önce bu problemlerle karşılaşan insanların bu problemlerin üstesinden nasıl geldiği araştırıldı ve farklı bakış açıları kazanıldı. Bulunan birçok farklı çözümünden problemlerimizin çözümlerine uygun olanları seçildi.

Program geliştirirken adım adım gidildi. İlk önce tek bir dosyadan nasıl veri okunulacağı yapıldı. Sonra tek dosyadan okunulan verilerin doğruluğunun kontrolünün nasıl olacağı yapıldı. Karşılaşılan problemler halledildi. Sonra menü yapıldı. Menüde belirtilen işlemlerin hesaplanacağı fonksiyonlar, fonksiyonlarda yapılacak işlemler için değişkenler ve sonuçların tutulduğu structlar tanımlandı. Binary

dosyalardan veri alınması ve kontrollerinin yapılması tamamlandı.

Daha sonra klasörde bulunan tüm dosyalara erişim sağlandı. Erişimden sonra bizim işimize yarayacak olan “.nkt” uzantılı dosyaların ayrımları yapıldı ve bir structta tutuldu.

Dosyalar kontrol edildikten sonra içerdikleri bilgilere göre ayrı ayrı gruplandırıldı. Hatalı olan dosyaların ilgili hata mesajlarının yazımı yapıldı. Hatalı dosyaların uygun dosyaların bulunduğu structa atılmaması sağlandı. Dosyalar ancak doğru haldeyse bilgilerinin tutulması sağlandı.

Fonksiyonlarda dosyalar bulundukları grubun veri okuma ve hesaplama işlemlerine tabi tutuldular.

Fonksiyonlar farklı dosyalardan veri okuma işlemine uygun hale getirildi.

Daha sonra menü ilk olarak dosya kontrolü işlemine izin verecek, diğer hesaplamalar dosyalar kontrol edildikten sonra yapılacak şekilde düzenlendi.

Menünün her seçimde her dosya için ekranda gözükmemesi için yani menüdeki her işlem seçimi sonrası sadece sonuçların gözükmesi ve sonuçlar bittikten sonra menünün tekrar gösterilmesi için gereken işlemler yapıldı.

Geliştirme aşamasında tek dosya ile işlemler yapılırken binary dosyadan bilgiler çekildi fakat tüm dosyalar okunurken içeriğin binary olup olmadığının dosyaya girmeden önce tam olarak bilinmemesi sebebiyle binary okuması entegre edilemedi. Çözüm geliştirilemedi.

4-)Sonuç

Sonuçta C dili kullanılarak dosyalama işlemlerinin yapılması, dosyalardan okuma yapılması, struct yapılarının kullanılmasıyla nesneye yönelik programlama düşünce tarzının ve algoritmik düşünme yetisinin gelişmesi sağlandı.

5-)Deneysel Sonuçlar

SECIM 1:
veriler12.nkt DOSYASINDA 1.satir b bilgileri hatali girilmistir.
2.satir g b bilgileri hatali girilmistir.
3.satir z r g b bilgileri hatali girilmistir.
5.satir y z r g b bilgileri hatali girilmistir.
6.satir z r g b bilgileri hatali girilmistir.
veriler2.nkt dosya PROTOTIP HATASI

SECIM 2:

veriler123.nkt DOSYASI

4.981094 0.087787 99.749025 81 178 204
94.957733 98.022101 0.558709 10 192 126
97.079500 33.694392 22.512338 76 44 81
97.035800 33.674645 22.449870 182 107 105

veriler1231223.nkt DOSYASI

79.911334 51.105907 12.947192
11.946037 40.354486 49.804250
78.604502 47.935201 15.789463
79.965671 49.165689 14.851328

SECIM 3

veriler123.nkt DOSYASI
0.015070 0.003270 0.012146
0.015070 0.003270 99.996373
0.015070 99.987497 0.012146
99.999297 0.003270 0.012146
99.999297 0.003270 99.996373
99.999297 99.987497 0.012146
99.999297 99.987497 99.996373

veriler1231223.nkt DOSYASI
11.946037 21.804353 9.585374
11.946037 21.804353 82.178798
11.946037 94.397777 9.585374
84.539461 21.804353 9.585374
84.539461 21.804353 82.178798
84.539461 94.397777 9.585374
84.539461 94.397777 82.178798

SECIM 4:

cx=79.000000-cy=49.000000-cz=14.000000-cr=5.000000

veriler123.nkt DOSYASI
79.911334 51.935201 12.947192 95 227 28
82.905980 50.862317 16.200685 77 228 110
77.461915 51.358116 13.258157 196 34 43
77.611566 45.663866 14.497531 161 25 31

veriler1231223.nkt DOSYASI
79.911334 51.105907 12.947192
78.604502 47.935201 15.789463
79.965671 49.165689 14.851328

SECIM 5:

veriler123.nkt DOSYASI:
65.854827

veriler1231223.nkt DOSYASI:
48.776556

6-)Kaba Kod

- 1-)Başla.
- 2-)Dosya dizin oku ve dosya isimlerini struct dosyaisimleri dizisine kaydet.
- 3-)Dosya nkt uzantılı ise struct nktdosyalar dizisine kaydet.
- 4-)Dosyayı aç ve ilk 5 satırını oku ve bilgileri kontrol edip uyumluluğu kontrol et ve kaydet.
- 5-)Koordinat ve renk bilgilerini kontrol et hatalı noktaları struct hatalinoktalar *hatali_noktalar dizisine ve hatasız noktaları struct nokta *noktalar dizisine bilgileri kaydet.Hatali nokta sayisini(int hn) ve doğru nokta sayisini(int n) say.
- 6-)Menüyü göster.Kullanıcı çıkış için 0'a basmadığı sürece önce 1 basmasını diret. Ardından dosyanın dosya kontrolünü 1'i seçince yap. Doğruluk sonuçlarını output.nkt'ye yaz.
- 7-)Tekrar menüyü göster ve seçtiği işleme gidip doğru olan dosyalar için işlemleri sırasıyla yap.
- 8-)
 - 2.işlem noktalar dizisinin iki tane döngü içinde elemanlarını double uzaklık fonksiyonuna göndererek sonuç üret ve bu sonuçları kendisinden bir önceki sonuçla ile karşılaştırarak en uzak ve en yakın noktaları kaydet. En son çıkan sonucu ekrana ve output.nkt'ye yazdır.

3.işlem dizideki en küçük xyz ve en büyük xyz koordinatlarını bulup x'leri y'leri ve z'lerin kendi arasında farklarını bulup bunları karşılaştırarak

en büyük farkın diğer 2 koordinat arasında da olmasını sağla ve olasılık kullanarak bulunan 6 koordinatla 8 noktayı bul ve sonucu ekrana ve output.nkt'ye yazdır.

4.işlem kullanıcıdan girilmesi istenen kürenin merkez koordinatları ve yarıçapını her dosya için, noktalar dizisindeki xyz koordinatları ile kürenin merkez koordinatlarını uzaklık fonksiyonuna gönderip çıkan sonucun yarıçapdan küçük olanlarını ekrana ve output.nkt'ye yazdır.

5. işlem noktalar dizisinin iki tane döngü içinde elemanlarını double uzaklık fonksiyonuna göndererek sonuç üret ve sonuçları ekleyerek devam et , döngü sayısına böl ve ortalama sonucunu ekrana ve output.nkt'ye yazdır.

7-)Referanslar

1. <https://pubs.opengroup.org/onlinepubs/7990989775/xsh/dirent.h.html>
2. <https://www.geeksforgeeks.org/c-program-list-files-sub-directories-directory/>
3. <https://stackoverflow.com/questions/17598572/read-write-to-binary-files-in-c>
4. <http://bilgisayarkavramlari.sadievrenseker.com/2008/10/22/c-ile-dosya-islemleri/>

8-)Akış Şeması

