Kocaeli Üniversitesi
Bilgisayar Mühendisliği
Yasin Çalışkan Atakan Atalar
180201060
190201038

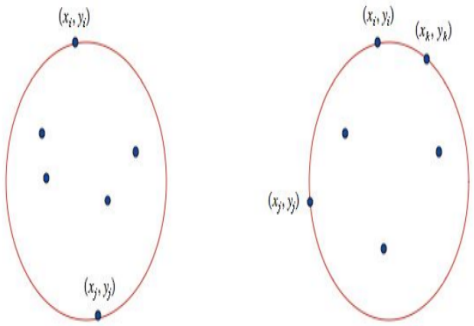
Programlama Laboratuvarı I

PROJE#1 MINIMUM ENCLOSING CIRCLE

Proje Teslim Tarihi : 13.11.2020

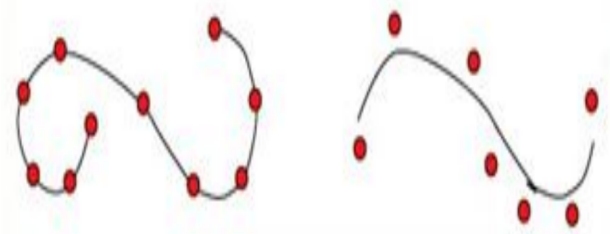
Özet

Bu projede dosyaya kaydedilmiş ve içinde sıralı x y koordinatları bulunan noktaları okuyup bu noktaların bir arayüz yardımı ile ekranda gösterme işleminin yapılması, yine dosyadaki veriler yardımıyla bu noktaları içine alacak en küçük çemberin çizilmesi ve bu noktaların her birinin üzerinden geçen eğrinin ekranda gösterilme işlemleri yapılmıştır. Çizilen çemberin yarıçapı ile merkez noktası tespit edilmiş ve kullandığımız yardımcı kütüphane ile bu veriler ekranda gösterilmiştir. En küçük çemberin ve b-spline eğrilerinin çizimi için sırasıyla Frank-Wolfe ve Catmull-Rom algoritmalarının matematiksel işlemleri analiz edilerek projemize uygun bir hale getirilerek kullanılmıştır.Geliştirme ortamı olarak Visual Studio 2019 kullanılmış olup arayüz tasarımı için allegro kütüphanesi tercih edilmiştir.



Giriş

Spline eğrileri noktalar kümesi çevresinde düzgün bir eğri oluşturmak için kullanılan bir şerittir. Problemin tanımı Minimum Çevreleyen Çember problemi, aynı zamanda En Küçük Çember problemi veya Minimum Kapsayan Daire problemi olarak da bilinmektedir. Projede bizden istenen kullanıcı tarafından tamsayı koordinatlı 2 boyutlu bir düzlemde N nokta verildiğinde, tüm noktaları içeren minimum çevreleyen yarıçaplı daireyi çizdirmemiz istenmektedir
Verilen N noktanın en yakınından geçen eğriyi çizdirmemiz istenmektedir.
Çizdirmiş olduğunuz dairenin ise yarıçapını ve merkezini hesaplamamız istenmektedir.

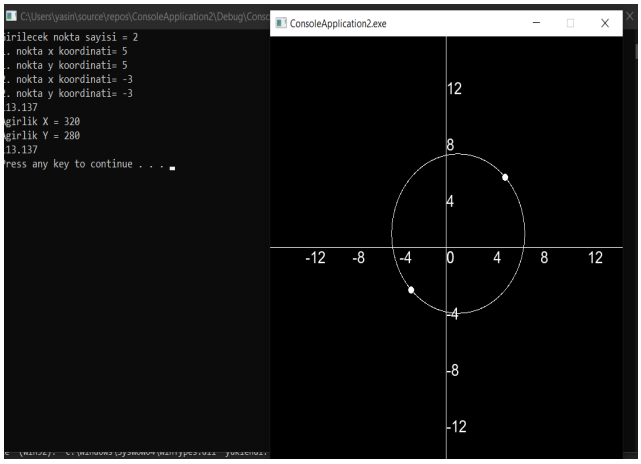


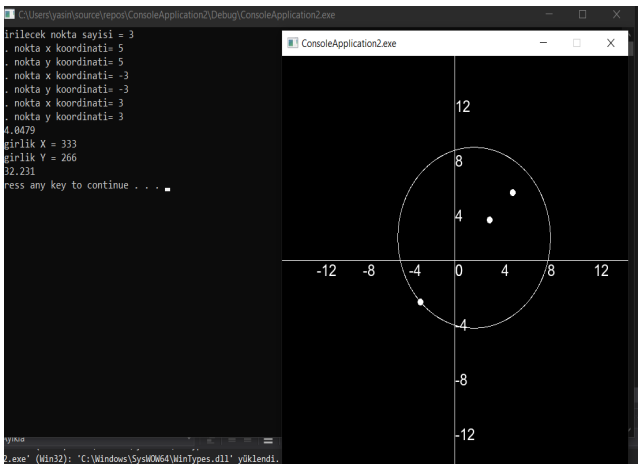
Yöntem

Öncelikle projeye dosyadan veri okuma işlemiyle başlanmıştır. Proje konumundaki sayilar.txt isimli metin dosyasından x,y ikilisi şeklinde yazılmış verileri projede iki ayrı değişkende tutulmuştur. Bu değişkenleri ve ileride çizeceğimiz çemberi arayüz üzerinde gösterebilmek için allegro kütüphanesi aracılığıyla bir ekran oluşturulmuştur. Bu noktaları ekranda gösterebilmek için ekran genişliğini, yüksekliğini ve dosyadan okunan değerleri baz alarak dinamik ölçeklendirme hesabı

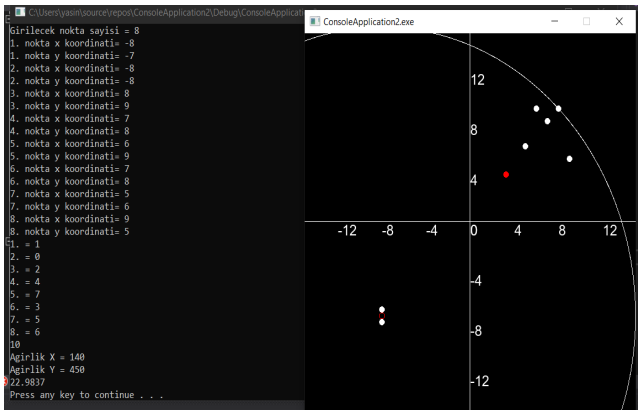
yapılmıştır.Kullanıcı eğer isterse fare tekerleği aracılığı ile ölçeklendirme ayarını yapabilir.

Problemlerin çözümü için birçok algoritma geliştirilmiş fakat birçoğunda başarısız olunmuştur. İlk etapta en küçük çemberi oluştururken dosyadan alınan noktaların x ve y ağırlık merkezleri hesaplanmış, çemberin merkez noktası orası kabul edilmiş ve bu merkez noktasına en uzak nokta ile arasındaki mesafe iki nokta arası mesafe hesaplama fonksiyonu ile hesaplanıp çap kabul edilerek çember çizilmiştir.

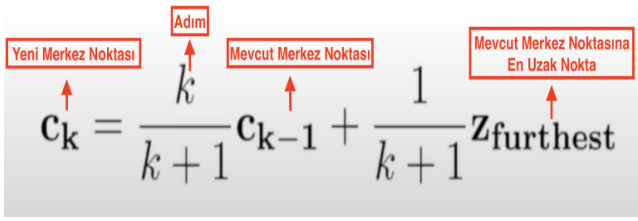




Fakat yukarıda örneğini göreceğiniz şekilde 3 ve 3'den fazla veri girişi yapıldığında merkez noktası yanlış tespit edilmiş ve en küçük çember doğru çizilememiştir. Burada oluşan hataya farklı çözümler getirmiş olsak da girilen her değer için tam anlamıyla doğru sonuca ulaşılamamıştır.



Yukarıdaki örnekte ise kırmızı nokta, girilen tüm noktaların ağırlık merkezi olarak bulunmuş ve kendisine en uzak iki noktanın orta noktasını (içi boş kırmızı nokta) merkez kabul etmiş ve bu merkez noktasına en uzak noktayı çap olarak kabul ederek en küçük çemberi çizdirilmeye çalışılmış fakat yine en küçük çemberi çizdirme işlemi başarısız olmuştur. Bu işlemlerden sonra en küçük çemberi çizdirebilmek için gerekli olan şeyin ideal merkez noktasını bulmak olduğu anlaşılmıştır. Bu noktayı bulabilmek için Frank-Wolfe algoritmasının matematiksel işlemleri projede kullanılmıştır. İdeal merkez noktasının bulunması için projede hesapladığımız ağırlık merkezi bilgisini mevcut merkez noktası kabul ederek yeni merkez noktasını hesaplamak için kullanılmıştır. Aşağıda gösterilen matematiksel işlemlerle beraber merkez noktası her döngüde daha ideal bir konuma yaklaşmıştır.



Programın ideal merkez noktasının tespit edilmesi ve döngüden çıkması için;
1)Çember üzerinde iki nokta varsa ve bu iki nokta çapı oluşturuyor ise
2)Çember üzerinde üç ve üçten fazla nokta var ise
şartları kullanılmıştır.

Hesaplanmış ideal merkez noktası ile kendisine en uzak nokta arası mesafe ise

çap kabul edilmiştir.Elimizde bulunan merkez noktası ve çap bilgisi ile çizilebilecek en küçük çemberin çizim işlemi tamamlanmış oldu.

B-spline için Catmull-Rom algoritmasını kullanılmıştır. Algoritmaya göre dört nokta belirlenir, bu noktalardan iki tanesi eğrinin çizileceği noktalar ve diğer ikisi de kontrol noktası olarak seçilir.

$$q = 0.5\left((2P\cdot 1) + (-P\cdot 0 + P\cdot 2)t + (2P\cdot 0 - 5P\cdot 1 + 4P\cdot 2 - P\cdot 3)t^2 + (-P\cdot 0 + 3P\cdot 1 - 3P\cdot 2 + P\cdot 3)t^3\right)$$

Formülü böyle olmak üzere P0,P3 kontrol noktası P1,P2 eğrinin çizileceği noktalardır. t her döngüde çok küçük miktarda artan bir değişken olmak üzere 0 ile 1 arasında değerler alır.q(t) ise P1 ile P2 arasında t değişkenine bağlı olarak eğrinin geçeceği noktaların koordinat bilgilerini tutar. Değişim t miktarına bağlı olarak arttığı için iki nokta arasında bir çok değer alır ve bu noktaların ekranda gösterimiyle beraber iki nokta arasında eğrinin oluşturulması sağlanır. Eğrinin tüm noktalar üzerinden geçişini sağlamak için P0,P1,P2 ve P3 noktalarının t=1 için yeni değerler alması gereklidir. Örneğin kullanıcı tarafından altı nokta girilmişse;

P0 = i % noktaSayisi => 0 1 2 3 4 5 6

P1 = (i+1) % noktaSayisi => 1 2 3 4 5 6 0

P2 = (i+2) % noktaSayisi => 2 3 4 5 6 0 1

P3 = (i+3) % noktaSayisi => 3 4 5 6 0 1 2

i değişkeni her t=1 için 1 artmakta olup (noktaSayisi-1) olana kadar b-spline eğrilerinin çizdirilmesi sağlanmıştır.

Kodları daha düzenli ve okunabilir hale getirmek amacıyla kodlar metodlara ayrılmış ve bunların sonucunda geriye birden fazla değer return edebilmek için structlardan yararlanılmıştır.

Projenin arayüzünü oluşturduğumuz allegro kütüphanesi ile pencerenin oluşturulması için gerekli genişlik ve yükseklik değerleri global alanda sabit bir değer olarak tanımlanmıştır ve bu değerlere göre pencere oluşturulmuştur. Koordinat düzleminin eksen uçlarına çizilen üçgenlerin çizilme işlemleri yapılmıştır. Renk bilgileri için özel veri türü ayarlanmış ve renk bilgilerinin tekrar tekrar kullanılmaması için değişkenlerde tutulmuştur. Font tipi olarak arial.ttf seçilmiş olup globalde sabit bir değişken olarak font büyüklüğü belirlenmiştir. Koordinat düzlemi üzerindeki sayısal değerler için ve pencerenin sol üst kısmındaki yarıçap ile merkez noktası bilgilerinin yazılması için tanımladığımız font özellikleri kullanılmıştır. Dosyadan alınan noktaların koordinat düzlemindeki yerlerinin daha net ve anlaşılır gözükmesi için ızgaralar oluşturulmuştur. Kullanıcının fare tekerleği aracılığıyla ölçekleme yapması sağlanmıştır.

Kullanıcı noktaları dosyadan girmek yerine koordinat sistemi üzerinden mouse ile yeni noktalar girebilirler. Bunun için allegro kütüphanesinden yararlanıldı ve kullanıcının tıkladığı noktanın x ve y değerleri hesaplanıp koordinat değişkenlerine atıldı.Kullanıcıdan sürekli nokta bilgisi alabilmek için program sonsuz döngüde çalıştırılması sağlandı. Kullanıcının dosyaya girdiği verilere göre ölçeklendirme sabit tutuldu, fare tekerleği ile ölçeklendirme yapabilmesi sağlandı , eski ekran penceresi silindi ve üzerine yeniden çizimler yapıldı. Konsol ekranı temizlendi ve kullanıcının girdiği yeni koordinat bilgisi eklendi, bununla birlikte merkez noktası ve çap bilgileri tekrar hesaplanıp konsol ekranına yazdırıldı.

Zaman Karmaşıklığı

Algoritmanın karmaşıklık analizini yaparken sabit zamanlı veriler göz ardı edilerek sadece oluşturulan iç içe döngülerin hesabı yapılmıştır.Big o analizine göre sadece iç içe döngülerin hesabının yapılması

yeterlidir.

```
for (int i = 0; i < noktaSayisi; i++) // Döngüyü
{
    for (int j = i + 1; j < noktaSayisi; j++)
```

Buradaki verinin karmaşıklığı n*(n-1)/2 şeklindedir. Big o analizine göre karmaşıklık analizi en yüksek dereceli polinomun kat sayısına göre belirlenir. Projenin karmaşıklık analizi $O(n^2)$ ‘ dir.

Sonuçlar

Yapılan tüm işlemlerin sonucunda dosyadan alınan koordinat bilgileri ile hesaplanan yarıçap ve merkez noktası bilgileri konsol ekranında gösterilmiştir. Dosyada ikiden az olacak şekilde x,y koordinat bilgisi girilmesi durumunda kullanıcıya hata mesajı verilmiştir. Merkez noktasının konumu içi dolu kırmızı nokta şeklinde gösterilip kendisine en uzak nokta ile arasında yarıçap çizgisi çizilmiştir ve bu çizginin orta noktasına ‘r’ yazılmıştır. B-spline eğrileri ekranda pembe renginde ve çizilen en küçük çember ise sarı renkte gösterilmiştir. Kullanıcı mouse ile nokta girişi yapabileceği hakkındaki uyarı konsol üzerinden gösterildi.

Kaba Kod

1-Dosyadan N sayıda (x,y) koordinat bilgilerini oku.

2-Okunan bilgileri (x,y) değişkenlerinde tut.

3-Allegro kütüphanesi aracılığıyla arayüzü oluştur.

4-(x,y) değişkenleri ile ölçeklendirme yap.

5-(x,y) noktalarının ağırlık merkezini hesapla ve geçiçi merkez noktasını ata.

6-Merkez noktası olarak atanan veri ve Frank-Wolfe algoritması ile ideal merkez noktasını hesapla.

7-Merkez noktasına en uzak nokta ile arasındaki mesafeyi çap olarak kabul et.

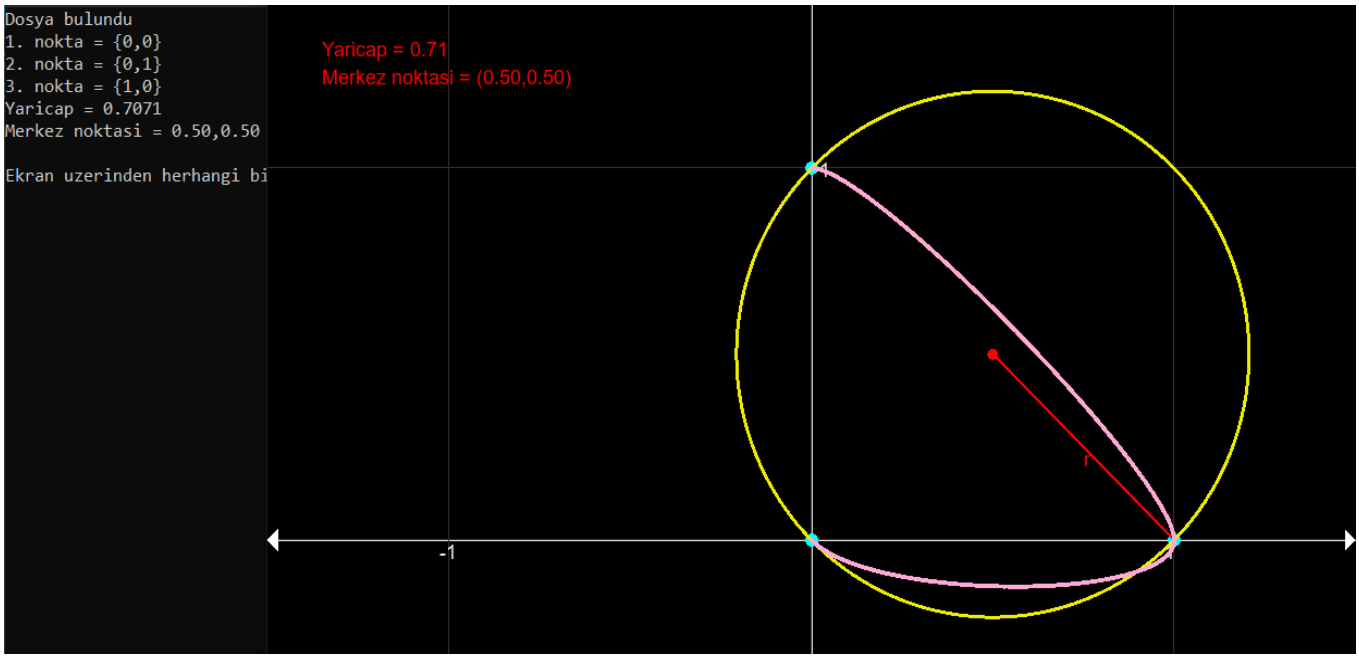
8-Gerekli ölçeklendirmeleri yaparak çemberi ve dosyadan okunan (x,y) noktalarını ekranda göster.

9-B-spline için Catmull-Rom algoritmasını kullan ve eğri noktalarını ekrana bastır.

10-Kullanıcının mouse ile nokta girişi yapabilmesi sağla.

11-Kullanıcının fare tekerleği ile ölçeklendirme yapabilmesini sağla.

Deneysel Sonuçlar



Giriş: {{0, 0}, {0, 1}, {1, 0}}

Çıktı: Merkez = {0,5, 0,5} , Yarıçap = 0,7071

Açıklama: Yukarıdaki daire 0.707 yarıçaplı ve merkez (0.5, 0.5) ile çizildiğinde, belirtilen tüm noktaların dairenin içinde veya üzerinde olduğu görülmektedir.

Kaynakça

http://bilgisayarkavramlari.com/2009/08/10/ splines-seritler/?highlight=b%20spline

https://franknielsen.github.io/blog/kernelCoresetMEB/index.html

https://en.wikipedia.org/wiki/Centripetal_Catmull%E2%80%93Rom_spline

https://www.allegro.cc/manual/5/graphics.html