Kocaeli Üniversitesi

Bilgisayar Mühendisliği Bölümü

Programlama Laboratuvari-1

1.proje

Muhammed Koç

190202040

Proje Özeti:

Projede bizden 2 boyutlu bir düzlemde rastgele verilen tamsayı koordinatlı noktalar etrafında çizilebilecek en küçük daire ve bu noktalara en yakın yerden geçen eğri çizdirilmesi istenmiştir.

Dairenin merkezinin ve yarı çapının hesaplanmalı ve hepsi ekrana bastırılması istenmiştir.

Proje yi C dili üzerinden yaptım ve Graphics.h kütüphanesi kullandım.

Noktalar ise sayi+1boşluk+sayi formatında girilmiş ve kod bu formata uygun yazılmıştır

Arayüzde koordinat sistemi her bir boşluk 1 tamsayı değeri alacak şekilde çizgilerle ölçeklendirilmiştir

Pdf teki verilen iki fotoğraftan herhangi birine uygun yapılabileceği düşünülerek en uygun sonuç alınan hermite spline kullanılmıştır. Ama asıl istenilenin B spline

olabileceği düşünülerek B splinde dahil edilmiştir

Minimum boyuttaki daire oluşturulurken Welzl algoritması baz alınmıştır.

Çıktıda uygun şekilde koordinat sistemi, noktalar ve daire renklendirilerk anlaşılır kılınmıştır.

GİRİŞ

Verilen noktalara göre çizilecek en küçük daireyi çizdirebilmek ve noktalara en yakından geçecek şekilde olan eğriyi çizdirebilme için Graphics.h kullanıldı ve C dilinde yazıldı ama Graphics.h c derleyicisinde hata verdiği için C++ kodu kullanılmadan Graphics.h a uygun C++ derleyicisi kullanıldı.

Codeblocks ide si kullanıldı

Yöntem

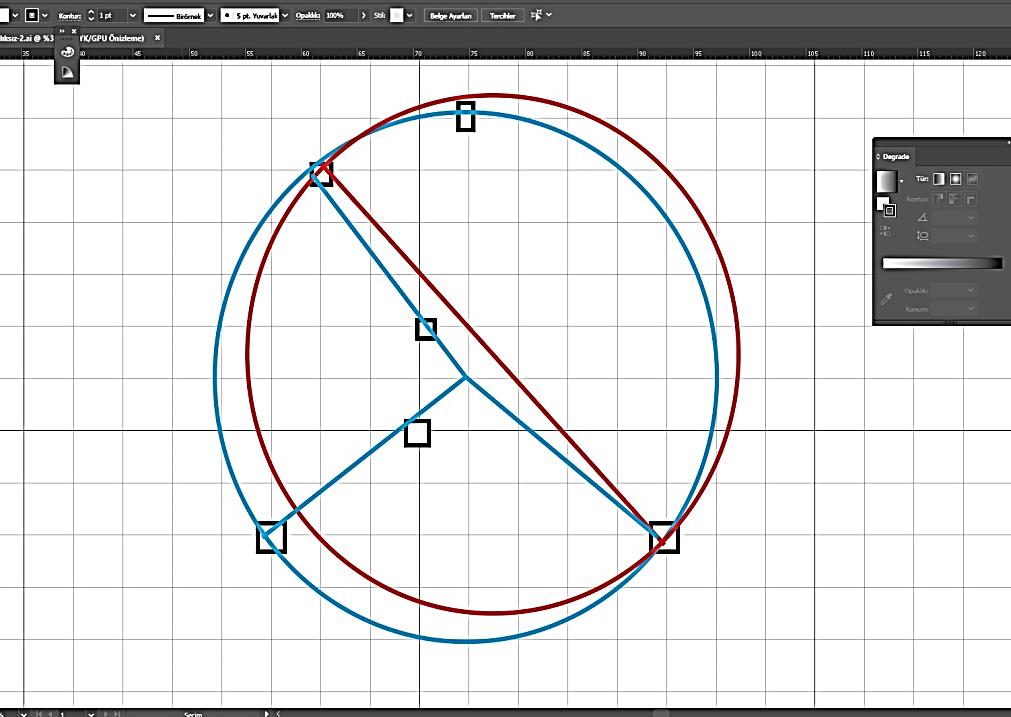
Minumum boyuttaki daire oluşturulurken 2 temel nokta bulunmaktadır:

1. 2 noktanın daireyi gerdirerek maksimum boyuta çıkarması

2. 3 noktanın daireyi gerdirerek maksimum boyuta çıkarması

Diğer durumlarda ise yani 4 veya daha fazla noktanın daireyi gerdirmesi mümkün olmamakta tekrar ilk iki seçeneğe düşerek fazla olan noktaların dairenin çevresinden teması kopacaktır.

Yani ilk iki durum değerlendirilmiş diğer durumlar fiziksel olarak imkansız olduğu için önemsenmemiştir.



(Mavi daire en uzak iki nokta baz alınarak Kırmızı daire ise cevrel çemberin formülü baz alınarak çizilmiştir. )

Şekildeki gibi eğer sadece en uzak 2 nokta değerlendirilirse dairenin dışında hala nokta kalabileceği için hatalı sonuca ulaşılır.

1.seçenği kontrol edebilmek için birbirine en uzak 2 nokta çap kabul edilip bunların orta noktası merkez aralarındaki mesafenin yarısı yarı çap kabul edilmiştir ve tüm noktaların merkeze uzaklığı karşılaştırılarak dairenin içinde kalıp kalmadığı sorgulanmıştır

2. seçeneği kontrol edebilmek için seçilen üçerli noktalar çevrel çember formülü uyarlanarak merkezi hesaplanmış ve yarıçapı bulunmuştur. Noktaların hepsi üçlü döngüye sokularak tüm olasılıklar hesaplanarak tüm noktaları içine alan çember yapılmıştır.

Eğer 3. Seçenekte 2 seçenkten daha küçük bir daire kurallara uyarak oluşuyorsa bu çember seçilmektedir.

Minimum 3 nokta ile bu işlemler uygulanmaktadar.

Daha az nokta için direk atama yapılarak ilk seçenek kullanılmaz.

Hermite spline da

-t\*t\*t + 2\*t\*t - t;

3\*t\*t\*t - 5\*t\*t + 2f;

-3\*t\*t\*t + 4\*t\*t + t;

t\*t\*t – t\*t;

formülleri dörderli olarak noktalar ile çarpıştırıldı.

t noktası ise 0 ile 1 çok yavaş artan bir parametre olarak kullanılarak eğrinin oluşması ve noktalar üzerindeki etkinin artıp azaltılmasında kullanıldı.

Big O analizi ise O^4 bulundu

ana\_fonk() gönderilen n değer girilen 3 lü for döngüsünde

for (int i = 0; i < N; i++)

for (int j = i + 1; j < N; j++)

for (int k = j + 1; k < N; k++)

n^3 olur.

Buradan da gönderilen doğrula() ki for döngüsü ile

for(int i=0 ; i<N ; i++)

O^4 olmuştur

SÖZDE KOD

Uzaklık\_bul{

Girdi: nokta1,nokta2

Çıktı: sonuc

İki nokta arsı uzaklığı bul sonuç a ata }

Cap\_belirle1{

Girdi: nokta1, nokta2

Çıktı: cap

Noktaları x ve y diye topla 2 ye böl

cap.yaricap=uzaklık bul()/2}

cap\_belirle2{

Girdi: n1, n2, n3

Çıktı: cap

Cevrel çemberi formülü ile merkez bul daire.x,daire.y ata

cap.yaricap=uzaklik\_bul()}

doğrulama{

Girdi: daire\_dogrula, nokta1, N

Çıktı:0 veya 1

a=uzaklık bul()

if(a> daire\_dogrula.yaricap) return 1

else return 0}

Ana\_fonk{

Girdi : nokta, N

Çıktı: buyuk

yaricap=2 daire.y=2 daire.x=2

daire.y=nokta[0].y daire.x=nokta[0].x yaricap=0

if(N==0) return daire1

if(N==1) return daire2

buyuk.daire.x=0;

buyuk.daire.y=0;

buyuk.yaricap=INFINITY;

Tüm noktaları cap\_belirle1() e gönder

Daireana1 eşitle

if((daireana1.yaricap<buyuk.yaricap) && dogrulama(daireana1,nokta,N))

{daireana1 i buyuk e at}

Üçlü döngüyle tüm noktaları sırayla capbelirle2() gönder

daireana1 ye eşitle

if((daireana1.yaricap<buyuk.yaricap) && dogrulama(daireana1,nokta,N))

{daireana1 i buyuk e at}}

Cizgiler{

Girdi: a,dairex

dairex.x, dairex.y, dairex.yaricap ı string e çevir

grafik ekranına yazi yazi2 yazi3 bas

koordinat çizgilerini bas

}

yol {

Girdi: nokta,t

Çıktı: tx,ty

spline formülünü uygula

noktalarla t nin yerine yazıldığı formül sonuçlarını çarp

}

N=1 İ=0

DOSYAYI OKU İF (YOKSA) print (BULUNAMADI)

while(Dosyanın sonuna kadar)

İf(satır sonuna gelirse) N 1 artır

Dosya kapat

DOSYAYI OKU İF (YOKSA) print (BULUNAMADI)

while(Dosyanın sonuna kadar)

dosyayı oku

Nokta[i].x ve nokta[y].y ye ata

Noktaları bastır

Dosyayı kapat

Dairex e ana\_fonk() fonksiyonunu ata

dairex.x, dairex.y, dairex.yaricap ekrana bastır

1500\*1000 px ekran oluştur

x0=750 ,a=15 , y0=500

Cizgiler() fonksiyonunu kullan

grafik ekranına noktaları bas

while(0 dan N-1 e 0.0005 artır){

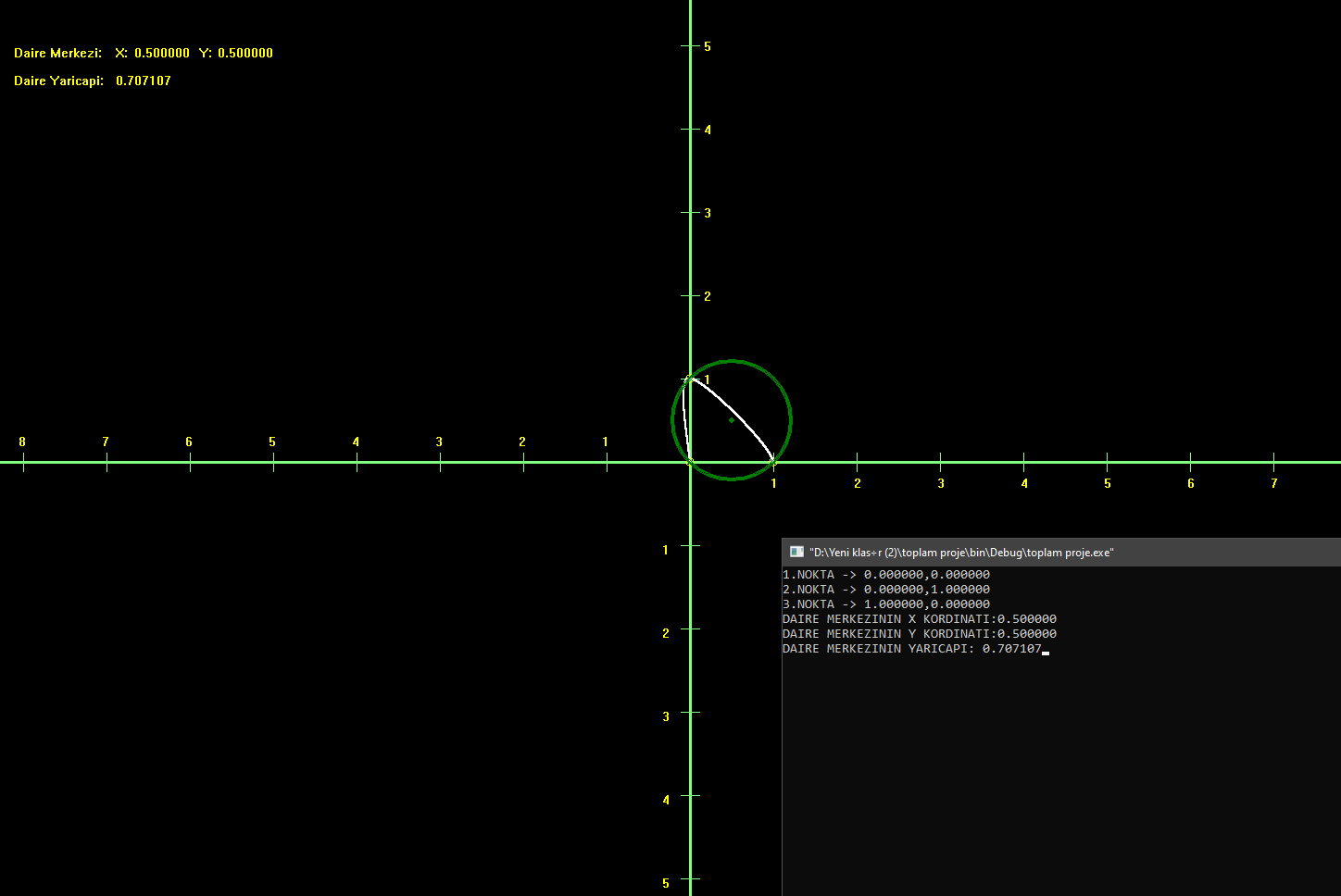
yol fonksiyonunu nokta2 ye ata

nokta2.x ve nokta2.y bas

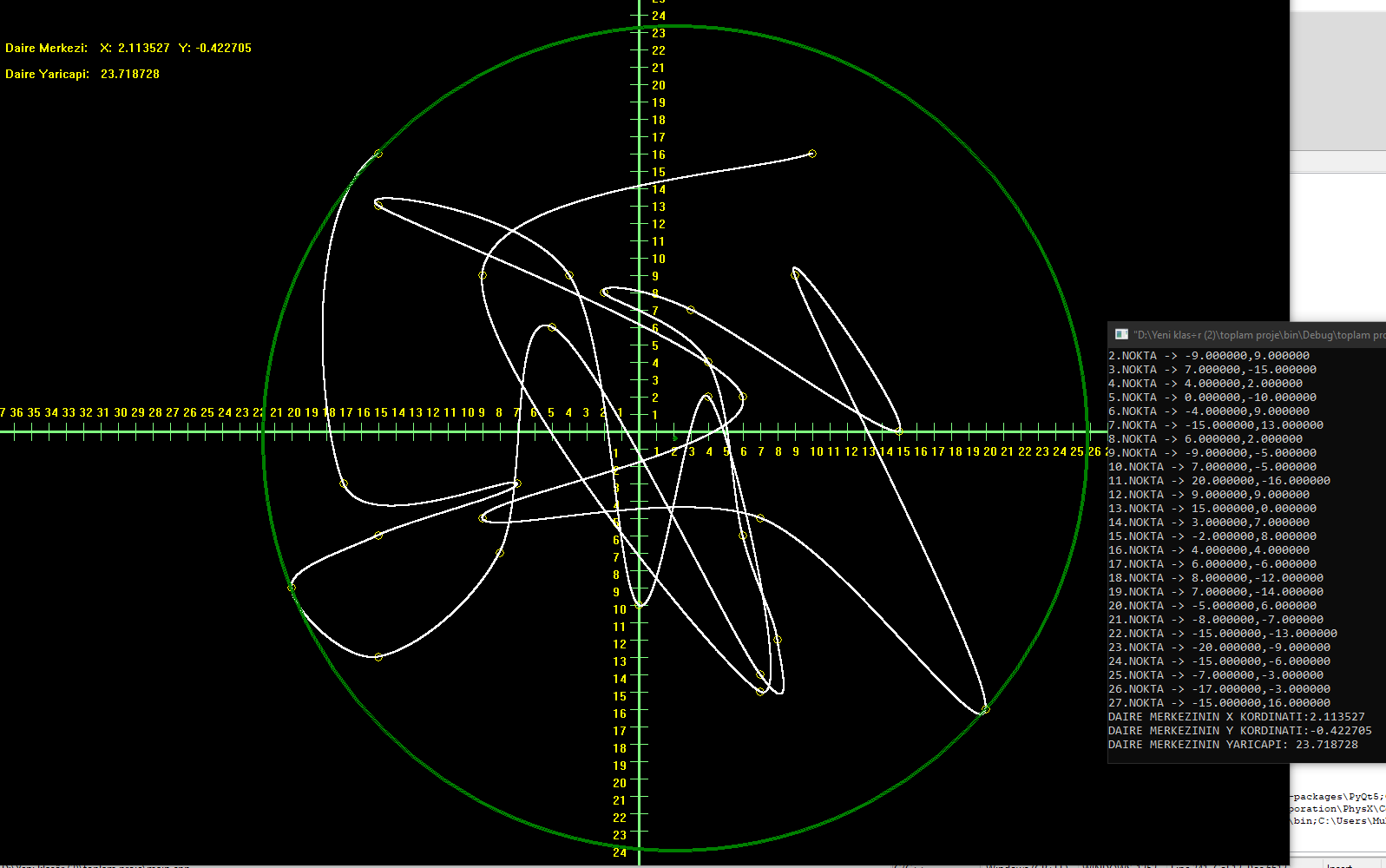
}

Deneysel Sonuçlar

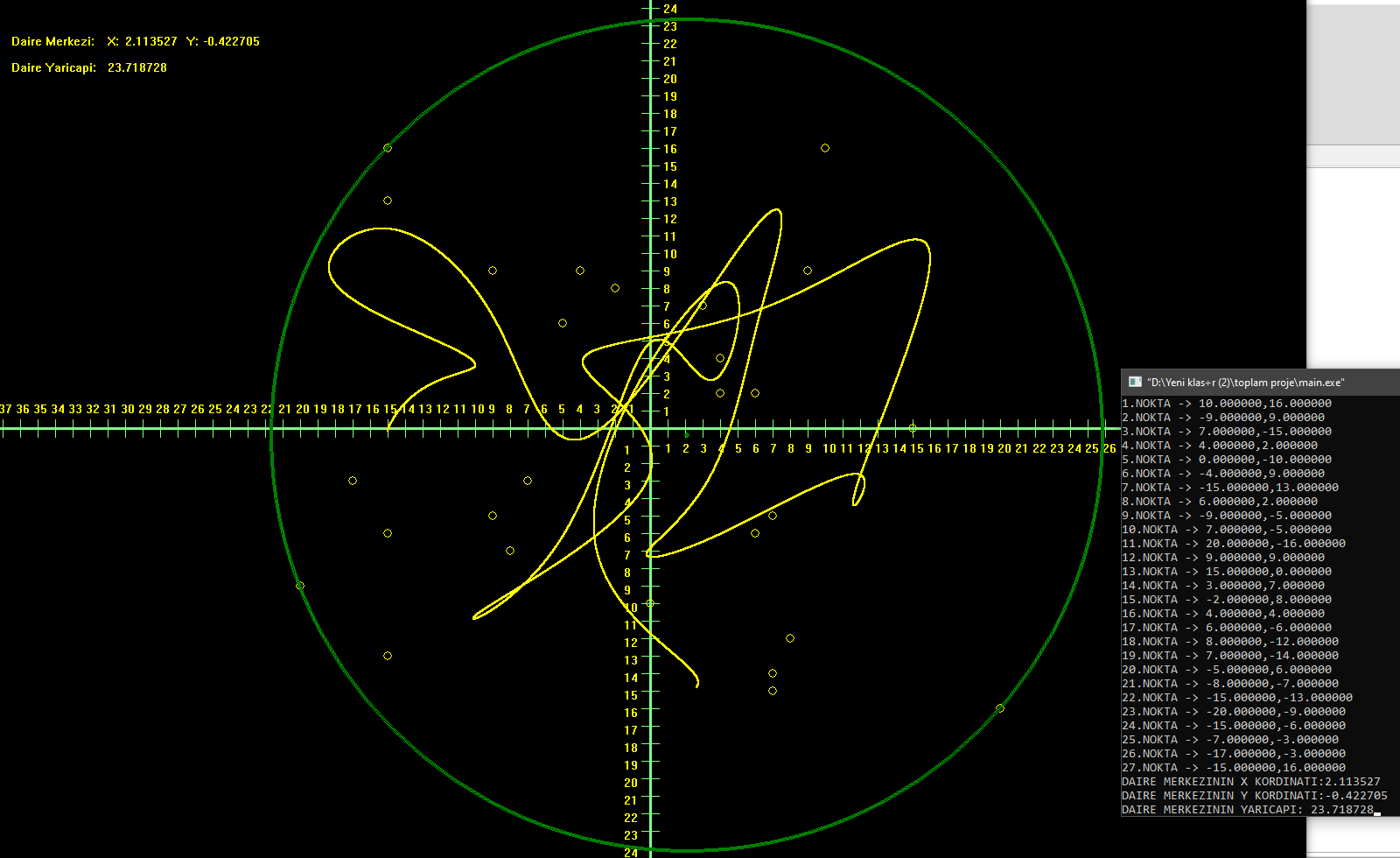
Hermite Spline

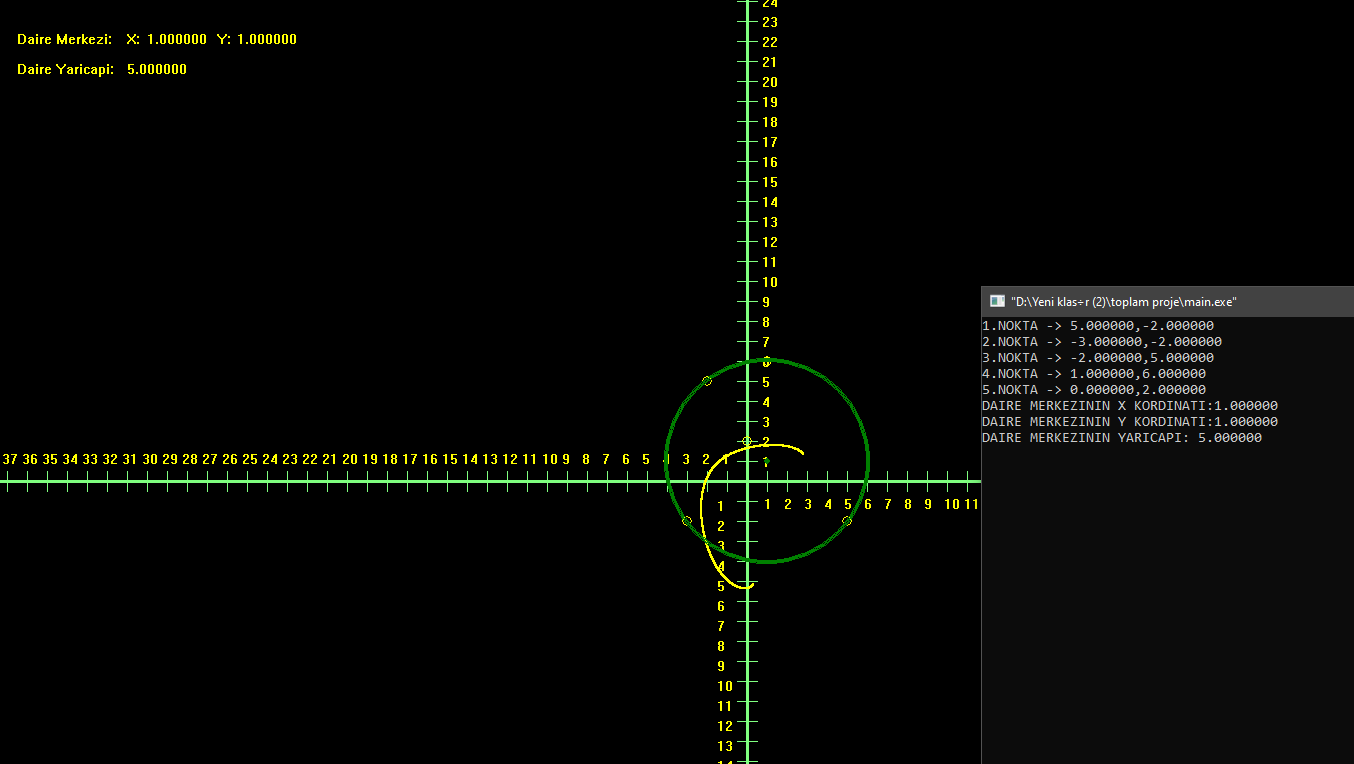


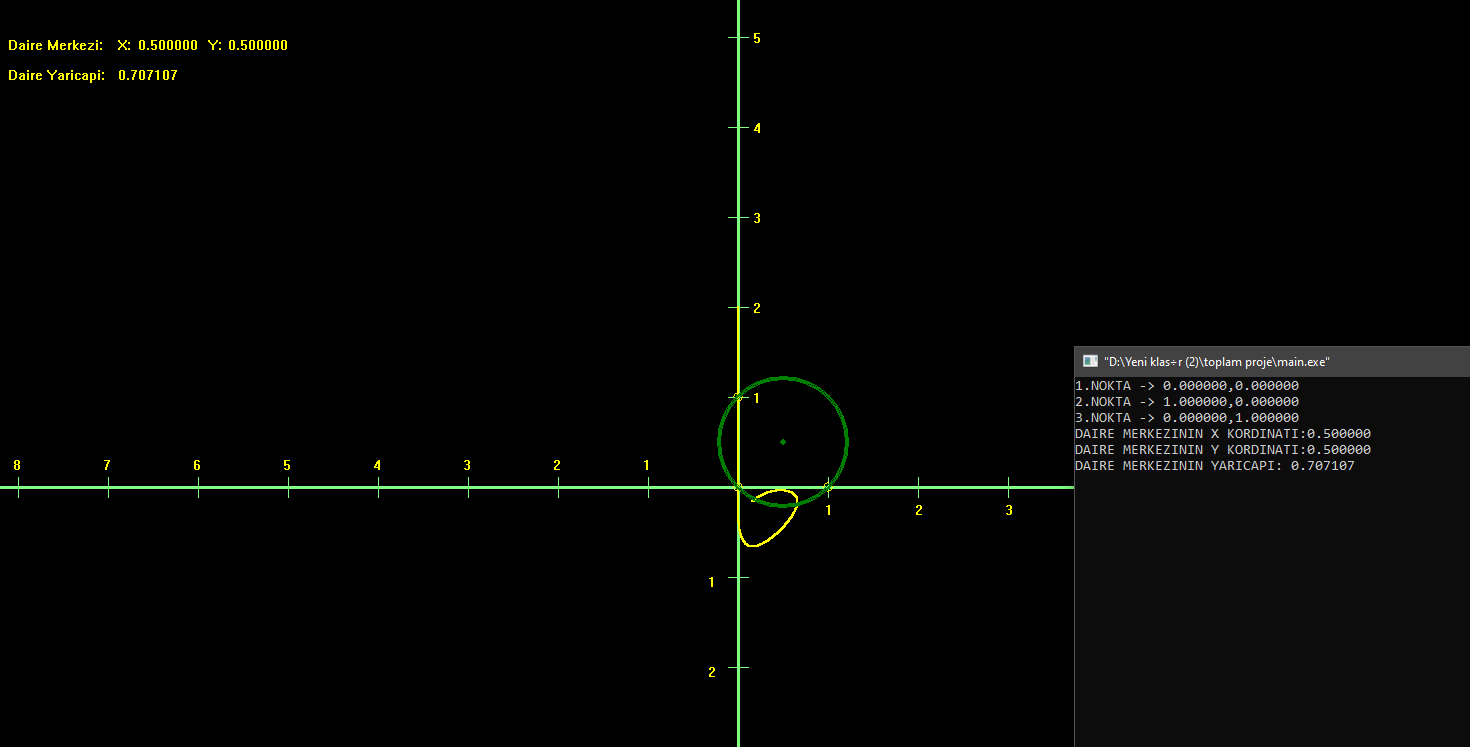




B spline







SONUC:

Projeden matematik formüllerinin nasıl bil yazılım diline uyarlanabileceği uzun kodlarda fonksiyonların düzenleyici ve işlevsel olduğu ve çok uzun süren projelerin hangi aşamalardan ve sıkıntılardan geçerek oluşturulduğu öğrenildi.

Grafik kütüphanelerinin kullanımı öğrenildi. Rapor yazmak hakkında ilk tecrübe edinildi.

İnternetten bilinmeyen bir konuda nasıl araştırma yapılabileceği ve bu konuda İngilizcenin ne kadar etkili olduğu keşfedildi.

Matematik ile yazılım dünyasının ne kadar iç içe olduğu anlaşıldı.

KAYNAKÇA

<https://www.geeksforgeeks.org/draw-circle-c-graphics/>

<https://www.geeksforgeeks.org/minimum-enclosing-circle-set-1/>

<https://www.geeksforgeeks.org/minimum-enclosing-circle-set-2-welzls-algorithm/?ref=lbp>

<https://www.geeksforgeeks.org/equation-of-circle-when-three-points-on-the-circle-are-given/?ref=rp>

<https://www.gamedev.net/tutorials/_/technical/graphics-programming-and-theory/welzl-r2484/>

<https://www.wikiwand.com/en/Smallest-circle_problem>

<https://stackoverflow.com/questions/9653072/return-a-struct-from-a-function-in-c#:~:text=You%20can%20return%20a%20structure,a%20will%20work%20just%20fine>.

<https://www.cs.mcgill.ca/~cs507/projects/1998/jacob/problem.html>

<https://www.geeksforgeeks.org/how-to-write-a-pseudo-code/?ref=leftbar-rightbar>

<https://www.nayuki.io/page/smallest-enclosing-circle>

<https://www.cs.princeton.edu/courses/archive/spring09/cos226/assignments/circle.html>

<http://www.turkcewiki.org/wiki/%C3%87evrel_%C3%A7ember>

<http://www2.cs.uregina.ca/~anima/408/Notes/Interpolation/UniformBSpline.htm>

<https://stackoverflow.com/questions/30035970/b-spline-algorithm>

<http://bilgisayarkavramlari.sadievrenseker.com/2009/08/10/splines-seritler/>

<https://forum.donanimhaber.com/c-dili-ile-grafik-dersleri-graphics-h-quincy--89514478>

<https://youtu.be/9_aJGUTePYo>

<https://www.geeksforgeeks.org/cubic-bezier-curve-implementation-in-c/>

<http://www.nar-associates.com/nurbs/c_code.html#chapter3>

<https://github.com/OneLoneCoder/videos/blob/master/OneLoneCoder_Splines1.cpp>

<https://en.wikipedia.org/wiki/Cubic_Hermite_spline>

<http://www2.cs.uregina.ca/~anima/408/Notes/Interpolation/UniformBSpline.htm>

<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwi8htye7P3sAhWkxYUKHe_lCHcQFjABegQIARAC&url=https%3A%2F%2Fweb.itu.edu.tr%2Fyukselen%2FHM504%2F02Ek-%2520Bezier%2520e%25F0rileri.pdf&usg=AOvVaw1lmGAiOXE0SimwNbIhaSKj>

<https://en.wikipedia.org/wiki/B-spline>