Kocaeli Üniversitesi

Bilgisayar Mühendisliği Bölümü

Programlama Laboratuarı I

Minimumu Çevreleyen Çember-B Spline

Taha Uz

maskeden_2000@hotmail.com

Projenin Özeti:

Programlama Laboratuarı I Projesi olarak bizden "Minimumu Çevreleyen Çember ve B-Spline" adlı uygulama geliştirmemiz beklenmektedir.

Ben proje için C programlama dilini ve CodeBlocks geliştirme ortamını seçtim.

Proje dokümanında bizden beklenen koordinatları belirlediğimiz (.txt veya .csv) dosyanın okunması ve bu dosyadan belirlenen noktalarla oluşan ve bu noktaları kapsayan en küçük çemberin koordinat sisteminde ara yüz tasarımı kullanılarak gösterilmesi bununla birlikte girilen bu noktaların yakınlarından geçen bir eğri oluşturup bu eğrinin de ara yüz tasarımı ile koordinat sisteminde gösterilmesi istenmiştir.

Projede ben C programlama dilini kullanarak dosyadan (.txt) belirlenen koordinatları okuyup bu koordinatları çevreleyen minimum çemberi iki nokta arasındaki uzaklık yardımıyla çemberin yarıçapını ve çemberin sınırında 3 nokta tekniği ile çemberin merkezini determinant yöntemiyle hesapladım. C dilinde bulunan Gaphics.h kütüphanesini kullanarak koordinat sistemi ara yüzü tasarlayıp noktaları ve çemberi bu koordinat sisteminde gösterdim. Bununla birlikte Bezier Spline yöntemini kullanarak noktaların en yakınından geçen eğriyi koordinat sisteminde çizdim.

1.GİRİŞ

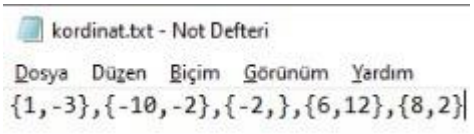
Proje için C programlama dilini ve CodeBlocks geliştirme ortamını kullandım. C programlama dili; taşınabilirliği ve ifade gücü çok yüksek, okunabilirlik özelliği güçlü olan esnek bir dildir. CodeBlocks platformu; özgür açık kaynak kodlu bir C/C++ tümleşik geliştirme ortamıdır. wxWidgets tabanlı tamamen özelleştirilebilir arabirimiyle gelişmiş plugin desteğiyle birçok yardımcı fonksiyon sunan bir geliştirme platformudur.

2.TEMEL BİLGİLER

Projemi; dosyadan koordinat noktalarını okuma, noktaları çevreleyen en küçük çemberin merkez ve yarıçap hesabı, noktaların en yakınından gecen eğri hesabı ve son olarak bu yapılan işlemleri koordinat sisteminde Graphics.h kütüphanesi yardımıyla çizdirmek olmak üzere 4 temel bölüme ayırdım. Bu projede "stdio.h , stdlib.h , math.h , graphics.h , Windows.h" kütüphanelerini kullandım .

2.1.Dosyadan Koordinat Okuma:

Dosyadan koordinatları okuyan algoritma tasarımında ilk olarak dosyanın bulunamaması durumunda kullanıcıya "dosya açılamadı" uyarısını veren koşul durumu ile tasarladım. Bu koşulu sağlar ise devamında dosyadan karakter alma işlemine geçtim böylece dosyadaki tüm koordinatları örnek olarak "{0,0},{0,1},{1,0}" formatıyla alınacak şekilde tasarladım algoritmayı. Burada "char" tipinde olan koordinatlarımı örnek kod olarak bu formatta "dizi[j] = buf[i]-48 " yöntemini kullanarak "int" tipine çevirip teker teker diziye attım. Bu attığım değerler dizide sırasıyla "x1, y1, x2, y2… " şeklinde yerleştirdim. Devamında dizideki değerleri benim işlemlerimde başkarakter olan matrisime attım " For " döngüsü yardımıyla. Böylece bu işlemler sonucunda koordinatların hepsini matrise atamış oldum. Bu değerler pozitif veya negatif sayı olacak şekilde okuyabilen; 1 basamak ve 2 basamak okuyabilecek şekilde tasarladım.Birde kullanıcı örnek olarak "{5,},{2,6},{,0},{,}" bu şekilde değer girdiğinde unutulan o sayılar otomatik "0" değeri atılacak şekilde tasarladım.



Örnek Koordinat. txt Dosyası

2.2.Noktaları Çevreleyen Minimum Çember Hesabı:

Bu işlemi bir fonksiyon başlığı altında yapılcak şekilde algoritmayı oluşturdum. " min_cember_hesap " isimli "void" tipinde fonksiyona ilk olarak 3 tane parametre yolluyoruz. Bu parametrelerin ilki bu işlemler sonunda çemberimizin merkez koordinatları ve yarıçap bilgisini alacak olan "float" tipinde "merkezandyaricap" adında dizi parametresini pointer olarak yolluyorum. İkinci parametre dosyadan okuduğumuz koordinatları tutan "matris" adlı matrisimizi yolluyorum. Son parametre olan ve matrisimizin kaç tane koordinat değeri tutuğuna dair bilgisini atadığım "boyut2" adlı "int" değeri atıyorum "min_cember_hesap" adlı fonksiyona. Bu işlemlerin ardından teker teker matrisin boyutunu kontrol ediyorum çünkü matrisin boyutunu "0" ve "1" olmasına karşın özel işlemler uyguluyorum. Eğer bu koşulları sağlamazsa o zaman asıl zor olan işlemlere başlıyorum. Bu işlemlerde ilk olarak kendim belirlediğim koşulları sağlaması açısından karşılaştıracak ilk çemberi elimle rastgele büyük bir değerde belirliyorum çemberin merkez koordinatını ve yarıçapını. Bu işlemden sonra ilk döngüye giriyorum. Bu döngü ilk kontrol olan "merkez_yaricap" adlı fonksiyona gidiyor. Bu fonksiyon girilen rastgele iki noktayı alıp, iki noktaya eşit uzaklıkta ve aynı hizada olan orta noktayı belirliyor ve bu orta noktayı geçici olarak çemberin merkez noktası olarak belirliyor. Bu kontroller sırasında "dist" adlı fonksiyona orta noktayı ve bu iki noktadan herhangi birini alıp iki nokta arasındaki uzaklığı hesaplıyor. Bu uzaklık çemberimizin yarıçapı oluyor. Bu işlemlerin hepsi bittiğinde ise elimizde geçici çemberimizin hem merkez koordinatı hem de yarıçapı oluyor. Bir sonraki aşama diğer girilen noktalar

çemberin içinde veya sınırında mı olduğunu kontrol eden “nokta_icerde_mi” adlı fonksiyona atıyor bu fonksiyon da “icerde_mi” adlı fonksiyona geçici merkez koordinatı, yarıçapını ve tüm alınan noktaları(dosyadan okuduğumuz koordinatları) yolluyor. “icerde_mi” adlı fonksiyon aldığı noktaları çemberin merkez koordinatları ile arasındaki uzaklığı karşılaştırıyor eğer tüm noktalar çemberin içinde ise yeni geçiçi çember olarak belirleniyor ve bu işlemler en küçük çevreleyen çemberi bulana kadar devam ediyor. Bu “merkez_yaricap ”adlı fonksiyon eğer en küçük çemberi bulamaz ise “merkez_yaricap_2” adlı bu fonksiyon devreye giriyor ve bu fonksiyon içinde bulunduğu döngü ile 3 adet noktayı alarak çemberin merkezini ve yarıçapını verilen 3 noktadan geçen çemberin denklemi hesaplamak için araştırmalarımla bulduğum 7 yöntemden programlanabilirliği en uygun olan “determinant” yöntemi kullanarak çemberin merkez koordinatını ve yarıçapını hesaplayıp yine aynı şekilde noktaları teker teker “noktalar_icerde_mi” ve “icerde_mi” adlı iki fonksiyona yollayarak çemberin denklemini sağlayıp sağlamadığına bakıyorum.Böylece bu döngü noktaları çevreleyen en küçük çemberi bulana kadar devam ediyor .

2.3.Koordinat Noktalarının Yakınından Geçen Eğri:

Bu işlemi bir fonksiyonda yapmayıp “main” fonksiyonun içinde gerçekleştirdim. Çünkü koordinatları içinde bulunduran “matris” adlı matris değişkenini fonksiyona yollamak biraz işleri zorlaştırdığını düşündüm. Belirlenen noktaların yakınından geçen eğriyi oluşturmak için internetten yaklaşık 2-3 günümü alan araştırmalar yaptım. Bezier Spline yönteminin kodlanması ve matematiğinin kolay olması sebebiyle bu yöntemin algoritmasını geliştirmeye başladım.

$$P(u) = \sum_{i=0}^{n} p_i B_{i,n}(u)$$

1.Görsel Eğrinin Ara Kontrol Noktalarını Bulan Formül

$$B_{i,n}(u) = C(n,i)\,t^i\,(1-t)^{n-i}$$

2.Görsel Bi,n(u) Hesaplayan Denklem

Bu yöntemin bir formüle dayalı olduğunu bu formülü “Graphics.h” kütüphanesindeki “putpixel” adlı fonksiyon ile dosyadan aldığımız koordinatların (ilk ve son koordinatların) arasında ara kontrol noktaları ile belirlenen noktalarla tek tek o noktalara çizilen pikseller ile oluşturdum bu eğriyi.

```
for(t=0;t<=1;t+=0.00001){
    x=0;
    y=0;
    for(i=0;i<=boyut2-1;i++){
        x=x+(comb(boyut2-1,i)*pow(1-t,boyut2-1-i)*pow(t,i)*matris[i][0]);
        y=y+(comb(boyut2-1,i)*pow(1-t,boyut2-1-i)*pow(t,i)*matris[i][1]);
    }
    putpixel((x*scale)+midx,(y*(-scale))+midy,YELLOW);
}
```

3.Görsel Bezier Spline Yöntemi Kullanarak Yazdığım Algoritma

2.4.Koordinat Sistemi Ara Yüzü Tasarımı:

Bu ara yüz tasarımında “Graphics.h” kütüphanesini tercih ettim. Bu ara yüzü tasarımını “main” fonksiyonda tasarladım. İlk olarak hesaplamalarımla bulduğum çemberin merkezinin koordinatlarını ve yarıçapını ardından dosyadan okuduğumuz koordinat değerlerini “char” tipine dönüştürdüm. Bunun nedeni “Graphics.h” kütüphanesinde koordinat noktalarını “string” olarak kullanan “outtextxy” adlı fonksiyon . Bu dönüştürmeleri yaptıktan sonra koordinatların değerlerine göre 3 tane ölçek belirledim. Bunları belirlememdeki sebep çok küçük koordinat değerleri girildiğinde kullanıcıya; çok küçük ve iç içe giren nokta değerlerinin ara yüzde birbirine girmesiydi. Bundan dolayı 3 ölçek belirledim ve ölçekleri sırasıyla “5-5 ”,”10-10” ve “25-25”(maksimum x ve y değerleri) olarak belirledim ve kullandığım ekranımın maksimum “y” koordinat düzlemine kadar “x” ‘si sınırladım. Böylelikle daha anlaşılır bir görüntü elde ettim. Alınan koordinatları hem 2 basamaklı hem 1 basamaklı ve bunların negatiflerini duyarlı olabilen hem de merkez koordinatı negatif ve ondalıklı, yarıçapı ise pozitif ve ondalıklı değer alabilecek şekilde tasarladım ve kullanıcıya sundum.

3.KULLANILAN FONKSİYONLAR:

3.1 Benim Oluşturduğum Fonksiyonlar:

int fact();

Bu fonksiyon girilen parametrelerin faktöriyel hesabını yapar.

int comb();

Bu fonksiyon girilen parametrelerin kombinasyon hesabını yapar.

float dist();

Bu fonksiyona girilen (x,y) şeklinde koordinat noktalarının arasındaki uzaklığı hesaplar.

void merkez_yaricap();

Bu fonksiyon ilk matematiksel (2 nokta ile çember oluşturma)yöntemi kullanarak çemberin merkez koordinat hesabını ve yarıçap hesabını yapar.

void merkez_yaricap_2();

Bu fonksiyon 3 koordinat noktası alarak determinant yöntemi ile çemberin merkez koordinatlarını hesaplar ve “dist();” fonksiyonunun yardımı ile bu çemberin yarıçapını bulur.

void nokta_icerde_mi();

Bu fonksiyon girilen koordinatların hepsini ve geçici çemberin tüm bilgilerini alarak döngü yardımıyla “icerde_mi” fonksiyonuna aktarılmasını sağlar.

void icerde_mi();

Bu fonksiyon alınan koordinatlar ile çemberin denklemini kullanarak noktaların çemberin içinde mi dışında mı olduğunu kontrol ediyor.

void min_cember_hesap();

Bu fonksiyon girilen koordinatları parametre olarak alıyor ve “dist”,”merkez_yaricap”,” merkez_yaricap_2”,”noktalar_icerde_mi” ve son olarak “icerde_mi” fonksiyonlarını kullanarak girilen koordinatları çevreleyen en küçük çemberi çizmeye yarıyor.

3.2 Hazır Kullandığım Fonksiyonlar:

fgets();

Dosyadan karakter okumanızı sağlıyor.

fclose();

Dosya ile yapılan işlemler bitince dosyayı kapatmanız gerekli bunun için kullanılan bir fonksiyon.

isdigit();

İçine girilen parametrenin rakam olup olmadığına bakıyor ona göre "True" ya da "False" değer döndürüyor.

initwindow();

Girilen değerlerle ara yüz penceresi oluşturmanızı sağlıyor.

GetSystemMetrics();

Sizin donanımınızın desteklediği maksimum ekran boyutunu alıyor.

textwidth();

Girilen "string" 'in uzunluğunu ölçer.

textheight();

Girilen "string" 'in yüksekliğini (boyutunu)ölçer.

line();

"Graphics.h" kütüphanesine yer alan bir fonksiyondur ve girilen koordinatlar ile size bir çizgi çizmenizi sağlıyor "a" noktasından "b" noktasına.

outtextxy();

Graphics.h kütüphanesine yer alan bir fonksiyondur ve girdiğiniz koordinatlarda ekrana karakter veya karakterler basmanıza yardımcı oluyor.

circle();

Graphics.h kütüphanesine yer alan bir fonksiyondur ve bu fonksiyona girilen "x","y" ve "yariçap" değerleri sırasıyla girildiğinde ekrana çember çizdiriyor.

pow();

Bu fonksiyon " Math.h" kütüphanesinde yer alan bir fonksiyondur ve girilen değerlerin üssü alınmasını sağlar.

sqrt();

Bu fonksiyon " Math.h" kütüphanesinde yer alan bir fonksiyondur ve girilen değerlerin kök alınmasını sağlar.

putpixel();

"Graphics.h" kütüphanesine yer alan bir fonksiyondur ve girilen koordinatların renkli bir şekilde belirtilmesini sağlar.Bunu daha çok döngü yardımıyla bir şeylerin çiziminde kullandım.

getch();

Klavyeden karakter almak için kullanılır.

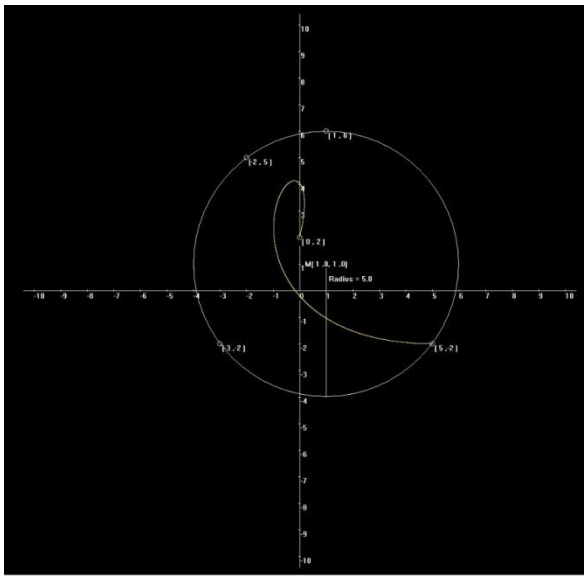
closegraph();

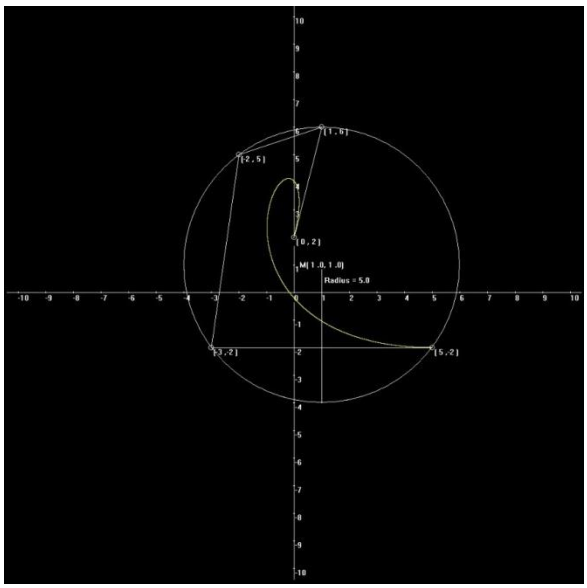
"Graphics.h" kütüphanesine yer alan bir fonksiyondur ve ara yüz ile yapılan işlemler bittiğinde bu fonksiyon ile kapatılması tavsiye edilir.

4.Algoritmanın Zaman Karmaşıklığı Analizi:

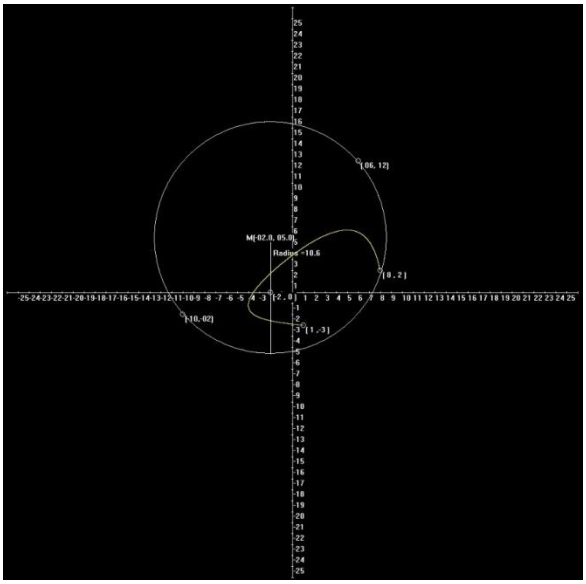
Zaman karmaşıklığı(Time Complexity) , bir programın ya da fonksiyonun işlevini tam anlamıyla yerine getirebilmesi için her işlemden kaç kere yapması gerektiğini gösteren bir bağıntıdır. Benim bu projede kullandığım algoritmalara sırasıyla bakalım. "Main" fonksiyondan ilerlemek gerekirse ilk olarak bizi sınırı sabit "100" olan dosya okuma döngüleri karşılıyor bunu aklımızda tutalım. Sonrasında "min_cember_hesap" fonksiyonuna geçiyor oranın Big O 'suna baktığımızda iç içe 2 tane döngümüz var bunlardan bir tanesi "n^2" kadar diğeri ise "n^3" kadar işlem yapıyor. Bu "min_cember_hesap" fonksiyona baktığımızda **O(n^3)** olduğunu görüyoruz. Main fonksiyonundan devam ettikçe "scale" adlı sabit değişkenin ve "limit" adlı sabit değişkenin burada sınır olduğunu görüyoruz. Bunlar logaritmik artışla devam ediyor her zaman. Yani **O(logn)** oluyor. Devamındaki döngüler yine "boyut2" dediğimiz değişkenle sınırlı kalıyor yani **O(n)** oluyor çünkü her defasında bu "boyut2 " adlı değişken 1 tane döngü ile sınırlı kalıyor. En sonunda kombinasyon ve faktöriyel hesabı yaptığımız fonksiyonlarımıza geliyoruz. Bu fonksiyonda faktöriyel hesabının **O(n)** eşit oluyor işlem hesabının Big O 'su. Böylece tüm programı gözden geçirdiğimizde "min_cember_hesap" adlı fonksiyonda kullandığımız iç içe yapısına sahip olan 3 tane "for" döngüsü bizim zaman karmaşıklığımızın en yüksek derecesi O**(n^3)** bizim en büyük göstergemiz olduğundan bu programın zaman karmaşıklığının **O(n^3)** olduğunu söyleyebiliriz.

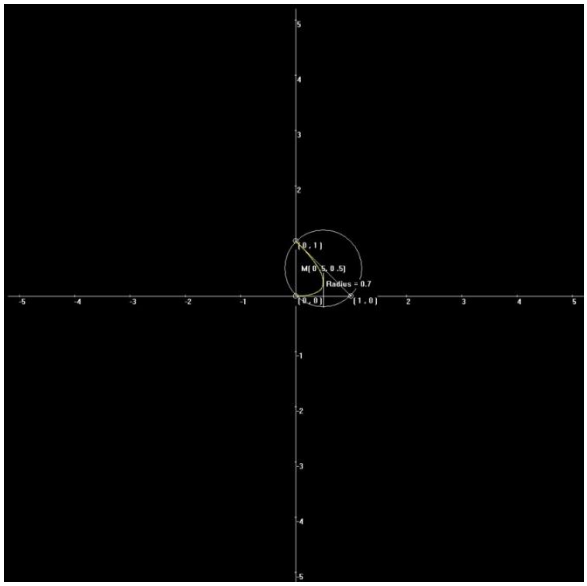
5.DENEYSEL SONUÇLAR:



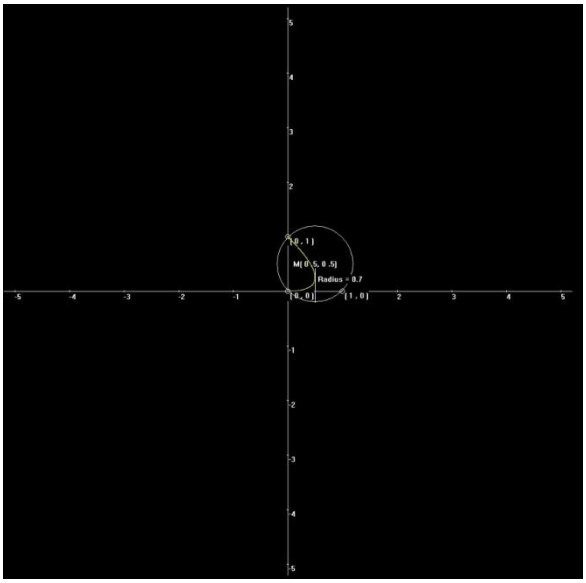



```
Merkez kordinati = {1.0,1.0} ve yaricapi = 5.00(main)

Process returned 0 (0x0)   execution time : 6.426 s
Press any key to continue.
```





```
Merkez kordinati = {-2.0,5.0} ve yaricapi = 10.63(main)

Process returned 0 (0x0)   execution time : 57.890 s
Press any key to continue.
```



6.KARŞILAŞILAN SIKINTILAR ve ÇÖZÜMLERİ:

1.Dosyadan Değer Okurken Fazladan Koordinat Ekleme:

Dosyadan koordinatları okuyum değerleri diziden matrise aktarırken fazladan değer okuduğunu fark ettim dosyanın ve uygulamayı her çalıştırdığımda olmuyordu. 5 kere çalıştırdığımda en fazla 2 çalıştırmamda fazladan koordinat okuyordu. Bu sıkıntıyı dosyayı okurken "{0,0},{1,0},{0,1}" formatında okurken bir koşul durumu belirttim;

```
if(buf[i] == '}' && buf[i+1] != ','){

 break;

}
```

Bu koşul durumuyla sorunu çözdüm.

2.Merkez Koordinat Noktamın Değerlerinden Herhangi Biri "0" ise Noktayı Okumuyordu:

Çemberin merkez koordinatlarını, koordinat noktasına "int" değerden "char" değerine dönüştürürken koşul durumlarında "0" durumunu belirtmeyi unuttuğum için merkez koordinat değerini okumuyormuş. Koşul durumunda ">=0 " belirtince hata ortadan kalktı.

3.Koordinat Sistemine Sınırları Belirlerken "Outtextxy" Fonksiyonunun Kullanımını Bilmediğimden Kaynaklanan "Warning" Hataları:

Koordinat sistemine sınırları belirlerken "outtextxy" fonksiyonunun kullanımını bilmediğimden kaynaklanan "Warning" hataları sayesinde "outtextxy" 'yi yanlış kullandığımı fark ettim.

Outtextxy(midx,midy,"0");(**HATALI KULLANIMIM**)

Yukarıdaki gibi kullanınca uyarı alıyordum. Bu uyarıyı "0" 'ı bir "char" tipindeki "a" değişkenine atınca düzeldi. Bunun nedeni "outtextxy" 'nin "char" değil "char dizisi" (yani String) ile değer aldığını fark ettim.

4. **Koordinat Sistemine Sınırları Belirlerken "Outtextxy" Fonksiyonunun Kullanımını Bilmediğimden Kaynaklanan Gereksiz Çok Satır Kullanımı:**

Koordinat sisteminde sınır rakamlarını girerken örnek olarak ;(**HATALI KULLANIMIM**)

Outtextxy(midx,midy,"0");

Outtextxy(midx,midy,"1");

……

Gibi devam eden satırları "0" ve "1" yazmak yerine "string" tipindeki "a" 'değişkenini yazıp ona döngüler ile teker teker değer atayarak 150 satırı 20 satıra indirdim.

7.SONUÇ:

Bu proje sayesinde; C" diline yeniden hâkim olmayı ve "C" dilinde nasıl ara yüz tasarlanacağına, tasarlanırken nasıl fonksiyonlar kullanılacağını bana öğretti. Bununla birlikte bir algoritmanın zaman karmaşıklığının nasıl hesaplanacağını, bir proje tasarlarken nasıl araştırma yapılacağını ve nelere dikkat edileceğini bana zor yoldan öğretti.

8.KAYNAKÇA:

1.Spline Çeşitleri ve B-Spline hakkında bilgi.

http://acikerisimarsiv.selcuk.edu.tr:8080/xmlui/bitstream/handle/123456789/9508/212423.pdf?sequence=1&isAllowed=y

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

2.Code Blokcs hakkında bilgi.

https://tr.wikipedia.org/wiki/Code::Blocks

3."putpixel()" fonksiyonunun kullanımına dair bilgi.

https://www.geeksforgeeks.org/putpixel-function-c/#:~:text=The%20header%20file%20graphics.,the%20color%20of%20the%20pixel

4.”outtextxy()”fonksiyonunun kullanımına dair bilgi.

https://www.youtube.com/watch?v=pOarmtkSI3U&ab_channel=techNprog

5. “Graphics.h” kütüphanesindeki tüm fonksiyonlar

https://www.programmingsimplified.com/c/graphics.h

6. “Graphics.h” kütüphanesindeki fonksiyonların kullanımlarını öğreten eğitim video serisi.

https://www.youtube.com/watch?v=WzYQMLyoDEk&list=PL5UFsTza4wWSNhe0xuO6ELw7ORU-UHNDO&index=6&ab_channel=VCRGames

7. Minimumu Çevreleyen Çember Merkezini hesaplarken kullandığım matematiksel yöntemler .

https://www.youtube.com/watch?v=nDDYfLVxixw&ab_channel=SundarMath

https://www.qc.edu.hk/math/Advanced%20Level/circle%20given%203%20points.htm

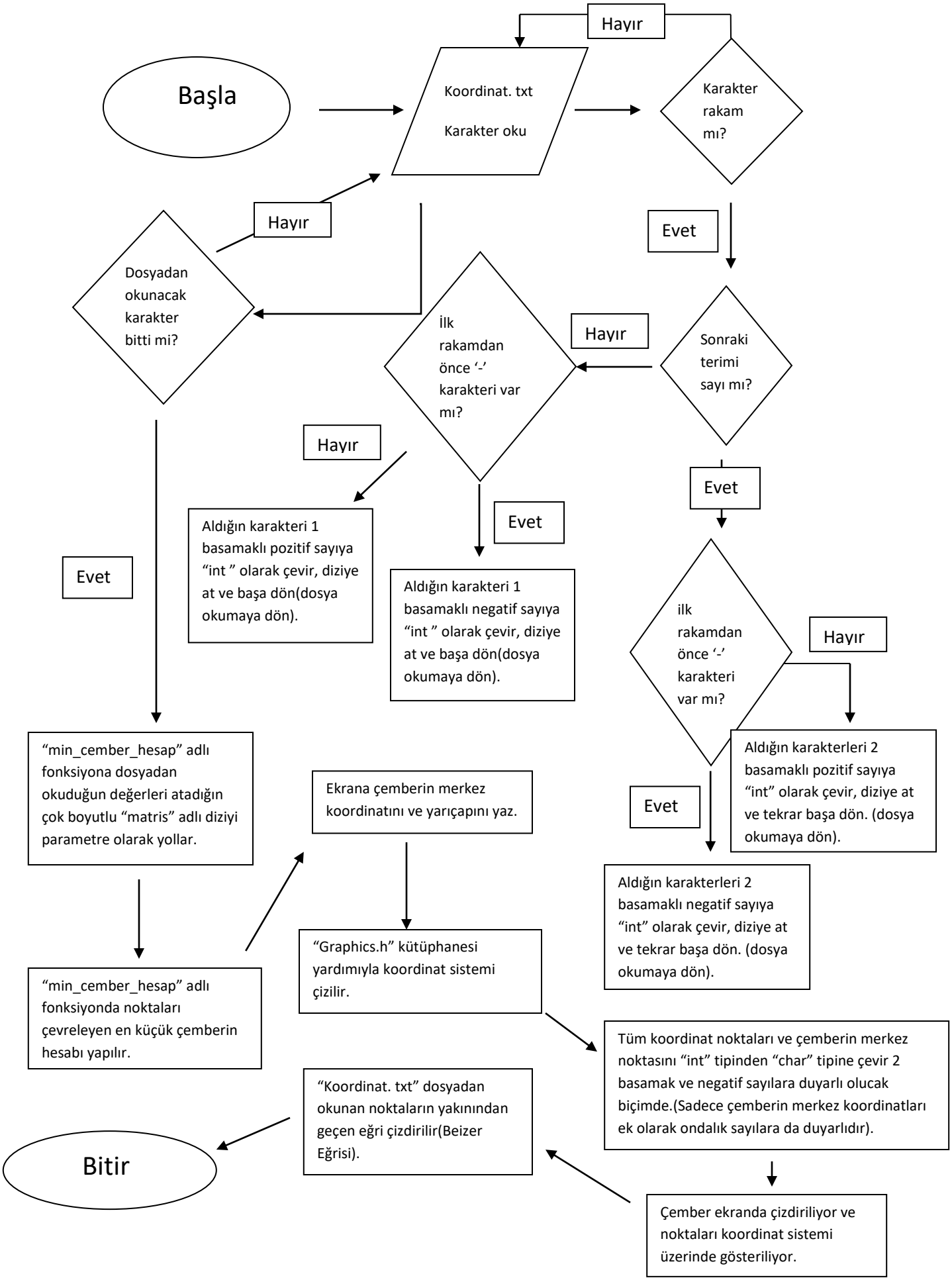
8.”Graphics.h” kütüphanesinin CodeBlocks 17.12 sürümüne kurmak için yardımcı video.

https://www.youtube.com/watch?v=GM0kni4jdPY&ab_channel=CodeWar

9.Kodda zaman karmaşıklığının nasıl hesaplanır [Big O()].

http://cagataykiziltan.net/programin-calisma-hizi-ve-algoritma-verimliligi/zaman-karmasikligi-ve-buyuk-o-notasyonu-time-complexity-and-big-o-notation/

https://www.youtube.com/watch?v=XQqjUSOi45o&ab_channel=BilgisayarKavramlari


Başla
Koordinat. txt
Karakter oku
Karakter rakam mı?
Hayır
Evet
Hayır
Dosyadan okunacak karakter bitti mi?
İlk rakamdan önce '-' karakteri var mı?
Hayır
Sonraki terimi sayı mı?
Hayır
Evet
Hayır
Aldığın karakteri 1 basamaklı pozitif sayıya "int " olarak çevir, diziye at ve başa dön(dosya okumaya dön).
Evet
Aldığın karakteri 1 basamaklı negatif sayıya "int " olarak çevir, diziye at ve başa dön(dosya okumaya dön).
Evet
ilk rakamdan önce '-' karakteri var mı?
Hayır
Aldığın karakterleri 2 basamaklı pozitif sayıya "int" olarak çevir, diziye at ve tekrar başa dön. (dosya okumaya dön).
Evet
Aldığın karakterleri 2 basamaklı negatif sayıya "int" olarak çevir, diziye at ve tekrar başa dön. (dosya okumaya dön).
Evet
"min_cember_hesap" adlı fonksiyona dosyadan okuduğun değerleri atadığın çok boyutlu "matris" adlı diziyi parametre olarak yollar.
"min_cember_hesap" adlı fonksiyonda noktaları çevreleyen en küçük çemberin hesabı yapılır.
Ekrana çemberin merkez koordinatını ve yarıçapını yaz.
"Graphics.h" kütüphanesi yardımıyla koordinat sistemi çizilir.
Tüm koordinat noktaları ve çemberin merkez noktasını "int" tipinden "char" tipine çevir 2 basamak ve negatif sayılara duyarlı olucak biçimde.(Sadece çemberin merkez koordinatları ek olarak ondalık sayılara da duyarlıdır).
Çember ekranda çizdiriliyor ve noktaları koordinat sistemi üzerinde gösteriliyor.
"Koordinat. txt" dosyadan okunan noktaların yakınından geçen eğri çizdirilir(Beizer Eğrisi).
Bitir