

# Minumum Çevreleyen Çember (Minimum Enclosing Circle(MEC))-B Spline

Arda Talu

Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümü

170202037 [ardatalu.98@gmail.com](mailto:ardatalu.98@gmail.com)

## Özet

Projede istenen kullanıcı tarafından tamsayı koordinatlı 2 boyutlu bir düzlemde N nokta verildiğinde, tüm noktaları içeren minimum çevreleyen yarıçaplı daireyi çizdirmek, verilen N noktanın en yakınından geçen eğriyi çizdirmek ve çizdirmiş olduğumuz dairenin ise yarıçapını ve merkezini hesaplamamız istenmektedir.

## 1.Giriş

Programlama Dili: Proje C dili kullanılarak gerçekleştirilmiştir.

Arayüz Tasarım Kütüphanesi: OpenGL

Kullanılan Araçlar: Proje Code::Blocks 17.12 sürümü ve Notepad++ araçlarından faydalanılarak yapılmıştır.

## 2.Algoritma

Program başlatıldığında kullanıcıdan noktaların bulunduğu dosyanın adını yazması istenir.

Eğer dosya bulunamazsa hata verilip çıkılır.

Dosya bulunabilirse dosyanın içindeki noktalar okunur.

Okunan noktalar fonksiyonlardaki yerlerine göre çalıştırılır.

Noktalar, noktaları çevreleyen en küçük yarıçaplı çember ve B-Spline eğrisi çizdirilir.

Çizdirilen dairenin yarıçapı ve merkezi hesaplanır. Program kapatılır.

## 3.Sözde Kod

1. Başla
2. GL/gl.h kütüphanesini ekle
3. GL/glu.h kütüphanesini ekle
4. GL/glut.h kütüphanesini ekle
5. iostream kütüphanesini ekle
6. fstream kütüphanesini ekle
7. stdio.h kütüphanesini ekle
8. stdlib.h kütüphanesini ekle
9. conio.h kütüphanesini ekle
- 10.string.h kütüphanesini ekle
- 11.math.h kütüphanesini ekle
- 12.vector kütüphanesini ekle
- 13.GLfloat ctrlpoints[100]
- 14.void init(void)
- 15.glClearColor
- 16.glShadeModel(GL\_FLAT)
- 17.glMap1f(GL\_MAP1\_VERTEX\_3, 0.0, 1.0, 3, 4, &ctrlpoints[0][0])
- 18.glEnable(GL\_MAP1\_VERTEX\_3)
- 19.void display(void)
- 20.int i
- 21.glClear(GL\_COLOR\_BUFFER\_BIT)
- 22.glColor3f(1.0, 1.0, 1.0)
- 23.glBegin(GL\_LINE\_STRIP)
- 24.İçin(i = 0; i <= 30; i++)
- 25.glEvalCoord1f((GLfloat) i/30.0)
- 26.glEnd()
- 27.glPointSize(10.0)
- 28.glColor3f(1.0, 1.0, 0.0)
- 29.glBegin(GL\_POINTS)
- 30.İçin (i = 0; i < 4; i++)
- 31.glVertex3fv(&ctrlpoints[i][0])
- 32.glEnd()
- 33.glFlush()
- 34.glColor3ub(511,0,0)
- 35.glLineWidth(1)
- 36.glBegin(GL\_LINES)

```

37.glVertex2f(0.0,10.0)
38.glVertex2f(0.0,-10.0)
39.glEnd()
40.glFlush()
41.glColor3ub(511,0,0)
42.glLineWidth(1)
43.glBegin(GL_LINES)
44.glVertex2f(10.0,0.0)
45.glVertex2f(-10.0,0.0)
46.glEnd()
47.glFlush()
48.glColor3ub(511,0,0)
49.glLineWidth(1)
50.glBegin(GL_LINES)
51.glVertex2f(9.5,0.5)
52.glVertex2f(10.0,0.0)
53.glEnd()
54.glFlush()
55.glColor3ub(511,0,0)
56.glLineWidth(1)
57.glBegin(GL_LINES)
58.glVertex2f(0.0,10.0)
59.glVertex2f(-0.5,9.5)
60.glEnd()
61.glFlush()
62.glColor3ub(511,0,0)
63.glLineWidth(1)
64.glBegin(GL_LINES)
65.glVertex2f(0.0,10.0)
66.glVertex2f(0.5,9.5)
67.glEnd()
68.glFlush()
69.glColor3ub(511,0,0)
70.glLineWidth(1)
71.glBegin(GL_LINES)
72.glVertex2f(10.0,0.0)
73.glVertex2f(9.5,-0.5)
74.glEnd()
75.glFlush()
76.glutSwapBuffers()
77.void reshape(int w, int h)
78.glViewport(0, 0, (GLsizei) w, (GLsizei) h)
79.glMatrixMode(GL_PROJECTION)
80.glLoadIdentity()
81.Eğer (w <= h)
82.glOrtho(-10.0, 10.0, -
    10.0*(GLfloat)h/(GLfloat)w,10.0*(GLfloat
    )h/(GLfloat)w, -10.0, 10.0)
83.Değilse

```

```

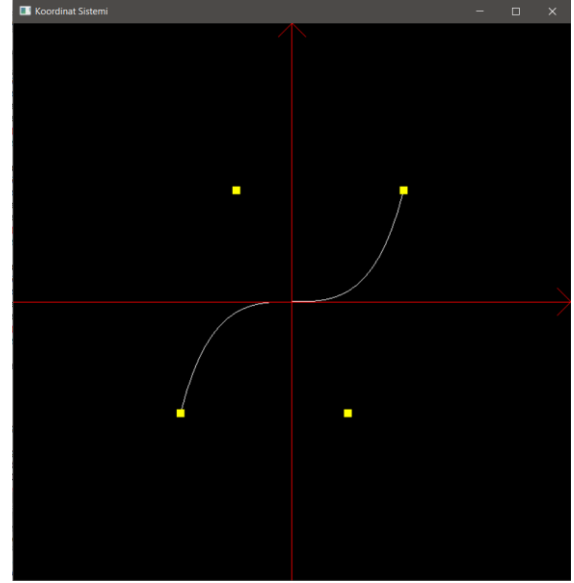
84.glOrtho(-
    10.0*(GLfloat)w/(GLfloat)h,10.0*(GLfloat
    )w/(GLfloat)h, -10.0, 10.0, -10.0, 10.0)
85.glMatrixMode(GL_MODELVIEW)
86.glLoadIdentity()
87.using namespace std
88.double mesafe(double a, double b)
89.return a*a + b*b
90.int main(int argc, char** argv)
91.char dosyaAdi[20]
92.double x[1005], y[1005], X, Y, d, e
93.int i,n,satir=0
94.Oku ("Dosya adini yaziniz: ")
95.Yaz ("%s",dosyaAdi)
96.FILE *dosya = fopen(dosyaAdi, "r")
97.Eğer (dosya==NULL)
98.Oku("Dosya acilamadi!\n")
99.Çıkış(1)
100.char ch
101.İken(ch!=EOF)
102.ch=fgetc(dosya)
103.Eğer(ch=='\n')
104.satir++
105.Oku("%d satir\n",satir)
106.Oku("Noktalar: \n")
107.İken(fscanf(dosya,"%lf , %lf", &x[i],
    &y[i])!=EOF)
108.İçin(i=0; i<satir; i++)
109.fscanf(dosya,"%lf , %lf", &x[i], &y[i])
110.Oku("{ %lf , %lf},", &x[i], &y[i])
111.Oku("\n")
112.İçin (int i = 0; i < n; i++)
113.scanf("%lf%lf", &x[i], &y[i])
114.X += x[i]; Y += y[i]
115.X /= n; Y /= n
116.double P = 0.1
117.İçin (int i = 0; i < 30000; i++)
118.int f = 0
119.d = mesafe(X - x[0], Y - y[0])
120.İçin (int j = 1; j < n; j++)
121.e = mesafe(X - x[j], Y - y[j])
122.Eğer (d < e) { d = e; f = j
123.X += (x[f] - X)*P
124.Y += (y[f] - Y)*P
125.P *= 0.999
126.Oku("%.3lf %.3lf\n%.3lf", X, Y, sqrt(d))
127.glutInit(&argc, argv)
128.glutInitDisplayMode (GLUT_SINGLE |
    GLUT_RGB)
129.glutInitWindowSize(720,720)
130.glutInitWindowPosition(360,50)

```

```
131. glutCreateWindow("Koordinat Sistemi ")
132. init()
133. glutDisplayFunc(display)
134. glutReshapeFunc(reshape)
135. glutMainLoop()
136. fclose(dosya)
137. return 0
```

## 4.Sonuç

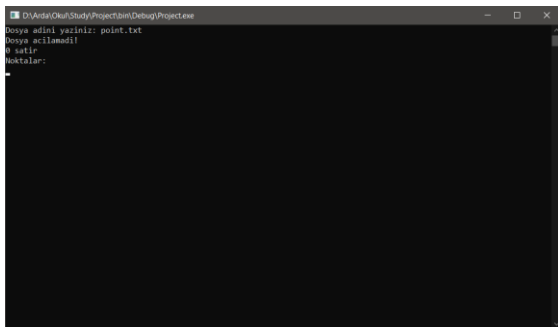
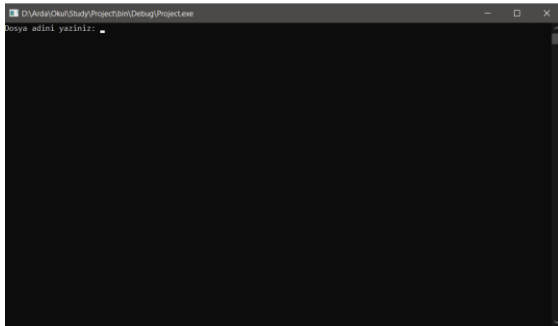
Kullanıcıya dosyada verdiği noktalara göre çevreleyen en küçük çember, çemberin yarı çapı, merkezi, noktalara göre hesaplanmış B-Spline'ı gösteren program yazılmıştır.



## 5. Karşılaşılan Problemler

OpenGL kütüphanesinin kaynaklarının çoğunlukla C++ diline uyumlu olması ve henüz tam olarak kaynağını fark edemediğim bir şekilde aynı kodların başka projelerde dosya adı yazıldığında açmasına rağmen bu projede fopen dosya açma komutuyla dosyayı okuma modunda açamadım.

## 6.Ekran Görüntüleri



## 7. Kaynakça

[1] <https://www.geeksforgeeks.org/getting-started-with-opengl/>

[2] <https://docplayer.biz.tr/60393521-B-spline-egrisi.html>

[3] <https://www.glprogramming.com/red/chapter12.html>

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

[5] <https://www.geeksforgeeks.org/minimum-enclosing-circle-set-2-welzl-algorithm/>

[6] <https://ideone.com/vju6Uh>