

II. gyakorlat - Jegyzőkönyv

Ekart Csaba - ZWPMKP

1. feladat

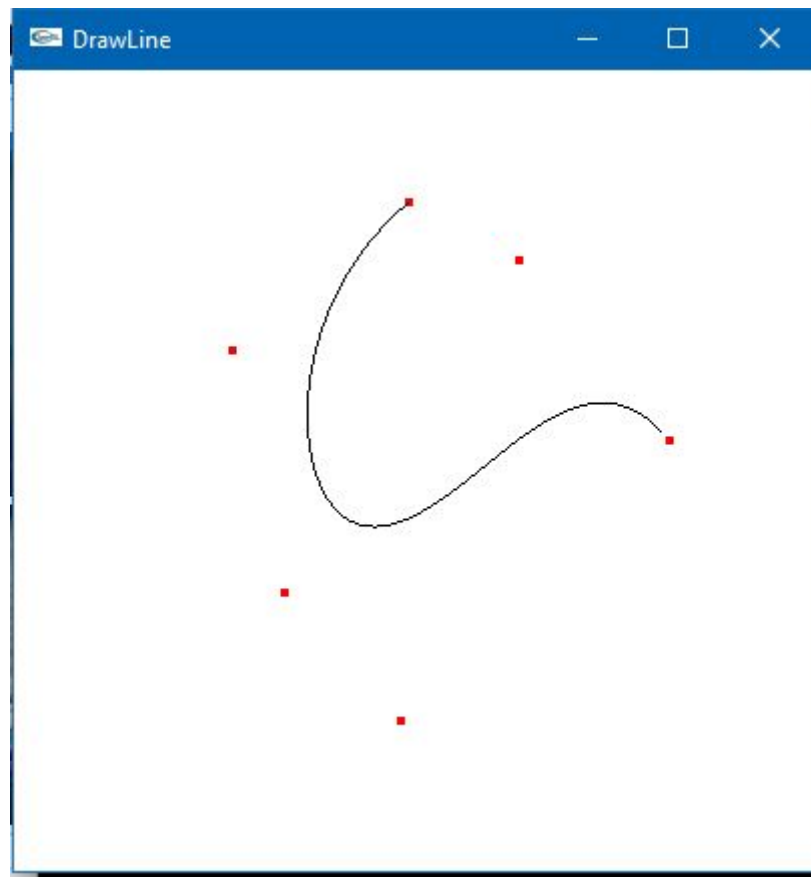
Az első feladatban a Lagrange görbét kellett elkészíteni. A pdf-ben olvasottak, és az órán látottak alapján csináltam meg. Az előző órán elkészített pontrajzoló programot felhasználtam. A piros pöttyök kirajzolásához a GL_QUADS parancsot használtam, és a glColor3ub paranccsal állítottam a színükön. A "pont" típus megalkotása már előző órán megtörtént.

Az érdekes kód részlet a megadottakon kívül a drawLagrange függvény volt. Ebben ténylegesen kirajzoljuk a pontokat az eddig megadott függvények segítségével. A j ebben az esetben a "beosztás" sűrűségét adja meg, minél kisebbre vesszük annál simább görbét kaphatunk.

```
70 void drawLagrange()
71 {
72     for (int i=0; i< points.size(); i++) {
73         knotVector[i]=(double)i/(double(points.size()-1));
74     }
75
76     point a(0,0);
77     point b(0,0);
78
79     a = CalcLagrangePoint(0);
80     for(double j = 0.01; j <= 1; j += 0.01)
81     {
82         b = a;
83         a = CalcLagrangePoint(j);
84
85         glBegin(GL_LINES);
86
87         glVertex2d(b.x, b.y);
88         glVertex2d(a.x, a.y);
89
90         glEnd();
91     }
```

2. feladat

A feladat megoldásához a diában megadottak alapján csak az L függvényt cseréltem B-re, illetve a calcLagrangePoint-ot calcBezierPointra, és a drawLagrange-ot drawBezierre cseréltem. A megadott függvényekben lévő hibák javítás után ez a program is tökéletesen működött.



3. feladat

A harmadik feladatban a NURBS görbét kellett megvalósítani. A pdf-ben megadottak alapján az inicializáló függvényem:

```
42 void SajatInit() {
43     theNurb= gluNewNurbsRenderer();
44     gluNurbsProperty(theNurb, GLU_SAMPLIN_TOLERANCE, 25.0);
45     gluNurbsProperty(theNurb, GLU_DISPLAY_MODE, GLU_FILL);
46 }
```

A rajzoló pedig az alábbi. Az egyetlen érdekesség amit el kellett végezni, hogy a knots beosztását a megfelelő szinten tartsuk a feladatban leírtak alapján. Ezenkívül mindenütt módosítani kellett a myPoint típuson, amely innentől kezdve nem használt, hanem helyette egy 2 dimenziós tömb (szintén a feladatban leírtak miatt).

```
glColor3ub(0,0,0);
for(int i = 0; i <= ptnum + ORDER; i++) {
    knots[i] = i;
}

gluBeginCurve(theNurb);
gluNurbsCurve(theNurb, ptnum + ORDER, knots, 3, &ctrlpoints[0][0], ORDER, GL_MAP1_VERTEX_3);
gluEndCurve(theNurb);

glFlush();
```

4. feladat

A negyedik feladatban Catmull-Clark algoritmus implementálása volt a feladat. Sajnos az algoritmusról csak 3d-ben találtam információkat az interneten, de még arról is keveset, így a megvalósítása nem sikerült.