

Oppgave 4: Rasterisering

- a) Beskriv hvordan man kan utvide DDA-algoritmen til å rasterisere sirkler. (8%)

DDA (Digital Differential Analyzer) er i pensumboken beskrevet som en linjerasteriseringsalgoritme som benytter stigningstallet for å beregne neste piksel som skal tegnes. Dersom man itererer over x (x er en fri variabel), vil neste y -verdi finnes ved hjelp av:

$$y_{n+1} = y_n + \frac{\Delta y}{\Delta x} = y_n + m$$

Dersom man itererer over y , vil man beregne neste x ved hjelp av følgende formel:

$$x_{n+1} = x_n + \frac{1}{m}$$

For linjer må man skifte mellom å iterere over x og å iterere over y basert på om stigningstallet, er større eller lik en. Ved rasterisering av sirkler benytter man seg i stedet av symmetri.

Poenget er å legge merke til at man finner neste verdi ved hjelp av stigningstallet og så benytte stigningstallet til en sirkel. Stigningstallet til en sirkel kan, for eksempel, finnes ved implisitt derivasjon av sirkelligningen:

$$\begin{aligned}x^2 + y^2 &= r^2 \\2x + 2y \cdot y' &= 0 \\y' &= -\frac{2x}{2y} = -\frac{x}{y}\end{aligned}$$

Ettersom stigningstallet forandrer seg hele tiden blir formelen for å approksimere y_{n+1} som følger:

$$y_{n+1} = y_n + y'_n = y_n + m_n = y_n - \frac{x_n}{y_n}$$

Denne tilnærmingen tar med enkelte piksler som ligger utenfor sirkelen som man vil finne (eksakt) ved hjelp av sirkelligningen.

Enkelte studenter benyttet følgende formel:

$$y_{n+1} = y_n + y'_n = y_n - \frac{x_n}{\sqrt{r^2 - x_n^2}}$$

Denne gir riktig resultat, men roten under brøkstreken er jo et uttrykk for y_n og er dermed overflødig (utfører unødvendige og krevende beregninger – $\text{sqrt}()$ og $\text{pow}()$). Om x_{n+1} benyttes i roten gir det en feil som fører til at sirkelen blir mindre enn om x_n benyttes.

Dersom man utnytter sirkelens symmetri er det kun nødvendig å regne ut verdier i sirkelens første oktant ($0 \leq \theta \leq \frac{\pi}{4}$).

b) Skriv pseudokode for algoritmen du har beskrevet i a). (8%)

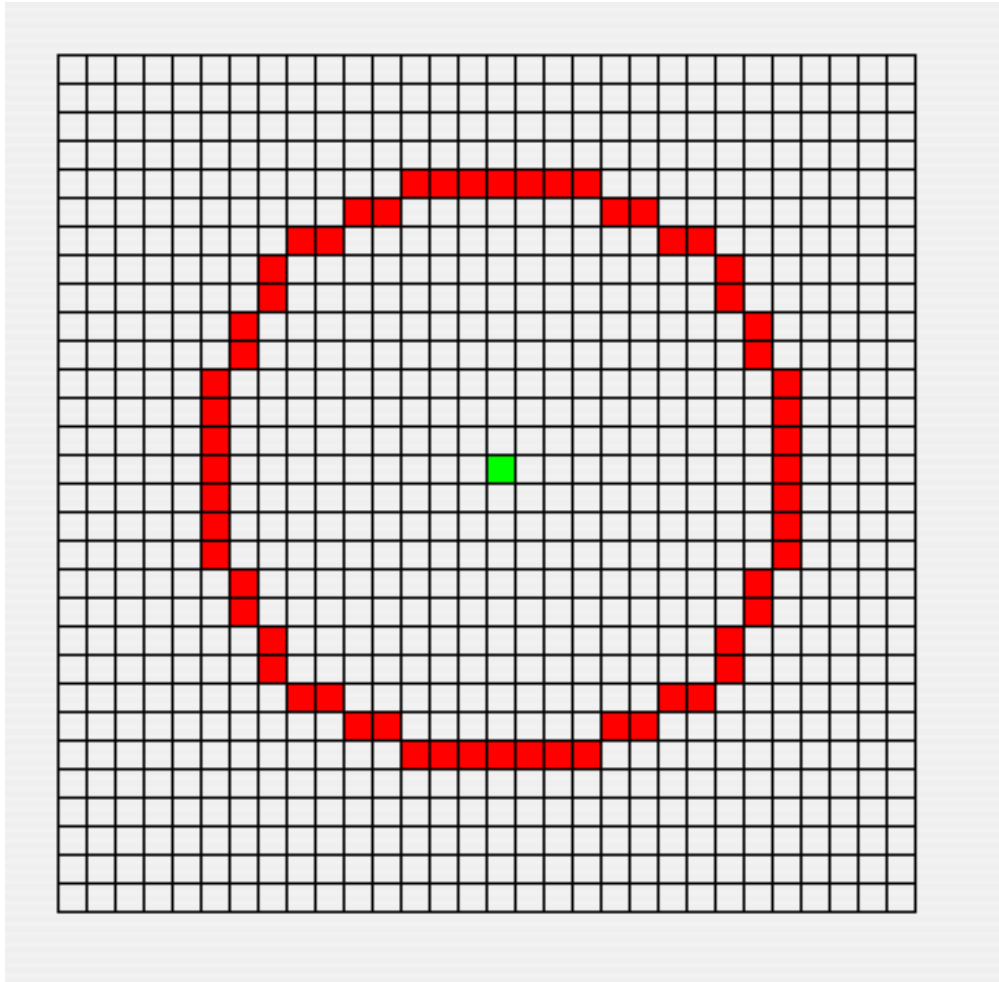
```
ddaCircle (int xOrigin, int yOrigin, int radius) {
    double x = 0;
    double y = radius;
    double m = -x/y;

    while (Math.round(x) <= Math.round(y)) {
        plot8(xOrigin, yOrigin, Math.round(x), Math.round(y));
        m=-x/y;
        x+=1;
        y+=m;
    }
}

plot8(int xOrigin, int yOrigin, int xOffset, int yOffset){
    plot4(xOrigin, yOrigin, xOffset, yOffset);
    plot4(xOrigin, yOrigin, yOffset, xOffset);
}

plot4(int xOrigin, int yOrigin, int xOffset, int yOffset){
    setPixel(xOrigin + xOffset, yOrigin + yOffset);
    setPixel(xOrigin + xOffset, yOrigin - yOffset);
    setPixel(xOrigin - xOffset, yOrigin + yOffset);
    setPixel(xOrigin - xOffset, yOrigin - yOffset);
}
```

c) Tegn en sirkel med sentrum i (0,0) og radius 10 ved hjelp av pseudokoden du har beskrevet i b). List opp i en tabell verdiene til viktige variable for hvert punkt som tegnes. (4%)



Figur 1: Grønn prikk viser piksel (0,0).

Tabell 1: Lister opp viktige variable ved eksekvering av pseudokode gitt i b)

x	y	m
0.0	10.0	0.0
1.0	10.0	0.0
2.0	9.90	-0.1
3.0	9.70	-0.20
4.0	9.39	-0.31
5.0	8.92	-0.43
6.0	8.40	-0.56
7.0	7.69	-0.71

Oppgave 5: Linjeklipping

- Beskriv Liang-Barskys algoritme for linjeklipping. (8%)
- Klipp linjen med endepunkter i (0,0) og (5,4) mot klippevinduet med nedre venstre hjørne (2,0) og øvre høyre hjørne i (6,4) ved hjelp av Liang-barskys linjeklippingsalgoritme. (8%)

- c) Hvorfor er Liang-Barskys algoritme for linjeklipping kjappere enn Cohen Sutherland-algoritmen? (4%)

Oppgave 6: Transformasjoner og projeksjoner

- a) Utled rotasjonsmatrisen for å rotere et punkt P om x-aksen slik at det får en ny posisjon P'. **Hint:** $\sin(A+B) = \sin(A)\cos(B) + \cos(A)\sin(B)$, $\cos(A+B) = \cos(A)\cos(B) - \sin(A)\sin(B)$ (10%)
- b) Gitt et syntetisk kamera med projeksjonscenter i punktet (0,0,5,1) og bildeplanet definert for $z=0$. Projiser punktet $P=(10,7,-9,1)$ som er angitt i kamerakoordinater inn i bildeplanet. Hva blir kartesiske koordinatene til det projiserte punktet? (10%)

Task 4: Rasterizing

- a) Describe how to extend the DDA algorithm to rasterizing circles. (8%)
- b) Write the algorithm you have described in a) in pseudo code (8%)
- c) Execute the pseudo code for a circle with center in (0,0) and a radius of 10 using the pseudo code you have written in b). Show, for each pixel drawn, all the important variables in a table (4%)

Task 5: Line clipping

- a) Describe Liang-Barsky's line clipping algorithm. (8%)
- b) Clip the line with end points (0,0) and (5,4) against the clipping window with lower left corner at (2,0) and upper right corner at (6,4) using Liang-Barsky's line clipping algorithm. (8%)
- c) Why is Liang-Barsky's line clipping algorithm more computational efficient than Cohen-Sutherland's algorithm? (4%)

Task 6: Transformations and Projections

- a) Develop the 4x4 matrix that will rotate a point P about the x-axis. **Hint:** $\sin(A+B) = \sin(A)\cos(B) + \cos(A)\sin(B)$, $\cos(A+B) = \cos(A)\cos(B) - \sin(A)\sin(B)$ (10%)
- b) Given a synthetic camera with the center of projection at the point (0,0,5,1) and the projection plane $z=0$. Project the point $P=(10,7,-9,1)$ given in camera coordinates into the projection plane. What are the cartesian coordinates of the projected point? (10%)