

Grafikus primitívek kitöltése

Téglalap kitöltése

Poligon kitöltése

Kör, ellipszis kitöltése

Kitöltés mintával

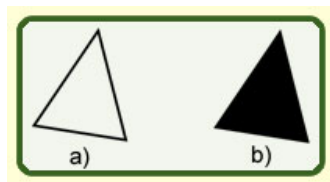
Grafikus primitívek kitöltése

Területi primitívek:

Zárt görbék által határolt területek (pl. kör, ellipszis, poligon)

Megjelenítendők

- a) Csak a határvonalat reprezentáló pontok kirajzolásával (kitöltetlen)
- b) Minden belső pont kirajzolásával (kitöltött)

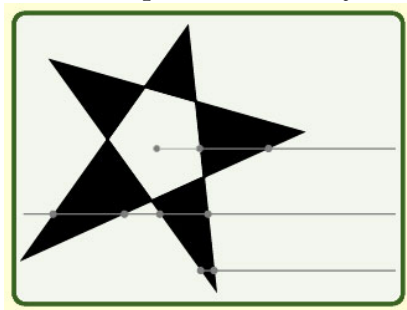


Grafikus primitívek kitöltése

Alapkérdés:

Mely képpontok tartoznak a grafikus primitívhez?

Páratlan paritás szabálya:



Páros számú metszéspon:
külső pont

Páratlan számú metszéspon:
belső pont

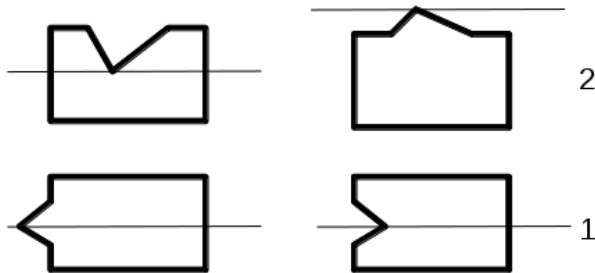
Grafikus primitívek kitöltése

Primitívek kitöltésének az elve:



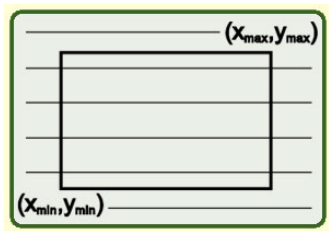
Balról jobbra haladva minden egyes pásztázó (scan) vonalon kirajzoljuk a primitív belső pontjait (egyszerre egy szakaszt kitöltve)

Csúcsponatok metszésekor:



Ha a metszett csúcsponatok lokális minimum vagy maximum, akkor kétszer számítjuk, különben csak egyszer.

Téglalap kitöltése



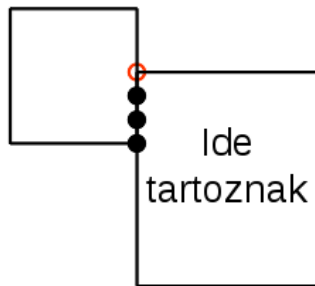
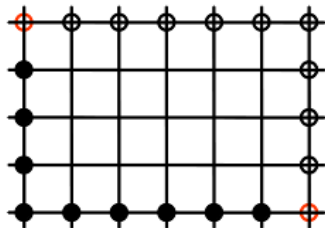
```
for ( $y = y_{min}; y < y_{max}, y++$ )  
  for ( $x = x_{min}; x < x_{max}, x++$ )  
    WritePixel( $x, y, value$ )
```

Probléma:

Egész koordinátájú határpontok hova tartozzanak?

Téglalap kitöltése

Legyen a szabály pl.: Egy képpont akkor nem tartozik a primitívhez, ha rajta áthaladó él, és a primitív által meghatározott félsík a képpont alatt, vagy attól balra van. Pl.:

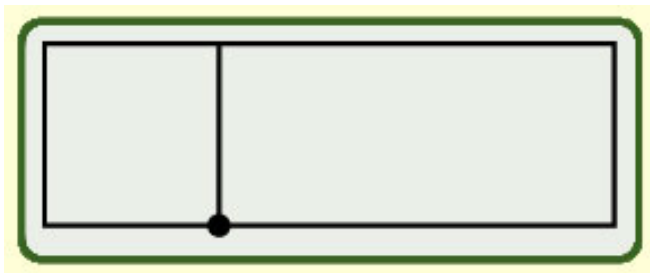


Vagyis a pásztázó vonalon a kitöltési szakasz balról zárt, jobbról nyitott

Téglalap kitöltése

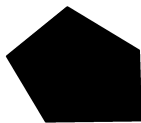
Megjegyzések:

- a) Általánosítható poligonokra
- b) A felső sor és a jobb szélső oszlop kimarad
- c) A bal alsó sarok kétszeresen tartozhat a téglalaphoz



Poligonok kitöltése

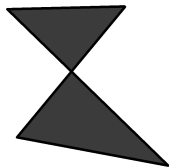
A poligon lehet:



konvex



konkáv



önmagát metsző

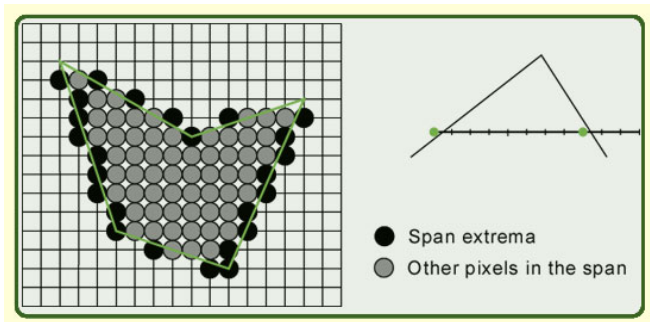


lyukas

Haladjunk a pásztázó egyeneseken és keressük a kitöltési szakaszok végpontjait.

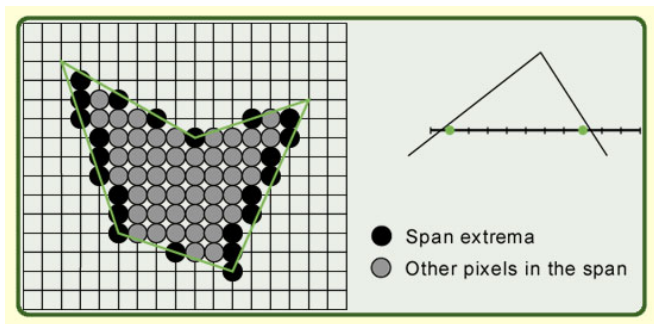
Poligon kitöltése

a) A felezőpont algoritmus szerint választjuk a végpontokat (azaz, nem számít, hogy azok a poligonon kívül, vagy belül vannak)



Diszjunkt poligonoknak lehet közös képpontjuk

b) A végpontokat a poligonhoz tartozó képpontok közül választjuk



Algoritmus poligonok kitöltésére

Minden pásztázó egyenesre:

1. A pásztázó egyenes és a poligon élei metszéspontjainak a meghatározása
2. A metszéspontok rendezése növekvő x -koordinátáik szerint
3. A poligon belsejébe tartozó szakasz(ok) végpontjai közötti képpontok kirajzolása Használjuk a páratlan paritás szabályát: Tegyük fel, hogy a bal szélén kívül vagyunk, utána minden egyes metszéspont megváltoztatja a paritást.

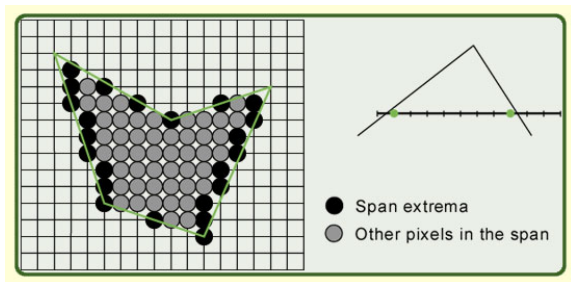


Algoritmus poligonok kitöltésére

3.1. Adott x nem egész értékű metszéspont.

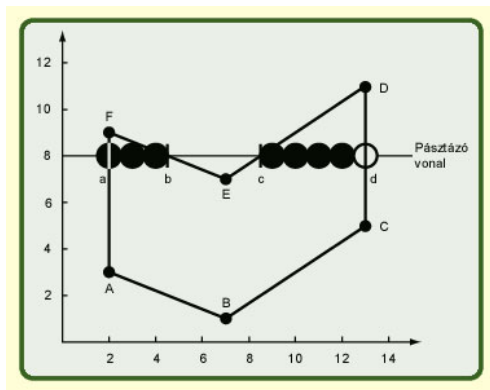
Ha kívül vagyunk, akkor legyen a végpont a fölfelé kerekített x

Ha belül vagyunk, akkor legyen a végpont a lefelé kerekített x



Algoritmus poligonok kitöltésére

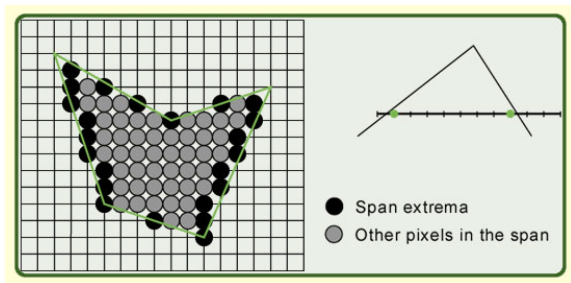
- 3.2. Adott x egész értékű metszéspont
Ha ez bal végpont, akkor ez belső pont.
Ha ez jobb végpont, akkor ez külső pont



Algoritmus poligonok kitöltésére

3.2.1. A poligon csúcspontjaiban:

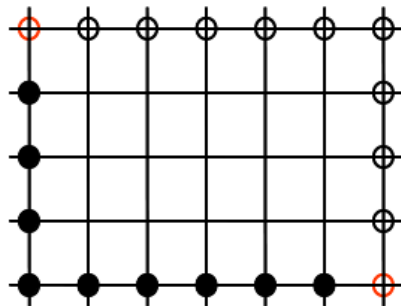
- ▶ y_{min} csúcspont beszámít a paritásba
- ▶ y_{max} csúcspont nem számít a paritásba, tehát y_{max} csúcspont csak akkor lesz kirajzolva, ha az a szomszédos él y_{min} pontja is



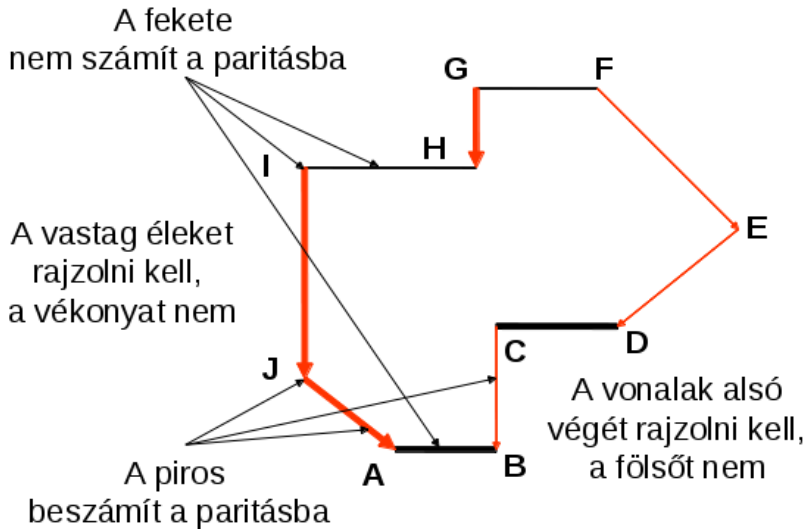
3.2.2. Vízszintes él esetén:

Az ilyen élek csúcspontjai nem számítanak a paritásba

Egész y koordináta esetén az alsó élet rajzoljuk, a felsőt nem

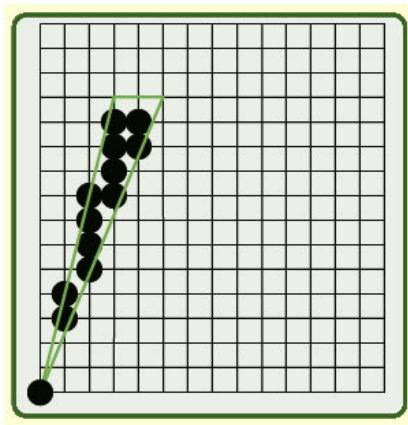


Példa poligon kitöltésére



Poligon kitöltése

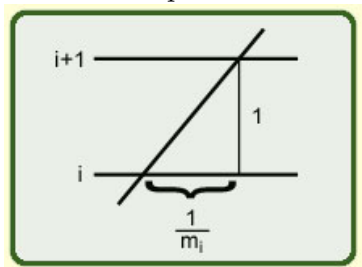
Szilánkok: olyan poligon-területek, amelyek belsejében nincs kitöltendő szakasz = hiányzó képpontok



Implementáció:

Nem kell minden egyes pásztázó vonalra újra kiszámolni minden metszéspontot, mert általában csak néhány metszéspont érdekes az i -edik pásztázó vonalról az $i + 1$ -dikre átlépve.

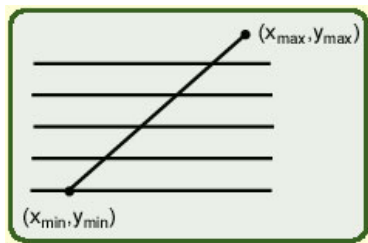
$$x_{i+1} = x_i + \frac{1}{m_i}$$



Poligon kitöltése

Tegyük fel hogy: $m > 1$

($m = 1$ triviális, $m < 1$ kicsit bonyolultabb)



$$\Delta x = \frac{1}{m} = \frac{x_{max} - x_{min}}{y_{max} - y_{min}} \quad (< 1)$$

$$x = \underset{\text{egész rész}}{[x]} + \underset{\text{tört rész}}{\{x\}}$$

$$\begin{aligned} [x_{i+1}] &= [x_i] \\ \{x_{i+1}\} &= \{x_i + \Delta x\} \end{aligned}$$

$$\begin{aligned} \text{vagy } [x_i] + 1 \\ \text{vagy } \{x_i\} + \Delta x - 1 \end{aligned}$$

Tegyük fel, hogy bal határon vagyunk!

- ▶ Ha $\{x_i\} = 0$, akkor (x, y) -t rajzolni kell (a vonalon van)
- ▶ Ha $\{x_i\} \neq 0$, akkor fölfelé kell kerekíteni x -et (belső pont)

Egész értékű aritmetika használható:

törtrész helyett a számlálót és nevezőt kell tárolni

Poligon kitöltése

```
void LeftEdgeScan(int xmin, int ymin,
                  int xmax, int ymax, int value) {
    int x, y, numerator, denominator, increment;
    x = xmin;
    numerator = xmax - xmin;
    denimonator = ymax - ymin;
    increment = denominator;
    for(y = ymin; y < ymax; y++) {
        WritePixel(x,y,value);
        increment += numerator;
        if(increment > denominator) {
            x++;
            increment -= denominator;
        } // END IF
    } // END FOR
}
```

Adatstruktúrák:

ÉT: (Élek táblázata)

A kisebbik y értékük szerint rendezve az összes élet tartalmazza.

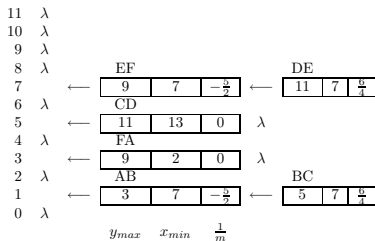
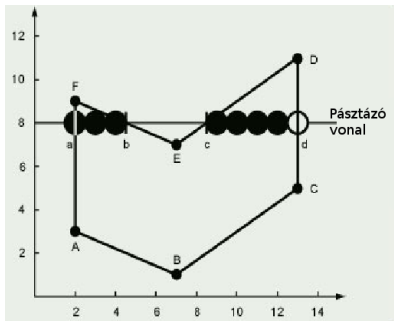
A vízszintes élek kimaradnak!

Annyi lista van, ahány pásztázó vonal. Minden listában azok az élek szerepelnek, amelyek alsó végpontja a pásztázó vonalon van.

A listák az élek alsó végpontjának x koordinátája, ezen belül a meredekség reciproka szerint rendezettek.

Minden lista elem tartalmazza az él y_{max} , x_{min} koordinátáját és a meredekség reciprokát.

Poligon kitöltése



AÉT: Aktív Élek Táblázata)

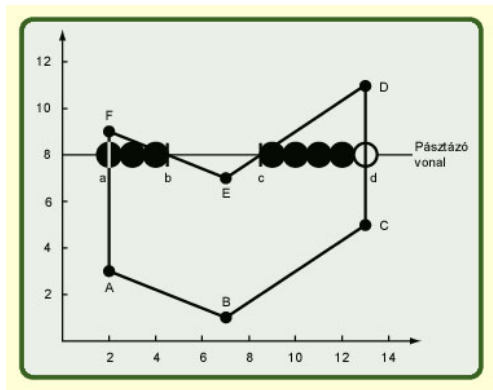
A pásztázó vonalat metsző éleket tartalmazza a metszéspontok x koordinátája szerint rendezve. Ezek a metszéspontok kitöltési szakaszokat határoznak meg az aktuális pásztázó vonalon.

Ez is lista.

Algoritmus poligon kitöltésére

0. **ÉT** kialakítása
 1. y legyen az **ÉT**-ben lévő nem üres listák közül a legkisebb y
 2. **AÉT** inicializálása (üres)
 3. A továbbiakat addig ismételjük, amíg **ÉT** végére érünk és **AÉT** üres lesz:
 - 3.1 **ÉT**-ből az y -hoz tartozó listát – a rendezést megtartva – **AÉT**-hez másoljuk
 - 3.2 **AÉT**-ből kivesszük azokat az éleket, amelyekre $y_{max} = y$ (a felső éleket nem töltjük ki)
 - 3.3 A kitöltési szakaszok pontjait megjelenítjük
 - 3.4 $y = y + 1$
 - 3.5 Minden **AÉT**-beli élben módosítjuk x -et
-

Poligon kitöltése



AÉT az $y = 8$ pásztázóvonalon:

$$\leftrightarrow \begin{array}{|c|c|c|} \hline 9 & 2 & 0 \\ \hline \end{array} \leftrightarrow \begin{array}{|c|c|c|} \hline 9 & 4 & -\frac{5}{2} \\ \hline \end{array} \leftrightarrow \begin{array}{|c|c|c|} \hline 11 & 9 & \frac{6}{4} \\ \hline \end{array} \leftrightarrow \begin{array}{|c|c|c|} \hline 11 & 13 & 0 \\ \hline \end{array}$$

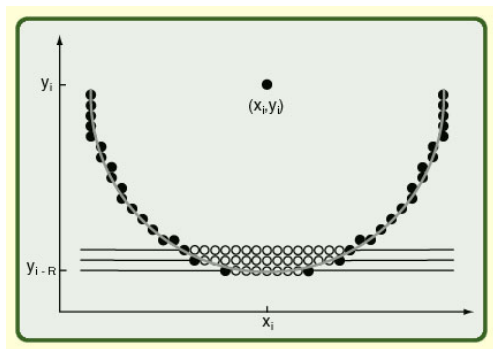
x_{max} x $\frac{1}{m}$

Megjegyzés:

Háromszögekre, trapézokra egyszerűsíthető az algoritmus, mert a pásztázó egyeneseknek legfeljebb 2 metszéspontja lehet egy háromszöggel vagy egy trapézzal (nem kell **ÉT**).

Kör, ellipszis kitöltése

P belül van, ha $F(P) < 0$, de most is használható a felezőpont módszer. Hasonló algoritmussal számíthatók a kitöltési szakaszok.



Háromszög kitöltése (OpenGL)

Egyetlen színnel:

```
glBegin(GL_TRIANGLES);  
    glColor3f(0.1, 0.2, 0.3);  
    glVertex3f(0, 0, 0);  
    glVertex3f(1, 0, 0);  
    glVertex3f(0, 1, 0);  
glEnd();
```

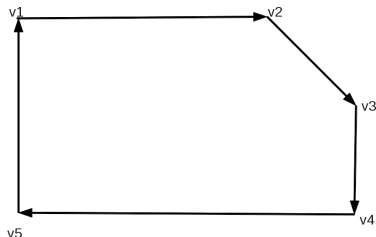
Háromszög kitöltése (OpenGL)

Több színnel (Gouraud-féle módon interpolálva):

```
glShadeModel(GL_SMOOTH); // G-arnyalas
glBegin(GL_TRIANGLES);
    glColor3d(1.0,0.0,0.0);
    glVertex3d(5.0, 5.0, 0);
    glColor3d(0.0,1.0,0.0);
    glVertex3d(195.0, 5.0, 0);
    glColor3d(0.0,0.0,1.0);
    glVertex3d(100.0, 195.0, 0);
glEnd();
```

Poligon létrehozása (OpenGL)

```
glBegin(GL_POLYGON);  
    glVertex3d(100,0,0);  
    glVertex3d(0,0,0);  
glEnd();
```



Az OpenGL csak síkbeli konvex sokszögek helyes kirajzolását garantálja.

Az elsőként specifikált csúcspont színe lesz a primitív színe, ha `glShadeModel(GL_FLAT);`

Poligon (OpenGL)

3D-s poligonoknak két oldaluk van: elülső és hátulsó oldal. Alapértelmezésben mindkét oldal ugyanúgy rajzolódik ki, de ezen lehet változtatni:

```
void glPolygonMode(enum face, enum mode);
```

face:

- ▶ GL_FRONT_AND_BACK
- ▶ GL_FRONT
- ▶ GL_BACK

mode:

- ▶ GL_POINT csak a csúcspontokat rajzolja ki
- ▶ GL_LINE a határvonalat rajzolja ki
- ▶ GL_FILL kitölti a poligont

