


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)
 Megjeleníthetők

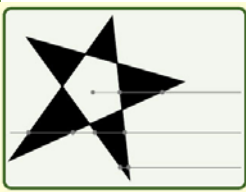
- Csak a határvonalat reprezentáló pontok kirajzolásával (kitöltetlen)
- Minden belső pont kirajzolásával (kitöltött)



GRAFIKUS PRIMITÍVEK KITÖLTÉSE

Alapkérdés:
 Mely képpontok tartoznak a grafikus primitívekhez?

Páratlan paritás szabálya:




Páros számú metszéspont:
külső pont

Páratlan számú metszéspont:
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)

GRAFIKUS PRIMITÍVEK KITÖLTÉSE

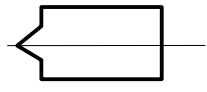
Csúcsponatok metszésekor:



2



2



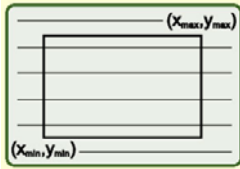
1



1

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

Téglalap kitöltése



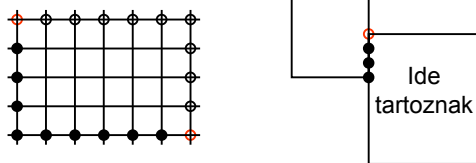
```

for y from y_min to y_max
  for x from x_min to x_max
    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.:

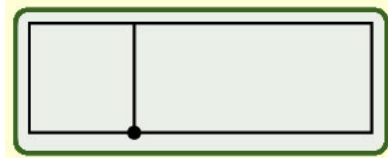


Vagyis a pásztázó vonalon a kitöltési szakasz balról zárt, jobbról nyitott

Téglalap kitöltése

Megjegyzések:

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



Poligon kitöltése

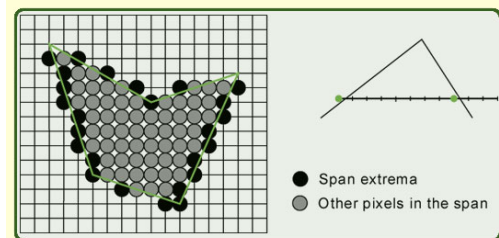
A poligon lehet:

- konvex,
- konkáv,
- önmagát metsző,
- lyukas

Haladjunk a pásztázó egyeneseken és keressük a kitöltési szakaszok végpontjait:

Poligon kitöltése

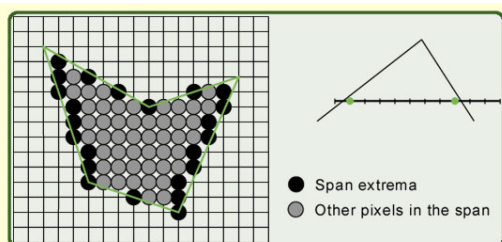
- a) A felezőpont algoritmus szerint választjuk a végpontokat (azaz, nem számít, hogy azok a poligonon kívül, vagy belül vannak);



Diszjunkt poligonoknak lehet közös képpontjuk

Poligon kitöltése

- b) A végpontokat a poligonhoz tartozó képpontok közül választjuk



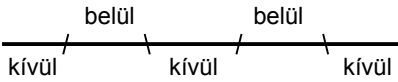
Algoritmus poligonok kitöltésére

Minden pásztázó egyenesre:

1. A pásztázó egyenes és a poligon élei metszéspontjainak a meghatározása
2. A metszéspontok rendezése növekvő x-koordinátáik szerint

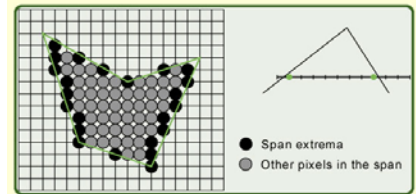
Algoritmus poligonok kitöltésére

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



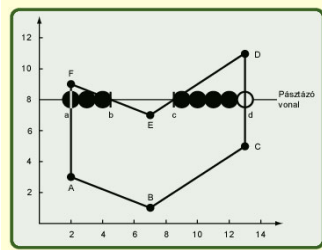
Algoritmus poligonok kitöltésére

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



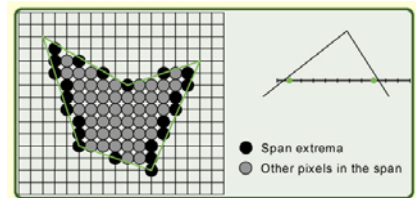
Algoritmus poligonok kitöltésére

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



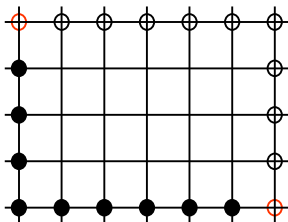
Algoritmus poligonok kitöltésére

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

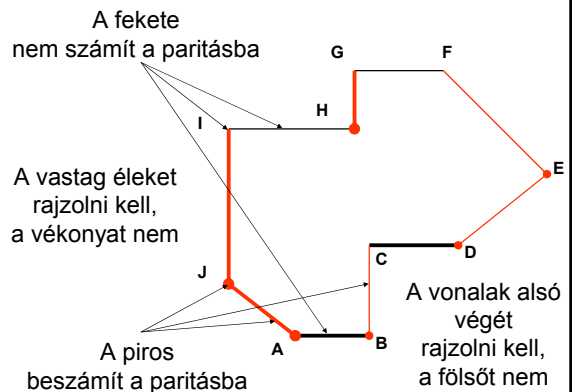


Algoritmus poligonok kitöltésére

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

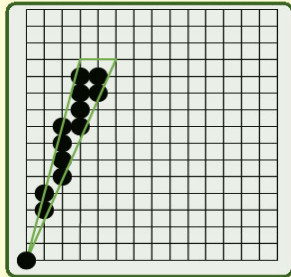


Példa poligon kitöltésére



Poligon kitöltése

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

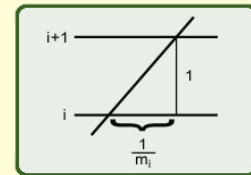


Poligon kitöltése

Implementáció:

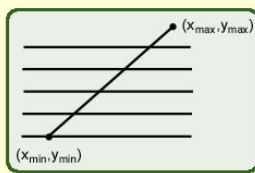
Nem kell minden egyes pásztázó vonalra újra kiszámolni minden metszéspontot, mert általában csak néhány metszéspont érdekes az i -dik pásztázó vonalról az $i+1$ -dikre átlépve

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



Poligon kitöltése

Tegyük fel hogy: $m > 1$
($m = 1$ triviális, $m < 1$ kicsit bonyolultabb)



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

$x =$ egész rész + tört rész
 $[x_i] \quad \{x_i\}$

$$[x_{i+1}] = [x_i] \quad \text{vagy} \quad [x_i] + 1$$

$$\{x_{i+1}\} = \{x_i\} + \Delta x \quad \text{vagy} \quad \{x_i\} + \Delta x - 1$$

Poligon kitöltése

Tegyük fel, hogy a 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:
tötrész helyett a számláló és nevező tárolása

Poligon kitöltése

```

procedure LeftEdgeScan (xmin, ymin, xmax, ymax, value
:integer);
var x, y, numerator, denominator, increment : integer;
begin
  x := xmin; numerator := xmax-xmin;
  denominator := ymax-ymin;
  increment := denominator;
  for y := ymin to ymax do
    begin WritePixel (x,y,value);
      increment := increment+numerator;
      if increment > denominator then
        begin
          x := x+1;
          increment := increment-denominator
        end;
    end
  end;
end;
```

Poligon kitöltése

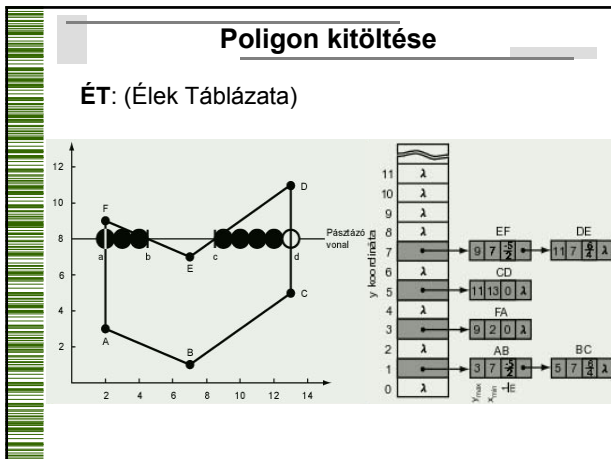
Adatstruktúrák:

ÉT: (Élek Táblázata)

A kisebbik y értékük szerint rendezve az összes éleket 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 szerint rendezettek

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



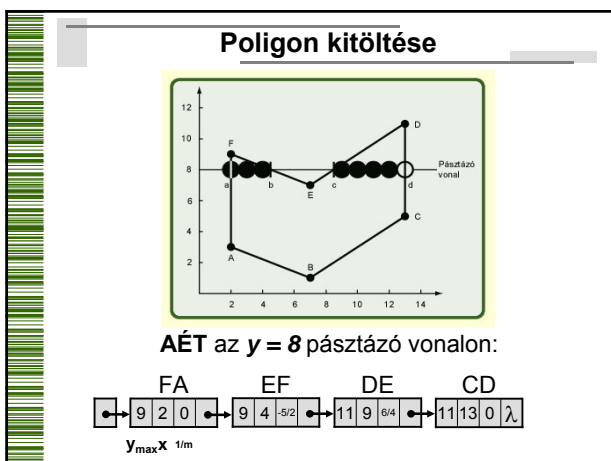
Poligon kitöltése

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 levő nem üres listák közül a legkisebb y
 2. **AÉT** inicializálása (üres)
 3. A továbbiakat addig ismételjük, amíg **ÉT** végére érünk és **AÉT** üres lesz:

- ### Algoritmus poligon kitöltésére
- 3.1 **ÉT**-ből az y -hoz tartozó listát **AÉT**-hez másoljuk
 - 3.2 **AÉT**-ből kivesszük azokat az éleket, amelyekre $y_{max} = y$ (a felső élet nem töltjük ki)
 - 3.3 **AÉT**-t x szerint rendezzük
 - 3.2 A kitöltési szakaszok pontjainak a megjelenítése
 - 3.4 $y = y + 1$
 - 3.5 Minden **AÉT**-beli élben módosítjuk x -et

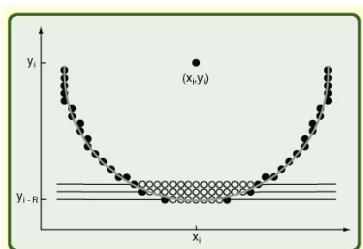


Poligon kitöltése

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)

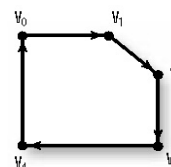
Több színnel (Gouraud-féle módon interpolálva)

```
glShadeModel(GL_SMOOTH); //G-árnyalás
glBegin(GL_TRIANGLES);
glColor3d(1.0, 0.0, 0.0);
glVertex3d(0.0, 0.0, 0.0);
glColor3d(0.0, 0.0, 1.0);
glVertex3d(1.0, 0.0, 0.0);
glColor3d(0.0, 1.0, 0.0);
glVertex3d(0.5, 1.0, 0.0);
glEnd();
```



Poligon létrehozása (OpenGL)

```
glBegin(GL_POLYGON);
glVertex3d(0, 100, 0);
glVertex3d(50, 100, 0);
glVertex3d(100, 50, 0);
glVertex3d(100, 0, 0);
glVertex3d(0, 0, 0);
glEnd();
```



Az **OpenGL** csak 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:

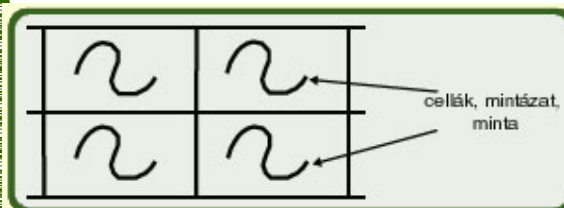
- GL_FRONT_AND_BACK
- GL_FRONT
- GL_BACK;

mode:

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

Kitöltés mintával

Általában: terület kitöltése szabályosan ismétlődő grafikus elemekkel



Képmátrixok (raszter) esetében a cella egy (kisméretű) mátrix

Kitöltés mintával

Példa:



Tégla minta

Lehet a kitöltés "átlátszó" is: nem minden képpontot írunk felül, csak azokat, ahol a minta nem 0

Kitöltés mintával

Fajtái:

- Válasszunk egy pontot a **primitívben** (pl. bal felsőt), egy pontot a mintában (pl. bal felsőt), illesszük azokat egymásra, a többi pont illeszkedése már kiszámítható
- Válasszunk egy pontot a **képernyőn** (pl. bal felsőt), egy pontot a mintában (pl. bal felsőt), illesszük azokat egymásra, a többi pont illeszkedése már kiszámítható (most a mintázat a képernyőhöz van rögzítve)

Kitöltés mintával

Legyen:
 minta $M * N$ -es mátrix
 minta $[0,0] \gggg$ képernyő $[0,0]$
 ekkor

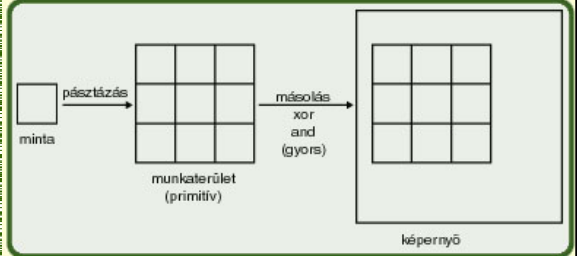
1. módszer: Pásztázás soronként (átlátszó)

```
if minta [x mod M, Y mod N] then
    WritePixel (x, y, érték)
```

Gyorsabb: több képpont (sor) egyszerre történő másolásával (esetleg maszkolás is szükséges a sor elején vagy végén)

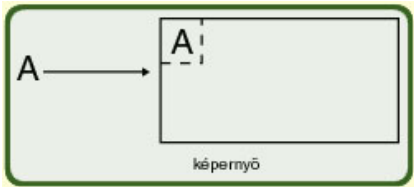
Kitöltés mintával

2. módszer: Téglalap írás



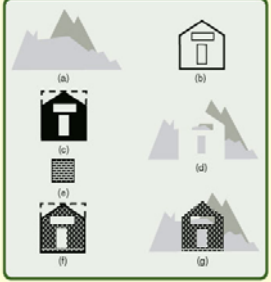
Kitöltés mintával

Csak akkor érdemes használni, ha a primitívet sokszor kell használni
 Pl. karakterek megjelenítése



Kitöltés mintával

A téglalap írás kitöltés kombinálható képek közötti műveletekkel, így bonyolult ábrák készíthetők:

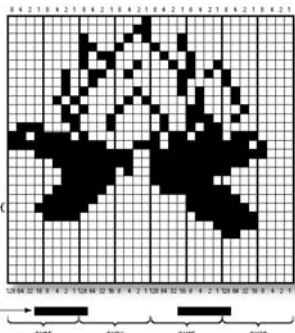


(a) hegyek, (b) ház vonalai, (c) a ház kitöltött bitmap képe, (d) (a)-ból kitöröltük (c)-t, (e) tégla minta, (f) (b) tégla mintával kitöltve, (g) (e) (d)-re másolva

Kitöltés mintával (OpenGL)

```
void glPolygonStipple(const ubyte *mask);
```

Kitöltési minta beállítása
mask: egy 32×32-es
bittérkép (minta)



Kitöltés mintával (OpenGL)

A kitöltési minta

glEnable(GL_POLYGON_STIPPLE) engedélyezése
glDisable(GL_POLYGON_STIPPLE) tiltása

Pl.:

```
void display() {  
    glClear(GL_COLOR_BUFFER_BIT);  
    //Képernyő törlés  
    glShadeModel(GL_FLAT);  
    //Árnyalási mód: FLAT  
    glPolygonStipple(mask); // A minta  
    glEnable(GL_POLYGON_STIPPLE); //engedélyezés  
    rajz(); //Az alakzat kirajzolása  
    glDisable(GL_POLYGON_STIPPLE); //tiltás  
}
```

Kitöltés mintával (OpenGL)

Példa:

