Grafikus primitívek kitöltése

Téglalap kitöltése

Poligon kitöltése

Kör, ellipszis kitöltése

Kitöltés mintával

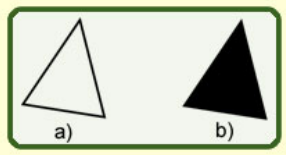
Grafikus primitívek kitöltése

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)

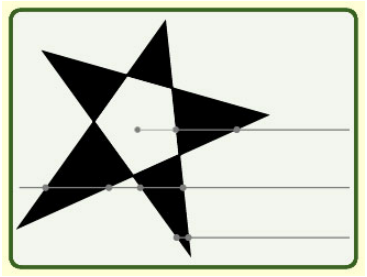


A tertületi primitívek zárt görbével határolt területek, amelyeket megjelníthetünk körvonalaikkal, vagy pedig kitölthetünk valamilyen színnel.

Grafikus primitívek kitöltése

Alapkérdés:
Mely képpontok tartoznak a grafikus primitívhez?

Páratlan paritás szabálya:



Páros számú metszéspont:
külső pont

Páratlan számú metszéspont:
belső pont

Poligonok esetében problémába ütközhetünk a kitöltést illetően, hiszen létezik olyan poligon, amely elmetszi önmagát. Kérdés, hogy ilyenkor hogyan járjunk el.
Szkenneljük sorfolytonosan képet, és figyeljük meg, hogy mikor haladunk át egy poligon élén. Ha egy sorban haladunk, számoljuk ezeket az áthaladásokat: minden páratlan áthaladás után a poligon belsejében vagyunk, és minden páros áthaladás után a poligonon kívül vagyunk.

Grafikus primitivek kitöltése

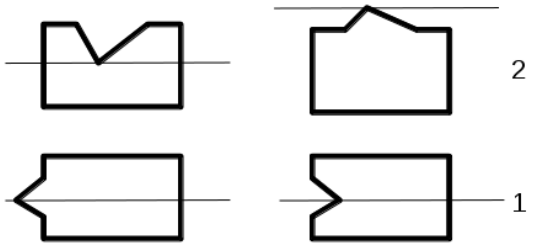
Primitívek kitöltésének az elve:

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

Ezzel a módszerrel szakaszonként ki tudjuk tölteni a poligont, ráadásul az eljárás során csak egyszer kell végigszkennelni a képet.

Grafikus primitívek kitöltése

Csúcspontok metszésekor:

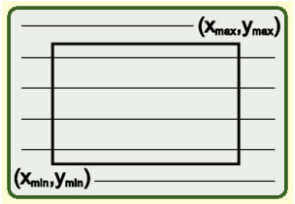


Ha a metszett csúcspont lokális minimum vagy maximum, akkor kétszer számítjuk, különben csak egyszer.

Oldalak metszésekor az előző módszer viszonylag egyszerű, de mi történik, ha csúcson haladunk át.
Ilyen esetben az a szabály, hogy

- Ha az elmetszett csúcspont lokális minimum (mint a bal felső képen), akkor kétszeresen számítjuk (vagyis egy pillanatra kikerülünk a poligonból, majd visszatérünk belülre).
- Ha lokális maximum, akkor is kétszer számítjuk, hiszen mielőtt elértük volna ezt a csúcspontot, akkor kívül voltunk, egy pillanatra bekerülünk a poligonba, majd ismét kívülre kerülünk.
- Különben pedig egyszer számítjuk, hiszen vagy belépünk a poligon belsejébe, vagy kilépünk onnan.

Téglalap kitöltése



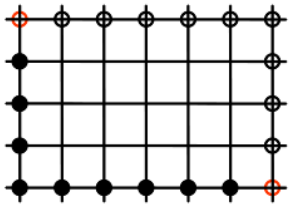

```
for (y = y_min; y < y_max, y++)
  for (x = x_min; x < x_max, x++)
    WritePixel(x,y,value)
```

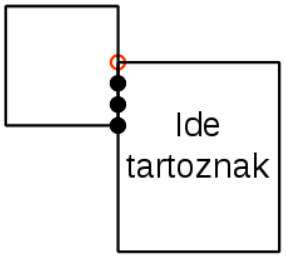
Probléma:
Egész koordinátájú határpontok hova tartozzanak?

A legegyszerűbb eset a téglalap kitöltése, hiszen ez vízszintes szakaszok rajzolásával megtehető. Ahogy a példában is látjuk, az implementációja során csupán egy dupla `for`-ciklussal kell végigmenni a sarokpontok koordinátái között.

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

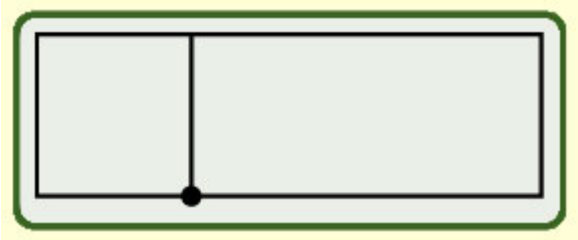
Téglalap kitöltése

Megjegyzések:

a) Általánosítható poligonokra

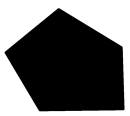
b) A felső sor és a jobb szélső oszlop kimarad

c) A bal alsó sarok kétszeresen tartozhat a téglalaphoz



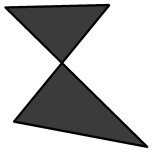
Poligonok kitöltése

A poligon lehet:

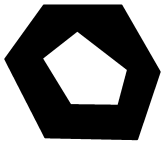


konvex

konkáv



önmagát metsző

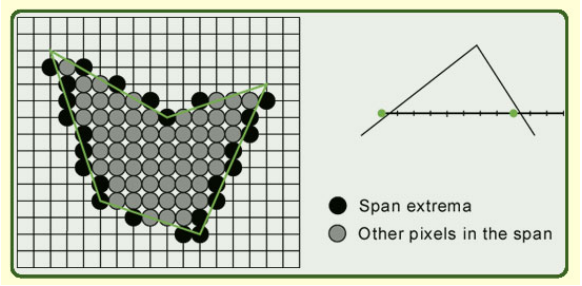


lyukas

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)

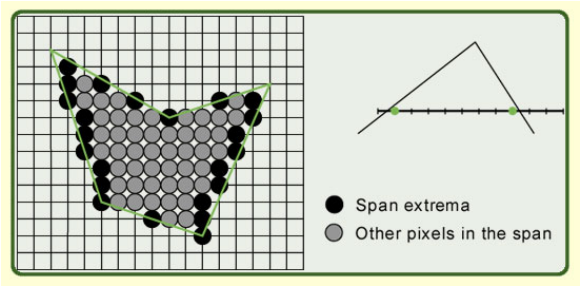


Diszjunkt poligonoknak lehet közös képpontjuk

Ha általános poligonokat tekintünk, akkor ezek a poligonok lehetnek konvexek, konkávok, önmagukat metsző, illetve lyukas poligonok. A pásztázó módszer minden típusú poligon esetében működik.

Poligon kitöltése

b) A végpontokat a poligonhoz tartozó képpontok közül választjuk



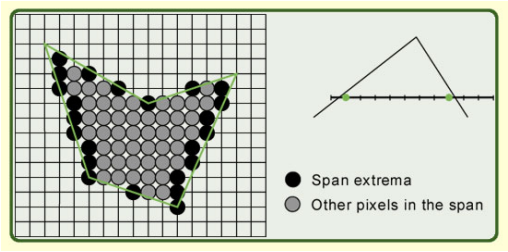
Algoritmus poligonok kitöltésére

Minden pásztázó egyenesre:

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

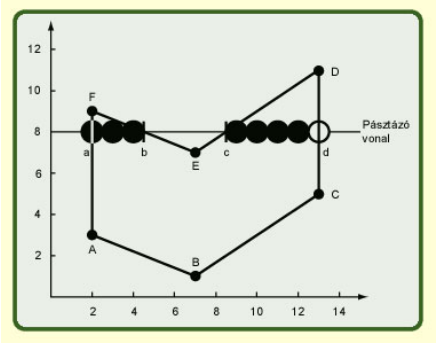
Algoritmus poligonok kitöltésére

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

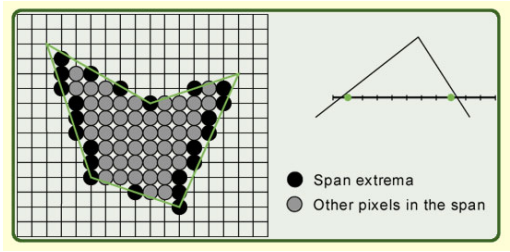


A kitöltés során a bal oldali élek mentén a pontok a poligonhoz tartoznak, a jobb oldali végpontok pedig nem tartoznak a poligonhoz.
Ez az elv összhangban van azzal, hogy a poligonok felső valamint a jobb oldali élek nem tartoznak a poligonhoz, a bal oldali élek pedig a poligonhoz tartoznak.

Algoritmus poligonok kitöltésére

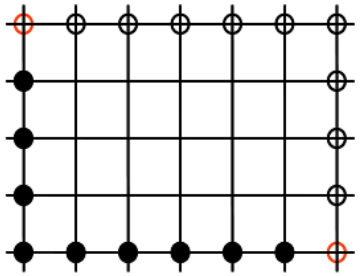
3.2.1. A poligon csúcspontjaiban:

- 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

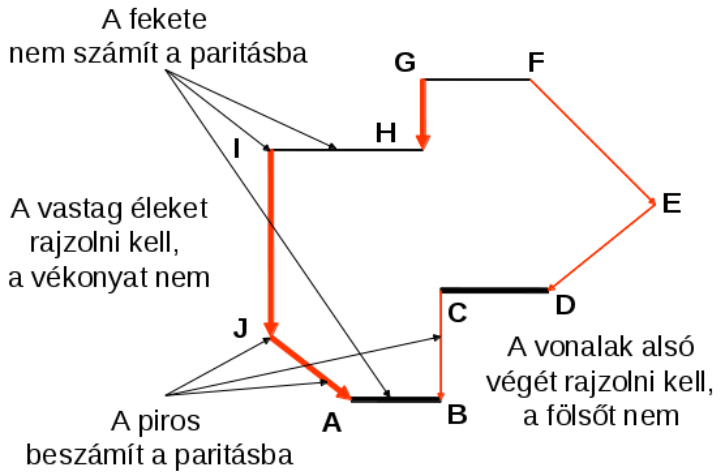


Algoritmus poligonok kitöltésére

3.2.2. Vízszintes él esetén:
Az ilyen élek csúcspontjai nem számítanak a paritásba
Egész y koordináta esetén az alsó élet rajzoljunk, a felsőt nem

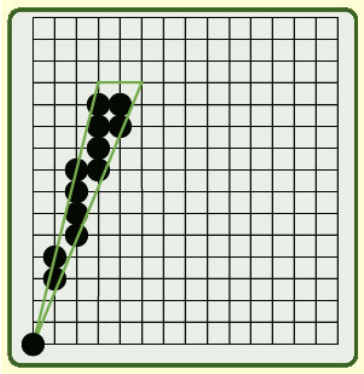


Példa poligon kitöltésére



Poligon kitöltése

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

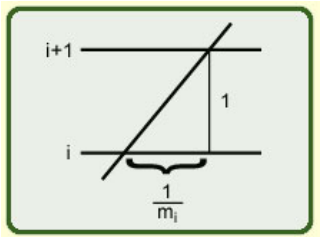


A szilánkok nagyon elfajuló alakú poligonok. Mivel vannak hosszú keskeny csúcsai, ezért előfordulhat, hogy bizonyos részeken nem lesz olyan rácspont, amely a poligon határán belül van. Ugyanakkor más pontok pedig benne lehetnek a poligonban, ezeket ki kell rajzolni. Ez azt eredményezi, hogy látszólag a diszkrét rácson „megszakad” az objektum.

Poligon kitöltése

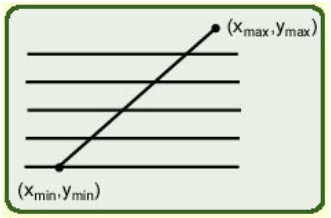
Implementáció:
Nem kell minden egyes pásztázó vonalra újra kiszámolni minden metszéspontot, mert általában csak néhány metszéspont érdekes az i-edik pásztázó vonalról az $i+1$-dikre átlépve.

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



Poligon kitöltése

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



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

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

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

A poligon oldalainak egyenleteit a végpontok alapján ismerjük. Nem kell minden pásztázó egyenesre kiszámolni a metszéspontokat, hiszen tudjuk azt, hogy a pásztázó egyenesek pontosan 1 y-koordinátában térnek el egymástól és tudjuk az egyenes meredekségét is, így az x_{i+1} koordinátát könnyedén kiszámolhatjuk.

Poligon kitöltése

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

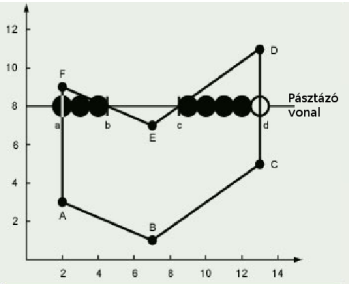
Poligon kitöltése

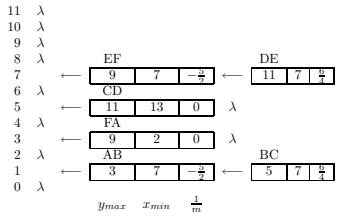
```
void LeftEdgeScan(int xmin, int ymin,
         int xmax, int ymax, int value) {
  int x, y, numerator, denominator, increment;
  x = xmin;
  numerator = xmax - xmin;
  denimonator = ymax - ymin;
  increment = denominator;
  for(y = ymin; y < ymax; y++) {
    WritePixel(x,y,value);
    increment +=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

Poligon kitöltése

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





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)

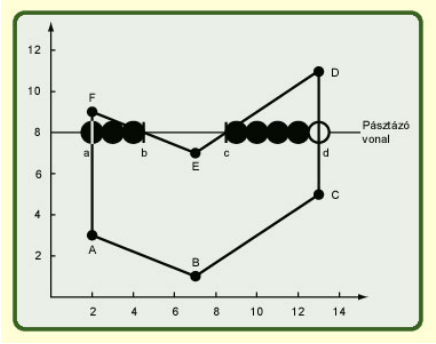
3.3 A kitöltési szakaszok pontjait megjelenítjük

3.4 $y = y + 1$

3.5 Minden **AÉT**-beli élben módosítjuk x-et

Az 1. lépésben megvizsgáljuk az Élek Táblázatát (ÉT), és csak az első pásztázó egyenes azon az y-koordinátán lesz, amelyik a legkisebb az ÉT-ben.
A 2. lépésben az Aktív Élek Táblázata még üres.
A 3. lépésben az ÉT-ből sorra vesszük az aktuális pásztázó vonalat metsző éleket, és bemásoljuk őket az AÉT-be. Ezután az AÉT-ből sorra megvizsgáljuk az éleket, és amennyiben az y_{max} érték megfelel az aktuális pásztázó egyenes y koordinátájának, kivesszük az AÉT-ből (hiszen a következő pásztázásnál, amikor y-t növeljük, már nem felette halad a pásztázó egyenes.
Figyeljük meg, hogy az x értékeket a 3.5 lépésben mindig módosítjuk, így könnyedén kitölthetjük a pontokat. Az $\frac{1}{m}$ értéket sem kell mindig újraszámolni, hiszen eltároltuk minden egyes szakaszhoz.

Poligon kitöltése



AÉT az $y = 8$ pásztázóvonalon:

$\leftrightarrow$ | 9 | 2 | 0 | $\leftrightarrow$ | 9 | 4 | $-\frac{5}{2}$ | $\leftrightarrow$ | 11 | 9 | $\frac{6}{4}$ | $\leftrightarrow$ | 11 | 13 | 0 | λ

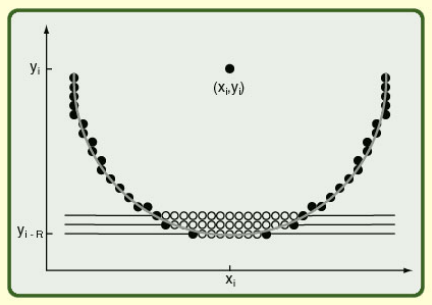
x_{max} x $\frac{1}{m}$

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();
```

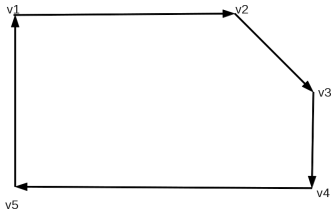
Háromszög kitöltése (OpenGL)

Több színnel (Gouraud-féle módon interpolálva):

```
glShadeModel(GL_SMOOTH); // G-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

Számítógépes grafika alapjai