Bevezetés a számítógépi grafikába

Kitöltő algoritmusok

Troll Ede Mátyás

Matematikai és Informatikai Intézet
Eszterházy Károly Katolikus Egyetem

Eger, 2024

Áttekintés

1. Színinformáción alapuló eljárások
   - Él-flag módszer
   - Többirányú rekurzív módszer

2. Csúcsaival adott poligon kitöltése
   - Téglalap kitöltése
   - Poligon kitöltése
     - Polárkoordináták
     - Konvex burok meghatározása
     - Poligon kitöltése (algoritmus)

Kitöltendő alakzatok

A kitöltendő alakzatok megadása az alábbiak alapján lehetséges

Kitöltendő alakzatok

A kitöltendő alakzatok megadása az alábbiak alapján lehetséges

- Az alakzatot határoló „görbét” ismerjük (színinformáción alapuló eljárások)

Kitöltendő alakzatok

A kitöltendő alakzatok megadása az alábbiak alapján lehetséges

- Az alakzatot határoló ,,görbét'' ismerjük (színinformáción alapuló eljárások)
  - Egymással határos raszterek

Kitöltendő alakzatok

A kitöltendő alakzatok megadása az alábbiak alapján lehetséges

- Az alakzatot határoló ,,görbét'' ismerjük (színinformáción alapuló eljárások)
    - Egymással határos raszterek
    - A háttérszíntől különböző raszterek

Kitöltendő alakzatok

A kitöltendő alakzatok megadása az alábbiak alapján lehetséges

- Az alakzatot határoló ,,görbét'' ismerjük (színinformáción alapuló eljárások)
  - Egymással határos raszterek
  - A háttérszíntől különböző raszterek
- Poligon adott csúcsokkal

Áttekintés

Él-flag módszer

Él-flag módszer

- Legyen adott a határszínnel definiált zárt alakzat

Él-flag módszer

- Legyen adott a határszínnel definiált zárt alakzat
- Elindítunk egy vízszintes scanline-t

Él-flag módszer

- Legyen adott a határszínnel definiált zárt alakzat
- Elindítunk egy vízszintes scanline-t
  - Ezzel együtt egy logikai változót (flag) inicializálunk hamis értékkel

Él-flag módszer

- Legyen adott a határszínnel definiált zárt alakzat
- Elindítunk egy vízszintes scanline-t
  - Ezzel együtt egy logikai változót (flag) inicializálunk hamis értékkel
  - Ezt minden határszínnél negáljuk

Él-flag módszer

- Legyen adott a határszínnel definiált zárt alakzat
- Elindítunk egy vízszintes scanline-t
    - Ezzel együtt egy logikai változót (flag) inicializálunk hamis értékkel
    - Ezt minden határszínnél negáljuk
        - Igaz érték: az alakzaton belül vagyunk

Él-flag módszer

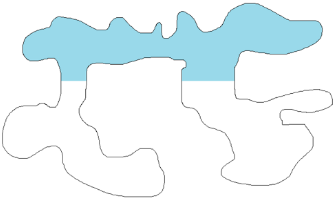
- Legyen adott a határszínnel definiált zárt alakzat
- Elindítunk egy vízszintes scanline-t
  - Ezzel együtt egy logikai változót (flag) inicializálunk hamis értékkel
  - Ezt minden határszínnél negáljuk
    - Igaz érték: az alakzaton belül vagyunk
    - Hamis érték: az alakzaton kívül vagyunk

Él-flag módszer

- Legyen adott a határszínnel definiált zárt alakzat
- Elindítunk egy vízszintes scanline-t
  - Ezzel együtt egy logikai változót (flag) inicializálunk hamis értékkel
  - Ezt minden határszínnél negáljuk
    - Igaz érték: az alakzaton belül vagyunk
    - Hamis érték: az alakzaton kívül vagyunk
  - Felmerülő probléma: egymás melletti határpontok kezelése

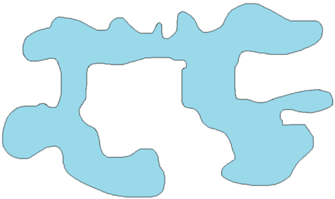
Él-flag módszer

```
ELJÁRÁS KITÖLT_ÉL_FLAG(SZIN: HÁTTÉR, SZÍN SZ);
  VÁLTOZÓK
    EGÉSZ: X, Y;
    LOGIKAI: BENN;
  ALGORITMUS
    CIKLUS Y <- Y_MIN..Y_MAX
      BENN <- HAMIS;
      CIKLUS X <- X_MIN..X_MAX
        HA (SZÍN(X, Y) <> HÁTTÉR) AKKOR
          BENN <- NEM BENN;
        HA_VÉGE;
        HA (BENN = IGAZ) AKKOR
          PIXEL(X, Y, SZ);
        HA_VÉGE;
      CIKLUS_VÉGE;
    CIKLUS_VÉGE;
ELJÁRÁS_VÉGE;
```



Él-flag módszer

```
ELJÁRÁS KITÖLT_ÉL_FLAG(SZIN: HÁTTÉR, SZÍN SZ);
  VÁLTOZÓK
    EGÉSZ: X, Y;
    LOGIKAI: BENN;
  ALGORITMUS
    CIKLUS Y <- Y_MIN..Y_MAX
      BENN <- HAMIS;
      CIKLUS X <- X_MIN..X_MAX
        HA (SZÍN(X, Y) <> HÁTTÉR) AKKOR
          BENN <- NEM BENN;
        HA_VÉGE;
        HA (BENN = IGAZ) AKKOR
          PIXEL(X, Y, SZ);
        HA_VÉGE;
      CIKLUS_VÉGE;
    CIKLUS_VÉGE;
ELJÁRÁS_VÉGE;
```



Él-flag módszer

```
ELJÁRÁS KITÖLT_ÉL_FLAG(SZIN: HÁTTÉR, SZÍN SZ);
  VÁLTOZÓK
    EGÉSZ: X, Y;
    LOGIKAI: BENN;
  ALGORITMUS
    CIKLUS Y <- Y_MIN..Y_MAX
      BENN <- HAMIS;
      CIKLUS X <- X_MIN..X_MAX
        HA (SZÍN(X, Y) <> HÁTTÉR) AKKOR
          BENN <- NEM BENN;
        HA_VÉGE;
        HA (BENN = IGAZ) AKKOR
          PIXEL(X, Y, SZ);
        HA_VÉGE;
      CIKLUS_VÉGE;
    CIKLUS_VÉGE;
ELJÁRÁS_VÉGE;
```

4 irányú rekurzív módszer

4 irányú rekurzív módszer

- Legyen adott a határszínnel definiált zárt alakzat

4 irányú rekurzív módszer

- Legyen adott a határszínnel definiált zárt alakzat
- Legyen adott az alakzat egy belső pontja (ahová klikkelünk)

4 irányú rekurzív módszer

- Legyen adott a határszínnel definiált zárt alakzat
- Legyen adott az alakzat egy belső pontja (ahová klikkelünk)
- Az algoritmus megvizsgálja, hogy az adott pixel nincs-e már megfestve

4 irányú rekurzív módszer

- Legyen adott a határszínnel definiált zárt alakzat
- Legyen adott az alakzat egy belső pontja (ahová klikkelünk)
- Az algoritmus megvizsgálja, hogy az adott pixel nincs-e már megfestve
    - Ha nincs, akkor megfesti

4 irányú rekurzív módszer

- Legyen adott a határszínnel definiált zárt alakzat
- Legyen adott az alakzat egy belső pontja (ahová klikkelünk)
- Az algoritmus megvizsgálja, hogy az adott pixel nincs-e már megfestve
  - Ha nincs, akkor megfesti
  - Rekurzív módon megvizsgáljuk a szomszédos 4 pixelt

4 irányú rekurzív módszer

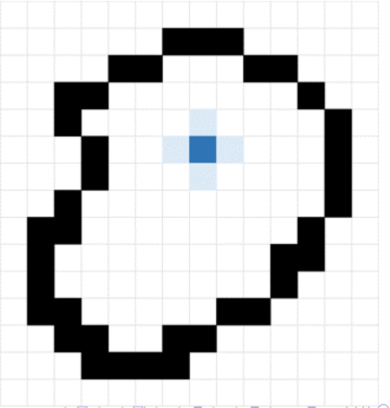
- Legyen adott a határszínnel definiált zárt alakzat
- Legyen adott az alakzat egy belső pontja (ahová klikkelünk)
- Az algoritmus megvizsgálja, hogy az adott pixel nincs-e már megfestve
    - Ha nincs, akkor megfesti
    - Rekurzív módon megvizsgáljuk a szomszédos 4 pixelt
- Az eljárás hátránya a rekurzív hívások száma

4 irányú rekurzív módszer

- Legyen adott a határszínnel definiált zárt alakzat
- Legyen adott az alakzat egy belső pontja (ahová klikkelünk)
- Az algoritmus megvizsgálja, hogy az adott pixel nincs-e már megfestve
  - Ha nincs, akkor megfesti
  - Rekurzív módon megvizsgáljuk a szomszédos 4 pixelt
- Az eljárás hátránya a rekurzív hívások száma

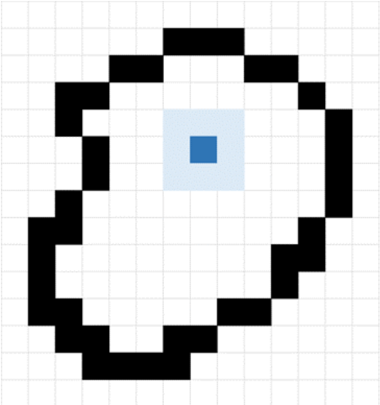
4 irányú rekurzív módszer

```
ELJÁRÁS KITÖLT_REK4(SZÍN: HÁTTÉR, SZÍN SZ, EGÉSZ X, EGÉSZ Y);
  ALGORITMUS
    HA (SZÍN(X, Y) = HÁTTÉR) AKKOR
      PIXEL(X, Y, SZ);
      KITÖLT_REK(X    , Y + 1, SZ);
      KITÖLT_REK(X    , Y - 1, SZ);
      KITÖLT_REK(X + 1, Y    , SZ);
      KITÖLT_REK(X - 1, Y    , SZ);
    HA_VÉGE;
ELJÁRÁS_VÉGE;
```



8 irányú rekurzív módszer

```
ELJÁRÁS KITÖLT_REK8(SZÍN: HÁTTÉR, SZÍN SZ, EGÉSZ X, EGÉSZ Y);
  ALGORITMUS
    HA (SZÍN(X, Y) = HÁTTÉR) AKKOR
      PIXEL(X, Y, SZ);
      KITÖLT_REK(X    , Y + 1, SZ);
      KITÖLT_REK(X    , Y - 1, SZ);
      KITÖLT_REK(X + 1, Y    , SZ);
      KITÖLT_REK(X - 1, Y    , SZ);
      KITÖLT_REK(X + 1, Y + 1, SZ);
      KITÖLT_REK(X + 1, Y - 1, SZ);
      KITÖLT_REK(X - 1, Y + 1, SZ);
      KITÖLT_REK(X - 1, Y - 1, SZ);
    HA_VÉGE;
ELJÁRÁS_VÉGE;
```



4 irányú kitöltés veremmel

Az eljárás ugyanazt fogja csinálni, mint a rekurzív megoldás, csak új rekurzív hívás helyett magunk kezeljük a vermet.

4 irányú kitöltés veremmel

Az eljárás ugyanazt fogja csinálni, mint a rekurzív megoldás, csak új rekurzív hívás helyett magunk kezeljük a vermet.
Ennek segítségével egyszerű iterációval meg tudjuk valósítani az alakzat kitöltését.

4 irányú kitöltés veremmel

```
ELJÁRÁS KITÖLT_VEREM4(SZÍN: HÁTTÉR, SZÍN SZ, EGÉSZ X, EGÉSZ Y);
  VÁLTOZÓK
    EGÉSZ[]: DX, DY
    EGÉSZ: I, NX, NY;
    VEREM[EGÉSZ, EGÉSZ]: V;
  ALGORITMUS
    DX <- [0, 1, 0, -1];
    DY <- [-1, 0, 1, 0];
    V.PUSH(X, Y);
    CIKLUS_AMÍG(V.POP(X, Y))
      PIXEL(X, Y, SZ);
      CIKLUS I <- 1..4
        NX <- X + DX[I];
        NY <- Y + DY[I];
        HA (SZÍN(NX, NY) = HÁTTÉR) AKKOR
          V.PUSH(NX, NY);
        HA_VÉGE;
      CIKLUS_VÉGE;
    CIKLUS_VÉGE;
ELJÁRÁS_VÉGE;
```

8 irányú kitöltés veremmel

```
ELJÁRÁS KITÖLT_VEREM8(SZÍN: HÁTTÉR, SZÍN SZ, EGÉSZ X, EGÉSZ Y);
  VÁLTOZÓK
    EGÉSZ[]: DX, DY
    EGÉSZ: I, NX, NY;
    VEREM[EGÉSZ, EGÉSZ]: V;
  ALGORITMUS
    DX <- [0, 1, 1, 1, 0, -1, -1, -1];
    DY <- [-1, -1, 0, 1, 1, 1, 0, -1];
    V.PUSH(X, Y);
    CIKLUS_AMÍG(V.POP(X, Y))
      PIXEL(X, Y, SZ);
      CIKLUS I <- 1..8
        NX <- X + DX[I];
        NY <- Y + DY[I];
        HA (SZÍN(NX, NY) = HÁTTÉR) AKKOR
          V.PUSH(NX, NY);
        HA_VÉGE;
      CIKLUS_VÉGE;
    CIKLUS_VÉGE;
ELJÁRÁS_VÉGE;
```

Áttekintés

Téglalap kitöltése

Téglalap kitöltése

- Legyen adott a koordinátatengelyekkel párhuzamos oldalú téglalap

Téglalap kitöltése

- Legyen adott a koordinátatengelyekkel párhuzamos oldalú téglalap
- Ebben az esetben egy egyszerű egymásba ágyazott ciklussal kitölthetjük a téglalapot

Téglalap kitöltése

- Legyen adott a koordinátatengelyekkel párhuzamos oldalú téglalap
- Ebben az esetben egy egyszerű egymásba ágyazott ciklussal kitölthetjük a téglalapot
- Probléma lehet az egymással közös élben találkozó téglalapok megjelenítése, ekkor ugyanis fellép a „szőrösödési” probléma

Téglalap kitöltése

- Legyen adott a koordinátatengelyekkel párhuzamos oldalú téglalap
- Ebben az esetben egy egyszerű egymásba ágyazott ciklussal kitölthetjük a téglalapot
- Probléma lehet az egymással közös élben találkozó téglalapok megjelenítése, ekkor ugyanis fellép a ,,szőrösödési'' probléma
  - A megjelenítés sorrendjétől függ az éldarab színe

Téglalap kitöltése

- Legyen adott a koordinátatengelyekkel párhuzamos oldalú téglalap
- Ebben az esetben egy egyszerű egymásba ágyazott ciklussal kitölthetjük a téglalapot
- Probléma lehet az egymással közös élben találkozó téglalapok megjelenítése, ekkor ugyanis fellép a „szőrösödési” probléma
  - A megjelenítés sorrendjétől függ az éldarab színe
- A probléma megoldása az, hogy a belső pixel definícióját az élekre vonatkozóan az alábbiak szerint adjuk meg

Téglalap kitöltése

- Legyen adott a koordinátatengelyekkel párhuzamos oldalú téglalap
- Ebben az esetben egy egyszerű egymásba ágyazott ciklussal kitölthetjük a téglalapot
- Probléma lehet az egymással közös élben találkozó téglalapok megjelenítése, ekkor ugyanis fellép a „szőrösödési” probléma
  - A megjelenítés sorrendjétől függ az éldarab színe
- A probléma megoldása az, hogy a belső pixel definícióját az élekre vonatkozóan az alábbiak szerint adjuk meg
  - Északi él pontjai: belső pontok (megjelenítjük)
  - Keleti él pontjai: külső pontok (nem jelenítjük meg)
  - Déli él pontjai: külső pontok (nem jelenítjük meg)
  - Nyugati él pontjai: belső pontok (megjelenítjük)

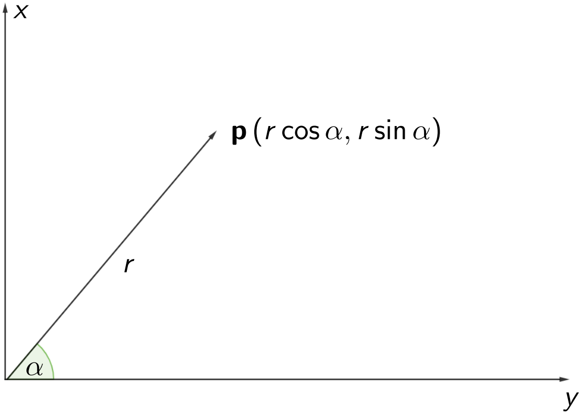
Polárkoordináták

Polárkoordináták

A polárkoordináta rendszer olyan koordináta rendszer, melynek pontjait egy szög és egy távolság adat alapján adjuk meg.

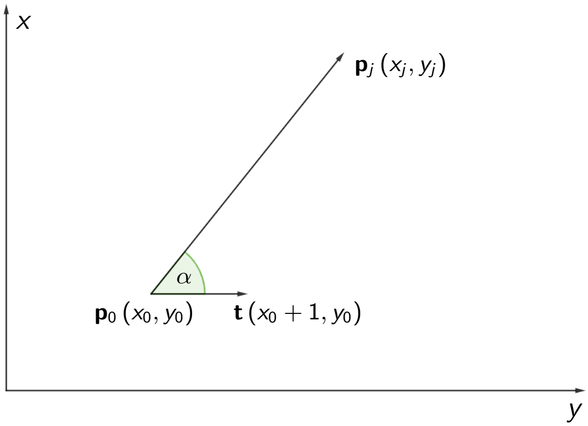
Polárkoordináták

A polárkoordináta rendszer olyan koordináta rendszer, melynek pontjait egy szög és egy távolság adat alapján adjuk meg.



Polárkoordináták

Egy adott $\mathbf{p}_j$ pont $\mathbf{p}_0$-ra vonatkozó polárkoordinátái



Polárkoordináták

A későbbiekben szükséges lesz arra, hogy meghatározzuk a $\mathbf{p}_j$ pont $\mathbf{p}_0$-ra vonatkozó polárszögét.

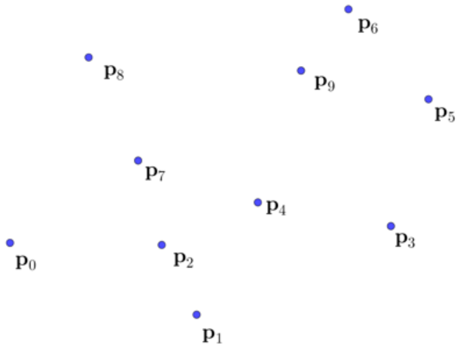
Polárkoordináták

A későbbiekben szükséges lesz arra, hogy meghatározzuk a $\mathbf{p}_j$ pont $\mathbf{p}_0$-ra vonatkozó polárszögét.

$$\alpha = \begin{cases} \arcsin\left(\frac{y_j - y_0}{\sqrt{(x_j - x_0)^2 + (y_j - y_0)^2}}\right) & \text{ha } x_j > x_0 \\ \pi - \arcsin\left(\frac{y_j - y_0}{\sqrt{(x_j - x_0)^2 + (y_j - y_0)^2}}\right) & \text{egyébként} \end{cases}$$

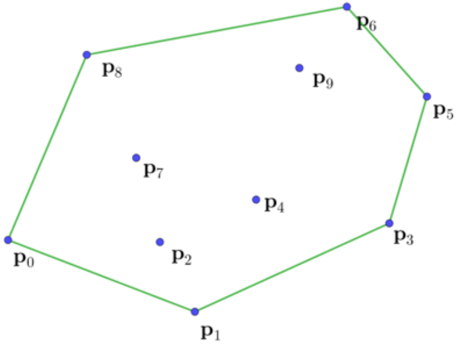
Konvex burok algoritmus (Graham pásztázás)

Legyenek adottak a $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n$ pontok.



Konvex burok algoritmus (Graham pásztázás)

Legyenek adottak a $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n$ pontok.
Keressük a $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n$ pontok konvex burkát, azaz azt a legszűkebb konvex poligont, mely vagy belső pontként, vagy csúcspontként tartalmazza azokat.



Konvex burok algoritmus (Graham pásztázás)

Szükségünk lesz egy verem adatszerkezetre az alábbi eljárásokkal

Konvex burok algoritmus (Graham pásztázás)

Szükségünk lesz egy verem adatszerkezetre az alábbi eljárásokkal

- Verem inicializálása

Konvex burok algoritmus (Graham pásztázás)

Szükségünk lesz egy verem adatszerkezetre az alábbi eljárásokkal

- Verem inicializálása
- Verem tetejére elem beszúrása (Push)

Konvex burok algoritmus (Graham pásztázás)

Szükségünk lesz egy verem adatszerkezetre az alábbi eljárásokkal

- Verem inicializálása
- Verem tetejére elem beszúrása (Push)
- Verem tetejéről elem kivétele (Pop)

Konvex burok algoritmus (Graham pásztázás)

Szükségünk lesz egy verem adatszerkezetre az alábbi eljárásokkal

- Verem inicializálása
- Verem tetejére elem beszúrása (Push)
- Verem tetejéről elem kivétele (Pop)
- Legfelső elem értékének kiolvasása eltávolítás nélkül (Top)

Konvex burok algoritmus (Graham pásztázás)

Szükségünk lesz egy verem adatszerkezetre az alábbi eljárásokkal

- Verem inicializálása
- Verem tetejére elem beszúrása (Push)
- Verem tetejéről elem kivétele (Pop)
- Legfelső elem értékének kiolvasása eltávolítás nélkül (Top)
- Legfelső elem alatti elem értékének kiolvasása eltávolítás nélkül (Top2)

Konvex burok algoritmus (Graham pásztázás)

Az algoritmus menete a következő.

Konvex burok algoritmus (Graham pásztázás)

Az algoritmus menete a következő.

- Legyen $\mathbf{p}_0$ a pontok közül a minimális y koordinátájúak közül az, amelyiknek x koordinátája szintén minimális

Konvex burok algoritmus (Graham pásztázás)

Az algoritmus menete a következő.

- Legyen $\mathbf{p}_0$ a pontok közül a minimális y koordinátájúak közül az, amelyiknek x koordinátája szintén minimális
- Legyenek a $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n$ pontok a $\mathbf{p}_0$-ra vonatkozó polárszögeik szerint rendezve. (Ha több pontnak is ugyanaz a polárszöge, akkor csak a $\mathbf{p}_0$-tól legtávolabbi pontot tartjuk meg további feldolgozásra)

Konvex burok algoritmus (Graham pásztázás)

Az algoritmus menete a következő.

- Legyen $\mathbf{p}_0$ a pontok közül a minimális y koordinátájúak közül az, amelyiknek x koordinátája szintén minimális
- Legyenek a $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n$ pontok a $\mathbf{p}_0$-ra vonatkozó polárszögeik szerint rendezve. (Ha több pontnak is ugyanaz a polárszöge, akkor csak a $\mathbf{p}_0$-tól legtávolabbi pontot tartjuk meg további feldolgozásra)
- Inicializáljuk a vermet

Konvex burok algoritmus (Graham pásztázás)

Az algoritmus menete a következő.

- Legyen $\mathbf{p}_0$ a pontok közül a minimális y koordinátájúak közül az, amelyiknek x koordinátája szintén minimális
- Legyenek a $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n$ pontok a $\mathbf{p}_0$-ra vonatkozó polárszögeik szerint rendezve. (Ha több pontnak is ugyanaz a polárszöge, akkor csak a $\mathbf{p}_0$-tól legtávolabbi pontot tartjuk meg további feldolgozásra)
- Inicializáljuk a vermet
- Elhelyezzük benne a $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ pontokat

Konvex burok algoritmus (Graham pásztázás)

Az algoritmus menete a következő.

- Legyen $\mathbf{p}_0$ a pontok közül a minimális y koordinátájúak közül az, amelyiknek x koordinátája szintén minimális
- Legyenek a $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n$ pontok a $\mathbf{p}_0$-ra vonatkozó polárszögeik szerint rendezve. (Ha több pontnak is ugyanaz a polárszöge, akkor csak a $\mathbf{p}_0$-tól legtávolabbi pontot tartjuk meg további feldolgozásra)
- Inicializáljuk a vermet
- Elhelyezzük benne a $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ pontokat
- A többi ponton iterációval haladunk végig

Konvex burok algoritmus (Graham pásztázás)

Az algoritmus menete a következő.

- Legyen $\mathbf{p}_0$ a pontok közül a minimális y koordinátájúak közül az, amelyiknek x koordinátája szintén minimális
- Legyenek a $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n$ pontok a $\mathbf{p}_0$-ra vonatkozó polárszögeik szerint rendezve. (Ha több pontnak is ugyanaz a polárszöge, akkor csak a $\mathbf{p}_0$-tól legtávolabbi pontot tartjuk meg további feldolgozásra)
- Inicializáljuk a vermet
- Elhelyezzük benne a $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ pontokat
- A többi ponton iterációval haladunk végig
  - Minden pont esetén egy újabb iterációt hajtunk végre addig, amíg a Top2, Top és $\mathbf{p}_j$ pontok által alkotott szög nem végez balfordulatot

Konvex burok algoritmus (Graham pásztázás)

Az algoritmus menete a következő.

- Legyen $\mathbf{p}_0$ a pontok közül a minimális y koordinátájúak közül az, amelyiknek x koordinátája szintén minimális
- Legyenek a $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n$ pontok a $\mathbf{p}_0$-ra vonatkozó polárszögeik szerint rendezve. (Ha több pontnak is ugyanaz a polárszöge, akkor csak a $\mathbf{p}_0$-tól legtávolabbi pontot tartjuk meg további feldolgozásra)
- Inicializáljuk a vermet
- Elhelyezzük benne a $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ pontokat
- A többi ponton iterációval haladunk végig
  - Minden pont esetén egy újabb iterációt hajtunk végre addig, amíg a Top2, Top és $\mathbf{p}_j$ pontok által alkotott szög nem végez balfordulatot
    - Kivesszük a legfelső elemet (Pop)

Konvex burok algoritmus (Graham pásztázás)

Az algoritmus menete a következő.

- Legyen $\mathbf{p}_0$ a pontok közül a minimális y koordinátájúak közül az, amelyiknek x koordinátája szintén minimális
- Legyenek a $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n$ pontok a $\mathbf{p}_0$-ra vonatkozó polárszögeik szerint rendezve. (Ha több pontnak is ugyanaz a polárszöge, akkor csak a $\mathbf{p}_0$-tól legtávolabbi pontot tartjuk meg további feldolgozásra)
- Inicializáljuk a vermet
- Elhelyezzük benne a $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ pontokat
- A többi ponton iterációval haladunk végig
    - Minden pont esetén egy újabb iterációt hajtunk végre addig, amíg a Top2, Top és $\mathbf{p}_j$ pontok által alkotott szög nem végez balfordulatot
        - Kivesszük a legfelső elemet (Pop)
        - Beszúrjuk a verembe a $\mathbf{p}_j$ pontot (Push)

Poligon kitöltése

Az általános módszer ebben az esetben is az él-flag módszer

Poligon kitöltése

- Az általános módszer ebben az esetben is az él-flag módszer
- A legdélibb és legészakibb csúcsok között minden scanline esetén az alábbi lépéseket eszközöljük.

Poligon kitöltése

Az általános módszer ebben az esetben is az él-flag módszer
A legdélibb és legészakibb csúcsok között minden scanline esetén az alábbi lépéseket eszközöljük.

- Meghatározzuk a scanline metszéspontjait a poligon minden élével

Poligon kitöltése

Az általános módszer ebben az esetben is az él-flag módszer
A legdélibb és legészakibb csúcsok között minden scanline esetén az alábbi lépéseket eszközöljük.

- Meghatározzuk a scanline metszéspontjait a poligon minden élével
- A metszéspontokat rendezzük az x koordináta szerint

Poligon kitöltése

Az általános módszer ebben az esetben is az él-flag módszer
A legdélibb és legészakibb csúcsok között minden scanline esetén az alábbi lépéseket eszközöljük.

- Meghatározzuk a scanline metszéspontjait a poligon minden élével
- A metszéspontokat rendezzük az x koordináta szerint
- Egy flag segítségével megfestjük a metszéspontok közötti pixeleket

Poligon kitöltése – belső- és külső pontok

Poligon kitöltése – belső- és külső pontok

- A poligon csúcsai egész koordinátákkal adottak

Poligon kitöltése – belső- és külső pontok

- A poligon csúcsai egész koordinátákkal adottak
- A poligon éleit egy szakaszrajzoló algoritmussal (DDA, MidPoint) meg lehet határozni

Poligon kitöltése – belső- és külső pontok

- A poligon csúcsai egész koordinátákkal adottak
- A poligon éleit egy szakaszrajzoló algoritmussal (DDA, MidPoint) meg lehet határozni
- Kérdés, hogy az egymással szomszédos (azonos y koordinátájú) pixelek esetén melyiket tekintsük belső, és melyiket külső pontnak

Poligon kitöltése – belső- és külső pontok

- A poligon csúcsai egész koordinátákkal adottak
- A poligon éleit egy szakaszrajzoló algoritmussal (DDA, MidPoint) meg lehet határozni
- Kérdés, hogy az egymással szomszédos (azonos y koordinátájú) pixelek esetén melyiket tekintsük belső, és melyiket külső pontnak
    - Megoldás, hogy megkülönböztetünk bal- és jobboldali éleket aszerint, hogy párosadik, vagy páratlanodik metszéspontról van szó

Poligon kitöltése – belső- és külső pontok

- A poligon csúcsai egész koordinátákkal adottak
- A poligon éleit egy szakaszrajzoló algoritmussal (DDA, MidPoint) meg lehet határozni
- Kérdés, hogy az egymással szomszédos (azonos y koordinátájú) pixelek esetén melyiket tekintsük belső, és melyiket külső pontnak
  - Megoldás, hogy megkülönböztetünk bal- és jobboldali éleket aszerint, hogy párosadik, vagy páratlanodik metszéspontról van szó
  - Bal oldali esetén lefelé, jobb oldali esetén felfelé kerekítünk (valami kerekített MidPoint)

Poligon kitöltése – belső- és külső pontok

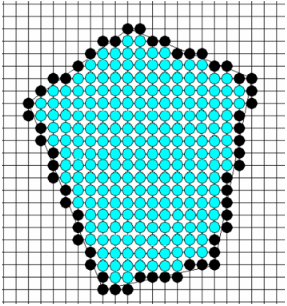
- A poligon csúcsai egész koordinátákkal adottak
- A poligon éleit egy szakaszrajzoló algoritmussal (DDA, MidPoint) meg lehet határozni
- Kérdés, hogy az egymással szomszédos (azonos y koordinátájú) pixelek esetén melyiket tekintsük belső, és melyiket külső pontnak
  - Megoldás, hogy megkülönböztetünk bal- és jobboldali éleket aszerint, hogy párosadik, vagy páratlanodik metszéspontról van szó
  - Bal oldali esetén lefelé, jobb oldali esetén felfelé kerekítünk (valami kerekített MidPoint)

Poligon kitöltése – belső- és külső pontok

- A poligon csúcsai egész koordinátákkal adottak
- A poligon éleit egy szakaszrajzoló algoritmussal (DDA, MidPoint) meg lehet határozni

Poligon kitöltése – belső- és külső pontok

- A poligon csúcsai egész koordinátákkal adottak
- A poligon éleit egy szakaszrajzoló algoritmussal (DDA, MidPoint) meg lehet határozni
- Kérdés, hogy az egymással szomszédos (azonos y koordinátájú) pixelek esetén melyiket tekintsük belső, és melyiket külső pontnak



Poligon kitöltése – belső- és külső pontok

Melyik pixeleket tekintsük belső-, és melyiket külső pontnak?

Poligon kitöltése – belső- és külső pontok

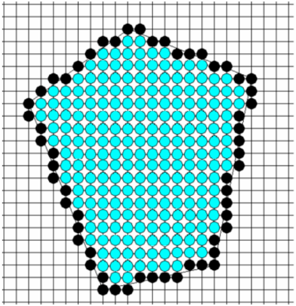
Melyik pixeleket tekintsük belső-, és melyiket külső pontnak?

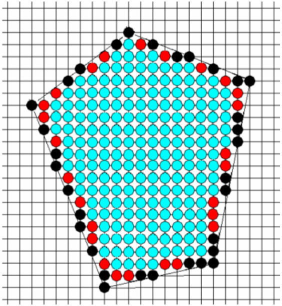
- Megoldás, hogy megkülönböztetünk bal- és jobboldali éleket aszerint, hogy párosadik, vagy páratlanodik metszéspontról van szó

Poligon kitöltése – belső- és külső pontok

Melyik pixeleket tekintsük belső-, és melyiket külső pontnak?

- Megoldás, hogy megkülönböztetünk bal- és jobboldali éleket aszerint, hogy párosadik, vagy páratlanodik metszéspontról van szó
- Bal oldali esetén lefelé, jobb oldali esetén felfelé kerekítünk (valami kerekített MidPoint)





Poligon kitöltése – Felmerülő problémák

Egész koordináta esetén külső, vagy belső pixel legyen?

Poligon kitöltése – Felmerülő problémák

Egész koordináta esetén külső, vagy belső pixel legyen?

- Bal oldali él esetén belső, jobb oldali esetén külső.

Poligon kitöltése – Felmerülő problémák

Egész koordináta esetén külső, vagy belső pixel legyen?

- Bal oldali él esetén belső, jobb oldali esetén külső.

Mi történjen, ha csúcsponton halad át a scanline?

Poligon kitöltése – Felmerülő problémák

Egész koordináta esetén külső, vagy belső pixel legyen?

- Bal oldali él esetén belső, jobb oldali esetén külső.

Mi történjen, ha csúcsponton halad át a scanline?

- A flag-et csak déli csúcs esetén állítjuk.

Poligon kitöltése – Felmerülő problémák

Egész koordináta esetén külső, vagy belső pixel legyen?

- Bal oldali él esetén belső, jobb oldali esetén külső.

Mi történjen, ha csúcsponton halad át a scanline?

- A flag-et csak déli csúcs esetén állítjuk.

Mi a teendő vízszintes élek esetén?

Poligon kitöltése – Felmerülő problémák

Egész koordináta esetén külső, vagy belső pixel legyen?

- Bal oldali él esetén belső, jobb oldali esetén külső.

Mi történjen, ha csúcsponton halad át a scanline?

- A flag-et csak déli csúcs esetén állítjuk.

Mi a teendő vízszintes élek esetén?

- A déli élet belsőként, az északi élet külsőként definiáljuk (hasonlóan, mint téglalap esetén)

Poligon kitöltése – Felmerülő problémák

Egész koordináta esetén külső, vagy belső pixel legyen?

- Bal oldali él esetén belső, jobb oldali esetén külső.

Mi történjen, ha csúcsponton halad át a scanline?

- A flag-et csak déli csúcs esetén állítjuk.

Mi a teendő vízszintes élek esetén?

- A déli élet belsőként, az északi élet külsőként definiáljuk (hasonlóan, mint téglalap esetén)
- Ez egyébként automatikusan következik az előző pontból

Poligon kitöltése – Él tábla

Az algoritmus megvalósításához érdemes használni egy él táblát (ET) az alábbiak szerint

Poligon kitöltése – Él tábla

Az algoritmus megvalósításához érdemes használni egy él táblát (ET) az alábbiak szerint

- Az éltábla annyi elemű, ahány scanline van

Poligon kitöltése – Él tábla

Az algoritmus megvalósításához érdemes használni egy él táblát (ET) az alábbiak szerint

- Az éltábla annyi elemű, ahány scanline van
- A tábla elemei listák, melyekben azon élekről van adat, melyek az adott scanline-t metszik

Poligon kitöltése – Él tábla

Az algoritmus megvalósításához érdemes használni egy él táblát (ET) az alábbiak szerint

- Az éltábla annyi elemű, ahány scanline van
- A tábla elemei listák, melyekben azon élekről van adat, melyek az adott scanline-t metszik
- A vízszintes lista élei x_{min} koordinátájuk szerint vannak rendezve.

Poligon kitöltése – Él tábla

Az algoritmus megvalósításához érdemes használni egy él táblát (ET) az alábbiak szerint

- Az éltábla annyi elemű, ahány scanline van
- A tábla elemei listák, melyekben azon élekről van adat, melyek az adott scanline-t metszik
- A vízszintes lista élei x_{min} koordinátájuk szerint vannak rendezve.
- Tárolt adatok az élekről:

Poligon kitöltése – Él tábla

Az algoritmus megvalósításához érdemes használni egy él táblát (ET) az alábbiak szerint

- Az éltábla annyi elemű, ahány scanline van
- A tábla elemei listák, melyekben azon élekről van adat, melyek az adott scanline-t metszik
- A vízszintes lista élei x_{min} koordinátájuk szerint vannak rendezve.
- Tárolt adatok az élekről:
  - az y koordináták az élek alacsonyabb csúcsának y koordinátái

Poligon kitöltése – Él tábla

Az algoritmus megvalósításához érdemes használni egy él táblát (ET) az alábbiak szerint

- Az éltábla annyi elemű, ahány scanline van
- A tábla elemei listák, melyekben azon élekről van adat, melyek az adott scanline-t metszik
- A vízszintes lista élei x_{min} koordinátájuk szerint vannak rendezve.
- Tárolt adatok az élekről:
    - az y koordináták az élek alacsonyabb csúcsának y koordinátái
    - az y_{max} az él maximális y koordinátája

Poligon kitöltése – Él tábla

Az algoritmus megvalósításához érdemes használni egy él táblát (ET) az alábbiak szerint

- Az éltábla annyi elemű, ahány scanline van
- A tábla elemei listák, melyekben azon élekről van adat, melyek az adott scanline-t metszik
- A vízszintes lista élei x_{min} koordinátájuk szerint vannak rendezve.
- Tárolt adatok az élekről:
    - az y koordináták az élek alacsonyabb csúcsának y koordinátái
    - az y_{max} az él maximális y koordinátája
    - az x_{min} az él alacsonyabb csúcsának az x koordinátája

Poligon kitöltése – Él tábla

Az algoritmus megvalósításához érdemes használni egy él táblát (ET) az alábbiak szerint

- Az éltábla annyi elemű, ahány scanline van
- A tábla elemei listák, melyekben azon élekről van adat, melyek az adott scanline-t metszik
- A vízszintes lista élei x_{min} koordinátájuk szerint vannak rendezve.
- Tárolt adatok az élekről:
    - az y koordináták az élek alacsonyabb csúcsának y koordinátái
    - az y_{max} az él maximális y koordinátája
    - az x_{min} az él alacsonyabb csúcsának az x koordinátája
    - $\frac{1}{m}$ az él meredeksége

Poligon kitöltése – Él tábla

```
STRUKTÚRA _ET //Éltábla
  EGÉSZ: X;
  _ETL: ETL;
  _ET: KÖVETKEZŐ;
STRUKTÚRA_VÉGE;

STRUKTÚRA _ET //Éltáblához tartozó éllista
  EGÉSZ: X1, X2;
  VALÓS: M;
  _ETL: KÖVETKEZŐ;
STRUKTÚRA_VÉGE;

STRUKTÚRA _AET //Aktív éltábla
  VALÓS: X, M;
  EGÉSZ: Y2;
  _AETL: KÖVETKEZŐ;
STRUKTÚRA_VÉGE;
```

Poligon kitöltése – Él tábla (algoritmus)

Poligon kitöltése – Él tábla (algoritmus)

- Töltsük fel az ET listát

Poligon kitöltése – Él tábla (algoritmus)

- Töltsük fel az ET listát
- Legyen y az ET lista első elemének az y koordinátája

Poligon kitöltése – Él tábla (algoritmus)

- Töltsük fel az ET listát
- Legyen y az ET lista első elemének az y koordinátája
- Inicializáljuk üresnek az AET listát

Poligon kitöltése – Él tábla (algoritmus)

- Töltsük fel az ET listát
- Legyen y az ET lista első elemének az y koordinátája
- Inicializáljuk üresnek az AET listát
- Ismételjük a következőket, amíg az ET és AET listák üresek nem lesznek

Poligon kitöltése – Él tábla (algoritmus)

- Töltsük fel az ET listát
- Legyen y az ET lista első elemének az y koordinátája
- Inicializáljuk üresnek az AET listát
- Ismételjük a következőket, amíg az ET és AET listák üresek nem lesznek
  - Tegyük az AET listába azokat az éleket, amelyekre $y = y_{min}$, majd rendezzük az AET-ben lévő éleket az x koordináta szerint

Poligon kitöltése – Él tábla (algoritmus)

- Töltsük fel az ET listát
- Legyen y az ET lista első elemének az y koordinátája
- Inicializáljuk üresnek az AET listát
- Ismételjük a következőket, amíg az ET és AET listák üresek nem lesznek
  - Tegyük az AET listába azokat az éleket, amelyekre $y = y_{min}$, majd rendezzük az AET-ben lévő éleket az x koordináta szerint
  - Rajzoljuk ki az y scanline-t, az AET-ben lévő x koordináta-párok között, figyelembe véve a paritást

Poligon kitöltése – Él tábla (algoritmus)

- Töltsük fel az ET listát
- Legyen y az ET lista első elemének az y koordinátája
- Inicializáljuk üresnek az AET listát
- Ismételjük a következőket, amíg az ET és AET listák üresek nem lesznek
  - Tegyük az AET listába azokat az éleket, amelyekre $y = y_{min}$, majd rendezzük az AET-ben lévő éleket az x koordináta szerint
  - Rajzoljuk ki az y scanline-t, az AET-ben lévő x koordináta-párok között, figyelembe véve a paritást
  - y értékét növeljük 1-gyel

Poligon kitöltése – Él tábla (algoritmus)

- Töltsük fel az ET listát
- Legyen y az ET lista első elemének az y koordinátája
- Inicializáljuk üresnek az AET listát
- Ismételjük a következőket, amíg az ET és AET listák üresek nem lesznek
  - Tegyük az AET listába azokat az éleket, amelyekre $y = y_{min}$, majd rendezzük az AET-ben lévő éleket az x koordináta szerint
  - Rajzoljuk ki az y scanline-t, az AET-ben lévő x koordináta-párok között, figyelembe véve a paritást
  - y értékét növeljük 1-gyel
  - Távolítsuk el azokat az éleket az AET-ből, amelyekre $y = y_{max}$

Poligon kitöltése – Él tábla (algoritmus)

- Töltsük fel az ET listát
- Legyen y az ET lista első elemének az y koordinátája
- Inicializáljuk üresnek az AET listát
- Ismételjük a következőket, amíg az ET és AET listák üresek nem lesznek
  - Tegyük az AET listába azokat az éleket, amelyekre $y = y_{min}$, majd rendezzük az AET-ben lévő éleket az x koordináta szerint
  - Rajzoljuk ki az y scanline-t, az AET-ben lévő x koordináta-párok között, figyelembe véve a paritást
  - y értékét növeljük 1-gyel
  - Távolítsuk el azokat az éleket az AET-ből, amelyekre $y = y_{max}$
  - Minden nem függőleges AET-beli él esetén legyen $x = x + \frac{1}{m}$

Köszönöm a figyelmet!