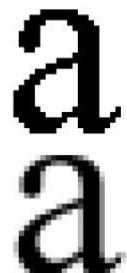
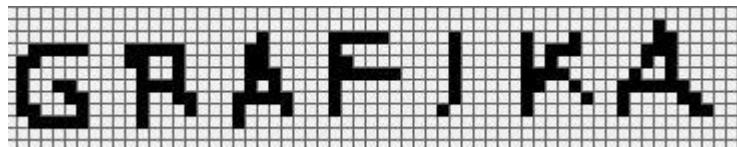


## 4. RASTERSKA GRAFIKA

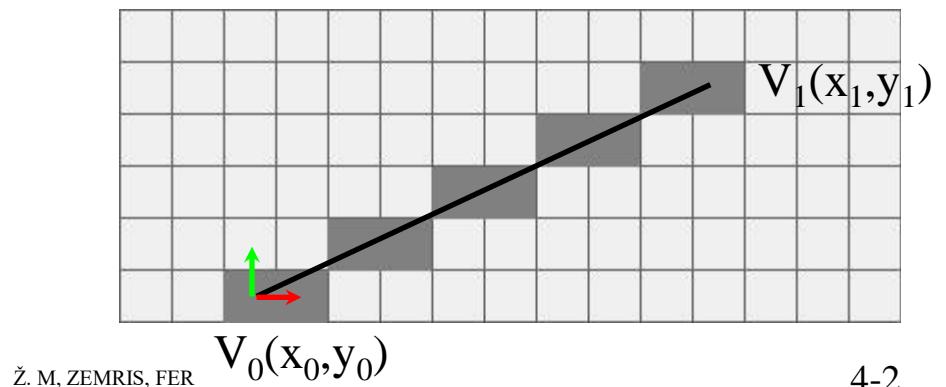
### Prikaz na zaslonu računala

- potrebna je rasterizacija objekata koje prikazujemo (*pixel shader engine*)
- kontinuirane objekte potrebno je diskretizirati u 2D prostoru
  - prikaz dužine, poligona, kružnice, zahtijeva diskretizaciju
  - prikaz slike na računalu zahtijeva diskretizaciju (uzorkovanje), kao bi svakom slikovnom elementu pridijelili intenzitet (boju), uređaji za snimanje npr. digitalni fotoaparat obavlja postupak uzorkovanja
  - postupak uzorkovanja – javlja se neželjeni učinak alias koji se očitije u nazubljenim linijama ili neželjenim uzorcima na teksturi (*moire*)



## 4.1 Bresenhamov postupak

- Prikaz dužine
  - određuje koje točke rastera trebaju biti osvijetljene kako bi načinili prikaz ravne linije [http://threejs.org/examples/webgl\\_interactive\\_voxelpainter](http://threejs.org/examples/webgl_interactive_voxelpainter)
  - postupak je u konačnici ostvariv upotrebom cjelobrojnog zbrajanja (oduzimanja) i posmaka
- osnovni algoritam 0-45°
  - odabiremo cjelobrojne vrijednosti koje odgovaraju središtimu slikovnih elemenata (pixela)  $V_0(x_0, y_0)$ ,  $V_1(x_1, y_1)$
  - jednadžba pravca kroz  $V_0, V_1$ 
$$y - y_0 = \frac{y_1 - y_0}{x_1 - x_0} (x - x_0), \quad y = \frac{\Delta y}{\Delta x} (x - x_0) + y_0, \quad \Delta y = y_1 - y_0 \\ \Delta x = x_1 - x_0$$
  - x, y su koordinate slikovnog elementa kojeg ćemo osvijetliti  
potrebno je zaokruživanje  
(cjelobrojne koordinate)

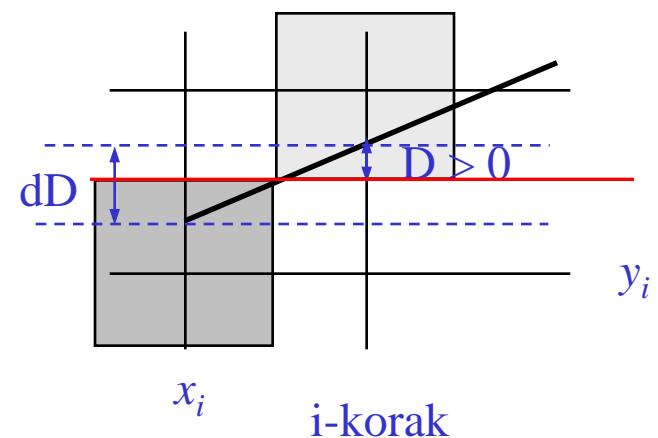
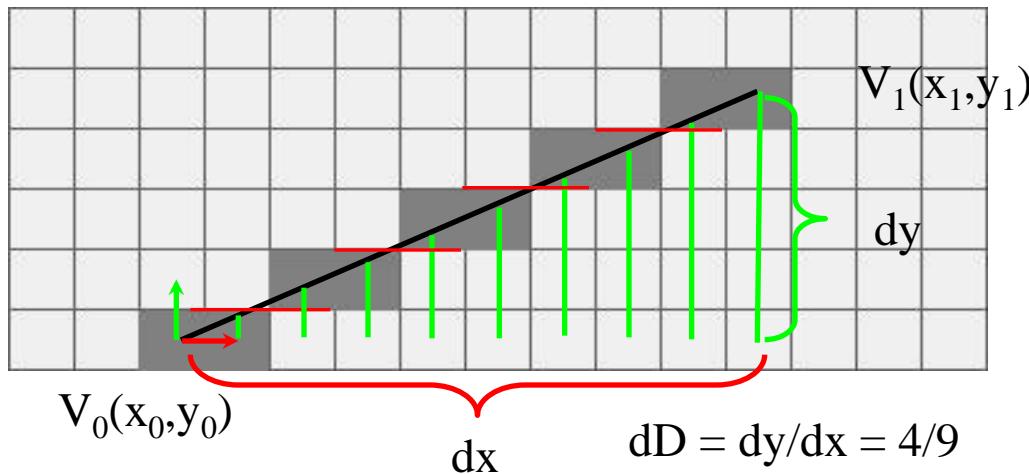


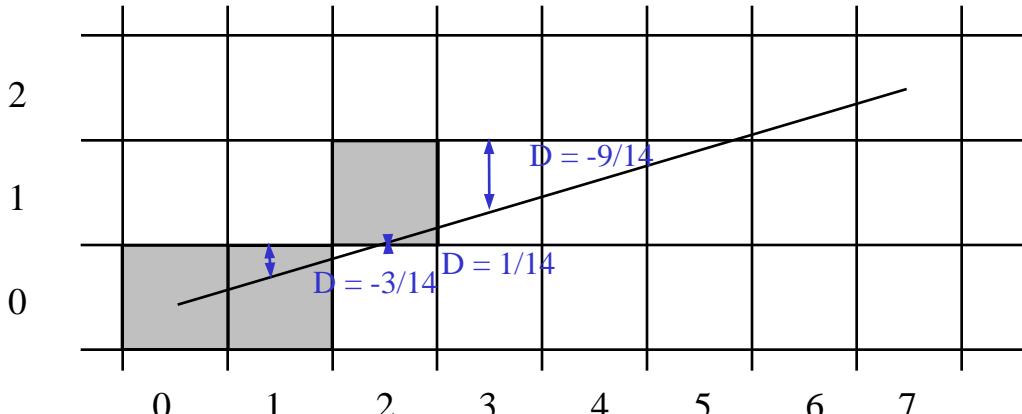
```

// osnovni algoritam Bresenhama za cjelobrojne koordinate x0, x1, y0, y1
line(x0, x1, y0, y1) {
    int dx = x1 - x0, dy = y1 - y0; // Pretpostavimo da linija nije vertikalna
    float dD = abs(dy / (float) dx);           // tj. dx != 0
    float D = dD - 0.5,
    int y = y0;

    for (x= x0 to x1) {
        crtaj(x,y);
        if ( D ≥ 0 ) {
            y = y + 1;
            D = D - 1.0;
        }
        D = D + dD;
    }
}

```

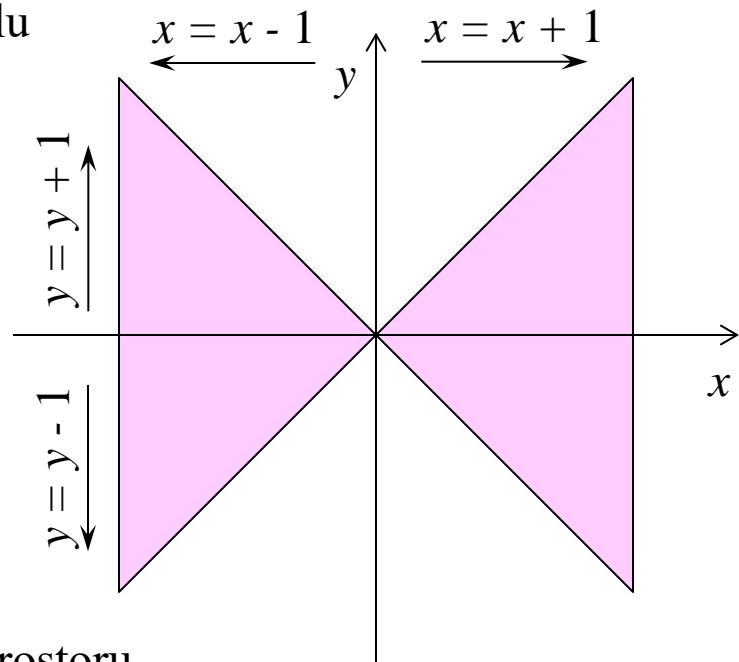




$$\begin{aligned} dx &= 7, dy = 2 \\ dD &= 2/7, \\ D &= 2/7 - 1/2 = -3/14, \\ y &= 0 \end{aligned}$$

```
for (x = 0 to 7) {
    crtaj(x,y);
    if (D ≥ 0) {y++; D- -};
    D = D + 2/7;
}
```

- ishodište zaslona je obično u gornjem lijevom uglu
- proširenje postupka  $0 - 45$  na sve kutove
  - razlikujemo 2 područja
    - $x$  se povećava/smanjuje za 1
    - $y$  se povećava/smanjuje za 1
- <http://bert.stuy.edu/pbrooks/graphics/demos/BresenhamDemo.htm>
- algoritam koji radi s cjelobrojnim vrijednostima
  - prilagođeno sklopovskoj implementaciji
  - cijeli algoritam pomnožimo s  $dx$   
(sve naredbe na koje ima utjecaj)
- postupak je lako proširiv na diskretizaciju u 3D prostoru



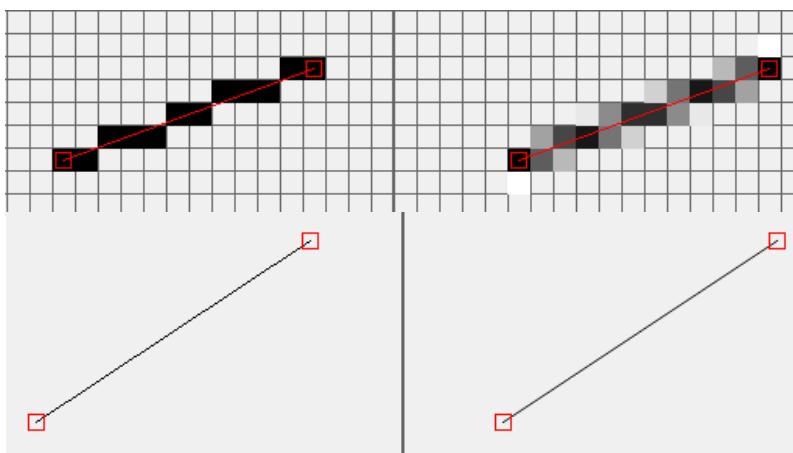
## 4.2 Neželjeni učinci uslijed diskretizacije

- dva načina umanjivanja učinka diskretizacije
  - udaljenost D određuje hoćemo li osvijetliti gornji slikovni element ili donji možemo koristiti D za određivanje intenziteta (sive razine) slikovnog elementa
    - npr. ako je  $D = 0$ , tada je  $D_0 = 0,5$ ,  $D_1 = 0,5$ ,  $(D_1 + D_0 = 1)$ , 0-crno, 1-bijelo)

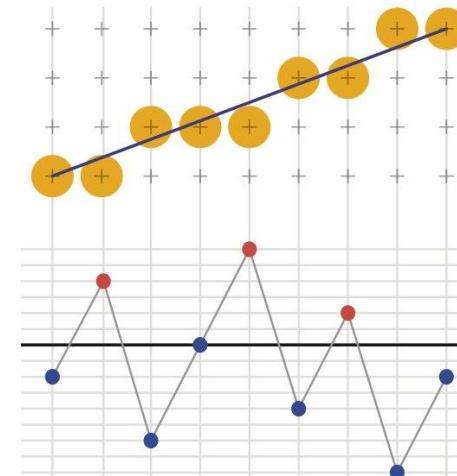
<https://ferko.fer.hr/minilessons/ml/minilesson/cc=DEFAULT,c=77f27c05d7164a7cb227cbdf8321d051/170f0e78317b45faab1318bd520ac279/item/4/star>

<http://members.chello.at/easyfilter/canvas.html>

```
glLineWidth(1.5);           // potreban je RGBA mod  
glEnable(GL_LINE_SMOOTH); // ili glEnable(GL_POINT_SMOOTH);  
glHint(GL_LINE_SMOOTH_HINT, GL_NICEST); // GL_NICEST ili GL_DONT_CARE  
glEnable(GL_BLEND); // omogućeno stapanje  
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
```



Ž. M. ZEMRIS, FER



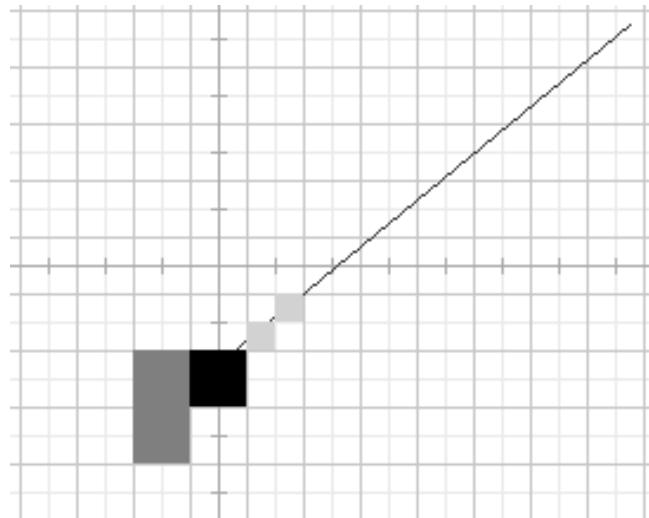
# Neželjeni učinci uslijed diskretizacije

- povećano uzorkovanje – povećamo razlučivost rastera

<http://jyunming-chen.github.io/tutsplus/tutsplusS-5.html>

- odredimo slikovne elemente koji bi bili osvijetljeni, na osnovi toga određujemo konačan intenzitet

<http://www.kevs3d.co.uk/dev/shaders/polyhedra3.html>

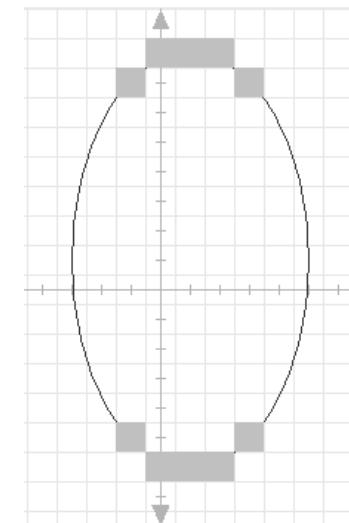
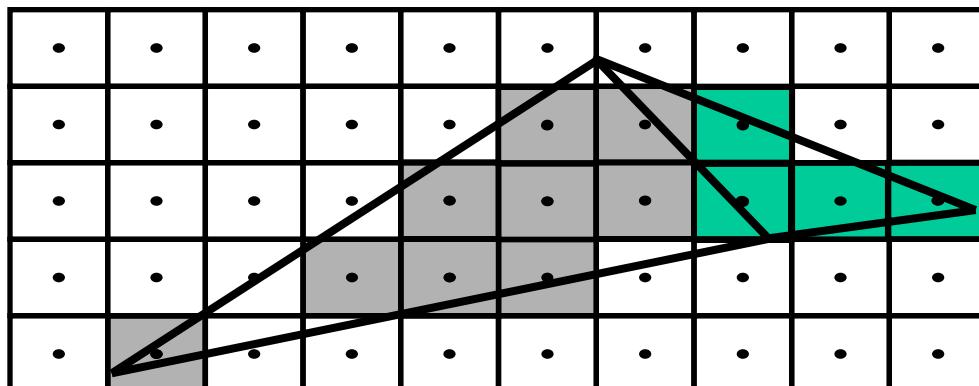


## 4.4 Rasterizacija dvodimanskih objekata

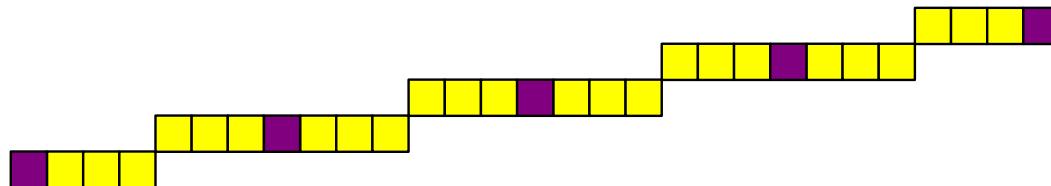
- kružnica – osnovna ideja Bresenhamovog postupka primjenjiva je na analitički definirane objekte npr. kružnica, elipsa

<http://members.chello.at/easyfilter/canvas.html>

- rasterizacija poligona - rubovi
- popunjavanje poligona (uzorkovanje)
  - određivanje rubnih točaka, DDA algoritam
  - popunjavanje liniju po liniju



# DDA algoritam (Digitalni Diferencijalni Analizator)



Slikovni elementi osvjetljeni Bresenhamovim postupkom.

Slikovni elementi osvjetljeni DDA postupkom.

- za svaki  $y$  imamo samo **jedan** slikovni element
- <http://www.sunshine2k.de/coding/java/Polygon/Filling/FillPolygon.htm>

```
// DDA algoritam
```

```
xi = x0;  
xf = -0.5;  
mi = (x1 - x0) div (y1 - y0);  
mf = (x1 - x0)/(y1 - y0) - mi;  
for (y = y0; y <= y1; y=y+1) {  
    crtaj (xi,y);  
    xi = xi + mi;  
    xf = xf + mf;  
    if (xf>=0) {  
        xi = xi + 1;  
        xf = xf - 1;  
    }  
}
```

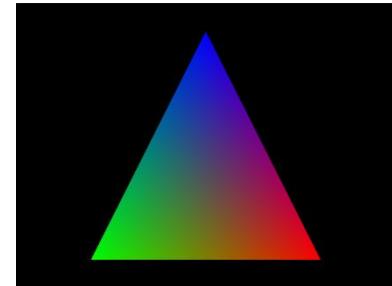
```
// DDA - Cjelobrojne varijable
```

```
xi = x0;  
xf = -(y1-y0);  
mi = (x1 - x0) div (y1 - y0)  
mf = 2 * ((x1 - x0) mod (y1 - y0))  
for { y = y0; y <= y1; y=y+1) {  
    crtaj (xi,y);  
    xi = xi + mi;  
    xf = xf + mf;  
    if (xf>=0) {  
        xi = xi + 1;  
        xf = xf - 2 * (y1 - y0);  
    }  
}
```

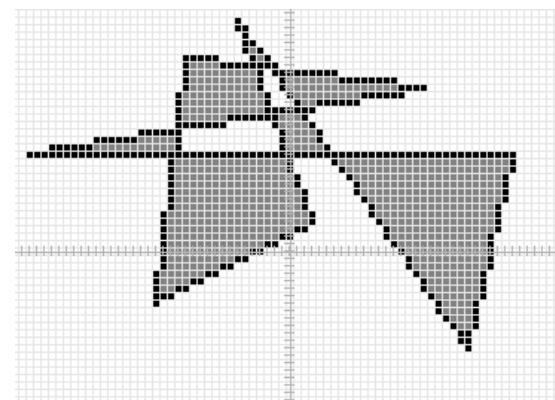
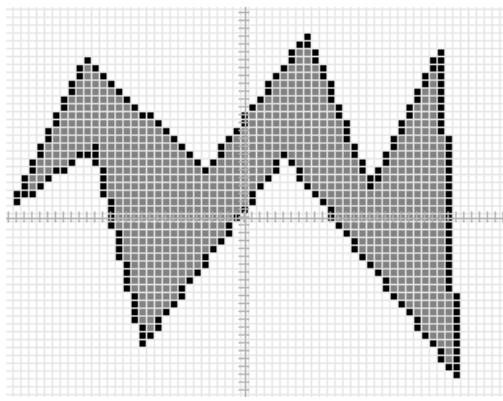
## Popunjavanje poligona

- konveksni poligoni – DDA određuje jednu lijevu i jednu desnu točku a može se odrediti i boja

<http://www.sunshine2k.de/coding/java/TriangleRasterization/TriangleRasterization.html>



- konkavni poligoni – bojanje od neparnih do parnih točaka  
(kompleksni poligoni, s rupom)



- problem popunjavanja i prianjanja poligona
  - nazubljeni rubovi <http://acko.net/files/fullfrontal/fullfrontal/webglmath/online.html>
- određivanje odnosa točke i poligona

# Primjeri drugih primjena Bresenhamovog i DDA algoritma

- određivanje vidljivosti scene za zadani položaj promatrača i smjer gledanja

<http://www.emanueleferonato.com/2015/12/10/calculating-dynamic-light-and-shadows-in-tile-based-roguelike-games-part-1-bresenhams-line-algorithm/>

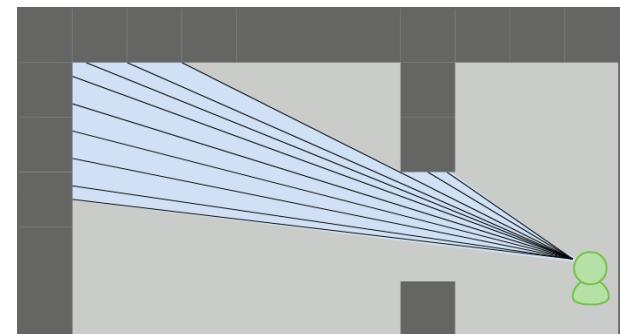
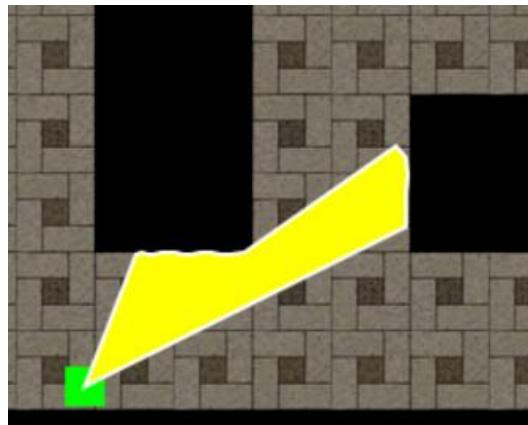
<http://www.emanueleferonato.com/2014/10/10/a-quick-html5-survival-horror-prototype-made-with-phaser/> (strelice)

- određivanje vidljivosti scene u labirintima

– <http://www.spacegoo.com/maze/>

- proširenje na 3D prostor

– <https://classic.minecraft.net/?join=-sKovfE6lcQJO10A>  
ili <http://didact.us/spatial-events/>



## Alias učinci

- frekvencija uzorkovanja (gustoća slikovnih) elemenata je premala
- teorija uzorkovanja Shannon'ov torom uzorkovanja vrijedi i za 2D, 3D ... (također i u vremenskoj domeni, odnosno broj slika/s)
- prije uzorkovanja potrebno je načiniti filtriranje nisko propusnim LP filtrom, tako da frekvencijski ograničimo signal (sliku)
- različiti rekonstrukcijski filtri kojima možemo rekonstruirati ponovo kontinuirani oblik
- <https://lspo.feri.um.si/CoLoS/applets/fft/ftd.html>

