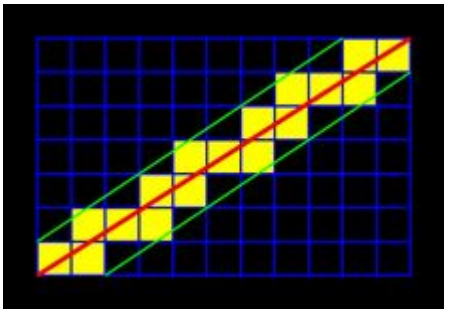
Modelització i Visualització

Exercici 1. Dibuixar una recta

David Domènech Vilà
2018-2019

Algorisme de Bresenham

Per poder solucionar aquesta pràctica implementarè l'algorisme de Bresenham. A continuació explicarè en que consisteix i el codi de la meva solució.



És un algoritme per poder dibuixar línies gràfiques mitjançant càlculs amb enters, l'algoritme calcula quin és el següent pixel a dibuixar.
Quan dibuixem una linea en un mapa de pixels les lineas sempre tindran errors, excepte per les lineas horitzontals i verticals, amb aquest algoritme calcularem els errors per poder pintar sobre la pantalla els píxels que estan per damunt d'on passa la recta.

El pseudocodi per generar una linea és el següent:

1. Obtenim el punt inicial P1 (x1,y1) i P2(x2,y2).
2.Calcular DeltaX, DeltaY i constant P.
3.Calcular quin punt és inici o final.
4.Mentres X<Últim
5.Segons el paràmetre P, prenem una de les dos opcions:
    -Si P<0 → (X+1,Y)
    -Si P>=0 → (X+1,Y+1)
6.Pintem el pixel fins arriba al final del bucle.

Codi solució

```
int x, y, dx, dy;
int P, incE, incNE, stepx, stepy;

dx =x2 - x1;   //DeltaX
dy =y2 - y1;   //DeltaY

if (dx >= 0) { //Calculem sentit X
  stepx = 1;
}else{
  dx =x1-x2;
  stepx = -1;
}
if (dy >= 0) {  //Calculem sentit Y
  stepy = 1;
}else{
  dy =y1-y2;
  stepy = -1;
}
x = x1; //Punts inicials
y = y1;
drawPoint(x,y);       //Pintem primer punt de la recta

if(dx>dy){
  P=2*dy - dx;
  incE = 2*dy;
  incNE = 2*(dy-dx);
  while (x != x2){
     x = x + stepx; //Desplacem columna
     if (P < 0){   //Segons constanP pintem pixel (x+1,y) o (x+1,y+1)
        P = P + incE;
     }
     else {
        y = y + stepy; //Desplacem fila
        P=P+incNE;
     }
     drawPoint(x,y);
  }
}
else{
```

```
  P=2*dx-dy;
  incE=2*dx;
  incNE=2*(dx-dy);
  while (y!=y2){
     y=y+stepy;
     if (P<0){
        P=P+incE;
     }
     else {
        x=x+stepx;
        P=P+incNE;
     }
     drawPoint(x,y);
  }
}
```

Respostes

•Perquè en les implementacions de algorisme de Bresenham es fa una consulta sobre la pendent de la recta? Identifica com has fet aquesta consulta en el codi que has utilitzat i quin significat i raó de ser li dones.

Mitjançant la consulta sobre la recta podem obtenir quin serà el següent pixel a dibuixar en la recta, després de obtenir el pendent podem saber si el pixel que continuarà la línea serà el següent de la columna actual (x+1,y) o si és el següent de la columna i fila actual (x+1,y+1).

En el codi solució comprovem la pendent dintre del bucle per cada pixel que anem a dibuixar i obtenim els paràmetres x i y per la següent trucada al mètode drawPoint.

```
while (x != x2){
   x = x + stepx;
       if (P < 0){
                P = P + incE;
        }
        else {
                y = y + stepy; //Desplacem fila
                P=P+incNE;
       }
   drawPoint(x,y);
}
```

•Si ens demanen de pintar una recta vertical, podríem optimitzar l'algorisme d'alguna forma? Com ho faries? Ho té en compte codi que has fet?

Si ens demanen pintar una recta vertical la solució es simplifica notablement, ja que no tenim problemes per decidir quin és el següent pixel a pintar, només caldria detectar el sentit de la recta i fer un bucle mentres cridem el mètode drawPoint incrementan o decrementant la Y en cada trucada segons el sentit de la recta.

• Les mateixes preguntes si ens demanen de pintar una recta de pendent 1.

Si ens demanen pintar una recta de pendent 1, el problema també es simplifica notablement, només caldria fer un bucle on truquem al mètode drawPoint on en cada trucada incrementem la X i la Y en un pixel, (x+1,y+1).

• Quina és la principal diferència entre l'algorisme DDA i el Bresenham? Quin efecte té en els casos particulars de les dos preguntes anteriors?

La principal diferencia es que Bresenham és útil en la generació de rectes i corbes, en canvi el DDA només pot generar rectes.
Bresenham també és més ràpid que DDA ja que només utilitza sumes i DDA fa servir divisions, Bresenham és també molt més eficient quan ha de dibuixar les línies.

Joc de proves

En el joc de proves es dibuixaran les lines del programa i també totes les lines que complexin les seguents caracteristiques:

-Recta vertical on els vèrtexs s'indiquen de dalt a baix. **(200,300,200,0)**
-Recta vertical on els vèrtexs s'indiquen de baix a dalt. **(250,0,250,300)**

-Recta horitzontal on els vèrtexs s'indiquen d'esquerra a dreta. **(0,200,400,200)**
-Recta horitzontal on els vèrtexs s'indiquen de dreta a esquerra. **(400,250,0,250)**

-Recta amb pendent=1 on els vèrtexs s'indiquen d'esquerra a dreta. **(0,0,300,300)**
-Recta amb pendent = 1 on els vèrtexs s'indiquen de dreta a esquerra. **(300,300,600,0)**

-Recta amb pendent < 1 on els vèrtexs s'indiquen d'esquerra a dreta. **(300,300,700,0)**
-Recta amb pendent < 1 on els vèrtexs s'indiquen de dreta a esquerra. **(750,0,300,300)**

-Recta amb pendent > 1 on els vèrtexs s'indiquen d'esquerra a dreta.**(300,300,450,0)**
-Recta amb pendent > 1 on els vèrtexs s'indiquen de dreta a esquerra. **(500,0,300,300)**