

**Шулуун зурах алгоритм.**

Үргэлжилсэн шулууны налалт-огтлолын тэгшитгэл (slope-intercept equation):

$$y = mx + b \quad \text{энд } m = \text{шулууны налуу, } b = y\text{-огтлолцол.}$$

$(x_1, y_1)$  болон  $(x_2, y_2)$  цэгүүд өгөгдсөн бол

$$\text{slope (m)} = \frac{y_2 - y_1}{x_2 - x_1}$$

$$\therefore b = y - \frac{y_2 - y_1}{x_2 - x_1}x \quad \text{Дээрх тэгшитгэлээс жнь} \quad y = mx + b$$

Аль ч цэгт  $(x_k, y_k)$

$$y_k = mx_k + b \dots\dots\dots 1$$

$(x_{k+1}, y_{k+1})$  дээр,

$$y_{k+1} = mx_{k+1} + b \dots\dots\dots 2.$$

2-оос 1 хасвал бид дараах тэгшитгэлийг авна

$$y_{k+1} - y_k = m(x_{k+1} - x_k)$$

у;

Энд  $(y_{k+1} - y_k)$  нь х-ийн өсөлтөд харгалзах у-ийн өсөлт.

$$\therefore \Delta y = m \Delta x$$

$$\text{буюу } m \equiv \frac{\Delta y}{\Delta x}$$

Шулуун зурах өсөлтийн алгоритмын хувьд ,

- $x$ -ийг 1-ээр нэмэгдүүлнэ
- Computer corresponding  $y$  and display pixel at position  $(x_i, \text{round}(y_i))$

Асуудал: Хөвөгч таслалтай (бугархай) цэгийн үржвэр болон нэмэх

- round функц.

**DDA шулууны алгоритм:**

Тоон дифференциал задлан шинжлэгч (digital differential analyzer (DDA)) нь  $\Delta x$  эсвэл  $\Delta y$  тооцоололд суурилсан скан хөрвүүлэлт шулуун зурах алгоритм (scan conversion line drawing algorithm)-ын томъёо,

$$\Delta y = m \Delta x.$$

Бид шулууныг нэг координат дахь нэгж интервал дээр түүвэрлэлт хийж, бусад координатын хувьд шулуунтай ойр орших тоон утгуудыг тодорхойлно.

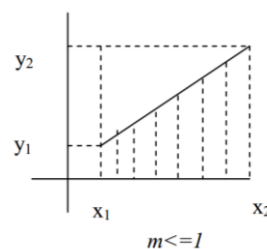
Эерэг налуутай шугамыг авч үзье.

Хэрэв  $m \leq 1$  бол, бид  $x$  координат түүвэрлэнэ.

Иймд  $\Delta x = 1$  ба дараалсан уутга бүрийг

тооцоолно:

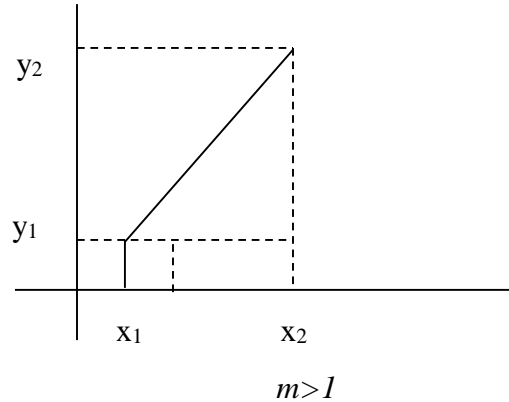
$$y_{k+1} = y_k + m \quad \therefore m = \frac{\Delta y}{\Delta x}, \Delta x = 1$$



Энд  $k$  эхлэлийн цэгээс эхэлж утга авах ба төгсөлийн цэг хүртэл 1-ээр өснө.  $m$  нь 0 ба 1 хоорондох бодит утга байж болно.

1-ээс их эерэг налуутай шулууны хувьд  $\Delta y = 1$  түүвэрлэн, харгалзах  $x$  утгыг дараах байдлаар тооцоолно

$$x_{k+1} = x_k + \frac{1}{m} \quad \because m = \frac{\Delta y}{\Delta x}, \Delta y = 1$$



Дээрх тэгшитгэлүүд нь шулууныг зүүнээс баруун тийш боловсруулах байдлаар таамагласан ба өөрөөр хэлбэр зүүн төгсгөлийн цэгээс эхэлнэ. Хэрэв боловсруулалт баруунаас зүүн тийш бол  $|m| < 1$  шулууны хувьд  $\Delta y = -1$  байна

$$\therefore y_{k+1} = y_k - m,$$

Хэрэв  $|m| > 1$ ,  $\Delta y = -1$

$$x_{k+1} = x_k - \frac{1}{m} \text{ гэж тооцоолно.}$$

### **DDA алгоритмд зориулсан C функц.**

```
void lineDDA (in x1, int y1, int x2, int y2)
{
    int dx, dy, steps, k;
    float incrx; incry; x,y;
    dx=x2-x1;
    dy=y2-y1;
    if (abs(dx)>abs(dy))
        steps=abs(dx);
    else
        steps=abs(dy);
    incrx=dx/steps;
    incry=dy/steps;
    x=x1; /* first point to plot */
    y=y1;
    putpixel(round(x), round(y),1); //1 is used for color
    for (k=1;k<=steps;k++)
    {
        x = x + incrx;
        y = y + incry;
        putpixel(round(x), round(y),1);
    }
}
```

DDA алгоритм нь пикселийн байрлалыг тооцоолоход илүү хурдтай арга боловч дараахь асуудал байна:

- $m$  нь бутархай тоогоор хадаглагдана.
- round алдаа үүсдэг
- шулууныг боловсруулах үед алдаа хуримтлагддаг.
- Иймд урт шулууны хувьд бодит шулууны замаас шулуун холдох болно

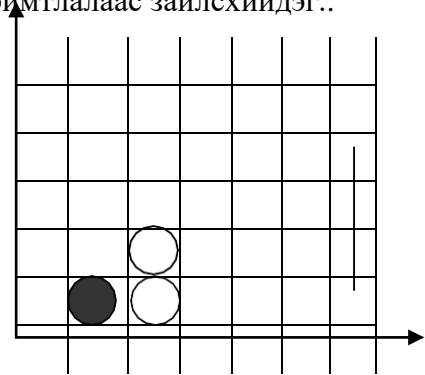
### Брезенхамын шулууны алгоритм (Bresenham's Line algorithm):

Брезенхамын хөгжүүлсэн үнэн зөв, үр ашигтай шулуун үүсгэх алгоритм нь scan converts lines нь дараагийн  $(x, y)$  байрлалыг тэмдэглэхдээ зөвхөн бүхэл тоон тооцоолол ашигладаг. Энэ нь өсөн нэмэгдэх алдааны хуримтлалаас зайлсхийдэг..

#### 1-ээс бага эерэг налуутай шулуун ( $0 < m < 1$ )

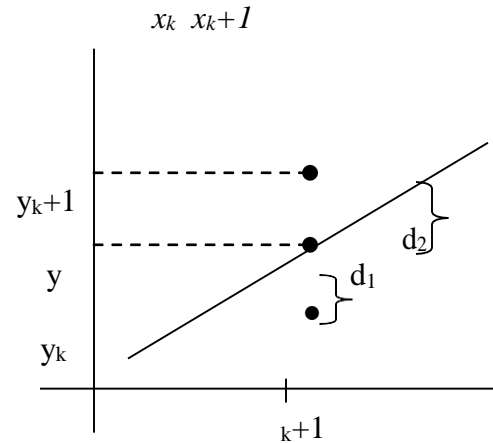
Шулууны дагуу орших цэгийн байршилүүд нь  $x$  нэгж интервалд түүвэрлэн тодорхойлно. Зүүн төгсгөлийн цэгээс эхлүүлэн бид дараалсан багана бүрт шулууны замд ойр цэгүүдийг тэмдэглэнэ.

$(x_k, y_k)$  нь  $k$  дахь алхам дээрх пиксел ба дараагийн тэмдэглэх цэг нь  $(x_k + 1, y_k)$  эсвэл  $(x_k + 1, y_k + 1)$  байна.



$x_k + 1$  түүврийн байршил дээр бид шулуунаас салаалж буй босоо тэнхлэгийн дагуух цэгүүдийг  $d_1$  &  $d_2$  гэж зурагт тэмдэглэсэн.

$x_k + 1$  пиксел баганын математик шулууны зам дээрх  $y$ -координат нь  $y = m(x_k + 1) + b$  байна.



Дараа нь  $d_1 = y - y_k$

$$= m(x_k + 1) + b - y_k$$

$$d_2 = (y_k + 1) - y$$

$$= (y_k + 1) - m(x_k + 1) - b$$

$$\text{Одоо } d_1 - d_2 = 2m(x_k + 1) - (y_k + 1) - y_k + 2b$$

$$= 2m(x_k + 1) - 2y_k + 2b - 1$$

Шулууны алгоритмын  $k$  дахь алхам дахь шийдвэрийн параметр  $p_k$ -г дараах байдлаар гарган авна:

Дээрх томъёонд тодорхойлсон  $m = \frac{\Delta x}{\Delta y}$  орлуулна.

$$\begin{aligned}
p_k &= \Delta x(d_1 - d_2) \\
&= \Delta x \left[ 2 \frac{\Delta y}{\Delta x} (x_k + 1) - 2y_k + 2b - 1 \right] \\
&= 2\Delta y \cdot x_k - 2\Delta x \cdot y_k + 2\Delta y + \Delta x(2b - 1) \\
&= 2\Delta y \cdot x_k - 2\Delta x \cdot y_k + c
\end{aligned}$$

Энд  $c$  тогтмол бие даасан тогтмол байна.

Хэрэв шийдвэрийн параметр  $p_k$  нь сөрөг бол өөрөөр хэлбэл  $d_1 < d_2$  бол  $y_k + 1$  дээрх пикселээс илүү  $y_k$  дээрх пиксель шулууны замд ойр байна. Энэ нөхцөлд доод пиксел (lower pixel) ( $x_k + 1, y_k$ ) тэмдэглэнэ. Эсрэг тохиолдолд дээд пиксел ( $x_k + 1, y_k + 1$ ) тэмдэглэнэ.

Шулууны дагуух координатын өөрчлөлт нь  $x$  эсвэл  $y$  чиглэлд нэгж алхамуудад явагдана. Тиймээс бид дараалсан шийдвэрийн параметрийн утгыг өсөн нэмэгдэх бүхэл тооцооллын арга ашиглан олж авах боломжтой.

$k+1$  алхам дахь  $p_{k+1}$  дараах байдлаар тооцоолно.

$$p_{k+1} = 2\Delta y \cdot x_{k+1} - 2\Delta x \cdot y_{k+1} + c$$

$$\therefore p_{k+1} - p_k = 2\Delta y(x_{k+1} - x_k) - 2\Delta x(y_{k+1} - y_k)$$

$x_{k+1} = x_k + 1$  болсоноор

$$\therefore p_{k+1} = p_k + 2\Delta y - 2\Delta x(y_{k+1} - y_k)$$

$y_{k+1} - y_k$  нь  $p_k$ -ийн тэмдгээс хамаарч 0 эсвэл 1 байна.

Анхны шийдвэрийн параметр  $p_0$  дараах байдлаар тооцоолох ба

$$p_0 = 2\Delta y - \Delta x$$

шийдвэрийн параметрийг дараалан дараах байдлаар тооцоолно

$$p_{k+1} = p_k + 2\Delta y - 2\Delta x(y_{k+1} - y_k)$$

Хэрэв  $p_k$  нь сөрөг бол  $y_{k+1} = y_k$  байх ба  $p_{k+1} = p_k + 2\Delta y$  болно.

Эсрэг тохиолдолд  $y_{k+1} = y_k + 1$  ба  $p_{k+1} = p_k + 2\Delta y - 2\Delta x$

### **Алгоритм:**

1. Шулууны хоёр төгсгөлийн цэг авах ба зүүн цэгийг ( $x_o, y_o$ ) байрлалд хадаглана.
2. ( $x_o, y_o$ ) frame buffer-т ачааллах ба өөрөөр хэлбэл эхний цэгийг тэмдэглэнэ.
3.  $\Delta x, \Delta y, 2\Delta x, 2\Delta y$  тогтмолуудыг тооцоолж, эхний шийдвэрийн параметрийн утгийг авна.

$$p_0 = 2\Delta y - \Delta x$$

4. Шулууны дагуух  $x_k$  бүрт  $k = 0$  эхлэн дараах туршилтыг гүйцэтгэнэ.

if  $p_k < 0$ , next point is ( $x_k + 1, y_k$ )

$$p_{k+1} = p_k + 2\Delta y$$

otherwise

next point to plot is ( $x_k + 1, y_k + 1$ )

$$p_{k+1} = p_k + 2\Delta y - 2\Delta x$$

5.  $\Delta x$  удаа 4 дэх алхамыг давтана.

**Функцийг C хэлэнд хэрэгжүүлсэн байдал**

```

void lineBresenham (int x1, int y1, int x2, int y2)
{
    int x, y, dx, dy, pk, k xEnd;
    dx=abs(x2-x1);
    dy=abs(y2-y1);
    if(x1>x2)
    {
        x = x2;
        y = y2;
        xEnd = x1;
    }
    else
    {
        x = x1;
        y = y1;
        xEnd = x2;
    }

    putixel (x,y,1);
    pk=2*dy-dx;
    while (x<=xEnd)
    {
        if(pk<0)
        {
            x=x+1;
            y=y;
            pk=pk+2*dy;
        }
        else
        {
            x=x+1;
            y=y+1;
            pk= pk+2*dy-2*dx
        }

        putpixel (x,y,1);
    }
}

```

Брензенхамын алгоритм нь ху хавтгайн өөр өөр октант ба квадрантуудын хоорондох тэгш хэмийг харгалзан дурын налуу шугамуудыг үүсгэдэг. 1 их эерэг налуу шугамын хувьд x болон y үүргийг сольж хэрэгжүүлдэг.

**Даалгавар:** Шулууныг зурах энгийн алгоритм болон Брензенхамын алгоритмыг хэрэгжүүлсэн програм бич. Хоорондын ялгааг тайлбарлана уу.