

一：直线的绘制(DDA 算法)

使用微分分析来绘制直线，这种算法很简单，就是算出直线的斜率，然后步进就行了。假设绘制两点 $A(x_1, y_1)$ ， $B(x_2, y_2)$ ，我们选择任意一点(假设为 A)作为起点，然后计算出斜率

$K = \frac{y_1 - y_2}{x_1 - x_2}$ ，假设当前起点为 x 值较小的那点 (x, y) ，则下一点 $(x+1, y')$ 中可以计算出 $y' = y + k$ ，也

就是在屏幕中绘制一系列这样的点 $\{(x, y), (x+1, y+k), (x+2, y+2k) \cdots \cdots (x+n, y+nk) | n = |x_2 - x_1|\}$ ， x 每次增加 1， y 每次比上一次增加 k 即可。执行下面的伪代码即可绘制出线段 AB：

```
Line(A, B)
{
    if (A.x >= B.x)
    {
        swap(A, B); //使A的x小于B的x
    }
    x=A.x;
    y=A.y;
    dy = (A.y-B.y)/(A.x-B.x);
    for (;x<B.x;x++)
    {
        setPixel(x, y, WHITE);
        y = y + dy;
    }
}
```

这样就可以绘制出一条线段。但是有一个问题，如果 $|k|$ 大于 1 的时候，绘制的线段就会断裂，如下图所示



线段不连续了，这是因为 $|k| > 1$ 。假设已经绘制了 (x, y) 点，假设 $k > 1$ ，则下一点应该是 $(x+1, y+2)$ 或者 $y+3$ 甚至更大，这样的话，就会出现不连续的情况。所以假如 $|k| > 1$ ，可以采用按照 y 步

进，而 $y' = y + \frac{1}{k}$ ，并且 $|\frac{1}{k}| < 1$ 。新绘制的线段就连续了：

