

案例 3 圆中点 Bresenham 算法

本案例知识要点

- 掌握八分法中点 Bresenham 算法绘制圆的原理。
- 设计八分法绘制圆的中点 Bresenham 算法。
- 编写八分法绘制圆的 CirclePoint(x,y)子函数。
- 编写绘制整圆的 Mbcircle()子函数。

一、案例需求

1. 案例描述

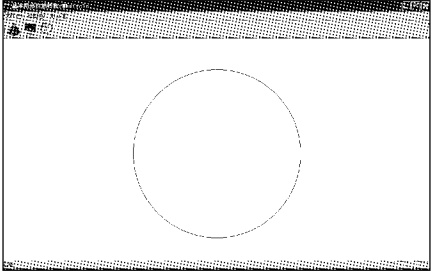
使用中点 Bresenham 算法绘制圆心位于屏幕客户区中心的圆。

2. 案例效果图

案例输入对话框及效果如图 3-1 所示。



(a) 输入对话框



(b) 效果

图 3-1 案例 3 输入对话框及效果

3. 功能说明

- (1) 要求使用对话框输入圆的半径。
- (2) 圆的颜色为蓝色。

二、算法设计

- (1) 输入圆的半径 R 。
- (2) 定义圆当前点坐标 x 和 y , 定义中点偏差判别式 d , 定义像素点颜色 rgb 。
- (3) 计算 $d=1.25-R, x=0, y=R, rgb=RGB(0,0,255)$ 。
- (4) 绘制点 (x,y) 及其在八分圆中的另外 7 个对称点。
- (5) 判断 d 的符号。若 $d<0$, 则 (x,y) 更新为 $(x+1,y)$, d 更新为 $d+2x+3$; 否则 (x,y) 更新为 $(x+1,y-1)$, d 更新为 $d+2(x-y)+5$ 。
- (6) 当 x 小于等于 y , 重复步骤(4)和(5), 否则结束。

三、案例实现

1. TestView.h 文件

代码如下:

```
// TestView.h : interface of the CTestView class
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
#ifdef _AFX_TESTVIEW_H__A75FDCFB_621C_4E38_A154_C344803E6372__INCLUDED_
#define AFX_TESTVIEW_H__A75FDCFB_621C_4E38_A154_C344803E6372__INCLUDED_
#endif _MSC_VER>1000
#pragma once
#endif // _MSC_VER>1000
#include "InputDlg.h" //包含对话框头文件
class CTestView : public CView
{
protected: // create from serialization only
    CTestView();
    DECLARE_DYNCREATE(CTestView)
// Attributes
public:
    CTestDoc * GetDocument();
// Operations
public:
    void GetMaxX(); //获得屏幕的最大x值函数
    void GetMaxY(); //获得屏幕的最大y值函数
    void CirclePoint(double x,double y); //八分法画圆子函数
    void Mbcircle(); //圆中点 Bresenham算法
// Overrides
    // ClassWizard generated virtual function overrides
    //{AFX_VIRTUAL(CTestView)
public:
    virtual void OnDraw(CDC * pDC); //overridden to draw this view
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
protected:
    virtual BOOL OnPreparePrinting(CPrintInfo * pInfo);
    virtual void OnBeginPrinting(CDC * pDC,CPrintInfo * pInfo);
    virtual void OnEndPrinting(CDC * pDC,CPrintInfo * pInfo);
    }AFX_VIRTUAL
// Implementation
public:
    virtual ~CTestView();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif
}
```

protected:

```
int MaxX,MaxY;           //屏幕 x 和 y 的最大坐标
double R;                //圆的半径
```

// Generated message map functions

protected:

```
//{{AFX_MSG(CTestView)
afx_msg void OnMENU_Mbcircle();
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
```

};

#ifndef _DEBUG // debug version in TestView.cpp

inline CTestDoc* CTestView::GetDocument()

```
{ return (CTestDoc*)m_pDocument;}
```

#endif

////////////////////////////////////

//{{AFX_INSERT_LOCATION}}

// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif // !defined(AFX_TESTVIEW_H_A75FDCFB_621C_4E38_A154_C344803E6372__INCLUDED_)

2. TestView. cpp 文件

代码如下:

// TestView.cpp : implementation of the CTestView class

#include "stdafx.h"

#include "Test.h"

#include "TestDoc.h"

#include "TestView.h"

```
#define ROUND(a) int(a+0.5) //四舍五入
```

#ifdef _DEBUG

#define new DEBUG_NEW

#undef THIS_FILE

static char THIS_FILE[]=__FILE__;

#endif

////////////////////////////////////

// CTestView

IMPLEMENT_DYNCREATE(CTestView,CView)

BEGIN_MESSAGE_MAP(CTestView,CView)

```
//{{AFX_MSG_MAP(CTestView)
```

```
ON_COMMAND(ID_MENU_Mbcircle,OnMENU_Mbcircle)
```

```
//}}AFX_MSG_MAP
```

// Standard printing commands

```
ON_COMMAND(ID_FILE_PRINT,CView::OnFilePrint)
```

```
ON_COMMAND(ID_FILE_PRINT_DIRECT,CView::OnFilePrint)
```

```
ON_COMMAND(ID_FILE_PRINT_PREVIEW,CView::OnFilePrintPreview)
```

END_MESSAGE_MAP()

```

////////////////////////////////////
// CTestView construction/destruction
CTestView::CTestView()
{
    // TODO: add construction code here
}
CTestView::~CTestView()
{
}
BOOL CTestView::PreCreateWindow(CREATESTRUCT& cs)
{
    // TODO: Modify the Window class or styles here by modifying
    // the CREATESTRUCT cs
    return CView::PreCreateWindow(cs);
}
////////////////////////////////////
// CTestView drawing
void CTestView::OnDraw(CDC * pDC)
{
    CTestDoc * pDoc=GetDocument();
    ASSERT_VALID(pDoc);
    // TODO: add draw code for native data here
}
////////////////////////////////////
// CTestView printing
BOOL CTestView::OnPreparePrinting(CPrintInfo * pInfo)
{
    // default preparation
    return DoPreparePrinting(pInfo);
}
void CTestView::OnBeginPrinting(CDC * /* pDC */, CPrintInfo * /* pInfo */)
{
    // TODO: add extra initialization before printing
}
void CTestView::OnEndPrinting(CDC * /* pDC */, CPrintInfo * /* pInfo */)
{
    // TODO: add cleanup after printing
}
////////////////////////////////////
// CTestView diagnostics
#ifdef _DEBUG
void CTestView::AssertValid() const
{
    CView::AssertValid();
}
}

```

```

void CTestView::Dump(CDumpContext& dc) const
{
    CView::Dump(dc);
}

CTestDoc* CTestView::GetDocument() // non-debug version is inline
{
    ASSERT(m_pDocument->IsKindOf(RUNTIME_CLASS(CTestDoc)));
    return (CTestDoc*)m_pDocument;
}

#endif // _DEBUG

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CTestView message handlers

void CTestView::GetMaxX() //得到客户区的最大横坐标
{
    CRect Rect;
    GetClientRect(&Rect);
    MaxX=Rect.right;
}

void CTestView::GetMaxY() //得到客户区最大纵坐标
{
    CRect Rect;
    GetClientRect(&Rect);
    MaxY=Rect.bottom;
}

void CTestView::Mbcircle() //圆中点 Bresenham算法
{
    double x,y,d;
    d=1.25-R;x=0;y=R;
    for(x=0;x<y;x++)
    {
        CirclePoint(x,y); //调用八分法画圆子函数
        if (d<0)
            d+=2*x+3;
        else
        {
            d+=2*(x-y)+5;
            y--;
        }
    }
}

void CTestView::CirclePoint(double x,double y) //八分法画圆子函数
{
    CClientDC dc(this);
    COLORREF rgb=RGB(0,0,255); //定义圆的颜色

```

```

dc.SetPixel(ROUND(x)+MaxX/2,ROUND(y)+MaxY/2,rgb); //x,y
dc.SetPixel(ROUND(y)+MaxX/2,ROUND(x)+MaxY/2,rgb); //y,x
dc.SetPixel(-ROUND(y)+MaxX/2,ROUND(x)+MaxY/2,rgb); //y,-x
dc.SetPixel(ROUND(x)+MaxX/2,-ROUND(y)+MaxY/2,rgb); //x,-y
dc.SetPixel(-ROUND(x)+MaxX/2,-ROUND(y)+MaxY/2,rgb); //-x,-y
dc.SetPixel(-ROUND(y)+MaxX/2,-ROUND(x)+MaxY/2,rgb); //-y,-x
dc.SetPixel(ROUND(y)+MaxX/2,-ROUND(x)+MaxY/2,rgb); //-y,x
dc.SetPixel(-ROUND(x)+MaxX/2,ROUND(y)+MaxY/2,rgb); //-x,y
}

void CTestView::OnMENU_Mbcircle() //菜单函数
{
    // TODO: Add your command handler code here
    InputDlg dlg;
    if(dlg.DoModal() != IDOK)
    {
        R=dlg.m_R;
    }
    AfxGetMainWnd()->SetWindowText("基本图形扫描转换:Mbcircle");
    RedrawWindow();
    GetMaxX();GetMaxY();
    Mbcircle();
}

```

四、案例说明

因为设备坐标系的原点在屏幕左上角,所以为了在屏幕中心绘制圆,定义了 GetMaxX() 和 GetMaxY() 函数来获得屏幕的 MaxX 和 MaxY,参见案例 1 的实现部分。在 SetPixel 函数中使用 MaxX/2 和 MaxY/2 来获得屏幕中心。