计算机图形学是利用计算机生成、处理和显示图形的学科，那么具体怎么进行计算机绘图呢？最简单的就是利用现成的绘图软件，比如计算机自带的绘图，Photoshop,3d Max等；第二种是使用专用的绘图语言和函数，如OpenGL等；除了专门的软件和专门语言外，很多软件本身也提供了简单的绘图功能，比如说VC++也有很多绘图函数，MFC的CDC类就封装了绘图函数，利用CDC类实现是计算机绘图的第三种方式；最后一种，Win32应用程序。

以下介绍利用MFC的CDC类常用的画图方法。

利用MFCAppWizard 建立单文档工程文件，左边有文件组织列表窗口，分别为源文件、头文件、资源文件。展开源文件文件夹，双击所建工程名对应的CexampleView.cpp,找到Ondraw函数，

1. **划线、划圆**

在其中添加代码pDC->LineTo(200,100)（默认坐标左上角0,0）pDC->MoveTo（）表示把焦点移到某点。pDC->Ellipse(30,20,300,200)表示椭圆外接矩形的左上为(30,20),右下为(300,200).

**２．划点函数**

pDC->SetPixel(i,j,RGB(0,0,0))

RGB每个分量的取值范围为0-255，r为255表示最红的红色成分，绿蓝也是如此。(255,255,255)表示白色。

**３．动画**

for(int i=0;i<628;i++)

{

int y=100\*sin(float(i)/100);//628/100约等于２pai

pDC->SetPixel(i,y+120,0); //函数中也可以用一个长整型表示颜色，0为黑色。最大16777215(2的24次方减一，24位真彩色)

Sleep(10); //#include<windows.h>

}

矩形

for(int i=5;i<60;i=i+5)

{

pDC->Rectangle(i,i,100-i,100-i);

Sleep(500);

}

**4．真实感**

void paintBall(int x,int y,int r, CDC \*pDC)

{

double w, u,d;

int R=0, G=0, B=0;

w=r;d=w/255;

while(w>=0)

{

for(u=0;u<628;u=u+1)

{

pDC->SetPixel((int)(x+w\*cos(u/100)),(int)(y+w\*sin(u/100)),RGB(R,G,B));

}

R++;G++;B++;

w=w-d;

}

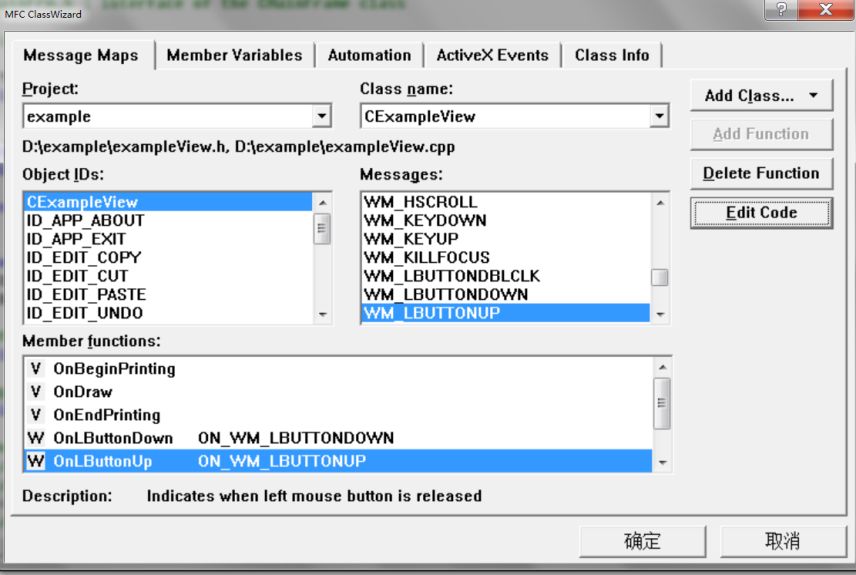
}

在ondraw函数中调用此函数。

U的取值范围正确吗？

**5．交互式**

查看菜单，建立类向导



void CExampleView::OnLButtonDown(UINT nFlags, CPoint point)

{

// TODO: Add your message handler code here and/or call default

m\_StartPoint=point;

CView::OnLButtonDown(nFlags, point);

}

void CExampleView::OnLButtonUp(UINT nFlags, CPoint point)

{

// TODO: Add your message handler code here and/or call default

CClientDC dc(this); //由CDC类派生出来的

dc.MoveTo(m\_StartPoint);

dc.LineTo(point);

CView::OnLButtonUp(nFlags, point);

}

在文件exampleView.cpp的前部添加ＣPoint类型的变量定义

CPoint m\_lastEndPoint;

CPoint m\_StartPoint;

**6．画笔与画刷**

小球下落1

pDC->MoveTo(100,10);

int k=1;

for(int i=10;i<250;i=i+k\*4.9)

{

pDC->Ellipse(100,10+i,150,60+i);

Sleep(100);

k++;

}

小球下落2（对比二者不同）

CDC \*pDC=GetDC();

CPen P1,P2;

P1.CreatePen(1,1,RGB(0,0,0));

P2.CreatePen(1,1,RGB(255,255,255));

pDC->MoveTo(100,10);

int k=1;

for(int i=10;i<250;i=i+k\*4.9)

{

pDC->SelectObject(&P1);

pDC->Ellipse(100,10+i,150,60+i);

Sleep(100);

pDC->SelectObject(&P2);

pDC->Ellipse(100,10+i,150,60+i);

k++;

}