OWL2.0 下妙用鼠标绘图

罗汉军　(华中理工大学　430074)

一、引言

Borland C + + 4.0/4.5 功能强大是众所周知的,但关于其编程的参考书中很难找到富有特色的绘图例程。要充分发挥其功能,必需潜心地研究其类库。

我们经常看到画直线或圆时可以用鼠标拉着走,但是这一般只针对用圆心,半径画圆,而针对于不规则画圆或圆弧则难以实现,笔者在开发软件的过程中,对 Borland C + + 4.5 的 OWL2.0 下的 TDC 类作了一个较深入的分析,终于成功地实现了动态三点画圆和三点画圆弧.特在此公布,以便与广大读者共享,以达到抛砖引玉的目的。

二、具体实现

其关键在于巧妙构造鼠标响应函数,尤其要巧妙组建响应鼠标移动的函数,并充分利用 OWL2.0 中 TDC 类的 SetROP2 (R2-NOT)函数的反极作用,先产生两个对象 1, 对象 2, 并精心安排对象的显示顺序,使对象 2 总是覆盖对象 1,从而实现动态三点画圆。

下面只就在 MDI 子窗口如何实现动态三点画圆作一个简单的介绍:

1.定义 TDrawMDIChild 类

```
class TDawMDIChild :public TMDIChild
{ public:
  TDrawMDIChild(TDrawMDIClient& parent, const char * title = 0);
  TDrawMDIChild()
  {
    delete dc;
    delete Pen;
  }
protected:
  void EvLButtonDown(UINT, TPoint&);
  void EvMouseMove(UINT, TPoint&);
protected:
  TDrawDC * dc; //TDrawDC 是 TClientDC 的派生类
  TPen * Pen;
  TPoint first,second; //第一点, 第二点
  TPoint center1,center2; //对象 1, 对象 2 的圆心
  int ClickNum; //鼠标点按次数
  int r1,r2; //对象 1, 对象 2 的半径
  }
```

2.声明子窗户的响应列表如下

```
DEFINE-RESPONSE-TABLE1 ( TDrawMDIChild, TMDIChild)
  EV-WM-LBUTTONDOWN,
  EV-WM-RBUTTONDOWN,
  EV-WM-MOUSEMOVE,
END-RESPONSE-TABLE;
```

3.定义构造函数和鼠标响应函数

```
TDrawMDIChild:: TDrawMDIChild ( TDrawMDIClient& parent, const char * title) :
  TMDIChild(parent, title)
  {
  Attr.Style | = WS-HSCROLL | WS-VSCROLL;
  dc = 0;
  Pen = 0;
  ClickNum = 0;
  }
  void
  TDrawMDIChild::EvLButtonDown(UINT, TPoint& point)
  {
  dc = new TDrawDC( * this);
  Pen = new TPen(TColor(100,100,0),1);
  dc->SelectObject( * Pen);
  dc->SetROP2(R2-NOT);
  dc -> SelectStockObject(NULL-BRUSH);

  switch( ClickNum )
  {
  case 0:
    ClickNum + + ;
    first = point;
  break;
  case 1:
  ClickNum + + ;
  second = point;
  center1.x = center2.x = (first.x + second.x)/2;
  center1.y = center2.y = (first.y + second.y)/2;
  r1 = r2 = (int)sqrt((long)(first.x - second.x) * (first.x - second.x) +
      (long)(first.y - second.y) * (first.y - second.y));
```

```
        dc->Ellipse(center2.x-r2,
                center2.y-r2,
                center2.x+r2,
                center2.y+r2); //用椭圆函数画圆
        center1 = center2;
        r1 = r2;
        break;
    case 2:
        ClickNum = 0;
        dc -> SetROP2( R2-COPYPEN );
        dc->Ellipse(center2.x-r1,
                center2.y-r1,
                center2.x+r1,
                center2.y+r1);
        delete dc;
        dc = 0;
        }
}
    void
    TDrawMDIChild::EvMouseMove(UINT, TPoint& point)
    {
    float k1,k2;
    float x,y;
    float radius;
    if( ClickNum == 2 )
    {
    if(first.y != second.y)
    {
        if(first.y != point.y)
        {
        k1 = (second.x-first.x)/(first.y-second.y);
        k2 = (point.x-first.x)/(first.y-point.y);
            if(k1 != k2)
            {
        x = (second.y-point.y+k2*(first.x+point.x)
            -k1*(first.x+second.x))/(2.0*(k2-k1));
            y = (first.y+second.y)/2.0+k1*(x-(first.x+sec
    ond.x)/2.0);
            }
        else
            {
            x = (first.x+second.x)/2.0;
            y = (first.y+second.y)/2.0;
            }
        }
    else
        {
        k1 = (second.x-first.x)/(first.y-second.y);
            x = (first.x+point.x)/2.0;
            y = (first.y+second.y)/2.0+k1*(x-(first.x+sec
    ond.x)/2.0);
            }
    }
    else
    {
        if(first.y != point.y)
        {
            k2 = (point.x-first.x)/(first.y-point.y);
            x = (first.x+second.x)/2.0;
            y = (first.y+point.y)/2.0+k2*(x-(first.x+
    point.x)/2.0);
            }
        else
        {
            x = (first.x+second.x)/2.0;
            y = (first.y+second.y)/2.0;
            }
    }
    radius = sqrt((double)(first.x-x)*(first.x-x)+
            (double)(first.y-y)*(first.y-y));
    r2 = (int)radius;
    center2.x = (int)x;
    center2.y = (int)y;
    dc->Ellipse(center1.x-r1,
            center1.y-r1,
            center1.x+r1,
            center1.y+r1); //画对象 1
        dc->Ellipse(center2.x-r2,
            center2.y-r2,
            center2.x+r2,
            center2.y+r2); //画对象 2 以覆盖对象 1
        center1 = center2;
        r1 = r2;
    }
```

(来稿时间:1996 年 11 月)