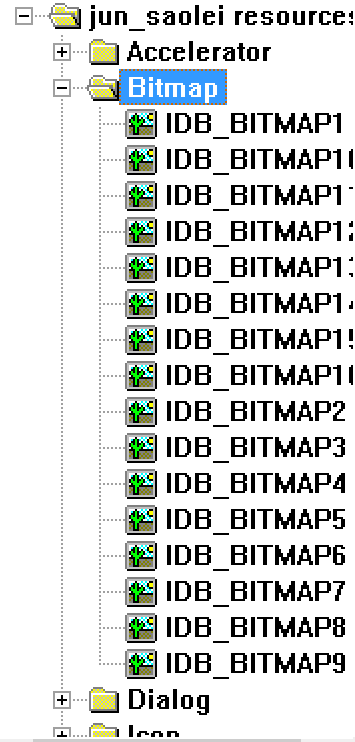
**扫雷实验报告**

**1400012799 吴文俊**

**亮点：**

1. **利用位图，减少了程序画图的繁琐过程。**
2. **定义全局变量weitu,geshu,有利于函数的编写。同时，个数-1表示有雷，正数表示周围的雷数，0.用一个变量表示多个含义，减少了程序变量的使用。**
3. **采用1秒内多次扫描非雷区，代替了递归函数**
4. **依据人体原理，通过方块上边左边画白线，下边右边画黑线产生立体感。**
5. **通过zhong,gao函数修改全局变量，起到修改函数难度的作用。**

**定义新类class Mine**

{

public:

int weitu;

int geshu;

};

**插入位图**

**便于后期的使用**

**程序中包含的函数:**

**int CMainFrame::OnCreate**

**(LPCREATESTRUCT lpCreateStruct)**

功能：调整系统自带的框架

设计思路：去除相应的函数

**BOOL CMainFrame::PreCreateWindow(CREATESTRUCT& c)**

功能：设置窗口大小

**CJun\_saoleiView::CJun\_saoleiView()**

{

// TODO: add construction code here

for ( int mi = 0 ; mi < 12 ; mi ++ )

m\_Bitmap[mi].LoadBitmap(IDB\_BITMAP1 + mi);//插入位图

for ( int vj = 0 ; vj < 4 ; vj ++)

m\_Button[vj].LoadBitmap( IDB\_BITMAP13 + vj);

sTime = 0;//经过的时间

timeStart = 0;//每20次算一秒

mRow = 25;//设置地雷的行数

mCol = 16;//设置地雷的列数

mineNum = 10;//默认状态为初级10颗雷

leftNum = mineNum;//剩余的雷赋初始值

isFinish = false;//记录是否碰到地雷

int ma = 0 ;

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

for( int j = 0 ; j < mCol ; j ++ )

{

mine[i][j].geshu = 0 ;

mine[i][j].weitu = 0 ;

}//雷区初始化

CTime time = GetCurrentTime();

int s = time.GetSecond(); //获得时间

srand(s);//以当前的时间为种子，避免了产生相同的随机数

while(ma < mineNum)//放置50个雷

{

int k = (int)( mRow \* rand() /(RAND\_MAX + 1.0)) ;//随机数产生行

int l = (int)( mCol \* rand() /(RAND\_MAX + 1.0)) ;

if( mine[k][l].geshu != -1 )//如果该地方已近是雷则跳到下一个地方

{

mine[k][l].geshu = -1;

ma ++ ;//记录已近布置的雷数

}

}

for( int a = 0 ; a < mRow ; a ++ ) //寻找边上雷数，并给非雷的方块赋值

for( int b = 0 ; b < mCol ; b ++)

if( mine[a][b].geshu == 0)

{

for( int c = a - 1 ; c <= a + 1 ; c ++ )

for ( int d = b - 1 ; d <= b + 1 ; d ++)

if( c >= 0 && c < mRow && d >= 0 && d < mCol)

{//判断是否在雷区内部

if( mine[c][d].geshu == -1)

mine[a][b].geshu ++ ;

}

}

}

**void CJun\_saoleiView::OnDraw(CDC\* pDC)**

{

CJun\_saoleiDoc\* pDoc = GetDocument();

ASSERT\_VALID(pDoc);

// TODO: add draw code for native data here

CBrush mbrush1;//添加刷子画背景颜色

mbrush1.CreateSolidBrush(RGB(100,130,130));//方块原始的颜色为青色

CRect mrect1( 0 , 0 , 1200 , 400 );

pDC->FillRect( mrect1 , &mbrush1 );//填充背景颜色

CBrush mbrush2;

mbrush2.CreateSolidBrush( RGB(0, 0 , 0)) ;

CRect mrect2(20, 10 , 70 , 40);//为左右显示数字栏填充黑色，为凹陷感做准备

pDC->FillRect(mrect2 , &mbrush2);

CRect mrect3(325 , 10 , 375 ,40);

pDC->FillRect(mrect3 , &mbrush2);

CPen mpen;

CPen \*oldPen;

mpen.CreatePen(PS\_SOLID , 2 , RGB(255 ,255,255));//添加白色笔

oldPen = pDC->SelectObject(&mpen);//为复原做准备

pDC->MoveTo( 20 , 40 );//在数字栏下边和右边画白线，产生凹陷感

pDC->LineTo( 70 , 40 );

pDC->LineTo( 70 , 10 );

pDC->MoveTo( 325 , 40 );

pDC->LineTo( 375 , 40 );

pDC->LineTo( 375 , 10 );

for( int i = 0 ; i < mRow ; i ++ )//方块左边上边画白线

for ( int j = 0 ; j < mCol ; j ++ )

{

pDC->MoveTo( 10 + i\*15 , 64 + j\*15 );

pDC->LineTo( 10 + i\*15 , 50 + j\*15 );

pDC->LineTo( 24 + i\*15 , 50 + j\*15 );

}

pDC->SelectObject(oldPen) ;//恢复原来的画笔

CPen mpen2;

CPen \*oldPen2;

mpen2.CreatePen(PS\_SOLID , 2 , RGB(0,0,0));//引入黑笔

oldPen2 = pDC->SelectObject(&mpen2);

for( int mi = 0 ; mi < mRow ; mi ++ )

for( int vj = 0 ; vj < mCol ; vj ++ )

{

pDC->MoveTo(10 + mi\*15 , 50+ vj\*15 + 14 );

pDC->LineTo(24 + mi\*15 , 64 + vj\*15 );

pDC->LineTo(24 + mi\*15 , 50 + vj\*15 );

}

pDC->SelectObject(oldPen2);//在方块右边下边画黑线

CDC Dc;

if(Dc.CreateCompatibleDC(pDC)==FALSE)

AfxMessageBox("Can't create DC");

Dc.SelectObject( m\_Button[0] );//开始笑脸

pDC->BitBlt( 180 , 10 , 40 , 40 , &Dc , 0 , 0 , SRCCOPY );

for( int a = 0 ; a < mRow ; a ++ )

for( int b = 0 ; b < mCol ; b ++ )

{

if( mine[a][b].weitu == 1 )//按下并显示数字

{

Dc.SelectObject(m\_Bitmap[mine[a][b].geshu] );//选入相应的位图

pDC->BitBlt( a\*15 + 10 , b\*15 + 50 , 40 , 40 ,&Dc , 0 , 0, SRCCOPY );

}

if( mine[a][b].weitu == 2 )//显示红旗

{

Dc.SelectObject(m\_Bitmap[9] );

pDC->BitBlt( a\*15 + 10 , b\*15 + 50 , 40 , 40 ,&Dc , 0 , 0, SRCCOPY );

}

if( mine[a][b].weitu == 3 )//显示问好

{

Dc.SelectObject(m\_Bitmap[10] );

pDC->BitBlt( a\*15 + 10 , b\*15 + 50 , 40 , 40 ,&Dc , 0 , 0, SRCCOPY );

}

if( isFinish == true && mine[a][b].geshu == -1 )//如果踩到地雷，则显示地雷

{

Dc.SelectObject(m\_Bitmap[11] );

pDC->BitBlt( a\*15 + 10 , b\*15 + 50 , 40 , 40 ,&Dc , 0 , 0, SRCCOPY );

Dc.SelectObject(m\_Button[3] );//显示哭脸

pDC->BitBlt( 180 , 10 , 40 , 40 ,&Dc , 0 , 0, SRCCOPY );

}

}

int ioldDC = pDC->SaveDC();//记录原始

pDC->SetTextColor(RGB(255,0,0));//设置文字的颜色

pDC->SetBkColor(RGB(0,0,0));

CFont font ;//插入文字

if(0==font.CreatePointFont(160,"Comic Sans MS")) //设置文字插入的地方

AfxMessageBox("Can't Create Font");

pDC->SelectObject(&font);

CString str ;//补充前面空余的0

if( leftNum < 10 )

str.Format( "00%d" , leftNum );

else

str.Format( "0%d" , leftNum );

pDC->TextOut(25,10,str);

if( sTime < 10 )

str.Format("00%d" , sTime);

else if( sTime < 100 )

str.Format("0%d" , sTime );

else

str.Format("%d" , sTime );

pDC->TextOut(330,10, str);

pDC->RestoreDC( ioldDC );//复原

**}**

**void CJun\_saoleiView::mineZero()**

{

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

for( int j = 0 ; j < mCol ; j ++ )

if( mine[i][j].geshu == 0 && mine[i][j].weitu == 1 )//判断是否无雷且按下

{

for( int n = i - 1 ; n <= i + 1 ; n ++ )

for( int m = j - 1 ; m <= j + 1 ; m ++ )

if( n >= 0 && n < mRow && m >= 0 && m < mCol)

if( mine[n][m].geshu != -1 && mine[n][m].weitu == 0 )

{

mine[n][m].weitu = 1;//展开无雷的地方

CRect rect;

rect.left = n\*15 + 10 ;

rect.right = n\*15 + 25 ;

rect.top = m\*15 + 50;

rect.bottom = m\*15 + 65 ;

InvalidateRect( &rect);//部分方格重画

}

}

}

**void CJun\_saoleiView::OnTimer(UINT nIDEvent)**

{

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

if( isFinish) return ;

mineZero(); // 显示周边非雷区

if( timeStart > 0 )

timeStart ++ ;

if( timeStart == 20 )

{

timeStart = 1;

sTime ++;

CRect rect3;

rect3.left = 325 ;

rect3.right = 375;

rect3.top = 10 ;

rect3.bottom = 40 ;

InvalidateRect( &rect3) ;//计算经过的时间，并显示出来

}

CView::OnTimer(nIDEvent);

}

**void CJun\_saoleiView::OnRButtonDown(UINT nFlags, CPoint point)**

{

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

if( isFinish ) return ;

if( point.x > 10 && point.x < 385 && point.y > 50 && point.y < 290 )//判断鼠标是否在正确的地方按下

{

int a = (point.x - 10) / 15 ;//转换鼠标坐标为方块的位子

int b = (point.y - 50) / 15 ;

if( mine[a][b].weitu == 0 || mine[a][b].weitu == 3 )//当前为未被按下或是？

{

mine[a][b].weitu = 2 ;

leftNum -- ;//剩余雷数减1

}

else if(mine[a][b].weitu == 2)//当前为红旗

{

mine[a][b].weitu = 3 ;

leftNum ++ ;

}

CRect rect1 ;

rect1.left = 20 ;

rect1.right = 70 ;

rect1.top = 10 ;

rect1.bottom = 40 ;

InvalidateRect( &rect1);//显示剩余雷数处

CRect rect2 ;

rect2.left = a\*15 + 10 ;

rect2.right = a\*15 + 25 ;

rect2.top = b\*15 + 50 ;

rect2.bottom = b\*15 + 65 ;

InvalidateRect( &rect2);//显示按下的地方

}

CView::OnRButtonDown(nFlags, point);

}

**void CJun\_saoleiView::OnLButtonDown(UINT nFlags, CPoint point)**

{

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

CDC \*pDC = GetDC();

CDC Dc;

if( Dc.CreateCompatibleDC(pDC) == FALSE )

AfxMessageBox( "error");

if( point.x > 180 && point.x < 210 && point.y > 10 && point.y < 40 )

{

Dc.SelectObject( m\_Button[3] );//开始处闪一下，换一张图片

pDC->BitBlt( 180,10, 160 , 160 , &Dc , 0 , 0 , SRCCOPY);

}

if( point.x >= 10 && point.x <= 385 && point.y >= 50 && point.y <= 290)

{

if( isFinish)

return ;

Dc.SelectObject(m\_Button[1]);//失败时显示哭脸

pDC->BitBlt( 180 , 10 , 160 , 160 , &Dc , 0 , 0 , SRCCOPY);

timeStart = 1 ;//开始循环

int a = (point.x - 10 ) / 15 ;//鼠标定位

int b = (point.y - 50 ) / 15 ;

if( mine[a][b].weitu == 0 || mine[a][b].weitu == 3)

{

if( mine[a][b].geshu == -1 )

{

isFinish = true ;

KillTimer(1) ;

Invalidate();//重画

}

else

{

mine[a][b].weitu = 1 ;

CRect rect ;

rect.left = a\*15 + 10 ;

rect.right = a\*15 + 25 ;

rect.top = b\*15 + 50 ;

rect.bottom = b\*15 + 65 ;

InvalidateRect( &rect) ;//重画当前方块

}

}

}

CView::OnLButtonDown(nFlags, point);

}

**void CJun\_saoleiView::OnLButtonUp(UINT nFlags, CPoint point)**

{

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

CDC \*pDC = GetDC();

CDC Dc ;

if( Dc.CreateCompatibleDC( pDC) == FALSE)

AfxMessageBox("Can't create DC");

Dc.SelectObject( m\_Button[0]);//显示开始

pDC->BitBlt( 180 , 10 , 160 , 160, &Dc , 0 , 0 , SRCCOPY);

if( isFinish )

{

Dc.SelectObject( m\_Button[2] );//显示结束

pDC->BitBlt( 180 , 10 , 160 , 160 , &Dc , 0 , 0 , SRCCOPY );

}

if( point.x > 180 && point.x< 210 && point.y > 10 && point.y < 40 )

OnStart();

CView::OnLButtonUp(nFlags, point);

}

**int CJun\_saoleiView::OnCreate(LPCREATESTRUCT lpCreateStruct)**

{

if (CView::OnCreate(lpCreateStruct) == -1)

return -1;

// TODO: Add your specialized creation code here

SetTimer( 1 , 50 , NULL ); //每一秒循环20次

return 0;

}

void CJun\_saoleiView::OnStart()

{

SetTimer( 1, 50 , NULL);

sTime = 0 ;

timeStart = 0 ;

leftNum = mineNum ;

isFinish = false ;

int ma = 0;

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

for( int j = 0 ; j < mCol ; j ++ )

{

mine[i][j].geshu = 0 ;

mine[i][j].weitu = 0 ;

}

while( ma < mineNum )

{

int k = (int)( mRow \* rand() / (RAND\_MAX + 1.0)) ;

int l = (int)( mCol \* rand() / (RAND\_MAX + 1.0)) ;

if( mine[k][l].geshu != -1 )

{

mine[k][l].geshu = -1 ;

ma ++ ;

}

}

for ( int a = 0 ; a < mRow ; a ++ )

for ( int b = 0 ; b < mCol ; b ++ )

if( mine[a][b].geshu == 0 )

{

for( int c = a - 1 ; c <= a + 1 ; c ++ )

for( int d = b - 1 ; d <= b + 1 ; d ++ )

if( c >= 0 && d >= 0 && c < mRow && d < mCol )

if( mine[c][d].geshu == -1 )

mine[a][b].geshu ++ ;

}

Invalidate();

}

**void CJun\_saoleiView::Onzhong()** //设置中级难度

{

// TODO: Add your command handler code here

mineNum = 40;

}

**void CJun\_saoleiView::OnGao()** //设置高级难度

{

// TODO: Add your command handler code here

mineNum = 100;

}