

```

// SquaresView.cpp : implementation of the CSquaresView class
//

#include "stdafx.h"
#include "SdiSquares.h"

#include "SquaresDoc.h"
#include "SquaresView.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////
// CSquaresView

IMPLEMENT_DYNCREATE(CSquaresView, CView)

BEGIN_MESSAGE_MAP(CSquaresView, CView)
    //{AFX_MSG_MAP(CSquaresView)
    ON_WM_LBUTTONDOWN()
    //}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// CSquaresView construction/destruction

CSquaresView::CSquaresView()
{
    // TODO: add construction code here
}

CSquaresView::~CSquaresView()
{
}

BOOL CSquaresView::PreCreateWindow(CREATESTRUCT& cs)
{
    // TODO: Modify the Window class or styles here by modifying
    // the CREATESTRUCT cs

    return CView::PreCreateWindow(cs);
}

////////////////////////////////////
// CSquaresView drawing

void CSquaresView::OnDraw(CDC* pDC)
{
    CSquaresDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);

    //
    // Set the mapping mode to MM_LOENGLISH.
    //
    pDC->SetMapMode (MM_LOENGLISH);

    //
    // Draw the 16 squares.
    //
    for (int i=0; i<4; i++) {
        for (int j=0; j<4; j++) {
            COLORREF color = pDoc->GetSquare (i, j);
            CBrush brush (color);
            int x1 = (j * 100) + 50;
            int y1 = (i * -100) - 50;
            int x2 = x1 + 100;
            int y2 = y1 - 100;
            CRect rect (x1, y1, x2, y2);
            pDC->FillRect (rect, &brush);
        }
    }
}

```

```

//
// Then the draw the grid lines surrounding them.
//
for (int x=50; x<=450; x+=100) {
    pDC->MoveTo (x, -50);
    pDC->LineTo (x, -450);
}

for (int y=-50; y>=-450; y-=100) {
    pDC->MoveTo (50, y);
    pDC->LineTo (450, y);
}
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CSquaresView diagnostics

#ifdef _DEBUG
void CSquaresView::AssertValid() const
{
    CView::AssertValid();
}

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

CSquaresDoc* CSquaresView::GetDocument() // non-debug version is inline
{
    ASSERT(m_pDocument->IsKindOf(RUNTIME_CLASS(CSquaresDoc)));
    return (CSquaresDoc*)m_pDocument;
}
#endif // _DEBUG

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// CSquaresView message handlers

void CSquaresView::OnLButtonDown(UINT nFlags, CPoint point)
{
    CView::OnLButtonDown(nFlags, point);

    //
    // Convert to click coordinates to MM_LOENGLISH units.
    //
    CClientDC dc (this);
    dc.SetMapMode (MM_LOENGLISH);
    CPoint pos = point;
    dc.DPtoLP (&pos);

    //
    // If a square was clicked, set its color to the current color.
    //
    if (pos.x >= 50 && pos.x <= 450 && pos.y <= -50 && pos.y >= -450) {
        int i = (-pos.y - 50) / 100;
        int j = (pos.x - 50) / 100;
        CSquaresDoc* pDoc = GetDocument ();
        COLORREF clrCurrentColor = pDoc->GetCurrentColor ();
        pDoc->SetSquare (i, j, clrCurrentColor);
    }
}

```