

도형 그리기

Drag하여 사각형 그리기

(1) 새 프로젝트 생성

("파일"->"새로만들기"->"프로젝트")

-> "Visual C++" -> "MFC" -> "MFC 응용프로그램")

-> 이름: **Figure**

-> 응용 프로그램 종류: "단일문서"

-> 프로젝트스타일 : "MFC Standard"

(1.5) main frame을 resize되게 하기

```
BOOL CMainFrame::PreCreateWindow(CREATESTRUCT& cs) {
    cs.style = WS_OVERLAPPED | WS_CAPTION | FWS_ADDTOTITLE
    | WS_MINIMIZEBOX | WS_MAXIMIZEBOX | WS_SYSMENU | WS_THICKFRAME;
    if( !CFrameWnd::PreCreateWindow(cs) )
        return FALSE;
    // TODO: CREATESTRUCT cs를 수정하여 여기에서
    // Window 클래스 또는 스타일을 수정합니다.
    //cs.style = WS_OVERLAPPED | WS_CAPTION | FWS_ADDTOTITLE
    // | WS_MINIMIZEBOX | WS_MAXIMIZEBOX | WS_SYSMENU | WS_THICKFRAME;

    return TRUE;
}
```

Drag하여 사각형 그리기

- (2) View 멤버변수 추가 : `CPoint m_ptStart;`
(3) `WM_LBUTTONDOWN, WM_LBUTTONUP`

```
void CFigureView::OnLButtonDown(UINT nFlags, CPoint point) {
    SetCapture();
    m_ptStart = point;
    CView::OnLButtonDown(nFlags, point);
}

void CFigureView::OnLButtonUp(UINT nFlags, CPoint point) {
    if (GetCapture() == this) {
        ReleaseCapture();

        CClientDC dc(this);
        CPen pen(PS_SOLID, 2, RGB(0,0,0));
        CPen *pOldPen = dc.SelectObject(&pen);
        CBrush nullbrush;
        nullbrush.CreateStockObject(NULL_BRUSH);
        CBrush *pOldBrush = dc.SelectObject(&nullbrush);
        dc.Rectangle(m_ptStart.x, m_ptStart.y, point.x, point.y);
        dc.SelectObject(pOldBrush);
        dc.SelectObject(pOldPen);
    }
    CView::OnLButtonUp(nFlags, point);
}
```

Drag하여 사각형 그리기

[4] Control 버튼 누르고 drag하면 원, Shift 버튼 누르고 drag하면 선 그리기.

```
void CFigureView::OnLButtonUp(UINT nFlags, CPoint point) {
    if (GetCapture() == this) {
        ReleaseCapture();
        CClientDC dc(this);
        CPen pen(PS_SOLID, 2, RGB(0,0,0));
        CPen *pOldPen = dc.SelectObject(&pen);
        CBrush nullbrush;
        nullbrush.CreateStockObject(NULL_BRUSH);
        CBrush *pOldBrush = dc.SelectObject(&nullbrush);
        if (nFlags & MK_CONTROL) {
            dc.Ellipse(m_ptStart.x, m_ptStart.y, point.x, point.y);
        }
        else if (nFlags & MK_SHIFT) {
            dc.MoveTo(m_ptStart.x, m_ptStart.y);
            dc.LineTo(point.x, point.y);
        } else
            dc.Rectangle(m_ptStart.x, m_ptStart.y, point.x, point.y);
        dc.SelectObject(pOldBrush);
        dc.SelectObject(pOldPen);
    }
    CView::OnLButtonUp(nFlags, point);
}
```

도형을 위한 최상위 클래스

[5] 도형을 위한 최상의 클래스 생성

프로젝트 명에 right click -> "추가" -> "새 항목..." : **item.h, item.cpp**

[item.h]

```
#pragma once
```

```
class CItem : public CObject
```

```
{
```

```
    DECLARE_SERIAL(CItem)
```

```
public:
```

```
    CItem() { m_color = 0; }
```

```
    CItem(int color, int x1, int y1, int x2, int y2)
```

```
{
```

```
    m_pt1 = CPoint(x1, y1);
```

```
    m_pt2 = CPoint(x2, y2);
```

```
    m_color = color;
```

```
}
```

```
virtual ~CItem() {}
```

```
virtual void Draw(CDC *pDC) {};
```

```
CPoint m_pt1, m_pt2;
```

```
int m_color;
```

```
};
```

사각형 클래스

[6] 사각형을 위한 클래스 생성

프로젝트 명에 right click -> "추가" -> "새 항목..." : **rect.h, rect.cpp**

[rect.h]

```
#include "Item.h"
class CRectangle : public CItem
{
    public:
    DECLARE_SERIAL(CRectangle)

    CRectangle();
    CRectangle(int color, int x1, int y1, int x2, int y2);
    ~CRectangle();
    virtual void Draw(CDC *pDC);
};
```

[7] 사각형을 위한 클래스 생성

[rect.cpp]

```
#include "stdafx.h"
```

```
#include "rect.h"
```

```
#include "resource.h"
```

```
IMPLEMENT_SERIAL(CRectangle, CItem, 1)
```

```
CRectangle::CRectangle() { }
```

```
CRectangle::CRectangle(int color, int x1, int y1, int x2, int y2)
```

```
: CItem(color, x1, y1, x2, y2) { }
```

```
CRectangle::~CRectangle() { }
```

```
void CRectangle::Draw(CDC *pDC) {
```

```
    COLORREF clrPen = (m_color == 0) ? RGB(255, 0, 0) : ((m_color ==  
    1) ? RGB(0, 255, 0) : RGB(0, 0, 255));
```

```
    CPen pen(PS_SOLID, 2, clrPen);
```

```
    CPen *pOldPen = pDC->SelectObject(&pen);
```

```
    CBrush nullbrush;
```

```
    nullbrush.CreateStockObject(NULL_BRUSH);
```

```
    CBrush *pOldBrush = pDC->SelectObject(&nullbrush);
```

```
    pDC->Rectangle(m_pt1.x, m_pt1.y, m_pt2.x, m_pt2.y);
```

```
    pDC->SelectObject(pOldBrush);
```

```
    pDC->SelectObject(pOldPen);
```

```
}
```

사각형 클래스

원 클래스

[8] 원을 위한 클래스 생성

프로젝트 명에 right click -> "추가" -> "새 항목..." : **circle.h, circle.cpp**

[circle.h]

```
#include "Item.h"
class CCircle : public CItem
{
    public:
    DECLARE_SERIAL(CCircle)

    CCircle();
    CCircle(int color, int x1, int y1, int x2, int y2);
    ~CCircle();
    virtual void Draw(CDC *pDC);
};
```


[9] 원을 위한 클래스 생성

프로젝트 명에 right click -> "추가" -> "새 항목..." : **circle.h, circle.cpp**

[circle.cpp]

```
#include "stdafx.h"
#include "Circle.h"
IMPLEMENT_SERIAL(CCircle, CItem, 1)
CCircle::CCircle() { }
CCircle::CCircle(int color, int x1, int y1, int x2, int y2)
: CItem(color, x1, y1, x2, y2) { }
CCircle::~~CCircle() { }
void CCircle::Draw(CDC *pDC) {
    COLORREF clrPen = (m_color == 0) ? RGB(255, 0, 0) : ((m_color ==
1) ? RGB(0, 255, 0) : RGB(0, 0, 255));

    CPen pen(PS_SOLID, 2, clrPen);
    CPen *pOldPen = pDC->SelectObject(&pen);
    COLORREF clrBrush = (m_color == 0) ? RGB(255, 0, 0) : ((m_color
== 1) ? RGB(0, 255, 0) : RGB(0, 0, 255));
    CBrush NewBrush(HS_BDIAGONAL, clrBrush);
    CBrush* pOldBrush = pDC->SelectObject(&NewBrush);
    pDC->Ellipse(m_pt1.x, m_pt1.y, m_pt2.x, m_pt2.y);
    pDC->SelectObject(pOldBrush);
    pDC->SelectObject(pOldPen);
}
```

원 클래스

[10] doc 클래스에 도형 저장을 위한 리스트 멤버 추가

```
CObList m_list;
```

리스트에 도형 추가하는 멤버 함수 추가

[doc.h]

```
public:
```

```
void AddList(int color, int shape, int x1, int y1, int x2, int y2);
```

[doc.cpp]

```
#include "rect.h"
```

```
#include "circle.h"
```

```
.....
```

```
void CFigureDoc::AddList(int color, int shape,  
    int x1, int y1, int x2, int y2)
```

```
{
```

```
    CItem *pItem = NULL;
```

```
    if (shape == 1)
```

```
        pItem = new CRectangle(color, x1, y1, x2, y2);
```

```
    else if (shape == 2)
```

```
        pItem = new CCircle(color, x1, y1, x2, y2);
```

```
    m_list.AddTail(pItem);
```

```
}
```

리스트 사용

[11] 새 도형이 생기면 doc의 m_list에 추가하고 invalidate 시키기.

```
void CFigureView::OnLButtonUp(UINT nFlags, CPoint point) {
    if (GetCapture() == this) {
        ReleaseCapture();

        CFigureDoc* pDoc = GetDocument();
        int x1 = m_ptStart.x;
        int x2 = point.x;
        int y1 = m_ptStart.y;
        int y2 = point.y;
        int shape = 1;
        if (nFlags & MK_CONTROL)
            shape = 1;
        if (nFlags & MK_SHIFT)
            shape = 2;
        x1 = (m_ptStart.x < point.x) ? m_ptStart.x : point.x;
        x2 = (m_ptStart.x < point.x) ? point.x : m_ptStart.x;
        y1 = (m_ptStart.y < point.y) ? m_ptStart.y : point.y;
        y2 = (m_ptStart.y < point.y) ? point.y : m_ptStart.y;
        pDoc->AddList(RGB(0,0,0), shape, x1, y1, x2, y2);
        Invalidate();
    }
}
CView::OnLButtonUp(nFlags, point);
}
```

View에서 새 도형 처리

View에서 직접 그리지 않기

[12] OnDraw()에서 CItem 상속 클래스 객체의 Draw() 함수를 호출.

```
void CFigureView::OnDraw(CDC* pDC) {
    CFigureDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if (!pDoc)
        return;

    POSITION pos = pDoc->m_list.GetHeadPosition();
    while (pos)
    {
        CItem *pItem = (CItem *)pDoc->m_list.GetNext(pos);
        pItem->Draw(pDC);
    }
}
```

창을 resize 해도 document의 목록에 있는 도형 객체가 새로 그려주기 때문에 같은 내용을 볼 수 있다.