

객체 지향 프로그래밍 응용

Chap 7. GDI를 이용한 고급 그래픽



2012.11.05.

오 병 우

컴퓨터공학과
금오공과대학교

XOR를 사용한 Pen 및 Brush

● XOR의 특성

◆ XOR 연산을 두 번 수행하면 원래 데이터

● 배경을 깨뜨리지 않고 쉽게 그림 출력

◆ XOR을 두 번 수행하면 복원 되므로 출력할 그림 영역의 저장 필요없음

◆ 주로 임시의 영역 선택, 선 긋기 등에 사용됨

실습

OnDraw()에서는 배경만 색칠

```
class CXORLineView : public CView
{
protected:
    CPoint m_ptFrom, m_ptTo;
```

```
CXORLineView::CXORLineView()
{
    // TODO: add construction code here
    m_ptFrom = CPoint(0,0);
    m_ptTo = CPoint(0,0);
}
```

```
void CXORLineView::OnDraw(CDC* pDC)
{
    CXORLineDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    // TODO: add draw code for native data here

    CRect rect;
    GetClientRect(&rect);

    pDC->FillSolidRect(&rect, RGB(0,0, 255));
}
```

```
void CXORLineView::OnLButtonDown(UINT nFlags, CPoint point)
{
    if (nFlags & MK_CONTROL) // MK_SHIFT도 사용 가능
        m_ptFrom = m_ptTo = point;

    CView::OnLButtonDown(nFlags, point);
}

void CXORLineView::OnMouseMove(UINT nFlags, CPoint point)
{
    if (nFlags & MK_LBUTTON) {
        CClientDC dc(this);
        dc.SetROP2(R2_XORPEN);

        CPen pen(PS_SOLID, 10, RGB(255,255,255)), *oldPen;
        oldPen = dc.SelectObject(&pen);

        dc.MoveTo(m_ptFrom);
        dc.LineTo(m_ptTo);

        m_ptTo = point;

        dc.MoveTo(m_ptFrom);
        dc.LineTo(m_ptTo);
        dc.SelectObject(oldPen);
    }

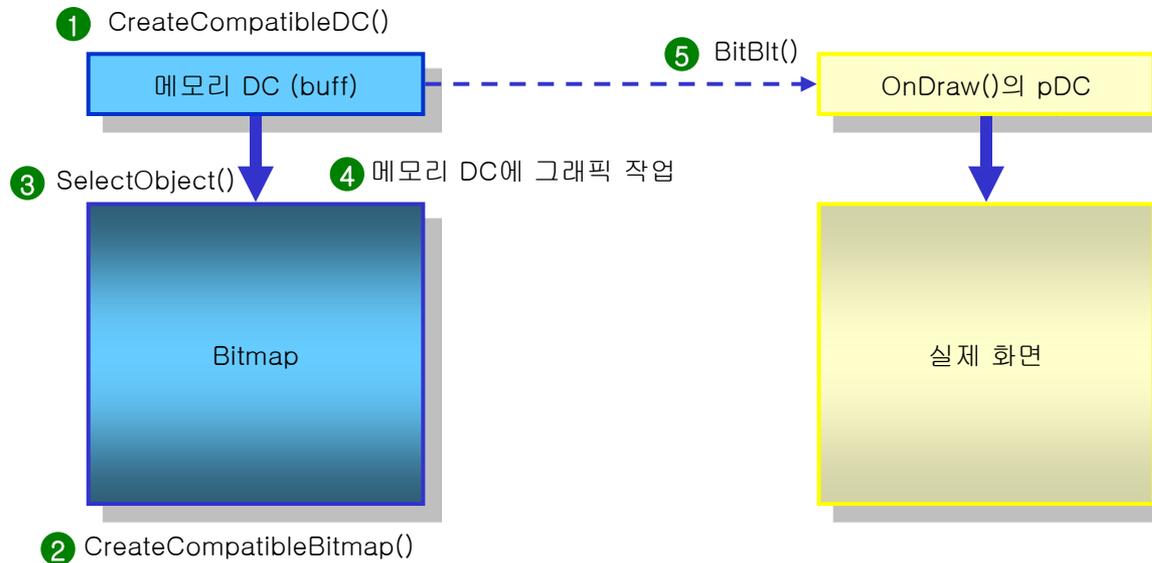
    CView::OnMouseMove(nFlags, point);
}
```

XOR

마우스 왼쪽 버튼이 누려 있는 상태인지 검사

Memory DC를 사용한 화면 출력

- 화면에 그리는 과정이 보여지면 깜빡임 발생
- 깜빡임 없는 고속 그래픽 출력
 - ◆ 메모리 DC 사용
 - 미리 메모리에 모두 그린 후에 한번에 화면으로 전송



실습

- 메모리 DC를 통한 그래픽 작업

- WM_ERASEBKGND 처리

```

BOOL CRandRectView::OnEraseBkgnd(CDC* pDC)
{
    // TODO: Add your message handler code here and
    // return CView::OnEraseBkgnd(pDC);
    return TRUE;
}
    
```

```

void CRandRectView::OnDraw(CDC* pDC)
{
    CRandRectDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    // TODO: add draw code for native data here

    int i;

    srand(100); // 초기값을 고정하여 항상 같은 결과를 얻음

    CRect rect;
    GetClientRect(&rect);

    if (m_bUseMemDC) {
        // 메모리 DC 생성
        CDC buff;
        buff.CreateCompatibleDC(pDC);

        // 메모리 DC에서 사용할 Bitmap 생성
        CBitmap bmp, *oldBmp;
        bmp.CreateCompatibleBitmap(&buff, rect.right, rect.bottom);

        // 메모리 DC에 생성된 Bitmap 선택
        oldBmp = buff.SelectObject(&bmp);

        // Bitmap 초기화
        buff.FillSolidRect(0, 0, rect.right, rect.bottom, RGB(255,255,255));

        for (i = 0; i < 1000; i++)
            buff.Rectangle(rand()%400, rand()%400, rand()%400, rand()%400);

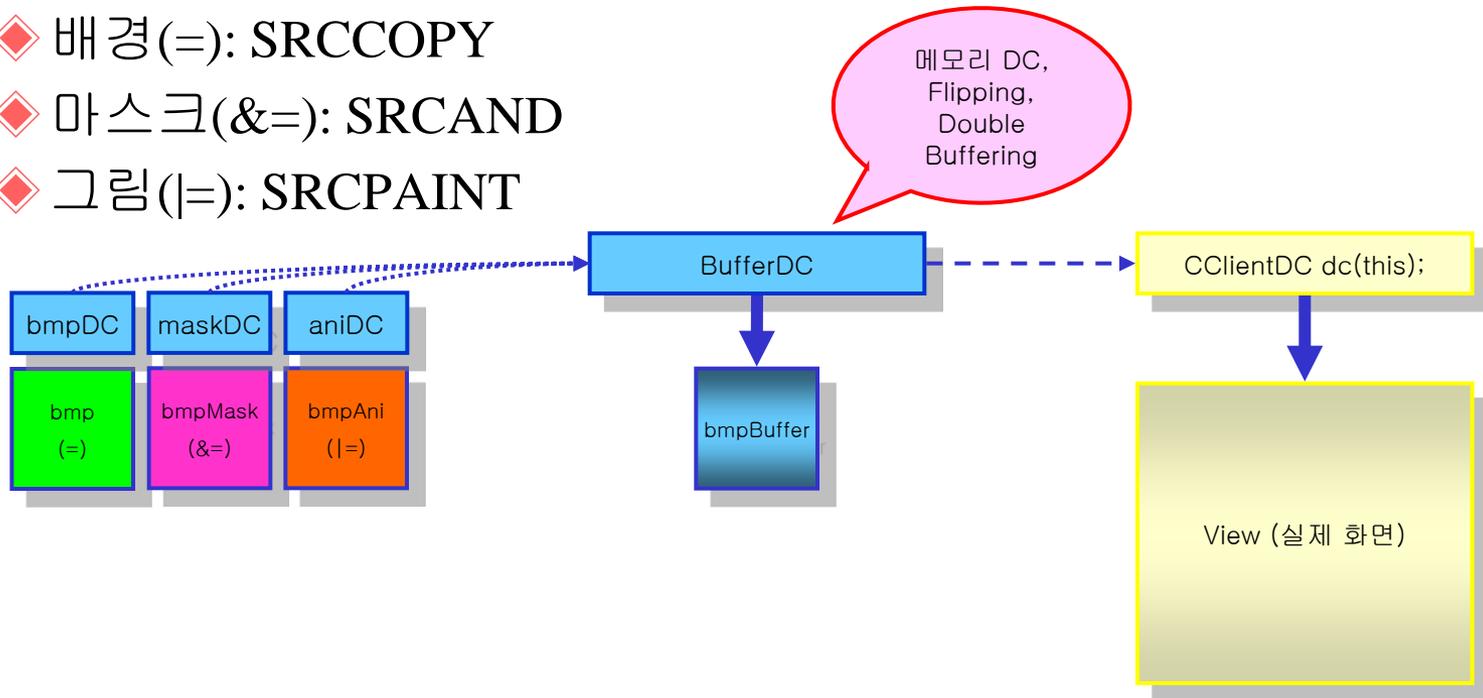
        // 메모리 DC로부터 화면으로 출력
        pDC->BitBlt(0, 0, rect.right, rect.bottom, &buff, 0, 0, SRCCOPY);

        buff.SelectObject(oldBmp);
    }
    else {
        pDC->FillSolidRect(&rect, RGB(255,255,255));

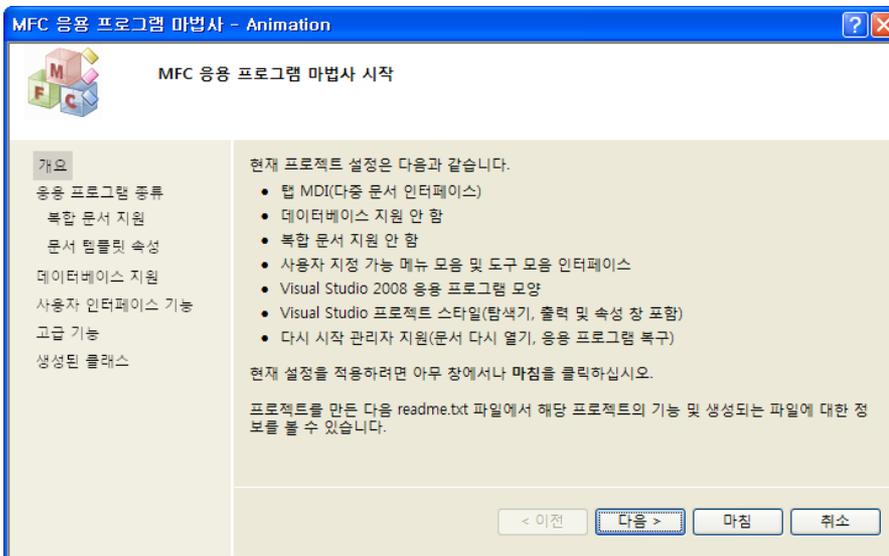
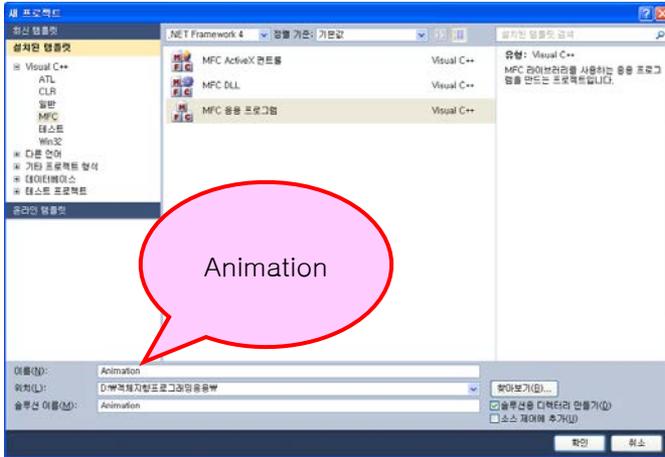
        for (i = 0; i < 1000; i++)
            pDC->Rectangle(rand()%400, rand()%400, rand()%400, rand()%400);
    }
}
    
```

Animation 출력

- 메모리 DC를 사용
 - ◆ 부드러운 Animation 출력
 - ◆ 배경, 마스크, 그림
- 메모리 DC
 - ◆ 배경(=): SRCCOPY
 - ◆ 마스크(&=): SRCAND
 - ◆ 그림(|=): SRCPAINT

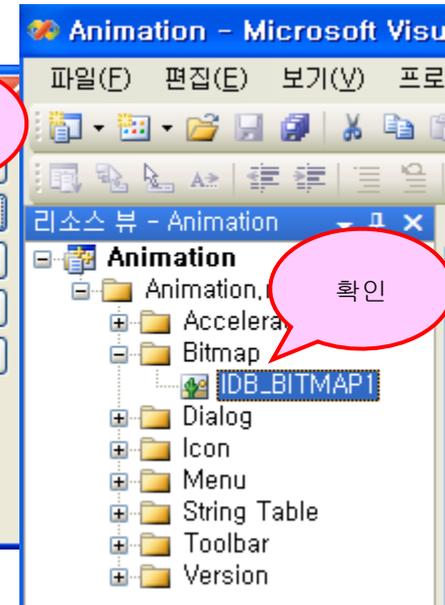
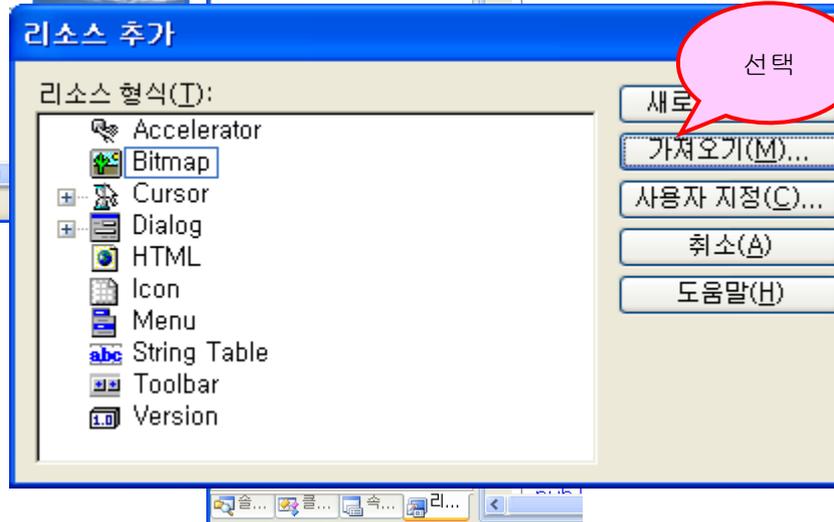
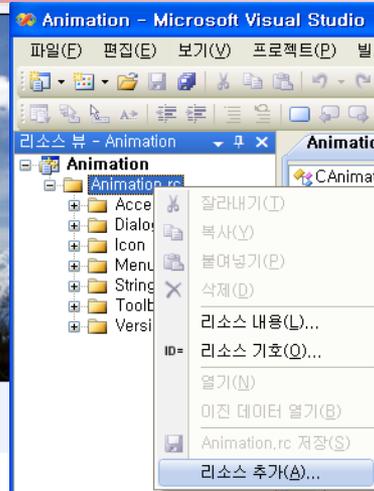
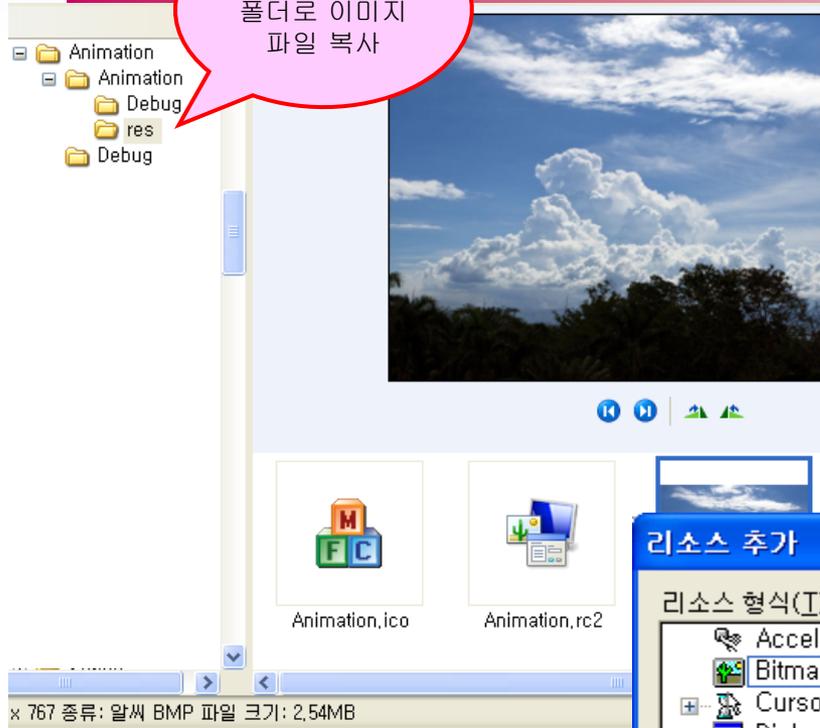


실습 프로젝트 생성

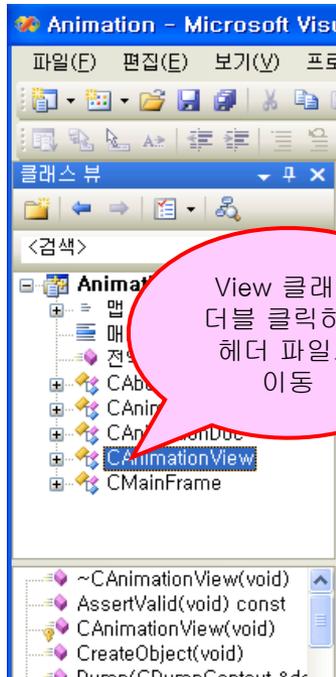


비트맵 이미지 추가

res 리소스
폴더로 이미지
파일 복사

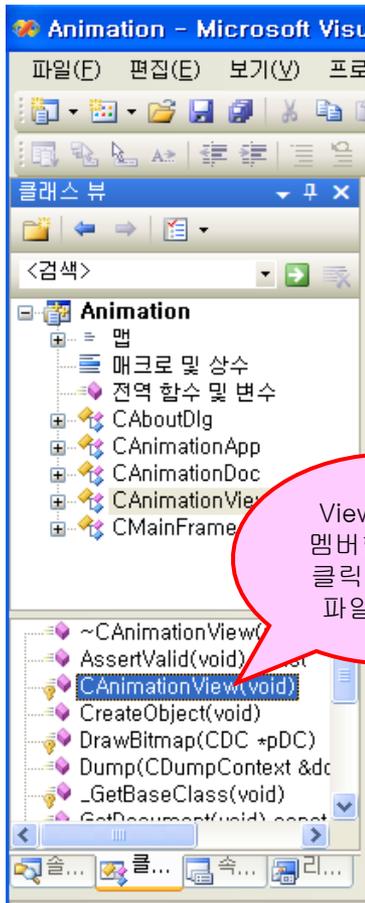


DrawBitmap() 함수 추가



```
class CAnimationView : public CView
{
protected:
    void DrawBitmap(CDC *pDC);
protected: // serialization에서만 만들어집니다.
    CAnimationView();
    DECLARE_DYNCREATE(CAnimationView)
};
```

DrawBitmap() 함수 구현



View 클래스
멤버함수 더블
클릭하여 cpp
파일로 이동

```

void CAnimationView::DrawBitmap(CDC *pDC)
{
    //-----
    // bmpDC 생성
    CDC bmpDC;
    bmpDC.CreateCompatibleDC(pDC);

    // 비트맵 로드
    CBitmap bmp;
    bmp.LoadBitmapW(IDB_BITMAP1);

    // 비트맵 정보 얻기 (크기)
    BITMAP bmpinfo;
    bmp.GetBitmap(&bmpinfo);

    // bmpDC에서 로드된 비트맵 사용
    CBitmap *oldBmp = bmpDC.SelectObject(&bmp);

    // 화면으로 bmpDC의 비트맵 전송
    pDC->BitBlt(0, 0, bmpinfo.bmWidth, bmpinfo.bmHeight,
               &bmpDC, 0, 0, SRCCOPY);

    // 비트맵 복원
    bmpDC.SelectObject(oldBmp);
}
    
```

View 클래스
cpp 파일 맨
뒤에 추가

애니메이션용 비트맵

res 폴더에 넣고 리소스 추가

◆ IDB_ANI

- 애니메이션용 그림
- 그림이 없는 곳은 검은색



◆ IDB_ANIMASK

- 애니메이션용 마스크
- 그림이 있는 곳은 검은색
- 그림이 없는 곳은 흰색



DrawBitmap()에 ANI 출력 추가

```
// 화면으로 memDC의 비트맵 전송
pDC->BitBlt(0, 0, bmpinfo.bmWidth, bmpinfo.bmHeight,
            &bmpDC, 0, 0, SRCCOPY);
```

```
bmpDC.SelectObject(oldBmp);
```

```
//-----
// 애니메이션 그림을 위한 aniDC 생성
CDC aniDC;
aniDC.CreateCompatibleDC(pDC);

// 비트맵 로드
CBitmap bmpAni;
bmpAni.LoadBitmapW(IDB_ANI);

// aniDC에서 로드된 비트맵 사용
oldBmp = aniDC.SelectObject(&bmpAni);

// 화면으로 aniDC의 비트맵 전송
pDC->BitBlt(300, 300, 62, 64,
            &aniDC, 0, 0, SRCCOPY);
aniDC.SelectObject(oldBmp);
//-----
}
```



DrawBitmap()에 Mask 출력 추가

```
// 화면으로 비트맵을 비트맵 인자
pDC->BitBlt(0, 0, bmpinfo.bmWidth, bmpinfo.bmHeight,
            &bmpDC, 0, 0, SRCCOPY);
```

```
bmpDC.SelectObject(oldBmp);
```

```
// 애니메이션 그림 마스크를 위한 maskDC 생성
CDC maskDC;
maskDC.CreateCompatibleDC(pDC);
```

```
// 비트맵 로드
CBitmap bmpMask;
bmpMask.LoadBitmapW(IDB_ANIMASK);
```

```
// maskDC에서 로드된 비트맵 사용
oldBmp = maskDC.SelectObject(&bmpMask);
```

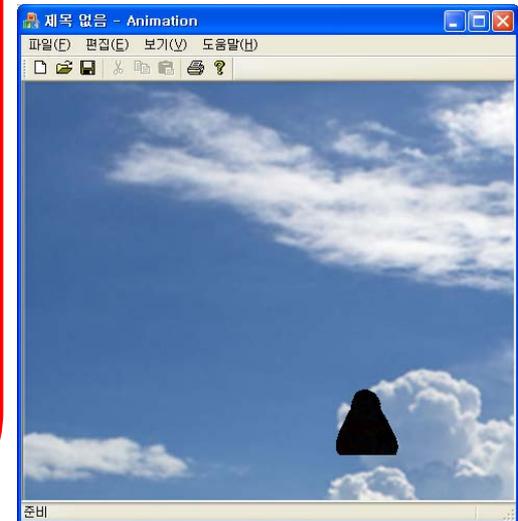
```
// 화면으로 maskDC의 비트맵 전송
pDC->BitBlt(300, 300, 62, 64,
            &maskDC, 0, 0, SRCAND);
```

```
maskDC.SelectObject(oldBmp);
```

```
/*
// 애니메이션 그림을 위한 aniDC 생성
CDC aniDC;
```

그림
그리는
부분은
주석 처리

SRCAND



DrawBitmap()에 ANI 출력 수정

```
// aniDC에서 로드된 비트맵 사용
oldBmp = maskDC.SelectObject(&bmpMask);
```

```
// 화면으로 memDC의 비트맵 전송
pDC->BitBlt(300, 300, 62, 64,
            &maskDC, 0, 0, SRCAND);
```

그림
그리는
부분은
주석 해제

```
maskDC.SelectObject(oldBmp);
```

```
//-----
// 애니메이션 그림을 위한 aniDC 생성
CDC aniDC;
aniDC.CreateCompatibleDC(pDC);
```

```
// 비트맵 로드
CBitmap bmpAni;
bmpAni.LoadBitmapW(IDB_ANI);
```

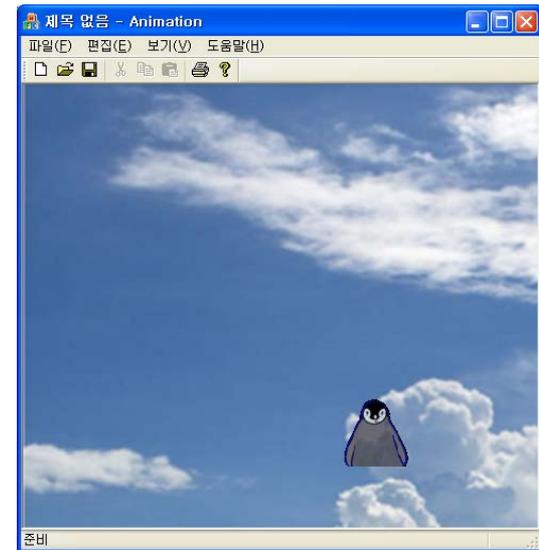
```
// aniDC에서 로드된 비트맵 사용
oldBmp = aniDC.SelectObject(&bmpAni);
```

SRCPAINT
로 변경

```
// 화면으로 memDC의 비트맵 전송
pDC->BitBlt(300, 300, 62, 64,
            &aniDC, 0, 0, SRCPAINT);
```

```
aniDC.SelectObject(oldBmp);
```

```
//-----
```



애니메이션 순서

```
void CAnimationView::DrawBitmap(CDC *pDC)
```

```
{
    static int nAnimate = 0;
```

```
    // memDC 생성
    CDC bmpDC;
    bmpDC.CreateCompatibleDC(pDC);

    // 비트맵 로드
```

```
    // 화면으로 maskDC의 비트맵 전송
    pDC->BitBlt(300, 300, 62, 64,
               &maskDC, 62 * nAnimate, 0, SRCAND);

    maskDC.SelectObject(oldBmp);
```

```
    // 애니메이션 그림을 위한 aniDC 생성
    CDC aniDC;
    aniDC.CreateCompatibleDC(pDC);

    // 비트맵 로드
    CBitmap bmpAni;
    bmpAni.LoadBitmapW(IDB_ANI);

    // aniDC에서 로드된 비트맵 사용
    oldBmp = aniDC.SelectObject(&bmpAni);
```

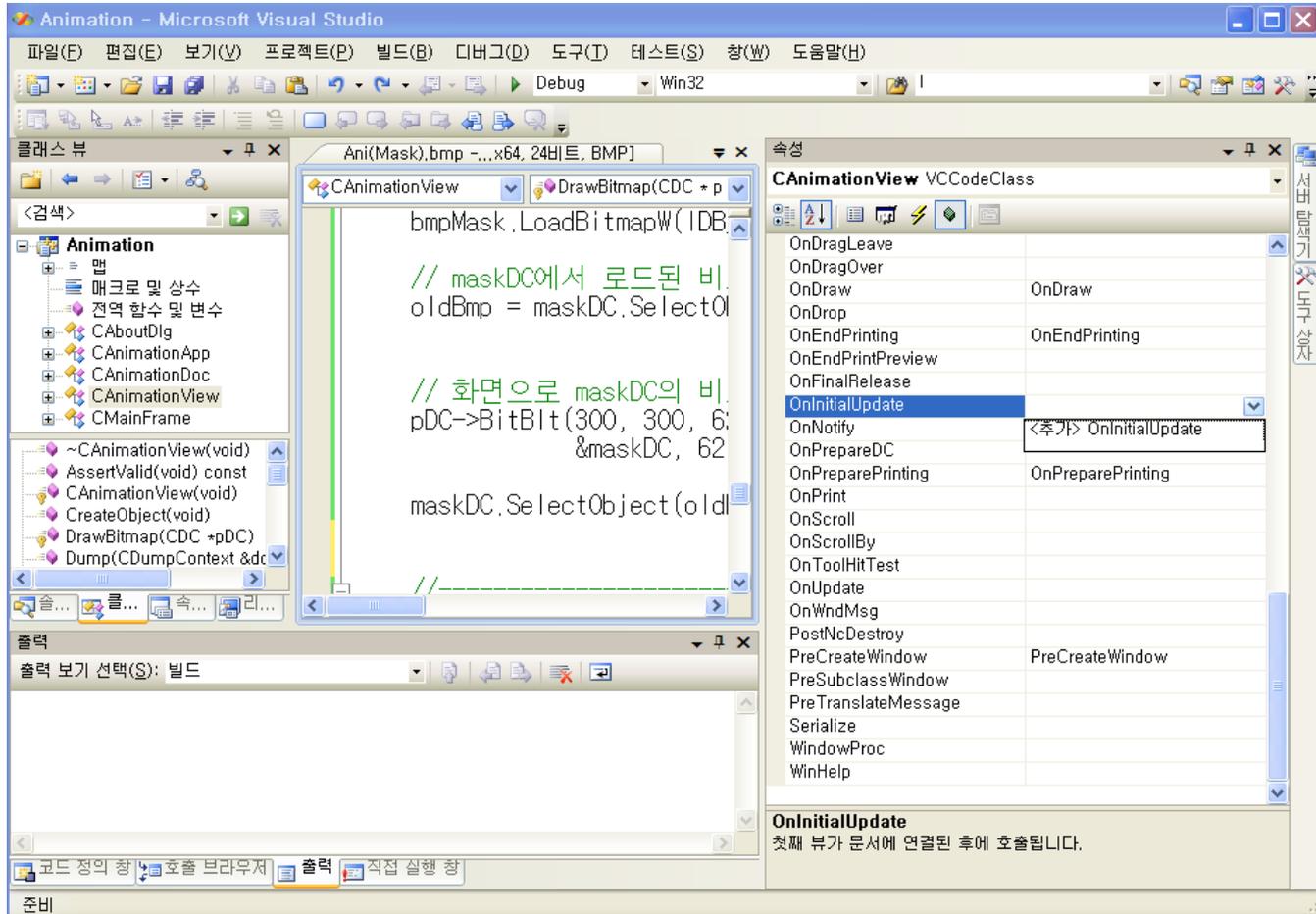
```
    // 화면으로 aniDC의 비트맵 전송
    pDC->BitBlt(300, 300, 62, 64,
               &aniDC, 62 * nAnimate, 0, SRCPAINT);

    if (++nAnimate > 2)
        nAnimate = 0;
```

```
    aniDC.SelectObject(oldBmp);
    //
```

OnInitialUpdate() 재정의

- 타이머 초기화를 위해 View를 처음 그리는 함수 재정의



OnInitialUpdate() 재정의

SetTimer()

```
void CAnimationView::OnInitialUpdate()
{
    CView::OnInitialUpdate();

    // TODO: 여기에 특수화된 코드를 추가 및/또는 기본 클래스를 호출합니다.
    SetTimer(0, 200, NULL);
}

```

타이머 ID
(여러 개의
타이머 사용
가능)

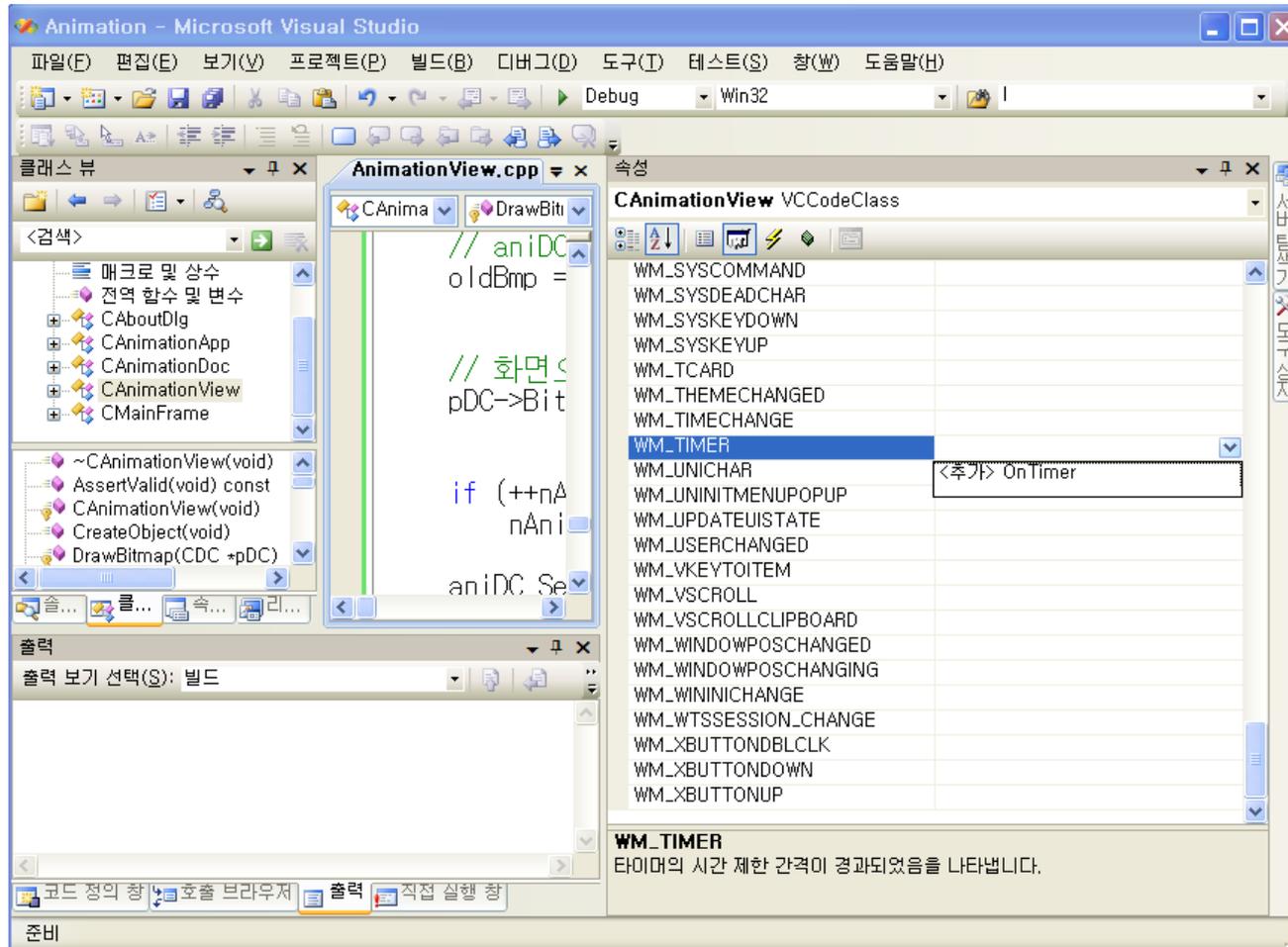
200 ms
(오차 있을
수 있음)

콜백 함수
주소 지정
가능하지만
주로 NULL

NULL일
때는
OnTimer()에
서 모두 처리

WM_TIMER 처리

타이머 메시지 처리



OnTimer()

타이머 메시지 핸들링

◆ Invalidate() 호출

- 화면 다시 그림

타이머 ID
(여러 개
세팅 가능)

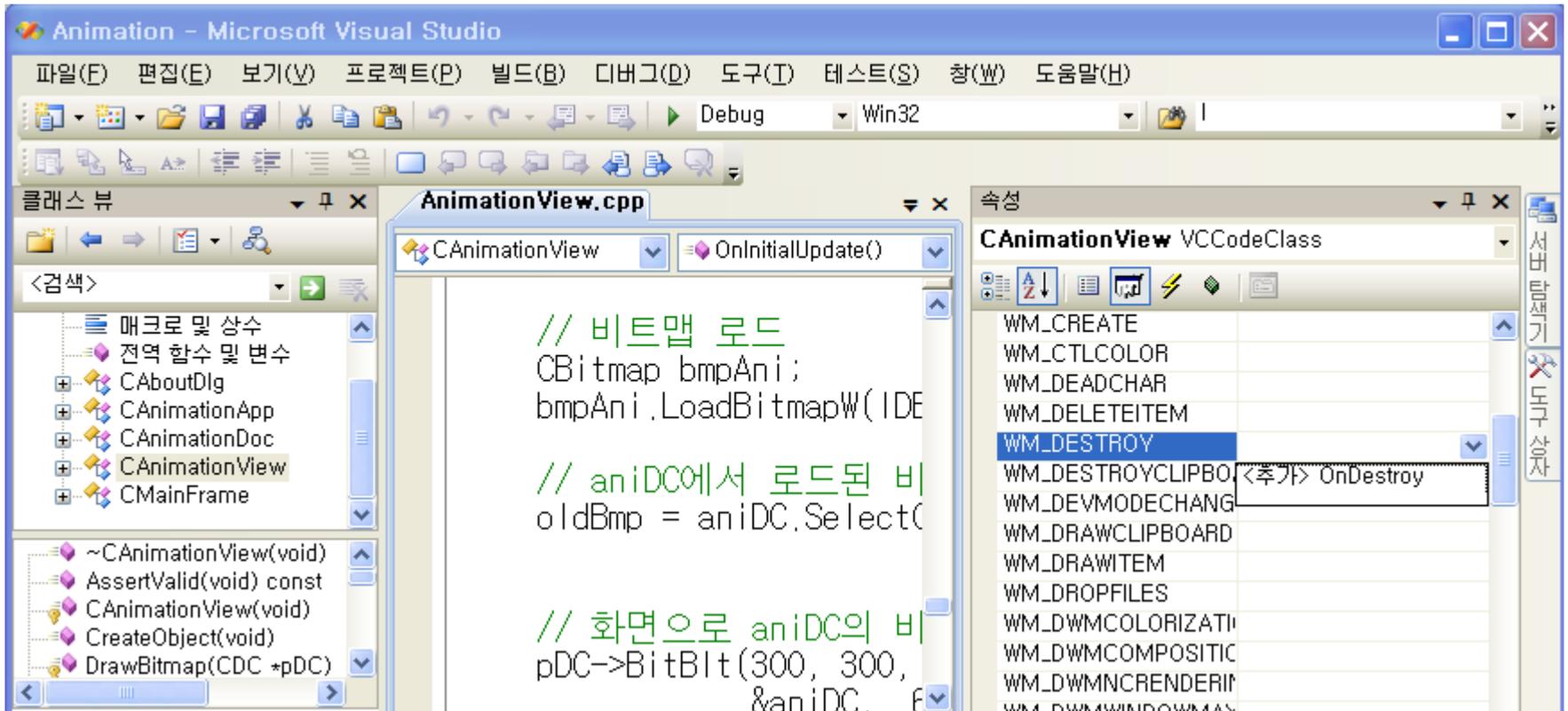
```

void CAnimationView::OnTimer(UINT_PTR nIDEvent)
{
    // TODO: 여기에 메시지 처리기 코드를 추가 및/또는 기본값을 호출합니다.
    Invalidate();
    CView::OnTimer(nIDEvent);
}

```

WM_DESTROY

- 프로그램 종료시 발생하는 WM_DESTROY 메시지 처리



AnimationView.cpp

```

// 비트맵 로드
CBitmap bmpAni;
bmpAni.LoadBitmapW(IDE

// aniDC에서 로드된 비
oldBmp = aniDC.SelectC

// 화면으로 aniDC의 비
pDC->BitBlt(300, 300,
&aniDC, f
    
```

속성

Message	Handler
WM_CREATE	
WM_CTLCOLOR	
WM_DEADCHAR	
WM_DELETEITEM	
WM_DESTROY	<추가> OnDestroy
WM_DESTROYCLIPBO	
WM_DEVMODECHANG	
WM_DRAWCLIPBOARD	
WM_DRAWITEM	
WM_DROPFILES	
WM_DWMCOLORIZATI	
WM_DWMCOMPOSITIC	
WM_DWMNCRENDERIN	
WM_DWMWINDOWMAX	

OnDestroy()

타이머 종료

```

void CAnimationView::OnTimer(UINT_PTR nIDEvent)
{
    // TODO: 여기에 메시지 처리기 코드를 추가 및/또는 기본값을 호출합니다.
    CClientDC dc(this);
    DrawBitmap(&dc);

    CView::OnTimer(nIDEvent);
}

void CAnimationView::OnDestroy()
{
    CView::OnDestroy();

    // TODO: 여기에 메시지 처리기 코드를 추가합니다.
    KillTimer(0);
}

```

타이머 ID

Buffer DC 사용

```

void CAnimationView::OnDraw(CDC* pDC)
{
    CAnimationDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if (!pDoc)
        return;

    // TODO: 여기에 원시 데이터에 대한 그리기 코드를 추가합니다
    CRect rect;
    GetClientRect(&rect);

    CDC buff;
    buff.CreateCompatibleDC(pDC);

    CBitmap bmp, *oldBmp;
    bmp.CreateCompatibleBitmap(pDC, rect.right, rect.bottom);

    oldBmp = buff.SelectObject(&bmp);

    // 애니메이션 그리는 함수 호출
    DrawBitmap(&buff);

    pDC->BitBlt(0, 0, rect.right, rect.bottom,
               &buff, 0, 0, SRCCOPY);

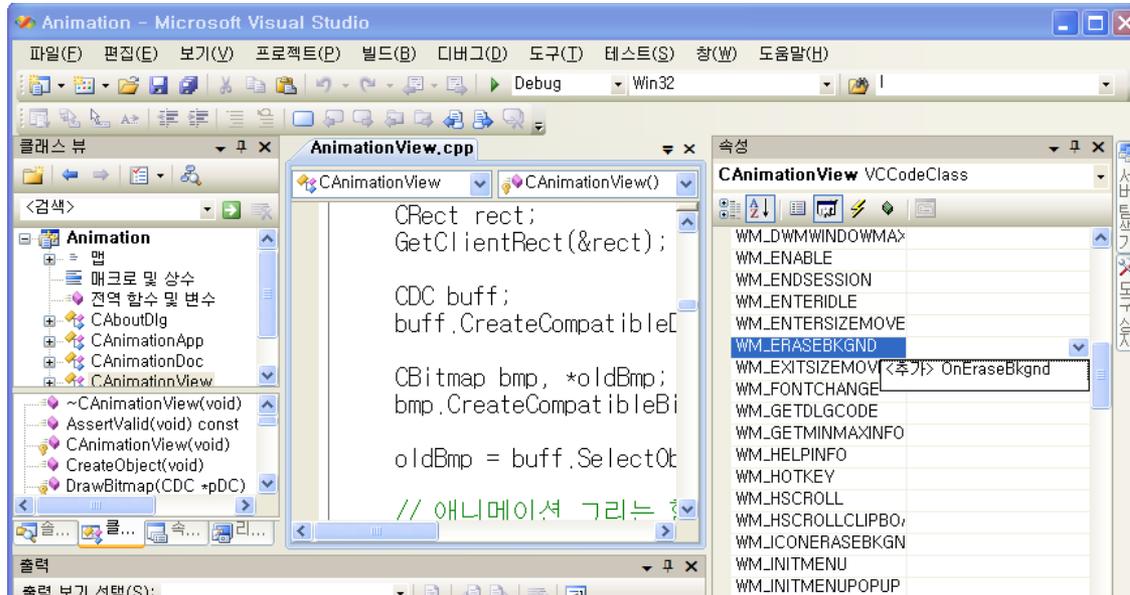
    buff.SelectObject(oldBmp);
}

```

화면의 전체
영역

WM_ERASEBKGD 처리

- 화면 전체 영역에 대해 BitBlt 하므로 지울 필요 없음



```

BOOL CAnimationView::OnEraseBkgnd(CDC* pDC)
{

```

// TODO: 여기에 메시지 처리기 코드를 추가 및/또는 기본값을 호출합니다.

```

//return CView::OnEraseBkgnd(pDC);
return TRUE;
}

```