

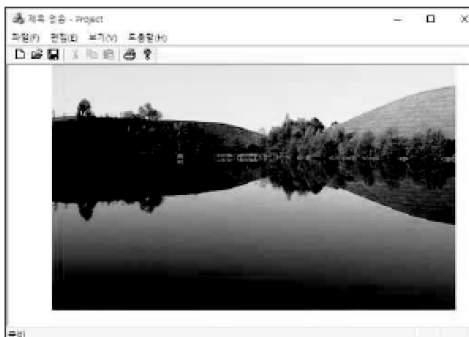


# 비트맵

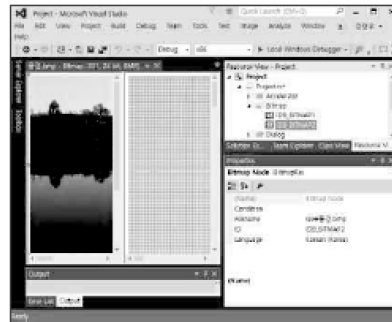
# 비트맵

## 비트맵 이미지 출력 과정

### 비트맵 파일



### 파일 편집(D 부여)



①  
비트맵 가져오기

②  
LoadBitmap()  
비트맵 로드  
CBitmap 객체  
bitmap

④  
SelectObject()  
사용할 비트맵 선택

⑤ BitBlt()

비트맵 전송

③  
CreateCompatibleDC()

dc  
CDC 객체

memdc  
CDC 객체

# 비트맵

## CDC::BitBlt() 함수

```
BOOL BitBlt(  
    int    nXD,  
    int    nYD,  
    int    nW,  
    int    nH,  
    CDC    memdc,  
    int    nXS,  
    int    nYS,  
    DWORD  dwRop  
);
```

# 비트맵

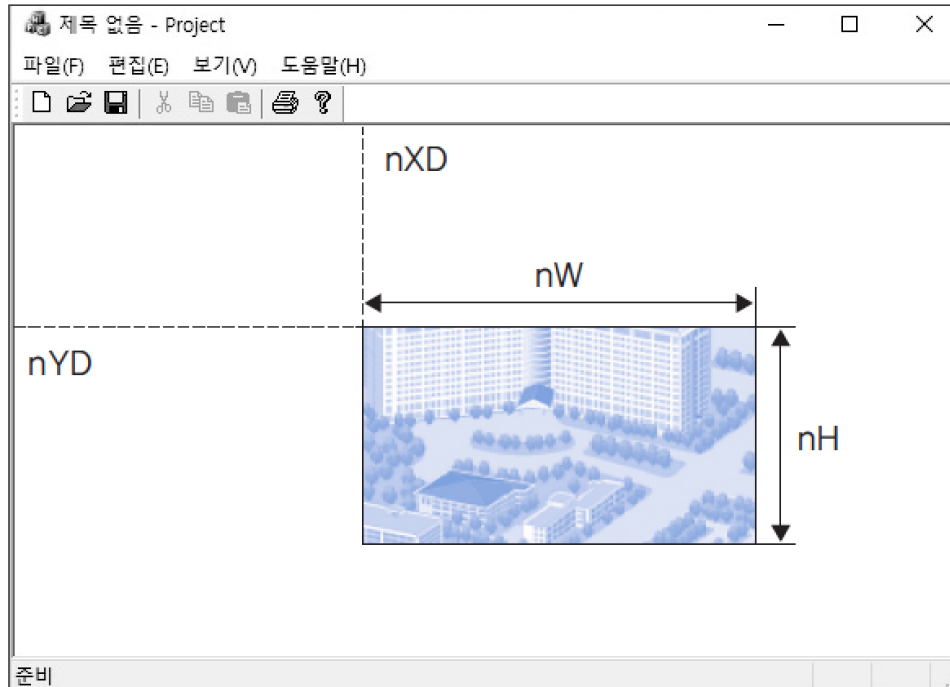
## StretchBlt() 함수

```
BOOL CDC::StretchBlt(  
    int        nXD,  
    int        nYD,  
    int        nW,  
    int        nH,  
    HDC        memdc,  
    int        nXS,  
    int        nYS,  
    int        nWS,  
    int        nHS,  
    DWORD      dwRop  
);
```

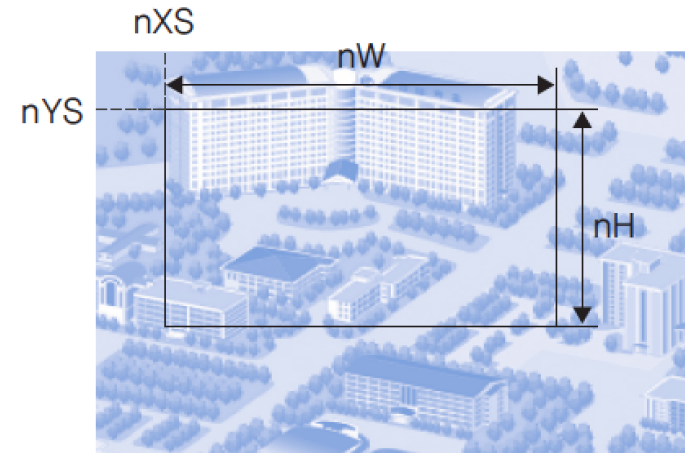
# 비트맵

BitBlt() 함수를 이용하여 화면에 비트맵 출력하기

dc



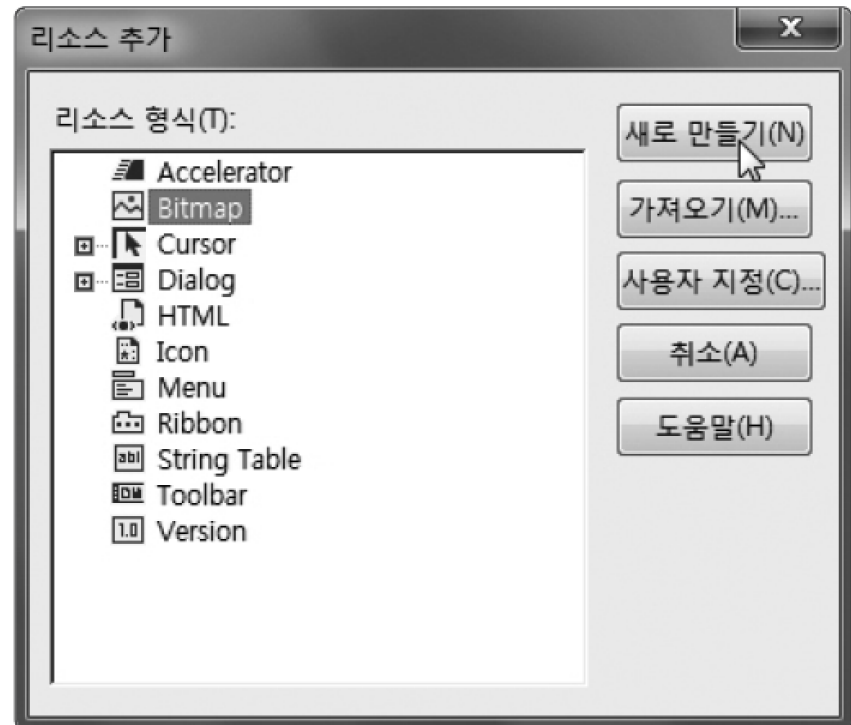
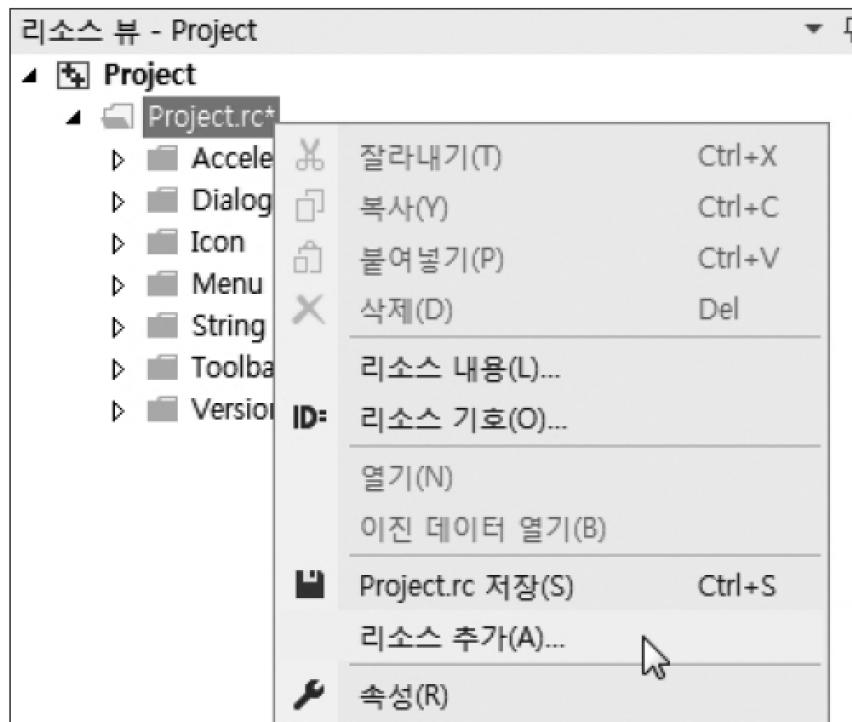
memdc



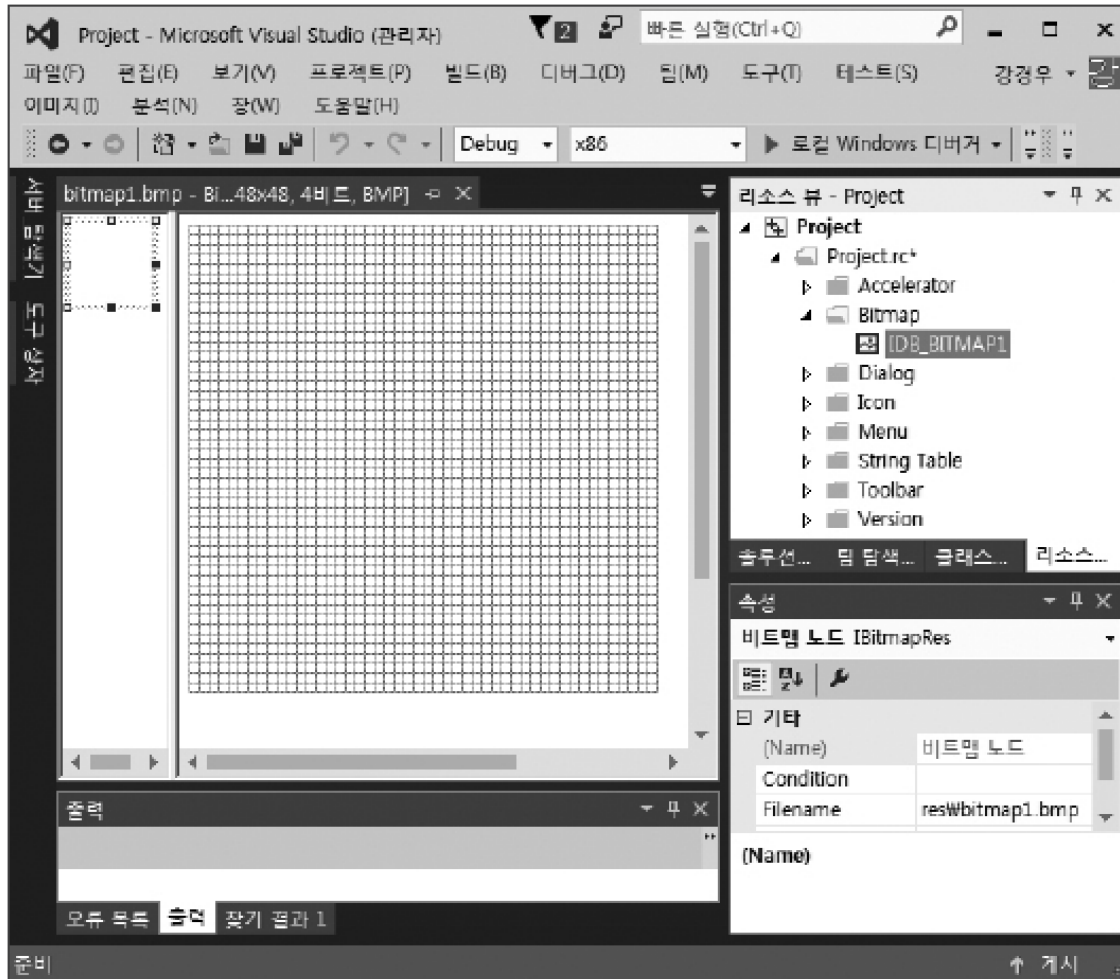
# 비트맵

## 비트맵 이미지 단순 출력하기

### • 비트맵 가져오기

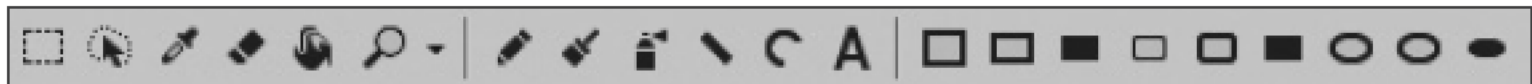


# 비트맵



◀ 비트맵 편집창

▼ 비트맵 편집 도구



# 비트맵

속성

비트맵 편집기 IBmpEd

기타

(Name)	비트맵 편집기
Compressed	False
Filename	C:\Users\User\Desktop\Proj
Image Type	BMP

모양

Colors	24비트
--------	------

위치

Height	4비트
Width	8비트
	24비트
	32비트

Colors

현재 이미지에 사용할 수 있는 색 수를 지정합니다.

속성

비트맵 편집기 IBmpEd

기타

(Name)	비트맵 편집기
Compressed	False
Filename	C:\Users\User\Desktop\Proj
Image Type	BMP

모양

Colors	24비트
--------	------

위치

Height	48
Width	48

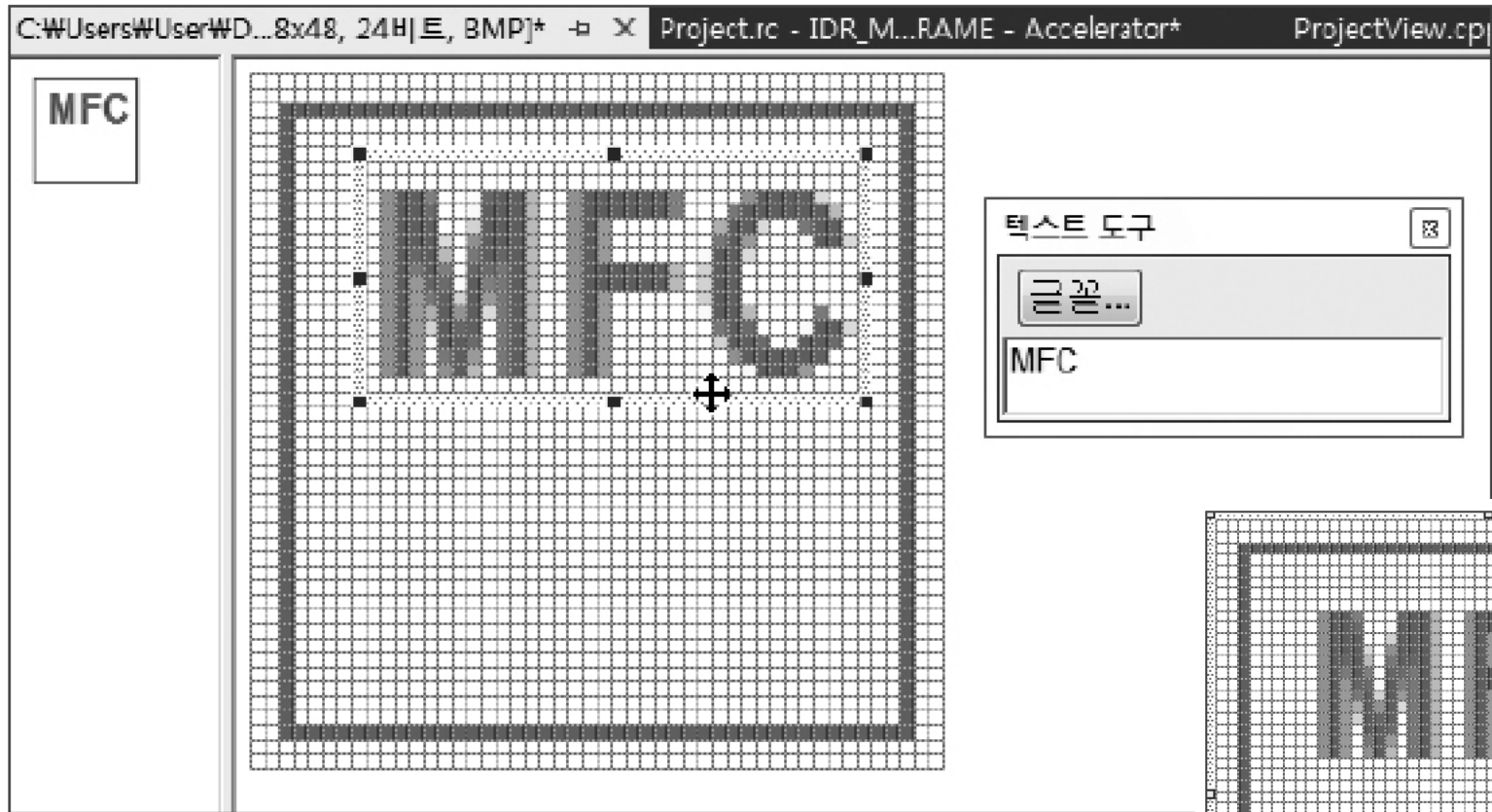
Height

이미지 높이(픽셀)를 지정합니다.

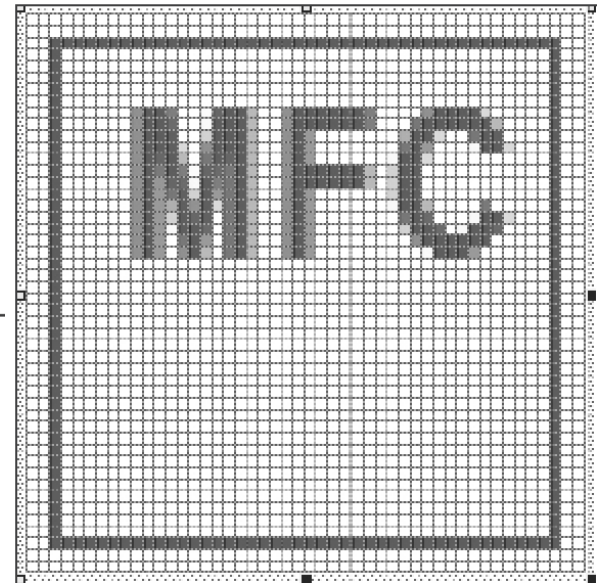
## ▲ 비트맵 속성 변경



# 비트맵



## ▲ 비트맵에 텍스트 나타내기



# 비트맵

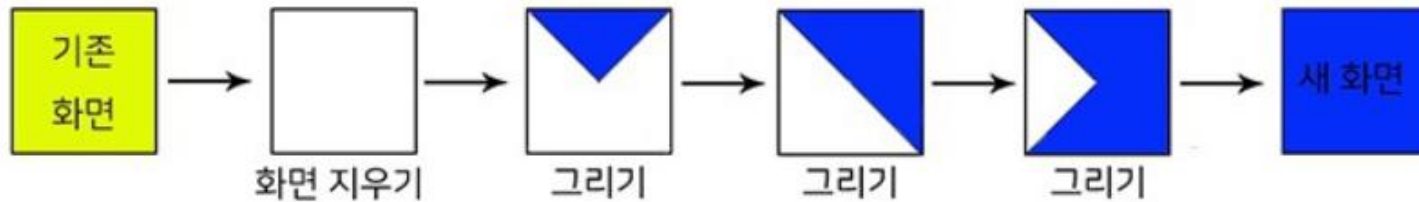


비트맵 가져오기

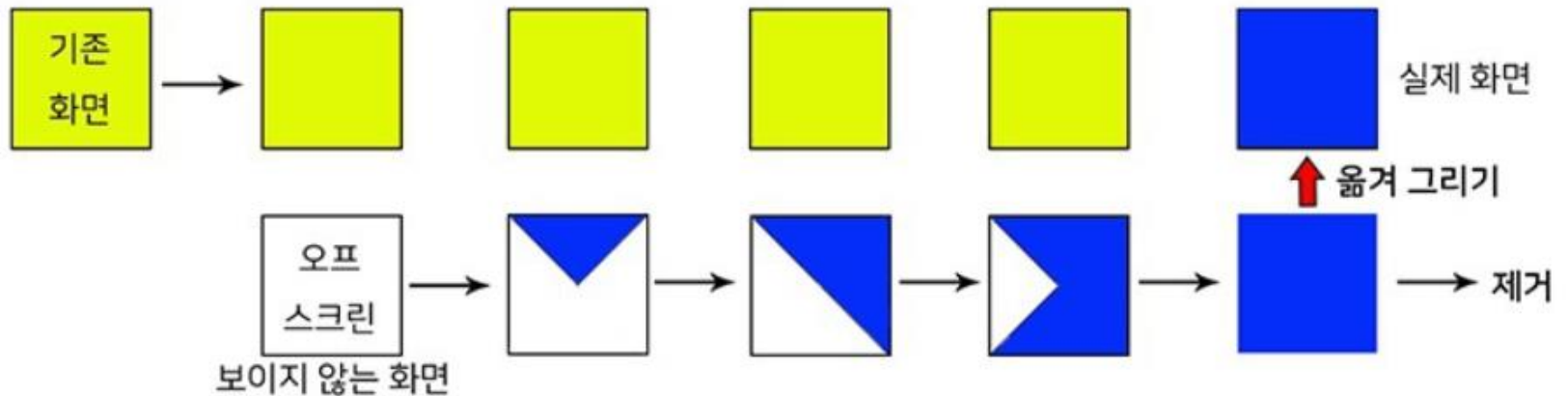


# 비트맵

더블 버퍼링 적용 전 화면 전환



더블 버퍼링 적용 후 화면 전환



# 비트맵

## • 함수의 코드 수정

```
01 void CProjectView::OnDraw(CDC* pDC)
02 {
03     CProjectDoc* pDoc = GetDocument();
04     ASSERT_VALID(pDoc);
05     if (!pDoc)
06         return;
07     CDC mdc;
08     CBitmap s_bit, l_bit, *oldbit;
09     mdc.CreateCompatibleDC(pDC);
10     s_bit.LoadBitmap(IDB_BITMAP1);
11     l_bit.LoadBitmap(IDB_BITMAP2);
12     oldbit = mdc.SelectObject(&s_bit);
13     pDC->BitBlt(0, 0, 48, 48, &mdc, 0, 0, SRCCOPY);
14     mdc.SelectObject(&l_bit);
15     pDC->BitBlt(60, 0, 546, 331, &mdc, 0, 0, SRCCOPY);
16     mdc.SelectObject(oldbit);
17     s_bit.DeleteObject();
18     l_bit.DeleteObject();
19     mdc.DeleteDC();
20 }
```

- 
- pDC->StretchBlt(100, 100, 200, 300, &mdc, 100, 100, 500, 500, SRCCOPY);
  - SRCAND : AND 연산자를 사용하여 대상 비트 맵과 소스 비트 맵의 픽셀을 결합  
SRCCOPY : 소스 비트 맵을 대상 비트 맵에 복사.  
SRCPAINT : OR 연산자를 사용하여 대상 비트 맵과 소스 비트 맵의 픽셀을 결합

# 래스터 연산

- 배경색과 출력하는 색 사이에 이루어지는 연산
- 비트맵 이미지 = 캐릭터 이미지 + 마스크 이미지

비트맵에서 얻은 캐릭터 이미지와 마스크 이미지



# 래스터 연산

마스크 출력



컬러



흰색 또는 검은색

$$\begin{array}{rcl} & \text{흰색}(255,255,255) & \\ \text{AND} & \text{컬러}(X,X,X) & \\ \hline & \text{컬러}(X,X,X) & \end{array}$$

$$\begin{array}{rcl} & \text{검은색}(0,0,0) & \\ \text{AND} & \text{컬러}(X,X,X) & \\ \hline & \text{검은색}(0,0,0) & \end{array}$$

# 래스터 연산

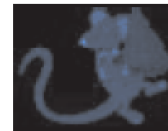
마스크 결과와 캐릭터 출력



검은색 또는 컬러

$$\begin{array}{rcl} & \text{검은색}(0,0,0) & \\ \text{OR} & \text{컬러}(X,X,X) & \\ \hline & \text{컬러}(X,X,X) & \end{array}$$

← OR



검은색 또는 컬러



# 래스터 연산

---

캐릭터 출력



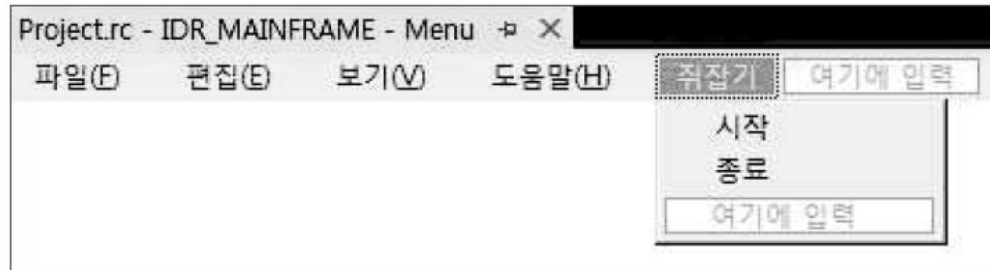
# 래스터 연산

## 래스터 연산을 이용하여 캐릭터 이미지 출력하기

발생하는 이벤트	이벤트 처리 함수	해야 할 일
WM_COMMAND	OnStart( )	• 타이머를 동작시킴
WM_COMMAND	OnEnd( )	• 타이머를 정지시킴
WM_TIMER	OnTimer( )	• 쥐의 위치를 이동함 • 뷰 영역 무효화
WM_MOUSEMOVE	OnMouseMove( )	• 마우스를 드래그 중이면 가늠자의 위치를 마우스 포인터의 위치로 변경함 • 뷰 영역 무효화
	OnDraw( )	• 배경 이미지 위에 쥐와 가늠자를 출력함

# 래스터 연산

## • 리소스 편집



메뉴 항목	ID	속성
취잡기		Pop-up: True
시작	ID_START	변경 없음
종료	ID_END	변경 없음



### ▲ 비트맵 이미지 추가

# 래스터 연산

## • 클래스에 멤버변수 추가

클래스	엑세스 지정자	자료형	변수 이름
CProjectView	public	CPoint	m_ptMouse
CProjectView	public	CPoint	m_ptTarget

## • 이벤트 처리 함수 추가

메뉴 항목 ID	메시지	클래스	처리 함수
ID_START	COMMAND	CProjectView	OnStart( )
ID_END	COMMAND	CProjectView	OnEnd( )

클래스	메시지	처리 함수
CProjectView	WM_MOUSEMOVE	OnMouseMove( )
CProjectView	WM_TIMER	OnTimer( )

# 래스터 연산

---

- 함수의 코드 수정

```
void CProjectView::OnStart()
{
    SetTimer(1, 700, NULL);
}

void CProjectView::OnEnd()
{
    KillTimer(1);
}
```

```
void CProjectView::OnTimer(UINT_PTR nIDEvent)
{
    m_ptMouse = CPoint(rand() % 819, rand() % 614);
    Invalidate();
    CView::OnTimer(nIDEvent);
}
```

# 래스터 연산

---

```
01 void CProjectView::OnMouseMove(UINT nFlags, CPoint point)
02 {
03     if (nFlags & MK_LBUTTON)
04     {
05         m_ptTarget = point - CPoint(24,24);
06         Invalidate();
07     }
08     CView::OnMouseMove(nFlags, point);
09 }
```

# 래스터 연산

---

```
01 void CProjectView::OnDraw(CDC* pDC)
02 {
03     CProjectDoc* pDoc = GetDocument();
04     ASSERT_VALID(pDoc);
05     if (!pDoc)
06         return;
07     CDC mdc;
08     CBitmap back, mouse, mouse_mask;
09     CBitmap target, target_mask, *oldbit;
10     mdc.CreateCompatibleDC(pDC);
11     back.LoadBitmap(IDB_BACK);
12     mouse.LoadBitmapW(IDB_MOUSE);
13     mouse_mask.LoadBitmapW(IDB_MOUSE_MASK);
```

# 래스터 연산

```
14     target.LoadBitmapW(IDB_TARGET);
15     target_mask.LoadBitmapW(IDB_TARGET_MASK);
16     oldbit = mdc.SelectObject(&back);
17     pDC->BitBlt(0, 0, 819, 614, &mdc, 0, 0, SRCCOPY);
18     mdc.SelectObject(&mouse_mask);
19     pDC->BitBlt(m_ptMouse.x, m_ptMouse.y, 68, 49, &mdc, 0, 0, SRCAND);
20     mdc.SelectObject(&mouse);
21     pDC->BitBlt(m_ptMouse.x, m_ptMouse.y, 68, 49, &mdc, 0, 0, SRCPAINT);
22     mdc.SelectObject(&target_mask);
23     pDC->BitBlt(m_ptTarget.x, m_ptTarget.y, 48, 48, &mdc, 0, 0, SRCAND);
24     mdc.SelectObject(&target);
25     pDC->BitBlt(m_ptTarget.x, m_ptTarget.y, 48, 48, &mdc, 0, 0, SRCPAINT);
26     mdc.SelectObject(oldbit);
27     target.DeleteObject();
28     target_mask.DeleteObject();
29     back.DeleteObject();
30     mdc.DeleteDC();
31 }
```





# Thank You !

