

# Window Programming

Visual C++ MFC Programming

Lecture 10

김예진

Dept. of Game Software

# Announcement

---

- 03/16: HW #1 on ClassNet (Due: 03/24) → Avg: 8.1
- 04/06: HW #2 on ClassNet (Due: 04/15) → Avg: 6.3
- 04/27: Midterm 실기 시험 (5 문제, ~90 min.) → Avg: 4.3
  - Sample 문제 on ClassNet
- 05/08: HW #3 on ClassNet (Due: 05/15)

# Plan: 사용자 인터페이스 (UI)

---

- 메뉴 (Menu)
  - 연습: 메뉴에서 사각형 색상 바꾸기
  - 연습: 메뉴 항목 갱신
  - 연습: 메뉴 만들어 붙이기
- 툴바 (Toolbar)
  - 연습: 툴바에서 도형 색상 바꾸기
  - 연습: 툴바에서 도형 모양 바꾸기
- 상태바 (Status Bar)
  - 연습: 상태바에 마우스 위치 표시

# 사용자 인터페이스 (UI) 메뉴 (Menu)



## THE MENU

### LUNCH & DINNER



### BREAKFAST



### DESSERT



### WINE LIST

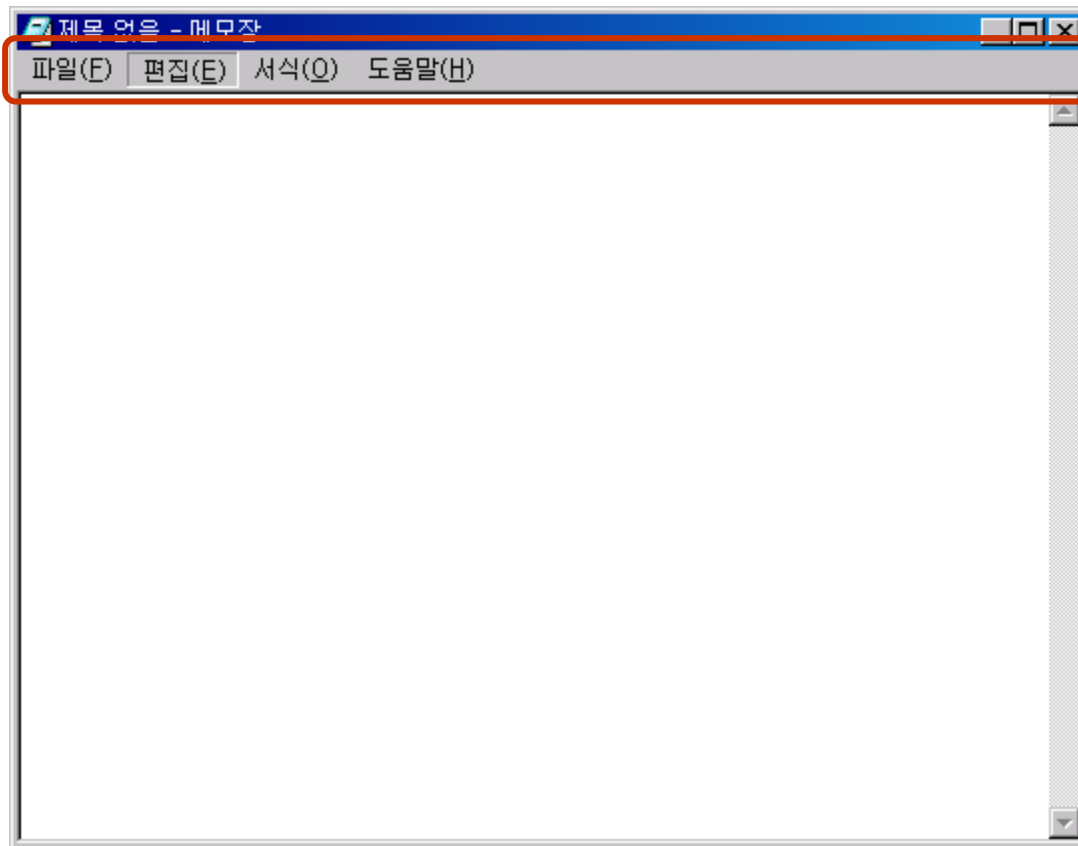


### SOUP OF THE DAY



# 메뉴 용어 (1/6)

- 최상위 메뉴 = 메뉴 바

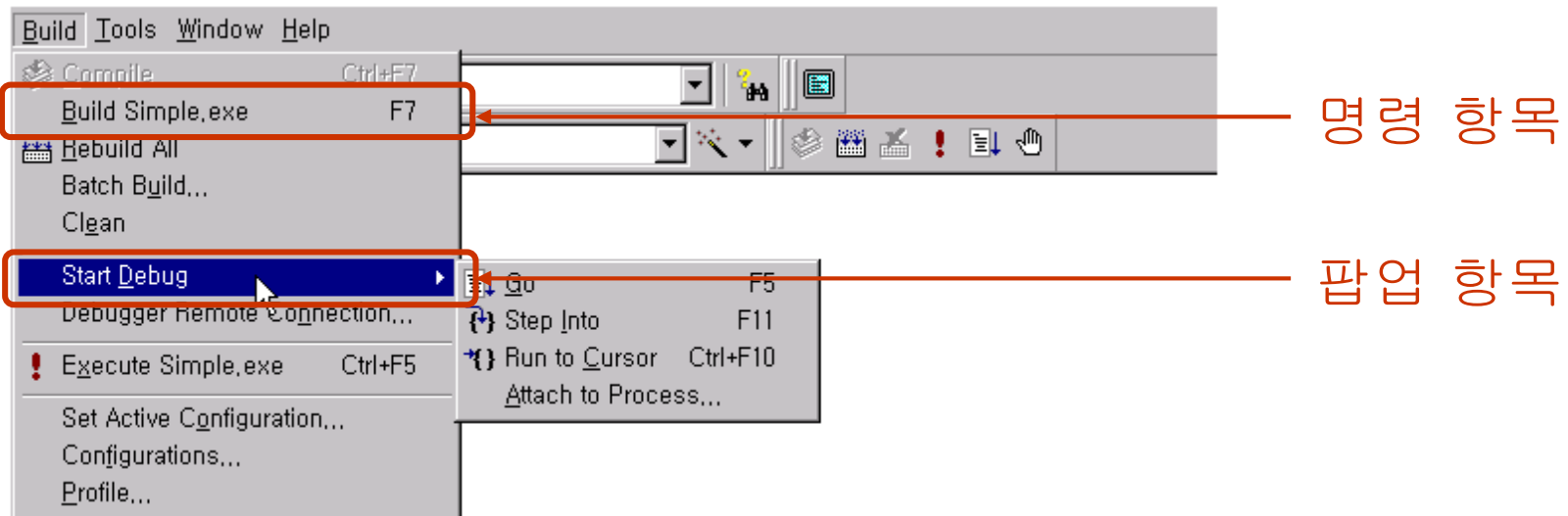


최상위 메뉴  
= 메뉴 바

# 메뉴 용어 (2/6)

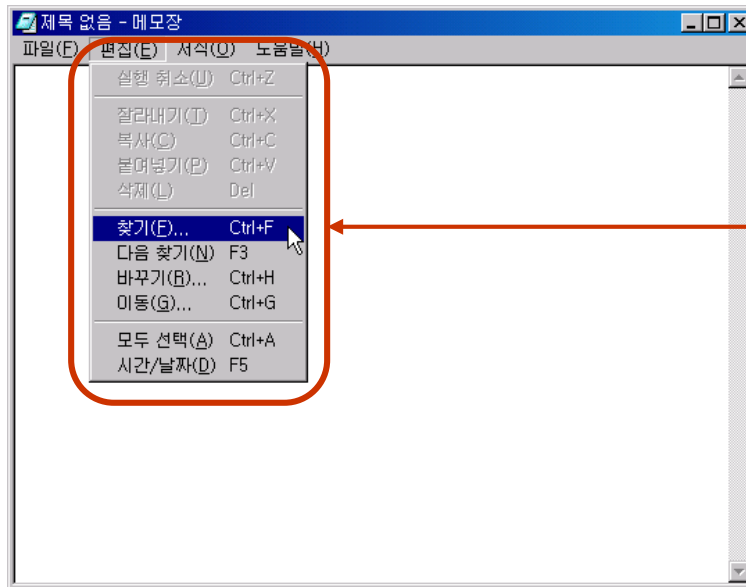
- 메뉴 항목

용어	의미
① 명령 항목	명령(Command)을 수행하는 메뉴 항목. 선택하면 WM_COMMAND 메시지가 발생한다.
② 팝업 항목	하위 메뉴를 화면에 표시하는 메뉴 항목. 선택해도 WM_COMMAND 메시지가 발생하지 않는다.



# 메뉴 용어 (3/6)

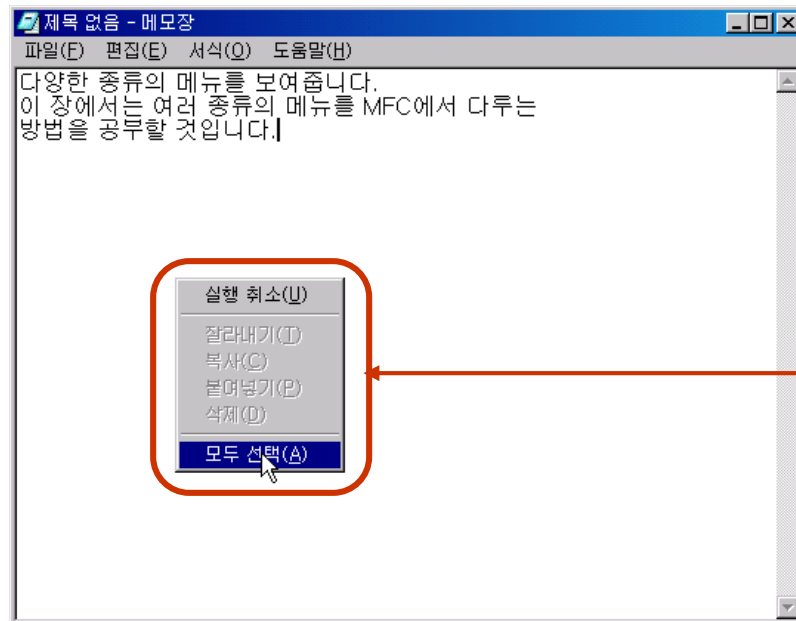
- Drop-down 메뉴
  - 최상위 메뉴 항목을 클릭했을 때 펼쳐지는 메뉴



Drop-down 메뉴  
= Pop-up 메뉴

# 메뉴 용어 (4/6)

- 컨텍스트 메뉴 = 단축 메뉴
  - 마우스 오른쪽 버튼을 누를 때 열리는 메뉴
  - 마우스 커서의 위치 또는 현재 작업하고 있는 내용에 따라 서로 다른 메뉴 항목이 표시됨

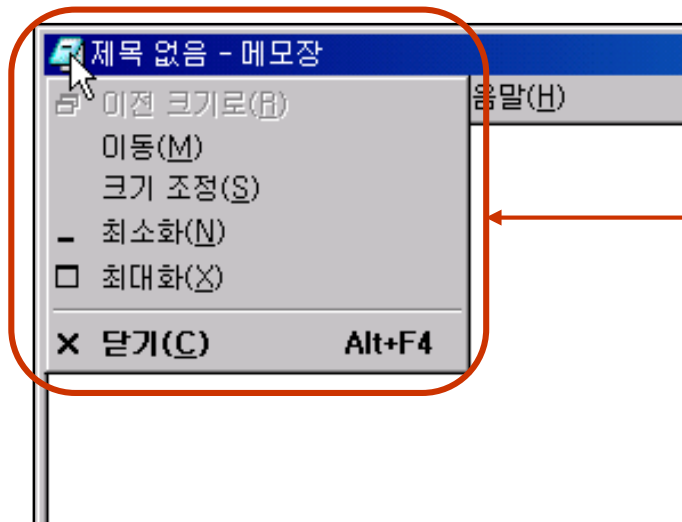


컨텍스트 메뉴  
= 단축 메뉴  
= 팝업 메뉴



# 메뉴 용어 (5/6)

- 시스템 메뉴 = 윈도우 메뉴



시스템 메뉴  
= 윈도우 메뉴

# 메뉴 용어 (6/6)

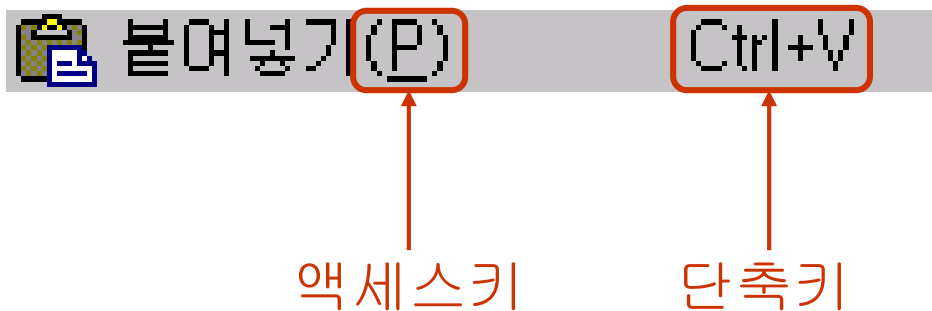
- 액세스키와 단축키

- 액세스키

- 메뉴가 열린 상태에서 특정 항목을 키보드로 빠르게 선택

- 단축키

- 메뉴가 열리지 않은 상태에서도 키 조합으로 메뉴 항목의 기능을 곧바로 실행



# 메뉴 클래스

- MFC 클래스

CObject

CCmdUI

└─ CMenu

- CMenu

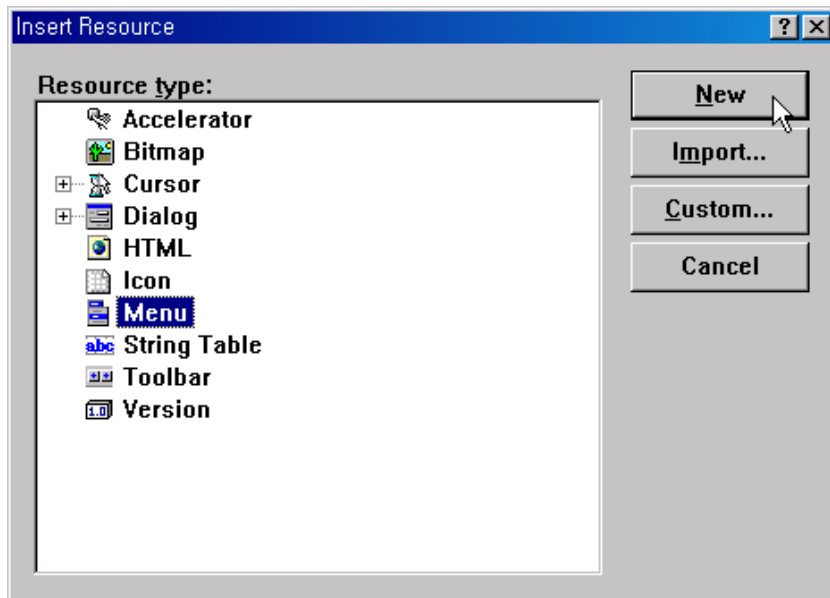
- 메뉴를 다룰 수 있는 다양한 함수 제공

- CCmdUI

- CObject의 파생 클래스가 아닌 독립된 클래스
    - 메뉴와 툴바 및 상태바에 필요한 기능 지원

# 메뉴 생성 (1/6)

- 두 가지 방법
  - 메뉴 리소스
  - 프로그램 실행 중 메뉴 생성
- 메뉴 리소스 작성



# 메뉴 생성 (2/6)

- AppWizard가 생성한 코드

```
BOOL CSimple2App::InitInstance()
{
    CMainFrame* pFrame = new CMainFrame;
    m_pMainWnd = pFrame;

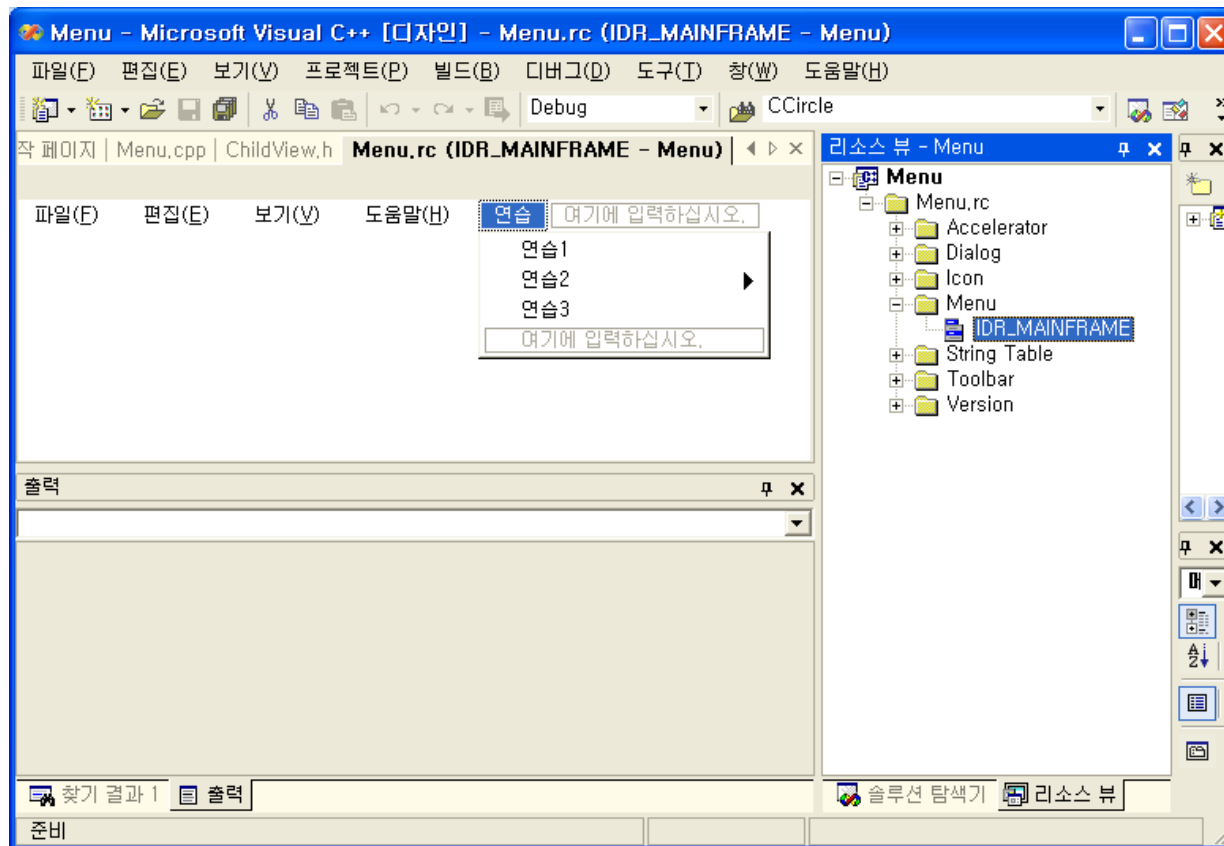
    pFrame->LoadFrame(IDR_MAINFRAME,
        WS_OVERLAPPEDWINDOW | FWS_ADDTOTITLE, NULL,
        NULL);

    pFrame->ShowWindow(SW_SHOW);
    pFrame->UpdateWindow();

    return TRUE;
}
```

# 메뉴 생성 실습 (3/6)

- 리소스 뷰 사용



# 메뉴 생성 (4/6)

- 메뉴 항목 속성

속성	의미
ID	내부적으로 메뉴 항목을 구분하는 번호이며 일반적으로 ID_메뉴이름_항목이름 형태로 만든다. 예) ID_EDIT_CUT
Caption	화면에 표시되는 문자열로 액세스키를 지정하려면 해당 문자 앞에 '&' 기호를 사용한다. 단축키를 사용할 경우 'Wt' 기호를 삽입하여 단축키를 나타내는 문자열이 탭 위치에 정렬되도록 한다. 예) 잘라내기(&T)WtCtrl+X
Separator	메뉴 항목을 구분하는 가로줄이 표시된다.
Pop-up	설정하면 명령 항목이 아닌 팝업 항목이 된다. 최상위 메뉴는 대개 Pop-up 속성을 가진다.

# 메뉴 생성 (5/6)

- 메뉴 항목 속성 (cont'd)

속성	의미
Inactive	메뉴 항목이 표시되지만 사용하지는 못한다.
Break	일반적으로 메뉴 항목은 하나의 열(Column)에 표시되지만 항목의 개수가 많을 경우 두 개 이상의 열에 표시되게 할 수 있다. Break 속성으로 Column 또는 Bar를 선택하면 이때부터는 다음 열에 메뉴 항목이 표시된다. Column과 Bar 속성은 기본적으로 같은 기능을 하지만 Bar 속성을 선택하면 열 구분선(세로줄)이 생긴다.
Checked	메뉴 항목의 왼쪽에 체크 표시를 한다.



# 메뉴 생성 (6/6)

- 메뉴 항목 속성 (cont'd)

속성	의미
Grayed	메뉴 항목이 흐리게 표시되어 현재 사용할 수 없음을 나타낸다.
Help	윈도우의 오른쪽 끝 위치에 메뉴가 표시되도록 한다. 주로 Help 메뉴 항목에 이 속성을 설정한다.
Prompt	MFC로 작성한 프로그램에서만 사용할 수 있는 속성으로, 툴바와 상태바에 표시될 문자열을 나타낸다. '\n'을 기준으로 앞쪽 문자열은 상태바에 표시되며 뒤쪽 문자열은 툴팁에 표시된다. 예) 선택 부분을 잘라내어 클립보드에 넣습니다\n잘라내기

# 메뉴 명령 처리 (1/4)

---

- 메뉴 명령 처리 과정
  - ① 명령 항목을 마우스나 키보드로 선택
  - ② **WM\_COMMAND** 메시지 발생
  - ③ WM\_COMMAND 메시지 핸들러에서 메뉴 명령 처리

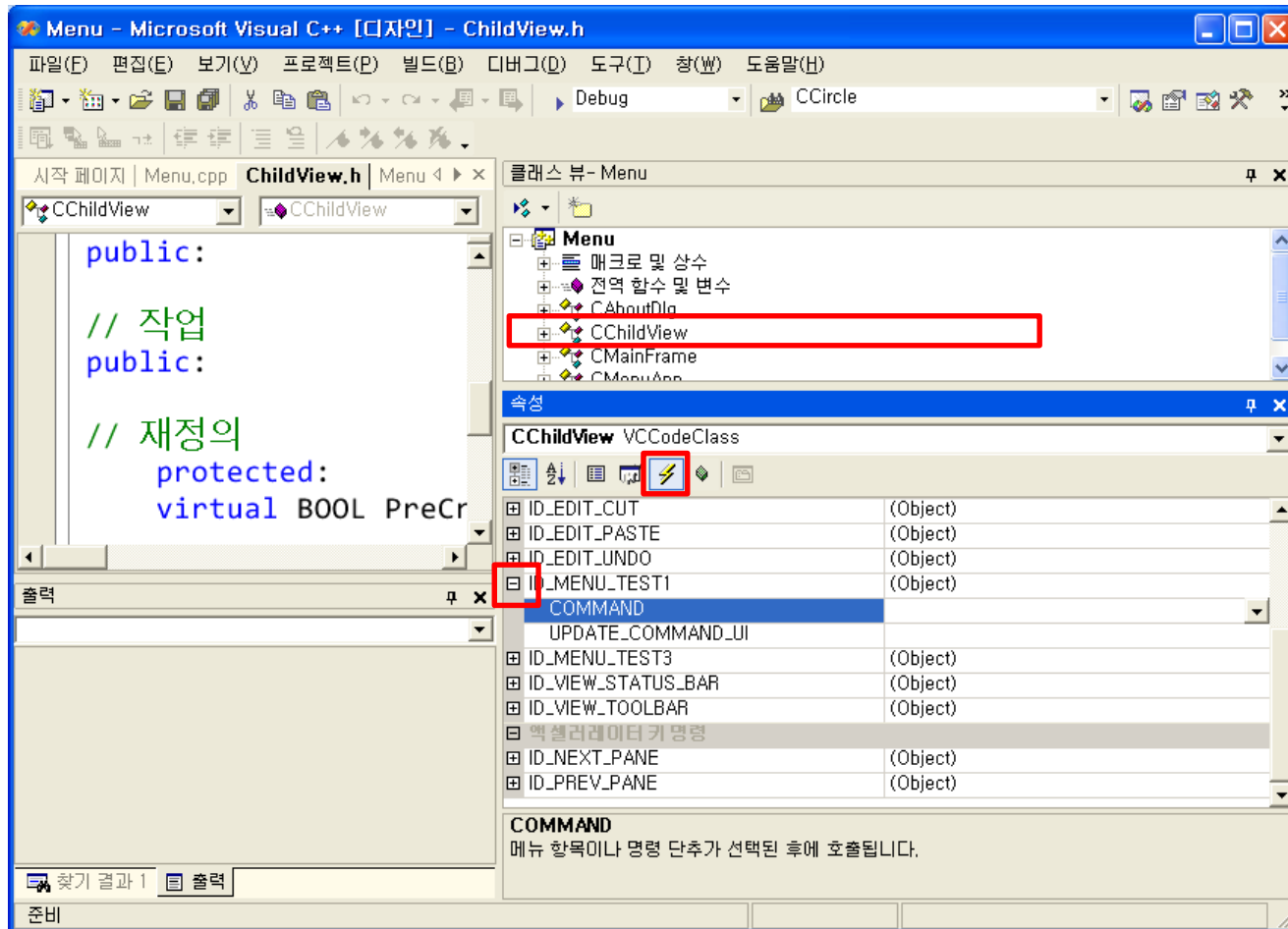
# 메뉴 명령 처리 (2/4)

---

- MFC의 메뉴 명령 처리 방법
  - 각각의 메뉴 항목에 대해 함수를 따로 작성
    - 명령 핸들러(Command Handler)
    - `ON_COMMAND(메뉴ID, 함수명)` 매크로를 이용하여 메뉴 항목과 함수 연결
- 명령 라우팅
  - 명령 핸들러를 작성하는 위치에 관계없이 처리

# 메뉴 명령 처리 (3/4)

- 속성 창을 이용



# 메뉴 명령 처리 (4/4)

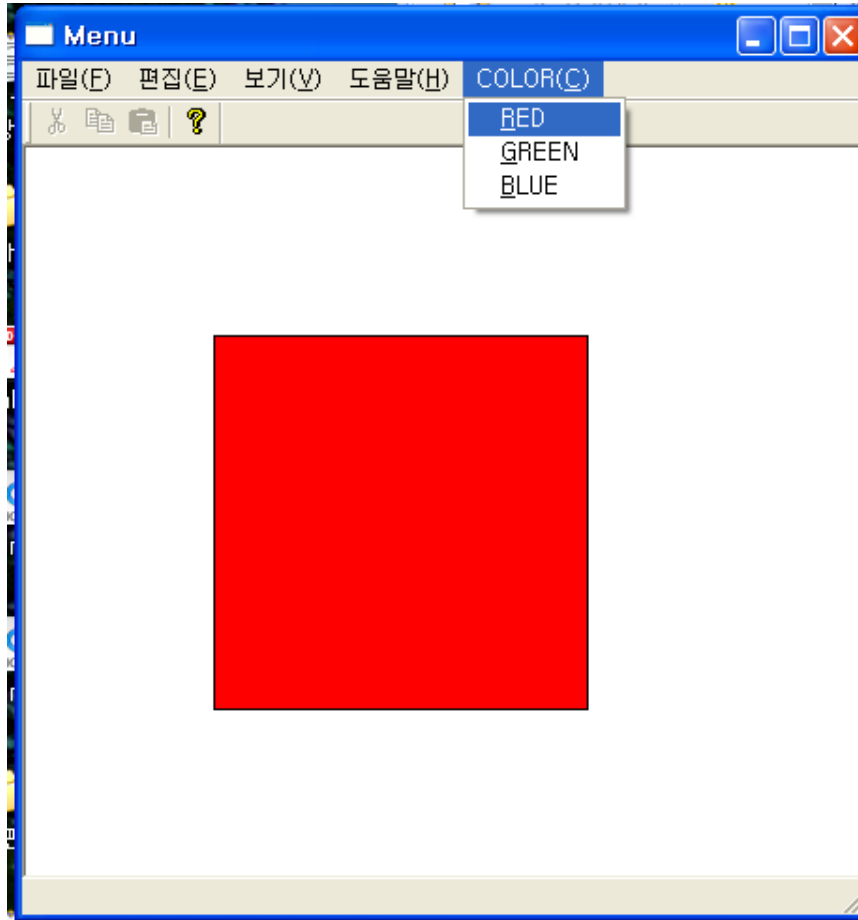
- 메뉴 명령 처리 예

```
BEGIN_MESSAGE_MAP(CChildView,CWnd )
    ...
    ON_COMMAND(ID_COLOR_RED, OnColorRed)
    ON_COMMAND(ID_COLOR_GREEN, OnColorGreen)
    ON_COMMAND(ID_COLOR_BLUE, OnColorBlue)
    ...
END_MESSAGE_MAP()

void CChildView::OnColorRed()
{
    m_color = RGB(255, 0, 0);
    Invalidate();
}
```

# 연습: 메뉴에서 사각형 색상 바꾸기

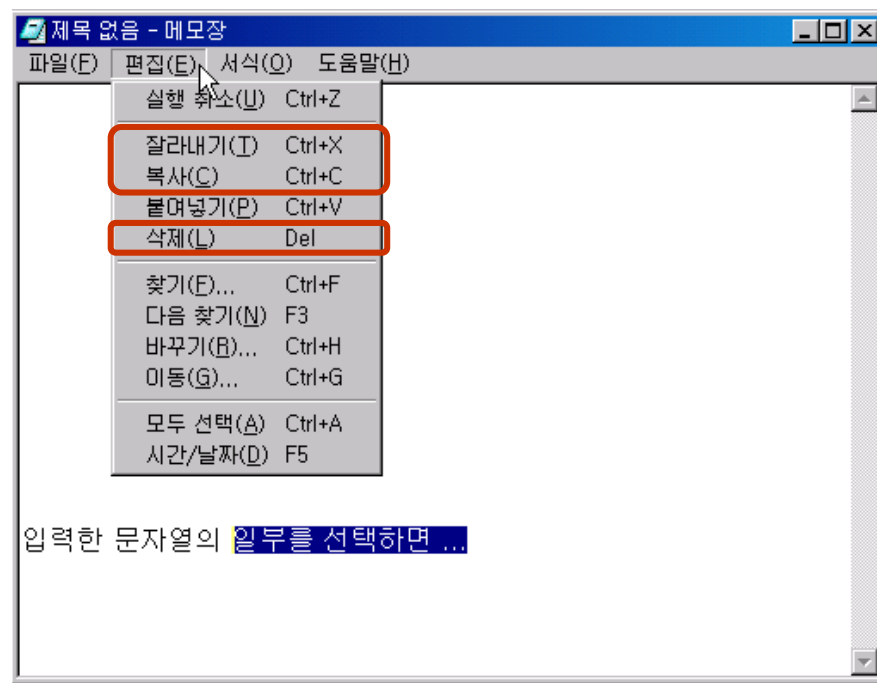
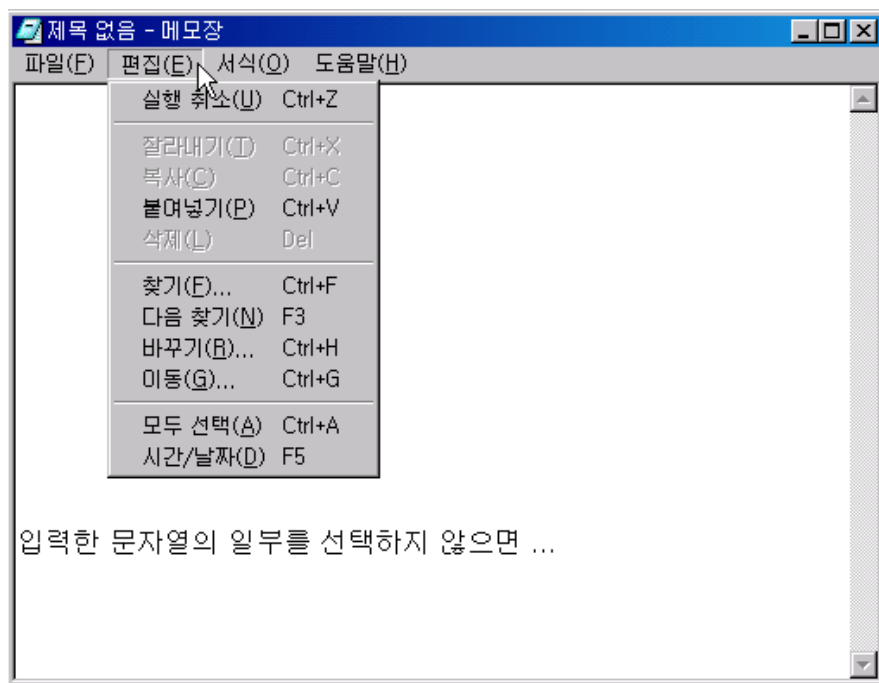
- 사각형을 그리고 메뉴를 통해 색 변경



1. Color를 저장할 변수 선언:  
`COLORREF m_color`
2. OnPaint에서 brush 설정:  
`CBrush brush(m_color);`  
`dc.SelectObject(&brush);`
3. 사각형 그리기
4. 메뉴 만들기
5. 메뉴에 핸들러 함수 연결
6. `m_color` 값 변경

# 메뉴 항목 갱신 (1/5)

- 메뉴 항목 갱신 예



# 메뉴 항목 갱신 (2/5)

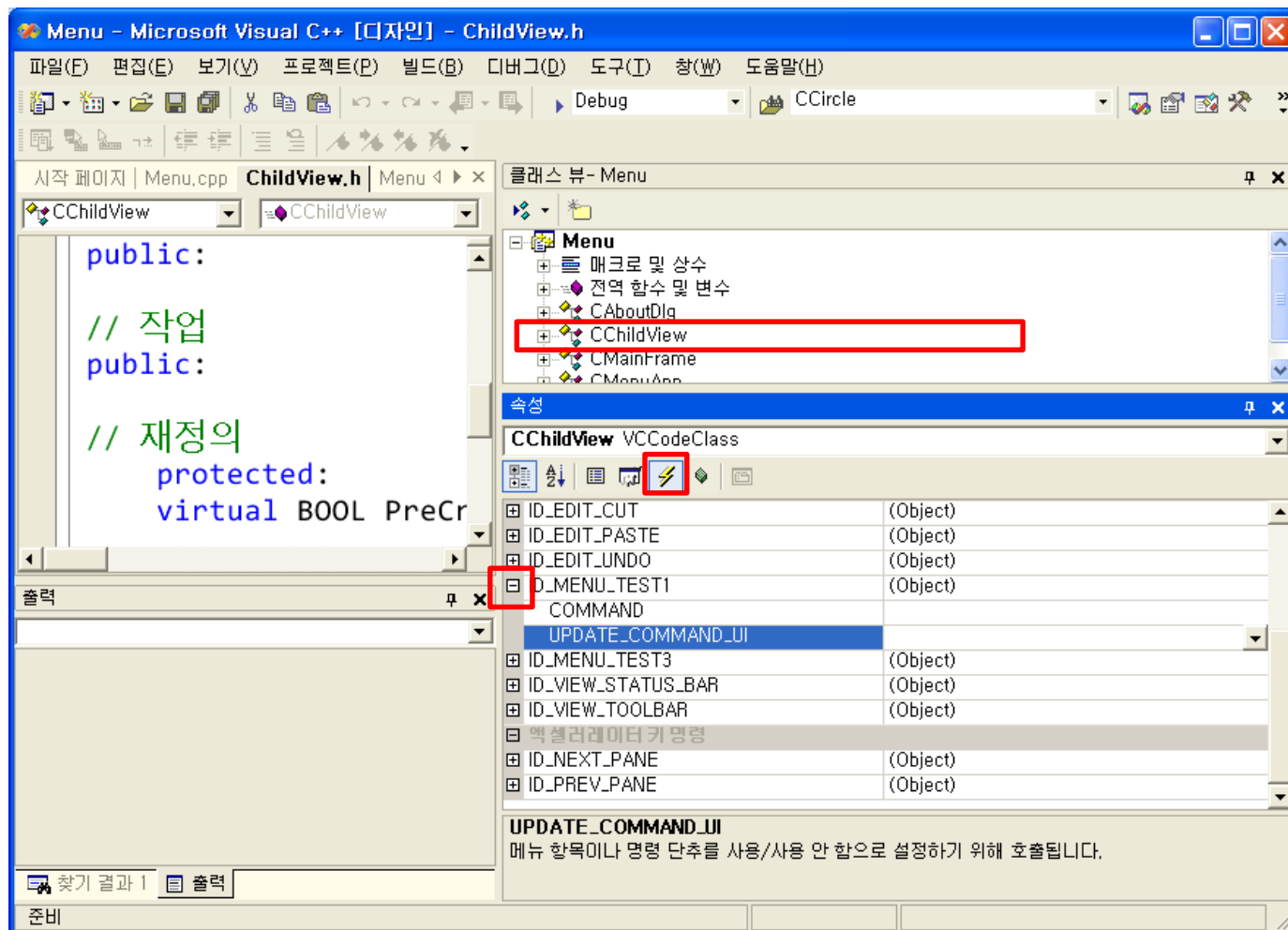
---

- MFC의 메뉴 항목 갱신 방법
  - 각각의 메뉴 항목에 대해 함수를 따로 작성
    - 명령 갱신 핸들러(Command Update Handler)
    - `ON_UPDATE_COMMAND_UI`(메뉴ID, 함수명) 매크로를 이용하여 메뉴 항목과 함수 연결
- 명령 라우팅
  - 명령 핸들러와 마찬가지로 명령 갱신 핸들러도 작성하는 위치에 관계없이 처리



# 메뉴 항목 갱신 (3/5)

- 속성 창을 이용



# 메뉴 항목 갱신 (4/5)

- 메뉴 항목 갱신 예

```
BEGIN_MESSAGE_MAP(CChildView,CWnd )
    ...
    ON_UPDATE_COMMAND_UI(ID_COLOR_RED, OnUpdateColorRed)
    ON_UPDATE_COMMAND_UI(ID_COLOR_GREEN, OnUpdateColorGreen)
    ON_UPDATE_COMMAND_UI(ID_COLOR_BLUE, OnUpdateColorBlue)
    ...
END_MESSAGE_MAP()

void CChildView::OnUpdateColorRed(CCmdUI* pCmdUI)
{
    pCmdUI->SetCheck(m_color == RGB(255, 0, 0));
}
```

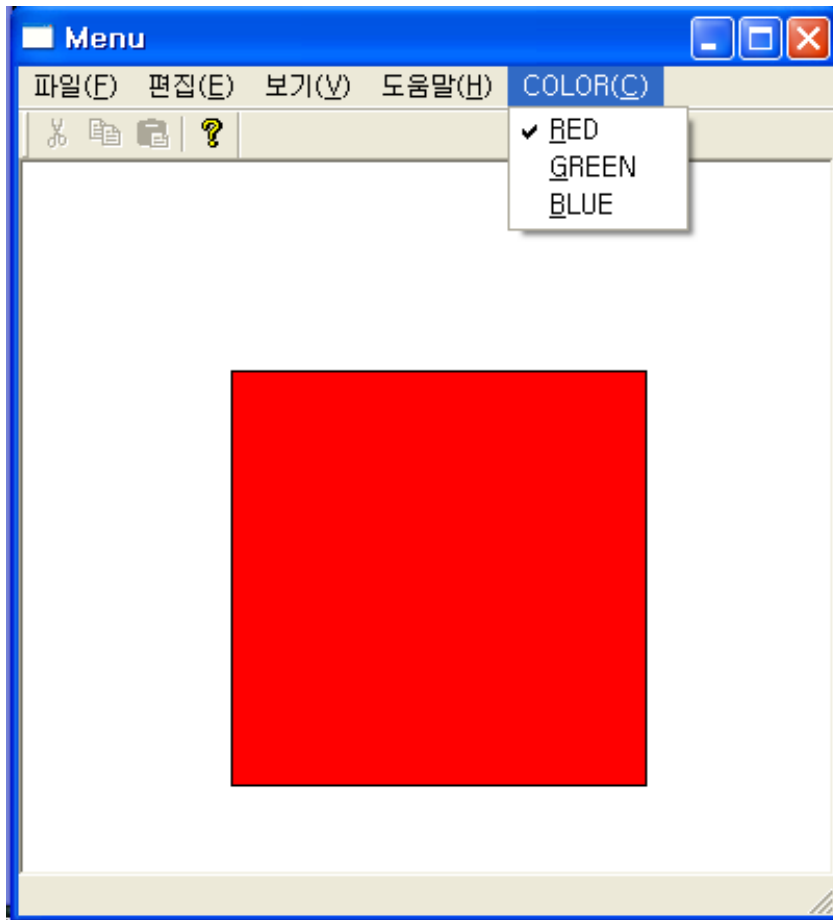
# 메뉴 항목 갱신 (5/5)

- CCmdUI 클래스 멤버 함수

멤버 함수	의미	사용 예
Enable()	활성화 상태 변경	pCmdUI->Enable(b_DrawMode);
SetCheck()	체크 상태 변경	pCmdUI->SetCheck(m_color == RGB(255, 0, 0));
SetRadio()	라디오 표시 상태 변경	pCmdUI->SetRadio(m_color == RGB(255, 0, 0));
SetText()	문자열 변경	pCmdUI->SetText("Light On");

# 연습: 메뉴 항목 갱신

- Update Command UI 실습



1. UpdateCommand 핸들러 연결

2. pCmdUI->SetCheck(...) 구현:

```
if (m_color==RGB(0,0,255))
    pCmdUI->SetCheck(true);
else
    pCmdUI->SetCheck(false);
```

# Menu 만들기의 2가지 방법

---

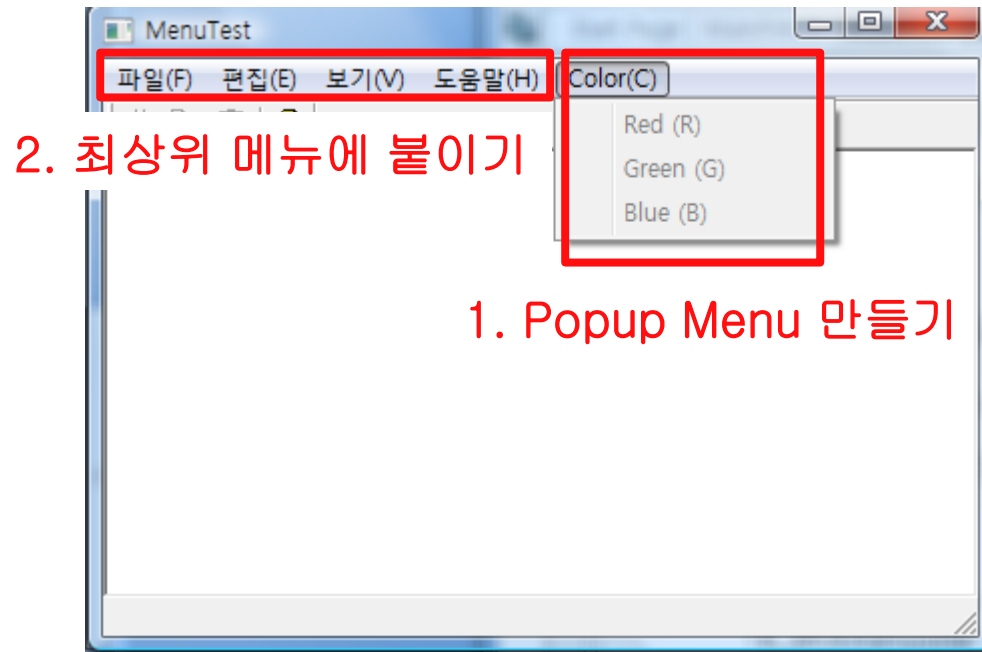
- Resource를 이용하기
- 직접 Coding을 통해 만들기
  - CMenu 객체를 활용

CObject

└ CMenu

# CMenu를 이용하여 Menu 만들기

1. Popup Menu 만들기
2. Popup Menu를 최상위 Menu에 붙이기



# CMenu를 이용하여 Menu 만들기

---

## 1. 붙일 팝업 메뉴 CMenu 객체 생성

- 새로 PopupMenu를 생성: `CreatePopupMenu()`
- 팝업 메뉴에 항목 추가
  - `AppendMenu()` 함수 이용

## 2. 최상위 CMenu 객체에 붙이기

- 최상위 CMenu 객체 생성
  - 새로 만들기 : `CreateMenu()`
  - 현재 메뉴의 포인터를 가져오기 : `GetMenu()`
- 팝업 메뉴를 최상위 메뉴에 연결
  - `AppendMenu()` 함수 이용

# 연습: 메뉴 만들어 붙이기

- Menu는 MainFrame 클래스의 **OnCreate** 속에...

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    // 생략...
    CMenu Popup;
    Popup.CreatePopupMenu();
    Popup.AppendMenu(MF_STRING, 201, _T("Red(&R)"));
    Popup.AppendMenu(MF_STRING, 202, _T("Green(&G)"));
    Popup.AppendMenu(MF_STRING, 203, _T("Blue(&B)"));

    CMenu *pMenuMain = GetMenu();
    pMenuMain->AppendMenu(MF_POPUP,
        (UINT_PTR) Popup.Detatch(), _T("Color(&C)"));
}
```



# 정리

- CMenu 객체: 메뉴의 생긴 모양, ID등을 생성
- 멤버함수: AppendMenu()

```
bool CMenu::AppendMenu( Flag, ID, Caption )
```

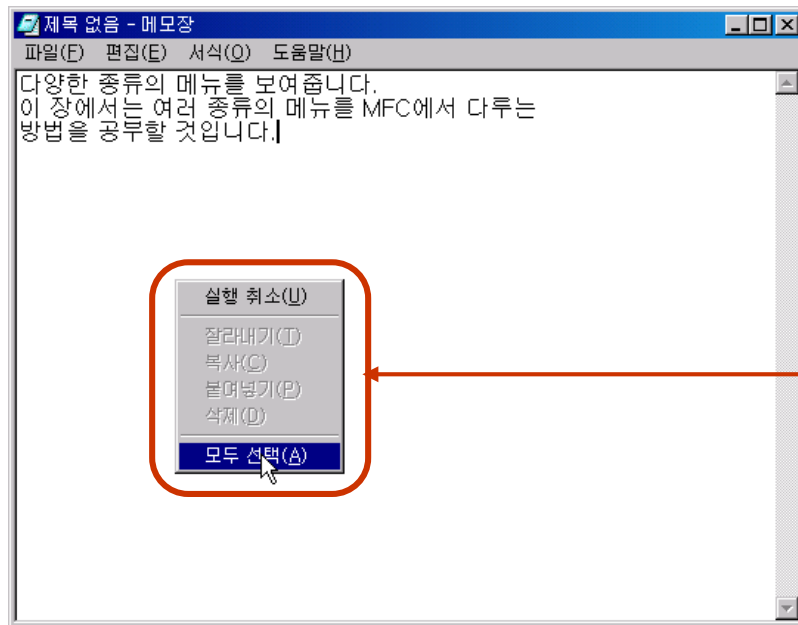
– Flag:

- MF\_STRING (일반적인 메뉴 항목)
- MF\_POPUP (하위 메뉴 연결 시)
- MF\_SEPARATOR

– ID:

- MF\_STRING 의 경우: Command ID
- MF\_POPUP 의 경우: 붙일 하위 메뉴의 주소 (Detach() 함수이용)


# 컨텍스트 메뉴



컨텍스트 메뉴  
= 단축 메뉴  
= 팝업 메뉴


# 컨텍스트 메뉴 동작 순서

---

1. 마우스 오른쪽 버튼 클릭 또는  누름
2. WM\_CONTEXTMENU 메시지 발생
3. 메시지 핸들러에서 메뉴를 띄우는 코드 작성

# 컨텍스트 메뉴 (1/6)

---

- WM\_CONTEXTMENU 메시지 발생 상황
  - 마우스 오른쪽 버튼을 클릭하는 경우  
(WM\_RBUTTONDOWN 핸들러가 없는 경우)
  - Shift + F10 키 조합을 누른 경우
  - 가상 키코드 VK\_APPS에 해당하는 키(를 누른 경우

# 컨텍스트 메뉴 (2/6)

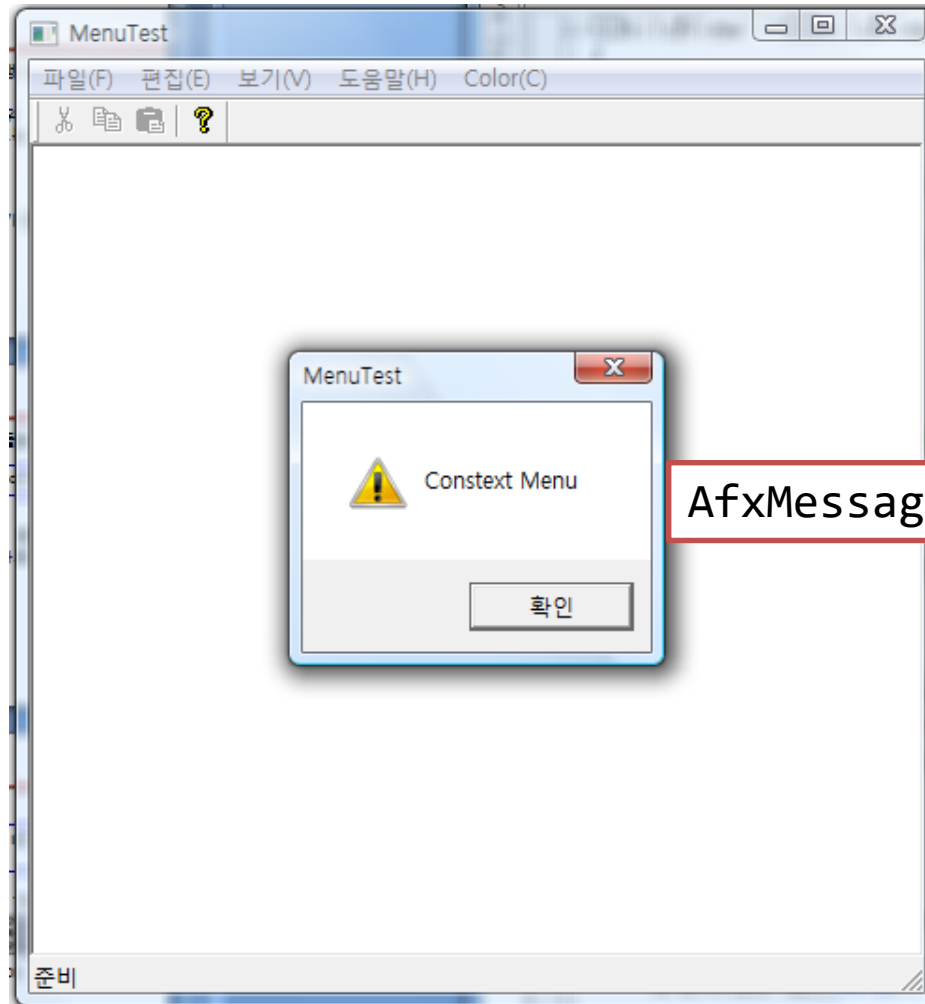
- WM\_CONTEXTMENU 메시지 핸들러

```
afx_msg void OnContextMenu(CWnd* pWnd, CPoint pos);
```

- pWnd - 마우스 커서 아래쪽에 있는 윈도우
- pos - 마우스 커서의 위치(스크린 좌표)

# 컨텍스트 메뉴 핸들러 연습

- WM\_CONTEXTMENU 핸들러를 만들어 보자



```
AfxMessageBox(_T("Context Menu"));
```

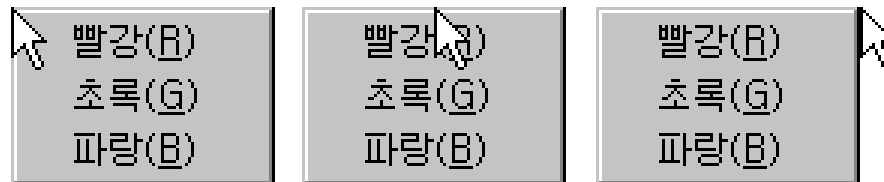
# 컨텍스트 메뉴 (3/6)

- CMenu::TrackPopupMenu() 함수

```
BOOL TrackPopupMenu (UINT nFlags, int x, int y, CWnd* pWnd,  
LPCRECT lpRect = 0) ;
```

– nFlags

- TPM\_LEFTALIGN, TPM\_CENTERALIGN, TPM\_RIGHTALIGN



- TPM\_LEFTBUTTON, TPM\_RIGHTBUTTON

# 컨텍스트 메뉴 (4/6)

- CMenu::TrackPopupMenu() 함수

```
BOOL TrackPopupMenu (UINT nFlags, int x, int y, CWnd* pWnd,  
    LPCRECT lpRect = 0) ;
```

- x, y
  - 컨텍스트 메뉴가 표시될 위치(스크린 좌표)
- pWnd
  - 컨텍스트 메뉴의 WM\_COMMAND 메시지를 받을 윈도우
  - **AfxGetMainWnd() 함수** 를 사용하여 창과 연결
- lpRect
  - 마우스 버튼을 클릭하더라도 컨텍스트 메뉴가 닫히지 않는 사각형 영역(스크린 좌표)



# 컨텍스트 메뉴 (5/6)

- 컨텍스트 메뉴 사용 예

```
void CChildView::OnContextMenu(CWnd* pWnd, CPoint point)
{
    CMenu menuPopup;

    menuPopup.CreatePopupMenu();
    menuPopup.AppendMenu(MF_STRING, 201, _T("Red (&R)"));
    menuPopup.AppendMenu(MF_STRING, 202, _T("Green (&G)"));
    menuPopup.AppendMenu(MF_STRING, 203, _T("Blue (&B)"));

    menuPopup.TrackPopupMenu(
        TPM_LEFTALIGN|TPM_LEFTBUTTON,
        point.x, point.y,
        AfxGetMainWnd());
}
```

# 컨텍스트 메뉴 (6/6)

- 컨텍스트 메뉴 사용 예

```
void CChildView::OnContextMenu(CWnd* pWnd, CPoint point)
{
    CMenu menu;
    menu.LoadMenu(IDR_MAINFRAME);

    CMenu* pMenu = menu.GetSubMenu(4);

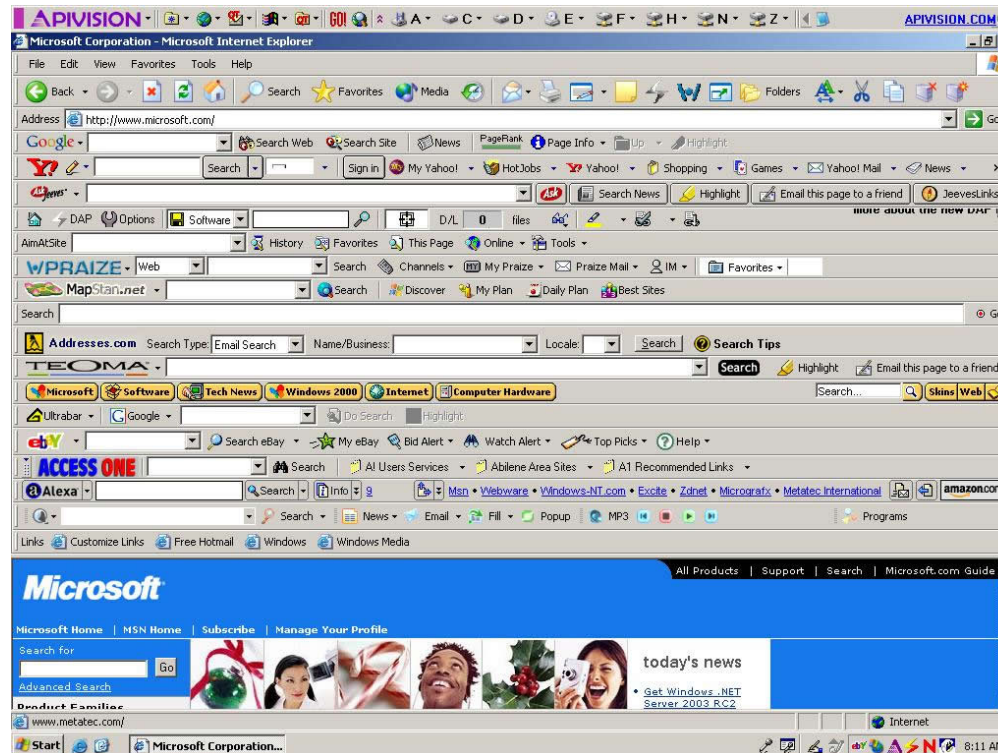
    pMenu->TrackPopupMenu(
        TPM_LEFTALIGN|TPM_RIGHTBUTTON,
        point.x, point.y, AfxGetMainWnd());
}
```

# 이 밖의 메뉴 및 관련사항

---

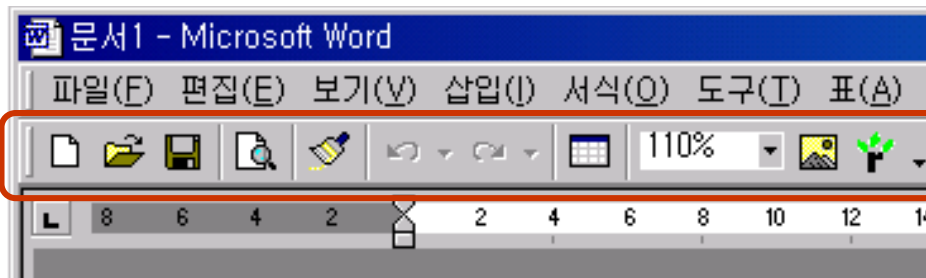
- 교재 p252 ~ 277
  - 시스템 메뉴
  - 가속기(=Accelerator, Hot Key)
  - 트레이 아이콘 및 메뉴

# 사용자 인터페이스 (UI) 툴바 (Toolbar)

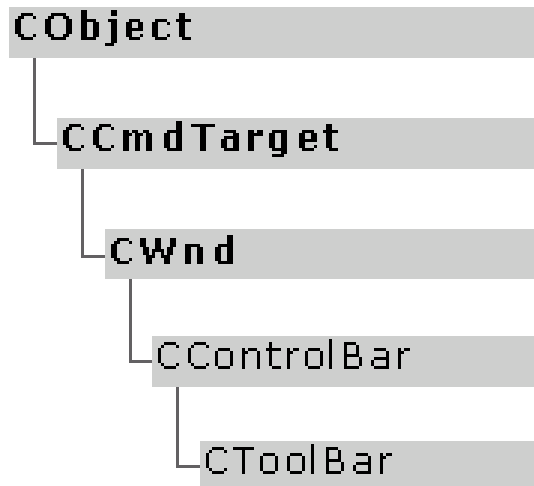


# 툴바 (1/4)

- 툴바

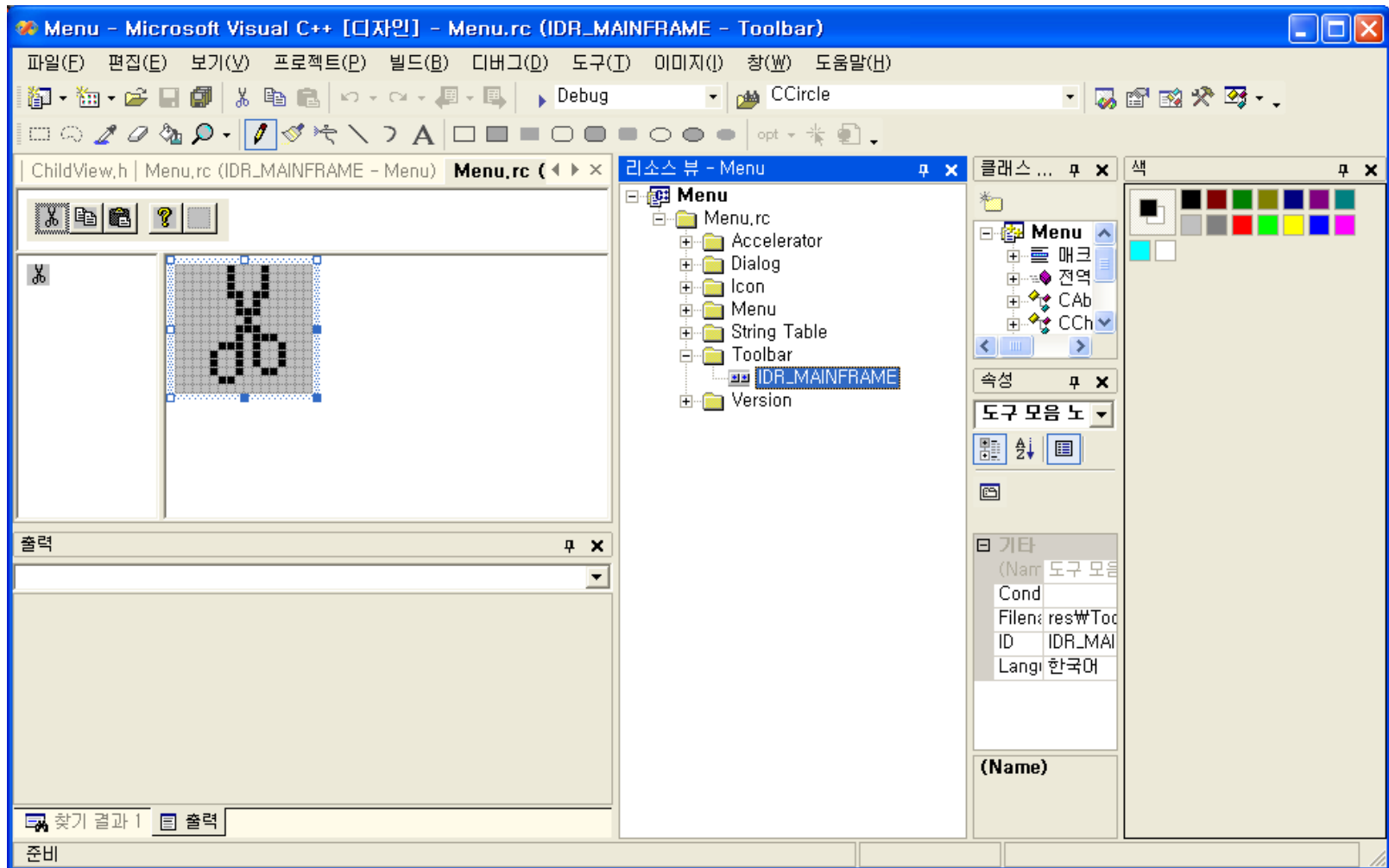


선택하면  
WM\_COMMAND  
메시지 발생!



# 툴바 (2/4)

- 툴바 리소스



# 툴바 (3/4)

- 툴바 코드

```
class CMainFrame : public CFrameWnd
{
    // 생략 ...
protected:
    CStatusBar m_wndStatusBar;
    CToolBar m_wndToolBar;
    CChildView m_wndView;
    // 생략 ...
};

int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    // 생략 ...
    if (!m_wndToolBar.CreateEx(this, TBSTYLE_FLAT, WS_CHILD |
        WS_VISIBLE | CBRS_TOP | CBRS_GRIPPER | CBRS_TOOLTIPS |
        CBRS_FLYBY | CBRS_SIZE_DYNAMIC) ||
        !m_wndToolBar.LoadToolBar(IDR_MAINFRAME))
```

# 툴바 (4/4)

- 툴바 코드 (cont'd)

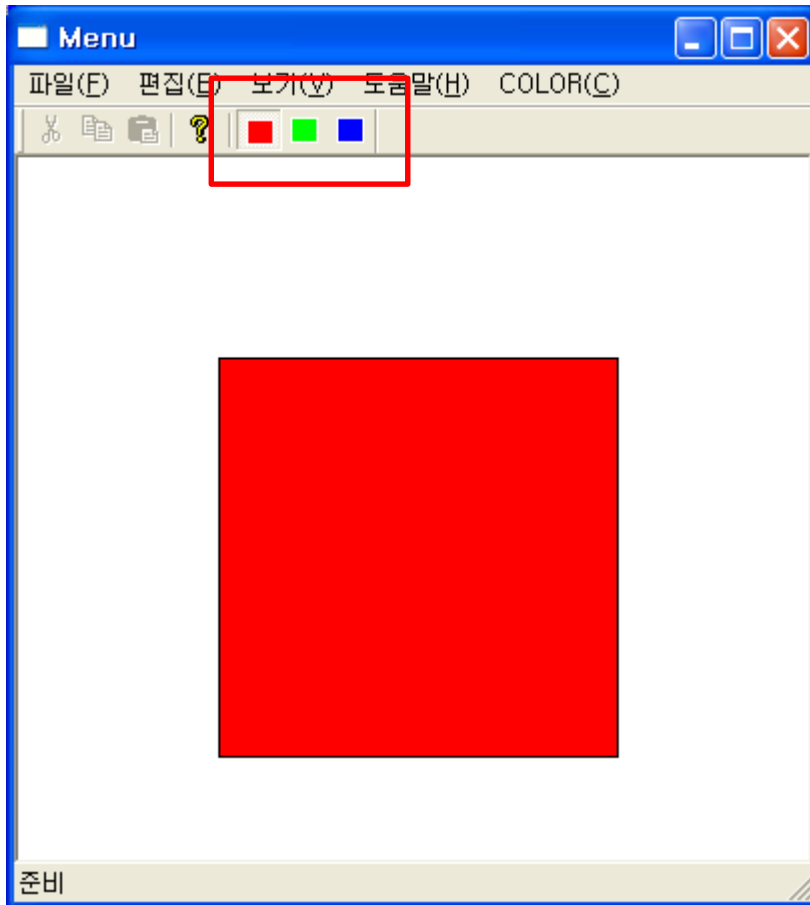
```
{
    TRACE0("Failed to create toolbar\n");
    return -1;
}
// 생략 ...
m_wndToolBar.EnableDocking(CBRS_ALIGN_ANY);
EnableDocking(CBRS_ALIGN_ANY);
DockControlBar(&m_wndToolBar);

return 0;
}
```



# 연습: 툴바에서 도형 색상 바꾸기

- color menu의 내용을 toolbar와 연결시킨다

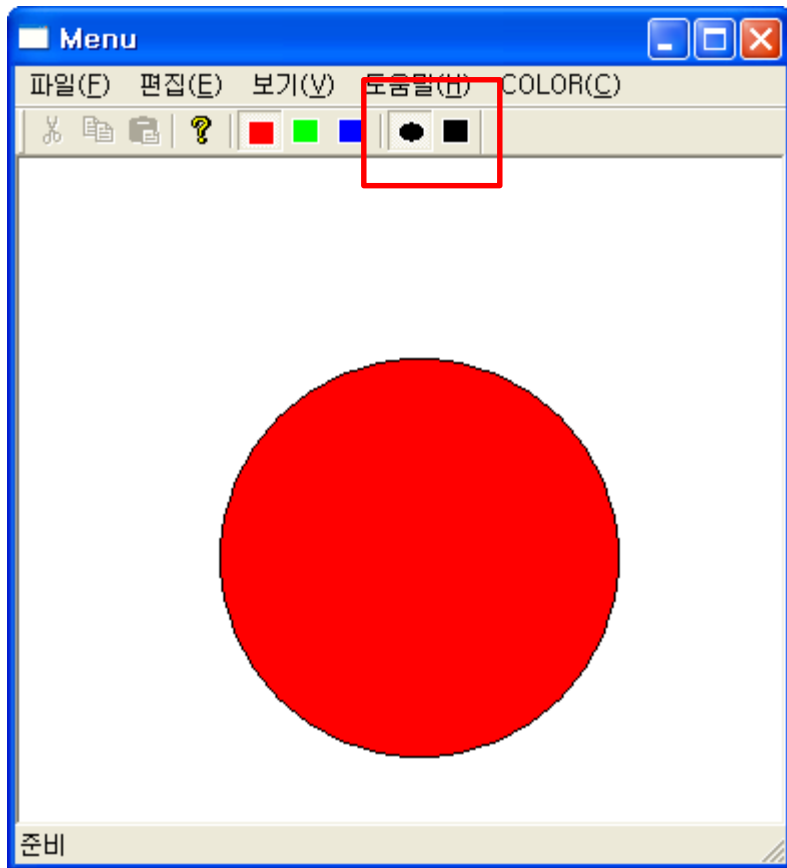


1. ToolBar에 RGB 버튼 추가

2. 각 버튼에 ID를 menu의 ID와  
똑같이 설정  
예) ID\_COLOR\_RED

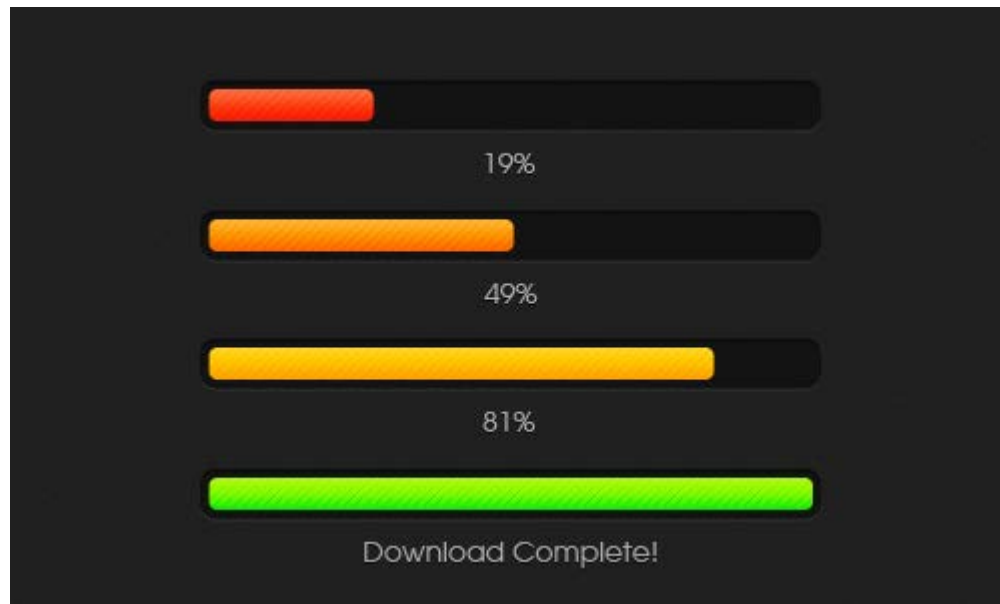
# 연습: 툴바에서 도형 모양 바꾸기

- Menu에 없는 작업을 toolbar에서 하고 싶다면...
  - 직접 메시지를 만들고 연결



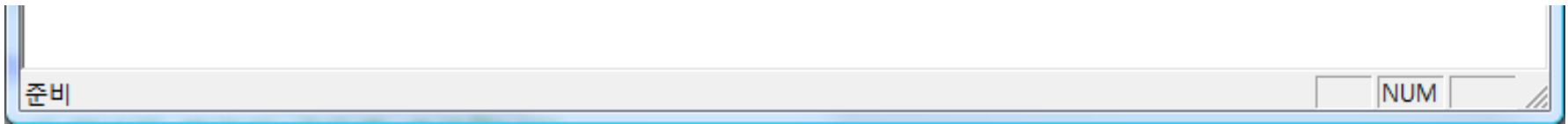
1. Toolbar에 타원, 사각형 버튼 추가
2. MessageMap 추가:  
`ON_COMMAND(...)`
3. 함수 정의/선언 추가:  
`afx_msg void OnSelectCircle();`

# 사용자 인터페이스 (UI) 상태바 (Status Bar)

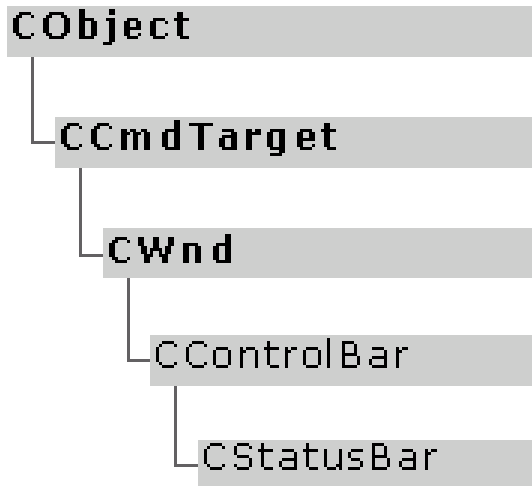


# 상태바 (1/4)

- 상태바

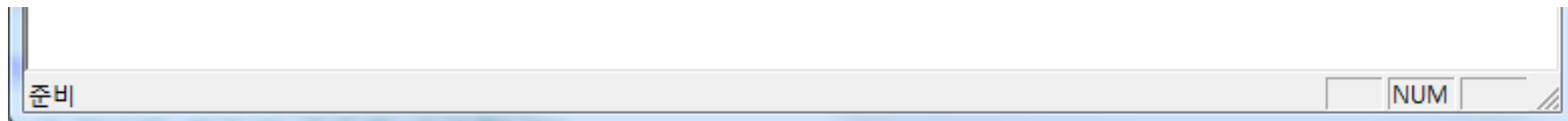


- MFC 클래스



# 상태바 (2/4)

- 상태바 리소스



```
static UINT indicators[ ] =  
{  
    ID_SEPARATOR,  
    ID_INDICATOR_CAPS,  
    ID_INDICATOR_NUM,  
    ID_INDICATOR_SCRL,  
};
```

Simple2 resources		ID_EDIT_REPLACE	57641	지정한
Accelerator		ID_EDIT_SELECT_ALL	57642	문서 전
Dialog		ID_EDIT_UNDO	57643	마지막
Icon		ID_EDIT_REDO	57644	이전에
Menu		ID_WINDOW_SPLIT	57653	현재 열
String Table		ID_APP_ABOUT	57664	프로그램
String Table	abc	ID_APP_EXIT	57665	응용 프
Toolbar		ID_NEXT_PANE	57680	다음 창
Version		ID_PREV_PANE	57681	이전 창
		ID_INDICATOR_EXT	59136	EXT
		ID_INDICATOR_CAPS	59137	CAP
		ID_INDICATOR_NUM	59138	NUM
		ID_INDICATOR_SCRL	59139	SCRL
		ID_INDICATOR_OVR	59140	OVR
		ID_INDICATOR_REC	59141	REC

# 상태바 (3/4)

- 상태바 코드

```
class CMainFrame : public CFrameWnd
{
// 생략 ...
protected:
    CStatusBar m_wndStatusBar;
    CToolBar m_wndToolBar;
    CChildView m_wndView;
// 생략 ...
};
```

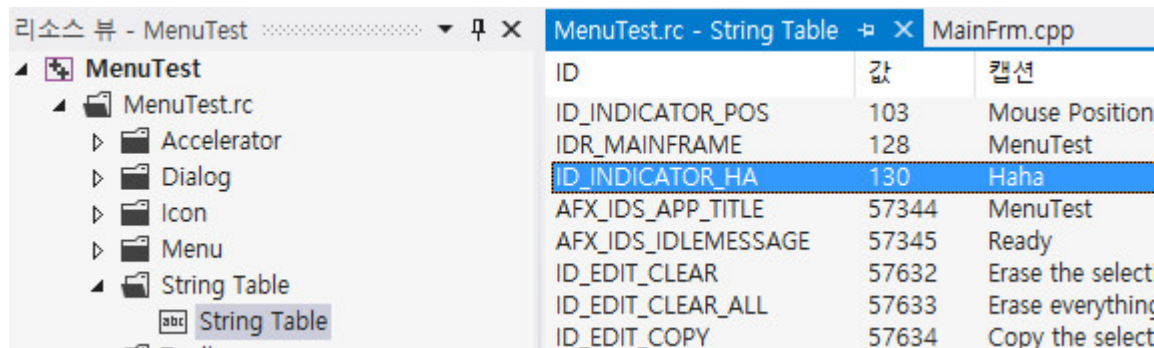
# 상태바 (4/4)

- 상태바 코드 (cont'd)

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    // 생략 ...
    if (!m_wndStatusBar.Create(this) ||
        !m_wndStatusBar.SetIndicators(indicators,
            sizeof(indicators)/sizeof(UINT)))
    {
        TRACE0("Failed to create status bar\n");
        return -1;
    }
    // 생략 ...
}
```

# 상태바에 정보 표시하기

## 1. 상태바에 정보를 표시할 공간 마련 스트링 테이블 추가



```
static UINT indicators[] =  
{  
    ID_SEPARATOR,  
    ID_INDICATOR_HA,  
    ID_INDICATOR_CAPS,  
    ID_INDICATOR_NUM,  
    ID_INDICATOR_SCRL,  
};
```

## 2. 상태바에 글쓰기

```
m_wndStatusBar.SetPaneText( 1 , _T("HaHa"));
```

구역 번호

상태바는 CMainFrame의 멤버변수로 존재  
어떻게 CChildView에서 CMainFrame으로 갈 수 있을까?



# 상태바에 정보 표시하기

---

1. 상태바에 정보를 표시할 공간 마련  
    **스tring 테이블 추가**
2. 코드 내에서 메인프레임 포인터를 얻어오기  
    **(CMainFrame \*) AfxGetMainWnd()**
3. 얻어온 메인프레임을 통해 상태바에 접근  
    **m\_wndStatusBar** (사전에 public으로 변경)
4. 상태바에 글쓰기  
    **m\_wndStatusBar.SetPaneText(1, \_T("HaHa"));**

# 연습: 상태바에 마우스 위치 표시

- 마우스의 위치를 표시



1. status bar에 indicator 추가
2. WM\_MOUSEMOVE 핸들러 추가 (CChildView)
3. SetPaneText 멤버함수 사용 위치 표시

# 연습: 상태바에 마우스 위치 표시

- 마우스의 위치를 표시

리소스 뷰 - MenuTest

MenuTest.rc - String Table\*    MainFrm.cpp    ChildView.h

ID	값	캡션
IDR_OLE_INIT_FAILED	100	OLE initialization failed. Make sur
ID_INDICATOR_POS	103	Mouse Position(9999,9999)
IDR_MAINFRAME	128	MenuTest
AFX_IDS_APP_TITLE	57344	MenuTest
AFX_IDS_IDLEMESSAGE	57345	Ready
ID_EDIT_CLEAR	57632	Erase the selectionWnErase
ID_EDIT_CLEAR_ALL	57633	Erase everythingWnErase All
ID_EDIT_COPY	57634	Copy the selection and put it on t

```
static UINT indicators[] =  
{  
    ID_SEPARATOR,  
    ID_INDICATOR_HA,  
    ID_INDICATOR_POS,  
    ID_INDICATOR_CAPS,  
    ID_INDICATOR_NUM,  
    ID_INDICATOR_SCRL,  
};
```

```
void CChildView::OnMouseMove(UINT nFlags, CPoint point)  
{  
    CString str;  
    str.Format(_T("Mouse Position(%d, %d)"), point.x, point.y);  
  
    CMainFrame * pMain = (CMainFrame *) AfxGetMainWnd();  
    pMain->m_wndStatusBar.SetPaneText(2, str);  
}
```

---

Q & A

# Homework #3 – Due: 05/15

- 간단한 그림판 만들기
  - CList, Timer, Menu의 적극 활용을 통한 프로그램 제작

