

Window Programming

Visual C++ MFC Programming

Lecture 07

김예진

Dept. of Game Software

Notices

- 03/20: HW 1 (Due: 03/26) → Avg: 8.85
- 04/05: HW 2 (Due: 04/13)

Plan

- MFC와 마우스
 - 연습 1: Line 그리는 프로그램
- MFC와 키보드
 - 연습 2: 사각형 움직이기

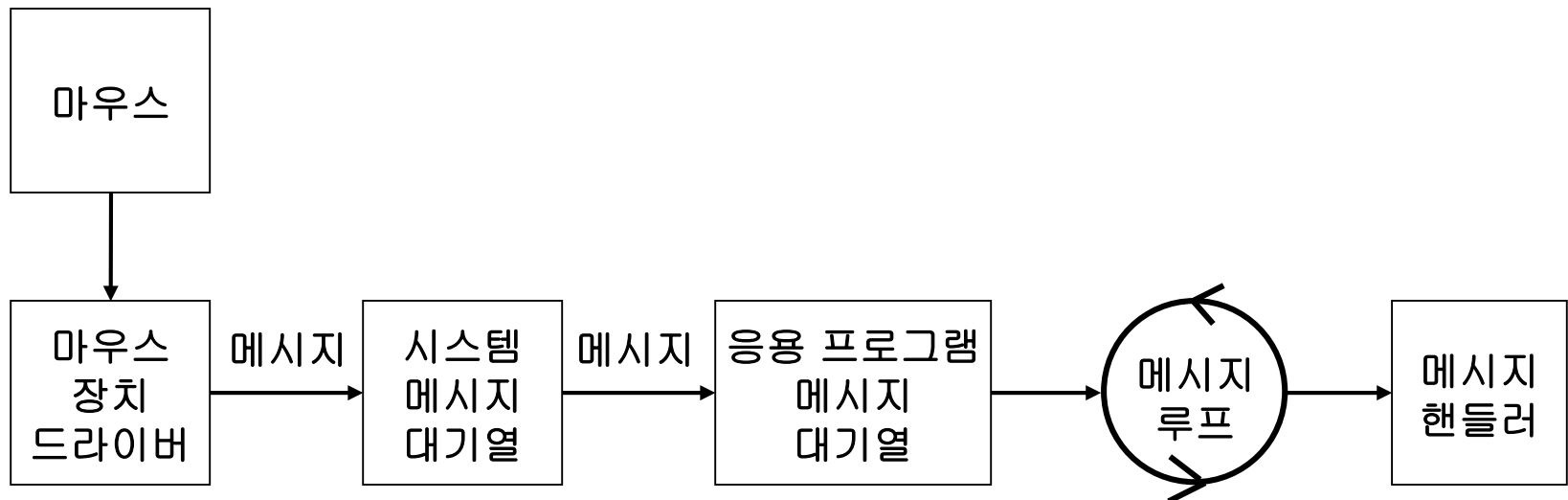
MFC와 Mouse



마우스 기초 (1/2)

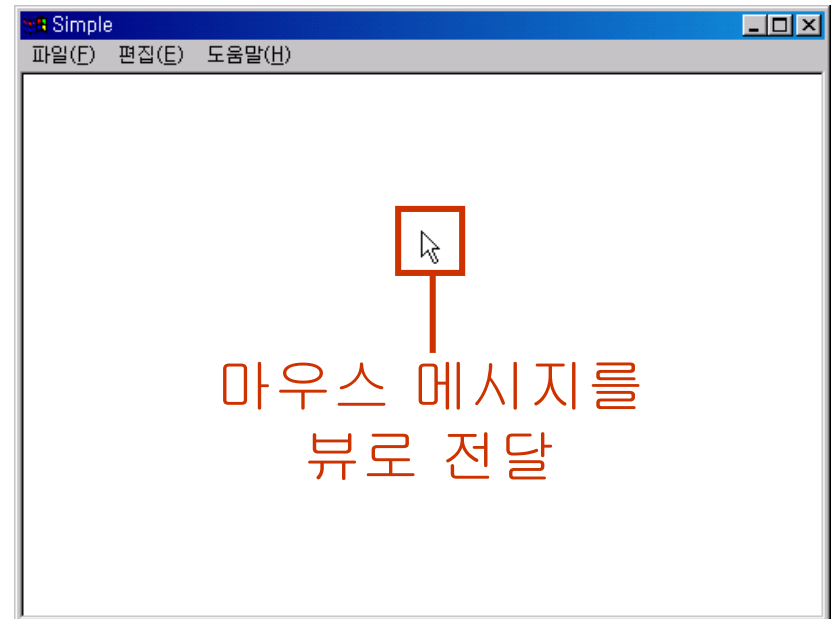
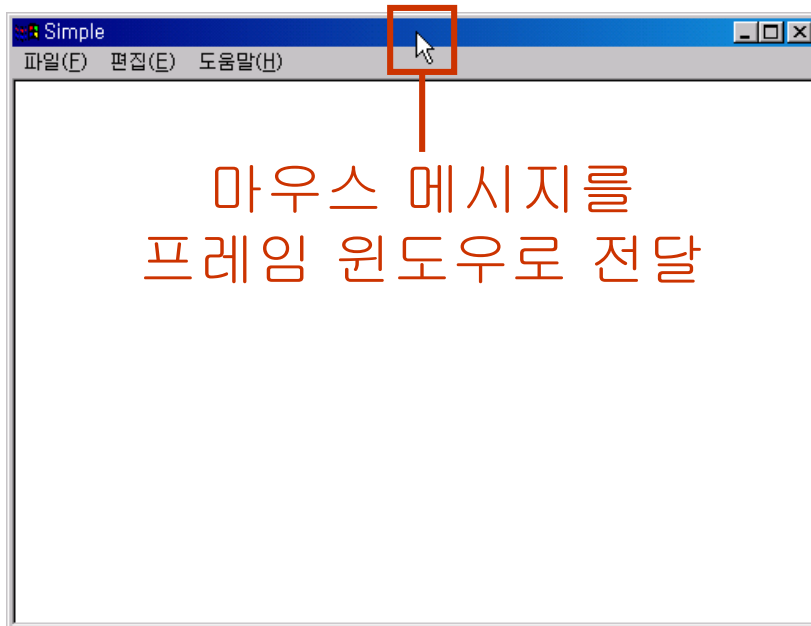
- 마우스 처리

- 윈도우 운영체제는 마우스와 관련된 모든 변화를 메시지 형태로 프로그램에게 전달



마우스 기초 (2/2)

- 마우스 메시지 전달
 - 마우스 메시지는 마우스 커서 밑에 있는 윈도우가 받음



클라이언트 영역 마우스 메시지 (1/6)

- 클라이언트 영역 마우스 메시지

메시지	발생 시점
WM_LBUTTONDOWN	왼쪽 버튼을 누를 때
WM_LBUTTONUP	왼쪽 버튼을 떼를 때
WM_LBUTTONDOWNBLCLK	왼쪽 버튼을 더블 클릭할 때
WM_MBUTTONDOWN	가운데 버튼을 누를 때
WM_MBUTTONUP	가운데 버튼을 떼를 때
WM_MBUTTONDOWNBLCLK	가운데 버튼을 더블 클릭할 때
WM_RBUTTONDOWN	오른쪽 버튼을 누를 때
WM_RBUTTONUP	오른쪽 버튼을 떼를 때
WM_RBUTTONDOWNBLCLK	오른쪽 버튼을 더블 클릭할 때
WM_MOUSEMOVE	마우스를 움직일 때

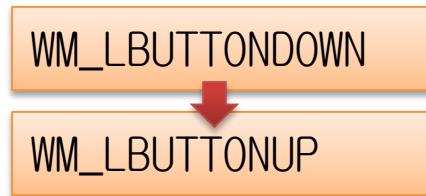
클라이언트 영역 마우스 메시지 (2/6)

- 메시지 발생 과정 (왼쪽마우스의 경우)

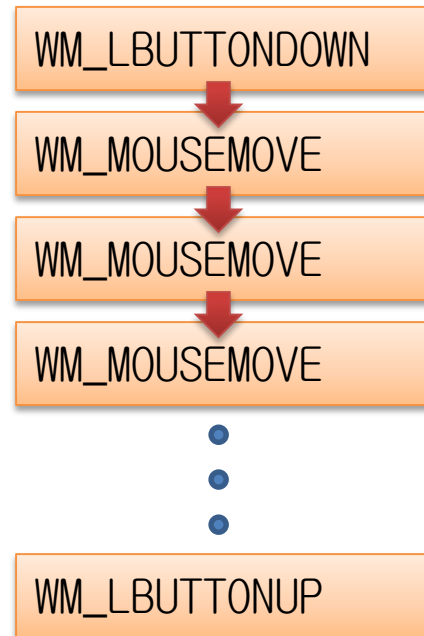
마우스를 움직일 때



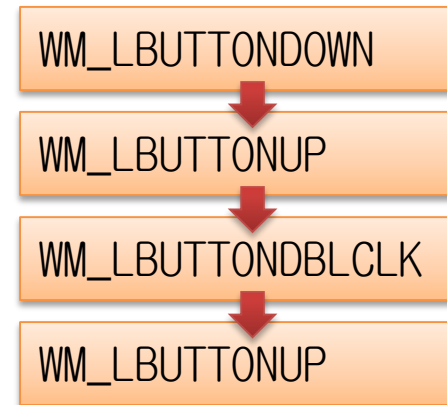
왼쪽 버튼 클릭



왼쪽 버튼 클릭
+ 드래그



왼쪽 버튼
더블 클릭



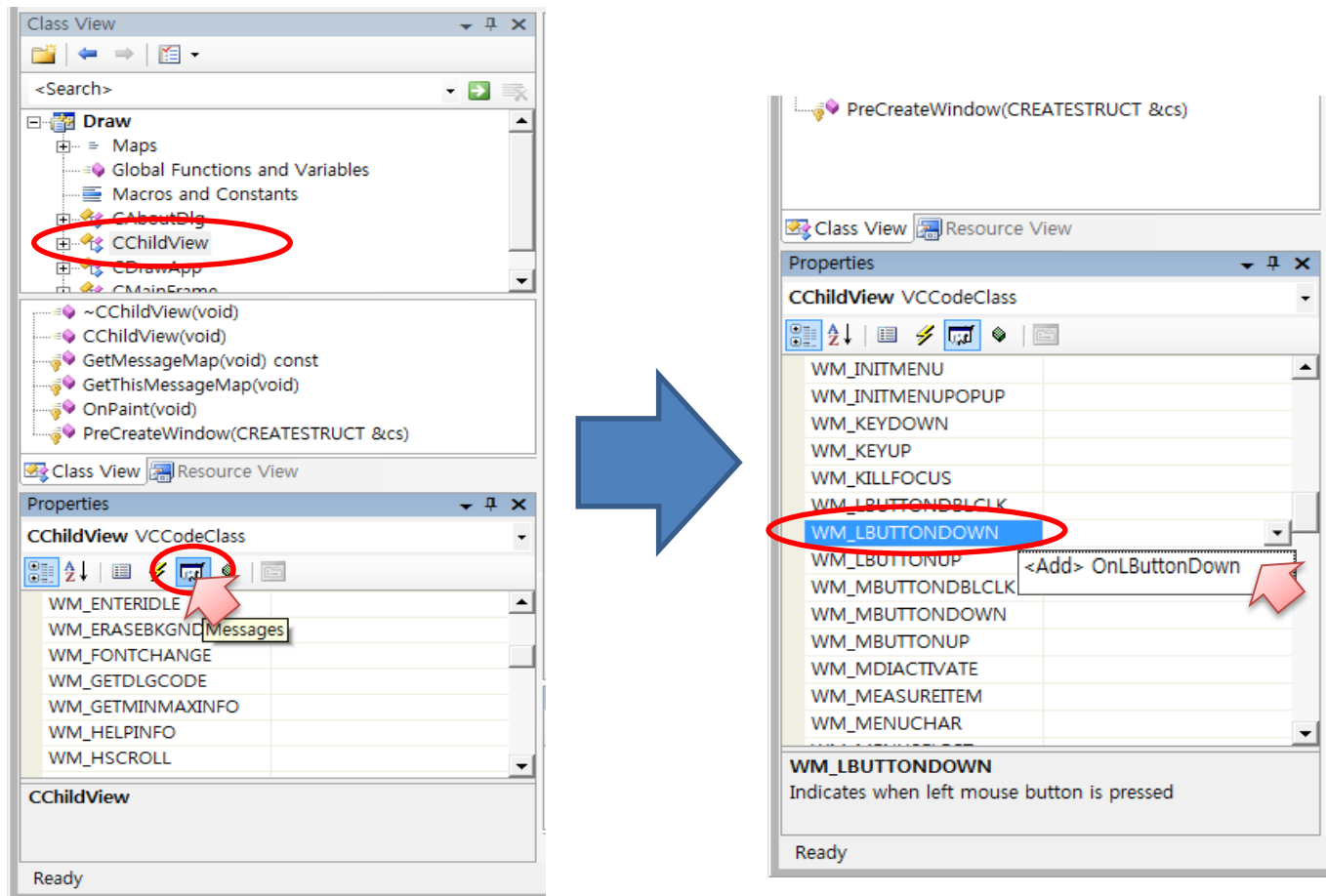
클라이언트 영역 마우스 메시지 (3/6)

- 클라이언트 영역 마우스 메시지 핸들러

메시지	메시지 맵 매크로	메시지 핸들러
WM_LBUTTONDOWN	ON_WM_LBUTTONDOWN()	OnLButtonDown
WM_LBUTTONUP	ON_WM_LBUTTONUP()	OnLButtonUp
WM_LBUTTONDOWNBLCLK	ON_WM_LBUTTONDOWNBLCLK()	OnLButtonDownBlCk
WM_MBUTTONDOWN	ON_WM_MBUTTONDOWN()	OnMButtonDown
WM_MBUTTONUP	ON_WM_MBUTTONUP()	OnMButtonUp
WM_MBUTTONDOWNBLCLK	ON_WM_MBUTTONDOWNBLCLK()	OnMButtonDownBlCk
WM_RBUTTONDOWN	ON_WM_RBUTTONDOWN()	OnRButtonDown
WM_RBUTTONUP	ON_WM_RBUTTONUP()	OnRButtonUp
WM_RBUTTONDOWNBLCLK	ON_WM_RBUTTONDOWNBLCLK()	OnRButtonDownBlCk
WM_MOUSEMOVE	ON_WM_MOUSEMOVE()	OnMouseMove

윈도우 메시지 핸들러 추가법

- CChildView의 Properties 창에서 추가할 수 있음
- 또는, 클래스 마법사(Class Wizard) 사용



클라이언트 영역 마우스 메시지 (4/6)

- 메시지 핸들러 형태

```
afx_msg void On##### (UINT nFlags, CPoint point) ;
```

- nFlags

- 메시지가 생성될 때의 키보드나 마우스 버튼의 상태를 나타내는 비트 마스크

비트 마스크	의미
MK_CONTROL	Ctrl 키가 눌렸을 때
MK_SHIFT	Shift 키가 눌렸을 때
MK_LBUTTON	마우스 왼쪽 버튼이 눌렸을 때
MK_MBUTTON	마우스 가운데 버튼이 눌렸을 때
MK_RBUTTON	마우스 오른쪽 버튼이 눌렸을 때

클라이언트 영역 마우스 메시지 (5/6)

- 메시지 핸들러 형태

```
afx_msg void On##### (UINT nFlags, CPoint point) ;
```

– nFlags

```
void CChildView::OnLButtonDown(UINT nFlags, CPoint point)
{
    if (nFlags & MK_SHIFT) {
        // 만약 Shift 키가 눌렀다면 ...
    }

    CWnd::OnLButtonDown(nFlags, point);
}
```

클라이언트 영역 마우스 메시지 (6/6)

- 메시지 핸들러 형태

```
afx_msg void On##### (UINT nFlags, CPoint point) ;
```




- point

- 메시지가 생성될 때의 마우스 커서 위치(클라이언트 좌표)

```
void CChildView::OnLButtonDown(UINT nFlags, CPoint point)
{
    CClientDC dc(this);
    CPoint pt = point;
    dc.Rectangle(pt.x - 100, pt.y + 100, pt.x + 100, pt.y - 100);

    CWnd::OnLButtonDown(nFlags, point);
}
```

마우스 캡처

- 문제점: 마우스가 밖으로 나가 버리면? 
- 용도
 - 마우스 캡처를 하면 마우스 커서의 위치에 관계없이 마우스 메시지를 받을 수 있음
- 관련 함수

API 함수	MFC 함수	의미
SetCapture()	CWnd::SetCapture()	마우스 캡처를 시작한다.
ReleaseCapture()	없음	마우스 캡처를 해제한다.
GetCapture()	CWnd::GetCapture()	어느 윈도우가 현재 마우스 캡처를 하고 있는지 알아낸다.

비 클라이언트 영역 마우스 메시지 (1/4)

- 비 클라이언트 영역 마우스 메시지

메시지	발생 시점
WM_NCLBUTTONDOWN	왼쪽 버튼을 누를 때
WM_NCLBUTTONUP	왼쪽 버튼을 떼를 때
WM_NCLBUTTONDBLCLK	왼쪽 버튼을 더블 클릭할 때
WM_NCMBBUTTONDOWN	가운데 버튼을 누를 때
WM_NCMBUTTONUP	가운데 버튼을 떼를 때
WM_NCMBUTTONDBLCLK	가운데 버튼을 더블 클릭할 때
WM_NCRBUTTONDOWN	오른쪽 버튼을 누를 때
WM_NCRBUTTONUP	오른쪽 버튼을 떼를 때
WM_NCRBUTTONDBLCLK	오른쪽 버튼을 더블 클릭할 때
WM_NCMOUSEMOVE	마우스를 움직일 때

비 클라이언트 영역 마우스 메시지 (2/4)

- 비 클라이언트 영역 마우스 메시지 핸들러

메시지	메시지맵 매크로	메시지 핸들러
WM_NCLBUTTONDOWN	ON_WM_NCLBUTTONDOWN()	OnNcLButtonDown
WM_NCLBUTTONUP	ON_WM_NCLBUTTONUP()	OnNcLButtonUp
WM_NCLBUTTONDBLCLK	ON_WM_NCLBUTTONDBLCLK()	OnNcLButtonDbLClk
WM_NCMBUTTONDOWN	ON_WM_NCMBUTTONDOWN()	OnNcMButtonDown
WM_NCMBUTTONUP	ON_WM_NCMBUTTONUP()	OnNcMButtonUp
WM_NCMBUTTONDBLCLK	ON_WM_NCMBUTTONDBLCLK()	OnNcMButtonDbLClk
WM_NCRBUTTONDOWN	ON_WM_NCRBUTTONDOWN()	OnNcRButtonDown
WM_NCRBUTTONUP	ON_WM_NCRBUTTONUP()	OnNcRButtonUp
WM_NCRBUTTONDBLCLK	ON_WM_NCRBUTTONDBLCLK()	OnNcRButtonDbLClk
WM_NCMOUSEMOVE	ON_WM_NCMOUSEMOVE()	OnNcMouseMove

비 클라이언트 영역 마우스 메시지 (3/4)

- 메시지 핸들러 형태

```
afx_msg void OnNc##### (UINT nHitTest, CPoint point);
```

- nHitTest

- 메시지가 생성될 때의 마우스 커서 위치를 나타내는 상수값 ⇒ 다음 페이지 표 참조

- point

- 메시지가 생성될 때의 마우스 커서 위치(스크린 좌표)
 - 클라이언트 좌표로 변환하려면 CWnd::ScreenToClient() 함수를 사용

비 클라이언트 영역 마우스 메시지 (4/4)

- nHitTest

상수값	의미
HTCAPTION	타이틀바
HTCLIENT	클라이언트 영역
HTCLOSE	종료 버튼
HTHSCROLL	가로 스크롤 바
HTMENU	메뉴
HTMAXBUTTON 또는 HTZOOM	최대화 버튼
HTMINBUTTON 또는 HTREDUCE	최소화 버튼
HTSYSTEMMENU	시스템 메뉴
HTVSCROLL	세로 스크롤 바

이외 마우스와 관련된 것들

- 집에서 읽어 볼 것: 교재, pp 214~220

- 마우스 포인터의 모양 지정:

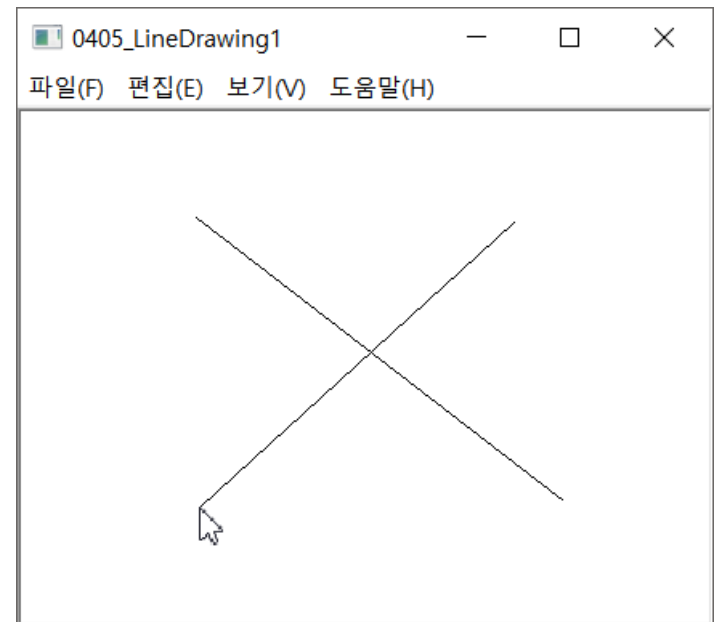
```
HCURSOR SetCursor (HCURSOR hCursor);
```

- 마우스 동작 범위 지정

```
::ClipCursor (lpRect pRect) ;
```

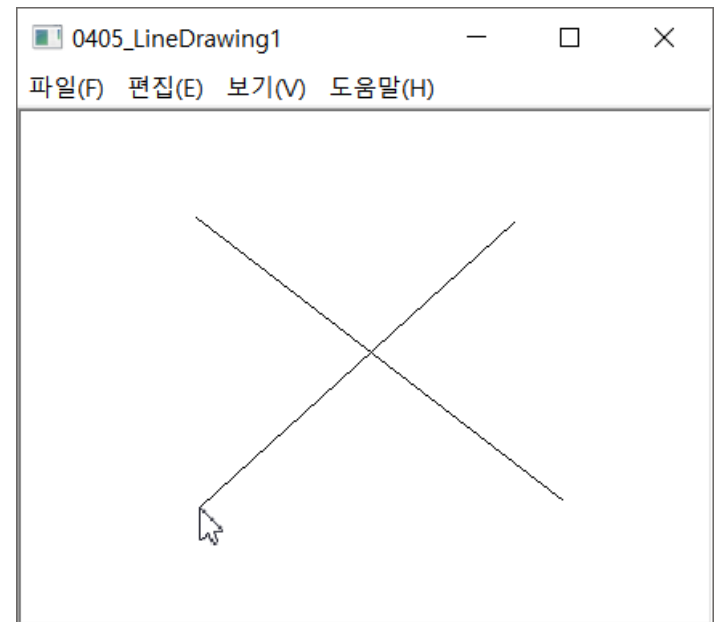
연습 1: Line 그리는 프로그램

- 화면에 마우스를 이용하여 line을 그리기
- Line 을 그리기 위해서는 시작점, 끝점을 기억
 1. 마우스 버튼 Down 시
 - 시작점을 입력
 2. 마우스 버튼 Up 시
 - 끝점을 입력
 - Line 그리기



연습 1: Line 그리는 프로그램

- 화면에 마우스를 이용하여 line을 그리기
- **마우스가 움직일 때 중간 과정을 보여줌**
- Line 을 그리기 위해서는 시작점, 끝점을 기억
 1. 마우스 버튼 Down 시
 - 시작점을 입력
 2. **마우스 Move 시**
 - **끝점을 입력**
 - **Line 그리기**
 3. 마우스 버튼 Up 시
 - 끝점을 입력
 - Line 그리기



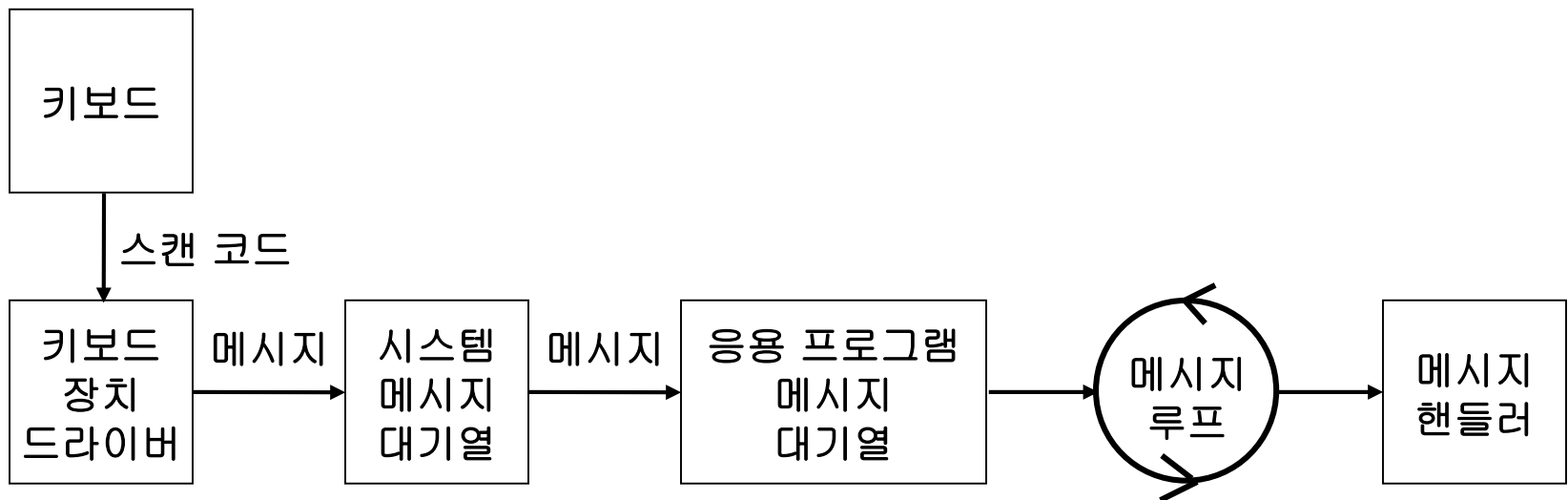
MFC와 Keyboard



키보드 기초 (1/2)

- 키보드 처리

- 윈도우 운영체제는 키보드와 관련된 모든 변화를 메시지 형태로 프로그램에게 전달

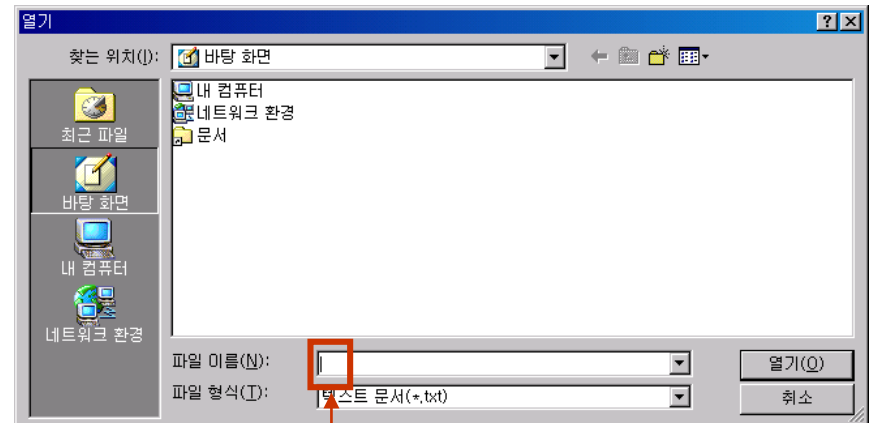
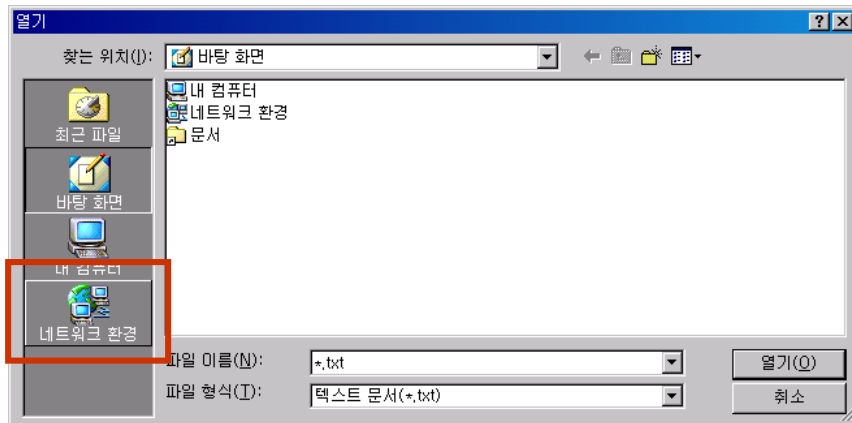


키보드 기초 (2/2)

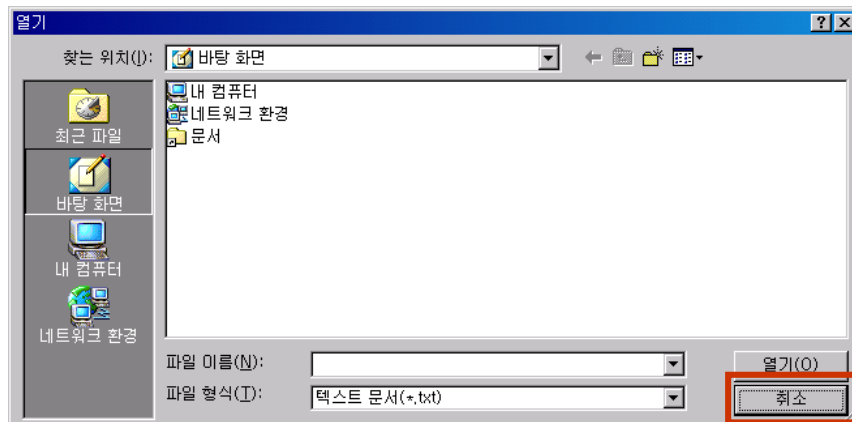
- 키보드 메시지 전달
 - 키보드 메시지는 키보드 포커스를 가진 윈도우가 받음
- 키보드 포커스
 - 활성 윈도우 또는 활성 윈도우의 자식 윈도우가 가지는 일종의 속성

키보드 포커스 (1/2)

- 키보드 포커스 유형

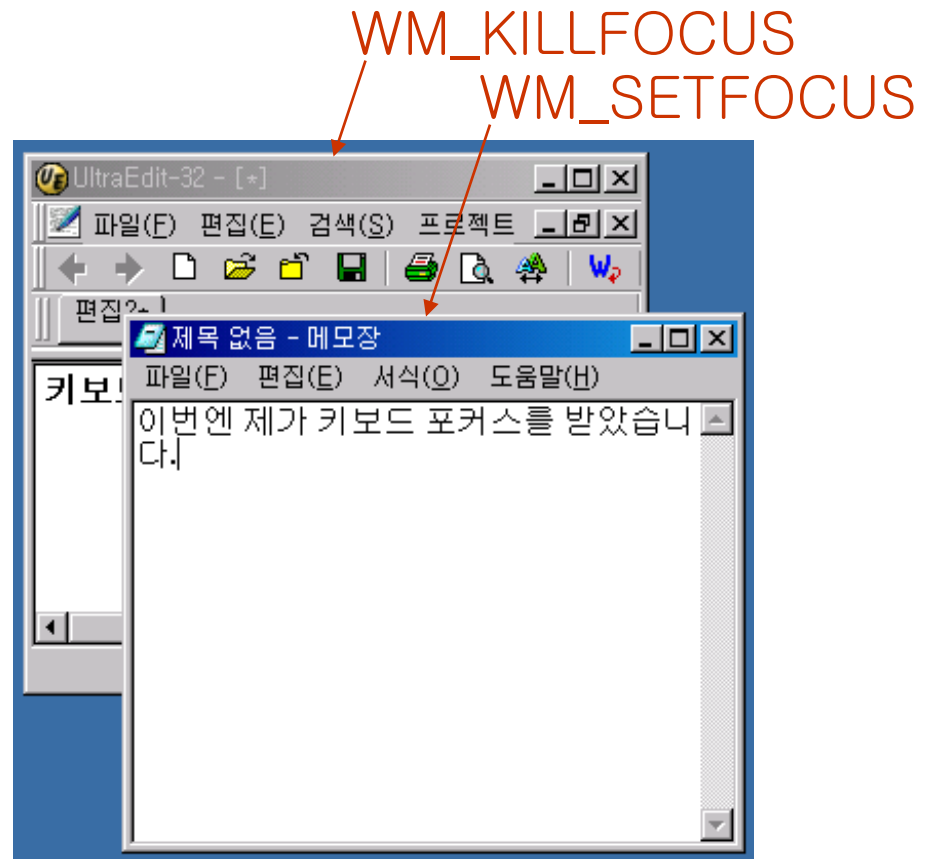
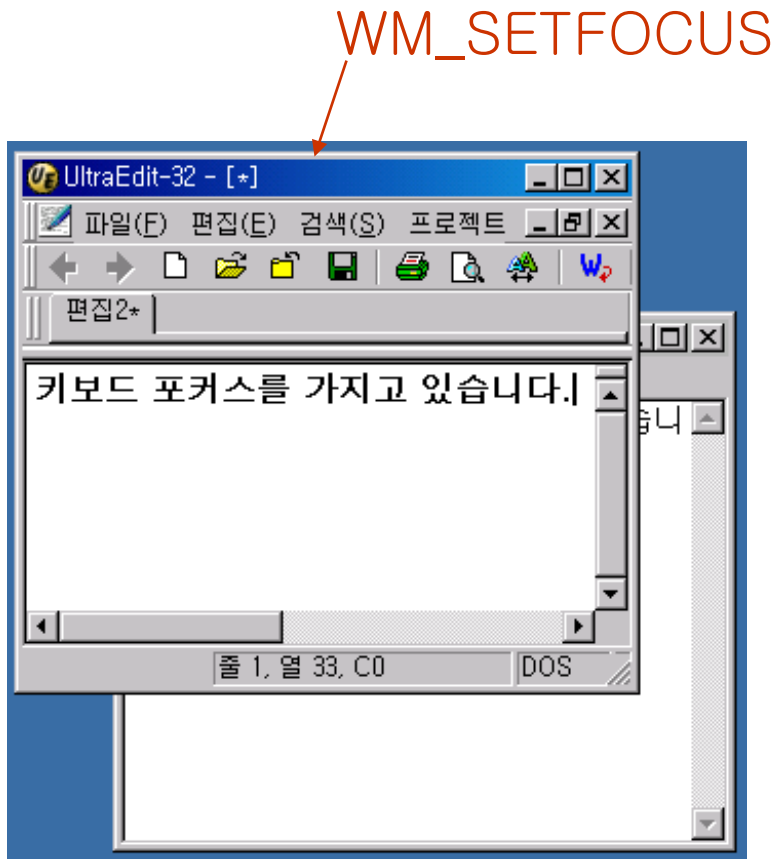


캐럿 (Caret)



키보드 포커스 (2/2)

- 키보드 포커스 변화



캐럿 (1/2)

- 관련 함수

함수 이름	역할
CreateCaret()	비트맵을 이용하여 캐럿을 생성한다.
CreateGrayCaret()	회색 사각형 모양의 캐럿을 생성한다.
CreateSolidCaret()	검정색 사각형 모양의 캐럿을 생성한다.
ShowCaret()	캐럿이 보이도록 한다.
HideCaret()	캐럿을 숨긴다.
GetCaretPos()	캐럿의 위치(클라이언트 좌표)를 얻는다.
SetCaretPos()	캐럿의 위치(클라이언트 좌표)를 변경한다.
::DestroyCaret()	캐럿을 파괴한다.
::GetCaretBlinkTime()	캐럿이 깜박이는 간격을 얻는다.
::SetCaretBlinkTime()	캐럿이 깜박이는 간격을 설정한다.

MFC에서 제공하지 않는 기능은 API 함수를 직접 호출

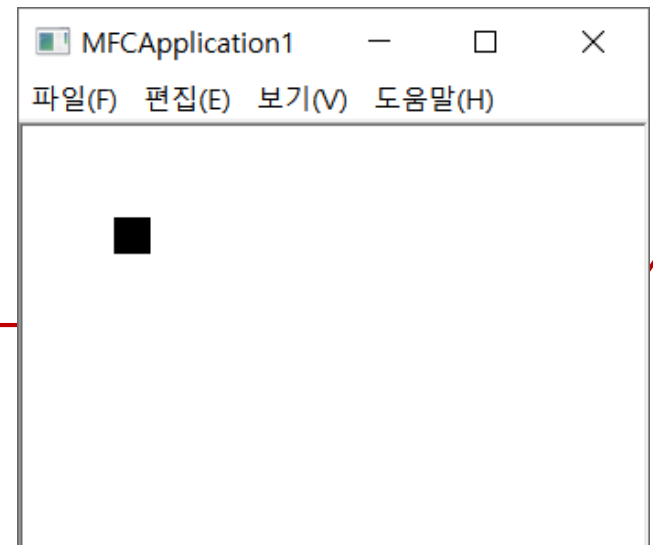
캐럿 (2/2)

```
void CChildView::OnSetFocus(CWnd* pOldWnd)
{
    CWnd ::OnSetFocus(pOldWnd);

    CreateSolidCaret(20, 20);
    SetCaretPos(CPoint(50, 50));
    ShowCaret();
}

void CChildView::OnKillFocus(CWnd* pNewWnd)
{
    CWnd ::OnKillFocus(pNewWnd);

    HideCaret();
    ::DestroyCaret();
}
```



키 누름 메시지 (1/3)

- 키 누름 메시지(Keystroke Message)
 - 키보드를 누르거나 떼는 동작에 의해 발생하는 메시지
- 키 누름 메시지 종류

메시지	의미
WM_KEYDOWN	F10, Alt 이외의 키를 누를 때
WM_KEYUP	F10, Alt 이외의 키를 떼를 때
WM_SYSKEYDOWN	F10, Alt, Alt + [키 조합]을 누를 때
WM_SYSKEYUP	F10, Alt, Alt + [키 조합]을 떼를 때

키 누름 메시지 (2/3)

- 메시지 핸들러 형태

```
afx_msg void On####(UINT nChar, UINT nRepCnt, UINT nFlags);
```

- nChar

- 키에 할당된 가상 키 코드값 ⇨ 다음 페이지 표 참조

- nRepCnt

- 키를 계속 누르고 있을 경우 1보다 큰 값을 가짐

- nFlags

- 키와 관련된 다양한 정보를 담고 있음 (MSDN 참조)

키 누름 메시지 (3/3)

- 가상 키 코드 (Virtual Key Code)
 - 운영체제가 각각의 키에 할당한 고유한 값 (장치 독립적)

가상 키 코드	해당 키	가상 키 코드	해당 키
VK_CANCEL	Ctrl-Break	VK_HOME	Home
VK_BACK	Backspace	VK_LEFT	←
VK_TAB	Tab	VK_UP	↑
VK_RETURN	Enter	VK_RIGHT	→
VK_SHIFT	Shift	VK_DOWN	↓
VK_CONTROL	Ctrl	VK_SNAPSHOT	Print Screen
VK_MENU	Alt	VK_INSERT	Insert
VK_PAUSE	Pause	VK_DELETE	Delete
VK_CAPITAL	Caps Lock	'0' - '9'	0 ~ 9
VK_ESCAPE	Esc	'A' - 'Z'	A ~ Z
VK_SPACE	Spacebar	VK_F1 ~ VK_F12	F1 ~ F12
VK_PRIOR	PgUp	VK_NUMLOCK	Num Lock
VK_NEXT	PgDn	VK_SCROLL	Scroll Lock
VK_END	End	VK_SCROLL	Scroll Lock

문자 메시지 (1/3)

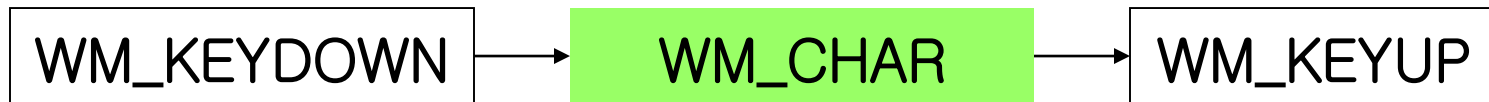
- 문자 메시지 필요성 - VK_R 키를 누른 경우?

문자	가상 키 코드 조합
r	영문 입력 모드에서 VK_R 또는 Caps Lock + Shift + VK_R 키를 누른 경우
R	영문 입력 모드에서 Caps Lock + VK_R 또는 Shift + VK_R 키를 누른 경우
ㄱ	한글 입력 모드에서 VK_R 키를 누른 경우
ㄲ	한글 입력 모드에서 VK_R + Shift 키를 누른 경우

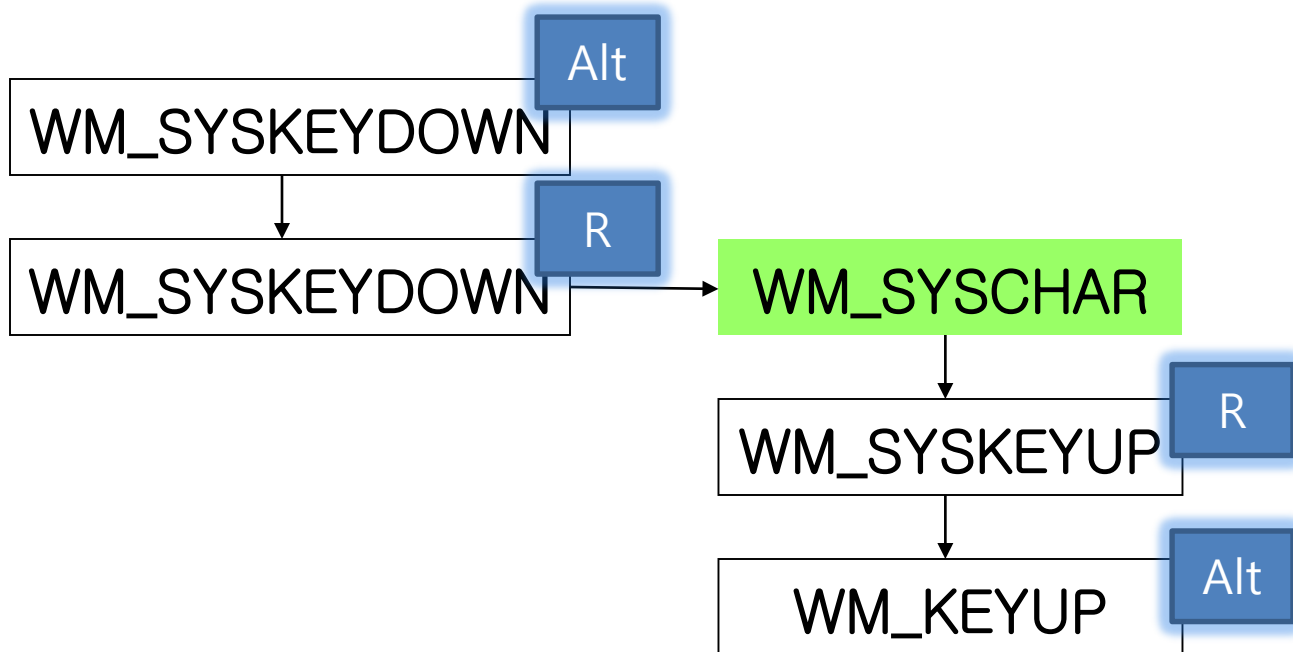
→ 키 누름 메시지 (WM_KEYDOWN) 외에
문자 메시지 (WM_CHAR)를 발생 시킴

문자 메시지 (2/3)

- 문자 메시지 - VK_R 키를 누른 경우



- 문자 메시지 - Alt + VK_R 키를 누른 경우



문자 메시지 (3/3)

- 메시지 핸들러 형태

```
afx_msg void OnChar(UINT nChar, UINT nRepCnt, UINT nFlags);  
afx_msg void OnSysChar(UINT nChar, UINT nRepCnt, UINT nFlags);
```

- nChar

- 키에 해당하는 문자 코드(Character Code) 값을 가진다.

- nRepCnt

- 키를 계속 누르고 있을 경우 1보다 큰 값을 가진다.

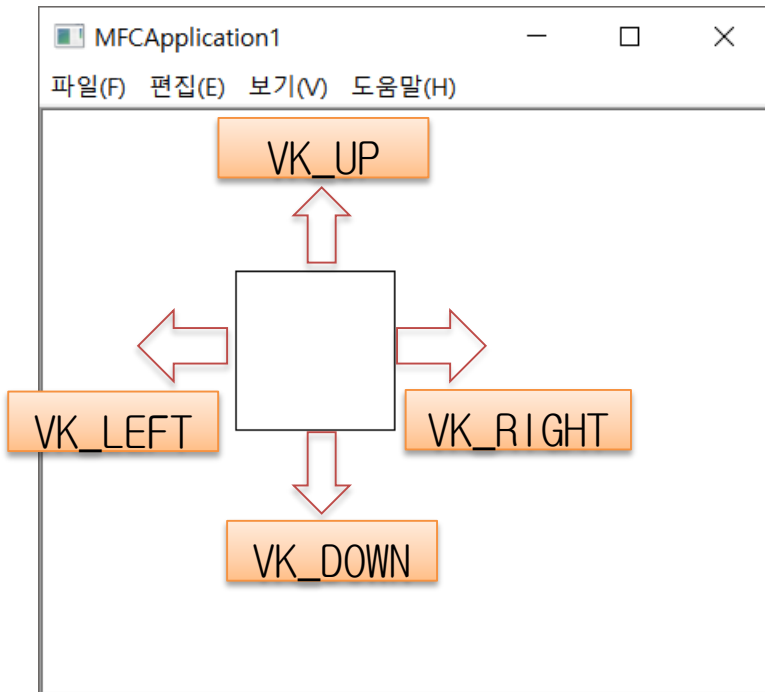
- nFlags

- 키와 관련된 다양한 정보를 담고 있다(MSDN 참조).

연습 2: 사각형 움직이기

- 사각형의 위치를 방향키로 조작

```
void OnKeyDown (UINT nChar, UINT nRepCnt, UINT nFlags) ;
```



1. 위치 저장 변수 선언
`CPoint m_pt;`
2. 위치에 사각형그리기
(OnPaint)
`dc.Rectangle(m_pt.x, ...);`
3. OnKeyDown 핸들러 추가
4. nChar 값 검사하여 위치변경
`if (nChar == VK_RIGHT)
 m_pt.x += 10;`

Q & A