

9장

키보드와 마우스

김성영교수
금오공과대학교
컴퓨터공학부

키보드 메시지 기본

키보드에서 어떤 키를 누르면 활성화된 윈도우의 어플리케이션 큐에 **WM_KEYDOWN**이나 **WM_SYSKEYDOWN** 메시지를 넣는다.

실습 9.1

키보드에서 입력한 문자열을 그대로 화면에 출력하자.

- My 프로젝트 기반 코드 작성
- 코드를 간단히 하기 위해 뷰에서 모두 처리
- **WM_KEYDOWN** 메시지 핸들러 사용

MyView.cpp

```
void CMyView::OnKeyDown( UINT nChar, UINT nRepCnt, UINT nFlags )
{
    CString strChar;
    strChar.Format( "%c", nChar );

    CClientDC dc( this );
    dc.TextOut( 0, 0, strChar, strChar.GetLength() );

    CView::OnKeyDown(nChar, nRepCnt, nFlags);
}
```

가상 키

- 입력한 키의 코드값 - OnKeyDown() 핸들러의 첫 번째 인자
- VK로 시작하는 상수로 정의

키 값	식별자	키
1	VK_LBUTTON	마우스 왼쪽 버튼
2	VK_RBUTTON	마우스 오른쪽 버튼
8	VK_BACK	역 공백 키
13	VK_RETURN	엔터 키
16	VK_SHIFT	쉬프트 키
17	VK_CONTROL	컨트롤 키
32	VK_SPACE	공백 키
35	VK_END	엔드 키
36	VK_HOME	홈 키
37	VK_LEFT	왼쪽 화살표 키

키 값	식별자	키
38	VK_UP	위쪽 화살표 키
39	VK_RIGHT	오른쪽 화살표 키
40	VK_DOWN	아래쪽 화살표 키
45	VK_INSERT	삽입 키
46	VK_DELETE	삭제 키
48 ~ 57	'0' ~ '9'	숫자 키 0 ~ 9
65 ~ 90	'A' ~ 'Z'	문자 키 A ~ Z, 소문자는 없음
112 ~ 127	VK_F1 ~ VK_F16	기능 키 F1 ~ F16

- 숫자와 알파벳 이외의 다른 키에 대한 입력을 막기 위한 조건

```
if( (nChar >= '0' && nChar <= '9') || (nChar >= 'A' && nChar <= 'Z') )
```

가상 키 상태 검사

SHIFT와 A키를 동시에 누른 경우 nChar에는 무슨값이 전달 되겠는가?

→ 'A'가 전달

쉬프트 키의 상태는 GetKeyState() API함수를 통해 알 수 있다.

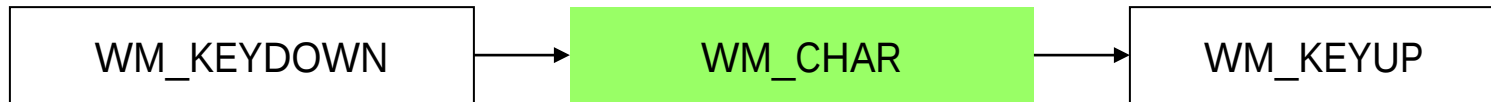
```
SHORT GetKeyState( int nVirtKey );
```

시프트 키 (Shift, Ctrl, Alt) 및 토글 키 (Caps Lock, Num Lock, Scroll Lock)
의 누름 여부 파악

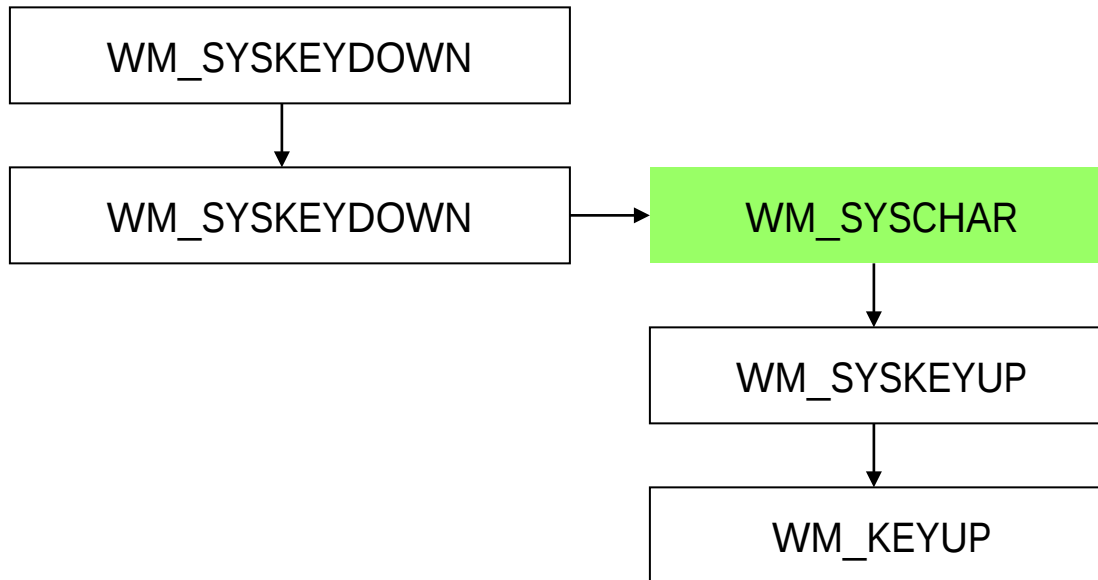
- `iState = GetKeyState( VK_SHIFT );` // 음수(눌러짐) 혹은 양수
- VK_SHIFT, VK_CONTROL, VK_MENU(Alt 키)
- VK_LSHIFT, VK_RSHIFT, VK_LCONTROL, VK_RCONTROL, VK_LMENU, VK_RMENU
- VK_LBUTTON, VK_RBUTTON, VK_MBUTTON

문자 메시지

문자 메시지 - VK_R 키를 누른 경우



문자 메시지 - Alt + VK_R 키를 누른 경우



문자 메시지

소문자를 출력하기 위해서는 소문자에 대한 **ASCII 코드값** 필요

키가 눌렸을 때 ASCII 코드 값을 제공하는 메시지

→ **WM_CHAR**

소문자 출력을 위해서는 **WM_KEYDOWN** 메시지 대신

WM_CHAR 메시지를 처리

OnKeyDown() 핸들러의 내용을 **OnChar()** 핸들러로 옮기고 입력 가능 문자를 숫자와 알파벳(대문자 및 소문자)로 제한

MyView.cpp

```
void CMyView::OnChar( UINT nChar, UINT nRepCnt, UINT nFlags )
{
    CString strChar;
    strChar.Format( "%c", nChar );

    CClientDC dc(this);
    if ( (nChar >= '0' && nChar <= '9') || (nChar >= 'A' && nChar <= 'z') )
        dc.TextOut(0, 0, strChar, strChar.GetLength() );

    CView::OnChar( nChar, nRepCnt, nFlags );
}
```

출력된 문자열이 지워지지 않도록 OnDraw() 핸들러에서 출력

MyView.h

```
class CMyView : public CView
{
    CString m_strNote;
}
```

MyView.cpp

```
CMyView::CMyView( )
{
    m_strNote = _T("");
}

void CMyView::OnDraw( CDC* pDC )
{
    pDC->TextOut( 0, 0, m_strNote );
}

void CMyView::OnChar(UINT nChar, UINT nRepCnt, UINT nFlags)
{
    CString strChar;
    strChar.Format( "%c", nChar );

    if ( (nChar >= '0' && nChar <= '9') || (nChar >= 'A' && nChar <= 'z') )
        m_strNote += strChar; //문자열 뒤에 연결하여 붙임

    Invalidate(); //WM_PAINT 발생
    CView::OnChar( nChar, nRepCnt, nFlags );
}
```