

02. 도형 그리기 및 마우스

```
struct _tagArea
{
    bool bStart;
    POINT ptStart;
    POINT ptEnd;
};

_tagArea g_tArea;
```

마우스 커서 위치와 클릭했는지의 여부를 저장할 전역 구조체 변수.

<WM_PAINT, 화면에 그리는 메시지>

```
case WM_PAINT:
{
    PAINTSTRUCT ps;
    HDC hdc = BeginPaint(hWnd, &ps);
    // TODO: Add any drawing code that uses hdc here...

    TextOut( hdc, 50, 50, TEXT( "한" ), 1 );
    TextOut( hdc, 65, 65, TEXT( "글" ), 1 );
    TextOut( hdc, 80, 80, TEXT( "Hangeul" ), 7 );

    TCHAR strMouse[64] = {};
    wsprintf( strMouse, TEXT( "Start = x : %d y : %d" ), g_tArea.ptStart.x, g_tArea.ptStart.y );
    TextOut( hdc, 600, 30, strMouse, lstrlen( strMouse ) );

    Rectangle( hdc, 100, 100, 200, 200 );
    Ellipse( hdc, 100, 100, 200, 200 );

    // 라인의 시작점을 정의.
    MoveToEx( hdc, 300, 100, NULL );
    // 끝 점을 정의
    LineTo( hdc, 600, 600 );
    LineTo( hdc, 800, 100 );
    MoveToEx( hdc, 100, 400, NULL );
    LineTo( hdc, 150, 800 );

    if( g_tArea.bStart )
    {
        Rectangle( hdc, g_tArea.ptStart.x, g_tArea.ptStart.y, g_tArea.ptEnd.x, g_tArea.ptEnd.y );
    }

    EndPaint(hWnd, &ps);
}
break;
```

HDC

) 화면에 문자열 출력하는 함수

사각형 그리기

원 그리기(사각형 크기 똑차게 그림)

) 이어서 LineTo 할 경우 이전 끝 지점에서 계속해서 그림.

) 새로운 지점에서 선을 그리기 위해 MoveToEx로 시작점 재지정.

<TEXT 매크로>

```
#define __TEXT(quote) L##quote // r_winnt
#else /* UNICODE */ // r_winnt

#ifdef _TCHAR_DEFINED
typedef char TCHAR, *PTCHAR;
typedef unsigned char TBYTE, *PTBYTE;
#define _TCHAR_DEFINED
#endif /* !_TCHAR_DEFINED */

typedef LPCH LPTCH, PTCH;
typedef LPCCH LPCTCH, PCTCH;
typedef LPSTR PTSTR, LPTSTR, PUTSTR, LPUTSTR;
typedef LPCSTR PCTSTR, LPCTSTR, PCUTSTR, LPCUTSTR;
typedef PZZSTR PZZTSTR, PUZZTSTR;
typedef PCZZSTR PCZZTSTR, PCUZZTSTR;
typedef PZPSTR PZPTSTR;
typedef PNZCH PNZTCH, PUNZTCH;
typedef PCNZCH PCNZTCH, PCUNZTCH;
#define __TEXT(quote) quote // r_winnt
#endif /* UNICODE */ // r_winnt
#define TEXT(quote) __TEXT(quote) // r_winnt
```

→ 유니코드

##은 앞과 뒤 붙여줌.

→ 멀티바이트

→ 최종 define

TEXT 매크로의 경우 유니코드/멀티바이트의 경우에 따라 각각 다르게 정의되기 때문에 두 경우 모두 사용 가능하다.

<HDC(Handle to a Device Context)>
Windows는 3개의 DLL(동적라이브러리)로 구성되어 있음.

- KERNEL: 메모리 관리 및 프로그램 실행
- USER: 유저 인터페이스와 윈도우 관리
- GDI: 화면 처리 담당

Windows의 API 함수 대부분은 이 세 DLL에 의해서 제공됨.

즉, 윈도우즈의 클라이언트 영역에 그리기를 하려면 GDI(Graphics Device Interface)를 이용해야 하며, HDC는 GDI의 중요 부분이다. 다시 말해 DC 객체 참조시 사용하는 번호.

*DC(Device Context)

출력에 필요한 모든 정보를 가진 구조체.

GDI 모듈에 의해 관리됨.

클라이언트 영역에 그리기 위한 도구. 도화지

<TCHAR>

char → CHAR

wchar_t → WCHAR

이를 유니코드/멀티 바이트일 때 동시에 상황에 맞게 사용할 수 있도록 다음과 같이 재정의.

```
#ifdef UNICODE
typedef WCHAR TCHAR;
#else
typedef CHAR TCHAR;
#endif
```

<wsprintf>

sprintf가 형식 문자열 사용해 버퍼로 넘겨준 문자에 문자열을 복사해줌. wsprintf는 유니코드 또한 복사해줌.

<lstrlen>

유니코드 문자열 길이 구해주는 함수.

<마우스 입력부분 메시지>

```
// 마우스 왼쪽이 눌렸을 때 들어오는 메시지.
case WM_LBUTTONDOWN:
    if( !g_tArea.bStart )
    {
        g_tArea.bStart = true;
        g_tArea.ptStart.x = lParam & 0x0000ffff;
        g_tArea.ptStart.y = lParam >> 16;
        g_tArea.ptEnd = g_tArea.ptStart;

        InvalidateRect( hWnd, NULL, TRUE );
    }
    break;
// 마우스가 움직일때 들어오는 메시지.
case WM_MOUSEMOVE:
    if( g_tArea.bStart )
    {
        g_tArea.ptEnd.x = lParam & 0x0000ffff;
        g_tArea.ptEnd.y = lParam >> 16;

        InvalidateRect( hWnd, NULL, TRUE );
    }
    break;
// 마우스 왼쪽 버튼을 누르다가 뗐을 때 발생하는 메시지.
case WM_LBUTTONUP:
    if( g_tArea.bStart )
    {
        g_tArea.bStart = false;
        g_tArea.ptEnd.x = lParam & 0x0000ffff;
        g_tArea.ptEnd.y = lParam >> 16;

        InvalidateRect( hWnd, NULL, TRUE );
    }
    break;
// 키가 눌렸을 때 들어오는 메시지.
case WM_KEYDOWN:
    // 이 메시지가 들어올 경우 wParam에 어떤 키가 눌렸는지가 들어옴.
    switch( wParam )
    {
        case VK_ESCAPE:
            DestroyWindow( hWnd );
            break;
    }
    break;
case WM_DESTROY:
    PostQuitMessage(0);
    break;
default:
    return DefWindowProc(hWnd, message, wParam, lParam);
}
return 0;
```

) 하위 16비트
얻어오기.

↳ 강제로 WM_PAINT
메시지를 호출해주는 함수

lParam >> 16 : 16비트 왼쪽으로 밀어버림.

16진수 값 1자리는 4비트이므로 4자리를 왼쪽으로 미는것.
이를 통해 상위 16비트 추출함.

< Invalidate Rect >

1번 인자) 윈도우 핸들

2번 인자) 갱신할 영역. NULL이면 전체화면을
대상으로 갱신함.

3번 인자) TRUE : 현재 화면을 지우고 갱신.

FALSE : 현재 화면을 안지우고 갱신.

< 출력 화면 >

