Window API Programming- 입문 2

afewhee@gmail.com

1. 도형 그리기 - 1

- 선그리기
  - BOOL MoveToEx(HDC hdc, int X, int Y, LPPOINT lpPoint);
    - 현재의 위치 이동
    - X, Y : 새로운 위치 좌표
    - lpPoint : 이전 위치. NULL 인 경우 화면의 절대 위치
  - BOOL LineTo(int x, int y), BOOL LineTo(POINT point); 현재의 위치에서 목적 위치까지 선을 그림
- 픽셀 색상 변경
  - SetPixel(int x, int y, COLORREF crColor), (POINT point, COLORREF crColor) : 특정위치의 픽셀 색상을 변경
  - 사용법: SetPixel(hdc, x, y, RGB(0,0,0));
- 매핑 모드: 논리적 좌표는 물리적 좌표로 Mapping
  - SetMapMode(HDC hdc, int iMode):
  - iMode:
    - MM_TEXT: 픽셀
    - MM_LOMETRIC: 논리적 단위 1을 물리적 단위 0.1mm로 매핑
    - MM_HIMETRIC: 논리적 단위 1을 물리적 단위 0.01mm로 매핑
- 원점 이동 함수
  - SetViewportOrgEx(hdc, 600, 300, NULL);

- 직사각형
  - Rectangle(HDC, left, top, right, bottom)
- 타원
  - Ellipse(HDC, left, top, right, bottom)
- 채우기
  - CreateSolidBrush(), CreateBrushIndirect()
- 선
  - CreatePen(), CreatePenIndirect()
- 다각형
  - 선 그리기: Polyline(HDC, POINTs, Count)
  - 채우기: Polygon(HDC, POINTs, Count)

- 도형 그리기 순서
  - Brush, Pen 객체를 생성한다.
    - CreateBrushIndirect()
    - CreatePen()
  - DC에 Brush, Pen 객체를 설정하고, 이전의 Pen, Brush객체를 저장한다.
    - OldBrush=(HBRUSH)SelectObject(hdc, NewBrush);
    - OldPen =(HPEN)SelectObject(hdc, NewPen);
  - 도형을 다 그렸으면 이전에 저장된 Brush, Pen 객체를 DC에 돌려 준다.
    - SelectObject(hdc,OldBrush);
    - SelectObject(hdc,OldPen);
  - 생성한 Brush, Pen 객체를 반환한다.
    - DeleteObject(NewBrush);
    - DeleteObject(NewPen);

- C함수:
  - clock(): 프로그램 시작 후부터 시간(millisecond)
  - _getsystime(&tmTime): 시스템의 현재 시간 알아오기

- WinAPI 함수:
  - GetTickCount(): 운영체제 시작 후부터 시간(millisecond)
  - timeGetTime() : 운영체제 시작 후부터 시간(millisecond) ➔ mmsystem.h, winmm.lib

  - ::GetLocalTime(&tmS) : 시스템의 현재 시간 알아오기

- 타임 이벤트 생성:
  - SetTimer(HWND, UINT_PTR, UINT, TIMERPROC pTimerFunc)
  - pTimerFunc: NULL이면 WM_TIMER 메시지 발생

- BOOL GetWindowRect(HWND, LPRECT):
  - 윈도우의 영역(현재위치, 크기) 반환
- GetClientRect(HWND, LPRECT):
  - 윈도우 작업 영역 크기 계산
- ScreenToClient(HWND, LPPOINT):
  - 바탕 화면에 의한 좌표 점을 해당 윈도우 작업 영역의 상대 좌표로 변환
- AdjustWindowRect()
- ➔ Photoshop으로 확인 필요!!!

4. Keyboard / Mouse

- WM_KEYDOWN:
  - 키보드를 누를 때 발생하는 메시지. Alt와 함께 키를 눌렀을 때는 WM_SYSKEYDOWN 메시지 발생
  - wParam : 가상 키 코드 값. 키의 종류.

```
Ex)
case WM_KEYDOWN:
{
    switch(wParam)
  {
      case VK_LEFT:
  ...
```

- WM_KEYUP:
  - 키보드를 뗄 때 발생하는 메시지

- WM_CHAR:
  - 문자에 해당하는 키에 대한 메시지. 문자열 출력 가능
  - wParam : 입력된 문자 코드 ➔ 아스키 코드 값.
  - 키보드의 문자 키를 눌렀다 떼면 WM_KEYDOWN ➔ WM_CHAR ➔ WM_KEYUP 순서로 메시지 발생

```
Ex)
case WM_CHAR:
{
    char c = (char)wParam;
    ...
```

- WM_{L|R|M}BUTTON{DOWN|UP}:
  - 마우스 버튼을 누를 때/ 뗄 때 메시지 발생
  - LOWORD(lParam), HIWORD(lParam) : 해당 윈도우의 작업 영역에 대한 상대적인 마우스의 X, Y 좌표.

  Ex)
  ```
  case WM_KEYDOWN:
  {
        x=LOWORD(lParam);
        y=HIWORD(lParam);
    ...
  ```

- WM_MOUSEMOVE:
  - 마우스가 움직일 때 발생

- WM_{L|R|M}BUTTONDBLCLK:
  - 더블 클릭에 대한 메시지
  - 윈도우를 생성할 때 WNDCLASS 구조체 변수 style의 값에 더블클릭에 대한 이벤트를 받을 수 있도록 다음과 같이 설정
  - WndClass.style |=CS_DBLCLKS;

- WM_MOUSEWHEEL:
  - 휠 메시지: _WIN32_WINNT가 최소한 0x0400 이상으로 선언되어 있어야 함
  - 휠의 상대적인 크기 계산: short v = short(HIWORD(wParam));

- 타이머를 이용해서 화면 보호 프로그램을 작성하시오.
- 타이머를 이용해서 시계를 만드시오.
- DC를 이용해서 태극기를 그리시오.
- 키보드 마우스에 대한 클래스 CInput 클래스를 만들고 구현하시오.