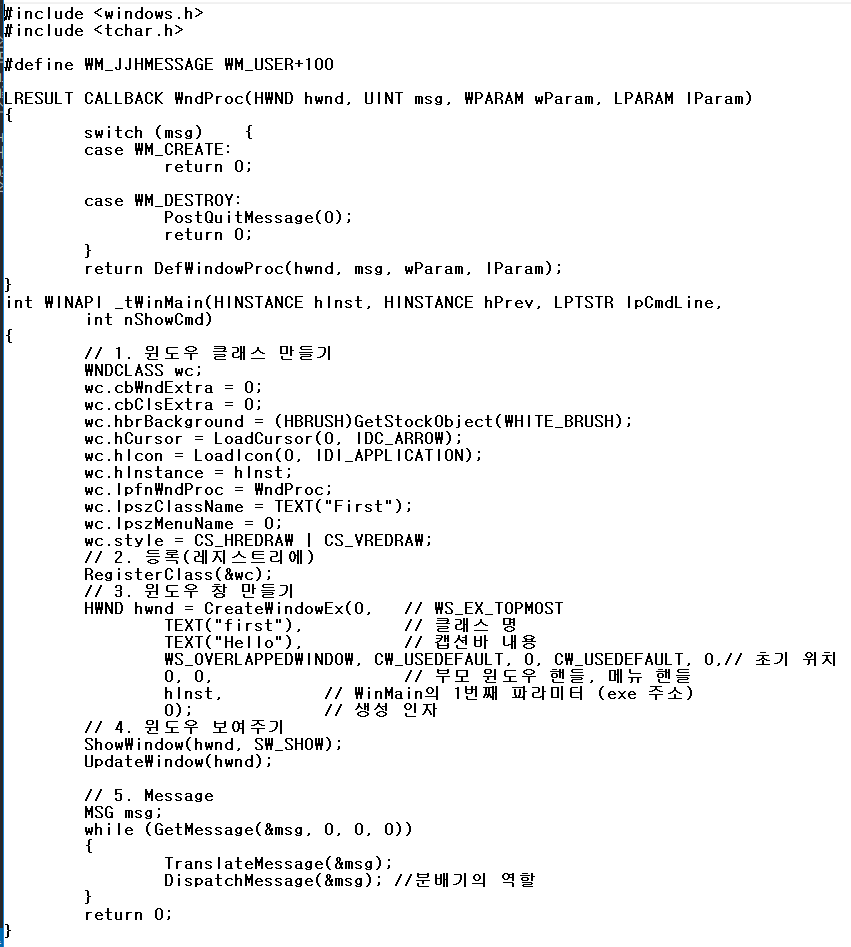
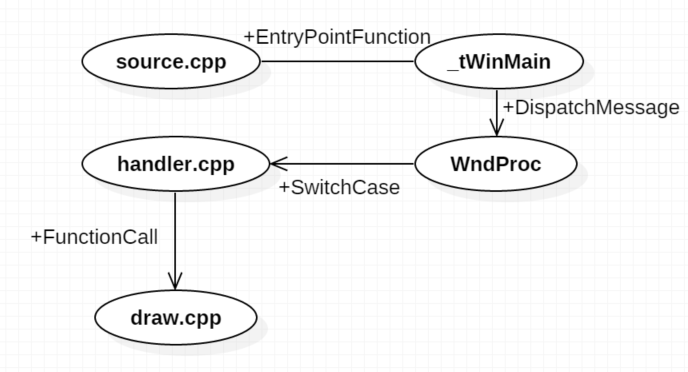
Window API \_Code 정리

* **Skelton Code Pattern**
  + Basic Code Pattern

Source.cpp



* + Bit27기 Code Pattern



* **Window Application System 실행 과정**

1. OS 에서 Application 실행
   1. Application handle이 분배기의 Application Queue에 전달됨
   2. Application에서 Message를 Message Queue에 보냄
   3. 분배기가 Message Queue에서 Message를 추출 후 분석
   4. 분석한 Message 값을 Application Queue에 있는 핸들 리스트에 대조해서 적절한 Application에 메시지 값을 전달함
2. Application 내부 상황
   1. GetMessage를 통해 Message Queue에서 Message를 달라고 요청함
   2. 1.C를 수행한 뒤 Message를 전달받아 MSG 타입의 변수에 저장함
   3. 받은 Message를 자신이 생성한 Application class의 프로시저 함수로 보내기 위해서 DispatchMessage를 이용함 이 함수는 윈도우 프로시저로 메시지를 전달하여 분배기가 메시지를 Application class에 setting한 프로시저에 전달 할 수 있도록 한다.
   4. 프로시저 함수 단에서 Message 값에 맞춰 처리함
   5. 처리하지 못한 Message 값은 DefWindowProc에 전달해서 처리함
   6. 처리가 완료되면 while을 이용한 루프를 만들지 않으면 프로그램이 종료하게된다.

* **Message에 대해서**
  + How to make Custom Message
    - WM\_USER 값 이후의 값들을 Custom Message로 이용 할 수 있음
    - #define 만들고싶은이름 WM\_USER + 양의정수값 으로 메시지 값을 설정 할 수 있다.
    - 보통 IPC 통신을 하려고 할 때 사용된다.
    - ~~그래봐야 숫자 놀이….~~
  + Function

|  |  |  |  |
| --- | --- | --- | --- |
| 함수 이름 | 반환형 | 매개변수 | 내용 |
| GetMessage | BOOL | LPMSG lpMsg,  HWND  hWnd,  UINT  wMsgFilterMin,  UINT  wMsgFilterMax | 메시지 큐에서 메시지를 가져오는 함수,메시지가 없으면 무한 대기 한다.  ipMsg에 MSG 타입의 변수 주소를 넣는다. 나머지는 0으로 해도 좋다. |
| TranslateMessage | BOOL | const MSG \* lpMsg | 가상 키 메시지를 문자 메시지로 변환한다. |
| DispatchMessage | LRESULT | const MSG \*lpmsg | 받은 메시지를 윈도우 프로시저에 전달한다. |
| SendMessage | LRESULT | \_In\_ HWND   hWnd,  \_In\_ UINT   Msg,  \_In\_ WPARAM wParam,  \_In\_ LPARAM lParam | hWnd의 핸들 값을 가진 윈도우에 Msg 값을 전달하는 함수 wParam, lParam에 값을 추가해서 전달 할 수 있다. |
| PostMessage | BOOL | \_In\_opt\_ HWND   hWnd,  \_In\_     UINT   Msg,  \_In\_     WPARAM wParam,  \_In\_     LPARAM lParam | SendMessage와 하는 짓은 같다 하지만 이 값은 bool 값을 성공값을 반환한다. 또 메시지를 던져 놓고 다른 작업을 할 수 있다. |
| PeekMessage | BOOL | \_Out\_    LPMSG lpMsg,  \_In\_opt\_ HWND  hWnd,  \_In\_     UINT  wMsgFilterMin,  \_In\_     UINT  wMsgFilterMax,  \_In\_     UINT  wRemoveMsg | 메시지 큐에 있는 메시지를메세지 lpMsg에 넣는다.  hWnd값이 0이면 현재 윈도우에 맞는 메시지를 검색한다.wMsg…0해도 됨  wRemoveMsg는 PM\_REMOVE는 큐에서 메세지를 제거 PM\_NOREMOVE는 메시지를 제거 안함  이 함수가 true면 메시지가 있다는 의미 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

* + Window Message 상수 값 리스트
    - System 정의 상수 값
      * 요령

시스템 상태를 전달하는 용도로 사용되는 경우가 많다.

* + - * 리스트 (대표 적인 비 큐 메세지다)

|  |  |
| --- | --- |
| 이름 | 내용 |
| WM\_CREATE | 윈도우 인스턴스가 생성되어 지면 날라오는 값 |
| WM\_DESTROY | 메시지 큐에 있는 WM\_QUIT 값을 가져 오면 실행되는 값 |
| WM\_PAINT | 무효화 영역이 발생하면 OS로부터 날라오는 메시지 계속 날라온다. 진짜 미친 듯이 날라온다.  ,메시지 큐에 좌표 값 만 업데이트 되고 2번 이상 들어오지는 않는다.  즉 2개 이상 메시지가 존재하지 않다는 의미 하지만 메시지는 계속 날라온다. |
|  |  |
|  |  |
|  |  |

* + - 키보드 입력
      * 요령

키보드 입력 값의 경우 실제 문자열이 들어오는 것이 아니라. 어셈블리 값이 들어오게 된다. 따라서 아무리 들어오는 메시지 값을 출력하려고 해도 원하는 문자열이 출력되지는 않는다.

또한 문자열 키패드 부분은 일반 KeyDown으로는 처리하기가 곤란하기에 그 처리 과정이 여타 Key들과는 다르다.

* + - * Function Key

WM\_KEYDOWN :

Switch(wParam)

{ case 해당 function key값 :}

=============================

VK\_SPACE – space bar 값

VK\_DOWN – 방향키 아래 값

* + - * 문자 Key

Switch(msg)

{

case WM\_CHAR :

switch(wParam) :

}

wParam에 값이 들어오는데 이 값을 (TCHAR)형태로 강제 캐스팅 하면 문자를 받을 수 있다.

* + - 마우스 입력

Switch(msg)

{

Case 해당 마우스 버튼 값:

}

=====================================

WM\_MBUTTONDOWN: 가운데 값

WM\_LBUTTONDOWN: 왼쪽 버튼 값

WM\_RBUTTONDOWN: 오른쪽 버튼 값

WM\_MOUSEMOVE : 마우스가 움직일 때 들어오는 값

* **GDI 모듈**
* InvalidateArea (무효화 영역)에 대해서
  + 무효화 영역이 필요한 이유

이미 gdi로 그려진 위치에는 다른 그림을 그릴 수 없다.

* + 무효화 영역 발생 시기

윈도우를 옮길 때

윈도우를 제거 했을 때

감추어졌던 윈도우 영역이 보이게 될 때

윈도우의 크기를 조정했을 때

ScrollWindwo(),ScrollDC()함수를 사용하는 경우

InvalidateRect(),InvalidateRgn()함수를 사용하는 경우

윈도우의 일부에 겹친 DialogBox나 메시지 박스가 제거 될 대

메뉴가 나타났다 사라질 때

풍선 도움말이 나타났을 때

* + 구조체

|  |  |  |
| --- | --- | --- |
| 이름 | 멤버 | 내용 |
| PAINTSTRUCT | HDC hdc; // dc handle  BOOL fErase; //배경을지운경우 FALSE, 지우지 않은 경우 TRUE  RECT rcPaint; // 클리핑 영역을 포함하는 최소 사각형  BOOL fRestore; //Reserved; used internally by the system  BOOL fIncUpdate; //Reserved; used internally by the system  BYTErgbReserved[32]; //Reserved; used internally by the system | 이 구조체는 발생된 무효화 영역에 대한 여러가지 정보를 얻어 오기 위해 사용한다. |
|  |  |  |

* + Function

|  |  |  |  |
| --- | --- | --- | --- |
| 이름 | 반환값 | 매개변수 | 내용 |
| BeginPaint | HDC | \_In\_  HWND  hwnd,  \_Out\_ LPPAINTSTRUCT lpPaint | WM\_PAINT 처리 루틴에서 DC를 얻을 때 사용해야 하는 특수 함수  DC를 만들고 UpdateRegion을 DC의 Cliping 영역으로 지정한다.  무효화 영역을 지우기 위해 WM\_ERASEBKGND 메시지를 보낸다  PAINTSTRUCT 구조체를 채운다.  Caret이 있을 경우 그리기 전에 Hide하고 EndPaint()에서 Show한다.  무효화 영역을 유효화 한다. |
| EndPaint | BOOL | HWND hwnd  Const PAINTSTRUCT \* lpPaint |  |
| GetDC | HDC |  | DC핸들을 얻는 일반적인 방법 |
| InvalidateRect | BOOL | \_In\_       HWND hWnd,  \_In\_ const RECT \*lpRect,  \_In\_       BOOL bErase | 일정한 공간 lpRect영역에 강제로 무효화 현상이 일어나게 하는 함수  hWnd 무효화할 해당 윈도우 핸들  lpRect 무효화 영역  bErase는 True면 바로 적용됨  false는 나중에 적용된 왜냐면 메시지 큐에 WM\_PAINT 메시지메 넣고 그 다음에 반영되게 끔 하기 때문이다. |
| InvalidateRgn |  |  | 강제로 무효화 현상이 일어나게 하는 함수 |

* + 클리핑
    - 무효영역 중에서도 화면에 보이는 가시 영역을 말한다. 그려져야 할 부분 중에서도 그리기가 허가된 영역
* **UI 모듈**
  + Define 문
* **Custom Function Library**
  + 윈도우
  + 글자
  + GDI

ㅇㅇㅇㅇㅇㅇㅇ