윈도우 프로시저(Window Procedure)라는 뜻으로 WndProc를 말하며, WinMain에서 호출하는 것이 아니라 운영체제에 의해 호출된다.

WinMain내의 메시지 루프는 메시지를 메시지 처리 함수로 보내기만 할 뿐이며 WndProc은 메시지가 입력되면 운영체제에 의해 호출되어 메시지를 처리한다.

운영체제에 의해 호출되는 응용 프로그램 내의 함수를 콜백 함수라고 한다.

GetMessage()의 단점

메세지가 들어올때까지 대기 상태

PeekMessage()의 활용

[공통점]

두 함수는 모두 메세지 큐로부터 메세지를 가져온다. 두 함수는 동일한 매개변수 4개를 가진다.

그중에 첫 번째 인자로 MSG 구조체를 받는다. 메세지 큐로부터 얻은 메세지 값을 MSG 구조체에 저장한다

[차이점]

GetMessage 함수는 메시지가 도착하여야지만 리턴되는데 반해(blocking 함수임),

PeekMessage 함수는 메시지 큐에 도착한 메시지가 없어도 그냥 리턴한다(non-blocking 함수임)는 점이다.

[GetMessage()]

GetMessage()는 메세지 큐에 메세지가 존재한다면 가져와서 MSG 구조체에 그 값을 저장하고 TRUE를 반환한다.

그런데 만약 읽은 메세지가 WM\_QUIT이면 FALSE를 리턴한다. WM\_QUIT은 메세지 루프를 종료하기 위한 조건으로 사용된다. 만약 메세지 큐에 메세지가 존재하지 않으면 메세지가 생성될 때까지 무한정 기다린다.

이는 scanf 또는 cin과 같은 함수 사용 시 사용자가 입력할 때까지 무한정 기다리는 것과 같다. 때문에 이러한 상황을 생각하여 사용하기 적절한지 상황인지 생각해야 한다.

이러한 메세지들을 가져오면 TranslateMessage() 라는 함수를 통해 번역을 하고, DispatchMessage()라는 함수를 통해 메세지 처리 함수(WndProc)으로 전달한다.

WndProc이 최종적으로 해당 메세지를 처리한다. 윈도우의 모든 메세지는 이처럼 3개의 함수를 거친다.

[PeekMessage()]

PeekMessage()는 메세지 큐에 메세지가 존재한다면 가져와서 MSG 구조체에 그 값을 저장하고 0이 아닌 값을 반환한다.

해당 함수는 메세지 큐가 비어있을 때 GetMessage() 처럼 무한정 기다리지 않고 바로 0을 리턴한다.

때문에 PeekMessage() 함수는 무한 대기에 빠지지 않고, 메세지 큐가 비었다면 다음과 같이 다른 작업을 해줄 수 있다.

해당 함수도 메세지를 읽어오면 GetMessage와 동일하게 메세지가 번역되어 메세지 처리 함수로 보내진다.

Unity

그래픽 파이프라인

|  |
| --- |
| 랜더링 파이프라인 |
| 그래픽스 API 초기화 |
| GPU 디바이스 생성// 현재 기기의 GPU를 추상적으로 표현하는 오브젝트, 커맨드 큐에서 명령을 받아들여 실행. |
| 커맨드큐 생성//GPU에게 전달할 명령을 쌓아두는 곳, CPU와 GPU가 직접 통신하지 않기 때문에 GPU가 성능이 더 좋아서 |
| 랜더링 파이프라인 상태 생성// |
| 드로우콜 생성 -> 커맨드 버퍼를 커맨드 큐에 전달 |
| 랜더링 파이프라인 |
| 정점 조립 |
| 버텍스 셰이더 |
| 래스트라이저 |
| 프래그먼트셰이더 |

D3D

그래픽 파이프라인

Textured Surface

Texture Sampler

Pixcel Rendering

Pixcel Processing

Geometry Processing

Vertex Processing

Tessellation

Primitive Data

Vertex Data