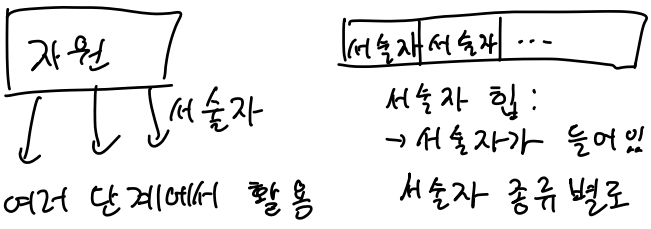
ComPtr 메소드
- Get : ptr ↔ 대입 연산자
- GetAddressOf : ptr의 주소 ↔ &(주솟값) 연산
- Reset

텍스쳐 형식 : DXGI_FORMAT_(색상 형식)_(자료형)

swap chain
- ResizeBuffers : 크기 변경
- Present : 버퍼 제시 (presenting, 버퍼 전환)

Descriptor (= view) : 자원을 파이프라인에 묶는 역할



여러 단계에서 활용

서술자 힙 :
→ 서술자가 들어있는 배열
서술자 종류별로 필요

종류
- CBV (constant buffer)
- SRV (shader resource)
- UAV (unordered access)
- sampler view
- RTV (render target)
- DSV (depth/stencil)

초기화 시점에서 생성할 것

무형식 (typeless)는 필요한 경우만

Feature level : 하드웨어가 지원하는 D3D 레벨

DXGI : 공통적인 기능들 처리 (ex : swap chain, 전체 화면 전환)

IDXGIFactory : DXGI의 핵심 인터페이스 중 하나
- 주요 기능 : IDXGISwapChain 생성, 디스플레이 어댑터 열거(enum)

디스플레이 모드 열거 과정

① 어댑터 열거 (from IDXGIFactory) ⇒ IDXGIAdapter 리스트

② 각 어댑터의 디스플레이 모드 출력 (from IDXGIAdapter)

기능 지원 점검 (CheckFeatureSupport) : Feature Level, MSAA등을 지원하는지 검사

Command queue : Command list 들의 대기열, GPU가 실행

⇒ ID3D12CommandQueue

Command list ⇒ ID3D12CommandList 상속⇒ ID3D12GraphicsCommandList

Command Allocator : Command List에 추가된 명령을 메모리에 저장하는 할당자

Command list 생성 과정

① ID3D12Device::CreateCommandAllocator로 할당자 생성

② ID3D12Device::CreateCommandList로 목록 생성, 이때 할당자를 넘겨줌

③ Command List의 메서드들을 호출하여 명령 추가

④ ID3D12GraphicsCommandList::Close를 호출하여 기록을 끝냄

⑤ ID3D12CommandQueue::ExecateCommandLists를 호출해 명령 실행

⑥ Command List 초기화, 명령들은 할당자에 남아있으므로 상관없음

⑦ Allocator 초기화, 이때는 GPU가 모두 실행하기 전까지 초기화하면 안됨

ⓗ의 문제를 해결하기 위해 Fence를 사용 (경계선 같은 거)
Command queue의 Signal 메서드 활용
64비트 숫자로 몇 번째 Fence 인지 알아냄
마지막에 넣은 Fence에 도달할 때까지 프로그램 대기 (효율적 X)

Transition : 자원의 상태를 변경 (resource hazard 를 회피) → ex: 자원을 기록하는 중 읽을 때
4장에서는 백버퍼를 present → render target
→ present 로 변경

구현
WARP (Windows Advanced Rasterization Platform)
→ 소프트웨어 래스터라이저 : D3D12 장치 생성 실패시 사용