**Direct3D 초기화**

**1.기본 지식**

**1-1. Direct3D 12**

Direc3D – 응용 프로그램에서 GPU를 제어하고 프로그래밍하는 데 쓰이는 저수준 그래픽 API이다.

※**GPU** = graphics processing unit : 그래픽 처리 장치

※**API** = application programming interface : 응용 프로그래밍 인터페이스

**Direct3D 12 변화**

- CPU 부담을 크게 줄이고 다중 스레드 지원을 개선을 위한 재설계.

- 전 버전 보다 훨씬 낮은 수준 API가 됨 (GPU 쪽에 가깝게)

- 이전보다 추상화가 줄고 개발자가 손수 관리해야 할 사항 증가.

- 현 세대 GPU 구조들을 좀 더 밀접하게 반영함.

- 사용하긴 어려우나 성능이 개선됨.

**1-2. COM ( Component Object Model )**

- DirectX의 프로그래밍 언어 독립성과 하위 호환성을 가능하게 하는 기술.

- COM 객체는 참조 횟수가 0이 되면 메모리 해제됨.

**1-3. 텍스처 형식**

- 텍스쳐의 각 원소는 픽셀 하나의 색상을 담는다. (그 뿐 아니라 다양한 용도로 사용됨)

- 특정 형식(format)의 자료 원소들만 담을 수 있는데, 구체적인 형식은 DXGI\_FORMAT이라는 열거형으로 지정.

**1-4. 교환 사슬과 페이지 전환**

- **전면 버퍼(front buffer)**와 화면 바깥의 텍스처 **후면 버퍼(back buffer)**가 서로 역할을 교환하며 껌뻑이는 현상을 피하는 기법을 **‘이중 버퍼링’**이라 한다.

- 전면 버퍼와 후면 버퍼는 하나의 **교환 사슬(swap chain)**을 형성.

※ **IDXGISwapChin** - 교환 사슬을 대표하는 인터페이스

※ **제시(Presenting)** – 후면 버퍼와 전면 버퍼의 역할을 교환해 페이지가 전환되게 하는 것.

**1-5. 깊이 버퍼링 (Depth buffer)**

- 각 픽셀의 깊이 정보를 담는다.

**1-6. 지원과 서술자**

- 그리기 명령을 제출 전에, 먼제 해당 그리기 호출이 참조할 자원들을 렌더링 파이프라인에 **묶어야(bind)**한다. 이를 가리켜 자원을 파이프라인에 **“연결한다(link)”** 또는 **“바인딩(binding)한다.”**

- 서술자는 자원 자료를 지정하는 수단일 뿐만 아니라, 자원을 GPU에 서술하는 수단.

- **서술자 = 뷰(View)**

- 응용 프로그램이 사용하는 서술자들이 저장되는 곳이 서술자 힙.

- 서술자 종류마다 개별적 서술자 힙이 필요.

- 같은 종류의 서술자들은 같은 서술자 힙에 저장.

- 한 종류의 서술자에 대해 여러 개의 힙을 둘 수 있음.

- 서술자들은 응용 프로그램의 초기화 시점에서 생성해야 함. 이는 그때 일정 정도의 형식 점검과 유효성 검증이 일어나기 때문이고 또한 초기화 시점에서 생성하는 것이 실제 실행 시점에서 생성하는 것보다 낫기 때문.

※ **서술자(descriptor) 객체** – 자원을 GPU에게 서술해주는 경량의 자료구조.

(본질적으로 이는 하나의 간접층 (Level of indirection)).

(GPU 자원들이 파이프라인에 직접 묶이는 것이 아니라 실제로 파이프라인에 묶이는 것은 해당 자원을 참조하는 이 객체가 묶는 것.)

※ **CBV** = contant buffer(상수 버퍼)

※ **SRV** = shader resource(셰이더 자원)

※ **UAV** = unordered access view (순서 없는 접근)

※ **RTV** = Render target

※ **DSV** = Depth / Stencil

※ **서술자 힙(descriptor heap)** = 서술자들의 배열.

**1-7. 다중표본화의 이론**

- 엘리어싱 제거 기법엔 초과표본화와 다중표본화가 있다.

※ **앨리어싱(Aliasing) 효과** – 계단 현상

※ **초과표본화(supersampling)**

– 후면 버퍼와 깊이 버퍼를 화면 해상도보다 4배(가로, 세로 두배씩) 크게 잡고, 3차원 장면을 4배 크기의 해상도에서 후면 버퍼에 렌더링한다. 이미지를 화면에 제시할 때가 되면 후면 버퍼를 원래 크기의 버퍼로 환원한다.

픽셀 처리량과 메모리 소비량이 4배라 비용이 높다.

(하향표본화(downsampling) 이라고도 한다.)

※ **다중표본화(multisampling)**

- 일부 계산 결과를 부분픽셀(subpixel)들 사이에서 공유하기 때문에 초과표본화보다 비용이 낮다.

- 이미지 색상을 각 부분픽셀마다 계산하는 것이 아닌 픽셀당 한 번만 계산(픽셀 중심에서), 그 그 생상과 부분픽셀들의 **가시성**(이를 위해 부분픽셀당 깊이.스텐실 판정)과 **포괄도**(부분픽셀을 다각형이 어느 정도나 덮고 있는지를 뜻하는 값)를 이용해 최종 색상을 결정.

**전체적 초기화 과정**

1. ID3D12Device 생성

2. ID3D12Fence 객체 생성 후 서술자들 크기 얻기

3. 4X MSAA(Multisample anti-aliasing) 품질 수준 지원 여부 점검.

4. 명령 대기열, 명령 목록 할당자, 주 명령 목록 생성

5 교환 사슬 서술 후 생성.

6. 필요한 서술자 힙들을 생성.

7. 후면 버퍼의 크기 설정, 후면 버퍼에 대한 렌더 대상 뷰 생성.

8. 깊이-스텐실 버퍼 생성, 연관된 깊이-스텐실 뷰 생성.

9. 뷰 포트와 가위 판정용 사각형 설정.

---------------------------------------------------------------

**1. 장치 생성**

장치(Device) – 디스플레이 어댑터를 나타내는 객체

WARP – Windows Advanced Rasterization Platform(Windows 고급 래스터화 플랫폼)

* mdxgiFactory 객체는 교환 사슬을 생성하는데 쓰인다.

(교환 사슬이 Direct3D가 아닌, DXGI의 일부이기 때문)

**2. 울타리 생성, 서술자 크기 얻기**

- CPU, GPU 동기화를 위한 울타리 객체 생성.

- 후에 필요한 서술자들의 크기를 미리 조회하여 설정.

**3. 4X MSAA 품질 수준 지원점검**

- 4X MSAA 항상 지원

**4. 명령 대기열, 명령 목록 생성**

ID3D12CommandQueue – 명령 대기열을 대표하는 인터페이스

ID3D12CommandAllocator – 명령 할당자를 대표하는 인터페이스

ID3D12GraphicCommandList – 명령 목록을 대표하는 인터페이스

**5. 교환 사슬의 서술과 생성**

DXGI\_SWAP\_CHAIN\_DESC 구조체 인스턴스의 멤버들을 생성하고자 하는 교환 사슬에 맞게 설정.

**6. 서술자 힙 생성**

응용 프로그램에 필요한 서술자/뷰들을 담을 서술자 힙을 생성.

ID3D12DescriptorHeap 인터페이스로 대표.

렌더 대상 뷰 – 교환 사슬에서 렌더링의 대상이 되는 버퍼 자원을 서술.

깊이\_스텐실 뷰 – 깊이 판정을 위한 버퍼 자원을 서술.

※ CreateEventEx(LPSECURITY\_ATTRIBUTES lpEventAttributes, LPCWSTR lpName, DWORD dwFlags, DWORD dwDesiredAccess)

- 이름이 정해진 또는 이름이 없는 이벤트 객체를 열거나 생성하고 핸들 값을 반환하는 함수.

**7. 렌더 대상 뷰(RTV) 생성**

-반드시 자원에 대한 뷰(서술자)를 생성해 그 뷰를 파이프라인 단계에 묶어야 함.

-특히, 후면 버퍼를 파이프라인의 출력 병합기(outpur merger) 단계에 묶으려면 후면 버퍼에 대한 렌더 대상 뷰를 생성해야 함.

-ID3D12Device::CreateRenderTargetView 메서드를 이용해 생성.

**8. 깊이\_스텐실 버퍼와 뷰 생성**

깊이 버퍼 – 그냥 가장 가까운 가시 물체들의 깊이 정보를 저장하는 2차원 텍스처

- GPU 자원들은 GPU 힙에 존재

- GPU 힙은 GPU 메모리 블록인데, 특정 속성을 가지고 있다.

- 깊이\_스텐실 버퍼를 사용하기 전에 반드시 연관된 깊이\_스텐실 뷰를 생성해 파이프라인에 묶도록 한다.

**9. 뷰 포트 설정**

뷰포트 – 장면을 그려 넣고자 하는 후면 버퍼의 부분직각사각형(Subrectangle) 영역.

※하나의 렌더 대상에 여러 개의 뷰포트를 지정할 수 없다.

※명령 목록을 재설정(Reset)하면 뷰포트들도 재설정 해야함.

**10. 가위 직사각형 설정**

가위 직사각형 (scissor rectangle) – 특정 픽셀들을 선별(culling)하는 용도.

후면 버퍼를 기준으로 가위 직각사각형을 정의, 설정하면, 렌더링 시 가위 직각사각형의 바깥의 픽셀들은 후면 버퍼에서 레스터화 되지 않음 (최적화 기법)

※하나의 렌더 대상에 여러 개의 가위 직사각형을 지정할 수 없다.

※명령 목록을 재설정(Reset)하면 가위 직사각형들도 재설정 해야함.