**Direct3D의 그리기 연산 제2부**

**7.1 프레임 자원**

- CPU와 GPU는 병렬로 작동

- CPU는 명령 목록들을 구축하고 제출.

- GPU는 명령 대기열에 담긴 명령들을 처리.

**※ FlushCommandQueue** 함수를 호출하여 해당 프레임의 모든 명령이 실행되기를 기다림. 동기화 문제를 확실히 해결하지만 비효율적임.

1. 한 프레임의 시작에서 GPU는 처리할 명령이 하나 없는 상황에 부닥침.

2. 반대로, 한 프레임 끝에서 CPU와 GPU가 명령들을 모두 처리할 떄까지 기다려야 함.

**※ 프레임 자원 (Frame resourse)** - 매 프레임 CPU가 수행해야 하는 자원들을 순환 배열(circular array)로 관리.

1. 프레임 n에서 CPU는 프레임 자원 배열을 훑으며 다음 번 가용 프레임 지원(즉, GPU가 사용하지 않는)을 찾음.

2. GPU가 이전 프레임을 처리하는 동안 CPU는 그런 가용 프레임 자원을 적절히 갱신하고 프레임 n을 위한 명령 목록들을 구축해서 제출

3. 프레임 n+1로 넘어가서 같은 과정을 반복

4. 프레임 자원 배열의 원소가 세 개라면 CPU는 GPU보다 최대 두 프레임 앞서갈 수 있으며, GPU는 CPU를 따라잡기 위해 쉼 없이 일하게 됨.

**7.2 렌더 항목**

- 하나의 완전한 그리기 호출 명령을 렌더링 파이프라인에 제출하는데 필요한 자료 집합을 **렌더 항목(Render Item)**이라 한다.

**7.3 패스별 상수 버퍼**

- 이 버퍼는 하나의 렌더링 패스 전체에서 변하지 않는 상수 자료를 저장.

**※** 셰이더가 사용하는 상수 버퍼가 너무 많아지지 않도록 할 것. [Thibieroz13]은 성능을 위해서는 그 수를 5 미만으로 두라고 권하고 있다.

**7.5 도형 예제**

**7.5.3 프레임 자원들과 상수 버퍼 뷰들**

- 여러 FrameResource들을 벡터 자료구조에 보관

- 각 FrameResource에는 패스별 상수 버퍼들과 장면의 모든 렌더 항목을 위한 상수 버퍼들을 담는 업로드 버퍼가 있음.

- 프레임 자원 3개, 렌더 항목 n개일 때, 전체적으로 물체별 상수 버퍼가 3n개, 패스별 상수 버퍼가 3개 다라서 총 3(n+1)개 상수 버퍼 뷰(CBV)가 필요.

- 힙의 첫 서술자의 핸들을 **ID3D12DescriptorHeap::GetCPUDescriptorHandleForHeapStart 메서드**로얻을 수 있음.

- 서술자 오프셋 증가치를 알고 있으면 두 **CD3DX12\_CPU\_DESCRIPTOR\_HANDLE::Offset 메서드** 중 하나를 이용해서 n번 서술자 핸들의 오프셋을 구할 수 있음.

**7.6 루트 서명 추가 설명**

**※ 루트 서명** - 그리기 명령을 제출하기 전에 파이프라인에 묶어야 할 자원들이 무엇이고 그 자원들이 셰이더 입력 레지스터들에 어떻게 대응되는지를 정의.

**7.6.1 루트 매개변수**

- 루트 서명이 루트 매개변수들의 배열로 정의.

- 하나의 루트 서명에는 최대 64개의 DWORD만을 넣을 수 있음.

- DWORD 64개라는 최대 용량을 넘기지 않는다면 세 종류의 루트 매개변수들을 임의의 방식으로 조합해서 루트 서명을 만들 수 있음.

- 루트 매개변수를 나타낼 때에는 **CD3DX12\_ROOT\_PARAMETER 구조체**를 사용.

**※ 루트 매개변수 종류**

1. **서술자 테이블 :** 힙 안에 있는 이련의 서술자들(묶을 자원들을 식별하는)의 구간을 지정.

2. **루트 서술자(Root descriptor, 인라인 서술자)**

묶을 자원을 직접 식별하는 하나의 서술자.

힙에 둘 필요가 없다.

상수 버퍼에 대한 CBV나 기타 자원 버퍼들에 대한 SRV/UAV만 루트 서술자로 묶을 수 있음.

텍스처에 대한 SRV는 루트 서술자로 묶을 수 없음.

3. **루트 상수(Root constant) :** 직접 묶을 32비트 상수 값들의 목록.

**※ 세 가지 루트 매개변수 비용**

1. **서술자 테이블 :** DWORD 하나

2. **루트 서술자 :** DWORD 두 개

3. **루트 상수 :** 32비트 상수당 DWORD 하나.

**7.6.2 서술자 테이블**

- 서술자 테이블 루트 매개변수를 정의하려면 **D3D12\_ROOT\_PARAMETER**의 DescriptorTable 멤버를 채워서 서술자 테이블에 대한 좀 더 구체적인 정보를 제공.

- 슬롯 매개변수 하나를 서술자 테이블로서 초기화하려면 **D3D12\_DESCRIPTOR\_RANGE 인스턴스**들의 배열을 만들어야 함.

- 하나의 테이블에 다양한 종류의 서술자들을 담을 수 있으므로 이런 배열이 필요.

- **D3D12\_DESCRIPTOR\_RANGE**를확장해서 편의용 메서드들을 추가하는 **CD3DX12\_DESCRIPTOR\_RANGE라는 구조체**.

- CBV 구간, SRV 구간, UAV 구간이 모두 레지스터 0으로 시작하지만, 시작 번호가 같을 뿐이지 묶이는 레지스터들의 종류는 각자 다르므로 '**중첩' 충돌**이발생하지 않음.

**7.6.3 루트 서술자**

- 루트 서술자 루트 매개변수를 정의하려면 **D3D12\_ROOT\_PARAMETER**의 Descriptor 멤버를 채워서 루트 서술자에 대한 좀 더 구체적인 정보를 제공해야 함.

**7.6.4 루트 상수**

- 루트 상수 루트 매배변수를 정의하려면 **D3D12\_ROOT\_PARAMETER**의 Constants 멤버를 채워서 루트 상수에 대한 좀 더 구체적인 정보를 제공해야 함.

- 셰이더의 관점에서, 응용 프로그램이 설정한 루트 상수는 그냥 상수 버퍼의 한 자료와 다를 바 없음.

**7.6.6 루트 인수의 버전 적용**

- 하드웨어가 각 그리기 호출 시점에서의 루트 인수들의 상태('스냅숏')를 자동으로 저장.

- 루트 서명이 셰이더가 실제로 사용하는 것보다 많은 필드를 제공할 수 있음을 주의.

- 루트 서명의 크기를 최소한으로 유지하는 것이 바람직.

- 루트 서명이 크면 루트 인수들의 그러한 스냅숏들도 커짐.

- 루트 서명의 전환을 최대한 피하라고 권함.

- 응용 프로그램이 생성하는 여러 PSO들이 가능하면 동일한 루트 서명을 공유하게 하는 것이 바람직.

- 여러 셰이더 프로그램들과 호환되는 '슈퍼'루트 서명을 두는 것도 도움이 됨.

**※ 루트 인수(Root argument)** - 응용 프로그램이 루트 매개변수에 실제로 전달하는 값.