**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.