**Direct3D의 그리기 연산**

**6-1. 정점과 입력 배치**

- 원하는 자료(‘특성’)를 가진 커스텀 **정점 형식(vertex format)**을 만들려면 우선 그러한 자료를 담을 구조체를 정의

**※ 입력 배치 서술(input layout description)** - 정점의 각 성분으로 무엇을 해야 하는지를 Direct3D에 알려주는 수단이며, **D3D12\_INPUT\_LAYOUT\_DESC**라는 구조체로 대표.

**※ D3D12\_INPUT\_ELEMENT\_DESC 배열** – 각 원소는 정점 구조체의 각 성분을 서술하고 이 배열의 원소들과 정점 구조체의 성분들은 일대일로 대응한다.

**6-2. 점정 버퍼**

- GPU가 점들의 배열에 접근하려면, 그 정점들은 **버퍼(Buffer)**라 불리는 **GPU 자원(ID3D12Resource)**에 넣어 두어야 함.

- 응용 프로그램에서 정점 같은 자료 원소들의 배열을 GPU에 제공해야 할 때에는 항상 버퍼를 사용.

- **정점 기하구조(=프레임마다 변하지 않는 기하구조)**를 그릴 때에는 최적의 성능을 위해 정점 버퍼들을 **기본 힘(D3D12\_HEAP\_TYPE\_DEFAULT)**에 넣는다.

- 정점 버퍼를 초기화한 후에는 GPU만 버퍼의 정점들을 읽으므로(기하구조를 그리려고), 기본 힙에 넣는 것이 합당하고, CPU는 기본 힙에 있는 정점 버퍼를 수정하지 못한다.

- 실제 정점 버퍼 자원을 생성하는 것과 더불어, 응용 프로그램은 **D3D12\_HEAP\_TYPE\_UPLOAD** 형식의 힙에 임시 업로드용 버퍼 자원을 생성해야 함.

- 정점 버퍼를 파이프라인에 묶으려면 정점 버퍼 자원을 서술하는 **정점 버퍼 뷰**를 만들어야 한다.

- 정점 버퍼 뷰에는 서술자 힙이 필요하지 않다.

- 정점 버퍼와 이에 대한 뷰를 생성했으면, 정점 버퍼를 파이프라인의 한 입력 슬롯에 묶을 수 있다. 그러면 정점들이 파이프라인의 입력 조립기 단계로 공급됨.

- 입력 슬롯에 묶은 정점 버퍼는 다시 변경하지 않는 한 계속 그 입력 슬롯에 묶임.

**※ 정점 버퍼(vertex buffer)** – 정점들을 저장하는 버퍼.

**※ 범용 GPU 자원으로서의 버퍼에서 너비(width)** – 가로 길이가 아닌 버퍼의 바이트 개수를 뜻.

**※ D3D12\_VERTEX\_BUFFER\_VIEW\_DESC** – 정점 버퍼 뷰를 대표하는 구조체.

**※ ID3D12GraphicsCommandList::DrawInstanced 메서드** – 정점들을 실제로 그리는 함수.

**6-3 색인과 색인 버퍼**

- GPU가 색인들의 배열에 접근할 수 있으려면 색인들을 버퍼 GPU 자원(ID3D12Resource)에 넣어둬야 함.

- 색인 버퍼를 파이프라인에 묶으려면 색인 버퍼 자원을 서술하는 **색인 버퍼 뷰**를 만들어야 함.

- 정점 버퍼 뷰와 마찬가지로 **서술자 힙이 필요 없다.**

- 색인 버퍼는 **Id3D12CommandList::SetIndexBuffer 메서드**를 통해서 입력 조립기 단계에 묶음.

**※ 색인 버퍼** – 색인(Index)들을 담는 버퍼

**※ D3D12\_INDEX\_BUFFER\_VIEW** – 색인 버퍼 뷰를 대표하는 형식

**※ ID3D12GraphicsCommandList::DrawIndexedInstanced 메서드** – 색인들을 통해 기본도형을 그리는 함수.

**※ 기준 정점 위치(base vertex location)** – 전역 정점 버퍼에서 한 물체의 첫 번째 정점의 위치.

**6-4 예제 정점 셰이더**

- 본질적으로 셰이더는 하나의 함수.

- HLSL에서 함수는 항상 인라인화.

- 첫 입력 매개 변수는 정점 셰이더의 **입력 서명(input signature)**을 형성.

- **입력 매개변수의 의미소**( :POSITION, :COLOR 등)는 정점 구조체의 멤버들을 정점 셰이더 입력 매개변수들에 대응시키는 역할.

- **출력 매개변수의 의미소**(:SV\_POSITION, :COLOR 등)는 정점 셰이더의 출력을 파이프라인의 다음 단계( 기하 셰이더 또는 픽셀 셰이더)의 해당 입력에 대응시키는 역할.

**※ HLSL(High level shading language)** – 셰이더 작성 언어

**※ SV** – 해당 값이 system value 의미소임을 뜻하고 해당 정점 셰이더 출력 성분이 정점의 위치(동차 절단 공간에서의)를 담고 있음.

**6-5 픽셀 셰이더**

- 픽셀 단편마다 실행

- 주어진 입력으로부터 픽셀 단편의 색상을 계산.

- 후면 버퍼의 한 픽셀에는 최종적으로 그 픽셀이 될 수 있는 ‘후보’로서의 픽셀 단편들이 여러 개 존재할 수 있다.

**※ SV\_TARGET 의미소** – 해당 함수의 반환 값의 형식이 랜더 대상(render target)의 형식과 일치해야 함을 의미.