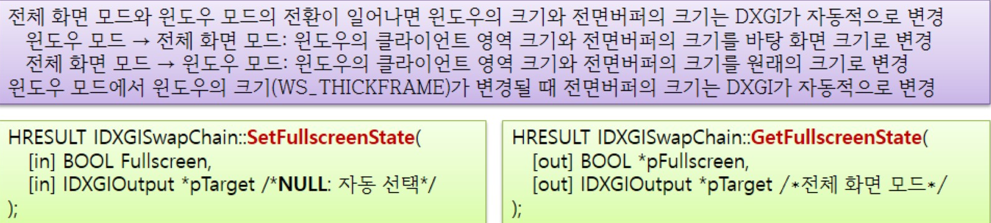
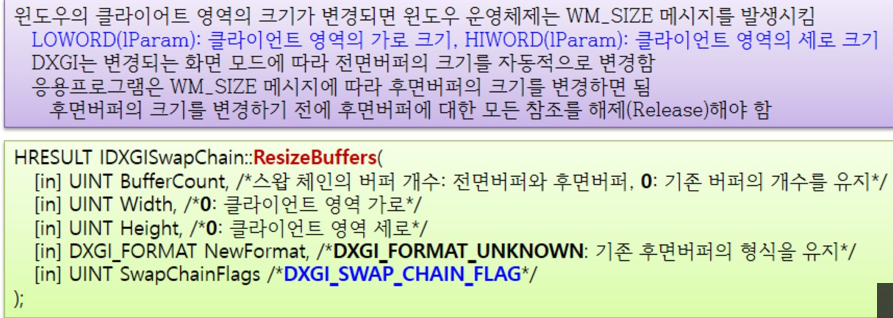
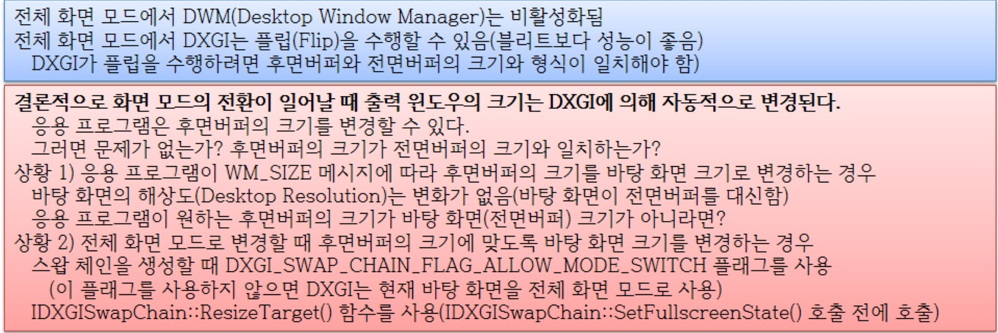
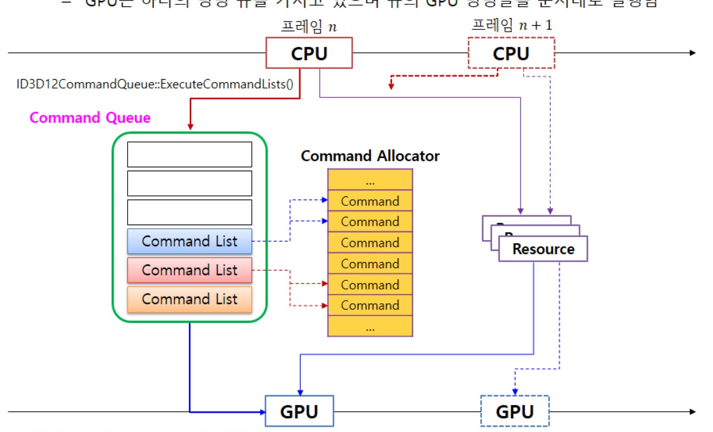
* 4.1.2 콤객체
* 콤객체 == c++클래스
* 콤 인터페이스를 가르키는 포인터를 특별한 함수를 이용해서 다른 콤 인터페이스 메서드를 이용해서 얻음.
* 컴은 release로. .
* Com은 참조횟수가 0이되면 메모리에서 해제
* 4.1.3 텍스쳐형식
* 텍스쳐는 2차원배열. 텍스처의 각 원소는 픽셀하나의 색상 담음.
* 128비트 색상은 정밀도가 높은 색상연산이 필요한곳에 쓰인다. 그러나 픽셀색상은 일반적으로 후면버퍼에 32비트색상으로 저장된다.
* 0726. 스왑체인할 때 창의 크기 🡪 dxgi가 자동으로 변경







frameadvance에서 command\_allocator reset하는건



여기있는 노란 커맨드들을 리셋하는거.

커맨드 얼로케이터는 같은 할당자에서 동시에 두개 이상의 명령리스트가 열린 상태가 될 수 없다🡪 executecommandist 호출해야 비로소 명령들이 명령 대기열에 추가되며, gpu가 그 명령을 뽑아서 실행. 명령들을 명령 목록에

두개이상 열리면안되는건 스고있으면연리면안되는거 싱글스레드는별문제없는데 멀티스레드로가면 추가했으면 close를 호출해서 명령들의 기록이 끝났음을 dx3d에 알려준다. 꼭닫아야함.

알로케ㅔ이터는 메모지묶음이다 냉장고에 붙여서 한장뜯어다써야함

commandlist에는 commandallocator형식의 메모리 할당자가 하나 연관된다. 명령목록에 추가된 명령들은 이 할당자의 메모리에 저장된다. Execute로 실행하면 명령 대기열은 할당자에 담긴 명령을 참조한다. 할당자는 device의 메서를 통해 생성한다.

할당자는 메모리풀이라고 생각하고 커맨드알로케이터에서 할당받아서만들면

씬에다 카메라두면 어떻게되나! 씬카메라 플레이어 독립적으로만들고 잘 연결해보자

씬에서 플레이어 만들면 씬지나가면 ㅡ플레이어는!

카메라 안움직이면 레이쏴서 오클루전맵.

정적인물체들은 미리만들어둠

Lod!중요하다 LOD! 디테일의정도는 애니메이션이든 그림자든조명이든 뭐로든!

콘스탄트버퍼를 따로 떼어서 생각하지말고 GPU가 동작하기 위해서 ㄴ데이터가 필요해 지피유가일을할라면 데이터가필요해 걔네를 리소스라 부러요 그 리소스중에 아주특별한 리소슥아있는데 상수버퍼라는 놈이 있어 상수버퍼라는놈은 C버퍼가됐던 뭐든 프로그램의 변수를 만들수가있어 리소스는 아주특별한규칙이있어 레지스터메모리특징하고 패킹이런게있어 그 리소스들은 기본적으로 리소르를만드는주체는 다렉을사용해서 씨피유에 만들어야대 리소스들을 사용하려면 메모리가 필요해 그게 리소스죠 그래서 컨스턴트버퍼 그리에이트커미티드뭐를마늗면 콘ㅇ스탄트버퍼륾ㄴ들수있어 그 버퍼를 접근할수있게하면 리소스 설명하기위한 정보가 필요해 그 설명하는정보가 디스크립터고 그래픽스 파이프랑니에서 디스크립터 한 개한개다루면 효율 하타ㅇ치니까 쉐이더 셋할때 지금부터 디스크립터가 몇 개가있고 콘스턴트버퍼뷰 그걸한번에 셋할거야 디스크립터들을 모은배열을 할당하는게 더 효ㅠ율적이니까 디스크립터 힙을만들어 디스크립터힙에는 콘스턴트버퍼뷰도있고 머ㅜ도있고 많아하든 힙에있는 거에다가 포인터로 설명할 수 있게 리소스 디스크립터. 타입같은게있어서 만듬. 디스크립터힙이 필요하고 디ㅅ크립터가있어야해. 얘를접근하기위한형태로 만들어야돼. 어떤거를 써야해 하려면 루트시그니처가 있는거 루트시그니처는 쉐이더에 변수들이 어떻게 연결될지 정보를 표현한거. 실제로 똑 같은 규약가지고 데이터들을.리소스들을 어딘가에 메모리롬로만드는게필요햇요 루트시그니터에 따라서 뭐 설명안써도 되는 루트상수. 일반적으로 디스크립터있어야대고.