

▶ 레벨 오브 디테일(LOD)

레벨: 카메라와 모델의 거리

카메라까지 거리에 따라 자동적으로 맵 레벨 선택

레벨에 따라 폴리곤 개수 결정

▶ 맵핑 비교 설명

일반: 객체 표면에 이미지를 덧씌워 색상/질감 표시, 입체감 없음

노멀(패럴랙스): 노멀 맵을 사용하여 빛이 표면에 부딪힐 때 각도를 조정하여 텍스처만으로 입체감 생성

변위(디플레이스먼트): 하이트 맵 사용해 실제 표면의 기하구조를 수정하여 입체감 강화

▶ 지연 렌더링

조명 계산을 후반에 모아서 하는 렌더링 방식

① 기하정보를 G버퍼에 저장

② 조명을 계산하여 최종적으로 합성

▶ 블렌드 스테이트 연결 방법

① D3D12_BLEND_DESC 생성 및 설정

② 렌더 타겟 배열 블렌드 설정

③ D3D12_GRAPHICS_PIPELINE_STATE_DESC 설정

④ ID3D12PipelineState 객체 생성

⑤ 명령 리스트에 PSO 설정

▶ 텍스 스텐실 설정 과정

① 리소스 설명자 설정

② 힙 속성 설정

③ 초기화 값 설정

④ 리소스 생성 및 뷰 생성

▶ 텍스 검사와 스텐실 검사 방법

텍스 검사: 픽셀의 깊이를 기준으로 앞뒤 관계를 판별하여 카메라에 가까운 객체가 화면에 먼저 보이도록 함

스텐실 검사: 마스크와 특정 조건을 설정해 지정된 영역에만 효과 적용

▶ 텍스 스텐실 버퍼 갱신 기준

각 프레임마다 깊이 정보가 갱신

새로운 물체가 렌더링될 때마다 해당 픽셀이 깊이 버퍼의 기존 값보다 가까우면 갱신

텍스는 텍스와 스텐실 모두 true면 갱신

스텐실은 스텐실만 활성화 되도 갱신

▶ 텍스 버퍼와 스텐실 버퍼 사용 용도

텍스 버퍼: Z축의 깊이를 기록해 물체의 화면 위치를 결정하는데 사용

스텐실 버퍼: 특정 픽셀을 마스킹하거나 특수 효과를 적용할 때 사용

▶ 아웃풋 머저

렌더링 파이프라인 최종 단계로 다양한 렌더 타겟을 결합
블렌딩, 텍스 스텐실 검사를 통해 최종 화면 구성

▶ 기하 셰이더

[maxvertexcount(n)]: 각 프리미티브에 대하여 실행되는 기하 셰이더의 인스턴수 개수 n개(최대 32개)

Append(StreamDataType): 출력 데이터를 스트림에 추가

RestartStrip(): 현재의 프리미티브 스트립을 끝내고 새로운 프리미티브 스트립 시작

▶ 스트림 아웃풋 설명

정점 셰이더의 출력을 GPU 메모리의 버퍼에 저장하는 기능

변환된 정점 데이터를 후속 렌더링 패스나 계산을 위해 사용 가능

▶ 스트림 아웃풋 작동 방법

① 스트림 아웃풋 버퍼 설정(동시에 4개 가능)

② 정점 셰이더 출력

③ 스트림 아웃풋 설정

④ 렌더링 및 데이터 전송