



🕶 Week08_Team04팀 – 그림자 렌더링 시스템 기술 문서

팀원 : 김윤수, 김상천, 김민찬, 주소연

🎯 그림자 렌더링 개요

- **지원:** 방향광 / 포인트광 / 스팟광 (DirectX 11 기반)
- **필터 모드:**
 - PCF, VSM, Hard(=NONE)
 - 방향광은 PCF만 지원
 - 스팟/포인트는 PCF 또는 VSM 선택 가능
- **CSM (Cascaded Shadow Map)**
 - 방향광 전용
 - 블렌딩 및 디버그 뷰 기능 제공

🧩 에디터 연동

- 뷰포트 툴바 → ShowFlag 드롭다운
 - Shadows, FXAA, Fog, Tile Culling, 통계 패널 등 토글 지원
- 셜도우 해상도 변경 UI
 - 방향/스팟/포인트 셜도우 해상도 변경 시 실시간 리소스 재할당
- 라이트 속성 패널
 - 라이트별 셜도우 파라미터 및 Depth Map 미리보기 제공
- 콘솔 위젯
 - STAT, SHADOW_FILTER 명령어 제공

섀도우 시스템 구조

구성 요소	경로
코어	Mundi/Source/Runtime/Renderer/ShadowSystem.{h, cpp}
방향광 CSM	Mundi/Source/Runtime/Renderer/CSM.{h, cpp}
통합 지점	SceneRenderer.cpp → RenderShadowMap, RenderDirectionalCSMSHadowMap, RenderOpaquePass
셰이더	- Shaders/Utility/DepthOnly.hlsl (깊이/모멘트 캡처) - Shaders/Materials/UberLit.hlsl (PCF/VSM/Hard 필터) - Shaders/Common/ShadowBuffers.hlsl (섀도우 버퍼 정의)

💡 라이트 타입별 처리

☀️ Directional Light

- 일반 방향 섀도우 + 선택적 CSM (`EDirectionalShadowMode::CSM`)
- 현재 파이프라인에서는 **VSM 미지원, PCF 경로** 사용

🔦 Spot Light

- 깊이 텍스처 배열 (DSV) + 선택적 VSM 모멘트 텍스처 배열 (RG Float, RTV|SRV)

💡 Point Light

- 큐브맵 배열 (DSV) + 선택적 VSM 큐브맵 배열 (RG Float, RTV|SRV)

🧠 필터 모드 (PCF / VSM / Hard)

- 전역 렌더 설정은 `RenderOpaquePass`에서 셰이더 매크로로 전달됨

모드	설명
PCF	포아송 16탭 필터링, 라이트별 <code>ShadowSharpen</code> 반경 적용
VSM	<code>DepthOnly.hlsl</code> 에서 1·2차 모멘트(RG16F) 기록 → <code>UberLit.hlsl</code> 에서 분산 기반 필터링
Hard	하드웨어 비교 전용, 추가 필터 없음

콘솔 명령:

SHADOW_FILTER NONE | PCF | VSM

Directional Light CSM

- 캐스케이드별 ViewProjection 행렬 계산 및 Uber 상수 업데이트
- DirectionalLight 선택 시 제어 항목:
 - Blending On/Off
 - Lambda (분할 분포)
 - Padding (AABB)
 - Debug Slice View
- 미리보기:
 - CSM 사용 시: **캐스케이드별 SRV**
 - 미사용 시: **단일 SRV**

라이트별 컨트롤

- 컴포넌트: **ULightComponentBase**, **ULightComponent**
 - **bCastShadows** – 색도우 사용 토글
 - **ShadowSharpen** – 소프트 ↔ 하드 가장자리 조절
 - **ShadowBias**, **ShadowSlopeBias**, **MaxSlopeDepthBias** – 색도우 깊이 보정
- UI:
 - TargetActor 패널에 슬라이더 제공
 - “**Reset Shadow Values To Default**” 버튼 포함

해상도 컨트롤

- **ShowFlag** 메뉴
 - 방향 / 스팟 / 포인트 해상도 항목 제공
- 내부 처리 흐름:

```
FShadowSystem::Set{Directional, Spot, Point}ShadowTextureResolution  
(...)
```

- 방향광의 경우:
 - **FCSM** 크기 변경과 리소스 재생성 수행

에디터 UI 및 통계

- ShowFlag 메뉴
 - Shadows, Grid, Decals, Fog, FXAA, Tile Culling 등 토글
- 콘솔 명령

```
STAT FPS  
STAT MEMORY  
STAT PICKING  
STAT DECAL  
STAT LIGHT  
STAT SHADOW  
STAT ALL  
STAT NONE
```

- 표시 내용
 - 라이트 종류별 개수
 - 메모리 사용량
 - 타일 컬링 통계

Depth Map 미리보기 (라이트 패널)

라이트 타입	미리보기 내용
Spot	선택된 라이트의 단일 SRV 표시
Point	큐브맵 6면 SRV 표시
Directional	CSM 사용 시 캐스케이드별 SRV / 미사용 시 단일 SRV

에디터 사용 방법

- ShowFlag 드롭다운
 - Shadows / 통계 패널 / 해상도 변경 / FXAA / Fog / Tile Culling 등
- 콘솔 명령

SHADOW_FILTER PCF | VSM | NONE

• 뷰 모드

- Lit(Gouraud / Lambert / Phong) , Unlit , Wireframe , BufferVis(SceneDepth / WorldNormal)
- 라이트별 설정
 - Bias , Sharpen 조절
 - Depth Map 미리보기 확인

구현 메모

- VSM
 - 깊이 대신 1차·2차 모멘트를 기록
 - RTV|SRV 가능한 RG 포맷 (R16G16_FLOAT) 사용
 - Spot/Point 전용 리소스 생성 ([CreateSpotShadowMomentTextureArray](#))
- PCF
 - 하드웨어 비교 샘플러 + 포아송 오프셋 사용
 - 반경은 라이트별 ShadowSharpen 적용
- Directional Light
 - 현재 파이프라인에서 VSM 미지원 → PCF/CSM으로 풀백
- 해상도 변경
 - FShadowSystem 의 해상도별 리소스 캐시를 통해 안전 재할당

빌드 및 실행

- 개발 환경

- Visual Studio 2022 (Windows 10/11, x64)
 - DirectX 11, D3DCompiler, Windows SDK
 - **프로젝트 열기**
 - `Mundi.sln`
 - **실행**
 - 시작 프로젝트: `Mundi`
 - Build → Run
-

주요 파일 구조

구분	파일 경로
렌더 오케스트레이션	<code>Renderer.{h,cpp}</code> , <code>RenderManager.{h,cpp}</code> , <code>SceneRenderer.{h,cpp}</code>
섀도우 시스템	<code>ShadowSystem.{h,cpp}</code> , <code>CSM.{h,cpp}</code>
라이트 관리	<code>LightManager.{h,cpp}</code> , <code>*LightComponent*.{h,cpp}</code>
UI	<code>SViewportWindow.{h,cpp}</code> , <code>ConsoleWidget.cpp</code> , <code>StatsOverlayD2D.{h,cpp}</code>
셰이더	<code>DepthOnly.hlsl</code> , <code>UberLit.hlsl</code> , <code>ShadowBuffers.hlsl</code>

Week08팀 목표:

- 모든 광원(방향/스팟/포인트)의 섀도우 렌더링 통합
- PCF/VSM 필터 모드 전환 및 CSM 지원
- 에디터 내 실시간 리소스 제어와 미리보기 제공
- 통계 및 상태 시스템(ShowFlag, Stat) 완비