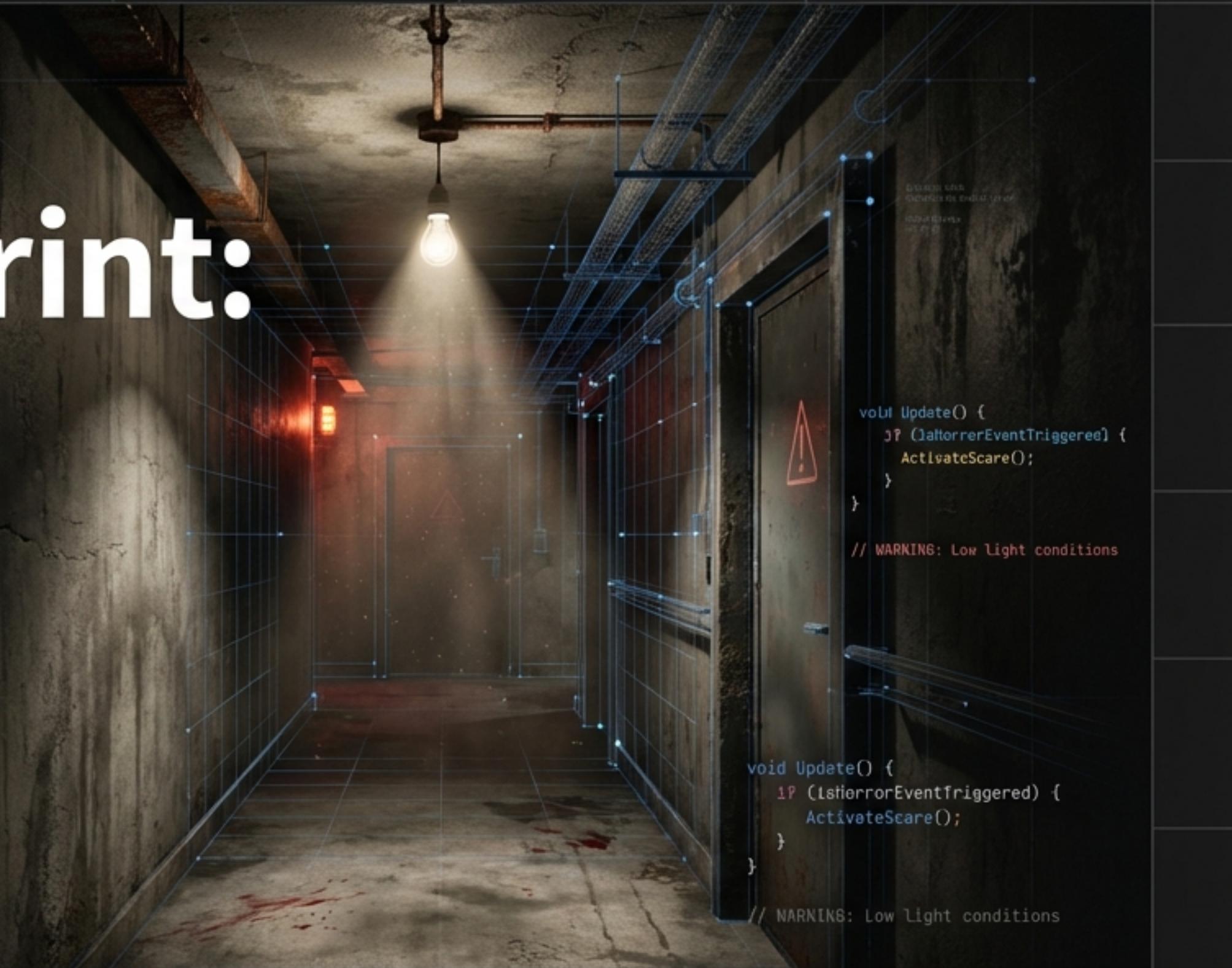


The Room Sprint: 5일 완성 유니티 공포 게임 개발

빈 공간에서 실행 파일까지,
몰입형 3D 공포 경험을 설계하다



커리큘럼 개요 & 개발자 로그

Day 1: Setup

Day 2: Mechanics

Day 3: Atmosphere

Day 4: Polish

Day 5: Build

PROJECT DEVELOPMENT TIMELINE

UNITY 3D HORROR PROTOTYPE PROCESS MAP



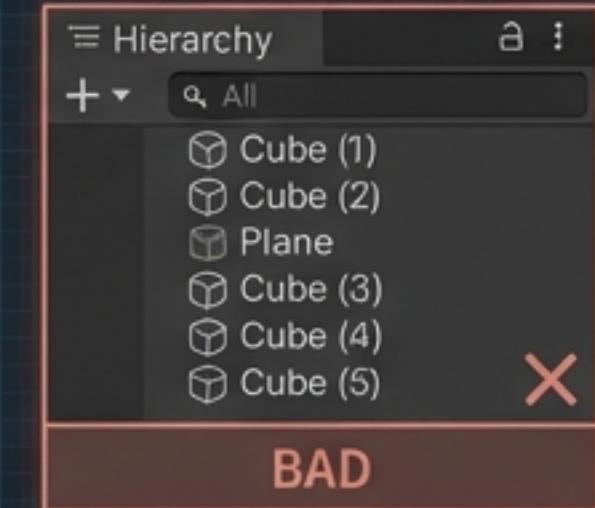
Day 1: 공간의 탄생과 구조화

UNITY 3D HORROR PROTOTYPE PHASE 1

Section 1: The Environment

- **기본 도형 (Primitives):** Plane(바닥), Cube(벽/천장)를 활용한 프로토타이핑.
- **Pro Tip:** Vertex Snap (V키)을 활용하여 틈새 없는 정밀한 벽 배치.

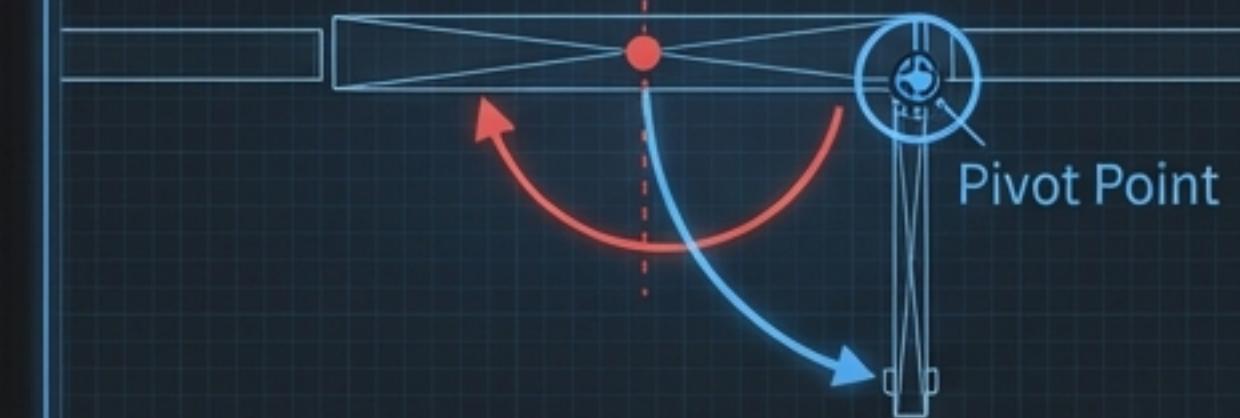
Hierarchy Comparison



Section 2: Hierarchy Management

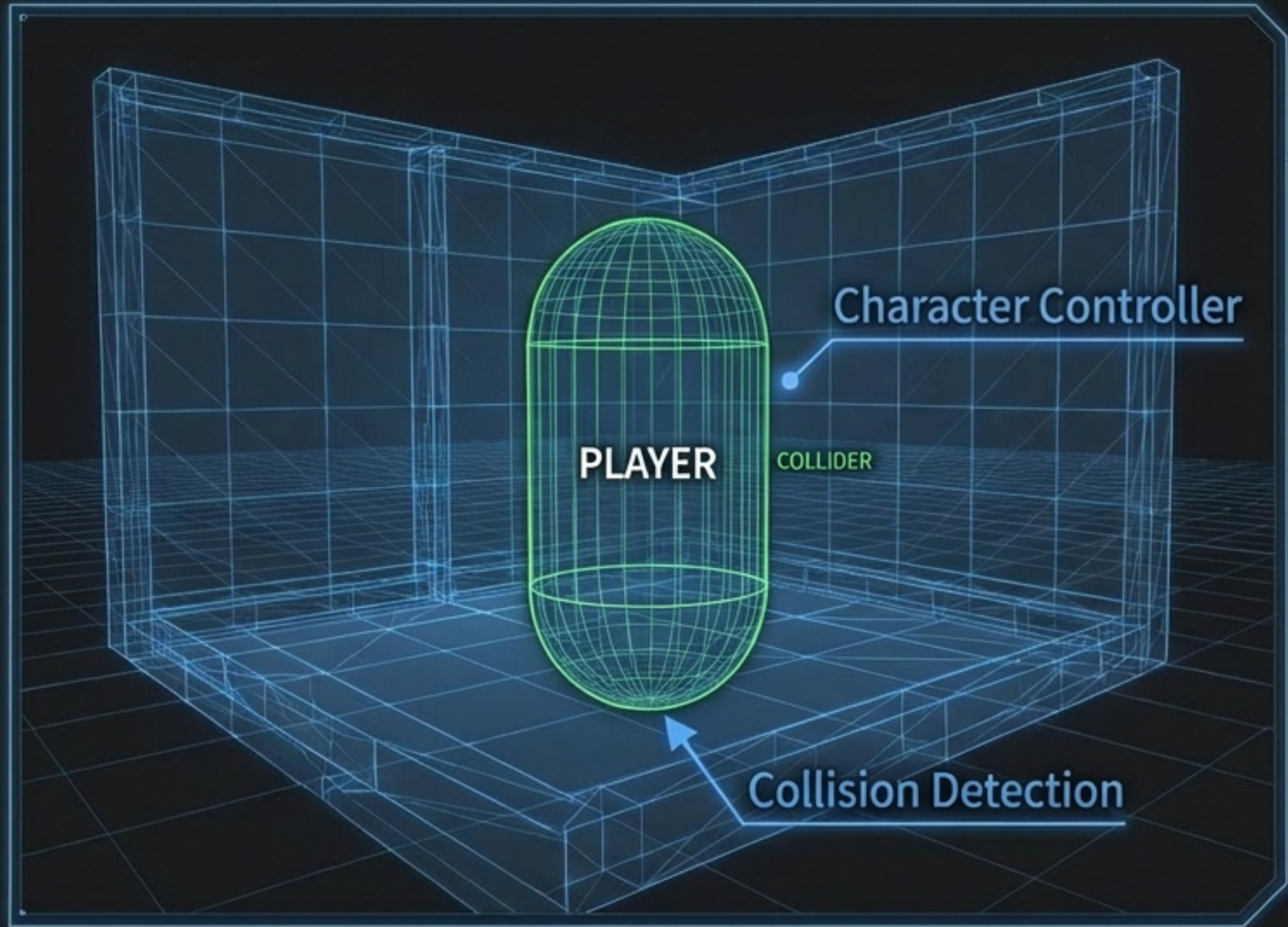
- **그룹화 (Grouping):** Empty Object를 부모로 설정하여 Wall, Floor, Ceiling 관리.
- **피벗 교정 (Pivot Calibration):** 문(Door) 오브젝트의 경첩 축을 위해 Empty Object를 활용, 회전 축 재설정.

Door Pivot Calibration



Day 1: 플레이어 투입 및 물리 법칙

UNITY 3D HORROR PROTOTYPE PHASE 1



Key Concepts

- **Character Controller:** 물리 엔진(Rigidbody)과 별개로 작동하는 플레이어 전용 이동 컴포넌트.
- **Starter Assets:** 1인칭 시점 (FirstPersonController) 프리팹 활용 및 Input System 분석.

Physics Logic

- **Collider:** Mesh Collider와 Box Collider를 통한 충돌 감지 (벽 뚫기 방지).
- **Limit:** 맵 이탈 방지를 위한 경계면 Collider 설정.

Day 2: 암전, 그리고 손전등

UNITY 3D HORROR PROTOTYPE PHASE 1



Step 1: Total Darkness

- **Skybox 제거:** Environment Lighting을 None으로 설정.
- **Ambient Color:** 검은색(#000000)으로 설정하여 완벽한 암전 상태 구현.

Ambient Color #000000 + Spot Light

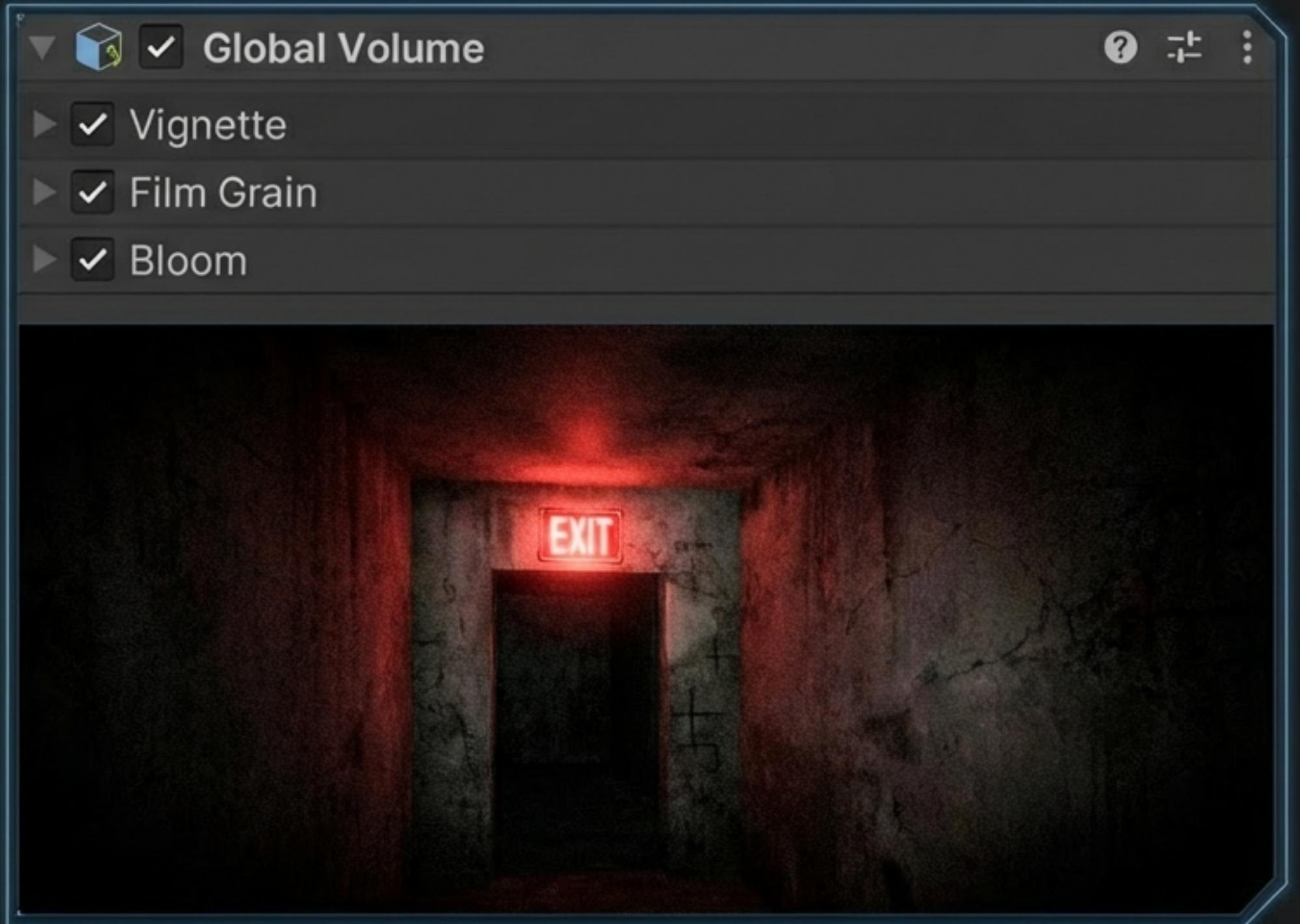


Step 2: The Flashlight

- **Spot Light:** 플레이어 메인 카메라(Main Camera)의 자식(Child)으로 배치하여 시선 추적.
- **설정값(Settings):** Range(거리), Spot Angle(광각), Intensity(강도) 조절.
- **주의 (Note):** 플레이어 자신의 몸체에 빛이 가려지지 않도록 Layer 및 Culling Mask 설정 필요.

Day 2: URP 포스트 프로세싱과 재질

UNITY 3D HORROR PROTOTYPE PHASE 1



Global Volume Effects

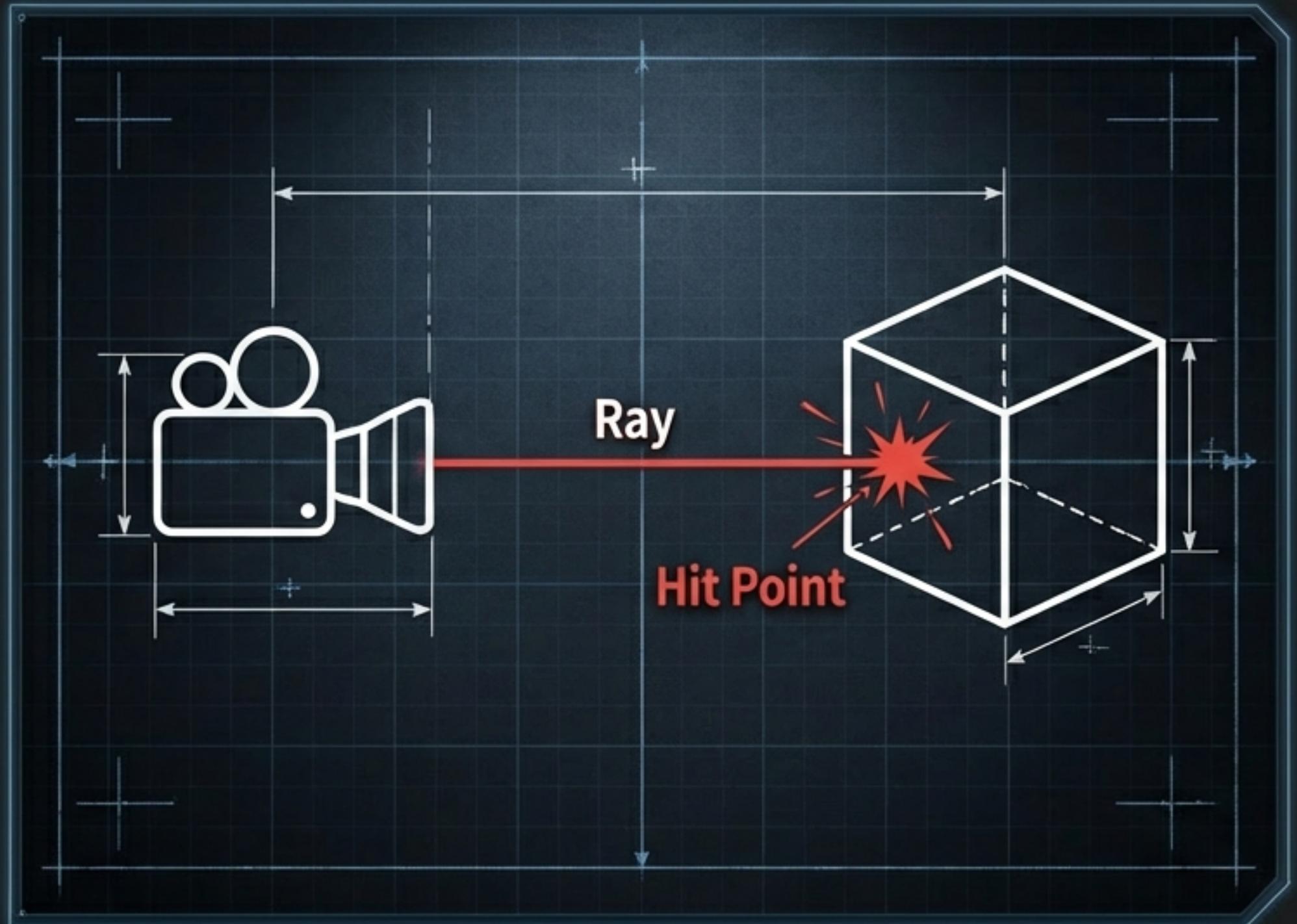
- **Vignette**: 화면 가장자리를 어둡게 하여 시야 제한 및 **공포감** 조성.
- **Film Grain**: 노이즈를 추가하여 거친 영화적 질감 표현.
- **Bloom**: 광원의 빛 번짐 효과 강화.

Material Depth

- **Reflection Probe**: URP에서 금속 재질이 검게 나오는 현상 해결 (Bake 필수).
- **Emission**: 스스로 빛을 내는 물체(모니터, 비상구) 구현으로 시각적 포인트 제공.

Day 3: 보이지 않는 손, Raycast

UNITY 3D HORROR PROTOTYPE PHASE 1



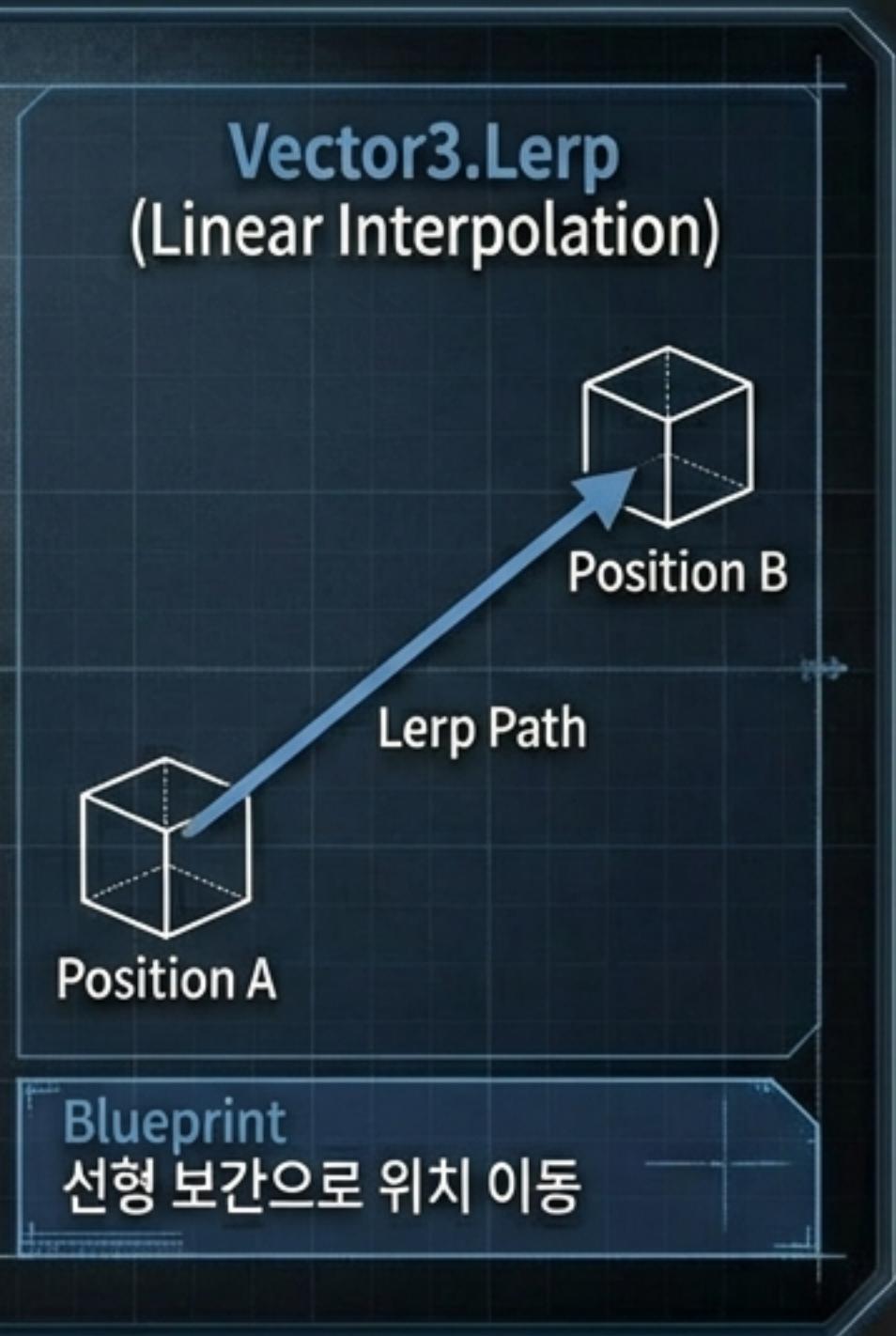
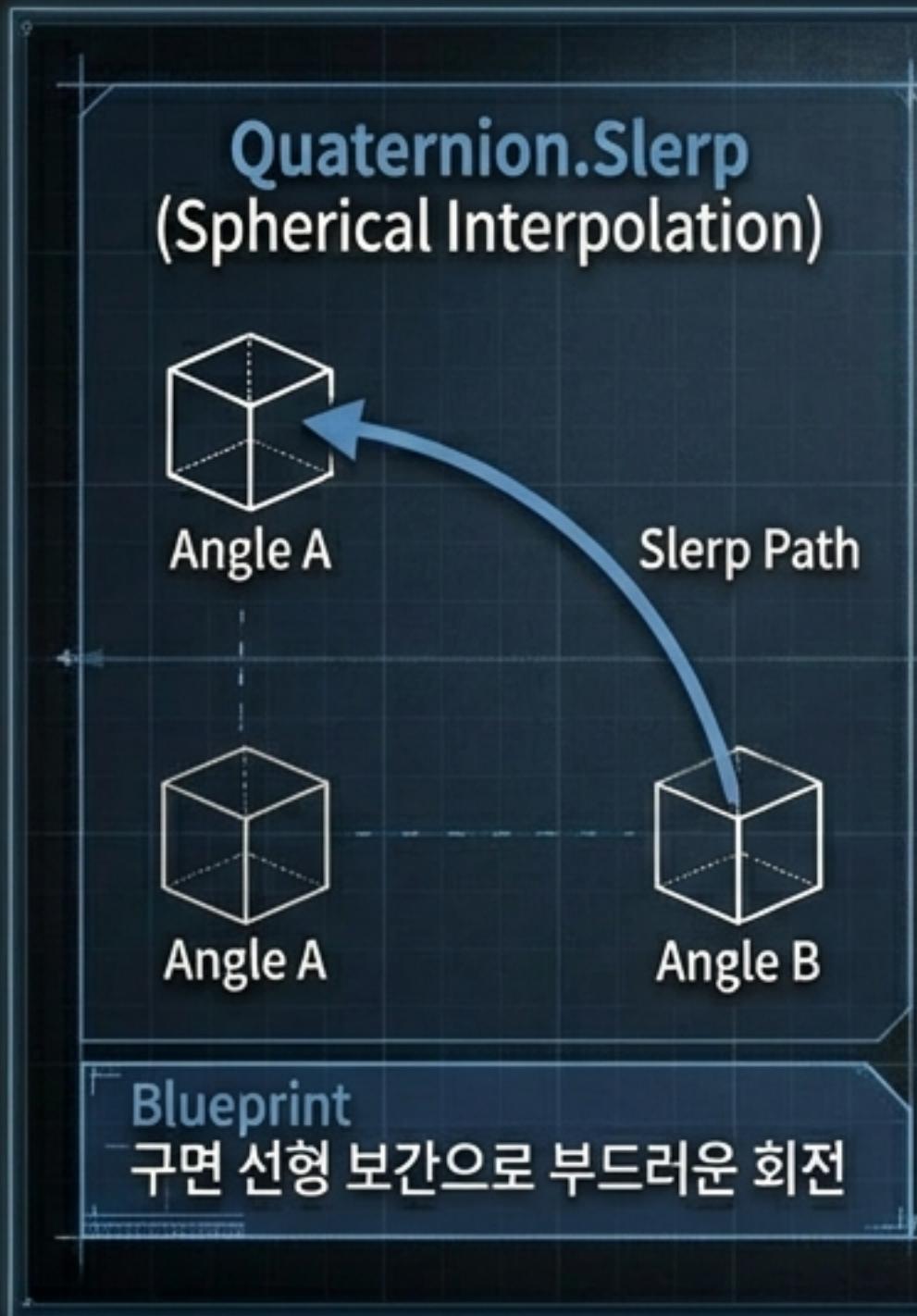
Core Logic

- 개념 (Concept): 카메라 중심에서 전방으로 보이지 않는 레이저(Ray)를 발사하여 오브젝트 감지.
- Code Snippet:

```
Physics.Raycast(ray, out hit,  
distance, layerMask)
```
- Debug: Debug.DrawRay를 사용하여 씬 뷰에서 레이저 시각화.
- Layer Mask: 상호작용 가능한 물체(Interactable)만 선별적으로 감지.

Day 3: 코드로 제어하는 애니메이션

UNITY 3D HORROR PROTOTYPE PHASE 1



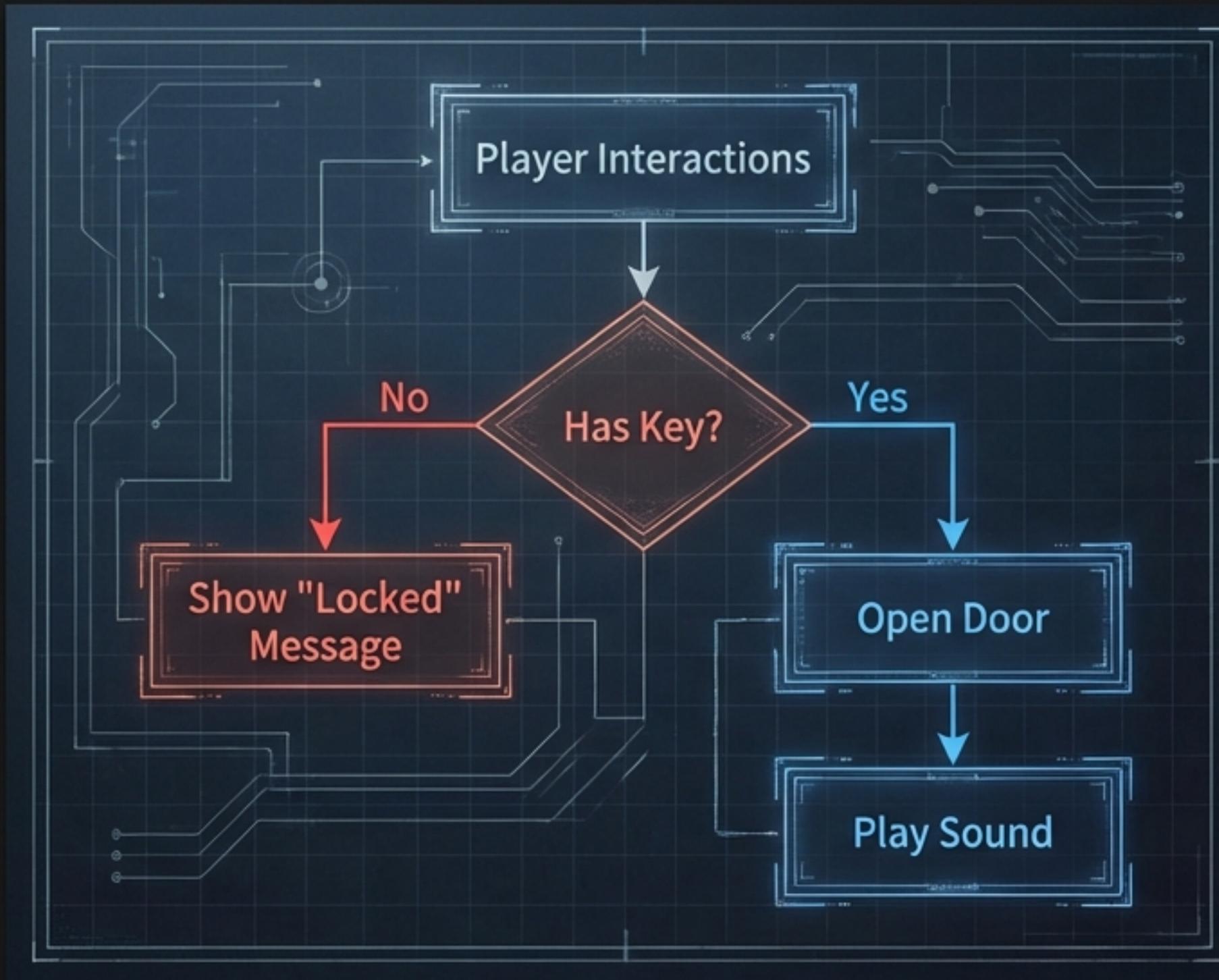
Comparison

- 문 (Door): 회전 운동.
Quaternion.Slerp 사용 (구면 선형 보간으로 부드러운 회전).
- 서랍 (Drawer): 직선 운동.
Vector3.Lerp 사용 (선형 보간으로 위치 이동).

The Math

- 상태 전환(Open/Close)을 위해
Time.deltaTime * smooth 값을 활용한 프레임 독립적 움직임 구현.

Day 3: 심화 로직과 조건문



Spice 1: The Poltergeist

- `Rigidbody.AddForce`를 활용해 물건이 스스로 날아가는 물리적 공포 연출.

Spice 2: Key System

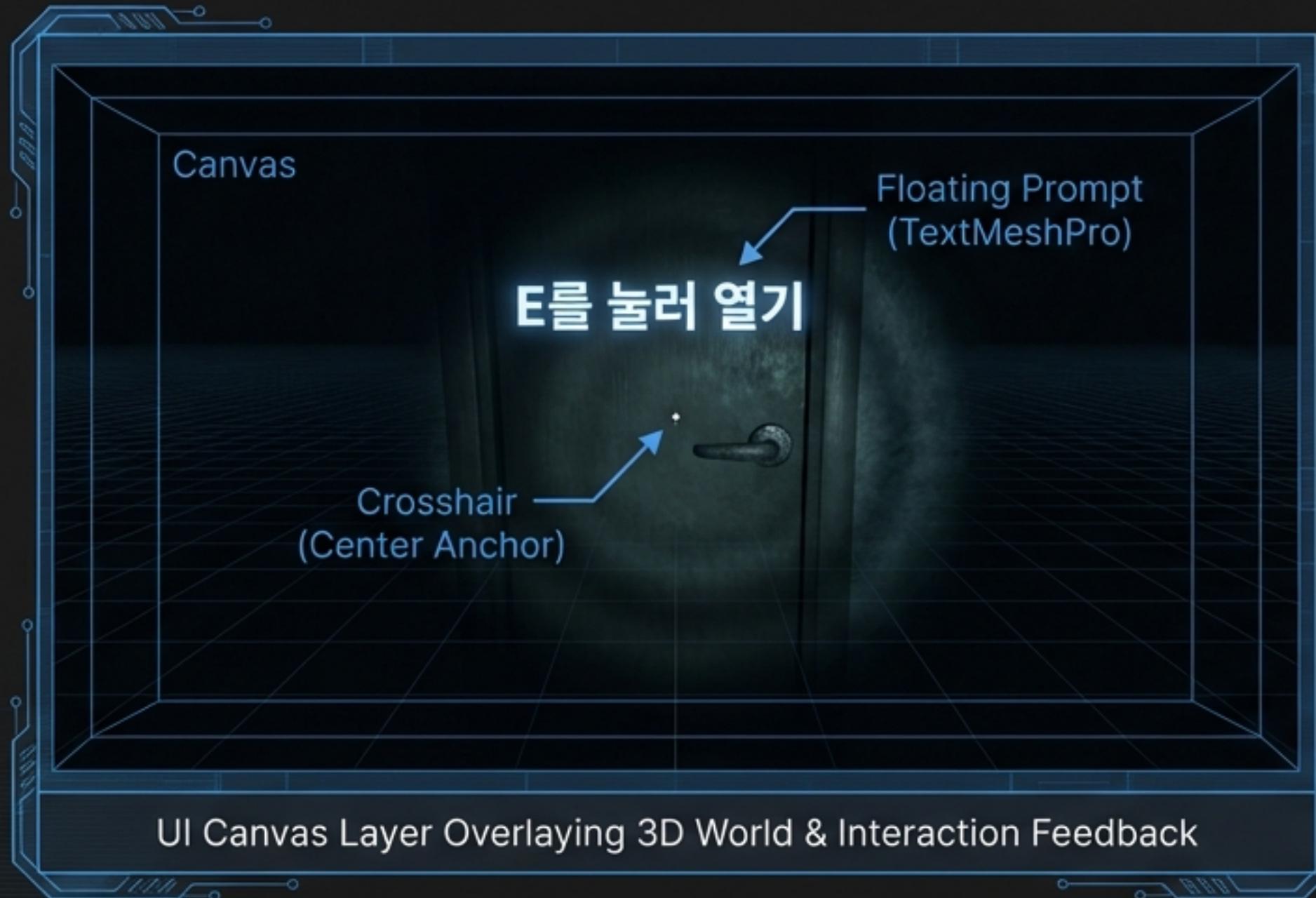
- Boolean Logic: `bool hasKey` 변수를 통한 조건부 상호작용.
- Flow: 열쇠 획득(`Destroy Object`) -> 변수 `True` 전환 -> 문 잠금 해제.

Lighting Logic

- Coroutines: `IEnumerator`와 `WaitForSeconds`를 사용한 전등 깜빡임(Flicker) 비동기 루프 구현.

Day 4: UI 시스템과 텍스트

UNITY 3D HORROR PROTOTYPE PHASE 1



Canvas Setup

- Render Mode: Screen Space - Overlay (화면 최상단 표시).
- Scale Mode: Scale With Screen Size (해상도 대응).

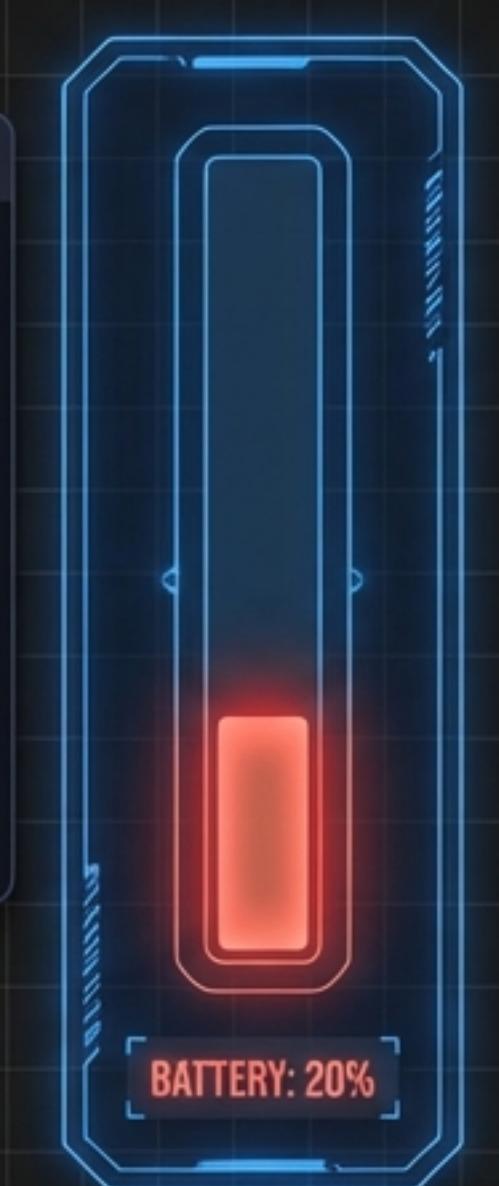
Interaction Feedback

- Crosshair: 화면 중앙 조준점 배치 (Anchor: Center).
- Information: Raycast 감지 시 'E를 눌러 열기' 텍스트 활성화 (text.enabled = true).
- Typography: TextMeshPro 및 한글 폰트 에셋(Font Asset Creator) 활용으로 텍스트 깨짐 방지.

Day 4: 게임 루프와 자원 관리

UNITY 3D HORROR PROTOTYPE PHASE 1

```
•••  
void Update() {  
    battery -= drainRate * Time.deltaTime;  
    if (battery <= 0) light.enabled = false;  
}
```



Game Flow

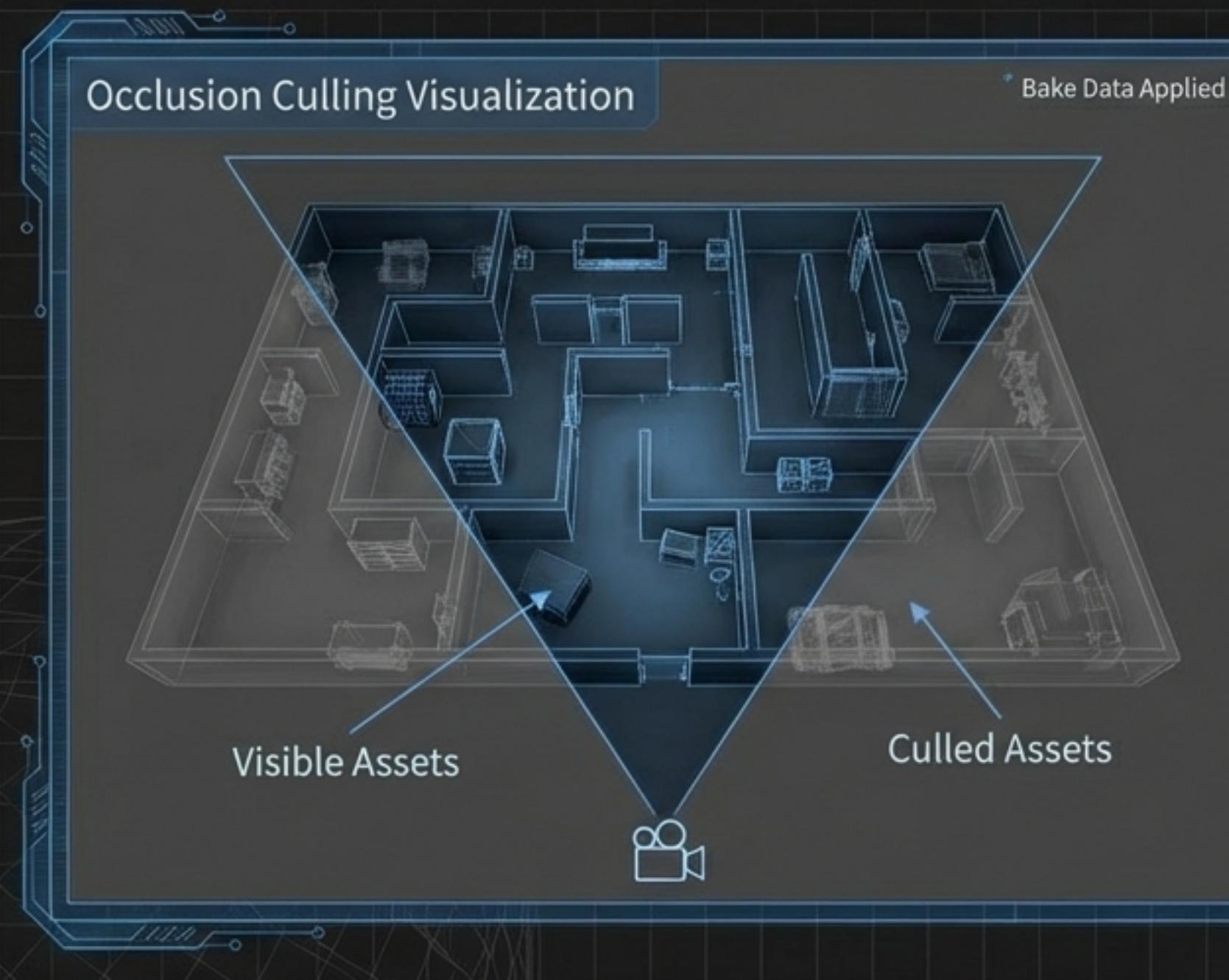
- **Scene Management:** Title \leftrightarrow Game 씬 전환 (LoadScene).
- **Pause Logic:** ESC 키 입력 시 Time.timeScale = 0 (시간 정지) 및 메뉴 활성화.

Spice: Battery System

- **Math:** $battery -= drainRate * Time.deltaTime$ (시간에 따른 지속적 감소).
- **Feedback:** UI Slider 값 연동 및 배터리 소진 시 Light.enabled = false.

Day 5: 최적화, 보이지 않는 기술

UNITY 3D HORROR PROTOTYPE PHASE 1



Rendering Optimization

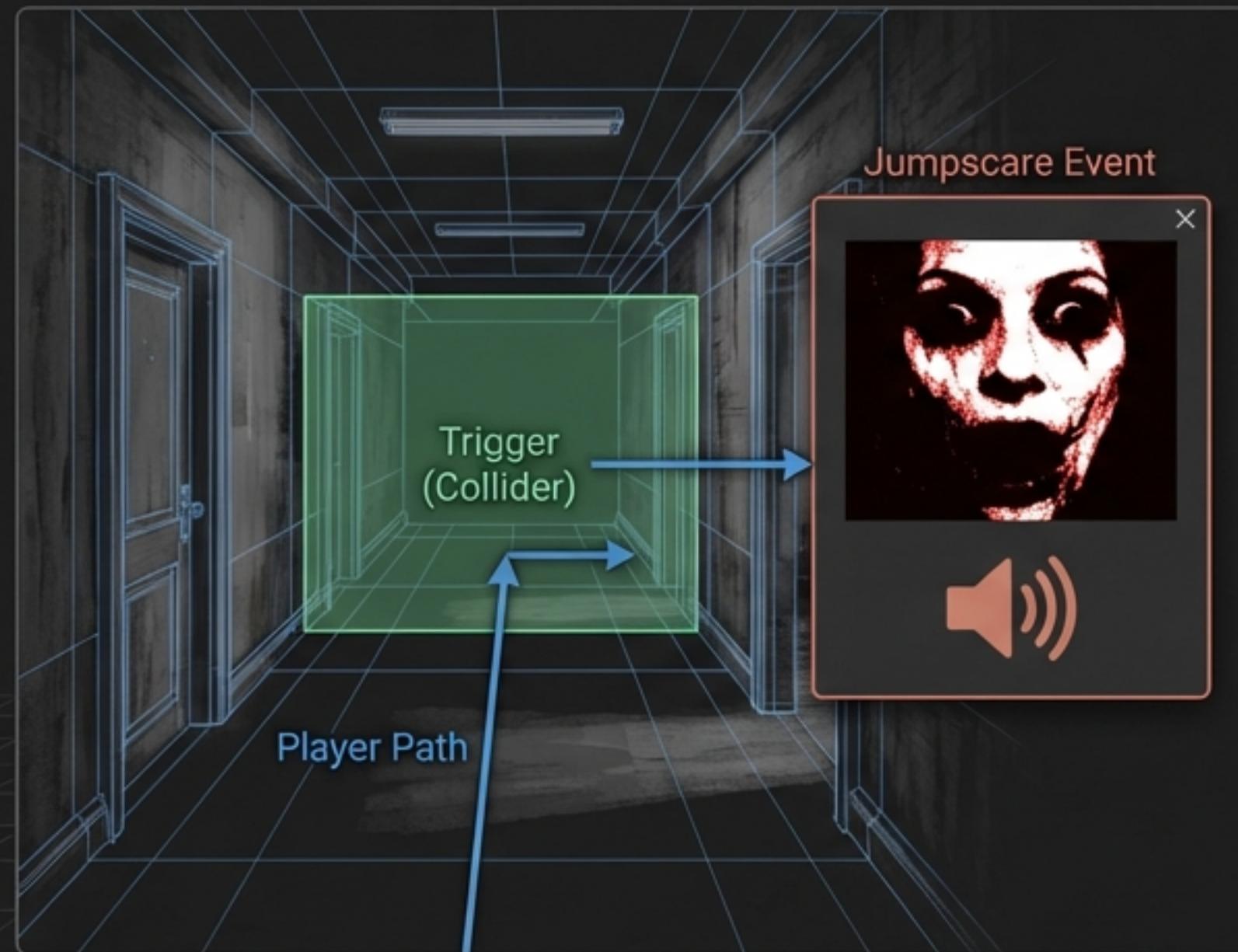
- **Static Batching:** 움직이지 않는 벽, 바닥, 천장 오브젝트를 'Static'으로 설정하여 배치 처리 (**Draw Call** 감소).
- **Occlusion Culling:** 카메라 시야에 가려진 벽 뒤의 오브젝트를 렌더링에서 제외 (**Bake** 과정 필요).

Asset Management

- 텍스처 압축('Compression') 및 'Max Size' 조절을 통한 빌드 용량 최적화.

Day 5: 🌶 공포의 설계, 점프스케어

UNITY 3D HORROR PROTOTYPE PHASE 1



Mechanism

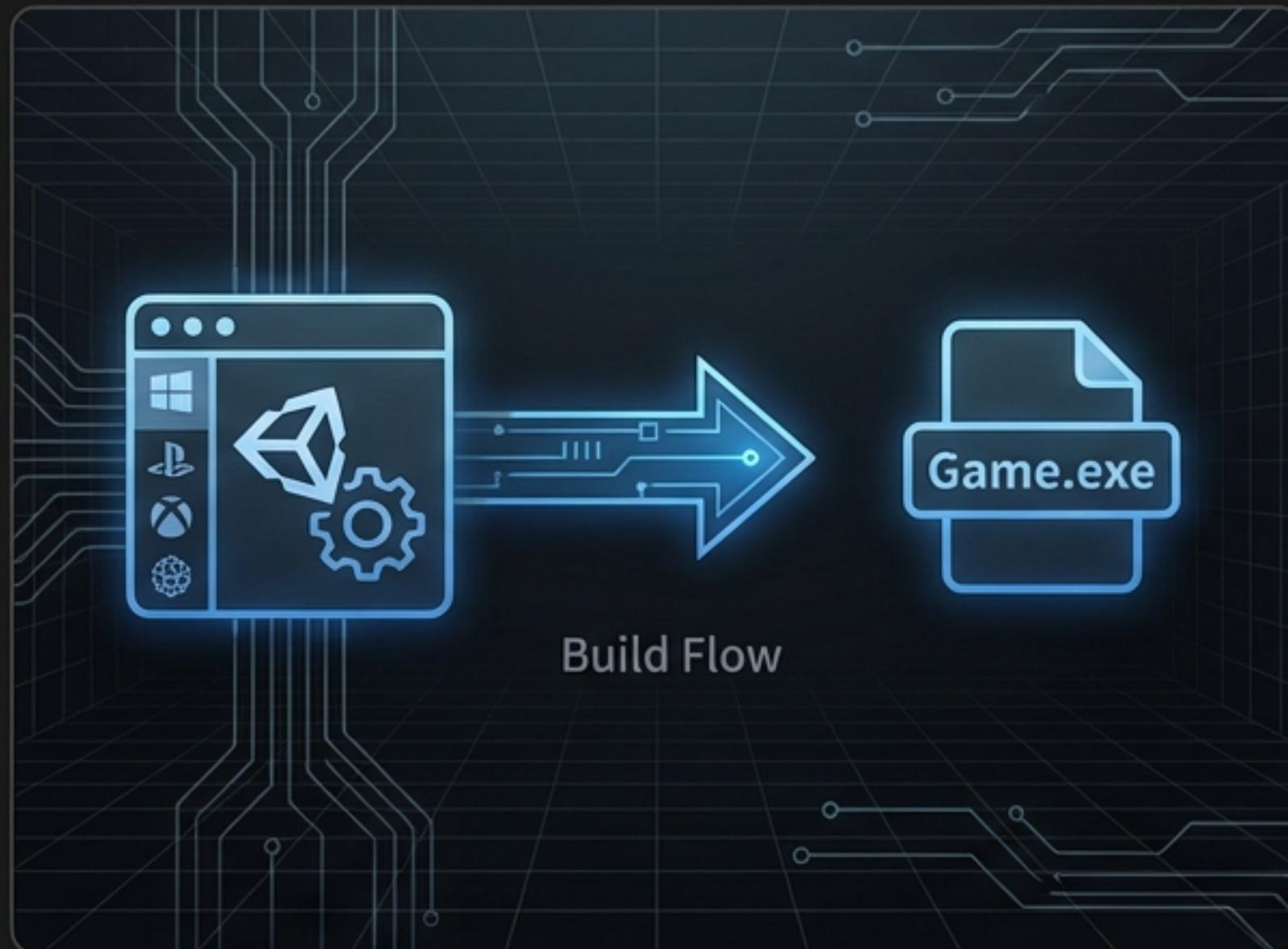
- **Trigger:** 투명한 `Collider` (`Is Trigger` 체크)를 플레이어 이동 경로에 배치.
- **Event:** `OnTriggerEnter` 함수 감지 시 발동.

Execution

- **Visual:** `Instantiate`로 공포 이미지 UI 팝업 생성.
- **Audio:** `PlayClipAtPoint`로 3D 사운드 즉발 재생.
- **Cleanup:** `Destroy(object, time)`로 일정 시간 후 자동 제거.

Day 5: 빌드 및 배포

UNITY 3D HORROR PROTOTYPE PHASE 1



Build Settings

- Scene List: TitleScene과 MainScene을 모두 포함 (**Scenes In Build**).
- Player Settings: 회사명, 제품명, 기본 아이콘(Icon) 설정.
- Resolution: 전체 화면 모드 및 기본 해상도 지정.

Process

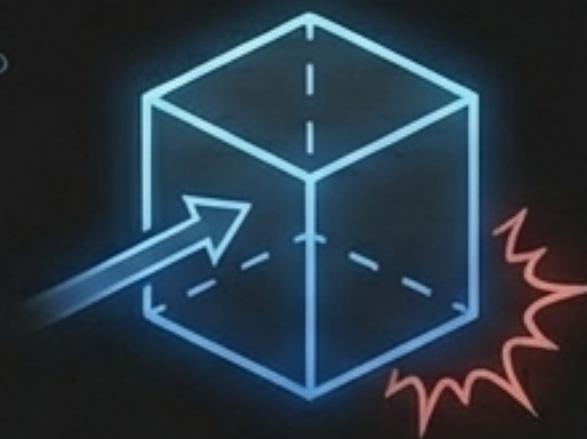
- **Build And Run (.exe 파일 추출)**.
- 일반적인 오류 점검 (씬 누락 확인).

5일간의 기술 스택 요약

UNITY 3D HORROR PROTOTYPE PHASE 1

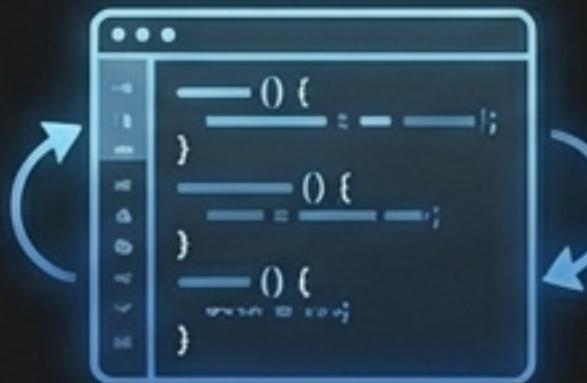
Physics

- Rigidbody
- Collider
- Raycast,
- Trigger



Code

- Start/Update, Coroutines
- Instantiate, Slerp/Lerp
- Singleton Pattern



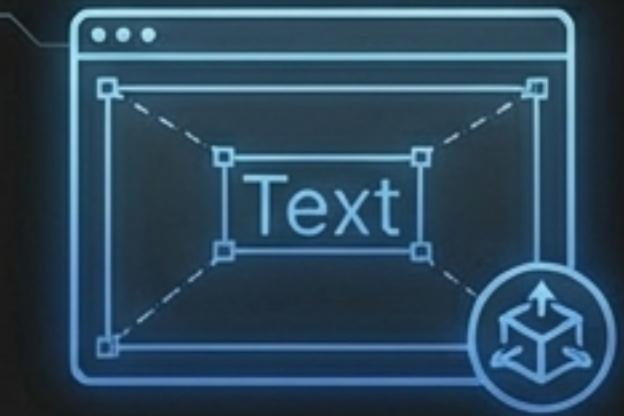
Rendering

- URP
- Global Volume
- Lighting (Spot/Point)
- Occlusion Culling



UI/System

- Canvas, TextMeshPro
- SceneManager
- PlayerPrefs



“이제 당신은 자신만의 공포를 설계할 준비가 되었습니다.”