딥웰(Deepwell) — 세부기획서: 레벨구조

> 재구성 기획서 (역설계) / 작성 기준 시점: 2026-08-31 (프로젝트 시작 시점) / 문서 버전: Ver 0.1 프로젝트 ID: P017 / 근거 수준: A / 작성: Claude (재구성) 이 문서는 결과물과 제작 기록으로부터 기획 의도를 역추정해 당시 작성한 것처럼 재구성한 것이다. 확인된 사실과 추정을 [근거]/[추정] 태그로 구분한다.

1. 목적과 범위

- 목적: 원통형 초고층을 사이드뷰로 번역한 층 씬(링 복도)과 홀 씬(수직 공동), 두 씬을 잇는 열쇠/계단실 규칙을 구현 가능한 수준으로 정한다.
- 범위: 층 씬 레이아웃, 방 규격, 계단실, 홀 씬, 열쇠/진입구, 절차 생성, 도달성 검증, 파밍 배치, 카메라. 적 AI와 아이템 수치는 02/03 인벤토리 문서에서 다룬다.
- 용어: **링 복도**=원통 한 층을 좌우로 펼친 1차원 복도. **계단실 코어**=층 이동 경로. **환승 층**=10, 20, 30, 40F 및 B10~B40처럼 두 홀 구획이 만나는 층. [근거: DEEPWELL_기획서_v0.4.md 부록 A, §5.3]

2. 설계

2.1 층 씬

링 복도 층 구조

- 의도(Why): 원통 평면을 가로 1차원으로 단순화해 2D 플랫포머로 구현 가능하게 하고, 방 단위 탐색/루팅이 본류가 되게 한다.
- 설계(How): 14~18칸의 링 슬롯. 고정 배치는 오른쪽 상/하행 계단실, 왼쪽 진입 홀(로비), 관리실 1개, 홀 진입문(1F/B1 및 5층 단위). 나머지 칸은 테마 풀에서 유닛 프리팹을 채운다. 방 종류와 폭(타일): studio 76, oneroom 98, tworoom 134, luxury 158, admin 112, utility 96, atrium 88, 계단실 코어 116. 오른쪽 끝까지 걸으면 같은 층 왼쪽 끝으로 이어진다. 층 전환 도착은 왼쪽 진입 지점이다. v0.3의 7×4 셀 그리드는 폐기했다. [근거: v0.4 §5.5, §10.6, v0.3 변경 4, 00_공통규격.md §B-9]
- 필요(What): 링 생성기, 유닛 프리팹, 연결부 검증기.
- 작업 방법: 슬롯 초기화 → 고정 배치 → `terrainRng`로 유닛 채움 → 좌우 연결부 정합 검증 → `spawnRng`로 컨테이너, 적, 잠금 롤 → 도달성 검증. 담당은 AI(코드)/도구(검증기).
- 완료 기준: 같은 시드/생성기 버전이면 같은 지형. 관리실, 코어, 진입문 모두 도달 가능.
- 리스크/대안: 세로 폭은 34→72타일로 확대(1,152px)되어 카메라가 세로로도 스크롤한다. 한 층 격자는 1,800×72타일이며 `game.step()` 0.31ms로 측정됐다. [근거: 00_공통규격.md §B-10]
- 근거: [근거: DEEPWELL_기획서_v0.4.md §5.5, 00_공통규격.md §B-9~B-10]

방 내부 — 발판, 사다리, 문

- 의도(Why): 방마다 위아래로 오르내리는 탐색을 만들되, 오를 수 없는 허공 발판은 없애 신뢰를 지킨다.
- 설계(How): 천장 높이 y 범위는 luxury 3~17, tworoom 13~31, atrium 9~21, admin 19~31, oneroom 23~39, utility 27~43, studio 31~45. 발판 형태는 선반 24%, 좌우 분리 26%, 징검다리 30%, 한쪽 발코니 20%. 사다리는 발판보다 2칸 위까지 연결. 방 사이 문 1은 지면 고정, 문 2는 공통 위층에 40% 확률로 두며

위층 문에는 문턱과 접근 사다리를 둔다. 층은 아래에서 위로 쌓으며 아래층 도달 구간과 겹치는 x에 사다리를 세우고, 겹침이 없으면 발판을 밀어 겹치게 한다. [근거: v0.4 §5.5, 00_공통규격.md §B-9, §B-10]

- 필요(What): 발판 생성기, 플러드필 도달성 검증(걷기, 사다리, 낙하, 점프 위 3타일/가로 4타일).
- 작업 방법: 생성 → 발판 단위 도달성 계산 → 14~16개 샘플 층 측정.
- 완료 기준: 깊이 d<15는 도달률 100%, 15≤d<30은 고립 허용 15%(실측 96.5~99.2%), d≥30은 30%(실측 92.3~97.7%). 샘플 14개 층에서 미도달 방 0, 전리품/적 스폰 도달률 100%. [근거: 00_공통규격.md §B-10, §B-13]
- 리스크/대안: 방 단위 검증이 공중 발판을 통과시킨 전례가 있어 검증 단위를 발판으로 낮췄다. 깊은 층의 고립은 제트팩/로프가 존재할 이유로 일부러 남긴다. [근거: 00_공통규격.md §B-10]
- 근거: [근거: 00_공통규격.md §B-9, §B-10]

계단실 — 층 이동

- 의도(Why): 층 이동을 현실의 한 곳(에스컬레이터 뱅크)에 모아 걷는 시간을 줄이고, 층 이동이 홀 열쇠와 별개로 열려 있게 한다.
- 설계(How): 오른쪽 끝 한 곳에 상행/하행을 모은다. 상행은 지면에서 최상단 층계참까지 직선 55단 45°, 하행은 바닥 슬래브를 뚫은 피트 9단(79행)이다. 코어 폭 116타일. 하행 구멍은 한방향 발판으로 덮어 위로는 지나가고 내려갈 때만 S+Space로 통과한다. 층 이동은 입력 E 1.2초 홀드, 로딩 0.5초 이내 목표. 지면에서 1칸(16px)까지 자동 step-up. [근거: 00_공통규격.md §B-13, v0.4 §5.5, §9.4]
- 필요(What): `FloorMap.ts`, `PlayerController.tryStepUp`, 코어 프리팹.
- 작업 방법: 지면→최상단 도달 시험(→ 홀드만, 점프 입력 0).
- 완료 기준: 도달 15.8초에 점프 프레임 0, E 1.2초로 1F→2F, 16개 층 위/아래 입구 도달. [근거: 00_공통규격.md §B-13]
- 리스크/대안: 꺾임 계단(switchback)은 세 번 시도 후 층계참 간섭으로 포기하고 직선 계단으로 정했다. 양끝 코어 각각 사용안은 편도 1,752타일(276초)이라 폐기했다. [근거: 00_공통규격.md §B-13]
- 근거: [근거: 00_공통규격.md §B-12, §B-13]

2.2 홀 씬

홀 슬라이스와 이동

- 의도(Why): 홀은 싸우는 공간이 아니라 건너는 공간이다. 지형만으로 긴장을 만든다.
- 설계(How): 10층 단위 구획(지상 1~10, …, 40~50F, 지하 B1~B10, …). 슬라이스 폭 75타일(16px 기준 1,200px, 화면 960px의 1.25배), 슬라이스 높이는 층간 72타일, 10개를 세로로 연결(140타일 기준 원본에서 확장). 구조물은 붕괴 계단, 발코니, 고정 사다리, 배관/잔해, 리프트 잔해. 이동 적은 없고 위험은 붕괴, 증기, 감전, 추락, 최하부 국소 오염×2. 장비 없이 계단/사다리만으로 완주 가능해야 하며 연속 붕괴 3구간 이상은 금지한다. 그래플링/제트팩은 속도를 벌지만 자동 탈출 수단이 아니다. [근거: v0.4 §5.4, §10.6, constants.ts]
- 필요(What): 홀 슬라이스 풀(30개 제안), `atriumRng` 선택, 홀 검증기(`validate-atrium-slices`).
- 작업 방법: 슬라이스 생성 → 조립(`assemble:atrium`) → 검증 → 타일 생성/검증.
- 완료 기준: "연속 붕괴 3구간 금지" 검증 통과, 최하부 오염 구역 고정.

- 리스크/대안: 홀이 지루한 이동 구간이 될 위험이 최우선. 배경이 곡률/멀리 있는 발코니/점광을 담당하고 충돌은 근경 타일만 담당하도록 노동을 나눴다. [근거: v0.4 §13, 00_공통규격.md §B-3]
- 근거: [근거: v0.4 §5.4, package.json, tests/atrium.test.ts]

홀 함정

- 의도(Why): 소음/부유 적을 지운 뒤 홀의 위협을 고정형 지형으로 일원화한다.
- 설계(How): 코드 상수 — 증기 주기 3.2초에 활성 0.9초, 밀침 260, 붕괴 지연 0.45초, 감전 피해 14, 재피격 간격 0.8초. 밸런스 초안 제안(미적용)은 증기 4초마다 1초, 감전 3초마다 0.6초 8피해로 구현과 다르다. 홀 바닥 오염은 기존 오염 소모×2이며 별도 중복 피해는 없다. [근거: AtriumHazard.ts, 밸런스_초안 §5]
- 필요(What): `AtriumHazard.ts`, 함정 데이터(`traps.json`은 제안).
- 작업 방법: 예고(점멸, 균열, 스파크) → 발생 순으로 구현하고 색/사운드로 예고한다.
- 완료 기준: 기본 경로에 대체 사다리가 있는 붕괴 발판.
- 리스크/대안: 구현값과 초안값 중 정본은 미확정이다.
- 근거: [근거: AtriumHazard.ts, DEEPWELL_밸런스_초안_v0.1.md]

2.3 열쇠와 진입구

홀 열쇠

- 의도(Why): 열쇠를 시간을 줄이는 소지품으로 만들어 "갖고 싶은 것"이 되게 하고, 영구 개통 대신 매 런 재획득의 긴장을 만든다.
- 설계(How): 열쇠 22종(1F, 5F~50F 10종, B1, B5~B50 10종), 소모 없음. 홀 진입/진출 모두 해당 층 열쇠가 필요하며 들어온 문도 예외가 아니다. 지상/지하 홀 사이 연결 통로는 열쇠가 필요 없다. 각 입구 층 관리실에 해당 열쇠를 고정 스폰한다. 열쇠가 없으면 계단실로 직접 층을 오갈 수 있다. 사망 시 일반 소지 열쇠는 상실, 보안 컨테이너 내부는 보존. [근거: v0.4 §5.3, §4.3]
- 필요(What): `keys.json`(22종, 키링 8개 제한), 관리실 프리팹, 열쇠 보유 판정(키링 내부 포함).
- 작업 방법: 홀 경계의 실제 출입문 목록을 기준으로 필요 열쇠를 산정 → `keys.test.ts` 검증.
- 완료 기준: 열쇠 판정 로컬 검사가 키링 내부를 포함한다.
- 리스크/대안: 재파밍 노동화 → 관리실 위치를 알려 두고 가는 길을 위험으로 만든다. 부피 문제 → 키링/보안 컨테이너. [근거: v0.4 §13]
- 근거: [근거: v0.4 §5.3, §9.4 F-DGN-08, tests/keys.test.ts]

2.4 파밍 배치와 카메라

컨테이너 파밍 배치

- 의도(Why): 컨테이너 파밍이 본류, 적 처치는 부수입이다. 노출 시간이 위험이다.
- 설계(How): 서랍 0.8초, 사물함 1.5초, 공구함 2초, 의료함 2초, 서류함 1.2초, 금고 8초, 적 사체 1초. 룸 성향은 studio=잡화/소모품, oneroom=재료, tworoom=장비, luxury=금고 확률 高, admin=열쇠 고정 스폰, utility=기계 부품. 컨테이너/적은 접지 좌표에 배치하고 컨테이너 앞 레이어에 그린다. [근거: v0.4 §5.5~5.6, 00_공통규격.md §B-11, §B-15]
- 필요(What): `loot_tables.json`(구역8×컨테이너6, 제안).

- 작업 방법: 아래 3절 표를 따른다.
- 완료 기준: 도달 가능 구간에서만 전리품/적 생성.
- 리스크/대안: 지면 좌표를 `(H-2)*TILE`로 잘못 계산한 버그가 있었고 B-15에서 정정했다. [근거: 00_공통규격.md §B-15]
- 근거: [근거: v0.4 §5.6, 00_공통규격.md §B-15]

카메라

- 의도(Why): step-up의 16px 점프가 화면 흔들림으로 보이지 않게 한다.
- 설계(How): 세로 데드존 46px, 지수 감쇠 7/초, 가로 감쇠 18/초, 최종 정수 스냅. 프레임률 독립을 위해 `1-e^(-k/dt)`를 쓴다. 홀 카메라는 별도. [근거: 00_공통규격.md §B-14]
- 필요(What): 층 씬 카메라 컨트롤러.
- 작업 방법: 52단 계단 등반 측정.
- 완료 기준: 카메라 1프레임 최대 이동 16px→3px, 4px 이상 튄 프레임 52→0. [근거: 00_공통규격.md §B-14]
- 리스크/대안: 추종 지연 걷기 5px/달리기 8px. 허용했다.
- 근거: [근거: 00_공통규격.md §B-14]

3. 데이터/리소스 명세

항목	필드	범위/예	상태
오염 곡선	d(0~49), 계수	d0~9: 1.00+0.10d … d40~49 끝 11.40	임시 [근거: v0.4 §3.3]
방 폭	종류, 폭(타일)	studio 76 … luxury 158, 코어 116	확정 구현값 [근거: 00_공통규격.md]
컨테이너	종류, 개방초, 가치	서랍 0.8 … 금고 8	기획 [근거: v0.4 §5.6]
층 프리팹	160개(8구역 ×20)	Tiled JSON	제안 [근거: 리소스_명세 §3]
홀 프리팹	30개, 폭 75	높이는 층고 결정 후	제안
열쇠	22종, 키링 8개	1F, 5~50F, B1, B5~B50	v0.4 정본, 현재 데이터는 B1 열쇠 없음 [근거: 리소스_명세 §2]
구역	8개	로비 1~9F … 심층 B40~50	기획 [근거: v0.4 §5.5]

4. 작업 순서

1. 홀 프로토(M1): 슬라이스, 낙하, 시야, 어둠.
2. 링 생성기/프리팹 로더/도달성 검증(M3).
3. 열쇠, 진입문, 관리실 고정 스폰.
4. 계단실, 카메라, 지면 오브젝트 좌표 정정.
5. 깊이별 고립 허용/낙사 기준 조정(R7).

5. 검수 기준

- 같은 시드/버전이면 같은 지형. 장비 없이 홀 완주. 샘플 16개 층 위/아래 입구 도달, 전리품 100%.
- 층 전환 0.5초 이내, 전환 중 타이머 지속.
- 실플레이로 1F 첫 열쇠 획득/복귀 시간은 아직 측정되지 않았다. [근거: v0.4 § 5.5]

딥웰(Deepwell) — 세부기획서: 인벤토리/경제

> 재구성 기획서 (역설계) / 작성 기준 시점: 2026-08-31 (프로젝트 시작 시점) / 문서 버전: Ver 0.1 프로젝트 ID: P017 / 근거 수준: A / 작성: Claude (재구성) 이 문서는 결과물과 제작 기록으로부터 기획 의도를 역추정해 당시 작성한 것처럼 재구성한 것이다. 확인된 사실과 추정을 [근거]/[추정] 태그로 구분한다.

1. 목적과 범위

- 목적: 소지품, 장비, 보안 컨테이너, 창고, 제작, 재보급, 분쇄 규칙과 데이터 필드를 구현 가능하게 정한다.
- 범위: 슬롯/그리드, 보관함 규칙, 장비 내구도, 은신처 시설, 재보급, 분쇄기/프린터, 시체 회수 규칙, 탐사 결산/바닥 물건. 수치 중 [제안값]은 코드 미반영 검토용이다. [근거: DEEPWELL_밸런스_초안_v0.1.md]

2. 설계

슬롯과 그리드 인벤토리

- 의도(Why): 가져갈 것과 버릴 것을 고르는 압박이 곧 파밍의 재미이므로, 테트리스형 그리드와 무게로 욕심의 대가를 만든다.
- 설계(How): 7슬롯(근접, 원거리, 방호복, 특수1, 특수2, 방어구, 가방). 가방 그리드는 회전 없음, 대부분 1×1이며 2칸은 큰 물건, 4칸 이상은 드물게. 가방 등급: G0 3×2(무게 16) … G5 7×4(무게 88). 보조 가방은 서류 케이스 3×1, 손가방 2×2, 사이드 파우치 2×2, 보안 컨테이너 2×2. 최대 스택은 아이템별 개별 지정. 무게는 장비/내부 물품/키링 열쇠를 합산하고 이동/로프 관성에 영향. [근거: v0.4 § 4.1~4.2, 3.2]
- 필요(What): `Inventory.ts`, `Equipment.ts`, 아이템 정의, 드래그&드롭 UI.
- 작업 방법: 그리드 겹침 금지 단위 테스트 → UI 연결 → 직렬화 시 키링 내부 배열까지 독립 복사. [근거: CONTINUATION_2026-09-04.md]
- 완료 기준: 슬롯 타입 제약과 스택 한도가 지켜진다.
- 리스크/대안: 가방 외형 점유 규격(G0/G1 2×2, G2/G3 2×3, G4/G5 3×3)은 제안이며 승인 전 적용 금지. [근거: 리소스_명세 § 2]
- 근거: [근거: v0.4 § 4.2, tests/items.test.ts]

보관함/보안 컨테이너

- 의도(Why): 사망해도 남길 물건을 플레이어가 고르게 하고, 열쇠/키링 부피 문제를 완화한다.

- 설계(How): 보안 컨테이너는 아이템 정의 1종이며 두 특수 슬롯에 각각 장착할 수 있고 각 개체가 독립 2×2 공간을 갖는다. 사망 시 본체/내용물이 원래 슬롯에 그대로 보존된다. 보관함 중첩은 금지하고 키툴(키링)만 예외. 내용물 있는 가방은 직접 교체 불가, 창고에 맡긴 뒤 교체한다. 창고에서는 가방 외형 칸만 점유하고 내부 격자는 보존. [근거: v0.4 §4.2]
- 필요(What): 보관 격자 데이터, 출처 필드, 보존 처리 코드.
- 작업 방법: 장착 개체별 내부 격자 직렬화 → 사망 정산 시 보안 컨테이너만 남기는 회귀 테스트(`run-state.test.ts`).
- 완료 기준: 사망 후 시체 스냅샷에서 보안 컨테이너 내용이 중복 제공되지 않는다.
- 리스크/대안: 중첩 금지와 무게 합산으로 우회 방지. 일괄 이동 시 보관함 이중 이동 UI는 별도 검토. [근거: 밸런스_초안 §6]
- 근거: [근거: v0.4 §4.2, §7.3, tests/run-state.test.ts]

장비 내구도와 탄약

- 의도(Why): 모든 장비에 사용 횟수를 두어 저렴한 비살상 무기의 존재 근거를 만든다.
- 설계(How): 전기충격봉 120, 나이프 90, 대형 둔기 60(사용당 2), 권총 200발, 소총 400발, 테이저 30발, 그래플링 훅 80회(모두 임시). 확인사살은 근접 내구 5. 탄약은 탄종별 스택(최대 30 제안), 일반 탄약과 재보급 탄약은 출처를 위해 별도 스택. [근거: v0.4 §4.4, §7.3, 밸런스_초안 §6]
- 필요(What): 내구도 필드, 출처 필드.
- 작업 방법: 데이터 → UI 표시(잔탄/내구).
- 완료 기준: 내구도 0 사용 불가, 분쇄/재보급 판정에서 예외 처리.
- 리스크/대안: 수치 전부 임시.
- 근거: [근거: v0.4 §4.4]

은신처 시설과 제작

- 의도(Why): 대기시간 없는 즉시 제작/수리로 "다시 하고 싶다"를 유지하고, 시설 등급이 제작 상한과 재보급 등급을 정한다.
- 설계(How): 9개 시설. 발전기가 다른 시설의 등급 상한. 저장고는 레벨당 +40칸(제안), 정수/여과기는 레벨당 시작 생명유지 +10초, 작업 벤치는 수리 비용 −10%/레벨. 수리는 은신처에서 최대 내구도 3% 영구 감소(제안 ×0.97), 현장 수리는 추가 2%p 손실, 하한은 출고값의 20%. 키링은 G0로 저렴히 상시 제작. 업그레이드 비용은 코드 기준으로 목표 레벨 L의 scrap=40×(L+1), alloy=12×L, L≥3이면 cell=4×L였고 제작 비용은 등급 g에서 scrap=10+14g, alloy=5g. v0.4는 scrap 아이템 통합으로 방향을 바꿨으나 코드 반영은 확인되지 않는다. [근거: v0.4 §7.1~7.2, 밸런스_초안 §1, §3, §4]
- 필요(What): `Economy.ts`, `shelter.json`(제안), 제작 레시피 `recipes.json`.
- 작업 방법: 로컬 입력/출력은 원자적으로 저장하고 대기 없이 완료한다.
- 완료 기준: 창고 포화 시 반출, 제작, 장비 교체 물품의 처리가 정해진다(다음 확인 항목).
- 리스크/대안: 활성 레시피 120개는 미정이라 확정 수로 인용하지 않는다. [근거: 리소스_명세 §5]
- 근거: [근거: v0.4 §7, 밸런스_초안, CONTINUATION_2026-09-04.md]

재보급

- 의도(Why): 전부 잃어도 다시 시작할 수 있는 안전망을 쿨타임 없이 제공하되 경제 악용은 막는다.
- 설계(How): 근접, 원거리, 방호복, 방어구, 가방 5분류가 대상(열쇠/특수 장비 제외). 지급 등급은 해당 시설 등급 −2, G0 하한. 해당 분류의 유효 장비가 창고, 장착, 가방, 수령 대기에 없을 때만 지급. 무기는 기본 단검/권총 고정이며 권총과 약 30발 탄약을 한 세트로 지급(공간이 전체 못 들어가면 지급 안 함). 비용/대기 없음. 판매가/분해 산출 0, 출처 표시는 이동, 저장, 시체 스냅샷에도 유지. 첫 시작 때만 G1 5종 전체 장착. [근거: v0.4 § 7.3, 00_공통규격.md § B-8]
- 필요(What): 출처 필드(`source=resupply`), 3D 프린터 신청 UI.
- 작업 방법: 지급 가능 판정 → 창고 공간 판정 → 지급 한 번에 저장.
- 완료 기준: 사망 후 재보급으로 재출격 가능, 재보급품 분쇄 산출 0.
- 리스크/대안: 재보급 반복으로 탄약만 얻는 행위는 권총 없이 탄약만 받는 기능을 없애 막는다. [근거: v0.4 § 7.3]
- 근거: [근거: v0.4 § 7.3, 00_공통규격.md § B-8]

분쇄기/3D 프린터/scrap

- 의도(Why): 창고 정리를 플레이어에게 맡기고(자동 폐기 없음), 공통 재료로 필요한 재료를 직접 살 수 있게 해 파밍의 확정성을 높이되 재료 파밍의 이유는 남긴다.
- 설계(How): 두 시설은 처음부터 제공. scrap은 창고 칸을 차지하는 실물 아이템(1×1, 최대 100, 무게 0.01 — 제안값). 분쇄 산출은 제작 가능 장비의 재료 가격 합×0.20×내구 계수(0.25~1), 재료 재출력은 25% 회수. 제작 재료 30종 가격표(분해 1~32, 프린터 4~128 scrap, 6묶음). 산출 공간 부족 시 전체 거래 취소, 입력 보존. 판매 통화는 미정. [근거: v0.4 § 7.1, 밸런스_초안 § 2]
- 필요(What): `economy.json`(제안), 분쇄 미리보기 UI.
- 작업 방법: 순환 차익(제작→분쇄, 출력→분쇄)에 손실이 있는지 표로 검증.
- 완료 기준: 공간 부족 시 원자적 취소, 재보급품 산출 0 우선.
- 리스크/대안: alloy/cell 처리, 판매 통화, scrap 크기/스택 미결(R1). [근거: v0.4 § 8]
- 근거: [근거: 밸런스_초안 § 1~2]

시체 회수와 탐사 결산

- 의도(Why): 타인의 죽음이 단서/보상이 되되 한 시체에서 무한 파밍은 막는다. 결산은 런이 남긴 성과를 한눈에 보이게 한다.
- 설계(How): 시체 회수는 시체마다 런당 1개, 스택 전체. 결산은 신규 획득, 사용, 이동 거리(타일), 층(최고 지상층/최저 지하층/방문 수), 방호복 피해(피격/낙하와 오염 구분), 처치 종류별 수. 바닥에 버리기는 Tab → 물건 선택 → 바닥에 버리기, 회수는 E. 내구도, 등급, 재보급 출처, 탄창, 키튤 내용/가방 내부를 보존. 탐사 바닥 물건은 해당 런에만, 은신처 바닥 물건은 계정에 남는다. [근거: v0.4 § 7.4, 탐사_결산과_바닥_물건.md]
- 필요(What): `ExpeditionLog.ts`, `GroundItems.ts`, `ExpeditionSummary.ts`.
- 작업 방법: 집계 재방문 중복 방지 → `report-drops.json` 증거.
- 완료 기준: 새로고침해도 결산 창 유지, 업데이트 전 런은 불완전 기록 안내. [근거: 탐사_결산과_바닥_물건.md]
- 리스크/대안: 시체 인플레이션은 밸런스 검증 항목. [근거: v0.4 § 13]
- 근거: [근거: 탐사_결산과_바닥_물건.md, qa/evidence/report-drops.json]

3. 데이터/리소스 명세

필드(items)	타입	범위/예	근거
id, name, category	문자열	분류 13종(melee, ranged, special, valuable, protectiveSuit, armor, bag, subbag, key, keyring, ammo, material, consumable)	[근거: 리소스_명세 § 2]
grade	정수	G0~G5	[근거: v0.4 § 2]
size, stack, weight	정수/실수	1×1~3×3, 개별	[근거: v0.4 § 4.2]
durability / maxDurability	실수	장비별 임시	[근거: v0.4 § 4.4]
source	열거	normal, resupply	[근거: v0.4 § 7.3]
storage grid	격자	가방 6~28칸, 키링 8	[근거: v0.4 § 4.3]
sellValue	정수	재보급품 0	[근거: v0.4 § 7.3]
정의 수	175(현재) → 155(제안)	삭제 −24, 추가 +4	[근거: 리소스_명세 § 2]

시설: 등급 상한 G0~G5, 저장고 칸, 시작 생명유지 보너스. 방어구 경감 6~42%, 내구 40~300. [근거: v0.4 § 4.8, § 7.1]

4. 작업 순서

1. 슬롯, 그리드, 스택, 무게 → 2. 보안 컨테이너/보관함 → 3. 은신처 입출고와 분쇄/프린터 → 4. 재보급과 시체 → 5. 결산/바닥 물건 → 6. 수치 승인 후 데이터 이관.

5. 검수 기준

- 키링 내부 열쇠가 보유 판정에 포함된다. 사망/탈출 정산은 활성 런에 한 번만 적용. 이중 보유(창고/로드아웃)가 없다.
- 창고 부족 시 물품 유실 없음(수동 이동 원칙). 단위 테스트 58개, `npm run verify` 통과. [근거: 탐사_결산과_바닥_물건.md, BUILD_BRIEF.md]

딥웰(Deepwell) — 세부기획서: 조작, UI, 리소스

> 재구성 기획서 (역설계) / 작성 기준 시점: 2026-08-31 (프로젝트 시작 시점) / 문서 버전: Ver 0.1 프로젝트 ID: P017 / 근거 수준: A / 작성: Claude (재구성) 이 문서는 결과물과 제작 기록으로부터 기획 의도를 역추정해 당시 작성한 것처럼 재구성한 것이다. 확인된 사실과 추정을 [근거]/[추정] 태그로 구분한다.

1. 목적과 범위

- 목적: 입력 체계, 화면/UI 구성, 리소스 물량과 생성 파이프라인, 오디오, 검수 기준을 정한다.
- 범위: 조작표, 화면 규격, 장비창, HUD, 결산 화면, 아트 규격, 리소스 파이프라인, 오디오 목록. 수치 밸런스는 02/인벤토리 문서에서 다룬다.

2. 설계

조작 체계

- 의도(Why): 마우스 조준과 키보드 이동의 분리로 로프/사격을 쓰되, 한 키에 한 의미만 두어 장비 유무로 키가 죽지 않게 한다.
- 설계(How): A/D 이동/진자 가속, W/S 사다리/로프 길이, Space 점프/제트팩 홀드/낙하산 탭, S+Space 발판 통과, Shift 달리기(부스터 시 대쉬), 좌클릭 원거리 장비, 우클릭 근접, E 홀드 상호작용(루팅, 확인사살, 홀 문, 탈출, 계단실 1.2초), R 재장전, Tab 인벤토리, M 원형 층 지도, G 손전등, C 웅크리기, Q/F 특수 장비. 입력은 탭/홀드를 구분한다. [근거: v0.4 §3.1, README.md]
- 필요(What): `input.ts`, 입력 설정(`input.json` 제안).
- 작업 방법: 키 충돌 점검 → 실제 입력 시험.
- 완료 기준: 입력 지연 2프레임 이하(기획 수용 기준). 실측은 기록 없음.
- 리스크/대안: Q/F 특수 장비의 개별 활성 규칙은 §8 보류. [근거: v0.4 §3.1]
- 근거: [근거: v0.4 §3.1, §9.1]

화면 규격과 축소

- 의도(Why): 캐릭터가 화면에서 차지하는 비율과 한눈에 보이는 방 수를 맞추고, 작은 창에서도 잘리지 않게 한다.
- 설계(How): 논리 해상도 960×540(48px 캐릭터가 세로 6.3%), 화면 60×34타일. 작은 창은 종횡비를 유지한 비정수 축소로 전체를 보이며 비율이 다른 창은 여백. UI와 포인터 좌표도 같은 배율. [근거: constants.ts, 00_공통규격.md §B-7, v0.4 §11]
- 필요(What): `Viewport.ts`.
- 작업 방법: `small-window.jpg`로 증거 확인.
- 완료 기준: 작은 창 스크린샷에서 화면이 잘리지 않는다(증거 있음). [근거: qa/evidence]
- 리스크/대안: 홀 폭이 화면의 1.25배로 줄어 가로 스크롤이 짧다. 필요하면 슬라이스 폭을 늘린다. [근거: 00_공통규격.md §B-7]
- 근거: [근거: constants.ts, 00_공통규격.md §B-7]

장비창과 HUD

- 의도(Why): 몸에 걸친 위치대로 슬롯을 놓아 학습을 줄이고, 컨테이너를 옆에 붙여 옮길 곳을 보면서 집게 한다.
- 설계(How): 장비창 배치는 특수1, 방호복, 특수2 / 근접, 방어구, 원거리 / 가방. 슬롯 46px, 가방 격자 34px. 컨테이너 열면 인벤토리 오른쪽 열에 붙이고 닫으면 원위치. 닫기 경로 4가지(Tab, Esc, ×, 바깥 클릭).

한글은 word-break:keep-all. 열쇠 20칸 고정 패널은 삭제. HUD 게이지는 생명유지, 방호복, 스태미나 3개. 컨테이너와 개방 게이지는 플레이어 앞 레이어(6~8 위, 12)에 그린다. 아이템 설명문은 고유 약 80개는 개별(BY_ID), 재료/귀중품은 분류 단위(BY_CATEGORY)로 쓴다. [근거: 00_공통규격.md §B-11]

- 필요(What): InventoryUI, LootUI, Hud, Flavor.ts.
- 작업 방법: 리스너/루트가 씬마다 누적되는 누수를 막도록 한 번만 걸고 인스턴스를 교체. [근거: 00_공통규격.md §B-11]
- 완료 기준: 은신처에서도 장비창을 닫을 수 있다.
- 리스크/대안: React DOM 오버레이로 계획했으나 실제 UI 구현 방식은 코드에서 확인하지 못했다. 미확인.
- 근거: [근거: 00_공통규격.md §B-11, src/ui]

결산/메뉴 화면

- 의도(Why): 런의 성과를 숫자로 보이게 해 다음 도전의 동기를 만든다.
- 설계(How): 생환/사망 후 은신처에 결산 창이 열리고, 닫지 않고 새로고침해도 유지된다. '지난 탐사 결산'으로 다시 볼 수 있다. 바닥 물건 UI는 은신처 '바닥 물건' 버튼. 정식 빌드는 로그인 필수 AuthScene, 테스트 빌드는 로그인/서버 생략. [근거: 탐사_결산과_바닥_물건.md, BUILD_BRIEF.md]
- 필요(What): ExpeditionSummary.ts, ShelterScene.ts.
- 작업 방법: expedition-summary.jpg, summary-small.jpg 증거.
- 완료 기준: 작은 창에서도 결산이 읽힌다.
- 리스크/대안: 원형 층 지도, 프로필, 리더보드 화면은 구현 여부 미확인.
- 근거: [근거: 탐사_결산과_바닥_물건.md]

아트 리소스와 파이프라인

- 의도(Why): 작화 전에 임시 리소스로 규격과 코드를 먼저 검증하고, 작화가 오면 코드 변경 없이 갈아 끼운다.
- 설계(How): 원본은 개별 PNG(assets/loose/), 아틀라스는 빌드 산출물(assets/atlas/)이며 손으로 편집하지 않는다. 도형 래스터라이저로 임시 리소스 1,698장을 생성하고(gen-placeholder.mjs), 이어받기 때 생성기가 기존 개별 PNG를 덮어쓰지 않도록 고쳤다. 제안 물량: 플레이어 118프레임, 적 134프레임, 타일 1,200종, 배경 28장, 층 프리팹 160/홀 30, 아이템 작화 135개, VFX 40종. 등급 팔레트는 6색 변주. [근거: README.md, CONTINUATION_2026-09-04.md, 리소스_명세 §3]
- 필요(What): 작화, 도구, 검증기(validate:art-v2, validate:char, validate:atrium-tiles).
- 작업 방법: 발주서(A1 홀타일셋, A2 홀프리팹, A3 플레이어스프라이트, D1/D2 데이터)로 작업 지시.
- 완료 기준: 검증기 통과. 최종 작화는 미완이다. [근거: BUILD_BRIEF.md]
- 리스크/대안: 1인 물량 병목은 8세트 색 램프, 상하체 분리로 줄인다. 폰트 라이선스 확인 필요. [근거: v0.4 §13]
- 근거: [근거: README.md, DEEPWELL_리소스_명세_v0.4.md, package.json]

오디오

- 의도(Why): 화면 밖 소리로 공포를 만든다.

- 설계(How): 제안 151클립. BGM 8, 앰비언스 9, 발소리 5재질×4, 근접/원거리 28, 적 6종×4, 기절/확인사살 6, 로프 5, 홀 전용 7, UI 14, 경고 10, 상호작용 16, 분쇄기/프린터 4. 소음 시스템은 없다. 원본 WAV → OGG/M4A, 총 9MB 목표는 측정 후 확정. [근거: 리소스_명세 §4]
- 필요(What): 음원/변환 도구.
- 작업 방법: 기록 없음. 구현/제작 여부 미확인.
- 완료 기준: 목록 연결(미확인).
- 리스크/대안: 홀 리버브 프리셋은 별도.
- 근거: [근거: DEEPWELL_리소스_명세_v0.4.md §4]

3. 데이터/리소스 명세

자산	수량	규격
타일	1,200(제안)	16×16, 블롭 47종 오토타일 포함
플레이어 프레임	118	48×48
적 프레임	134	32×32~64×32
아이템 아이콘	135	32×32
배경	28	960×540 패널
상호작용 오브젝트	23종/74프레임	컨테이너, 문, 홀 문/탈출 표식 등
VFX	40종	종당 4프레임 기본 예산
오디오	151클립	WAV→OGG/M4A
코어 아틀라스	최대 4장	2048×2048, 측정 전 예산 확정 금지
성능 목표: 초기 로드 6MB/3초, 1080p 60fps, 드로우콜 60 이하, 씬 전환 0.5초, 메모리 400MB. [근거: v0.4 §10.12] 현재 빌드 용량은 `dist-offline` 단일 HTML 기준이며 실측값은 읽지 않았다.		

4. 작업 순서

1. 조작/화면 규격 확정 → 2. 장비창/HUD → 3. 임시 리소스/아틀라스 파이프라인 → 4. 결산/바닥 물건 UI → 5. 작화 발주/교체 → 6. 오디오 제작/연결 → 7. 용량/성능 측정.

5. 검수 기준

- `npm run verify` 한 명령: 단위 테스트, 정식 빌드, 오프라인 테스트 빌드, 네트워크 차단 브라우저 검증이 모두 통과.
- 증거 이미지: shelter, floor, atrium, underground-hall, death, small-window, expedition-summary, dropped-item.
- 한계: 온라인 계정/서버 미검증, 밸런스/아트는 프로토타입 품질. [근거: qa/verification.json, BUILD_BRIEF.md]