하이넘 (High Num) — 메인기획서

> 재구성 기획서 (역설계) / 작성 기준 시점: 2026-08-03 (프로젝트 시작 시점) / 문서 버전: Ver 0.1 프로젝트 ID: P007 / 근거 수준: B / 작성: Claude (재구성) 이 문서는 결과물과 제작 기록으로부터 기획 의도를 역추정해 당시 작성한 것처럼 재구성한 것이다. 확인된 사실과 추정을 [근거]/[추정] 태그로 구분한다.

용어 정의: **홀스터** = 손에 든 주사위를 두는 칸(패). **장전** = 홀스터에서 총에 넣을 주사위를 고르는 행위. **정산** = 장전한 주사위로 데미지를 계산하는 단계. **귀속 다이스** = 항상 1이 나오고 소모되지 않는 주사위. **기믹** = 정예/보스가 하나씩 가진 고유 규칙. **막** = 한 구간(1막 1~11층, 2막 12~22층, 3막 23~34층). 층 번호는 막 안의 `row`(0~12)와 절대 층 `AbsFloor`(1~34) 두 가지다. [근거: README.md]

1. 핵심 메카닉스

1-1. 턴 구조: 장전 → 굴림 → 정산 → 발사 → 반격

- 의도(Why): 플레이어의 결정은 "무엇을 장전하느냐"에 집중시키고, 눈의 운은 한 순간(굴림)에 몰아 긴장을 만든다.
- 설계(How): 전투 한 턴은 ① 세팅(홀스터에서 장전할 주사위 선택, 칩을 세팅에 올림) ② 굴림(`DiceStage.PlayRoll`, 0.85초) ③ 계산 한 줄씩 공개(약 1.9초) ④ 확정값을 총에 물려 발사 ⑤ 적 반격 ⑥ 다음 턴 순이다. 화면을 누르면 공개를 즉시 건너뛴다. 옛 흐름은 발사가 먼저고 계산이 한꺼번에 떠서 "왜 이 값인지" 읽을 수 없었다. [근거: CombatDirector.cs 주석 2026-08-12]
- 필요(What): `CombatDirector`(진행), `CombatState`(전투 상태), `DiceStage`/`CombatHud`(연출), 계산 로그 `SettleLog`.
- 작업 방법: AI가 구현, 사용자/팀이 에디터 플레이로 템포 확인.
- 완료 기준: 한 턴이 효과가 없을 때 약 2.7초에 끝나고, 효과가 붙으면 자연히 길어진다.
- 리스크/대안: 연출 도중 저장하면 반격이 사라지는 판이 생긴다. 턴 시작 시점의 스냅샷(`StableSnapshot`)을 저장해 해결했다. 전투 이어 받기는 항상 "그 턴의 세팅 단계"에서 재개한다.
- 근거: [근거: CombatDirector.cs, README.md]

1-2. 정산식과 다이스 6종

- 의도(Why): 숫자 폭주의 쾌감을 만들되 규칙이 한 줄로 설명되게 한다.
- 설계(How): 최종 공격력 = (다이스 값 합계 + 합연산) × 곱연산 → 기믹 → 최종 데미지 후처리. 다이스는 6종이다. 귀속(ONE, 항상 1, 소모되지 않고 손으로 돌아옴, 패 상한에 불포함), 일반(NML, 눈 그대로), 고통(HRT, 눈이 두 번 적용, 주사위 두 개를 던진 것과 같음), 블록(BLK, 눈만큼 데미지 + 받는 데미지 감소), 치트(CHT, 얻을 때 눈이 1~6 중 하나로 정해져 쓸 때까지 고정, `fixedFace` 6 기본), 베팅(BET, 쓰면 이번 전투 보상 골드 +20). 고통을 "추가 데미지 한 번 더"에서 "눈 2회 적용"으로 바꾸면서 곱연산/족보 배수가 두 몫에 모두 걸리게 했다. 베팅은 옛 "홀드"를 대체하며 코드값 5는 세이브 호환 때문에 유지했다.
- 필요(What): `DiceDef`(id 101000~101005), 효과 표 `EffectTable`(102xxx), 족보 조건 `CondTable`(107xxx).
- 작업 방법: 코드값 숫자는 세이브에 적히므로 값을 바꾸지 않고 이름/능력만 갈아 끼운다.

- 완료 기준: 6종이 덱, 홀스터, 정산, 세이브에서 모두 같은 코드로 동작.
- 리스크/대안: 보스 1의 "특수 다이스 무효"는 종류를 NML로 바꾸고 `origCode`로 기억해 덱 정산이 흐트러지지 않게 한다. 귀속은 건드리지 않는다(홀스터에 눌어붙음).
- 근거: [근거: DiceDef.cs, CombatDirector.cs, Gimmick.cs]

1-3. 홀스터/장전 상한과 성장

- 의도(Why): 압박은 체력이 아니라 "장전 상한"과 "덱 잔량"이 만들게 한다.
- 설계(How): 드로우는 홀스터를 다 채울 때까지다. 매 턴 패가 가득 찬 상태로 시작하고 쏜 만큼만 보충된다. 상한은 홀스터 10~17, 장전 7~14, 드로우 4~7이다. 성장은 정예 처치와 막 전환마다 홀스터/장전 각 +1이며 한 런에서 최대 5번(정예 3 + 막 전환 2) 쌓인다. 상한 유물 2개(「약실 장전」 104116, 「제육감」 104151)까지 담도록 최대치를 시작치 + 5 + 2로 잡았다. 「전탄 장전」 버튼은 비어 있는 칸만 기댓값 높은 순으로 채우고 이미 올려둔 주사위는 건드리지 않는다.
- 필요(What): `BalanceTable.CapHand/CapLoad/CapDraw`, `GrowthCapPlus`.
- 작업 방법: 상한을 바꾸면 최대치도 함께 검토한다.
- 완료 기준: 성장을 다 쌓아도 최대치를 넘지 않고 유물 보너스가 잘리지 않는다.
- 리스크/대안: 「탐욕」 칩은 상한을 깎아 이미 넘친 상태가 될 수 있으며, 그때도 올려둔 주사위는 손대지 않는다.
- 근거: [근거: BalanceTable.cs, CombatDirector.cs]

1-4. 유물/칩

- 의도(Why): 장기 빌드는 유물, 한 판의 보험은 칩으로 나눈다.
- 설계(How): 유물 55종은 수치 가산/곱셈 계열과 족보(풀하우스, 스트레이트, 치또이츠 등) 계열로 나뉜다. 칩은 20종, 최대 5개, 사용 즉시 효과가 적용되는 1회성 소모재다. 수치상승형(Value)은 세팅에 올려두면 정산 때 합/곱연산에 참여하고, 규칙파괴형(Rule)은 `use`(누르는 즉시) 또는 `post`(굴림 직후 눈 손질)로 작동한다. 칩을 누르면 바로 쓰지 않고 [사용/돌아가기/버리기]를 묻는다. 칩 가격 100/150/200, 유물 가격 200/300/400.
- 필요(What): `RelicTable`(104xxx), `ChipTable`(108xxx), `ItemRoller`(잠김, 중복, 예약을 거른 추첨), 아이콘 75장.
- 작업 방법: 시트 갱신(아이템 시트 02, newchips.xlsx) 후 id 겹침 진단.
- 완료 기준: 유물/칩 이름, 설명, 아이콘, 효과가 id로 1:1 연결, 다국어 포함.
- 리스크/대안: 칩 그림 폴더(`Chips/`)와 유물 폴더(`Items/`)가 달라 아이콘이 안 뜬 버그가 있었다(`ItemArt`가 두 폴더를 순서대로 조회하도록 수정).
- 근거: [근거: ChipTable.cs, RelicTable.cs, BalanceTable.cs, HANDOVER.md]

1-5. 기믹

- 의도(Why): 빌드 한 가지로 끝까지 가지 못하게, 정예/보스마다 "빌드에 대한 질문"을 던진다.
- 설계(How): 기믹 10종은 몹의 그림 키(`art`)에 묶여 몹이 하나씩 들고 나오며 층/등급 추첨을 쓰지 않는다. 정예 6: 전염성 광견병(매 턴 적 공격력 +2, 화면 숫자도 함께 오름), 한 푼의 목숨 값(공격마다 골드 -50, 3턴 이내 처치 시 전액 반환), 세 번 보면 죽는 그림(장전 수가 3의 배수가 아니면 피해 -33%), 돼지 목에 진주

목걸이(특수 다이스를 쓴 턴마다 체력 -15, 턴당 1회), 채무불이행(매 턴 유물 5개 비활성), 네 탄생을 죽음으로 축복하리라(칩 미사용 시 피해 -25%, 사용 시 칩 1개당 체력 -10). 보스 3: 도망갈 수 없는 하루(특수 다이스 효과 무효), 가족, 친구, 이웃... 뭐 그런거였을지도.(특수 다이스를 안 쓴 턴 적 공격력 ×1.5), 고름에 뒤덮인 존엄(특수 다이스 개수 × 333,333 만큼 적 실드). 최종 보스 늑대와 개의 시간은 매 턴 보스 기믹을 무작위로 흉내 낸다. 기믹 파괴 칩이 올라가 있으면 기믹이 없는 것으로 친다.

- 필요(What): `Gimmick` 표(109101~109110), 적 상태(`Enemy.shield`, 잠금 목록), 화면 표시(적 체력바 옆 파란 실드 띠, 소지 유물 칸 어둡게).
- 작업 방법: 기믹 변경은 [날짜 요청]으로 이력을 남기고 의도(`intent`) 문장을 코드에 함께 적는다.
- 완료 기준: 각 기믹이 의도 문장 그대로 플레이어에게 읽히고 이어 하기 후에도 같은 기믹이 돌아온다.
- 리스크/대안: 체력을 깎는 기믹은 "맞아 죽을 수 있다"가 위협이라 1로 받쳐 주지 않는다. 정산 뒤 살릴 수단(노잣돈/불사 칩)을 찾는다.
- 근거: [근거: Gimmick.cs, CombatDirector.cs]

1-6. 길 선택과 지도

- 의도(Why): 고른 칸이 다음에 고를 수 있는 칸을 정하게 해 한 번의 선택에 무게를 준다.
- 설계(How): 막의 지도는 노드와 길(간선)의 그래프다. 경로 4개(`PathCount`)를 1층에서 위로 걸어 올리며 한 걸음은 바로 위 층의 같은 열, 왼쪽, 오른쪽 셋 중 하나로 간다. 길끼리 교차하지 않으며 열은 5개다(3에서 상향). 고정층: 1층 전투, 5층 정예, 10층 휴식, 11층 보스(최종 보스 12층은 3막만). 선택은 3D 표지판을 총으로 쏘는 방식이며 글자는 지도 절대 열, 화살표는 목록 순서라 좌표계가 둘인 점을 `NearestLane`과 "서 있는 칸 기준 왼쪽/오른쪽" 표기로 맞춘다.
- 필요(What): `StageMap`, `Signpost`, `RoomType`, 지도 UI.
- 작업 방법: 지도 생성 규칙 변경은 [지도 판만 다시 굽기] 도구로만 반영.
- 완료 기준: 어느 길을 타도 5층에서 정예가 고정되고 지도 화면에서 경로를 눈으로 따라갈 수 있다.
- 리스크/대안: 표지판 콜라이더가 판 간격보다 높아 "왼쪽"을 눌러도 "맨 왼쪽"이 잡히던 버그가 있었다.
- 근거: [근거: StageMap.cs, HANDOVER.md]

1-7. 보상, 상점, 이벤트, 휴식

- 의도(Why): 전투 사이에 재화, 체력, 빌드를 조정하는 휴식 리듬을 만든다.
- 설계(How): 보상 다이스는 먼저 전부 일반으로 나온 뒤 한 개씩 확률로 특수로 갈아 끼운다(일반 전투: 베팅 5/블록 10 → 일반 85, 정예: 베팅 5/블록 10/치트 5/고통 5, 보스: 베팅 5/블록 10/치트 12/고통 8). 보스 격파 시 특수 다이스 5개 확정(1막 블록, 2막 치트, 3막 고통). 회복은 휴식 20, 이벤트 5(깊이 파면 15), 막 클리어 12, 막 전환 10. 이벤트는 보물 45%/함정 25%/회복 30%이며 함정은 눈 4 이상으로 회피, 깊은 회복은 눈 4 이상 성공. 상점은 카테고리당 3칸, 등급 확률 55/30/15, 막 배수 1.0/1.2/1.5. 이벤트에서 눈을 굴리는 판정(BG3식)은 전투 주사위 굴림 애니메이션을 재활용한다.
- 필요(What): `RewardService`(보상, 상점, 이벤트, 보물 화면), `EventTable`(106101~106114), 이벤트 삽화 14장.
- 작업 방법: 시트 값을 `BalanceTable`에 상수로 못 박는다.
- 완료 기준: 모든 칸 종류가 입장→선택→결과→지도 복귀를 끝낸다.
- 리스크/대안: 새 보상 흐름(전리품 → 유물 카드) 도중 저장하는 경우는 미확인(D2).
- 근거: [근거: BalanceTable.cs, claude_531d5fea-ce9d-4786-b845-6bb2e2a53cf0, HANDOVER.md]

2. 구현 방법

2-1. 클라이언트/모듈 구조

- 의도(Why): 규칙(수치), 진행, 연출, UI를 분리해 AI와 사람이 각 부분을 따로 고칠 수 있게 한다.
- 설계(How): `Assets/Scripts/HighNum/` 아래 `RunDirector`(판 흐름, 타이틀로 오는 모든 길의 관문 `ShowTitle`), `RunState`(체력, 골드, 덱, 유물, 칩, 지도, `GainGold/GiveRelic/GiveChip/BreakFerrymanCoin`이 유일한 창구), `StageMap`, `SaveSystem`(세이브 3칸, 유물/칩은 id로 적고 불러올 때 Clone), `Combat/*`(전투 두뇌와 표), `Achieve/*`(업적 20), `View/*`(HUD/전투 화면 4,057줄, 타이틀, 연출), `Audio/*`, `Loc*`, `Editor/*`. UI는 `sortingOrder` 타이틀 30 > 전투 12 > 길 HUD 10. 진입점은 `HighNumBootstrap` → `RunDirector`.
- 필요(What): Unity 6000.3.19f1, 씬 `Stage1.unity` 하나.
- 작업 방법: 새 `.cs`는 `.meta`와 `.csproj` 항목을 함께 만든다.
- 완료 기준: `dotnet build` 두 프로젝트 경고 0/오류 0.
- 리스크/대안: `.csproj`가 브랜치를 옮길 때마다 어긋난다. 유니티에서 Regenerate project files를 쓴다.
- 근거: [근거: README.md, HANDOFF.md]

2-2. 씬이 정본인 UI(이중 구조)

- 의도(Why): 초기에 AI가 UI 전체를 코드로 조립했고 이후 씬에 굽는 방식으로 옮겨 사람이 인스펙터에서 고칠 수 있게 했다.
- 설계(How): 각 `Build()`는 참조가 이미 있으면 조기 반환한다. 그래서 코드를 고쳐도 씬이 우선한다. 규약: 조기 반환 경로에 "그 위젯만" 끼우는 보정을 추가한다(`UseIconGrid`, `EnsureHoverTip`, `SealButtonInners` 등). 프리팹 표준 패턴은 "템플릿+폴백": 템플릿이 있으면 `Instantiate` 후 `Bind()`는 값만 채우고, 없으면 옛 조립 경로를 유지한다. 자세한 UI 설계는 03 UIUX 문서에 둔다.
- 필요(What): 에디터 메뉴 `Tools ▸ High Num ▸` 폰트 ①②③, 프리팹 ①②③, 레이아웃 풀기, 현재 씬 보존 추가 굽기.
- 작업 방법: 변경 전 `grep -n " <필드>:" Stage1.unity`로 값까지 확인한다.
- 완료 기준: 인스펙터 수정이 Play 후 유지된다.
- 리스크/대안: 「전부 굽기」는 손 편집을 날리므로 메뉴에서 내리고 자동 굽기(`[InitializeOnLoad]`)를 껐다(커밋 `604fe06`).
- 근거: [근거: HANDOVER.md, README.md]

2-3. 데이터 파이프라인

- 의도(Why): 기획 시트와 코드가 어긋나는 사고를 막는다.
- 설계(How): 수치는 `BalanceTable`(BOM 있음), 아이템은 id 대역별 표, 글자는 `Loc.T(키, 한국어)` 단일 창구. `LocEn/LocKo` 1,305개는 엑셀에서 뽑아 생성하므로 손으로 고치지 않고, 덮어쓰기는 `LocExtra`(36개)에서 한다. 언어 변경 시 `TableReset`(표 먼저) → `LocSweep`(그려진 글자) → `Changed`(화면) 순으로 갱신한다. 처음 켜면 English이고 선택 후엔 `PlayerPrefs`가 이긴다(`ENLaunch`만). 세이브:

`%USERPROFILE%\AppData\LocalLow\DefaultCompany\High_Num\highnum_slot1~3.json`, `highnum_achieve.json`.
- 필요(What): 시트(MobStats, newMobBal, DiceDrop, newchips, MobGimmicks, ItemChanges), 에디터 진단 6종.
- 작업 방법: 표를 고치면 진단도 고친다(2막 시작방 명단 사례).
- 완료 기준: 진단(유물 중복/id 겹침/이벤트 유물 중복/다국어 빠짐) 통과.
- 리스크/대안: 이벤트 id 겹침은 진단 허점이 있다.
- 근거: [근거: README.md]

2-4. 파일 형식/브랜치 운영

- 의도(Why): 여러 사람/AI가 같은 저장소를 만지는 환경의 사고를 줄인다.
- 설계(How): 모든 `.cs`는 CRLF, BOM은 `BalanceTable.cs`/`RunState.cs`만. `core.autocrlf=true`와 `* text=auto`를 쓰고 줄바꿈은 git에 맡긴다. 기능은 `subTest`/`devmode2` 두 갈래에 넣고 체리픽 후 `git diff --quiet`로 같은지 확인한다. 두 브랜치 차이는 관리자 모드(`AdminMode.cs`, `RunDirector.Admin.cs`, `TitleScreen.Admin.cs`)뿐이어야 한다. 병합 후에는 빌드만 믿지 말고 겹치는 파일을 손으로 읽는다.
- 필요(What): git, 브랜치 규약.
- 작업 방법: 작업 시작 전 `git branch --show-current`, 미커밋 변경이 있으면 브랜치 이동 금지.
- 완료 기준: 두 갈래 차이가 관리자 파일뿐.
- 리스크/대안: 자동 병합이 컴파일되면서 모순되는 사례가 있었다(레이아웃 그룹 끔 vs 전제).
- 근거: [근거: README.md 4장 ⑤⑥]

3. 시스템 간 상호작용과 밸런스 기준

- 핵심 변수: 적 체력/공격력 표, 층별 기준 데미지 `RefDmg`(프로토타입 자동 플레이 1턴 데미지 중앙값, 1층 18, 11층 120, 28~32층 1,801~11,163, 33층 15,000, 34층 6,030으로 기재. 마지막 두 값이 역전된 이유는 기록 없음), 장전/홀스터 상한, 유물 곱연산, 난이도 보정. [근거: BalanceTable.cs]
- 층 구간 체력(시트): 1~4층 14, 5~10층 50, 11~16층 379, 17~21층 2,948, 22~27층 43,452, 28~32층 573,533, 33층 2,834,466, 34층 4,824,628. 현재 실사용은 몹 표(newMobBal)이며 시트값은 참고용으로 남아 있다. [근거: BalanceTable.cs] [추정: 표 변경 이후 `SheetHpBand`가 직접 쓰이는지 확인하지 않음]
- 몹 표(체력/공격력): 1막 첫 칸 21/5, 정예 전 70/5, 정예 690/8, 정예 후 550/6, 보스 4,100/10. 2막 정예 전 5,000/8, 정예 16,800/12, 정예 후 11,900/10, 보스 70,470/15. 3막 정예 전 51,200/10, 정예 505,000/15, 정예 후 379,000/12, 보스 1,004,040/18. 최종 보스 66,666,666/21. 쉬움 체력 ×0.9/공격 -3, 어려움 체력 ×1.3/공격 +3, 최종 보스는 난이도 보정을 받지 않는다. [근거: BalanceTable.cs]
- 시작: 체력 100(50에서 상향), 골드 100(0→50→100), 칩 최대 5개. [근거: BalanceTable.cs]
- 곡선의 의도: 체력이 막마다 약 10배씩 뛰므로 플레이어의 곱연산이 막마다 약 10배 자라야 한다. 이는 "숫자가 커지는 쾌감"을 만드는 핵심 곡선이다. [추정: 몹 표의 막별 정예 체력 690→16,800→505,000 배율로부터 역산]
- 시스템 연결: 난이도(막 시작 선택, 그 막 동안만) → 적 수치/보상 배수 → 정산 → 연출 강도 `ImpactPower(floor, dmg)`(흔들림, FOV, 반동, 화염, 조각 수에 일괄 연결). `MaxTier = 7`, `MaxPower`

2.33으로 후반 포화를 늘렸다. [근거: BalanceTable.cs, 연출_개선안_20260822.md]
- 난이도 보정(보상): 쉬움 다이스 ×1.3, 보통 골드 ×1.1, 어려움 특수 다이스/전설 유물 확률 +5%p. [근거: BalanceTable.cs]
- 불변 조건: 시각 개편 작업은 밸런스/피해 계산, 난이도, 보상량, 저장 데이터를 바꾸지 않는다. [근거: design/GAME_DESIGN.md]

4. 게임의 나머지 부분의 방향

- 콘텐츠 구조: 3막 34층. 1막 마른 황야, 2막 무너진 광산촌, 3막 검은 미사. 막 시작방에서 유물 2택(막별 명단: 1막 `StageOneGiftRelics`, 2막은 족보 계열 5종 명단) 후 난이도 3택. 상세는 03 콘텐츠 문서. [근거: BalanceTable.cs]
- 메타 진행: 업적 20종(걸린 아이템 + goal 진척 값), 도감 해금, 판을 넘어 남는 기록(`AchieveProfile`). 메타 성장(영구 능력)은 기록 없음. [근거: README.md]
- UI/UX 방향: 중앙 보호구역 원칙, 정보는 현재 선택 중심, TAB으로 보조 정보 접기, 한국어 14px 이상. 03 UIUX 문서. [근거: ART_DIRECTION.md]
- 아트/사운드 방향: 종이, 나무, 가죽, 잉크, 세피아, 금빛 확정, 혈색 발사. 03 연출, 리소스, 홍보 문서. [근거: ART_DIRECTION.md]
- 저장/설정: 수동 세이브 3칸, 업적 파일 별도, 화면 설정은 `PlayerPrefs`(세이브와 따로). 해상도 표 10종. [근거: README.md]
- 튜토리얼: 7장(한/영), 정보 창 접기 안내 토스트(전투/길 화면 모두). [근거: README.md, CombatDirector.cs]
- 관리자 모드: `devmode2` 브랜치에서 칸 종류 지정/지도 재생성/이벤트 강제 등. 정식 빌드에 포함되는지는 기록 없음. [근거: README.md]

5. 제작 과정 설계

5-1. 순서와 병렬

- 의도(Why): 규칙이 흔들리는 동안 시각 작업이 낭비되지 않게 한다.
- 설계(How): ① 컨셉/UI/리소스 명세 → ② 전투 코어/표 → ③ 시각 1차 스킨 → ④ 지도, 난이도, 수치 개편 → ⑤ 연출/리소스 완성 → ⑥ UI 하드코딩 분리 → ⑦ 다국어/영어 빌드 → ⑧ PV. ③과 ④는 브랜치(DW 시각 / HS UI / sound / JS)로 병렬 진행하고 HS 베이스에 DW 시각을 이식하는 통합 단계를 둔다(2026-08-20).
- 필요(What): 브랜치 규약, 통합 작업 시간.
- 작업 방법: AI 분담 — Codex는 imagegen 리소스/Unity 에디터 작업, Claude는 문서/코드 구조/검증, Antigravity는 PDF 분석. 사람(사용자/팀)은 방향 승인과 실플레이 확인.
- 완료 기준: 통합 브랜치에서 빌드 오류 0, 콘솔 오류 0.
- 리스크/대안: 병합 시 `.csproj`에서 새 `.cs` 누락으로 오류가 난다. 오류 개수와 훅 개수가 같은지가 단서였다.
- 근거: [근거: .git/logs/HEAD, claude_17e1d6b7-f554-44f1-be24-c0694455a13d, README.md]

5-2. 검증 절차

- 의도(Why): 렌더 결과를 눈으로 확인하지 않고 "적용됨"이라 보고하는 사고를 막는다.
- 설계(How): ① `dotnet build` 경고 0/오류 0 ② 에디터 진단 메뉴 ③ Unity MCP로 실제 Game View 캡처(목업/기존 스크린샷은 참고하지 않고 직접 촬영한 런타임 캡처만 근거로 삼음) ④ 콘솔 오류 0 ⑤ 사용자가 승인 목업과 대조. 리소스는 임포트 타입(Sprite 여부), 해상도, 9슬라이스 경계/NPOT 설정까지 로드 검증(52/52)한다.
- 필요(What): Unity MCP 연결, 캡처 폴더 `Captures/live_20260822/`.
- 작업 방법: 적용 전 예상 목업을 먼저 보여주고 승인 후 적용(2026-08-13 지시).
- 완료 기준: 증거(캡처/검증 json)가 남는다.
- 리스크/대안: MCP 연결이 끊기거나 외부 파일 변경 자동 새로고침을 못 받는 경우 임시 프로젝트에서 배치 컴파일/베이크 후 결과 씬만 되가져왔다.
- 근거: [근거: codex_rollout-2026-08-13T21-34-57-019ffb1e, codex_rollout-2026-08-23T22-19-49-01a02ec7-03c0-7db0-8b20-1f45848a031f, 연출_개선안_20260822.md]

5-3. 인수인계 방식

- 의도(Why): 대화가 바뀌어도 함정에 다시 빠지지 않게 한다.
- 설계(How): README에 소스 구조, 검증법, 함정 8개를, HANDOVER에 작업 원칙 5개, 도구, 남은 일/지뢰 목록을 쓴다. 작업을 하다 틀린 판단은 "해결한 주요 버그" 표(증상|진짜 원인)에 남긴다.
- 필요(What): 문서 2종 이상, 커밋 전 체크리스트(새 스크립트, 프리팹, `.meta`, 리소스, 씬).
- 작업 방법: 다른 컴퓨터로 옮기기 전 `git add -A` → 목록 눈 확인 → 커밋/푸시.
- 완료 기준: 새 컴퓨터에서 클론 → 브랜치 → 폰트 ① 진단 "이미 정리되어 있다" → 길 선택~전투 한 바퀴.
- 리스크/대안: 추적되지 않는 문서(루트 README)가 브랜치 이동 중 사라진 사례가 있어 문서는 커밋한다.
- 근거: [근거: HANDOVER.md, README.md]

6. 세부기획서 목록

문서	정해야 할 것
`03_세부기획서_콘텐츠-레벨-적.md`	막/층 구조, 지도 생성, 칸 종류, 몹, 정예, 보스, 최종 보스, 기믹 상세, 시작방, 보상, 상점, 이벤트
`03_세부기획서_전투-성장-수치.md`	정산식 예시, 다이스, 유물, 칩 표 명세, 상한/성장, 데이터 테이블 필드, 밸런스 기준과 진단
`03_세부기획서_UIUX-씬구조.md`	화면별 배치, 이중 구조 규약, 프리팹 템플릿+폴백, 레이아웃 원칙, 결산 창
`03_세부기획서_연출-리소스-홍보.md`	연출 강도 등급, 격발 타이밍, 리소스 명세 Part A~C, 52장 반입, PV, 영상, 영어판 (P008 통합)