ARK ON THE GROUND — 세부기획서 (레벨/서버 판정, 데이터, UI)

> 재구성 기획서 (역설계) / 작성 기준 시점: 2026-09-17 (프로젝트 시작 시점) / 문서 버전: Ver 0.1 프로젝트 ID: P044 / 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.

1. 목적과 범위

- 목적: 구현 팀이 바로 착수할 수 있도록 미니게임의 맵 레벨, 서버 판정/데이터, 화면, 아트/사운드 리소스를 한 문서에 정리한다. [추정: L 규모 기준 문서 1개]
- 범위: 맵 3종의 구간 설계, 서버 처리 순서와 예외, 19개 테이블 명세, 화면 S00~S07, 리소스 목록. 보상 총량 계산과 지표는 02/12 문서를 참조한다. [근거: 05_데이터_테이블.md, 08_화면_설계.md]
- 제외: 구현 코드, 실제 맵 좌표(기록 없음), 몬스터/성장 시스템(해당 없음, 구현 없는 이벤트 미니게임).

2. 설계

레벨: 4구간 허들 구성

- 의도(Why): 한 판 안에서 학습, 타이밍 학습, 숙련을 겪게 하고, 초회 이탈을 줄인다.
- 설계(How): 구간은 다음 연결 노드 번호로 전환한다.

구간	노드	감지 영역	요구 행동
A 학습	1~3	없음, 분기 없는 경로	이동/연결 조작 학습
B 고정	4~6	PULSE 1개, 주기적 점멸	주기 확인 후 통과
C 순찰	7~8	PATROL 1개, 좌우	이동 타이밍 조절
D 교차	9~10, 코어	CROSS 2개	안전 경로 선택

맵 3종은 방향 회전/반전만 달라 구간별 장애물 수/이동 거리 범위/점수 상한이 같다. 서버가 균등 확률로 배정한다. 이유: 패턴 간 유불리를 없애 순위를 공정하게 한다. - 필요(What): 맵 3종 좌표, 감지 영역 반경/속도, 레벨 제작 도구(미정). - 작업 방법: 기획이 구간 표/상한 확정 → 레벨 담당이 좌표 작성 → 테스트로 구간별 체류, 피격, 도달율 측정 → 조정. - 완료 기준: 첫 3노드 이탈이 낮고, 7번 이후 코어 완성율이 급락하지 않는다(수치 목표는 기록 없음, 점수대 20/40/30/10%가 난도 목표). - 리스크/대안: 코어 완성율이 낮으면 C/D 구간의 속도/반경을 먼저 조정, 보상 조건은 유지. - 근거: [근거: 02_보상_설계.md, 03_규칙_설계.md, 13_게임_콘셉트와_설계_의도.md]

서버 판정: 입장 → 플레이 → 결과 처리

- 의도(Why): 이벤트 기간, 티켓, 점수, 보상 전부를 서버가 확정해 부정/중복을 차단한다.
- 설계(How): ① 이벤트 상태 요청 → 서버가 UTC로 기간/일일 키 판정, 티켓 지급, 티켓, 최고점, 참여 횟수 반환. ② 입장 요청(요청 ID) → 기간, 티켓, 진행 중 세션 검사, 차감, 세션, 패턴 원자 생성. ③ 플레이 중 입력 순번 검증, 유효 입력 로그, 스냅샷 전송. ④ 코어 도달/180초 경과 시 COMPLETED, 끊김 30초 초과는

마지막 유효 상태로 자동 종료. ⑤ 결과 제출(요청 ID/세션 ID) → 점수 재계산, 엄격 초과일 때만 최고점 갱신, 유효 완료 횟수/일일 첫 완료 갱신, 보상 원장 생성, 우편 큐 투입, 재전송은 기존 결과 반환. ⑥ 종료 5분 유예 후 랭킹 고정, 순위 원장, 우편 발송.

- 필요(What): 서버 객체 5종, DB 19개, 기존 우편 서비스 호출 인터페이스.
- 작업 방법: 시퀀스(`07_시퀀스_다이어그램.svg`, 입장, 결과, 정산) 기준 구현 → 예외 표 검증.
- 완료 기준: 예외 표의 모든 항목이 재현 테스트로 통과한다.
- 리스크/대안: 입력 로그량(1입력 1행)이 커질 수 있다. 입력 샘플링/배치 저장은 대안이나 검증 가능성을 줄인다 [추정: 로그가 점수 재구성 근거이기 때문].
- 근거: [근거: 03_규칙_설계.md, 04_이벤트_시스템_설계.md, 10_시스템_흐름.md]

예외	처리	근거
동시 입장/중복 요청	요청 ID UQ, 최초 응답 재전송, ACTIVE 세션 1개	05
세션 생성 실패	같은 트랜잭션 롤백, 티켓 미차감	03
티켓 상한 초과	초과분 소멸, `discarded_count` 기록	05
접속 중단	30초 복귀, 이후 마지막 유효 상태로 종료	03
종료 경계	종료 전 승인 세션만 완료 가능, 종료 후 5분 제출, 이후 EXPIRED 0점	03
결과 중복 제출	`submit_request_id` UQ, 기존 결과 반환	05
우편 실패/가득 참	원장 유지, `reward_delivery_log` 재시도	02, 05

UI/UX: 화면 흐름과 상태 표시

- 의도(Why): 서버 확정 전에 잘못된 결과를 보이지 않고, 지연/오류에서도 이용자가 상태를 이해하게 한다.
- 설계(How): S00 로비 → S01 이벤트 허브 → S02 미니게임 메인(티켓 n/4, 다음 갱신, 최고점, 순위, 참여 횟수/56, 보상 목록; 기간 밖/티켓 0/세션 진행 중이면 입장 버튼 비활성) → S03 규칙 팝업 → S04 입장 로딩(승인 후) → S05 인게임(리소스 로드/세션 동기화 후; 드론/연결 노드, 코어, 감지 영역/남은 시간/연결 수/피격 수/예상 점수; 서버 불일치 시 서버 값으로 정정) → S06 결과(총점/4개 세부 점수, 신기록, 보상 상태, 제출 중 중복 탭 잠금) → S07 랭킹/보상(본인 순위, 구간 기준, 상위 기록, 정산 중 표시, 타인 계정 ID 비노출) → 기존 우편함. 종료 시각은 `5월 21일 04:59:59 KST`까지로 표기한다. 접근성: 위험 영역은 테두리, 아이콘, 경고음 병행.
- 필요(What): 화면 와이어프레임, 서버 응답 코드별 문구, 정적 규칙 캐시와 설정 버전 검증.
- 작업 방법: 화면 ID별 와이어프레임 → 상태(로딩, 실패, 지연) 문구 → 클라이언트 구현 → 서버 응답 목업 테스트.
- 완료 기준: 입장 실패/결과 조회 실패가 현재 화면에 사유와 재시도를 보여 준다.
- 리스크/대안: 실제 니케 UI와 다를 수 있어 제안안으로만 사용.
- 근거: [근거: 08_화면_설계.md, 00_니케_구조_확인.md]

수치/데이터: 보상 테이블과 가정 시나리오

- 의도(Why): 보상을 데이터로 분리해 운영 중 경제 검수와 교체가 가능하도록 한다.
- 설계(How): 보상 묶음과 구성품을 테이블로 분리하고 `R_*` 묶음을 조건 행(`reward_condition`)에 연결한다. 수량은 [제안값]이며 실측(M01~M07)으로 교체한다. 보상 ID와 아이템 매핑(`I_GROWTH_BOX` 등)

은 기획용 식별자다.
- 필요(What): 02 문서의 보상 표, 엑셀 허들_모델, 보상_총량.
- 작업 방법: 입력 교체 → 총량 재계산 → 경제 검수 → 승인.
- 완료 기준: 실측 기반 총량이 허용 범위 내.
- 리스크/대안: 총량 초과 시 출시 전 수량/구간 재조정.
- 근거: [근거: 02_보상_설계.md, 12_지표_조사_및_산정_근거.md]

아트/사운드 리소스

- 의도(Why): 복구의 인과와 위험을 시청각으로 전달한다.
- 설계(How): 드론, 노드(꺼짐/활성/연결됨), 코어, 선로 점등, 어두운→밝은 구획, 랩처 감지 영역(테두리/아이콘), 피격 정지 연출, 경고음, 연결 승인음, 복구 완료 연출. 구체 수량/규격은 기록 없음.
- 필요(What): 위 목록의 2D 리소스/음원, 생성 이미지는 콘셉트 참고용으로만 사용.
- 작업 방법: 리소스 목록 확정 → 제작 → 화면 통합 검수.
- 완료 기준: 색을 구별하지 못해도 위험 시점을 알 수 있다.
- 리스크/대안: 공식 아트와의 이질감. 이벤트 전용 스타일 가이드를 먼저 받는다.
- 근거: [근거: 13_게임_콘셉트와_설계_의도.md, 08_화면_설계.md] 상세 목록은 [추정: 연출 요소 문장에서 역산].

3. 데이터/리소스 명세

정적 테이블(서버 원본, 클라이언트 읽기 캐시)

테이블	주요 필드	제약/값
`event_config`	`event_id` PK, `start_at_utc`, `end_at_utc`, `reset_hour_kst`, `daily_ticket_grant`, `ticket_cap`, `entry_cost`, `session_limit_sec`, `reconnect_grace_sec`, `submit_grace_sec`	값 2, 4, 1, 180, 30, 300
`map_pattern`	`pattern_id` PK, `event_id` FK, `node_count`, `max_stun_count`	3개, 노드 10, 정지 상한 3
`map_node`	`node_id` PK, `pattern_id` FK, `node_order`, `pos_x`, `pos_y`, `core_flag`	UQ(pattern_id, node_order)
`map_hazard`	`hazard_id` PK, `pattern_id` FK, `start_node_order`, `end_node_order`, `pos_x`, `pos_y`, `radius`, `motion_code`, `speed`, `stun_sec`	패턴당 4행, `stun_sec=5`, `PULSE/PATROL/CROSS`
`score_rule`	`score_rule_id` PK, `event_id` FK, `rule_code` UQ, `points_per_unit`, `unit_cap`, `requires_core`	NODE 800×10, CORE 2000×1, TIME 20×180, NO_HIT 1000×1
`reward_bundle` / `reward_item` / `reward_condition`	묶음, 구성품, 조건(`condition_type`, `threshold_value`, `repeat_step`, `repeat_limit`)	만료 30일, UQ(bundle_id, item_id)

사용자/로그 테이블

테이블	주요 필드	제약
event_account	ticket_balance, valid_clear_count, best_score, best_score_at_utc, last_daily_key	UQ(event, account, server), 0≤잔액≤4
daily_ticket_grant	daily_key, granted_count, discarded_count	UQ(event_account, daily_key)
play_session	status, entry_request_id, last_seq, hit_count, node_count, core_reached	ACTIVE 계정당 1개
score_result	node_score, core_score, time_score, no_hit_score, total_score	session UQ, 0~14,600
daily_completion	daily_key, first_result_id	UQ(event_account, daily_key)
reward_grant	status, trigger_key, mail_id	UQ(event_account, condition, trigger_key)
ranking_snapshot / ranking_entry	participant_count, rank_no, best_score, bundle_id	UQ(event, server), UQ(snapshot, rank_no)
session_input_log	seq_no, input_type, server_at_utc, accepted, reject_code	UQ(session, seq_no)
reward_delivery_log, event_audit_log	시도 번호, 결과 코드, 액션 코드	재시도/감사 이력
[근거: 05_데이터_테이블.md]		

리소스 목록: 드론, 노드 3상태, 코어, 선로, 감지 영역 3종(PULSE/PATROL/CROSS), 구획 점등 스프라이트, 경고음, 승인음, 완료 연출, S00~S07 UI. 수량 기록 없음. [추정: 화면/연출 요소에서 역산]

4. 작업 순서

1. 레벨 규칙 확정(구간 표, 상한)과 맵 좌표 제작.
2. DB 스키마/서버 객체 구현 후 입장/결과 시퀀스 연동.
3. 클라이언트 화면 S01~S07과 서버 응답 연결.
4. 아트/사운드 통합.
5. 내부 플레이 테스트로 점수 분포/완성율 측정, 가정 수치 교체.
6. 보상 총량 경제 검수 후 확정. [추정: 02 메인기획서 §5의 제작 순서에서 구현 단계로 확장]

5. 검수 기준

- 입장 실패 시 티켓이 차감되지 않고, 중복 요청에서 세션이 1개만 생긴다. [근거: 03_규칙_설계.md]
- 클라이언트 점수를 조작해도 서버 점수가 바뀌지 않는다. [근거: 03_규칙_설계.md]
- 같은 입력 로그 재처리 시 동일 점수, 14,600 초과 없음. [근거: 05_데이터_테이블.md]
- 동점 순위가 확정 시각 → 계정 ID 순으로 한 명씩 매겨진다. [근거: 02_보상_설계.md]
- 보상 원장 재실행에도 중복 우편이 없다. [근거: 02_보상_설계.md]
- 위험 영역이 색 없이도 식별된다. [근거: 08_화면_설계.md]

- 첫 유효 완료율, 코어 완성율, 노드별 도달율, 구간 체류 시간, 피격 수를 측정한다. [근거: 13_게임_콘셉트와_설계_의도.md]