캐슬 브레이커 — 메인기획서

> 재구성 기획서 (역설계) / 작성 기준 시점: 2026-08-07 (프로젝트 시작 시점) / 문서 버전: Ver 0.1 프로젝트 ID: P016 / 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. 핵심 메카닉스

라운드 상태 머신

- 의도(Why): 시간이 정지된 선택 구간과 입력 없는 실시간 구간을 분리해 블라인드 배팅과 결정론 동기를 동시에 얻는다.
- 설계(How): 상태는 `SELECTION → REVEAL → REALTIME → (SELECTION …) → RESULT`다. SELECTION은 틱/수입 정지, 최대 60초, 양측 준비 완료 시 즉시 종료하고 미확정 예약도 유효하다. REVEAL은 2초 연출이며 누적 시간/수입에서 제외한다. REALTIME은 30Hz로 진행하며 라운드 텀은 `min(30×n, 120)`초다(R1=30, R2=60, R3=90, R4 이후 120). 누적 600초가 되면 성 HP 비교로 판정승을 가린다(동일하면 무승부). 총 7라운드이고 마지막 60초분 수입(420)은 쓰지 못한다(실질 예산 3,980). 상태 이름과 `LIMIT_SEC=600`, `REVEAL_SEC=2` 등은 상수로 확인된다. [근거: 오펜스게임_기획서.md §4, §15-6, constants.ts]
- 필요(What): `MatchState`(`round`, `phase`, `tick`, `phaseTick`, `termTicks`, `funds`, `spawnQueue` 등), `roundTermSec()`.
- 작업 방법: 상태 전이 함수 구현 → R1~R7 시작 시각 검산 → 핫시트 확인.
- 완료 기준: 선택 시작 시각이 0/30/90/180/300/420/540초이고 600초에 종료한다.
- 리스크/대안: 백그라운드 탭에서 `requestAnimationFrame`이 멈춰 전투가 정지한다(게임으로는 정상으로 간주). [근거: README.md]
- 근거: [근거: 오펜스게임_기획서.md §4.2, §18 M1]

경제와 지불

- 의도(Why): 라운드 텀이 길어질수록 예산이 커지게 해 별도 성장 시스템 없이 후반 판돈을 키운다.
- 설계(How): 시작 200, 초당 +7(30틱마다 정수 가산), REALTIME에서만 축적, 잔액 이월/상한 없음. 예약은 선택 종료 순간 일괄 결제하고, 예약 합계가 잔액을 넘으면 준비 완료가 막힌다. 30기 상한으로 소환되지 못한 큐는 다음 선택 진입 시 전액 환불한다. 라운드별 지급 누적은 200/410/830/1,460/2,300/3,140/3,980. [근거: 오펜스게임_기획서.md §5, §14.3]
- 필요(What): 상수 `START_FUNDS`, `INCOME_PER_SEC`, 환불 로직.
- 작업 방법: 틱 루프에서 `tick % 30 === 0` 시 가산, 선택 종료에서 결제.
- 완료 기준: 상한 도달 시 환불 금액이 정확하다.
- 리스크/대안: 퍽(개전 자금 +100, 경제 강화 +3/초)이 양측 대칭을 깨므로 퍽과 경제는 함께 검증한다. [근거: constants.ts]
- 근거: [근거: 오펜스게임_기획서.md §5, constants.ts]

조립과 스텟 파생

- 의도(Why): 가격과 스텟이 같은 파츠에서 나와 설계가 곧 실력이 된다.
- 설계(How): 기본 안정성 100, 기동 0, 공격 0, 사거리 20u, 주기 2.0초. 파츠 보정을 전부 유닛 단일 스텟에 합산하며 무기를 여러 개 달아도 통합 공격 1회다. 하한은 안정성 10, 사거리 0, 주기 0.8초. 이동속도는 기동×20 밀리유닛/틱(=기동×0.6 u/s), 회피는 기동×25bp(상한 4000bp). 주기 보정은 팔/특수에만 두어 화력×연사 곱연산을 막는다. 무장 0은 방벽, 다리 0은 고정 포탑으로 허용한다. [근거: 오펜스게임_기획서.md §6~§8, §15-1, constants.ts]
- 필요(What): 파츠 18종 정의, `assemble()`과 검증 V-1~V-7.
- 작업 방법: 데이터 표 → 계산 → 대표 빌드 14종 대조(`npm run check`).
- 완료 기준: 대표 빌드 수치가 §8.3과 정확히 일치한다.
- 리스크/대안: 열등 파츠가 재발하지 않도록 신규 파츠는 죽은 파츠 검사를 거친다.
- 근거: [근거: 오펜스게임_기획서.md §8, §16.3, package.json]

이동, 타겟, 교차 레인

- 의도(Why): 근접 물량도 화력으로 환산되게 하고, 하나의 숫자 문턱으로 항공-대공 계층을 만든다.
- 설계(How): 유닛은 아군/적군 모두 통과(겹침 허용). 매 틱 사거리 내 타겟이 있으면 정지, 없으면 전진하며 적 성 접촉 지점(d=588)을 넘지 않는다. 공격력 0 유닛은 같은 레인 최전방 적과 접촉 거리 12u에서 멈춘다. 타겟 우선순위는 ① 같은 레인 가장 가까운 적 ② 교차 레인(지대공 사거리 ≥ 50, 공대지 사거리 > 0) ③ 성(공중은 사거리 > 0) ④ 전진. 레인은 비행 가능 여부로 결정되며 플레이어가 고르지 않는다. 대안(충돌 유지, 기본 사거리 30, 간격 8u)은 각각 광선검 항공기 봉쇄 붕괴와 판독성 규약 재설계 때문에 기각했다. [근거: 오펜스게임_기획서.md §9, §14.2, §15-9]
- 필요(What): `findTarget`, 상수 `AA_MIN_RANGE=50`, `AG_MIN_RANGE=1`.
- 작업 방법: 사거리 49/50 경계 테스트 → 헤드리스 상성 측정.
- 완료 기준: 사거리 49는 공중을 못 때리고 50은 때린다.
- 리스크/대안: 세로 몸통 광선검 항공기는 의도적으로 열린 고위험 니치다(HP 70, 210원 이상).
- 근거: [근거: 오펜스게임_기획서.md §9.4, §15-12, constants.ts]

데미지, 회피, 자폭, 소환

- 의도(Why): 방어력 없는 단순 데미지로 스텟을 6종으로 유지하고, 겹침으로 생긴 뭉침을 자폭으로 처벌한다.
- 설계(How): 쿨다운이 0 이하이고 타겟이 있으면 회피 판정(`rng % 10000 < 회피bp`, 유닛 타겟만), 실패 시 공격력만큼 HP 감소. 스플래시는 같은 레인 반경 내 적 전원이며 회피는 대상별. 성은 회피/반격이 없다. HP 0 이하는 틱 종료 시 일괄 제거. 자폭 유닛은 첫 공격 틱에 소멸하며 처치 수로 잡지 않는다. 소환은 REALTIME 진입 후 0.4초(12틱) 간격으로 d=15u에 고정 소환, 진영 30기 상한. [근거: 오펜스게임_기획서.md §9.5~§9.7, §14, constants.ts]
- 필요(What): `Rng`(xorshift32), `SpawnOrder`, 소환 큐.
- 작업 방법: ID 오름차순으로 공격 처리해 RNG 소비 순서를 고정한다.
- 완료 기준: 동일 시드 재실행 결과가 비트 단위로 같다.
- 리스크/대안: 자폭의 강도는 실제 플레이 데이터 필요. 상수만 조정한다.
- 근거: [근거: 오펜스게임_기획서.md §16.5, rng 설계 §9.6]

퍽

- 의도(Why): 계정별 차이를 만들고 장기 정체성을 주는 메타 선택.
- 설계(How): 계정당 1개, 10종. 정적 보정(개전 자금 +100, 경제 +3/초, 사거리 +8u, 안정성 +12%, 이동 +15%, 주기 −8%[하한 6틱])은 덱에 적용하고 위치/시간 의존 효과(철벽 성채 HP +500/반사 15%, 방어 결계 100u 안 적 −30% 속도, 공습 45초마다 반경 30u/80 데미지)는 틱 로직에서 처리한다. NULLIFY는 상대 퍽을 무효로 하며 서로 들면 둘 다 꺼진다. [근거: perks.ts, constants.ts]
- 필요(What): `PerkId`, `effectivePerk`, `applyPerkToDeck`.
- 작업 방법: `npm run sim -- perk`로 미러 승률과 NULLIFY 검증.
- 완료 기준: 10종 모두 유리해지고 NULLIFY가 상대 효과를 지운다.
- 리스크/대안: 기획서 v1.0 본문에 퍽 절이 없어 밸런스 기준이 약하다.
- 근거: [근거: README.md "퍽"] [추정: 기획서 본문에 퍽 서술이 없음]

2. 구현 방법

클라이언트/코어 구조

- 의도(Why): 시뮬레이션을 UI/백엔드와 분리해 같은 코드를 클라, Worker, 헤드리스가 공유한다.
- 설계(How): `src/sim`(코어), `src/ai`, `src/backend`(추상화), `src/art`(SVG), `src/battle`(Canvas/30Hz 루프), `src/screens`(React), `src/storage`(localStorage). Worker는 `../../src/sim/*`를 그대로 import해 중복 정의가 없다. 전투 상태는 Zustand에 두지 않는다. [근거: README.md 구조 절, 오펜스게임_기획서.md §2.1]
- 필요(What): Vite 빌드, `DEPLOY_TARGET=cfpages` 빌드 변형.
- 작업 방법: sim → ai → battle → screens → backend 순.
- 완료 기준: `npm run build`(타입 체크+빌드) 통과.
- 리스크/대안: 기획서의 `packages/*` 모노레포는 실제로 `src/`+`worker/` 2프로젝트로 단순화되었다.
- 근거: [근거: README.md, package.json]

온라인 동기화

- 의도(Why): 무료 플랜에서 실시간 소켓 없이 두 사람이 같은 전투를 보게 한다.
- 설계(How): 매치 시드와 양측 라운드 예약만 주고받는다. 클라이언트는 `/api/match/:id/orders`(POST)로 예약을 올리고 `/api/match/:id/round/:n`(GET)을 0.8초 폴링해 양측 예약을 받는다. 상태 해시는 `/hash`로 제출한다. presence는 5초 핑, 60초 초과 시 stale로 표시한다. 종료는 `/end`에서 승자/사유를 기록한다. 실시간 구간 트래픽은 0이다. [근거: worker/src/index.ts, README.md, constants.ts]
- 필요(What): D1 `matches`, `match_rounds`, `queue`, `players` 테이블.
- 작업 방법: Worker 배포 → Pages 배포 → 두 브라우저로 완주 확인.
- 완료 기준: 두 브라우저로 한 판이 끝까지 진행된다.
- 리스크/대안: 해시 불일치 교정과 서버 권위 검증은 코드에서 확인하지 못했다. 몰수패 자동화는 Durable Objects(WebSocket) 전환 시 `backend/cloudflare/matchmaking.ts`, `match.ts`만 교체하도록 인터페이스를 유지했다. [근거: README.md]

- 근거: [근거: worker/src/index.ts, README.md "남은 한계"]

매칭

- 의도(Why): 동시 요청에서도 이중 매칭을 막는 서버 확정 방식.
- 설계(How): `/api/queue/poll`을 처리하는 Worker 핸들러가 직접 페어링한다. 후보 중 레이팅 차가 가장 작은 상대를 고르고, 조건부 `UPDATE … WHERE matched_match_id IS NULL`이 두 행을 모두 갱신한 요청만 매치를 생성한다. 시드는 현재시각과 두 uid로 만든 32비트 값이다. 클라이언트 폴링은 1.5초. [근거: worker/src/matchmaking.ts, README.md]
- 필요(What): 큐 테이블, `claimQueuePair`.
- 작업 방법: 두 클라이언트를 동시에 큐에 넣어 매치가 1개만 생기는지 확인.
- 완료 기준: 이중 매칭이 발생하지 않는다.
- 리스크/대안: 기획서의 FIFO와 달리 레이팅 근접 매칭이 되었다(확장 항목이던 것을 앞당김). 큐 정리(오래된 행 삭제) 로직은 미확인.
- 근거: [근거: matchmaking.ts, 오펜스게임_기획서.md §11.1]

데이터 파이프라인과 도구

- 의도(Why): 수치 검증을 사람의 플레이 없이 반복한다.
- 설계(How): `npm run check`(대표 빌드/교차 레인 문턱 검증), `npm run sim`(결정론/AI 라운드로빈/밸런스), `sim -- duel`(동일 예산 교환비), `sim -- bal`, `sim -- perk`. 배포는 `main` push 시 GitHub Actions가 Worker → Pages 순서로 재배포하며 D1 마이그레이션은 스키마 변경 시 수동이다. [근거: README.md, package.json]
- 필요(What): `scripts/check-assemble.ts`, `scripts/headless.ts`, Cloudflare API 토큰 비밀키.
- 작업 방법: 상수 변경 → `sim` 재측정 → 목표표와 대조.
- 완료 기준: BAL-1~3 합격(현재 미충족).
- 리스크/대안: 비밀키 미등록 시 자동 배포 실패.
- 근거: [근거: README.md]

3. 시스템 간 상호작용과 밸런스 기준

- 핵심 변수: 파츠 가격, 사거리(특히 50 문턱과 100~150 구간), 기동성(속도+회피), 주기, 라운드 예산 곡선, 30기 상한. [근거: 오펜스게임_기획서.md §8, §16]
- 상성: 장거리 → 근접 물량 → 고기동 → 장거리. 항공 → 지상 → 대공(사거리 50+) → 항공. [근거: 오펜스게임_기획서.md §15-9]
- 합격 기준: BAL-1(러시 vs 방벽) 어느 쪽도 60% 초과 금지, BAL-2(포병 vs 근접) 포병 55% 미만, BAL-3(항공 vs 대공) 대공 0기 85% 초과 / 3기 40% 미만. [근거: 오펜스게임_기획서.md §16.2]
- 현재 실측: BAL-1 100%, BAL-2 95%로 불합격, BAL-3 합격. 좌우 미러가 일치해 엔진은 대칭이다. [근거: 오펜스게임_기획서.md §16.5]
- 곡선 검산: 경보병 1기만으로는 성(3,500)을 못 끝낸다(약 199초). R7(540초)에 경보병 6기가 도달하면 약 33초에 파괴 가능. [근거: 오펜스게임_기획서.md §16.4]
- 조정 원칙: 로직은 건드리지 않고 상수만 조정한다. [근거: 오펜스게임_기획서.md §18 M1.5]

4. 게임의 나머지 부분의 방향

- 콘텐츠 구조: 스테이지/챕터 없음. 단일 대전 모드(AI 연습 / 온라인). 파츠는 해금 없이 처음부터 전부 개방. [근거: 오펜스게임_기획서.md §10]
- 메타 진행: Elo 레이팅(시작 1000, K=32, 하한 100, 몰수패도 일반 패배), 리더보드 상위 50위와 본인 고정 행, 퍽 선택. 프리셋 30슬롯/덱 5벌. [근거: constants.ts, 오펜스게임_기획서.md §12.5]
- UI/UX 방향: 격납고 중심 홈, 조립은 연출과 즉시 반영, 첫 접속 시 튜토리얼 6페이지, 한줄소개 60자로 매칭 화면에서 상대에게 표시. [근거: 오펜스게임_기획서.md §12.4, §12.10, constants.ts]
- 아트/사운드: 플랫 기하 SVG만. 사운드 기록 없음.
- 저장/설정: 온라인 미설정 시 `localStorage`, 설정 시 D1이며 같은 프로필 스키마를 쓴다. 설정 화면 기록 없음. [근거: README.md]
- AI 연습: 가상 플레이어 5인은 고정 덱/고정 퍽/고정 행동 패턴. 플레이어 레이팅에 가까운 상대 2명 중 하나를 고른다. AI는 전장에 나와 있는 유닛만 읽어 블라인드 규칙을 지킨다. [근거: opponents.ts, README.md]
- 후순위: 리플레이(시드+예약만 저장하면 재현), 관전, 모바일, PIN, 코스메틱. [근거: 오펜스게임_기획서.md §19]

5. 제작 과정 설계

- 순서: ① 상수, 파츠, 조립 계산기 → ② 전투 틱/결정론 검증 → ③ 헤드리스 밸런스 → ④ 아트 파이프라인/조립 연출 → ⑤ 셸 화면 → ⑥ 백엔드/온라인. 선행 조건은 ①②이며 ③은 ②와 병렬로 조정 가능하다. [근거: 오펜스게임_기획서.md §18]
- 병렬 가능: 아트(④)와 시뮬레이션 밸런스(③). 튜토리얼 일러스트와 로고는 ⑤와 병렬.
- AI/도구 분담: 기획자(사용자)가 방향과 규칙을 정하고, AI 에이전트가 기획서 명세대로 구현/검증한다. 구현 중 발견된 문제는 §15 결정 기록으로 환류한다. [근거: 오펜스게임_기획서.md 서문, §15] [추정: 서문의 "AI 에이전트" 전제]
- 검증 절차: `npm run check` → `npm run sim` → 브라우저 육안(30기 밀착, 조립 연출) → 온라인은 두 브라우저. 수용 기준은 §18 M0~M4. [근거: 오펜스게임_기획서.md §18]
- 인수인계: HANDOFF 없음. 인수인계 정보는 `README.md`와 기획서 §15에 분산되어 있다. 재개 시 먼저 기획서 §15를 읽도록 서문이 지시한다. [근거: 오펜스게임_기획서.md 서문]

6. 세부기획서 목록

문서	정해야 할 것
`03_세부기획서_시뮬레이션과_데이터.md`	파츠 표/상수 표/대표 빌드, AI, 퍽 수치, 밸런스 목표와 실측, 검산 방법
`03_세부기획서_온라인과_메타.md`	매칭/라운드 동기, presence, 몰수패 설계, D1 스키마, Elo, 리더보드, 프로필, 보안 정책
`03_세부기획서_UIUX와_아트.md`	화면별 명세, 조립 연출 AN-1~11, 전투 HUD/결과, 튜토리얼 6페이지, 플랫 아트, 팔레트, SVG 파이프라인