개인 프로젝트

# 캐슬 브레이커
# 메인 기획서



CASTLE BREAKER

조립식 오펜스 대전

GAME START

v0.1 오프라인 프로토타입 · 가상 플레이어 대전

조립 / 블라인드 라운드 / 결정론 전투 핵심 메카닉스

| 문서 최초 작성 | 문서 최종 수정 |
| --- | --- |
| 20260807 | 20260811 |
| 박도원 | 박도원 |

# 목차


## 개요 – 2P

1. 목차
2. 로그
3. 문서 계층

## 현황 – 5P

1. 컨셉 확정 시점 현황

## 목표 – 6P

1. 컨셉 목표 재확인
2. 이 단계의 목표

## 핵심 메카닉스 – 8P

1. 핵심 메카닉스 순환
2. 조립 슬롯 구조
3. 스텟 파생 공식
4. 주요 파츠 가격
5. 상성 순환
6. 전투 핵심 규칙

## 구현 – 15P

1. 구현 구조
2. 기술 선택과 이유
3. 결정론 동기화 흐름
4. 구현 필요 요소

## 작업 계획 – 19P

1. 구현 순서와 완료 기준
2. 리스크와 대응

## 리소스 – 21P

1. 리소스 수량

# 개요 – 로그

| 순번 | 작성일 | 작성자 | 작성 내용 | 비고 |
|---|---|---|---|---|
| 1 | 2026-08-07 | 박도원 | Ver 0.1 작성 | 핵심 메카닉스 초안, 기획서 v1.0 정리 |
| 2 | 2026-08-07 | 박도원 | 오프라인판 착수 | 조립 계산기와 로컬 전투 구현 시작 |
| 3 | 2026-08-09 | 박도원 | 전투 틱/AI 연습판 | 결정론 전투 코어와 AI 상대 연결 |
| 4 | 2026-08-11 | 박도원 | 규칙 개정 반영 | 겹침 허용/직결 슬롯 제거/자폭 파츠를 결정 기록에 반영 |
| 5 | | | | |
| 6 | | | | |

# 개요 – 문서 계층




컨셉 → 메인 → 세부 순으로 상위 문서의 결정을 목표로 받아 구체화함.
강조 표시 = 이 문서

# 현황 – 컨셉 확정 시점 현황

방향은 정해졌고 수치, 규칙, 구조는 문서화 단계임.

## 확정된 것

### 목표

설계하는 게임, 8~14분 1:1 대전

### 범위

AI 연습, 온라인, 조립, 덱, Elo, 튜토리얼

### 기술

TypeScript, React, Canvas 2D/정수 결정론

## 정해야 할 것

### 파츠/가격

파츠 18종의 보정치와 단가 곡선

### 전투 규칙

레인, 사거리, 충돌, 자폭 규칙

### 동기화

서버 구성과 라운드 경계 교환 방식

# 목표 – 컨셉 목표 재확인

**메카닉스가 달성할 목표**

### 조립 경제

가격과 스텟이 한 조립에서 파생됨

- 가격 = 파츠 합
- 죽은 파츠 0

### 블라인드 배팅

읽기 싸움이 성립하는 상성 순환

- 동시 예약
- 상성 3축

### 입력 없는 실시간

전투 중 트래픽 0의 동기화

- 정수 결정론
- 라운드 경계 교환

### 8~14분 1판

텀이 늘어나는 7라운드 구성

- 600초 제한
- 예산 곡선

# 목표 – 이 단계의 목표

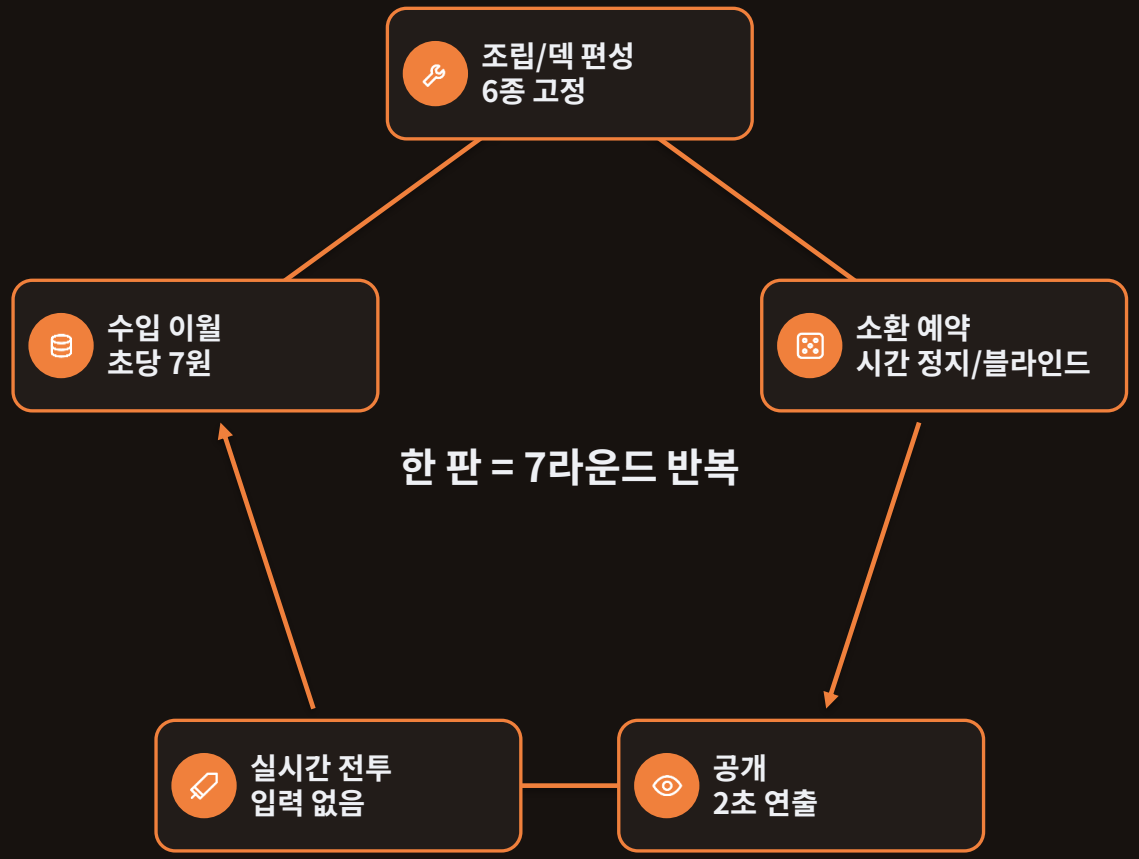
**조립 → 예약 → 결정론 전투가 한 판 안에서 끝까지 돌아가는 핵심 메카닉스를 수치로 확정함.**

| 18종 | 14종 | 3구도 |
| --- | --- | --- |
| 파츠 데이터 | 대표 빌드 검산 | 밸런스 검증 |

**의도** 세부 기획(UI, 아트, 온라인)이 얹힐 규칙 기반을 먼저 고정함.

**근거** 구현자가 AI라 모든 수치가 확정값이어야 재작업이 없음.

# 핵심 메카닉스 – 핵심 메카닉스 순환




**텀 증가**

라운드 텀 30초에서 120초까지 늘어남

# 핵심 메카닉스 – 조립 슬롯 구조

**유닛 1기**

## 몸통 (필수)

가로 30원 / 세로 40원

- 가로: 안정 +10
- 세로: 사거리 +20

## 팔

몸통 허용 수만큼 (1 / 2)

- 로봇 팔
- 120mm 활강포
- 무장창

## 무기

팔 마운트 합계만큼

- 광선검/미사일
- 항공폭탄/자폭 탄두

## 다리/특수

각 1개, 선택

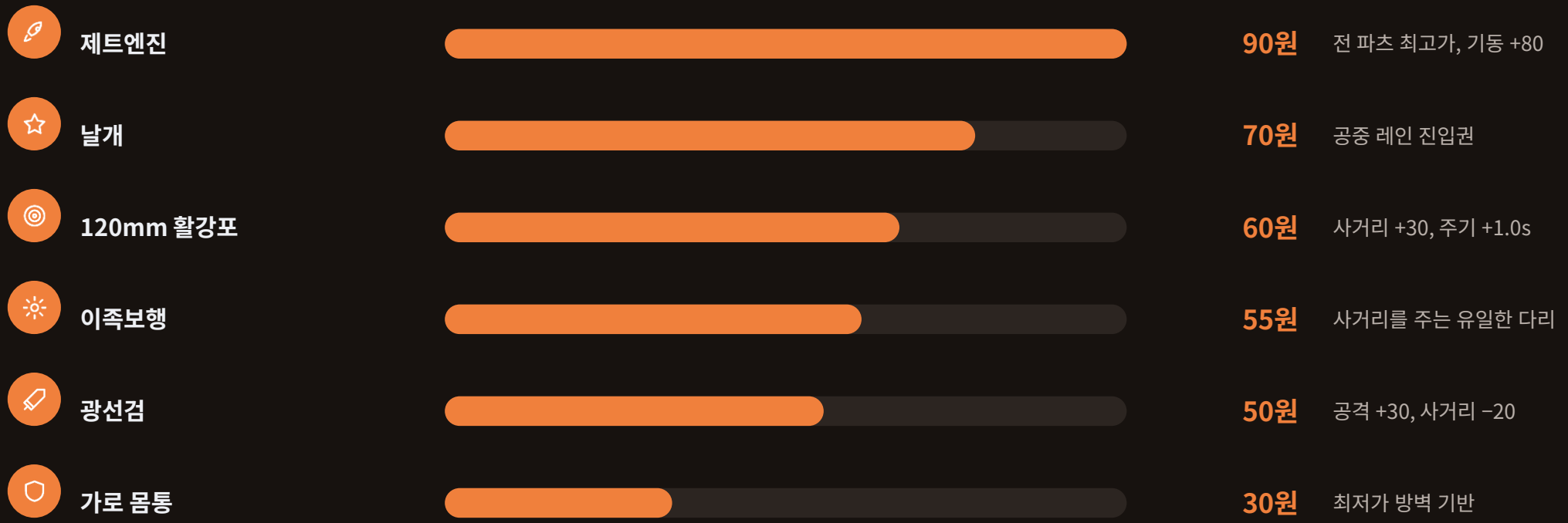
- 다리 5종
- 특수 4종

# 핵심 메카닉스 – 스텟 파생 공식

| 스텟 | 기본값 | 하한 | 파생 |
| --- | --- | --- | --- |
| 안정성(HP) | 100 | 10 | 파츠 보정 합산 |
| 기동성 | 0 | 0 | 이동속도 = 기동 × 0.6u/s, 회피 = 기동 × 25bp |
| 공격력 | 0 | 0 | 파츠 합산, 방어력/감쇄 없음 |
| 사거리 | 20u | 0 | 파츠 합산, 50 이상이면 대공 |
| 공격 주기 | 2.0초 | 0.8초 | 팔/특수 파츠만 보정, 30틱/초 |

# 핵심 메카닉스 – 주요 파츠 가격

단가는 제작 난이도 비율로 정함. 가격은 파츠 합계임.



# 핵심 메카닉스 – 상성 순환

기동성 →

### 러셔/가미카제

고기동이 장거리 병기를 잡음

### 폭격기

대공이 없으면 성까지 무저항

### 근접 물량/방벽

겹쳐 서서 화력을 합산함

### 포병/고정 포탑

접근하는 근접 편성을 녹임

사거리 →

# 핵심 메카닉스 – 전투 핵심 규칙

| 규칙 | 값 | 이유 |
|---|---|---|
| 유닛 겹침 | 충돌 없음, 아군/적군 통과 | 근접 1기만 공격하는 구조적 열세를 없앰 |
| 대공 문턱 | 지대공 사거리 50 이상 | 날개 70원을 최고가군에 두는 근거 |
| 공대지 문턱 | 사거리 0 초과 | 광선검 항공기를 기본 구성에서 봉쇄함 |
| 무장 0 유닛 | 최전방 적과 12u에서 정지 | 방벽이 앞에서 대신 맞는 방패가 됨 |
| 자폭 | 첫 공격 틱에 소멸, 반경 피해 | 겹침으로 뭉친 편성을 처벌하는 카운터 |

# 핵심 메카닉스 – 라운드 상태 전이

1

**선택**

시간/자금 정지, 최대 60초

2

**공개**

2초 고정, 시간 미포함

3

**실시간**

30Hz 진행, 텀만큼

4

**결과**

성 파괴 또는 600초

# 구현 – 구현 구조


castle-breaker
sim 코어
순수 함수, 클라/서버 공유
assemble
tick/round
rng(xorshift32)
client
React 셸 + Canvas 전투
screens
battle
art
server
매칭, 검증, 권위 실행
matchmaker
players
simcli
헤드리스 밸런스 측정
교환비
BAL 구도

# 구현 – 기술 선택과 이유

| 선택 | 내용 | 이유 |
| --- | --- | --- |
| 정수 연산 | 위치 밀리유닛, 시간 틱, 확률 bp | 부동소수점 차이로 인한 동기화 어긋남을 막음 |
| 결정론 RNG | xorshift32, 라운드 시드, ID 순 소비 | 양측이 같은 결과를 독립 계산함 |
| Canvas 2D | 유닛 최대 60기 규모 | WebGL이 필요 없는 규모임 |
| 상태 분리 | Zustand 셸, 전투 상태는 sim 안 | 렌더/UI가 시뮬 결과에 영향을 주지 않게 함 |
| 상수 단일화 | constants.ts에만 수치 기록 | 밸런스 조정을 한 곳에서 하기 위함 |

# 구현 – 결정론 동기화 흐름

1

## 예약 전송

프리셋 번호와 수량만 보냄

2

## 서버 검증

자금, 상한, 조립을 재계산함

3

## 공개 배포

양측 소환 목록과 라운드 시드

4

## 로컬 시뮬

전투 중 트래픽 0

5

## 해시 대조

불일치 시 스냅샷 교정

# 구현 – 구현 필요 요소

**핵심 메카닉스 구현**

### 시스템
조립, 전투, 라운드

- 유효성 V-1~V-7
- 소환 큐 12틱

### 데이터
파츠, 상수, 대표 빌드

- 파츠 18종
- 빌드 14종

### 도구
검산과 측정

- Vitest
- 헤드리스 sim

### 인력
기획 1인 + AI 구현

- 명세 작성
- 결과 검수

# 작업 계획 – 구현 순서와 완료 기준

1

## M0 조립 계산기

대표 빌드 스텟이 문서와 정확히 일치

check-assemble

2

## M1 로컬 전투

같은 시드면 해시 동일, 사거리 49/50 구분

핫시트 Canvas

3

## M1.5 밸런스

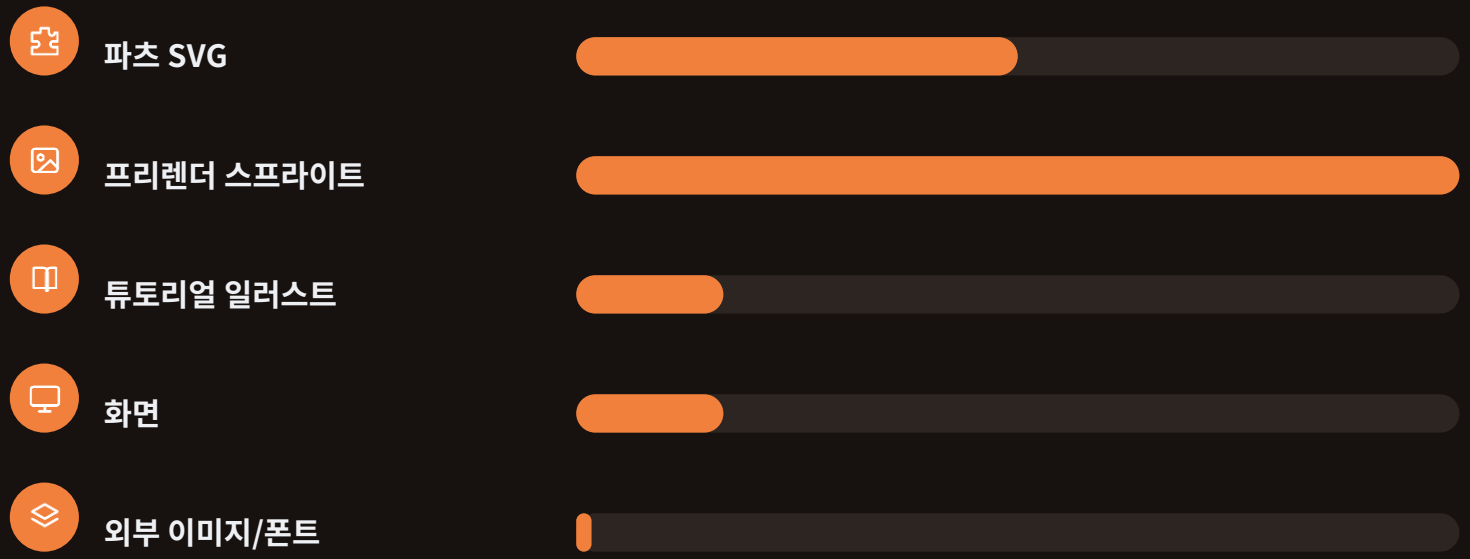
BAL-1~3 합격선 충족, 상수만 조정

simcli

# 작업 계획 – 리스크와 대응

| | |
|---|---|
| ⚠ 무기 다중 장착 시 화력과 연사가 곱해져 DPS가 폭주할 수 있음 | → 🛡 주기 보정을 팔/특수 파츠에만 부여해 DPS를 선형으로 둠 |
| ⚠ 다른 파츠의 완전 하위호환 파츠가 생길 수 있음 | → 🛡 죽은 파츠 검사로 파츠별 1순위 상황을 확인함 |
| ⚠ 근접 편성이 최전방 1기만 공격해 원거리가 압도할 수 있음 | → 🛡 유닛 겹침 허용으로 근접도 물량만큼 화력이 늘게 함 |

# 리소스 – 리소스 수량


파츠 SVG
18종
파츠 수와 동일, 기획서 표기 16종은 정정함
프리렌더 스프라이트
36장
파츠 18 × 진영 2색
튜토리얼 일러스트
6종
인라인 SVG 6페이지
화면
6종
메인, 인증, 격납고, 매칭, 전투, 결과
외부 이미지/폰트
0개
모든 그래픽을 코드로 그림

# 리소스 – 규격과 생산 방법

| 항목 | 규격 | 생산 방법/검수 |
| --- | --- | --- |
| 전장 캔버스 | 1280×720, 1u=2px, 전장 폭 1200px | Canvas 2D 직접 렌더, 1280×720 최소 해상도 확인 |
| 유닛 스프라이트 | 할당 24px, 실제 20px, 좌우 2px 여백 | 앵커 합성 후 캐시, 30기 밀착 스크린샷 검수 |
| 파츠 SVG | 직선/원 도형, 로컬 원점과 앵커 | 코드 정의 → 앱 기동 시 프리렌더 |
| 시뮬 시간 | 30Hz 틱, 렌더는 rAF | 동일 시드 해시 비교 |
| 명명 | 파츠 ID 접두 B_, A_, W_, L_, S_ | constants/parts 단일 출처 |