개인 프로젝트

쥐펀스게임 (찍찍찍 외전) 메인 기획서



핵심 메카닉스: 선회/자동 사격, 수류탄, 특수 쥐 상태

문서 최초 작성	문서 최종 수정
20261028	20261028
박도원	박도원

목차

개요 - 2P

1. 목차
2. 로그
3. 문서 계층

시작 현황 - 5P

1. 컨셉 확정 시점

목표 재확인 - 6P

1. 컨셉의 목표
2. 핵심 요소와 메카닉 대응

핵심 메카닉스 - 8P

1. 압박 → 선택 → 대가 루프
2. 선회 상태 설계
3. 쥐별 정답 매트릭스

구현 방법 - 11P

1. 구조와 판정 순서
2. 기술 선택과 이유

구현 필요 요소 - 13P

1. 필요한 시스템, 데이터, 도구

구현 계획 - 14P

1. 구현 순서와 완료 기준
2. 메카닉 리스크와 대안

리소스 - 16P

1. 재사용 리소스 수량
2. 규격과 생산 방법

개요 – 로그

순번	작성일	작성자	작성 내용	비고
1	2026-10-28	박도원	메인 기획 착수	제작 명세서 확정 직후 핵심 메카닉 설계 시작
2	2026-10-28	박도원	핵심 규칙 코어 설계 확정	선회, 사격, 수류탄, 특수 쥐 구현 순서 정리
3				
4				
5				
6				

개요 – 문서 계층



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

시작 현황 – 컨셉 확정 시점

확정된 것

컨셉 기획서

목표, 4요소, 루프, 제작 범위 확정

제작 명세서

선회, 사격, 수류탄, 쥐 8종 규칙 확정

근거 명세서 작성일 2026-10-28

재사용 대상

원작 리그/차력거인 관절/PNG 87개

남은 것

규칙 코어

구현된 코드가 아직 없음

수치 검증

추천값이 실제로 플레이 가능한지 미확인

원작 코드 연결

원작 리그를 새 규칙에 붙이는 방법 미정

목표 재확인 – 컨셉의 목표



방향 선택에 책임을 지게 하는 한 판짜리 디펜스를 규칙 코어로 먼저 구현함.

4종

핵심 메카닉(선회, 사격, 수류탄, 특수 쥐)

21개

규칙 테스트 목표 수

1/120초

고정 시뮬레이션 간격

의도 화면 없이 규칙부터 검증해야 AI 구현의 회귀를 빨리 잡을 수 있음.

근거 컨셉 목표 '조작은 줄이고 판단은 늘림'과 하루 단위 외전 규모.

목표 재확인 – 핵심 요소와 메카닉 대응

책임지는 방향 선택

멈출 수 없는 거인

자동 이동/자동 사격으로 구현

- 자동 이동
- 자동 사격

선회 지연

0.5초 선회 상태로 구현

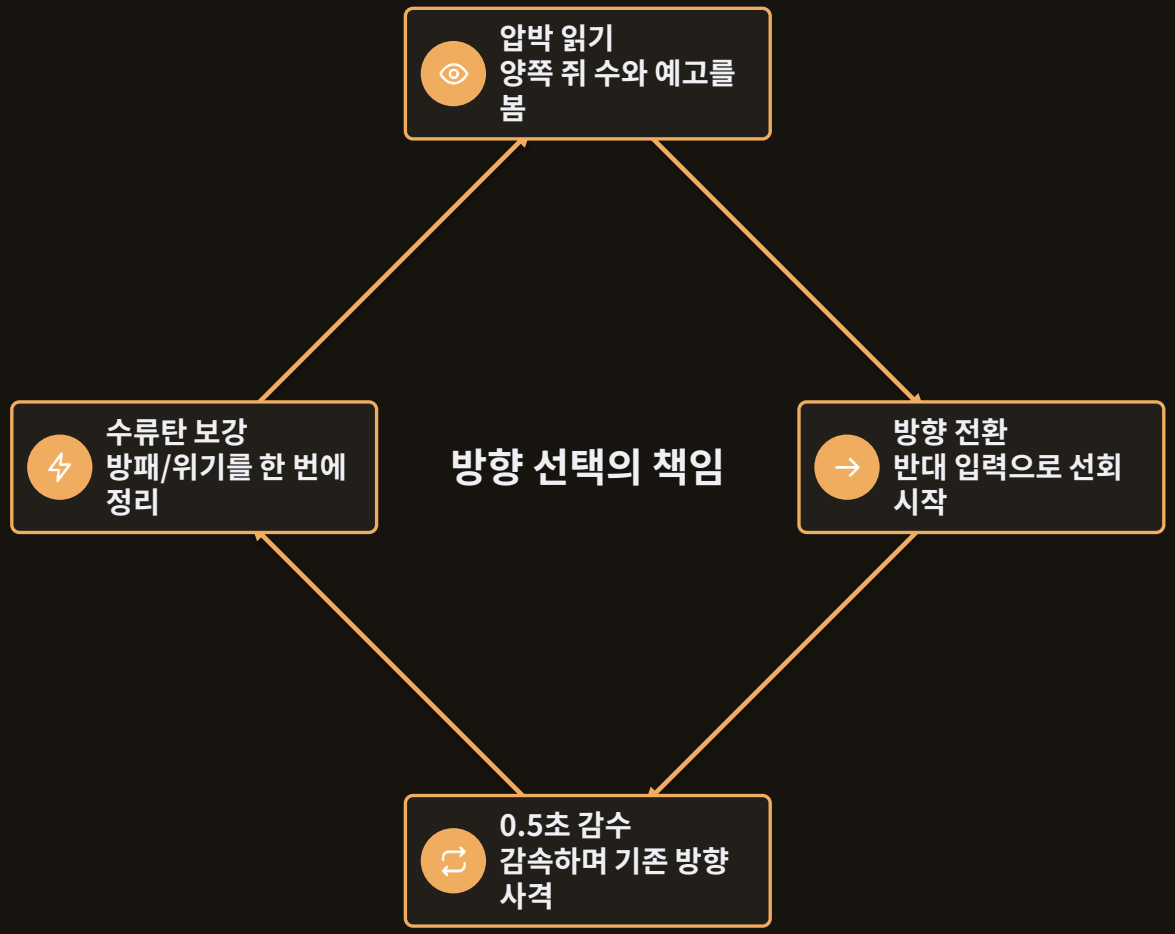
- 감속→반전→가속

쥐마다 다른 정답

수류탄과 특수 쥐 상태로 구현

- 수류탄
- 특수 쥐 7종

핵심 메카닉스 – 압박 → 선택 → 대가 루프



대가

선회 중 반대편이 0.5초 가까워짐

핵심 메카닉스 – 선회 상태 설계

단계	시간	움직임	의도
입력	즉시	반대 방향 지정	짧은 탭도 선회 요청으로 인정
감속	0.5초	속도 선형 감소, 기존 방향 사격	방향 선택의 대가를 만듦
반전	0.5초 시점	방향/이미지 뒤집기	사격 방향과 모습을 일치시킴
가속	0.2초	반대 방향으로 속도 회복	급정지 느낌을 피함
예외	화면 끝 140px	자동 선회	정지 불가 원칙 유지

핵심 메카닉스 – 쥐별 정답 매트릭스

대응 시점 (착지 후 → 즉시)

방패 쥐

총알 무효, 수류탄만 통함

맘스땃쥐

5발 또는 수류탄으로 처치

히어로/외계인

공중은 무적, 착지 뒤 처치

일반, 전기, 스트리머

한 발 처치, 우선순위 판단

총알로 처치 가능 (불가 → 가능)

구현 방법 – 구조와 판정 순서

1

config.js

속도, 시간, 점수 수치만 분리

의도 밸런스를 숫자만 바꿔 조정

2

core.js

프레임 독립 Game 클래스, 1/120초 고정 스텝

Node 테스트

3

view.js

입력, Canvas, 음향, 파편, 저장

의도 화면 없는 코어 검증을 위해 분리

4

판정 순서

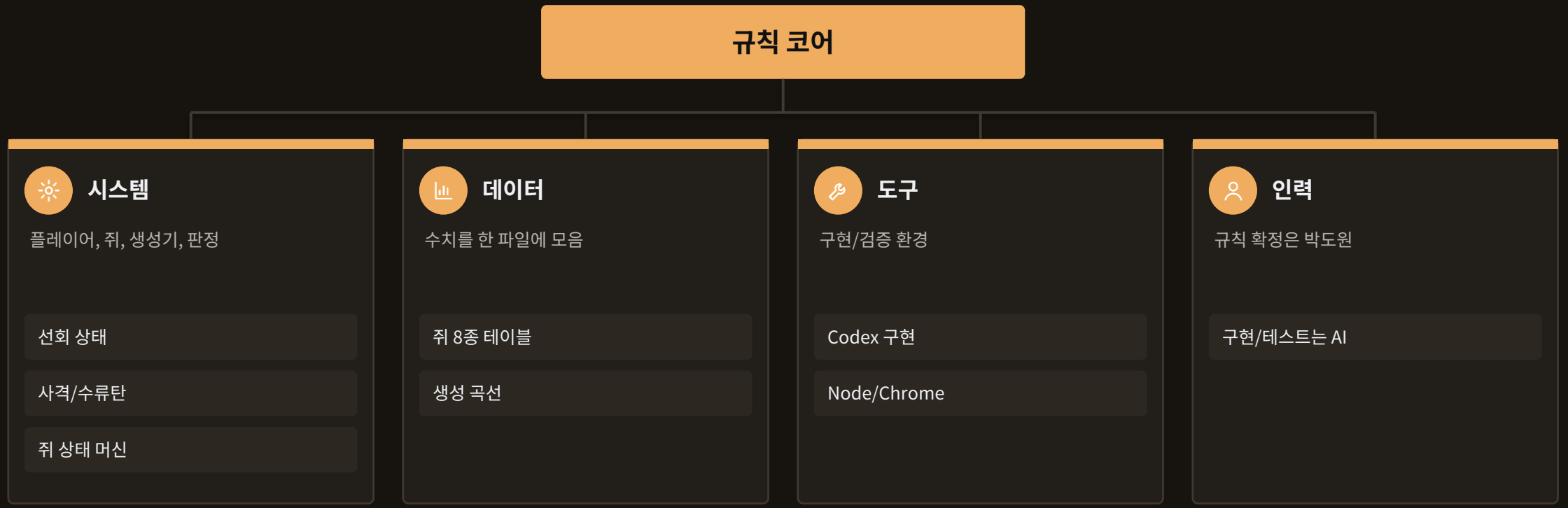
적중→사망 제거→접촉/영창 완료

근거 같은 프레임은 처치 우선

구현 방법 – 기술 선택과 이유

선택	대안	이유
HTML Canvas, 프레임워크 없음	Unity 이식	설치 없이 열어 검증하고 외전 공통 모듈을 재사용함
파일 분리(config, core, view)	단일 HTML	수정 지점을 나누어 AI가 숫자만 조정하게 함
탄환 선분-사각형 충돌	점 충돌	탄속 2304px/s에서 관통 누락을 막음
timeScale 2 시간 스케일	속도값 직접 상승	예고 타이머는 실제 시간 기준으로 유지함
Node 내장 테스트 러너	외부 테스트 도구	추가 설치 없이 규칙을 검증함

구현 필요 요소 – 필요한 시스템, 데이터, 도구



구현 계획 – 구현 순서와 완료 기준

1

기본 플레이

이동, 사격, 선회, 일반 쥐/접촉 종료

근거 선회 테스트 3종 통과

2

수류탄

쿨다운, 방향, 범위, 방패 차단

근거 수류탄 테스트 3종 통과

3

특수 쥐

예고, 무적, 착지, 영창, 말풍선

근거 종별 테스트 통과

4

통합 검수

전체 21개 테스트와 브라우저 확인

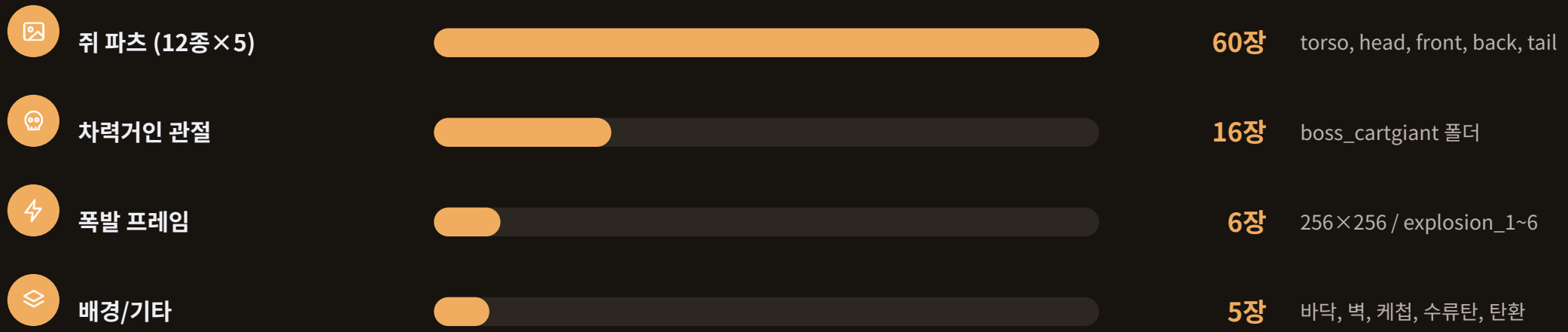
근거 처리량 테스트 포함

구현 계획 – 메카닉 리스크와 대안

⚠ 화면 끝 자동 선회가 명세서의 미확정 보완 규칙임	→ 🛡 벽 정지 대안은 정지 불가 원칙과 충돌해 배제함
⚠ 수류탄 도달거리 체감을 구현 전에 알 수 없음	→ 🛡 grenadeRange 수치만 조정할 예정임
⚠ 히어로/외계인 착지 직후 대응이 어려울 수 있음	→ 🛡 착지 후 회복 0.65초를 추가할 예정임

리소스 – 재사용 리소스 수량

새 일러스트 없이 원작 파일만 사용함. 수량은 원작 폴더의 PNG를 세어 확정함.



리소스 – 규격과 생산 방법

항목	규격	방법
쥐 파츠	PNG, 예: torso 434×264	원작 컷아웃을 그대로 복사해 리그로 조립
배경	floor 640×544 / wall 1782×156	원작 스테이지 파일 재사용
폭발, 케첩, 수류탄	PNG, 약 256~428px	FX 폴더를 복사, 폭발은 6프레임 순차 재생
BGM	mp3 1개, 약 3.8MB	원작 BGM_Hurry 재사용
방패	Canvas 도형	전용 일러스트 없이 buff 리그에 도형을 붙임