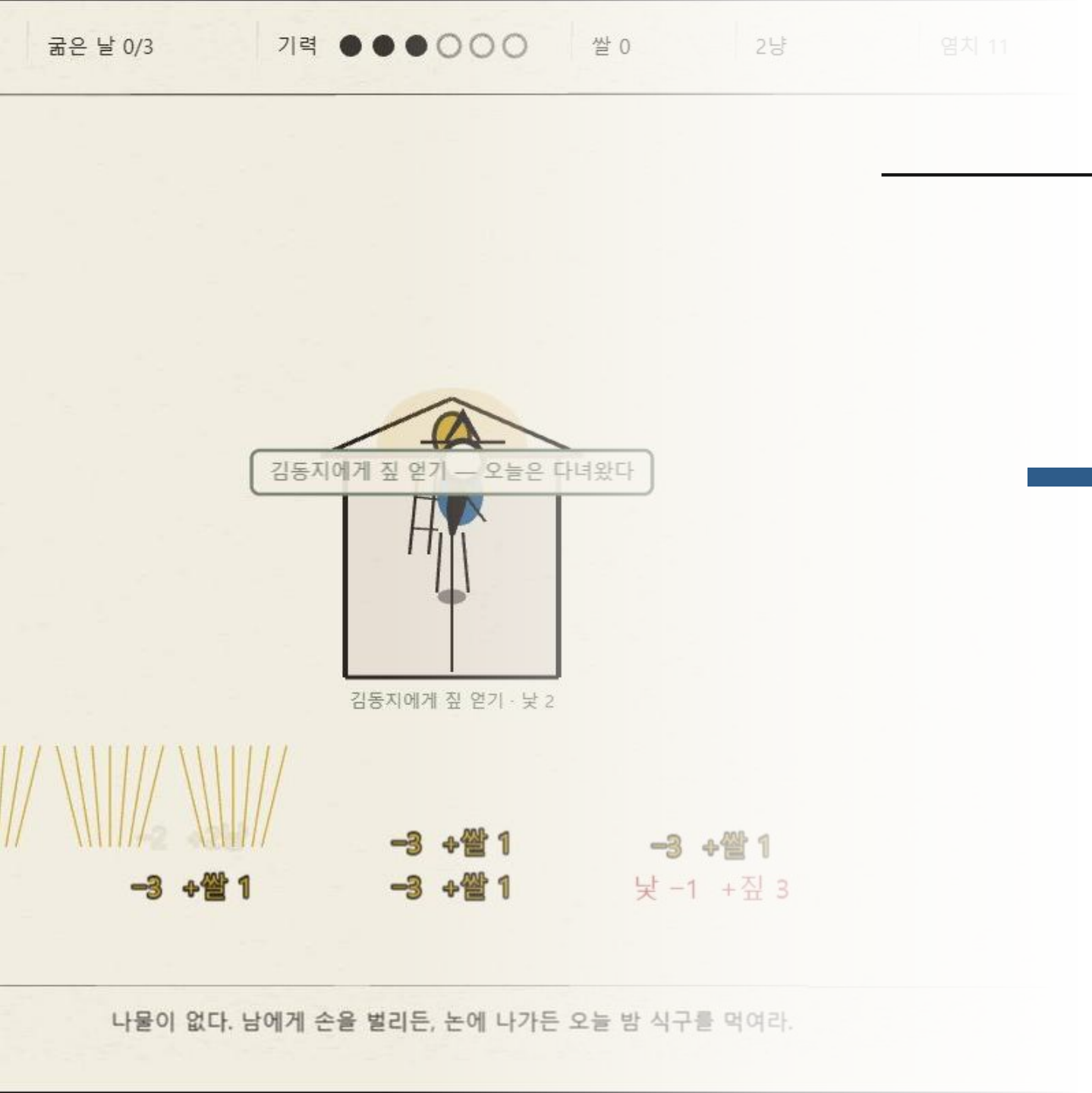


개인 프로젝트

홍부전 (소설->게임) 메인 기획서

염치, 기력, 박의 핵심 메카닉스와 구현 방법

문서 최초 작성	문서 최종 수정
20260804	20260804
박도원	박도원

목차

개요 - 2P

1. 목차
2. 로그
3. 문서 계층

현 상황 - 5P

1. 컨셉 확정 직후

목표 재확인 - 6P

1. 메인 단계의 목표
2. 컨셉 요소와 메카닉스의 대응

핵심 메카닉스 - 8P

1. 하루 루프
2. 하루 상태 전이
3. 염치와 문 규칙
4. 염치 잔량이 뒤집는 최적 행동
5. 박과 종료 흐름

구현 방식 - 13P

1. 구현 구조
2. 기술 선택과 이유

구현 필요 요소 - 15P

1. 구현에 필요한 것

구현 계획 - 16P

1. 구현 순서와 완료 기준

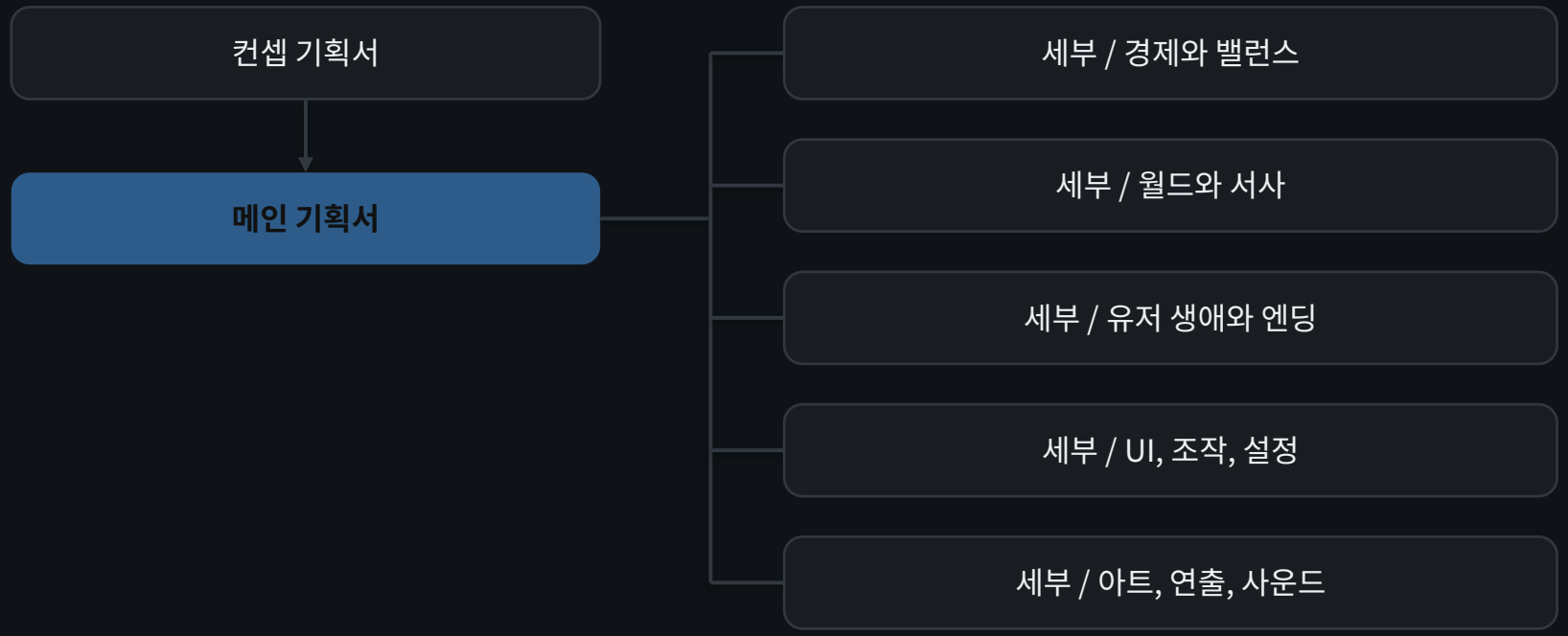
리소스 - 17P

1. 리소스 수량 (핵심 메카닉스 단계)
2. 리소스 규격과 생산 방법

개요 – 로그

순번	작성일	작성자	작성 내용	비고
1	2026-08-04	박도원	Ver 0.2 작성	핵심 메카닉스와 하루 상태 전이 설계
2	2026-08-04	박도원	순수 상태 분리	state.js와 화면 코드 분리 방침 확정
3	2026-08-04	박도원	상태 로직/화면 구현	상태, 화면, 음향 구현 착수
4	2026-08-04	박도원	핵심 구현 도달	플레이 가능 루프 확인, 세부 파트 착수
5				
6				

개요 – 문서 계층



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

현 상황 – 컨셉 확정 직후

확정된 것

염치 중심 생활경영, 웹, 민화 수묵/3막 16일

근거 컨셉 기획서 결과

점검된 수치 틀

톱질 하루 한 통/비용 1→4/후반 나물 제거

근거 극단 전략 점검 결과

남은 제약

외부 에셋 없음/별도 인력 없음/시연 인원 없음

목표 재확인 – 메인 단계의 목표

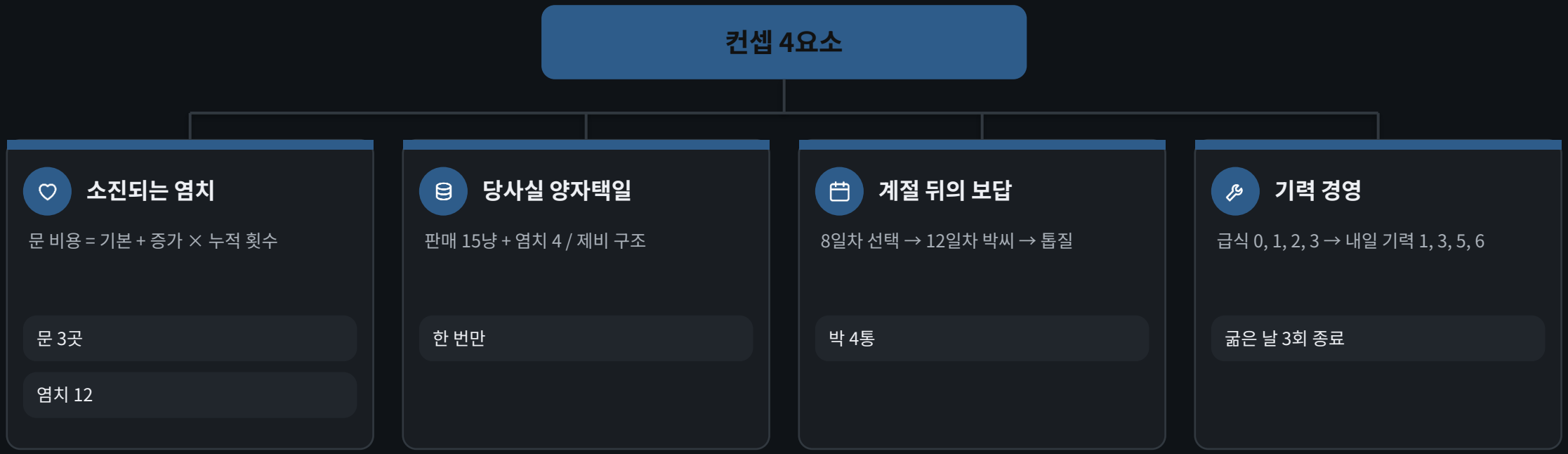
화면 없이도 한 해를 검증할 수 있는 순수 상태 규칙과 그 위의 Phaser 화면으로 핵심 루프를 완성함.

7종	3곳	3종
작업(노동) 종류	손 벌릴 문	종료 경로

의도 규칙 수정이 화면 전체를 흔들지 않게 하고, 설계 의도가 상태 전이로 이어지는지 확인함.

근거 수치 점검에서 극단 전략이 쉽게 생긴다는 점이 드러났음.

목표 재확인 – 컨셉 요소와 메카닉스의 대응



핵심 메카닉스 – 하루 루프



먹인 양 = 내일 기력

급식 0, 1, 2, 3에 기력 1, 3, 5, 6임

의도 가난을 대사 대신 행동 폭으로 보여 줌

굶은 날 3회

누적되면 즉시 종료함

근거 굶주림 엔딩의 근거

핵심 메카닉스 – 하루 상태 전이

1

기상

기력을 회복값으로 교체함

2

작업 판정

막→계절→오늘
수행→짚→기력 순 검사

의도 실패 시 자원이 변하지 않게 함

3

문 사용

누적 비용과 일일 제한을 분리함

4

잠자리

굶주림 한도를 먼저 검사함

근거 종료 우선순위 혼선 방지

5

날짜 전이

5, 8, 12일차 사건과 일일 제한 초기화

핵심 메카닉스 – 염치와 문 규칙

항목	규칙	의도
시작 염치	12	초중반 두세 번 구걸할 여유를 줌
김동지	1부터 +1씩, 짚 3	저렴하지만 반복에 취약함
형님	2부터 +2씩, 쌀 1	급할 때의 비싼 선택지임
아전	염치 3, 5냥, 단 한 번	영구 소진 사례를 둠
당사실 판매	15냥 + 염치 4	생계 회복과 구조 포기를 맞바꿈

핵심 메카닉스 – 염치 잔량이 뒤집는 최적 행동

박 진행 →

도달 불가

문이 닫혀 톱질 기력을 확보할 수 없음

형님 구걸

쌀 1을 얻고 톱질 #3까지 해 박 3통

생존 우선

염치 1이면 구걸 불가, 방아로 박 포기

소극적 노동

방아/무논으로 끼니만 해결, 박 0

염치 잔량 →

핵심 메카닉스 – 박과 종료 흐름

1

제비 선택

8일차 구조/방치를 확정함

2

12일차 분기

미구조면 그 해로 종료함

근거 선택의 결과를 계절 뒤에 돌려줌

3

심기

기력 1로 박씨를 심음

4

톱질

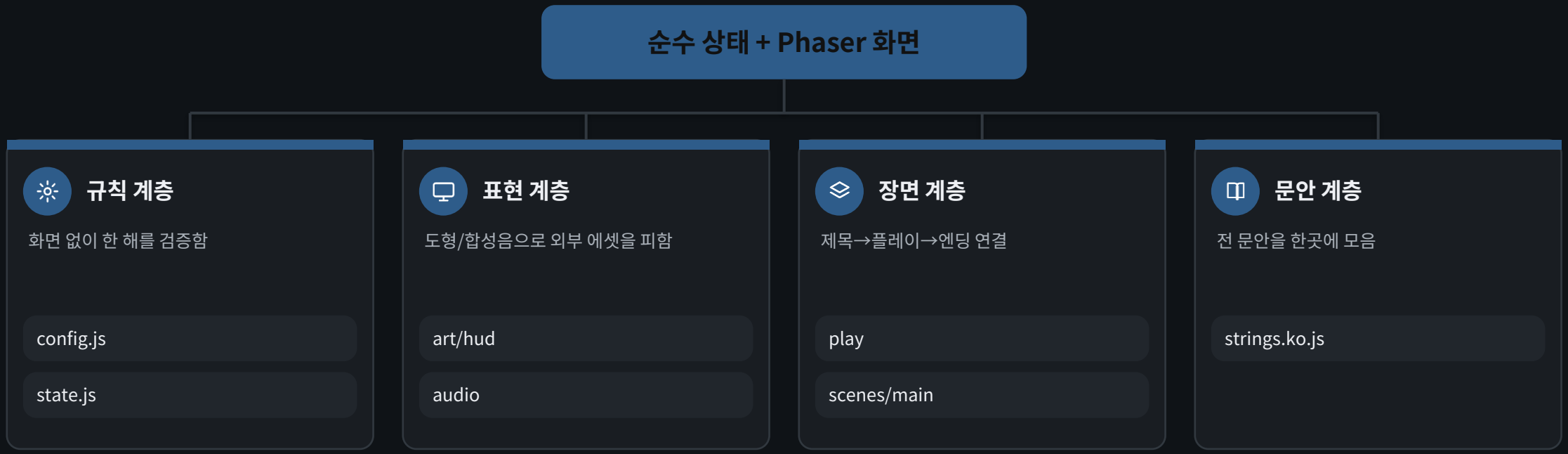
통마다 기력 1~4, 하루 한 통

5

16일 결산

켠 박/굶은 날/닫힌 문을 표시함

구현 방식 – 구현 구조


순수 상태 + Phaser 화면
규칙 계층
화면 없이 한 해를 검증함
config.js
state.js
표현 계층
도형/합성음으로 외부 에셋을 피함
art/hud
audio
장면 계층
제목→플레이→엔딩 연결
play
scenes/main
문안 계층
전 문안을 한곳에 모음
strings.ko.js

구현 방식 – 기술 선택과 이유

선택	이유	버린 대안
Phaser 4.2.1 (저장소 동봉)	웹/2D 이동/키 입력을 한 번에 얻음	순수 DOM
순수 상태 state.js 분리	같은 조작이 같은 결과여서 연산으로 검증 가능	씬 안에 규칙 혼재
수치는 config.js 한 곳	설계 문서 대조와 변경이 쉬움	곳곳에 하드코딩
Canvas 절차 도형 + WebAudio	총 5MB/3초 예산 안에서 외부 에셋 0	이미지/음원 파일
단일 HTML 포장 스크립트	파일 하나로 교부할 수 있음	폴더 묶음 교부

구현 필요 요소 – 구현에 필요한 것



구현 계획 – 구현 순서와 완료 기준

1

상수

규칙 값을 config로 확정함

JavaScript

의도 이후 모든 판정의 기준으로 삼음

2

순수 상태

화면과 분리해 판정함

state.js

근거 화면 없는 연도 검증이 필요함

3

문안/자산

한국어 문안과 절차 도형을 만듦

Canvas / WebAudio

4

장면 연결

Phaser 씬에 상태를 연결함

Phaser 4.2.1

5

검증

엔딩 3종이 모두 도달 가능해야 완료

Node / Puppeteer

근거 설계 문서의 검수 조건

리소스 – 리소스 수량 (핵심 메카닉스 단계)



리소스 – 리소스 규격과 생산 방법

구분	규격	생산/검수
화면	1280×800 가상 캔버스, 최저 1024×640	Phaser 캔버스, 두 해상도 캡처 확인
그래픽	먹선/한지 노이즈 절차 도형, 6색 팔레트	art.js 함수(시드 난수)로 생성
음향	WebAudio 합성 타악 12종	audio.js 노이즈/부분음 합성
명명	작업 id는 영문 소문자(meongseok 등)	config.js 한곳에서 관리
용량	총 5MB, 첫 화면 3초 이내	Phaser 약 1.3MB 포함 실측 확인

구현 필요 요소 – 구현 리스크

⚠ 당사실 판매로 염치가 회복되어 닫힌 문이 다시 열릴 수 있음	→ 🛡 영구 폐쇄 표현을 피하고 잔량 판정을 문안에 반영할 예정임
⚠ 심은 직후 톱질이 가능해 추석 수확 의도와 어긋날 수 있음	→ 🛡 규칙과 원작 의도의 차이를 기록하고 변경 여부를 판단함
⚠ 자동 이동 훅이 실제 길 찾기를 대신할 수 없음	→ 🛡 실제 보행 확인을 별도 절차로 둠