흥부전 (소설->게임) — 메인기획서

> 재구성 기획서 (역설계) / 작성 기준 시점: 2026-08-04 (프로젝트 시작 시점) / 문서 버전: Ver 0.1 프로젝트 ID: P012 / 근거 수준: A / 작성: Codex (재구성) 이 문서는 결과물과 제작 기록으로부터 기획 의도를 역추정해 당시 작성한 것처럼 재구성한 것이다. 확인된 사실과 추정을 [근거]/[추정] 태그로 구분한다.

1. 핵심 메카닉스

이동/대상 선택 — 장소에 가서 일을 한다

- 의도(Why): 생활경영의 반복 행동을 메뉴 선택만이 아니라 마을을 오가는 행위로 경험하게 한다. [근거: PRODUCT_BRIEF.md 공간 형태]
- 설계(How): 방향키/WASD의 입력 벡터를 정규화하여 대각선 속도를 맞추고 충돌 사각형을 검사한 뒤 위치를 갱신하며, 가장자리 출구에서 연결 구역으로 전환한다. [근거: play.js update, blocked, tryExit]
- 필요(What): 구역 연결표, 작업 지점 좌표, 충돌 영역, 현재 위치와 구역, 인접 대상 프롬프트가 필요하다. [근거: config.js ZONES/WORKS; play.js]
- 작업 방법: AI 구현이 구역과 대상을 배치하고 기획이 실제 보행으로 각 대상의 접근/출구 복귀/문구 표시를 확인한다. [추정: 자동 이동 훅만으로 보행 경험을 대신하지 않기 위한 절차]
- 완료 기준: 이동 후 대상의 비용과 결과가 표시되고 Z 또는 Space로 해당 행동을 실행한다. [근거: BUILD_BRIEF.md 핵심 동사; play.js]
- 리스크/대안: 제품 브리프는 이동을 기력/시간 비용으로 설명하지만 실제 이동은 기력을 깎거나 날짜를 넘기지 않으므로, 구현 설명에는 무료 이동으로 기록한다. [근거: PRODUCT_BRIEF.md; play.js update]
- 근거: [근거: config.js; play.js]

노동 — 유한한 하루 기력의 배분

- 의도(Why): 모든 일을 할 수 없게 하여 끼니, 현금, 박 작업 중 오늘 필요한 것을 고르게 한다. [근거: GAME_DESIGN.md §6/§11]
- 설계(How): 작업마다 해금 막→계절→오늘 수행 여부→짚 필요량→기력 순으로 가능 여부를 판정하며, 성공하면 기력을 차감하고 냥/쌀을 더한 뒤 오늘 완료 표시를 남긴다. [근거: state.js workStatus/doWork]
- 필요(What): 작업 ID, 기력 비용, 산출, 막, 구역, 좌표, 짚 필요 여부/봄 전용 여부가 필요하다. [근거: config.js WORKS]
- 작업 방법: 기획이 작업 표를 대조하고 AI가 판정, 소모, 보상을 한 경로로 처리하며, 같은 날 재입력과 기력 부족 입력을 검사한다. [추정: 중복 지급을 막기 위한 기존 규칙 기반 절차]
- 완료 기준: 하루 각 작업은 한 번만 보상하고 기력 부족이나 재료 부족이면 자원을 바꾸지 않는다. [근거: state.js doWork]
- 리스크/대안: 특정 작업의 효율만 높이면 모든 국면에서 같은 일을 하므로 계절 제한, 짚, 쌀 직접 산출의 차이를 함께 유지한다. [추정: 작업의 자원 형태와 조건 차이가 선택을 만드는 구조]
- 근거: [근거: config.js WORKS; GAME_DESIGN.md §9]

밤 정산 — 끼니가 다음 날의 행동 폭이 된다

- 의도(Why): 가난을 대사만으로 설명하지 않고 먹인 양과 내일의 노동 능력으로 연결한다. [근거: 세션 S1 15:35; GAME_DESIGN.md W1]
- 설계(How): 잠자리에서 `min(3, 보유 쌀)`을 자동 소비하며, 먹임 0, 1, 2, 3에 따라 다음 기력을 1, 3, 5, 6으로 정하고 0이면 굶은 날을 누적한다. [근거: config.js WAKE/FEED_MAX; state.js sleep]
- 필요(What): 날짜, 쌀, 기력, 굶은 날/마지막 급식량과 오늘 수행 기록, 정산 화면이 필요하다. [근거: state.js create/sleep; play.js]
- 작업 방법: AI가 정산을 단일 함수로 구현하고 기획이 급식량별 전이와 종료일의 판정 우선순위를 대조한다. [추정: 정산 누락과 중복 소비를 막기 위한 절차]
- 완료 기준: 굶은 날 3회에서 즉시 종료하고 생존한 다음 날에는 오늘 작업, 문, 톱질 제한이 초기화된다. [근거: state.js sleep]
- 리스크/대안: 굶은 날은 연속일이 아니라 누적일이며 쌀을 먹어도 감소하지 않고, 먹일 양을 임의 선택하는 기능도 없으므로 문안을 이에 맞춘다. [근거: state.js feedAmount/sleep]
- 근거: [근거: config.js; state.js sleep]

염치/문 — 도움을 미래의 비용으로 바꾼다

- 의도(Why): 도움을 모두 쓰는 전략보다 가을까지 남기는 판단이 유리해지도록 한다. [근거: GAME_DESIGN.md §9]
- 설계(How): 비용은 기본 비용 + 증가량 × 누적 횟수이며 김동지는 1부터 1씩 증가해 짚 3, 형님은 2부터 2씩 증가해 쌀 1, 아전은 염치 3으로 5냥을 단 한 번 준다. [근거: config.js DOORS; state.js doorCost]
- 필요(What): 염치 시작 12, 누적 사용 `doorUse`, 오늘 사용 `doorToday`, 형님 방문 `begCount`, 다음 비용 표시가 필요하다. [근거: config.js; state.js create]
- 작업 방법: 기획이 다음 비용/가을 기력과의 교환을 확인하고 AI가 누적 비용과 일일 제한을 분리해 구현한다. [추정: 다음 날 비용까지 초기화하는 오류를 피하기 위한 절차]
- 완료 기준: 같은 문은 하루 한 번, 아전은 전체 한 번이며 비용 미달에서는 지급 없이 닫힘을 표시한다. [근거: state.js doorStatus/useDoor]
- 리스크/대안: 당사실 판매의 염치 회복 예외가 있으므로 '두 번 다시 열리지 않는다'는 절대 표현을 구현 사실과 동일시하지 않는다. [근거: CONCEPT.md §6; state.js sellThread]
- 근거: [근거: config.js DOORS; state.js]

장/당사실 — 생계와 선의를 같은 재산에 묶는다

- 의도(Why): 구조의 대가를 돈/염치라는 이미 학습한 자원의 손실로 이해시키도록 한다. [근거: CONCEPT.md §2]
- 설계(How): 쌀은 3냥에 1을 구매하고, 당사실 판매는 보유 상태를 거짓으로 바꾸며 15냥과 염치 4를 지급하고, 제비 구조도 같은 보유 상태를 소모한다. [근거: config.js; state.js buyRice, sellThread, helpChick]
- 필요(What): 장의 두 선택지, 소지 여부, 판매 경고, 구조/지나침 결과 상태가 필요하다. [근거: play.js showMarket/showChick]
- 작업 방법: AI가 장 패널과 제비 선택을 같은 상태에 연결하고 기획이 판매 후 구조 불가/구조 후 판매 불가를 검사한다. [추정: 단일 재산의 이중 소비 방지 절차]
- 완료 기준: 재산을 한 번만 쓰며 판매, 구조, 방치가 이후 박씨 도착에서 구별된다. [근거: state.js]

- 리스크/대안: 제비 화면의 `Esc`는 단순 닫기가 아니라 지나침 확정이므로 취소로 오인할 위험을 03 UIUX에 남긴다. [근거: play.js chick 입력 분기]
- 근거: [근거: state.js; play.js]

박 — 남겨 둔 기력을 보상의 완성도로 돌려준다

- 의도(Why): 선행으로 받은 박씨를 자동 보상이 아닌 마지막 노동과 자원 배분으로 완성한다. [근거: GAME_DESIGN.md W6/W7; SOURCE_BIBLE.md R6]
- 설계(How): 박씨 도착 후 기력 1로 심고, 각 통 착수 때 기력 1, 2, 3, 4를 차감하며 하루 한 통만 켠다. [근거: config.js; state.js plant, beginSaw, finishSaw]
- 필요(What): `seedArrived`, `planted`, `sawed`, `sawedToday`, 개봉 내용 인덱스와 톱질 진행 상태가 필요하다. [근거: state.js create; play.js]
- 작업 방법: AI가 착수 판정/입력 연출/완료 지급을 나누고 기획이 같은 날 중복 착수와 기력 미달을 검사한다. [추정: 연출 도중 상태 중복 갱신 방지 절차]
- 완료 기준: 일반 모드는 좌우 교대 6회, 움직임 감소 모드는 확인키 3회로 동일한 한 통을 개봉하며 틀린 방향은 진행시키지 않는다. [근거: config.js SAW_STROKES; play.js sawInput]
- 리스크/대안: 심은 뒤 성숙 대기 없이 톱질할 수 있고 박 내용은 문안/인덱스이지 생계 자원 지급이 아니므로, 추석 전용 수확이나 보상으로 빈곤 즉시 해소를 구현 사실로 쓰지 않는다. [근거: state.js sawStatus/finishSaw; play.js sawInput]
- 근거: [근거: config.js; state.js; play.js]

종료/재시작 — 선택의 흔적을 돌려준다

- 의도(Why): 실패를 비난하는 대신 굶은 날/사용한 문/구조 결과로 다시 선택할 단서를 준다. [근거: GAME_DESIGN.md §14]
- 설계(How): 밤에는 굶주림 한도를 먼저 검사하고, 16일차는 보은 결산으로 종료하며, 12일차 진입에서 제비 미구조면 '그 해로 끝'으로 종료한다. [근거: state.js sleep]
- 필요(What): 엔딩 종류, 켠 박/굶은 날/닫힌 문/형님 방문/남은 염치의 결산과 제목 복귀가 필요하다. [근거: state.js tally; scenes.js Ending]
- 작업 방법: AI가 세 종류 종료와 제목 복귀를 연결하고 기획이 초기 자원/일일 기록/구조 상태가 새 판에서 초기화되는지 확인한다. [추정: 기록상 재시작 검사에 대응한 인수 절차]
- 완료 기준: 엔딩에서 확인키로 제목에 돌아가고 새 게임은 1일차/기력 6/염치 12/당사실 보유로 시작한다. [근거: scenes.js; state.js create]
- 리스크/대안: 최고 성과의 복합 조건은 엔딩 분기로 구현되지 않았으며 실패 조언도 과거 상태 기록 없이 생성하므로 평가 문구의 정확성을 별도 확인한다. [근거: GAME_DESIGN.md §9; scenes.js]
- 근거: [근거: state.js sleep/tally; scenes.js]

2. 구현 방법

클라이언트, 데이터, 상태 머신

- 의도(Why): 규칙 수정이 화면 전체를 흔들지 않게 하고, 원작 의도가 실제 상태 전이로 이어지는지 검증 가능하게 한다. [근거: state.js 머리말; S1 16:27]
- 설계(How): `window.HB` 아래 상수, 작업, 문, 구역을 두고 `HB.State`에서 판정/변경을 처리하며, Phaser의 제목→플레이→엔딩 씬과 플레이 내부 걷기, 장, 제비, 톱질, 막간 모드를 분리한다. [근거: config.js; state.js; main.js; play.js]
- 필요(What): `config.js`는 규칙, `state.js`는 상태, `strings.ko.js`는 문안, `hud.js`, `art.js`, `audio.js`는 표현, `play.js`/`scenes.js`는 흐름을 담당한다. [근거: build/app/src 파일 목록과 참조]
- 작업 방법: 상수→순수 상태→문안/자산→화면→검증 순으로 연결하며, `window.__HB`로 스냅숏과 규칙 호출을 제공한다. [근거: S1 구현 순서; main.js]
- 완료 기준: 화면 없이 규칙을 검사할 수 있고 화면에서도 같은 함수로 결과가 바뀐다. [근거: state.js; BUILD_BRIEF.md]
- 리스크/대안: 자동 검증 훅의 `go`/`at`는 이동 위치를 직접 바꾸므로 실제 길 찾기/충돌 검증을 대신하지 않는다. [근거: main.js 검증 훅]
- 근거: [근거: state.js; main.js; BUILD_BRIEF.md]

데이터는 별도 서버/DB 없이 고전 스크립트로 순서대로 로드하며, Phaser UMD를 동봉해 외부 요청 없는 실행을 지향한다. [근거: BUILD_BRIEF.md; bundle-single-file.mjs] 원 제품 브리프의 `file://` 불가 설명은 이후 고전 스크립트 구현/직접 열기 확인 기록으로 정정하여 교부 설명에 반영한다. [근거: PRODUCT_BRIEF.md; _progress.md 비고]

3. 시스템 간 상호작용과 밸런스 기준

계절, 노동, 문, 톱질의 교환 관계

- 의도(Why): 자력만 고집하거나 초반에 모든 도움을 소진하는 전략보다 필요한 때 문을 쓰는 판단을 보상한다. [근거: GAME_DESIGN.md §9]
- 설계(How): 봄의 나물은 기력 1로 쌀 1을 주지만 14~16일에는 사라지고, 큰 박 톱질에는 기력 3~4가 필요하므로 문에서 끼니를 구하면 노동 기력을 톱질에 쓸 수 있게 한다. [근거: config.js; GAME_DESIGN.md §9/§10]
- 필요(What): 봄 판정, 작업 비용/산출, 문 누적 비용과 박 비용, 생존 제약을 함께 계산하는 검증 경로가 필요하다. [근거: design/balance-reachability.mjs 존재; GAME_DESIGN.md]
- 작업 방법: 설계 기준을 구현 상수와 별도로 정리하고 자력, 소진, 절약, 숙련, 판매 전략의 결과를 비교한다. [근거: QA_REPORT.md §5]
- 완료 기준: 기록상 문 0, 1, 2회에 박 2, 3, 4통이라는 도달성 차이를, 해당 연산의 시작 상태/굶은 날 제약 아래 재현한다. [근거: GAME_DESIGN.md §9; QA_REPORT.md §5]
- 리스크/대안: 이 표를 임의의 자원 비축량/당사실 판매/모든 플레이 경로에 대한 수학적 상한으로 일반화하지 않는다. [추정: 제한된 연산 조건과 판매 회복 예외를 가진 구조]
- 근거: [근거: GAME_DESIGN.md §9; QA_REPORT.md §5]

밸런스의 최고 성과 목표는 4통/굶은 날 0/염치 잔존의 동시 달성이며, 구현은 이 값들을 개별 결산으로 보여주는 수준으로 정리한다. [근거: GAME_DESIGN.md §9; scenes.js Ending]

4. 게임의 나머지 부분의 방향

콘텐츠는 7개 화면 구역/7개 작업/3개 문과 제비 사건/박 4통으로 구성하며, 별도 전투, 몬스터, 보스는 해당 없음으로 둔다. [근거: config.js; PRODUCT_BRIEF.md] 새 회차의 이점은 저장 성장 대신 계절/문 사용 시점의 학습으로 두고, 박 보상과 놀부 후일담은 결산 문안으로 회수한다. [근거: GAME_DESIGN.md §6, §15, §17; scenes.js] UI는 상단 자원/대상 비용/하단 목표/반응 띠를 분리하고, 미술은 한지와 먹선, 음향은 사건별 국악 타악 계열의 합성음으로 구성한다. [근거: ART_DIRECTION.md; hud.js; audio.js] 설정은 T의 문안 100%/125% 전환과 M의 움직임 감소를 제공하며, OS의 움직임 감소 선호를 초기값으로 읽는다. [근거: hud.js; play.js; main.js] 저장, 이어하기, 계정, 클라우드는 구현하지 않으며 설정 영구 저장도 확인된 코드 범위에 없다. [근거: PRODUCT_BRIEF.md; main.js; hud.js]

5. 제작 과정 설계

검증, 교부, 인수인계

- 의도(Why): 규칙 일치, 실행 가능, 감정 전달을 서로 다른 증거로 확인하여 자동 검사의 과대 해석을 막는다. [근거: QA_REPORT.md; PLAYTEST_PROTOCOL.md]
- 설계(How): 상수 대조, 엔딩, 재시작, 화면 크기, 접근성, 성능 검사를 실행하고, 첫 이용자 이해도와 제비 선택/음향은 별도 사람 관찰로 남긴다. [근거: BUILD_BRIEF.md 검증 목록; PLAYTEST_PROTOCOL.md]
- 필요(What): 게임 파일, 검증 명령, 환경/결과 JSON, 장면 증거, 미수행 목록, 단일 파일 포장 스크립트가 필요하다. [근거: BUILD_BRIEF.md; QA_REPORT.md; bundle-single-file.mjs]
- 작업 방법: 규칙 확정 뒤 화면, 문안, 절차 자산은 인터페이스를 공유해 병행 가능하도록 나누고, 최종 통합 후 검증/인계 문서를 작성한다. [추정: 파일 책임에 근거한 재구성 작업 순서이며 당시 병행 인력 기록은 없음]
- 완료 기준: 기록상 12개 검사 묶음/64개 검사를 대조하되 미수행 항목은 남기며, 폴더판과 단일 파일판을 구분해 전달한다. [근거: QA_REPORT.md §2; S1 단일 파일 요청]
- 리스크/대안: 오래된 결과를 현재 실행 결과로 발표하지 않고, 이름 변경/재포장 뒤 동일성은 별도 확인 대상으로 둔다. [추정: QA 기록과 교부본의 시점/부제 차이]
- 근거: [근거: QA_REPORT.md; _progress.md; S1]

담당별 실제 인원 수/일정 일수/외주 비용은 기록 없음으로 두며, 소스와 문서의 상수가 달라지면 구현 수정인지 설계 변경인지 먼저 판정한다. [근거: GAME_DESIGN.md §9; 제작 인력 상세 기록 부재]

6. 세부기획서 목록

문서	확정할 범위
`03_세부기획서_콘텐츠와서사.md`	3막 진행/구역 구조/제비 사건, 박, 엔딩과 상태 전이 예외를 정한다. [근거: GAME_DESIGN.md; config.js; state.js]
`03_세부기획서_성장경제와데이터.md`	노동, 급식, 문, 재산, 톱질 수치와 상태 필드, 검증 조건을 정한다. [근거: config.js; state.js]
`03_세부기획서_UIUX와아트사운드.md`	입력/정보 배치, 반응, 접근성, 절차 자산, 음향, 교부 검수를 정한다. [근거: ART_DIRECTION.md; play.js; hud.js; audio.js]