치킨게임 — 세부기획서 (데이터, 수치, 밸런스)

> 재구성 기획서 (역설계) / 작성 기준 시점: 2026-08-07 (프로젝트 시작 시점) / 문서 버전: Ver 0.1 프로젝트 ID: P010 / 근거 수준: A / 작성: Claude (재구성) 이 문서는 결과물과 제작 기록으로부터 기획 의도를 역추정해 당시 작성한 것처럼 재구성한 것이다. 확인된 사실과 추정을 [근거]/[추정] 태그로 구분한다.

1. 목적과 범위

- 목적: 아이템 마스터, 상수, 경제/주문 수치, 밸런스 검증 기준과 난이도 조정 이력을 한곳에 정리해 수치가 문서, 코드, UI에서 어긋나지 않게 한다.
- 범위: 아이템 18종 + 가공품 3종, 시작 상태, 고정비, 위험 확률, 밸런스 시나리오, 테스트 매핑. 시스템 규칙은 `02_메인기획서.md`, 화면은 `03_세부기획서_ui-ux.md`.
- 원칙: 수치의 단일 원천은 `src/sim/constants.ts`이고 기획서 표는 그 초기값의 정본이다. 새 수치는 [제안값]으로 표시한다. [근거: 치킨게임_기획서.md 머리말, `src/sim/constants.ts`]

2. 설계

2.1 아이템 마스터

재료 티어와 위험 재료

- 의도(Why): 싼 재료로 시작해 비싼 재료로 갈아타는 성장 압력과, 위험하지만 싸고 강한 재료의 유혹을 같은 표에서 만든다.
- 설계(How): 4개 카테고리에 각각 표준, 상위, 위험 티어를 둔다. 밀가루는 맛 가산(전분 10, 박력분 14, 보약가루 26, 시멘트 4)이고 기름은 맛 배율(1.00/1.15/1.60/0.45)이다. 조미료는 맛 가산 5종(소금후추 5, 시즈닝 8, 인산염 12, 미원 15, 마법의 가루 32). 닭은 생닭 30, 냉동닭 17, 비둘기 6. 위험 재료는 가격 0(획득형)이거나 판매 리스크 태그를 가진다. 손질은 서로 다른 조미료 2종이 필요하다.
- 필요(What): `ItemDef` 필드(id, name, category, desc, price, shelfLifeMinutes, fridgeAffected, taste, tasteMult, riskTag, stackable).
- 작업 방법: 원안 수치를 기획서 표로 고정 → `items.ts` 전사(AI) → 빌드별 맛 테스트.
- 완료 기준: 21종이 모두 `ItemDef`로 존재하고 아이콘이 있다.
- 리스크/대안: 원안에 소금/후추 수치가 없어 에이전트가 정하지 않고 사용자에게 질문했다(1종 통합 5/400원 선택). 같은 이유로 원안에 없는 수치는 임의 확정하지 않는다.
- 근거: [근거: 치킨게임_기획서.md §5, 치킨게임.txt, claude_d88b9294-e9a5-4cb2-ad98-a3457ec5a033.txt]

2.2 시작 상태

3일치 시작 재고

- 의도(Why): 발주 리드타임 2일을 플레이어가 몸으로 배우게 하되, 첫 이틀에 게임이 끝나지 않게 한다.

- 설계(How): 시작 자금 5,000,000원, 평판 20, 위생 1.0, 집 보유, 가격 20,000원, 확장 0. 냉장고에 냉동닭 45(45칸), 선반에 전분 45/대두유 45/시즈닝 45/소금후추 45(각 3칸), 락스 5/청소용품 5. 원가 합계 약 288,000원으로 자금 대비 미미하다. 생존형 빌드(냉동닭 + 시즈닝 + 소금후추 → 숙성 → 전분 + 대두유)로 Day 1 맛 55.0이며, 숙성을 기다리지 않아도 40.0으로 성공한다.
- 필요(What): `START_STOCK_FRIDGE`, `START_STOCK_SHELF`.
- 작업 방법: 상수 정의 → 회귀 테스트(§17-19, §17-20).
- 완료 기준: Day 3 입고 전까지 재료가 고갈되지 않고 시작 재고만으로 손질이 성립한다.
- 리스크/대안: 손질 없이 튀기면 맛 27.0으로 항상 실패하며 이는 튜토리얼 압력으로 의도됐다.
- 근거: [근거: 치킨게임_기획서.md §11.1, §17-19/§17-20, `src/sim/constants.ts`]

2.3 주문/평판 수치

제한시간과 난이도 곡선

- 의도(Why): 압박의 원천이 "타이머 초과로 인한 몰수"가 아니라 "늦으면 맛이 떨어짐"이 되게 해 실패에 납득이 생기게 한다.
- 설계(How): 제한시간은 원안 수주 3분/전달 15분 → v1.1에서 3분/20분 → v1.2에서 9분/60분으로 변천했다. 이유는 15분이 최속 경로 16분보다 짧았고, 20분은 여유가 4분뿐이라 선행 조리가 강제됐고, 사용자가 "너무 어렵다"고 보고했기 때문이다. 전달 60분을 꽉 채우면 맛이 정확히 절반이 되어 기준 미달로 끝난다. 요구 맛 `base = 15 + rep × 0.85`, 지터 ±10%. 평판 보정: 성공 +0.5(인산염 +0.2, 마법의 가루 +0.5 추가), 실패 -1, 상함 -10, 가격 인상 -10/인하 +5, 클레임 -10, 입소문 +5, 영업정지 -20, 미납 -5. 완벽한 하루(성공률 90%↑, 주문 10건↑) +2, 최악의 하루(30%↓) -3.
- 필요(What): `ORDER_ACCEPT_LIMIT`, `ORDER_SERVE_LIMIT`, `REP_*`, `TASTE_REQ_*`.
- 작업 방법: 상수 수정 → 테스트를 상수 기준으로 작성(회귀 2개 추가) → 튜토리얼 문구/도해가 상수를 참조하는지 확인 → 기획서 §10.5, §17-28, 부록 A/변경 이력 갱신.
- 완료 기준: 75개 테스트 통과, 도해와 문구에 9분/60분이 자동 반영.
- 리스크/대안: 더 쉽게 하려면 맛 감쇠 반감기(`TASTE_DECAY_HALFLIFE = 60`)를 늘린다(다음 조절 손잡이).
- 근거: [근거: `src/sim/constants.ts`, 치킨게임_기획서.md §10.5/§17-28, claude_d88b9294-e9a5-4cb2-ad98-a3457ec5a033.txt]

2.4 경제 수치

고정비와 투자 곡선

- 의도(Why): 매달 고정비 압박과 확장이라는 장기 목표 사이의 간격을 크게 벌려, 평판을 올려야만 성장하게 한다.
- 설계(How): 월세 1,000,000(기본) + 500,000(확장당) + 500,000(집 보유), 생활비 200,000(10일), 식비 10,000원/끼. 일 순익은 14마리 기준 약 190,000원이고 30일 고정비(월세 1,500,000 + 생활비 200,000 + 식비 600,000 = 2,300,000) 차감 후 약 +3,400,000원이다. 확장비는 20M, 40M, 80M으로 단계 간 2배. 배송비 30,000원(500,000원 이상 무료), 배송 2일. 가격은 5,000원 단위, 평판 영향은 변경 시 1회성.
- 필요(What): 상수 표, 손익 시뮬레이션 테스트.

- 작업 방법: `balance.spec.ts`의 "고정비와 손익"으로 30일 후 자금을 검증한다.
- 완료 기준: Day 1 수준 유지는 흑자이나 확장비에는 도달하지 못한다.
- 리스크/대안: 장기 시뮬레이션은 에이전트 플레이 가정 기반이라 사람 플레이와의 차이는 미확인.
- 근거: [근거: 치킨게임_기획서.md §11, §18.1, `test/balance.spec.ts`]

2.5 세이브 데이터

스키마와 마이그레이션 방침

- 의도(Why): 같은 시드/같은 입력의 재현성을 세이브 복원에서도 유지한다.
- 설계(How): `{schema: 1, savedAtRealTime, slot, state}`를 `localStorage`의 `chickengame.save.<slot>`에 저장한다(슬롯 1~3, quicksave, autosave). `rngSeed`를 상태에 포함한다. 오토세이브는 매일 출근 시 1회. 스키마 불일치 시 v1은 로드를 거부하고 안내한다. 저장 접근 실패는 try/catch로 삼킨다.
- 필요(What): `serialize`, `save`, `load`, `clearSlot`, `slotInfo`.
- 작업 방법: 세이브 왕복 테스트, `reduce`가 원본을 변형하지 않는지 테스트.
- 완료 기준: 왕복 후 상태 동일. 진행 중 행동(busy) 포함 복원은 M5 기준으로 남아 있고 실구현 확인은 미확인.
- 리스크/대안: 키 이름 변경은 기존 세이브를 잃게 하므로 접두사 `chickengame.`를 유지한다.
- 근거: [근거: 치킨게임_기획서.md §15.4, `src/save/save.ts`, `test/decisions.spec.ts`]

3. 데이터/리소스 명세

3.1 아이템 마스터

id	이름	카테고리	가격	유통기한	냉장영향	맛 / 배율	리스크	스택
flour_starch	전분	flour	1,000	525,600분	아니오	10	-	예
flour_cake	박력분	flour	1,500	525,600분	아니오	14	-	예
flour_tonic	보약가루	flour	8,000	525,600분	아니오	26	-	예
flour_cement	시멘트	flour	0(획득)	525,600분	아니오	4	cement	예
oil_soy	대두유	oil	2,000	262,800분	아니오	×1.00	-	예
oil_sunflower	해바라기유	oil	3,200	262,800분	아니오	×1.15	-	예
oil_jet	제트항공유	oil	15,000	262,800분	아니오	×1.60	-	예
oil_grease	구리스	oil	0(획득)	262,800분	아니오	×0.45	grease	예
season_saltpepper	소금후추	seasoning	400	무기한	아니오	5	-	예
season_basic	시즈닝	seasoning	500	무기한	아니오	8	-	예
season_msg	미원	seasoning	900	무기한	아니오	15	-	예
season_phos	인산염	seasoning	1,400	무기한	아니오	12	phosphate	예
season_magic	마법의 가루(빨강)	seasoning	5,500	무기한	아니오	32	magic	예

id	이름	카테고리	가격	유통기한	냉장영향	맛 / 배율	리스크	스택
chick_fresh	생닭	chicken	4,200	240분	예	30	-	아니오
chick_frozen	냉동닭	chicken	2,500	2,880분	예	17	-	아니오
chick_pigeon	비둘기	chicken	0(포획)	240분	예	6	pigeon	아니오
clean_bleach	락스	hygiene	3,000	무기한	아니오	-	-	예
clean_tools	청소용품	hygiene	2,000	무기한	아니오	-	-	예
prepped_chicken / aged_chicken / fried_chicken	손질된/ 숙성된 닭 / 치킨	wip/product	제작	닭 승계 / 치킨은 감쇠	가공품은 닭 규칙 / 치킨 아니오	합산 / ×1.5 / 감쇠	승계	아니오

[근거: 치킨게임_기획서.md §5.2–5.7]

3.2 핵심 상수(발췌, `constants.ts`)

상수	값	의미
`MS_PER_TICK`	500	1분 = 500ms
`STACK_MAX`	20	스택 상한
`FRIDGE_DECAY_RATE`	0.25	냉장 신선도 소모 배율
`FRESHNESS_BUCKET`	10	스택 병합 신선도 버킷(분)
`AGING_MINUTES` / `AGING_MULT`	60 / 1.5	숙성
`OVERAGE_MINUTES` / `OVERAGE_MULT`	240 / 0.8	과숙성
`TASTE_DECAY_HALFLIFE`	60	치킨 맛 반감기
`T_PREP`, `T_FRY`, `T_CLEAN`	6, 12, 10	행동 시간(분)
`S_PREP`, `S_FRY`, `S_CLEAN`	2.0, 3.0, 3.0	행동 기력
`FILTH_DAILY`	0.25	일일 위생 자연 증가
`CLEAN_REDUCTION`	0.1	청소 1회 위생 감소(조정 후보 0.10~0.25)
`ORDER_ACCEPT_LIMIT` / `ORDER_SERVE_LIMIT`	9 / 60	수주/전달 제한(분)
`ORDER_SLOTS_BASE` / `ORDER_SLOTS_PER_EXP`	3 / 2	동시 수주 슬롯
`START_MONEY` / `START_REP`	5,000,000 / 20	시작 자금/평판
`EXPANSION_COSTS`	20M / 40M / 80M	확장
`PIGEON_SUCCESS_RATE` / `PIGEON_MAX_PER_DAY`	0.6 / 4	비둘기 포획
`PHOSPHATE_BUST_PER_SALE` / `MAGIC_BUST_PER_SALE`	0.005 / 0.012	영업정지(원안 0.05 / 0.10)
`LOG_MAX`	200	로그 상한

3.3 빌드 밸런스 표

빌드	구성	원가	즉시 맛	전건 통과 평판
생존형	냉동닭+시즈닝+소금후추 → 숙성 → 전분+대두유	6,400	55.0	약 40
표준형	생닭+시즈닝+미원 → 숙성 → 박력분+해바라기유	10,300	107.5	약 80

빌드	구성	원가	즉시 맛	전건 통과 평판
고급형	생닭+미원+마법가루 → 숙성 → 보약가루+제트항공유	33,600	226.4	100 여유
범죄형	비둘기+미원+마법가루 → 숙성 → 시멘트+구리스	6,400	37.6	약 20 이하

[근거: 치킨게임_기획서.md §8.5, §18.6] 범죄형의 통과 평판은 맛 37.6과 기준식 `15 + rep×0.85`에서 계산한 값이다. [추정: 기획서에 명시 없고 공식에서 역산]

3.4 테스트 매핑

테스트 파일	대상
`test/sim.spec.ts`	시간, 결정론, 유통기한, 숙성, 맛 공식, 감쇠, 주문, 고정비 부과일, 페이즈, 손님 수/시작 재고(M1)
`test/decisions.spec.ts`	결정 §17-19~23, 26, 1, 2, 상한 재료 전파, 세이브 왕복, 리듀서 불변성
`test/balance.spec.ts`	Day 1 손님 수, 빌드별 원가/맛, 처리량 상한, 위생 타임라인, 평판 성장, 리스크 확률 실측, 고정비/손익, 장기 안정성
`test/playthrough.spec.ts`	Day 1 완주, 30일 장기 시뮬레이션
세션 보고	75개 통과(2026-08-08 13:22)

4. 작업 순서

1. 원안 수치를 기획서 표로 확정하고 원안과 다른 값은 §17에 이유와 함께 기록한다.
2. 표를 `constants.ts`/`items.ts`에 전사한다.
3. 수식(`formulas.ts`)과 빌드별 맛 테스트.
4. 밸런스 시나리오를 테스트로 실측해 §18 표와 대조, 어긋나면 상수를 조정하고 기획서에 반영.
5. 사용자 플레이 피드백으로 난이도 조정(제한시간 3배) 후 문서/UI 문구/테스트를 같은 날 동기화한다.

5. 검수 기준

- 기획서 표와 `constants.ts`/`items.ts`가 일치한다(수치 전사 검사).
- 빌드별 맛 값이 소수 1자리로 재현된다(55.0 / 107.5 / 226.4 / 37.6).
- Day 1 손님 17~23명, 평판 0 회생 가능, 청소 0회 시 Day 5 위생 5.0.
- 수치 변경 시 튜토리얼, 도해, 기획서가 함께 갱신된다.
- 미해결: 기획서 본문의 구버전 수치(§1.2 "3분", §14.6.1 "3분/20분", §20 M1 "3분") 정리, 패키지 버전(1.1.0)과 기획서 버전(1.2) 정합.

치킨게임 — 세부기획서 (UI, UX, 아트 리소스)

> 재구성 기획서 (역설계) / 작성 기준 시점: 2026-08-07 (프로젝트 시작 시점) / 문서 버전: Ver 0.1 프로젝트 ID: P010 / 근거 수준: A / 작성: Claude (재구성) 이 문서는 결과물과 제작 기록으로부터 기획 의도를 역추정해 당시 작성한 것처럼 재구성한 것이다. 확인된 사실과 추정을 [근거]/[추정] 태그로 구분한다.

1. 목적과 범위

- 목적: 4화면 순환 구조에서 시간 압박 속에서도 조작과 정보 읽기가 부담이 되지 않도록 UI/UX와 아트 리소스를 정한다.
- 범위: 전역 레이아웃, 화면 4종, 키맵, 전달 확인, 가시성 보강, 튜토리얼, 타이틀, 토스트, 애니메이션, 그래픽 리소스. 수치/데이터는 `03_세부기획서_data-balance.md`.
- 제외: 사운드(기록 없음).
- 근거: [근거: 치킨게임_기획서.md §14, claude_d88b9294-e9a5-4cb2-ad98-a3457ec5a033.txt]

2. 설계

2.1 전역 레이아웃

레이아웃과 정보 배치

- 의도(Why): 판단에 필요한 정보(자원, 시간, 주문)를 어느 화면에서도 같은 위치에서 읽게 해 시선 이동 비용을 줄인다.
- 설계(How): 상단 HUD(Day, 요일, 시각, 배속, 돈, 생명력 N/100/기력 N/100, 평판, 위생, 숙련, 치킨값), 중앙 현재 화면, 우측 주문 대기 패널(요구 맛, 남은 시간 바)과 로그, 하단 인벤토리 3칸과 진행 중 행동 바. 최소 해상도 1280×720. 위생도는 구간 라벨(청결/보통/지저분/위험/폐쇄 임박)과 색을 병기해 색맹을 고려한다.
- 필요(What): `Hud`, `OrderPanel`, `Inventory`, `Stage` 컴포넌트.
- 작업 방법: 와이어 구성(기획) → 컴포넌트 구현(AI) → 브라우저에서 1280×720 스크롤 유무 확인.
- 완료 기준: 720px 높이에서 스크롤 없이 모든 패널이 보인다.
- 리스크/대안: 이모지 아이콘은 환경별 렌더가 달라 쓰지 않는다.
- 근거: [근거: 치킨게임_기획서.md §14.1, §9.5, §14.4]

2.2 화면별 설계

계산대(COUNTER)

- 의도(Why): 주문 수주, 전달, 경영 결정이 모이는 허브로, 손님을 눈에 보이게 해 시간 압박을 체감시킨다.
- 설계(How): 좌측 POS(수주, 발주, 가격표, 부동산, 확장, 뒷골목, 공사장, DOCK, 퇴근), 우측 전달대(스케치의 "치킨납품하는곳"). 두 지점 간 이동은 시간/기력이 들지 않는다. 전달 시 확인 다이얼로그에 요구 맛, 선택 치킨의 현재 맛, 성공/실패 예상, 상함/위험 태그 경고를 모두 표시한다. 줄 서기는 `orders`에서 파생: 최대 6명 표시, 초과분 `+N` 배지, 말풍선에 요구 맛과 타이머 바, 30초 이하일 때 붉은 점멸(기획서 문구는 3분 기준이었고 이후 9분으로 변경됨). 퇴근은 CLOSING에서만 활성. DOCK 입고 창은 자동으로 열지 않는다.
- 필요(What): `GuestQueue`, `ServeDialog`, `CounterDialogs`, 손님 SVG 9종.
- 작업 방법: 스케치의 항목을 기획서 항목에 대응(기획) → 대응이 없는 전달대는 SERVE 지점으로 확정 → 구현/확인.
- 완료 기준: 전달 확정 전 위 5개 정보가 모두 보인다.

- 리스크/대안: 전달대를 거치대로 만들면 3칸 병목이 무너져 단순 지점으로 했다.
- 근거: [근거: 치킨게임_기획서.md § 14.2.1, § 14.6.1, § 17-24, claude_d88b9294-e9a5-4cb2-ad98-a3457ec5a033.txt]

주방(KITCHEN)

- 의도(Why): 제작 행동의 시각적 보상과 클릭 가능한 직관성을 준다.
- 설계(How): 좌측 튀김기, 우측 도마. 튀김기 클릭 → 닭 튀기기 다이얼로그, 도마 클릭 → 닭 손질 다이얼로그(호버 시 점선 테두리와 튀기기/손질하기 라벨). 하단 버튼과 R/F 키도 유지해 키보드 완주를 보존한다. 실행 전 예상 맛을 표시(예: 손질 30.0 → 숙성 시 45.0). 진행 중 설비 위에 진행 바와 Esc 취소 힌트. 손질은 칼의 상하 운동 주기 400ms와 3단계 형태 변화, 튀기기는 기름 거품, 김, 튀김옷 색 보간(연노랑→황금)/완료 시 건지는 동작.
- 필요(What): `KitchenDialog`, 핫스팟 CSS, 조리 SVG.
- 작업 방법: 클릭 영역 추가(사용자 요청) → 정지 상태에서 도마/튀김기 클릭으로 다이얼로그가 정확히 열리는지 브라우저 확인.
- 완료 기준: 두 설비 클릭이 각각의 메뉴를 열고 기존 키 조작이 유지된다.
- 리스크/대안: 숙련도 가정 오차로 예상 맛이 어긋나지 않도록 실행 직전 숙련도로 계산한다.
- 근거: [근거: 치킨게임_기획서.md § 14.2.2, § 14.6.2, claude_d88b9294-e9a5-4cb2-ad98-a3457ec5a033.txt]

선반, 냉장고(SHELF, FRIDGE)

- 의도(Why): 보관량을 숫자 없이 느끼게 하고, 많은 칸에서 품목을 빠르게 구분하게 한다. 사용자가 "가시성이 너무 떨어진다"고 지적한 지점이다.
- 설계(How): 진입 즉시 아이템 칸이 자동으로 열린다(Enter 불필요, 계산대의 입고 창만 수동). 칸을 104×104로 키우고 아이콘 아래에 이름(긴 이름은 2줄), 그 아래에 남은 유통기한 또는 맛 값을 표시. 빈 칸은 점선/옅은 배경으로 눌러 채워진 칸이 먼저 보인다. 카테고리별 테두리와 상단 띠 색: 닭 붉은 살색, 손질/숙성 갈색, 치킨 황금, 밀가루 겨자, 기름 청색, 조미료 보라, 위생용품 초록. 창 위 범례는 실제 보유 카테고리만 표시. 인벤토리 3칸에도 같은 색. 자동 열림 시 배경 불투명도를 0.55→0.3으로 낮추고 칸을 아래에 붙여 뒤의 적재 일러스트가 비치게 한다(Esc로 닫으면 전체 그림). 선반 3단, 냉장고 4단에 카테고리별 더미를 그리고 스택 품목은 `ceil(개수/5)`개(최대 8), 닭류는 최대 12마리까지 그린다. 상함 더미는 붉은 테두리와 [상함] 뱃지. 증감은 200ms 페이드+스케일. 냉장 품목은 ❄×4 배지.
- 필요(What): `StorageGrid`, `ItemIcon` 21종, 카테고리 색 매핑.
- 작업 방법: 색 정의 → 그리드에 이름, 색, 범례 → 인벤토리 → 자동 열림 → 브라우저에서 범례/색 확인.
- 완료 기준: 냉장고 범례에는 닭 1종만, 선반 범례에는 밀가루, 기름, 조미료, 위생용품 4종이 나온다.
- 리스크/대안: 색에만 의존하지 않도록 이름을 항상 병기한다.
- 근거: [근거: 치킨게임_기획서.md § 14.2.3–14.2.4, § 14.6.3, claude_d88b9294-e9a5-4cb2-ad98-a3457ec5a033.txt]

2.3 조작

키맵과 포인터 조작

- 의도(Why): 시간 압박 게임에서 조작 실수가 곧 실패다. 키보드만으로 완주할 수 있게 하되, 마우스로도 같은 일을 하게 한다.
- 설계(How): ←→ 화면 이동(냉장고→계산대→주방→선반 순환), Enter 저장소 열기/확정, W A S D 커서, E 집기/Q 놓기, 1 2 3 인벤토리 선택, R 손질/F 튀기기, C 청소, Esc 취소, Space 일시정지, Ctrl+1/2/3 배속, F5/F9 퀵세이브/로드, Tab 주문 패널 포커스. 정지 상태에서 모든 조작이 가능하다.
- 필요(What): useKeymap 훅.
- 작업 방법: 키 목록 작성 → 훅 구현 → 하루 완주 확인.
- 완료 기준: 키보드만으로 Day 1을 완주한다.
- 리스크/대안: 같은 1/2/3이 인벤토리 선택과 배속에 겹치므로 배속은 Ctrl 조합으로 분리한다.
- 근거: [근거: 치킨게임_기획서.md § 14.3, 세션 08-08 키 조작 표]

2.4 온보딩

튜토리얼 10장

- 의도(Why): 수치가 많은 게임이라 설명 없이는 첫 화면에서 막힌다. 사용자가 "여러 수치 설명과 직관적 플레이 설명, 이미지 포함, 박스 확대"를 요구했다.
- 설계(How): 1000px 폭, 높이 543~590px의 대형 모달(1280×720에서 스크롤 없음). 구성은 제목 → 한 줄 요약 → 도해(946×280) → 요점 3~4개 → 진행 점. 10장: 목표와 월세 / 여섯 자원 읽는 법 / 하루 일과 / 네 화면 순환 / 인벤토리 3칸 제약 / 손질→숙성(17+8+5=30→×1.5=45) / 튀기기 공식과 기름 4종 배율표 / 맛 감쇠 곡선(55→27.5→13.8→6.9) / 주문 흐름 / 위생, 기력, 생명력. 수치는 constants.ts를 직접 참조한다. 시작 후 "설명서" 버튼으로 다시 열 수 있다.
- 필요(What): Tutorial, TutorialArt, HelpButton.
- 작업 방법: 도해 일러스트 → 본체 → 큰 모달 스타일 → 설명서 버튼 → 10장 전부 넘겨 잘림 확인(AI).
- 완료 기준: 10장 모두 스크롤 0건, 도해마다 라벨 6~23개.
- 리스크/대안: 수치를 문구에 하드코딩하면 난이도 조정 시 어긋나므로 상수를 참조한다.
- 근거: [근거: claude_d88b9294-e9a5-4cb2-ad98-a3457ec5a033.txt, 기획서 부록 B.3]

타이틀 화면

- 의도(Why): 시작과 이어하기를 분리하고, 시간이 흐르기 전에 게임을 시작할 준비를 한다.
- 설계(How): 밤의 가게 앞 배경에 창문 하나만 켜진 그림, 닭다리 로고와 치킨게임 / CHICKEN GAME, 태그라인 "몸이 부서지기 전에 월세를 내라.", 깜빡이는 Click to Start(아무 곳 클릭, Enter, Space), 세이브가 있을 때만 이어서 하기 — Day N, 별도 새로 시작. 새로 시작은 확인 1회("저장된 Day N 기록이 지워집니다. 되돌릴 수 없습니다.") 후 자동, 퀵, 슬롯 1~3/튜토리얼 완료 표시를 삭제하고 튜토리얼 1/10부터 시작한다. 타이틀에서는 시계를 막는다.
- 필요(What): TitleScreen, 스토어의 started 플래그, 클록 가드.
- 작업 방법: 컴포넌트 → 스토어 시작 상태 → 클록 차단 → 게임 오버 후 새로 시작도 같은 흐름에 연결 → 검증(시간 정지 1.4초, 취소 시 보존, 삭제 후 1/10).
- 완료 기준: 요구 3가지(시간 정지, 이어하기, 새로 시작 시 세이브 삭제/튜토리얼 재시작)를 브라우저에서 통과한다.

- 리스크/대안: 사용자는 "HTML만 수정"을 제안했지만 압축본 직접 수정이 위험해 소스 수정 후 재빌드를 택했다.
- 근거: [근거: claude_d88b9294-e9a5-4cb2-ad98-a3457ec5a033.txt]

2.5 피드백

토스트/로그

- 의도(Why): 알림이 화면에 남아 있으면 가려진다. 사라져도 기록은 남겨 놓친 정보를 복구한다.
- 설계(How): 토스트는 3.2초 선명 → 0.4초 페이드 → 3.8초에 DOM 제거. 클릭하면 즉시 닫힌다. 새 알림이 오면 타이머를 새로 시작하고, `clearToast(id)`는 현재 알림이 그 id일 때만 지운다. 모든 이벤트는 우측 로그 패널에 최대 200개까지 남는다(`LOG_MAX`). 심각도는 info/good/warn/danger.
- 필요(What): `Toast` 컴포넌트, 로그 상태.
- 작업 방법: 소멸 코드 부재 확인 → 타이머/페이드 → 연속 알림에서 타이머 간섭 검증.
- 완료 기준: opacity 1 → 0.24 → 제거가 시간대로 관찰되고 연속 알림이 서로 지우지 않는다.
- 리스크/대안: 브라우저 패널이 숨겨지면 타이머가 스로틀링되므로 시간 검증은 보이는 탭에서 한다.
- 근거: [근거: claude_d88b9294-e9a5-4cb2-ad98-a3457ec5a033.txt, `src/sim/constants.ts`]

2.6 아트 리소스

컬러 플랫 SVG 리소스

- 의도(Why): 외부 에셋 없이 단일 HTML로 배포하고, 상태가 그림에 드러나게 한다.
- 설계(How): 스타일 규칙: 면색 + 1단계 그림자, 외곽선은 동계열 얇게 또는 없음, 그라디언트/사실적 음영/텍스처 금지. 팔레트는 벽/바닥 `#F5EFE3`/`#E3D8C5`, 스테인리스 `#B8C4CC`/`#8A99A3`, 목재 `#D6A461`/`#A87840`, 튀김유/치킨 `#E8A33D`/`#C47F22`, 냉장고 `#7FB8B0`/`#4E8A82`, 강조 `#D9534F`, 선/텍스트 `#2B2622`(기획서 값, `art/palette.ts`에 집약). 상태 연동: 위생도에 비례한 얼룩 레이어, 탈진 시 채도 저하와 비네트, 영업정지 시 셔터. 접근성: `prefers-reduced-motion`에서는 이동/바운스를 끄고 상태 전환만 즉시 반영, 애니메이션은 transform/opacity만 사용한다.
- 필요(What): 배경 4종, 아이콘 21종, 손님 9종, 튜토리얼 도해.
- 작업 방법: 팔레트 → 배경 → 아이콘 → 손님 → 도해. 스케치는 배치 참고용으로만 쓰고 형태는 새로 그린다.
- 완료 기준: 외부 참조 0개. 단일 HTML이 `file://`에서 콘솔 에러 없이 동작한다.
- 리스크/대안: 이미지를 쓰면 단일 파일 배포가 불가능해 SVG 절차 생성을 택했다.
- 근거: [근거: 치킨게임_기획서.md § 14.5, claude_d88b9294-e9a5-4cb2-ad98-a3457ec5a033.txt]

3. 데이터/리소스 명세

3.1 UI 리소스

리소스	수량	위치	비고
화면 배경 SVG	4(계산대, 주방, 선반, 냉장고)	`src/ui/art/Scenery.tsx`	지배색 4종
아이템 아이콘	21	`ItemIcon.tsx`	재료 18 + 가공품 3

리소스	수량	위치	비고
손님 SVG	9(색3 × 실루엣3)	`Guest.tsx`	`order.id` 해시 결정
튜토리얼 도해	10장	`TutorialArt.tsx`	946×280
팔레트	1	`palette.ts`	7개 용도

3.2 카테고리 색

카테고리	테두리/띠 색	대상
chicken	붉은 살색	생닭, 냉동닭, 비둘기
wip	갈색	손질된 닭/숙성된 닭
product	황금	치킨
flour	겨자	전분 등 4종
oil	청색	4종
seasoning	보라	5종
hygiene	초록	2종

3.3 UI 상수(관찰값)

필드	값	근거
토스트 유지	3.2초 + 페이드 0.4초	claude_d88b9294…
저장소 칸 크기	104×104 px	claude_d88b9294…
튜토리얼 모달	1000px 폭, 543~590px 높이	claude_d88b9294…
줄 서기 표시 상한	6명	기획서 §14.6.1
로그 상한	200	`constants.ts`

4. 작업 순서

1. 레이아웃, HUD, 인벤토리, 저장소 그리드(M2).
2. 계산대 줄/전달 다이얼로그(M3).
3. 발주, 가격, 부동산 등 메뉴(M4).
4. 정산/게임 오버 화면(M5).
5. 아트/애니메이션(M6).
6. 사용성 보강: 토스트 → 튜토리얼 → 타이틀 → 저장소 가시성 → 주방 클릭 → 일시정지 정책.
7. 단일 HTML 빌드와 `file://` 확인.

5. 검수 기준

- 콘솔 에러 0건, 1280×720에서 스크롤 없음.
- 키보드만으로 Day 1 완주.
- 전달 다이얼로그에 요구 맛/현재 맛/예상 결과/리스크가 모두 표시된다.
- 선반/냉장고 진입 시 칸이 즉시 열리고 이름, 색, 범례가 맞다.

- 타이틀에서 시간이 흐르지 않고 새로 시작이 세이브를 전부 지운다.
- 튜토리얼 10장이 잘림 없이 보이고 수치가 상수와 일치한다.
- 구조적 점검 필요: 기획서 §14.6.1에 남은 구버전 타이머 문구(3분/20분)가 현행(9분/60분)과 맞는지 확인한다.