뒤로가는 RPG

JAMAIS VU

메인 기획서

20261008 / 박도원



목차

개요 – 2P

1. 목차
2. 로그
3. 문서 계층

구현 전제 – 5P

1. 현행 개정의 출발점
2. 컨셉의 수용 기준

핵심 메카닉스 – 7P

1. 상태의 연결
2. 전투 진입 계약
3. 라운드 처리 순서
4. 전술 규칙과 목적
5. 시간 능력의 경계

구현 구조 – 12P

1. 규칙과 화면의 분리
2. 화면 요청의 소비

구현 계획 – 14P

1. 필요한 제작 단위
2. 구현 게이트

리소스 – 16P

1. 아트 제작 규모
2. 생산과 납품 계약

개요 – 로그

순번	작성일	작성자	작성 내용	비고
1	2026-10-03	박도원	현행 통합 명세 착수	기존 코어와 새 서사의 연결 범위를 정함
2	2026-10-07	박도원	1부 연결 명세 보강	전투, 필드, 캠프 인계 조건을 반영함
3	2026-10-08	박도원	2부 예외 명세	엔딩/시간 장치/패배 결투를 반영함
4				
5				
6				

개요 – 문서 계층



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

구현 전제 – 현행 개정의 출발점

재사용 규칙

격자/판정 코어를 현행 서사에 맞춰 연결함

의도 기존 코어를 보존하면서 개정 연결부를 새로 설계함

근거 2026-10-03 StoryBible 개편 및 이전 코어 기록

새 구현 범위

장, 동료, 캠프와 시간 능력의 연결을 설계함

의도 기존 코어를 보존하면서 개정 연결부를 새로 설계함

근거 2026-10-03 StoryBible 개편 및 이전 코어 기록

정본 분리

GDD의 전투 규칙과 Story의 서사 기준을 구분함

의도 기존 코어를 보존하면서 개정 연결부를 새로 설계함

근거 2026-10-03 StoryBible 개편 및 이전 코어 기록

구현 전제 – 컨셉의 수용 기준



탐험, 전투, 정비의 선택을 같은 상태 데이터로 연결해야 함

0건

동일 입력의 결과 차이

9장

1차 연결 범위

3갈래

최종 결말 검증 범위

의도 상위 기획의 약속을 검수 가능한 기준으로 고정함

근거 제작기록 §4 및 Story_Overview_v1/Part2_Story_Summary_v1

핵심 메카닉스 – 상태의 연결



핵심 메카닉스 – 전투 진입 계약

1

접촉/트리거

맵 변형과 전투 ID를 결정함

2

시작점

맵별 1~4곳 중 선택하게 함

3

배치

기본 6명/맵별 4~8명으로 제한함

4

실행

목표와 패배 조건을 표시함

핵심 메카닉스 – 라운드 처리 순서

1

환경

타일/침수 효과를 적용함

2

순서

속도와 추가 기회를 계산함

3

턴

이동 뒤 행동 하나를 선택함

4

종료

지속 효과/쿨타임을 갱신함

핵심 메카닉스 – 전술 규칙과 목적

규칙	기준	플레이어 판단
거리/높이	맨해튼 / 높이 0~3	거리와 우회 비용을 읽게 함
추가 기회	상대 평균 1.5배 +1 / 2.5배 +2	속도 편성의 차이를 만들게 함
방향/약점	약점, 후방, 상태, 다운 각 +25%	위치와 연계를 선택하게 함
기본 피해	ATK × 계수 × 100/(100+유효 방어)	방어 상승의 효용을 완만하게 함
시야	아군 시야 합집합 / AI도 시야 적용	정찰 없이 안전한 판단을 못 하게 함

핵심 메카닉스 – 시간 능력의 경계

상황	처리	목적
평소 전투	일시정지, 무르기, 저장을 지원함	전술 학습 비용을 낮춤
장 진행	체크포인트로 저장/재도전을 관리함	역행 서사를 재도입하지 않음
시간의 틈	반복 전투와 별도 기록을 둠	성장/숙련 기회를 제공함
최종 장치전	시간 능력 사용 시 장치 파괴를 기록함	편의 기능을 내려놓는 선택을 만듦
장치전 재도전	새 전투에서 장치를 복구함	실패 뒤 다시 도전할 기회를 보장함

구현 구조 – 규칙과 화면의 분리

Unity 6000.3 / C# 9

Core

UnityEngine 없이 판정함

- Battle, Grid, Game
- 결정적 난수/검증

Presentation

입력과 결과 표현을 담당함

- 전투, 필드, 캠프
- 카메라, 음향, UI

Data/Tools

콘텐츠를 생성/검사함

- JSON 로더/생성기
- CoreTests/Sim

구현 구조 – 화면 요청의 소비

1

Core 효과

Dialogue, Travel, Battle 요청을 생성함

2

요청 큐

발생 순서를 보존함

3

화면 처리

대화 종료 뒤 후속 요청을 소비함

4

재저장

아웃트로 효과까지 스냅샷에 반영함

구현 계획 – 필요한 제작 단위

현행 코어 연결 작업

규칙

전투 경계값을 명세함

- 시야, 상태, 패배
- 보상 1회 처리

콘텐츠

장별 생성기를 준비함

- 전투/퀘스트 ID
- 동료/엔딩 조건

도구/검수

개인 제작의 반복 작업을 줄임

- .NET/Unity
- 데이터 검사/화면 캡처

구현 계획 – 구현 게이트

1

규칙 단위

경계값/난수 재현을 통과해야 함

2

한 장 연결

접촉부터 캠프 복귀까지 검증함

3

갈래 연결

미영입, 사망, 패배를 통과해야 함

4

전체 인계

1부 인계 후 2부 예외를 검증함

리소스 – 아트 제작 규모

현재 폴더의 파일 수를 제작량 산정에 사용함. 시안/사본이 포함된 수량임.



리소스 – 생산과 납품 계약

대상	규격/명명	생산/검수
인물	CHR_<ID> / PNG RGBA	승인 외형→표정→도트 / 캐릭터 수와 파일 수 분리함
맵	MAP_<ID>/layers.json / PPU 64	격자 확정→바닥/물체→가림 검사 순서임
도트	시트별 셀, 행, 프레임 메타	원안 180/240프레임 표를 그대로 적용하지 않음
데이터	<표>__Ch<n>.json / UTF-8	247파일 실측을 참고하며 참조 오류 0을 요구함
음향	sfx_<key> / bgm_<key>	절차 합성 우선 / 오디오 파일 0개 실측임