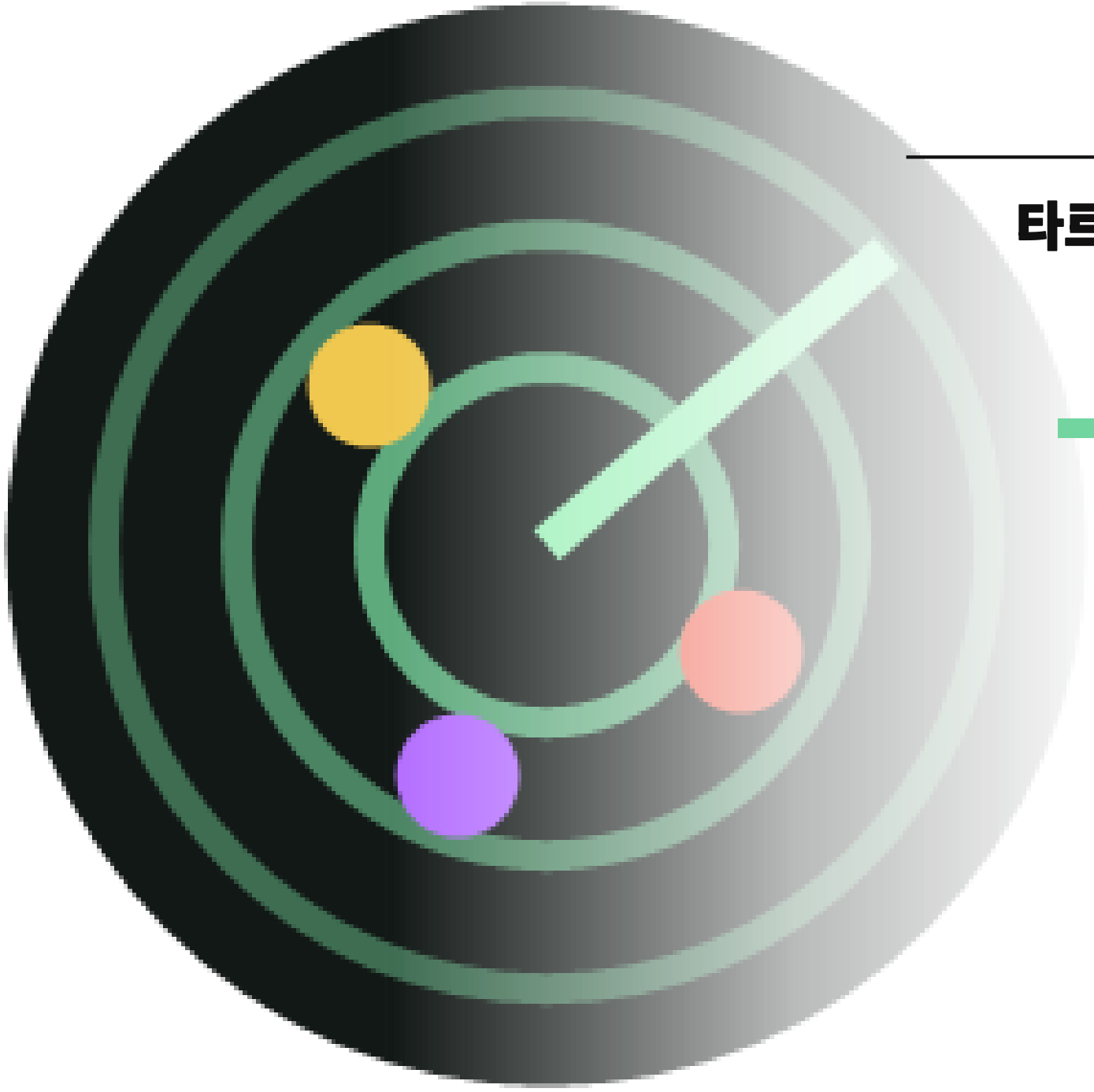


개인 프로젝트

타르코프 모니터 (분석 + 레이드 모니터)
메인 기획서

파서 / 레이드 엔진 / 라이브와 재생

문서 최초 작성	문서 최종 수정
20261026 12:04	20261026 12:20
박도원	박도원

목차

개요 – 2P

1. 목차
2. 로그
3. 문서 계층

현황 – 5P

1. 분석 완료 / 핵심 구현 전

목표 – 6P

1. 핵심 수용 기준

핵심 요소 – 7P

1. 핵심 시스템 4요소

작업 계획 – 8P

1. 입력에서 화면까지

본문 – 9P

1. 사망/가해자 판정 규칙
2. 위치와 상태의 보관 규칙
3. 레이드 관찰 루프
4. 목표 시각 언어

작업 계획 – 13P

1. 핵심 제작과 검증 순서

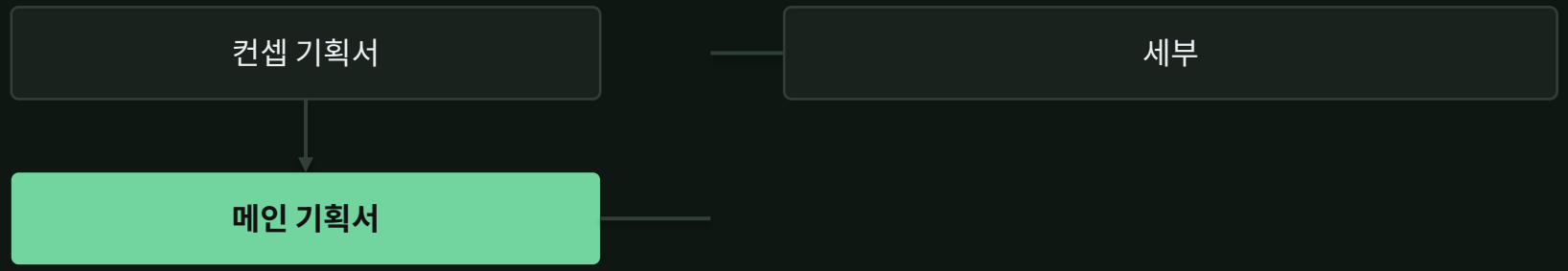
본문 – 14P

1. 예상 리스크와 대응

개요 – 로그

순번	작성일	작성자	작성 내용	비고
1	2026-10-26 12:04	박도원	메인 설계	수집, 판정, 전달 경계를 정함
2	2026-10-26 12:10	박도원	파서/카탈로그	타입 이벤트와 역할/맵 정보를 구성함
3	2026-10-26 12:13	박도원	소스 순서 보정	동일 밀리초 로그에 안정 정렬을 적용함
4	2026-10-26 12:20	박도원	핵심 검증 종료	라이브 따라잡기와 전체 재생의 일치를 확인함
5				
6				

개요 – 문서 계층



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

현황 – 분석 완료 / 핵심 구현 전

확보한 조사

939개 파일/AI 17개 파일의 패턴 분석을 확보함

의도 관측 범위에서 구현을 시작함

근거 10-26 11:33 분석 보고임

판정의 한계

BotDied에는 개체 ID가 없고 연속 좌표도 미확인임

의도 근접 이벤트 연결을 검증 대상으로 둠

근거 실제 게임로그 분석 문서임

구현 대상

파서, 엔진, 라이브 소스/재생 소스가 필요한 상태임

의도 입력과 표시를 분리함

근거 11:48 기능 가능성 판정까지 진행됨

목표 – 핵심 수용 기준



같은 로그를 넣으면 라이브 따라잡기와 전체 재생의 상태가 일치하도록 설계함

1초	0차이	3상태
읽기/스냅샷 갱신 기준	재생 간 최종 집계 목표	생존, 사망, 이탈

의도 실시간 표시 이전에 상태 계산의 재현성을 확보함

근거 같은 밀리초에도 순서가 다른 여러 이벤트가 존재함

핵심 요소 – 핵심 시스템 4요소

수집/정렬

파일별 오프셋과 시간 순서를 관리함

의도 누락/중복 처리를 줄임

근거 로그 채널이 여러 파일로 나뉨

개체 엔진

스폰, 사망, 이탈, 상태를 레이드 단위로 연결함

의도 서로 다른 레이드의 ID 혼선을 방지함

근거 로그에 레이드와 봇 식별자가 있음

좌표/보스 의미

이벤트 위치와 보스 예정/생성을 분리함

의도 서로 다른 확실성을 같은 표식으로 단정하지 않음

근거 Boss wave와 Bot activate는 별개임

전달/재생

동일 엔진의 스냅샷을 UI와 검증에 사용함

의도 별도 재생 로직의 불일치를 막음

근거 보관 로그를 다시 읽을 수 있음

작업 계획 – 입력에서 화면까지

1 LogSources

최신 세션의 추가 기록을 1초마다 읽음

.NET 10 / 파일 오프셋

의도 전체 파일 재분석 비용을 줄임

근거 파일이 계속 증가하는 구조임

2 LogParser

로그를 타입 있는 이벤트로 변환함

C# / 생성 정규식

의도 파싱과 판정을 분리함

근거 이벤트별 필드가 다름

3 RaidEngine

개체 상태와 궤적을 누적함

Core / SnapshotBuilder

의도 화면에 의존하지 않는 판정을 마련함

근거 동일 입력을 재생할 필요가 있음

4 Host / UI

스냅샷을 SSE로 지도와 목록에 전달함

WebView2 / HTTP / Leaflet

의도 창과 브라우저에 같은 UI를 사용함

근거 로컬 Windows 사용 환경임

본문 – 사망/가해자 판정 규칙

BotDied에 ID가 없어 인접 연결이 필요함. 동시 사망의 오귀속을 별도 검수함

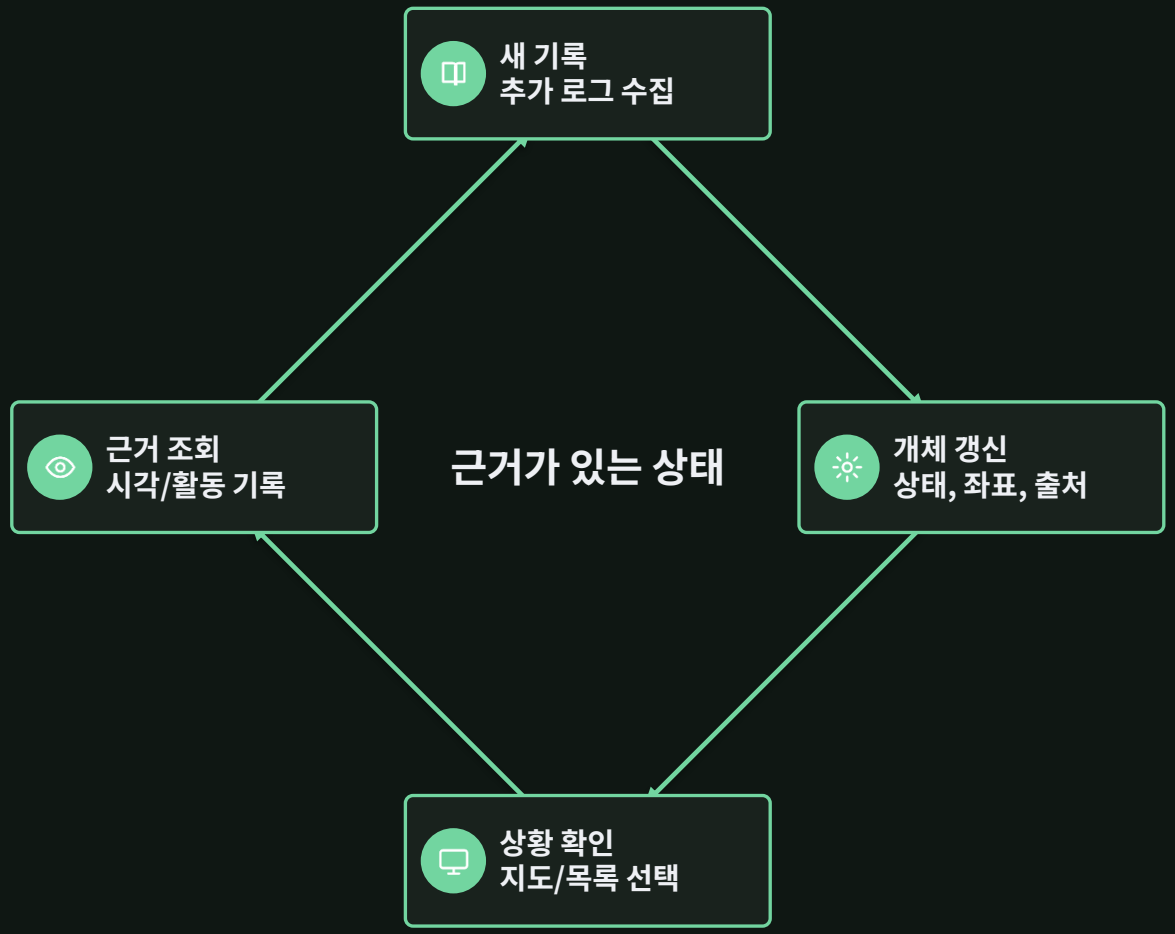
변수	설계값	적용 규칙
PairWindow	1초	dispose 직후 BotDied를 연결할 예정임
DisposeGrace	3초	사망 근거가 없는 소멸을 이탈로 처리함
KillerWindow	5초	사망 전 마지막 피격의 가해자를 연결함
생존 대조	allAlivePlayersList	개체 목록과 로그의 생존 ID를 대조함
미확인 가해자	기록 없음	피격 기록 부재를 임의 이름으로 채우지 않음

본문 – 위치와 상태의 보관 규칙

설계값으로 적용한 뒤 실제 재생의 가독성과 누락 여부를 검수할 예정임

대상	설계값	의도/근거
위치 노후화	45초	이벤트 공백을 흐린 마커로 드러냄
궤적	엔진 240개 / 전달 40개	누적 기록과 화면 전송량을 분리함
중복 억제	피격 2초 / 사격 5초 / 엄폐 10초	반복 로그가 이벤트 피드를 덮지 않게 함
보관 상한	피드 500 / 로그 80	장시간 누적에 필요한 한도를 둠
보스/보급	예정/확인 / 보급 시각	출현 예정은 실물 확인과 구분함

본문 – 레이드 관찰 루프



레이드 경계

종료 후 다음 레이드의 개체 상태로 전환함

의도 이전 봇이 새 레이드에 남지 않게 함

근거 레이드/맵 시작 로그가 존재함

위치 해석

목적지와 투척 방향을 현재 위치로 사용하지 않음

의도 거짓 좌표를 막음

근거 분석에서 벡터와 위치의 의미가 다름

본문 – 목표 시각 언어



어두운 전술 테마

정보 종류를 색상, 기호, 텍스트로 함께 구분함

의도 레이드 중 짧은 시선 이동을 지원함

근거 사용자의 UX 중시 요구임

출처와 경과 시간

좌표 출처와 마지막 갱신 시각을 함께 표시함

의도 오래된 점의 현재 위치 오인을 줄임

근거 이벤트가 생길 때만 좌표가 남음

목표 시각 예시임. 후속 제작된 P049 아이콘을 목표 목업으로 사용함

작업 계획 – 핵심 제작과 검증 순서

1

파서/카탈로그

이벤트와 한글 역할/맵 키를 정의함

C#

의도 판정 입력을 통일함

근거 역할 문자열이 로그에 존재함

2

엔진/소스

시간 워터마크와 안정 정렬을 적용함

Core

의도 같은 시각의 원래 순서를 보존함

근거 사망 연결은 인접 순서에 의존함

3

재생 대조

전체 재생과 라이브 따라잡기를 비교함

ReplayCheck

의도 집계 차이를 UI 이전에 제거함

근거 보관된 실제 세션을 이용 가능함

4

UI 인계

스냅샷 필드와 데이터 한계를 전달함

JSON / README

의도 표시 의미를 일관되게 유지함

근거 코어/UI가 같은 데이터 계약을 사용함

본문 – 예상 리스크와 대응

⚠ 동시 사망이 인접 이벤트 연결을 흐릴 수 있음	→ 🛡 BotDied 없는 소멸은 이탈로 두고 생존 목록과 대조함
⚠ 게임 설정 변경이 실행 환경을 손상할 수 있음	→ 🛡 설정은 읽기/진단으로 한정하고 파일 쓰기를 제외함
⚠ 지도 API 장애로 투영 검수가 지연될 수 있음	→ 🛡 메타/SVG를 캐시하고 구역명 대조와 독립 검증을 나눔