타르코프 모니터 (분석 + 레이드 모니터) — 메인기획서

> 재구성 기획서 (역설계) / 작성 기준 시점: 2026-10-26 (프로젝트 시작 시점) / 문서 버전: Ver 0.1 프로젝트 ID: P049 / 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. 핵심 메카닉스

봇 개체 식별과 집계

- 의도(Why): 봇 수, 종류, 개별 생존 목록의 기반.
- 설계(How): `Bot activate id:N role:… profileId … name … pos:(x,y,z)`로 개체 생성, `allAlivePlayersList: 1:assault, 10:pmcBEAR…`로 생존 목록 대조. 플레이어는 `id:1`로 판정한다 (`isAI:False`는 봇에도 찍히므로 부적합, 봇 id는 2부터). [근거: LogParser.cs, claude_a52b1ffe]
- 필요(What): aiData Trace 로그.
- 작업 방법: 실제 로그로 id 규칙을 확인한 뒤 판정식을 교체.
- 완료 기준: 레이드별 봇 수가 로그와 일치.
- 리스크/대안: 프로필 이름이 플레이어 스캐브와 섞임. 플레이어는 별도 "나"로 분리. [근거: claude_a52b1ffe]
- 근거: [근거: claude_a52b1ffe, LogParser.cs]

사망, 이탈, 가해자 판정

- 의도(Why): 사망 좌표와 가해자를 개체에 연결한다.
- 설계(How): `bot dispose:N` 직후 1초(`PairWindow`) 안의 `BotDied pos`를 연결하고, 사망 기록이 없으면 "이탈"로 둔다. 가해자는 사망 전 5초(`KillerWindow`) 내 마지막 `bot get hit from enemy`. `DisposeGrace`는 3초. [근거: RaidEngine.cs, README.md]
- 필요(What): `BotDied` 줄에는 봇 ID가 없다는 사실에 기반한 인접 연결.
- 작업 방법: 재생으로 사망/이탈 분포 확인.
- 완료 기준: 사망 좌표와 가해자 표시, 피격 기록 없으면 "기록 없음".
- 리스크/대안: 동시 사망 혼선. 기존 앱보다 생존 목록 대조를 추가해 정확도 보강. [근거: claude_a52b1ffe]
- 근거: [근거: RaidEngine.cs, README.md]

위치 추정(마지막 확인 위치)

- 의도(Why): 연속 위치가 없는 제약 아래 최대한 자주 갱신된 위치를 보여 준다.
- 설계(How): 위치 소스를 이벤트별로 정의한다: 구역 출입, 엄폐 탐색(FindPoint, 오차 수 m), 경로 보정(NMLink), 수류탄 판단 시점, 스폰 좌표. 궤적은 봇당 `TrailCap` 240개 보관, 스냅샷은 40개. 45초 넘게 갱신 없으면 흐리게 표시. [근거: RaidEngine.cs, SnapshotBuilder.cs, README.md]
- 필요(What): 좌표 포함 줄 패턴 목록.
- 작업 방법: FindPoint를 위치 소스로 추가 후 중앙값 비교.

- 완료 기준: 봇당 위치 기록 중앙값 증가. 실제 보고값: 레이드 1에서 2→6, 레이드 2에서 10→12, 커스텀 레이드 5. [근거: claude_a52b1ffe]
- 리스크/대안: 이동 목적지/구역 기준점을 현재 위치로 오인. 투척 방향 벡터는 위치로 쓰지 않음. [근거: 게임로그_분석.md]
- 근거: [근거: claude_a52b1ffe, 게임로그_분석.md]

봇 상태 추정

- 의도(Why): 이동, 교전, 수색 등을 개체별로 확인.
- 설계(How): 사격, 피격, 적 인지는 "교전", 엄폐는 "엄폐", 수색 지점은 "수색", 순찰 경로는 "순찰/이동", 그 외 "대기" 등. 최근 기록 기준의 추정. 로그 연속 기록은 중복 억제(피격 2초, 사격 5초, 엄폐 10초). [근거: RaidEngine.cs, README.md]
- 필요(What): 상태별 로그 줄 패턴.
- 작업 방법: 재생에서 상태 칩 확인.
- 완료 기준: 목록에 상태, 구역, 마지막 행동 표시.
- 리스크/대안: 기록 시각 기준이라 실제 상태와 지연. 문서에 한계 명시. [근거: README.md]
- 근거: [근거: README.md, RaidEngine.cs]

보스/보급 추적

- 의도(Why): 기존 앱의 핵심 기능을 승계.
- 설계(How): `Init Boss wave … ShallSpawn` 예정과 `Bot activate` 실제 생성을 구분. 호위병 생존 수. 보급은 `client/airdrop/loot` 요청과 상자 물리 경고로 "발생 시각만" 기록. [근거: README.md]
- 필요(What): backend 로그.
- 작업 방법: 보스 탭 카드에 호위 수 추가.
- 완료 기준: 예정/확인/생존/사망 상태 표시.
- 리스크/대안: 낙하 좌표/내용물 없음. 표시하지 않음.
- 근거: [근거: README.md, claude_a52b1ffe]

내 위치(스크린샷)

- 의도(Why): 공식 기능만으로 플레이어 위치를 얻는다.
- 설계(How): 스크린샷 파일명(날짜[시-분]_x, y, z_쿼터니언…png)을 1초 주기로 폴더 스캔해 좌표/방향을 얻는다. 맵은 촬영 시각에 진행 중인 레이드로 결정, 로비 촬영분은 무시. 이후 구역 출입으로 위치가 갱신되면 방향 화살표는 제거. [근거: Screenshots.cs, claude_a52b1ffe]
- 필요(What): 문서`\Escape from Tarkov\Screenshots`.
- 작업 방법: 10/25 레이드 스크린샷 8장으로 재생 검증.
- 완료 기준: 마커 좌표가 파일명 값과 일치(보고됨).
- 리스크/대안: 사용자가 스크린샷을 찍어야만 갱신.
- 근거: [근거: claude_a52b1ffe, README.md]

2. 구현 방법

- 구조: `Core`(LogParser → LogSources → RaidEngine → SnapshotBuilder → RaidMonitor), `Host`(WebServer: 정적 UI, SSE 스트림, 제어 엔드포인트, MainForm: WebView2 창), `wwwroot`. [근거: README.md, 파일 목록]
- 로그 소스: 라이브는 최신 세션 폴더를 추적하며 파일별 읽기 오프셋과 시간 워터마크를 쓴다. 한 파일당 읽기 예산 4MiB. 같은 밀리초 줄 순서를 지키기 위해 안정 정렬을 쓴다(`List.Sort`는 불안정). 재생은 배속 시계로 이벤트를 재주입하고 8초 초과 유휴 구간을 건너뛴다. [근거: LogSources.cs, claude_a52b1ffe]
- 스트리밍: 스냅샷을 최소 1초에 1회 SSE로 재전송. [근거: claude_a52b1ffe]
- 좌표 변환: tarkov.dev 맵 메타(변환/경계)를 받아 Leaflet 투영에 적용. [근거: MapService.cs, README.md]
- 검증 도구: `tools\ReplayCheck`로 기록 세션을 엔진에 통과시켜 요약 출력, 라이브 따라잡기와 전체 재생 비교. [근거: README.md, claude_a52b1ffe]
- 설정: 게임 폴더 자동 탐색(Tarkov Boss Monitor 설정 → Steam 라이브러리 → 기본 경로), 스캐브 추적 옵션 등은 `AppSettings`에 저장. [근거: README.md, claude_a52b1ffe]

3. 시스템 간 상호작용과 밸런스 기준

- 정확도 대 커버리지: 확정 정보(사망/스폰)와 추정 정보(상태/위치)를 UI에서 구분한다. [근거: README.md]
- 핵심 변수: 시간창 3종(1초/3초/5초), 갱신 흐림 임계 45초, 궤적 용량 240, 이벤트 피드 500, 로그 상한 80. [근거: RaidEngine.cs, README.md]
- 밸런스 대응 개념 없음(도구). 스캐브 제외 같은 필터가 정보 밀도와 가독성을 조절한다. [근거: claude_a52b1ffe]

4. 게임의 나머지 부분의 방향

- 콘텐츠 구조(제품): 맵별 보스/역할 카탈로그(한글 이름, 분류, tarkov.dev 맵 키). [근거: Catalog.cs, claude_a52b1ffe]
- 메타 진행: 해당 없음(도구).
- UI/UX: `03_세부기획서_log-analysis-ui-data.md` 참조.
- 아트/사운드: 독자 아이콘, 어두운 전술 테마. 사운드 기록 없음.
- 저장/설정: 설정 파일, 지도 캐시(`%LOCALAPPDATA%\TarkovRaidMonitor\cache`). [근거: README.md]
- 향후 후보: 탈출구 상태는 로그에 근거가 없어 보류, 남은 레이드 시간(`GameStarted`+맵별 제한)은 가능 판정이나 미구현. [근거: claude_a52b1ffe]

5. 제작 과정 설계

- 순서: 분석(Codex) → 가능성 판정 → 파서 → 엔진 → 로그 소스 → UI → 재생 검증 → 배포. 병렬 가능: UI와 엔진 후반. [근거: claude_a52b1ffe]
- 분담: 로그 조사/정적 분석은 Codex, 구현/배포는 Claude, 게임 내 실험/로그 설정 판단은 사용자. [근거: claude_a52b1ffe]

- 검증: 지난 세션 재생, 브라우저 `--headless`로 UI 확인, 배포본 스모크 테스트. [근거: claude_a52b1ffe]
- 인수인계: README, 분석 문서, 다음 세션용 메모(플레이어 id:1, 사망 판정 방식, 위치 비연속, 보급 시각만, 설정 수정 금지, aiMoveData 좌표 없음). [근거: claude_a52b1ffe]
- 배포 제약: 실행 중인 앱이 `dist` 파일을 잡고 있어 덮어쓰려면 앱 종료가 필요. [근거: claude_a52b1ffe]

6. 세부기획서 목록

- `03_세부기획서_log-analysis-ui-data.md` (L 규모 1개): 로그 이벤트 명세, 데이터 필드, UI/UX 구성, 리소스, 검수 기준을 하나로 다룬다.