김민혁

Minhyuck Kim



다양한 분야의 아이디어를 기획하고 실행으로 옮깁니다.

상명대학교 서울캠퍼스 휴먼AI학과
2027년 2월 학사 졸업 예정
2002년 4월 4일생 · 서울 거주

기능의 **동작 규칙과 완료 기준**을 정합니다. Codex로 구현과 검사를 진행하고, 결과를 보고 다음 수정 범위를 결정합니다.

상명대학교 휴먼AI학과 · 2027년 2월 졸업 예정

hollamba22@gmail.com
https://github.com/hollamba22

선별한 프로젝트

01

크릴리오네

투자정보 웹의 **프론트엔드·백엔드·웹디자인·홍보를 1인 개발로 진행했습니다**. GPT-5, 6로 화면, 기능과 전용 AI 판단 모델을 개발하고 갱신과 Instagram 발행을 자동화했습니다.

1인 기획·개발·운영 · 웹 개발·운영 / 모델 연구

02

Wi-Fi CSI 졸업작품

가정용 CCTV의 움직임 감지 기능을 Wi-Fi로 대신할 수 있는지 탐구한 졸업작품입니다. 두 강의실에서 CSI를 수집하고, 인원수 분류와 보행 감지 모델을 자체·외부 데이터로 검증했습니다.

5인 팀 · 장비·수집·전처리·모델 실험·검증 참여 · 팀 졸업 연구 · 결과 정리

03

주막

AI로 메뉴를 만들고 주막에서 판매·공유하는 경영 게임을 기획했습니다. 현재 운영 프로토타입을 개발하고 AI 제작 도구와 **커뮤니티**를 설계하고 있습니다.

1인 기획·개발 · 게임 프로토타입 개발 · AI 서비스 설계

04

어드벤처 게임

GPT-6 Luna로 세계관과 개인적인 고민을 나누는 동료 로보를 기획했습니다. 게임 속 GPU의 발견·장착·용량을 자유 대화와 **API 운영비**에 연결합니다.

1인 기획·개발 · 게임 프로토타입 개발 · AI 서비스 설계

원고 기준 2026-11-06 / 로컬 검토용 초안

크릴리오네 / 웹 개발

프론트엔드·백엔드·웹디자인·홍보

1인 기획·개발·운영 | 2026년 ~ 현재
공개 서비스 운영 | Python · FastAPI · JavaScript

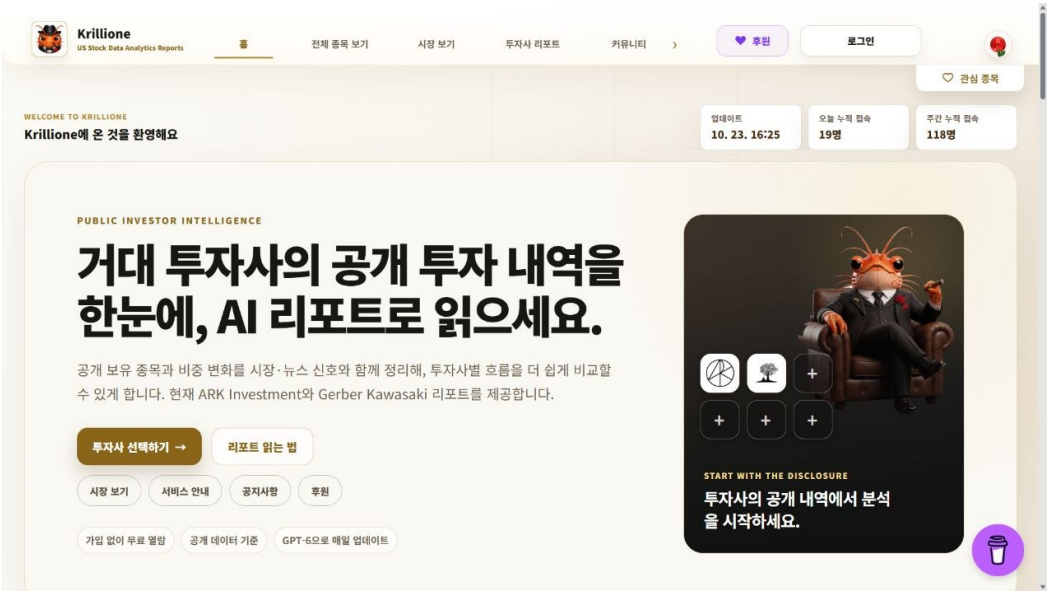
01 개발 과정

왜 만들었는가

투자사의 보유 정보는 매일 공개됩니다. 하지만 **일반 투자자**가 원문을 찾아 전날과 비교하고, 시장의 변화까지 읽기에는 **시간과 해석의 부담**이 큽니다.

저는 **ARK의 공개 보유 내역**을 중심으로 수량·비중의 변화에 가격·뉴스를 연결했습니다. 사용자가 '오늘 어떤 정보를 알아야 하는가'를 판단할 수 있는 서비스를 만들고자 했습니다.

많은 시장 정보와 시계열의 종가 관계를 **AI로 분석**하고자 했습니다. **0~100점**의 요약값에 해석과 기준일을 붙여, 사용자가 근거를 꼼꼼히 살펴 투자에 도움이 될 수 있게 했습니다.



크릴리오네 홈

탐색 구조와 화면 설계

처음에는 수많은 정보를 한꺼번에 보여주자 **결론과 상세 근거**가 뒤섞였습니다. 이는 기존의 읽기 쉬운 보고서를 만들겠다는 취지에 맞지 않았습니다. 따라서 최대한 많은 정보를 뒤로 배치하면서 사용자가 **시장→종목→리포트→기록** 순서로 화면을 탐색하며 정보를 얻을 수 있도록 구성하였습니다.

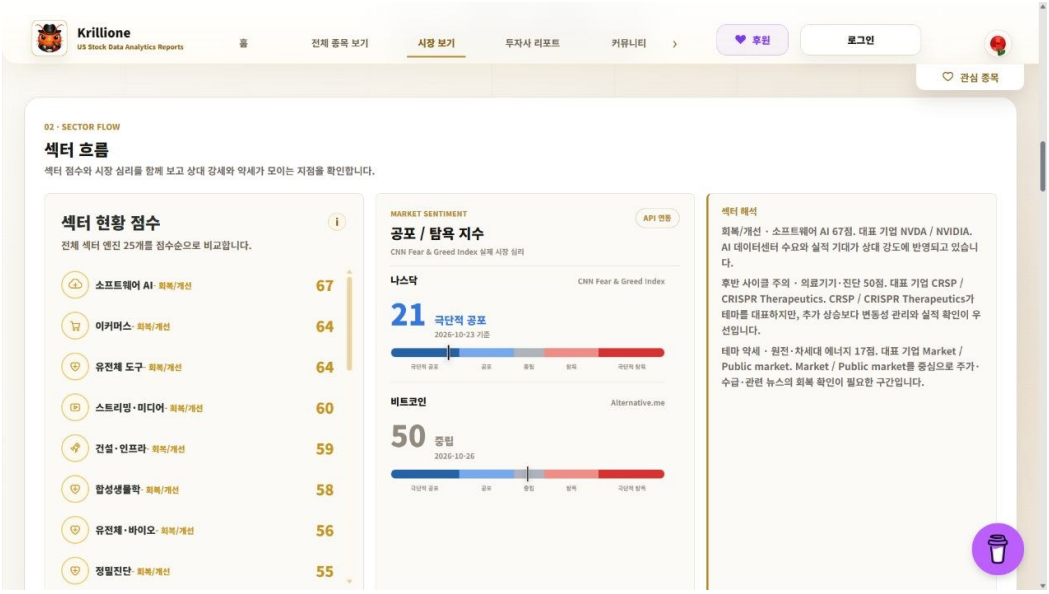
사용자가 먼저 볼 간단한 정보와 더 들어가 정보를 읽을 수 있도록 보고서 형식으로 제시했습니다. 매일 업데이트 되는 AI 보고서는 깊은 정보를 품고 있습니다.

시장 보기

개별 종목을 보기 전에 시장 전체를 읽도록 지수·변동성·원자재·가상자산을 배치했습니다. **25개 섹터** 점수와 공포·탐욕 지수로 분야별 흐름도 비교합니다.

TradingView P2P 계약으로 나스닥100 선물 차트를 도입했습니다. 나스닥100 선물 차트에 직접 그림을 그리거나 여러 지표를 추가할 수 있으며 로그인을 통해 저장됩니다.
FinancialJuice P2P 계약으로 실시간 거시 경제 뉴스를 도입했습니다. 따로 뉴스를 확인할 필요 없이 실시간으로 업데이트 되는 거시 경제 뉴스를 확인할 수 있습니다.

투자사 보유 변화·뉴스·실적 일정·예측 시장·FINRA ATS 거래량과 옵션 만기일의 거래량·미결제약정의 정보를 제공합니다.



섹터 흐름과 시장 심리

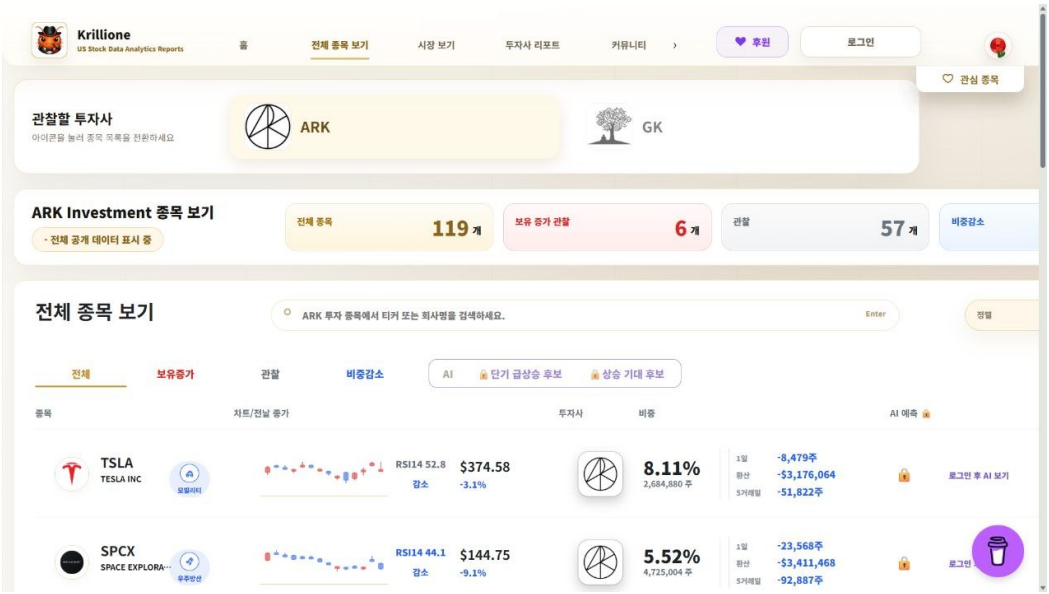
시장 보기 열기: https://krillione.com/market

전체 종목과 AI 점수

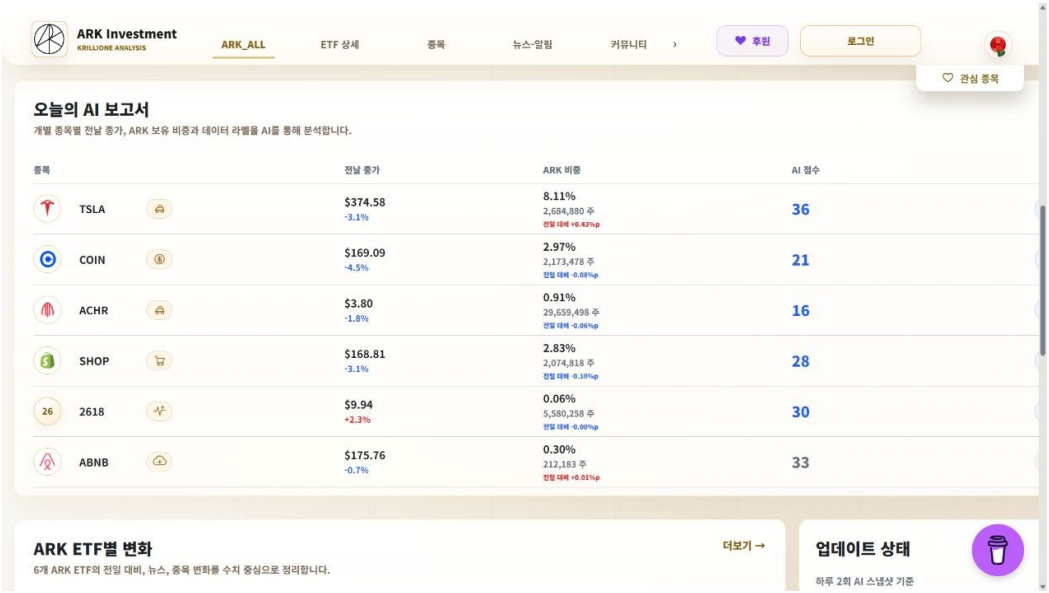
ARK·Gerber Kawasaki의 **공개 보유** 종목을 검색·정렬하도록 만들었습니다. 보유 증가·관찰·비중 감소 필터로 관심 대상을 좁힙니다.

가격·**차트**·보유 비중·**수량 변화**를 함께 배치했습니다. 가격 상승으로 비중만 커진 경우와 실제 수량이 늘어난 경우를 구분하기 위해서입니다.

기본 AI 점수에서 상세 분석으로 연결하고, 로그인 후 제공하는 예측 후보는 따로 관리했습니다. 금액 환산값은 공개 **수량 변화**와 기준 가격으로 계산한 값입니다.



전체 종목 목록



종목별 AI 점수

전체 종목 보기 열기: https://krillione.com/stocks

AI 점수 계산

보유 흐름·가격·뉴스·애널리스트·섹터 정보를 **0~100점**으로 묶었습니다. 점수는 더 살펴볼 종목을 고르는 지표이며, 좋은 신호가 많을수록 점수가 추가되는 방식입니다.

기관 점수에는 최근 5거래일 **수량 변화**와 20거래일 지속성, 가속·반전·여러 ETF의 흐름을 반영했습니다. 보유 감소와 자료 부족에는 **감점·상한·중립 보정**을 적용했습니다.

기준 점수에 **데이터 기반 AI 판단**을 연결했습니다. GPT-6.1 Terra·Sol은 뉴스 분석·설명을 맡습니다. 진입·핵심 보유·방어에 따라 비중을 조정하고, **상승 전용 예측 모델**은 별도로 연구를 진행하고 있습니다.

로컬 기본 점수 프로필의 구성

근거	기본 비중	함께 확인하는 내용
가격·타이밍	30%	가격 흐름과 기술적·시장 조건
종목·뉴스	20%	종목별 정보와 뉴스 근거
ARK 보유 흐름	35%	수량 변화·지속성·여러 ETF의 흐름
애널리스트	10%	분석 기준가 등 보조 정보
섹터	5%	관련 분야의 상대적인 환경

웹 점수의 흐름

자료	**공개 보유** 수량·비중·변화	**시장·종목** 가격·뉴스·섹터
계산	**기준 점수** 맥락별 가중치	**위험 보정** 감점·상한·결측
표시	**AI 설명** 뉴스 분석·근거	**0~100점** 점수·기준일·원문

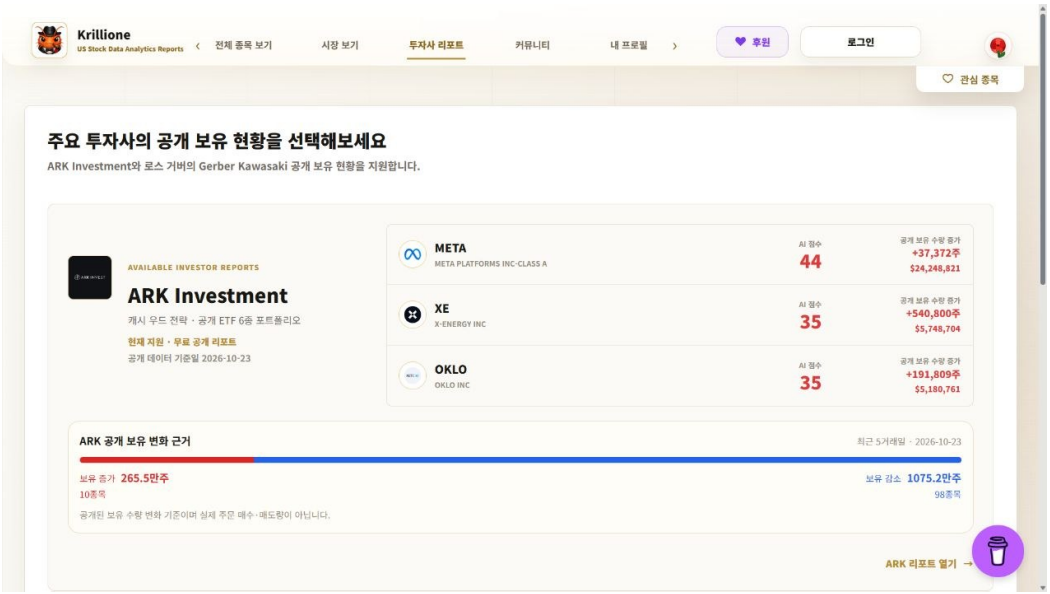
운영 점수와 EventMemoryModel 연구는 별개입니다.

투자사 리포트

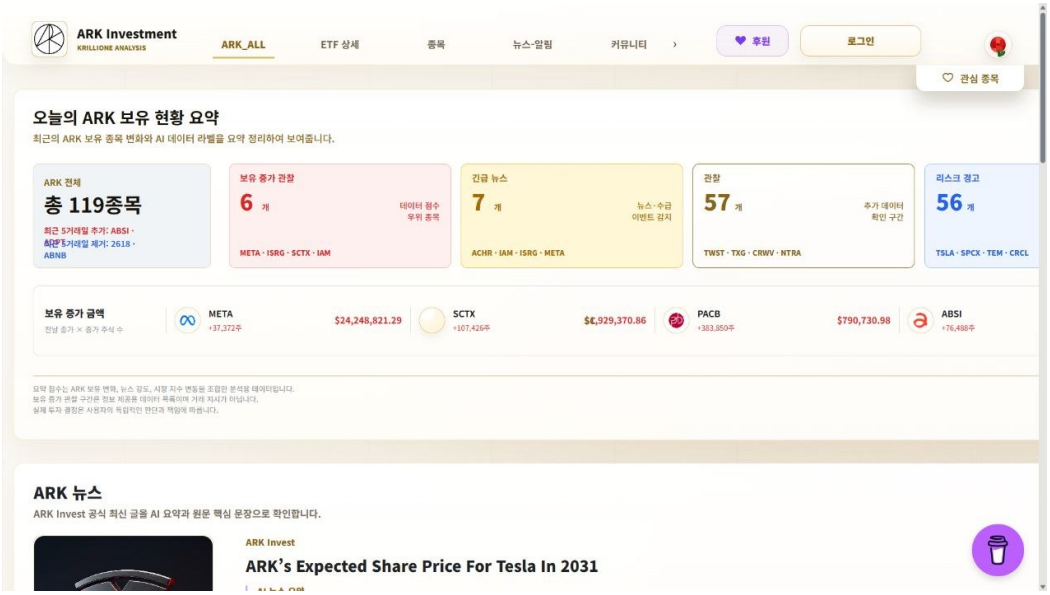
같은 종목도 한 투자사의 포트폴리오 안에서 보면 맥락이 달라집니다. **ARK**·Gerber Kawasaki의 **기준일**과 최근 변화를 보여주고 기관별 리포트로 연결했습니다.

ARK는 전체 보유, 추가·제거 종목, **수량 변화**의 금액 환산, AI 보고서와 6개 ETF의 변화를 묶었습니다. 공식 글과 원문도 확인할 수 있습니다.

공개 보유 변화만으로 주문 시각·체결 가격·기관의 의도를 알 수는 없습니다. 수량과 비중, 점수와 출처를 함께 보도록 구성했습니다.



투자사 리포트 선택



ARK 공개 보유 리포트

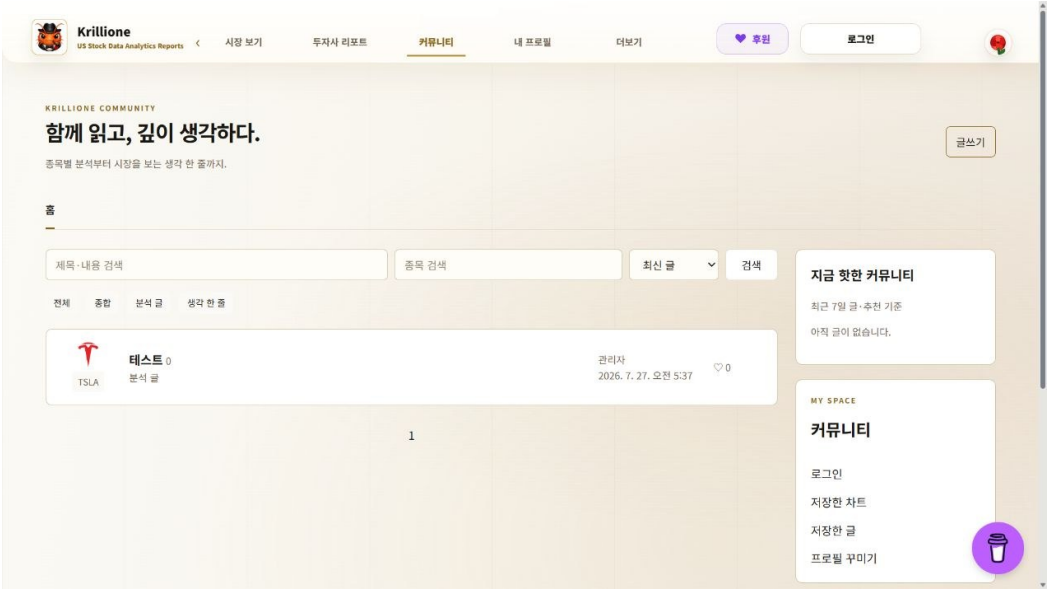
투자사 리포트 열기: https://krillione.com/investors

커뮤니티와 차트 공유

서비스의 해석을 읽은 뒤 자신의 생각도 남길 수 있게 **커뮤니티**를 만들었습니다. 종목별·종합 게시판에 검색·정렬·댓글·반응·북마크를 연결했습니다.

프로필과 개인 글에 전체 **공개·팔로워 공개·비공개**를 적용했습니다. 첨부 이미지와 **차트**에도 같은 접근 규칙을 적용합니다.

게시물에는 **당시 차트**의 레이아웃·선·**가격 시계열**을 복사했습니다. 원본 차트가 바뀌어도 게시한 근거와 작성자 표시가 남도록 했습니다.



커뮤니티와 개인 기록

커뮤니티 열기: https://krillione.com/commmunity

02 문제 해결

문제 사례와 해결과정

프론트엔드

정보를 유지하면서 종목 검색 지연을 줄였습니다

문제 검색어를 입력하거나 상세 화면으로 들어갈 때 화면이 멈춘 것처럼 보였습니다.

해결 기준 첫 수정 뒤 상세 뉴스·AI 분석 원문·아이콘이 빠졌습니다. 빠진 정보는 돌려놓고 지연만 줄이도록 기준을 다시 정했습니다.

바이브코딩으로 해결한 과정 Codex는 종목 정보와 아이콘을 복원했습니다. 검색에 필요한 목록을 먼저 준비하고, 느린 보조 목록 요청은 첫 화면 표시 뒤로 옮겼습니다.

확인한 결과 당시 로컬 브라우저에서 화면 준비 약 1.4초, 검색 반응 약 200ms를 기록했습니다. 상세 화면의 뉴스와 차트도 다시 열어 확인했습니다.

2026.06.25 ·「/stocks 렉 해결」로컬 브라우저 검수

종목마다 흩어져 있던 커뮤니티를 한곳에 모았습니다

문제 커뮤니티가 종목 상세 화면 안에 나뉘어 있어 글을 둘러보려면 종목을 먼저 골라야 했습니다.

해결 기준 커뮤니티를 독립 메뉴로 두고 종목은 필터로 고르게 했습니다. 종목과 무관한 글은 종합에서 읽을 수 있도록 했습니다.

바이브코딩으로 해결한 과정 Codex가 통합 목록과 종목 필터를 만들었습니다. 기존 종목 커뮤니티 링크도 통합 화면으로 연결해 해당 종목 필터가 바로 선택되도록 했습니다.

확인한 결과 로컬 한국어·영어 화면에서 데스크톱과 모바일을 확인했습니다. 임시 계정으로 익명 글과 댓글을 작성했고, 당시 통합 자동 검사 65건이 통과했습니다.

2026.09.12 · 「커뮤니티 기능 보완점 분석」 통합 화면 검수

글쓰기와 게시물 관리 순서를 다시 잡았습니다

문제 글쓰기와 글 관리에 버튼·검색·정렬이 흩어져 있었고, 본문에서 글자 굵기나 크기를 바꿀 수 없었습니다.

해결 기준 검색과 정렬을 묶고, 공개 범위와 삭제는 각 글에서 찾도록 정했습니다. 글을 쓰는 동안 기본 서식을 바꿀 수 있게 요청했습니다.

바이브코딩으로 해결한 과정 Codex로 게시 위치·제목·본문·설정·게시 순서를 정리했습니다. 굵기·크기·목록·인용과 실행 취소·다시 실행을 추가하고, 글 관리의 동작을 글별 메뉴로 옮겼습니다.

확인한 결과 로컬 한·영 화면에서 서식을 넣어 저장한 뒤 다시 열어 편집했습니다. 관련 자동 검사 45건이 통과했습니다.

2026.09.13 · 「커뮤니티 기능 보완점 분석」 편집기 검수

공유 차트를 가져와도 내 분석이 덮이지 않게 했습니다

문제 글에 공유된 차트를 내 종목 작업공간에 가져오면 기존 분석을 덮어쓸 수 있었습니다.

해결 기준 가져온 차트는 별도 사본으로 저장하고, 게시 당시의 선·설정·가격과 원작자 표시를 남기도록 정했습니다.

바이브코딩으로 해결한 과정 Codex가 공유 차트의 첨부·복제 기능을 구현했습니다. 미리보기를 본 뒤, 최신 가격은 버튼을 누르는 동안만 보이고 손을 떼면 게시 당시 차트로 돌아오도록 조건을 추가했습니다.

확인한 결과 로컬에서 차트를 저장하고 글에 첨부한 뒤 다시 열었습니다. 게시 당시 시세와 원작자 표시가 남는 것을 확인했습니다.

「커뮤니티 기능 보완점 분석」 · 차트 저장·첨부·재열람 검수

웹디자인

작은 화면에서도 검색창과 필터가 잘리지 않게 했습니다

문제 좁은 화면에서 홈의 갱신 문구와 종목 필터 일부가 잘렸습니다.

해결 기준 기존 색상과 아이콘, 메뉴 기능을 유지하면서 작은 화면에서도 모든 항목을 읽고 누를 수 있어야 했습니다.

바이브코딩으로 해결한 과정 Codex로 390px에서 넘치는 지점을 확인했습니다. 검색은 한 줄 전체를 쓰고 필터는 두 줄로 나눴습니다. 적용안을 보고 상하단 메뉴의 높이와 여백을 한 번 더 줄였습니다.

확인한 결과 360·390·430px 브라우저에서 가로 넘침과 메뉴·검색·필터 동작을 확인했습니다.

모바일 화면 수정 · 브라우저 너비별 검수 기록

데이터·점수

새 후보가 되는 조건과 남아 있는 조건을 분리했습니다

문제 일일 갱신 뒤 화면에서 후보가 모두 사라졌습니다. 기존 후보도 신규 진입 기준을 다시 통과해야 했습니다.

해결 기준 상태가 '강세'에서 '회복'으로 바뀌어도 수급과 위험 조건을 충족하면 기존 후보로 남도록 정했습니다.

바이브코딩으로 해결한 과정 Codex가 직접 동기화 과정에서 직전 정상 후보 이력이 끊긴 원인을 찾았습니다. 이전 이력을 연결하고 진입과 유지 검사를 분리했습니다. 유지된 후보는 최초 날짜·가격을 그대로 쓰도록 했습니다.

확인한 결과 로컬에서 당시 입력 147행을 다시 계산했습니다. 신규 0건, 기존 후보 6건 유지를 확인했고 관련 검사 271건이 통과했습니다.

2026.09.10 · 「확인 단기 급상승 후보」 및 후보 유지 규칙 검수

AI 리포트

한국어와 영어 보고서에서 같은 판단을 유지했습니다

문제 영어 화면에 한국어가 남았고, 보고서를 따로 생성하면 언어에 따라 수치와 판단 근거가 달라질 수 있었습니다.

해결 기준 화면 문구는 언어별로 관리하되, 보고서는 한국어판과 같은 모델로 영어판을 작성하고 수치·출처·구성은 보존하도록 했습니다.

바이브코딩으로 해결한 과정 Codex가 페이지에 남은 문구를 바꿔 끼우는 방식에서 언어별 문구집으로 전환했습니다. 보고서는 영어로 다시 쓴 뒤 수치·출처·항목 순서가 달라지면 저장하지 않도록 검사를 붙였습니다.

확인한 결과 당시 로컬 보고서 119건에서 한·영 API의 모델·제공자·점수와 한국어 원문 식별값이 일치하는 것을 확인했습니다.

2026.09.10 · 「영어 모드 잔존 한국어 수정」 및 한·영 보고서 검수

백엔드

게시물마다 반복하던 DB 조회를 묶었습니다

문제 커뮤니티 목록에서 글을 하나 읽을 때마다 공감·팔로우·북마크 상태를 다시 조회했습니다.

해결 기준 공개 이후 데이터베이스 조회를 점검하도록 요청했습니다. 조회 개선 전후에도 로그인 사용자가 보는 상태와 글의 공개 범위를 지켜야 했습니다.

바이브코딩으로 해결한 과정 Codex가 글별로 반복되던 상태와 차트 정보를 페이지 단위로 묶었습니다. 읽기 요청에서 쓰기 잠금을 미리 잡는 동작과 차트 목록의 불필요한 본문 읽기도 걷어냈습니다.

확인한 결과 로그인 상태의 로컬 회귀 검사에서 글 1개와 20개 모두 SQL 호출 7회였습니다. 커뮤니티 관련 자동 검사 28건이 통과했습니다.

2026.09.13 · 커뮤니티 운영 점검 및 조회 횟수 회귀 검사

예약은 실행됐지만 화면이 갱신되지 않는 문제를 고쳤습니다

문제 예약이 실행된 날에도 원천 자료가 멈추거나 상세 차트가 서버에 올라가지 않아 화면이 이전 상태에 머물렀습니다.

해결 기준 예약 성공 여부만 보지 않고 자료가 준비됐는지, 분석했는지, 웹에 반영됐는지를 각각 확인하도록 했습니다. PC 로그인 후에는 누락 작업부터 복구하도록 정했습니다.

바이브코딩으로 해결한 과정 Codex에 예약 로그·원천 자료·게시 상태의 대조를 맡겼습니다. 갱신이 멈춘 ARKW 자료를 공식 경로로 교체하고 상세 차트 전송을 보완했습니다. 지난 회차의 전송과 당일 분석을 따로 실행하게 했습니다.

확인한 결과 10월 9일 14:24에 전날 전송이 복구됐습니다. 당일 자료는 두 차례 대기한 뒤 15:47에 갱신과 서버 반영을 마쳤습니다. 미처리 작업은 0건이었습니다.

2026.10.09 · 「일일 업데이트 누락 보완 및 배포」 및 운영 이력(KST)

10월 9일 갱신 복구 실제 기록 (TXT)

홍보·운영

후보별 게시물을 스토리와 주간 묶음으로 나눴습니다

문제 처음에는 새 후보가 나올 때마다 피드에 별도 게시물이 올라갔습니다. 한 주에 어떤 종목이 나왔는지 한 번에 보기 어려웠습니다.

해결 기준 새 후보는 스토리로 알리고 주간 후보는 한국어·영어 캐러셀 하나씩에 묶기로 했습니다. 날짜와 가격은 추천이 처음 나온 시점을 쓰도록 정했습니다.

바이브코딩으로 해결한 과정 Codex로 웹 후보 갱신과 카드 생성·발행을 연결했습니다. 신규 스토리와 주간 캐러셀의 실행 조건을 나누고, 최초 추천값과 게시 결과를 남겨 재실행 시 중복을 막도록 했습니다.

확인한 결과 10월 30일 운영 기록에서 한·영 신규 스토리 4건이 하이라이트에 보관된 것을 확인했습니다.

2026.10.30 · Instagram 발행 이력 및 하이라이트 확인

핵심 변경 사항

· **게시 시점 보존** — 공유 차트에는 게시 당시 시세를 저장합니다. 최신 가격은 버튼을 누르는 동안만 보여주고, 버튼을 놓으면 게시 당시 차트로 돌아갑니다.

· **갱신 복구 분리** — 원천 확인·분석·서버 전송을 나눴습니다. 전송이 빠진 경우 분석을 다시 돌리지 않고 해당 자료만 올립니다. 갱신에 실패하면 마지막 성공본을 유지합니다.

· **발행 이력 저장** — 추천 당시 날짜와 가격을 기준으로 콘텐츠를 만듭니다. 발행 결과를 기록해 같은 작업을 다시 실행해도 중복 게시를 피합니다.

03 웹 운영

서버 공개와 갱신 자동화

Python·FastAPI로 **수집·분석** API를 만들고 JavaScript 화면과 연결했습니다. 로컬에서 분석을 마친 결과를 서버로 보내고 도메인으로 공개했습니다.

웹 요청마다 분석을 반복하지 않고 검증된 스냅샷을 읽도록 했습니다. **수집·분석·전송**이 실패하면 **마지막 성공 결과**를 유지합니다.

PC 종료나 미로그인으로 놓친 **갱신**도 복구하도록 정했습니다. 로그인과 예약 실행 때 원천 자료·**누락 작업**을 확인하고 입력이 바뀐 부분부터 분석을 재개합니다.

자동 갱신

점검	**예약·로그인** 자료·누락 작업 확인	
생성	**수집·분석** 보유·시장·뉴스 → 점수·리포트	
게시	**완료본 검증** 필수 자료·게시 상태	**서버 동기화** API·화면에 전달

실패하면 마지막 완료본을 유지합니다.

사업자 등록과 TradingView 계약

크릴리오네 운영을 위해 사업자를 등록하고 **TradingView**와 **P2P 계약**을 체결했습니다. 외부 기술 도입에 필요한 협의와 계약을 직접 진행했습니다.

도입한 차트를 종목 탐색·**공개 보유** 내역·**AI** 분석에 연결했습니다. 개발한 기능이 실제 서비스에서 제공되기까지의 절차를 함께 맡았습니다.

Instagram 발행 자동화

웹에서 게시를 마친 후보 기록으로 @krillione 콘텐츠를 만듭니다. 새 후보는 **한국어·영어** 스토리로, 한 주의 신규 후보는 다음 주 첫 성공 **갱신** 때 캐러셀로 묶습니다.

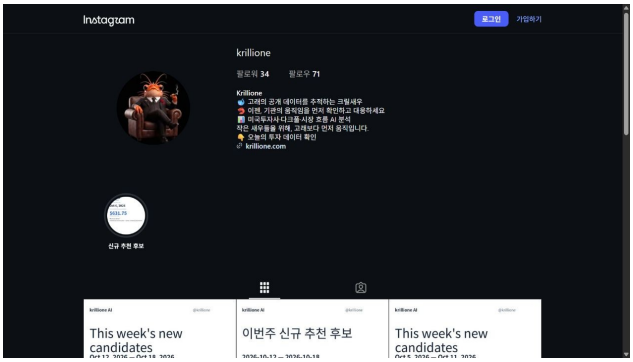
가격·점수만 바뀐 경우에는 신규 알림을 보내지 않습니다. 카드에는 **최초 날짜·당시 가격·근거**를 남기고, 필수 자료가 빠지면 발행을 기다립니다.

웹·서버 동기화가 성공한 뒤 **Meta API**로 게시합니다. 식별자·주차·게시 영수증으로 **중복**을 막고, 결과가 불확실하면 검토 대기로 남깁니다. 하이라이트 추가 여부는 별도로 확인합니다.

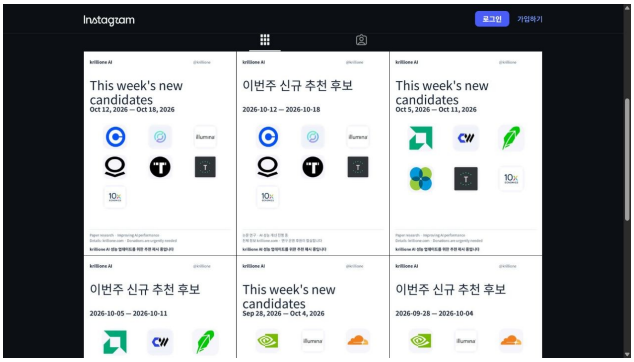
Instagram 발행

원천	**웹 후보 기록** 최초 날짜·가격·근거	
생성	**새 후보** 한·영 스토리	**주간 묶음** 한·영 캐러셀
발행	**이미지 검수** 자료·문구 확인	**Meta API** 게시·중복 방지·영수증

웹과 서버의 게시 성공 후 실행합니다.



크릴리오네 Instagram



한국어·영어 주간 게시물

@krillione 방문: https://www.instagram.com/krillione/

크릴리오네 / 모델 개발

기존 AI 점수 → 이벤트 V2 → 단기 상승 요건 추적

1인 연구 기획·데이터 구축·모델 설계·평가 | 2026년 중순 ~ 현재
V2 학습 완료 · 63거래일 관측 중 · 단기 웹 프로토타입 | Python · PyTorch

01 기존 AI 점수 판단 모델

왜 만들었는가

투자사가 보유량을 늘린 종목을 발견해도 차트가 이미 과열됐는지, 기업에 악재가 있었는지까지 한 번에 확인하기는 어려웠습니다. 심지어는 투자사 또한 판단이 잘못되어서 강하게 매수하는 날에도 하락하는 경우가 드물게 발생하였습니다. 그래서 매일 공개 보유 자료와 시장 정보를 모으고, 종목마다 어떤 근거가 점수에 영향을 줬는지 **백분율 점수로 한 눈에 볼 수 있는 구조**를 만들었습니다.

전체 종목의 **AI 점수**와 추천 후보의 선별 기준은 구분했습니다. 종목을 비교할 때는 공통 점수를 보여주고, 후보를 고를 때는 단기 급상승과 상승에 각각 **별도 조건**을 적용했습니다.

점수 계산과 후보 선정

차트와 진입 시점, 종목 뉴스, ARK 보유 변화, 애널리스트 목표가, 업종 흐름을 각각 점수화했습니다. 주가가 올라 펀드 내 비중만 커진 경우는 **주식 수가 늘어난 경우**와 나누었습니다. 시장이 위험회피 국면이거나 종목이 과열됐을 때는 종합 **점수에 상한**을 뒀습니다.

단기 급상승 후보는 위축된 업종에서 회복하는 흐름과 강한 업종에서 아직 덜 오른 종목을 따로 살폈습니다. 상승 후보에는 투자사의 보유 증가 근거와 기업·업종 흐름이나 조정 후 회복 조건을 함께 요구했습니다. 매일 **근거와 가격 기준일**을 갱신하고, 편입과 종료는 이력에 남겼습니다.

기존 AI 점수에서 추천 이력까지

입력	**투자사 보유 변화** 보유 수량과 비중	**종목·시장 자료** 차트·뉴스·목표가·업종
판단	**종합 점수** 항목별 점수와 위험 상한	**후보 선별** 단기 급상승 / 상승
기록	**추천 이력** 추천일·기준가·관찰 최고가·종료가	

전체 종목의 공통 점수와 후보별 선별 조건을 분리했습니다.

한계와 두 모델로의 전환

시장 위험회피를 점수에 반영했지만, **나스닥이 떨어질 때** 높은 점수의 종목이 얼마나 버텼는지는 따로 확인해야 했습니다. 관찰 중 한 번 올랐더라도 종료 시점에는 하락할 수 있습니다. 그래서 관찰 중 최고가와 함께 **종료 수익률, 하락폭, 시장 대비 움직임**을 봐야 한다고 판단했습니다.

먼저 기관 보유 변화의 순서와 시장 반응으로 한 달(21거래일)과 석 달(63거래일)의 수익과 하방을 추정하는 **이벤트 V2**를 만들었습니다. 그 전망에서 산출한 점수에 최근 가격과 거래량을 더해, 지금 진입을 검토할 만한지를 따로 판단하는 **단기 모델**로 이어갔습니다.

02 이벤트 모델 V2

왜 만들었는가

같은 보유 증가라도 앞선 감소를 되돌린 것과 여러 날 이어진 증가는 의미가 다를 수 있습니다. 저는 기관 행동을 점수 한 칸에 넣는 대신, **변화의 순서**와 당시까지의 시장 반응을 학습할 수 있는 구조를 설계했습니다.

초기 사건 기억 모델은 3·10거래일 SPY 대비 초과수익을 예측했습니다. 종목이 하락해도 시장보다 덜 떨어지면 초과수익은 양수입니다. 후속 장기 모델은 **21·63거래일**의 종목 자체 수익, SPY 대비 초과수익, 하방 움직임을 따로 추정합니다.

데이터 수집

기관 보유 자료가 올라오면 원문을 모으고, 시장 자료는 마감된 거래일의 시세에 맞춥니다. 원본 기준일, 추정한 경제적 보유일, **실제 확보 시각**을 따로 저장하여 분석합니다.

과거 ARK 아카이브는 SEC 보유 공시 표본과 대조했습니다. 예를 들어 두 자료를 같은 날짜로 맞췄을 때는 공통 식별자 1,010개 중 462개만 수량이 같았습니다. 다음 미국 거래일 기준으로 비교하니 **공통 1,010개**가 모두 일치했습니다. 과거 공개 시각의 지연 가정은 실제로 확보 시각을 기록한 전진 자료와 구분합니다.

수량 변화는 분할과 펀드 전체 유출입의 영향을 구분합니다. 보유 행이 빠졌을 때는 **결측으로 남깁니다**. 공개 보유량으로 확인할 수 있는 것은 잔고 변화까지입니다.

데이터 검증

ARK 아카이브와 SEC의 날짜 기준을 맞췄습니다

문제 같은 날짜 라벨로 맞춘 ARK·SEC 표본의 보유 수량이 서로 달랐습니다.

해결 기준 과거 ARK 자료에 시세와 기업행사를 붙이고, SEC 표본과 대조하도록 정했습니다. 과거 복구 자료와 실시간 관측은 분리했습니다.

바이브코딩으로 해결한 과정 Codex로 날짜를 같은 날과 다음 미국 거래일 기준으로 각각 대조했습니다. 아카이브 날짜와 경제적 보유일을 따로 저장하고, 과거 공개 시각의 지연 가정도 별도로 기록했습니다.

확인한 결과 표본 대조로 날짜를 하루씩 옮겨 검증했습니다. 양쪽에 각각만 있던 항목과 실제 과거 공개 시각은 별도로 미확인으로 남겼습니다.

2026.10.23 · 「논문 작성 방향 분석」 및 ARK 과거 보유 이력 실험

놓친 수집은 재개하고 성공분은 보존했습니다

문제 일부 종목을 돌아가며 수집할 때 실행 간격을 늘리면 전체 확인이 늦어졌습니다.

해결 기준 자료가 올라오는 시점에 맞춰 전체를 수집하고, 실패한 항목만 다시 요청하도록 했습니다. PC가 꺼져 놓친 실행도 다음 로그인 때 보충하도록 정했습니다.

바이브코딩으로 해결한 과정 Codex로 펀드·종목별 성공 여부를 저장하고, 실패만 재시도하는 흐름을 구성했습니다. 이후 실행에서는 남은 자료와 해시·거래일을 다시 확인합니다.

확인한 결과 한 실행에서 요청한 보유 자료 7개와 시세 21개를 모두 확보했습니다. 이후에도 성공·실패와 누락 여부를 실행별로 점검합니다.

2026.11.06 · 「기관 이벤트 수집 주기 찾기」 및 기관 이벤트 수집 완료 기록

모델 설계

초기에는 보유 변화마다 기억을 만들고, 유보·같은 방향·반대 방향 반응을 유지·수정 상태와 결합했습니다. 사건을 펀드로 모으고 다시 운용사 단위로 묶었습니다.

이벤트 모델 V2는 ARK 6개 펀드의 누적 보유 변화와 시장 특징을 각각 인코딩한 뒤, 시장 맥락에 따라 기관 특징을 얼마나 반영할지 학습합니다. **최근 3거래일**의 상대 거래량, 증가 흐름, 기울기도 시장 입력에 넣었습니다.

매 거래일 새 기관 자료와 완료 시세로 같은 학습 가중치에서 전망을 갱신합니다. 당시 입력과 예측은 보존합니다. V2가 추정한 21·63거래일 절대수익, SPY 대비 초과수익, 하방 위험은 뒤의 단기 모델에서 가점을 계산하는 데도 사용합니다.

입력 실험

거래량을 학습 입력으로 살렸습니다

문제 3거래일 거래량이 급증해도 이미 주가가 올라 있거나 변동성이 크면 같은 의미로 볼 수 없습니다.

해결 기준 최근 거래량을 모델 판단에 반영하도록 요청했습니다. 실제 영향과 성능은 고정된 비교 조건에서 확인했습니다.

바이브코딩으로 해결한 과정 Codex로 상대 거래량·증가 흐름·기울기 3개를 넣고, 23개 일봉이 불완전하거나 분할이 걸리면 결측 표시를 붙였습니다. 같은 V2 가중치에서 거래량 입력만 가린 결과도 비교했습니다.

확인한 결과 26,355행 중 47행은 결측 표시로 남았습니다. 63일 초과수익 추정이 거래량 입력 후 높아진 행과 낮아진 행을 모두 확인했습니다. 평가 오차 개선은 확인하지 못했습니다.

2026.11.04 · 「논문 작성 방향 분석」 및 장기 V2 거래량 학습 보고서

이벤트 모델 · 장기 V2

입력	**기관 누적 108열** 1·5·21·63거래일 보유 변화	**시장 25열** 가격·위험·최근 거래량 3개 특징
결합	**기관·시장 인코더** 각각 24차원으로 변환	**가중 결합** 시장 맥락에 따라 기관 특징 조절
출력	**절대·초과수익** 종목 자체 수익과 SPY 대비 수익	**하방 움직임** 진입 후 불리한 가격 움직임

133개 입력과 결측 표시를 사용합니다. 21·63거래일마다 수익과 하방을 함께 출력합니다.

초기 3·10거래일 사건 기억 모델

입력	**기관 특징 8개** 수량·비중·시점·마스크	**시장 특징 16개** 가격·거래량·마스크
표현	**사건 인코더** 변화별 기억 생성	**시장 GRU** 과거 가격 문맥
기억	**반전 처리** 같은 펀드·종목	**6개 상태** 유지·수정 × 반응 3종
통합	**사건 주의집중** 사건 → 펀드	**운용사 통합** 내부 → 운용사 간
출력	**기억 + 시장 문맥** 상대 수익률·방향	

과거 20거래일 입력으로 SPY 대비 초과수익과 방향을 예측한 초기 구조입니다.

학습과 검증

기존 ARK 보유 이력과 시세로 **V2 학습을 마쳤습니다**. 시간순으로 학습 17,070건과 검증 3,800건을 나눴고, 이전 실험에 사용한 2026년 상반기 2,370건에서 모의 평가를 진행했습니다.

63거래일 뒤 수익이 양수인지 맞힌 비율은 **52.03%**였습니다. 날짜별 예상 수익 상위 3종목을 뽑은 모의 선별에서는 왕복 비용 1%를 빼고 75건 중 **28건(37.33%)**의 수익이 양수였습니다. 63거래일 절대수익 오차(MSE)는 V2가 0.18505로, 상수 기준 0.09874와 시장 전용 0.10304보다 컸습니다.

매 거래일 새 예측을 고정해 두고 21·63거래일이 지나면 실제 수익과 하락폭을 연결합니다. **2027년 초**부터 새 데이터의 63거래일 결과를 확인할 예정입니다. 두 기간 모두 서로 다른 **진입일 20일 이상**의 성과가 확보되면 비용 후 수익과 하방을 비교해 검증을 마칠 계획입니다.

전진 검증

학습 완료와 추천 자격을 분리했습니다

문제 학습 오차가 줄어도 새 기간의 성과를 보장하지 않습니다. 이미 확인한 구간의 성과를 다시 독립 검증처럼 해석할 수 있습니다.

해결 기준 새 추천은 미사용 기간에서 거래비용과 하방 위험까지 검증한 뒤에만 열도록 정했습니다. 과거 재현과 실제 전진 검증은 별도로 집계합니다.

바이브코딩으로 해결한 과정 Codex로 동일 입력의 기준선과 시장 전용 모델을 비교하고, 분할 경계에 겹치는 결과를 제거했습니다. 추천 검증에는 왕복 비용 1%, 하방 한도, 서로 다른 진입일 수를 기록하도록 했습니다.

확인한 결과 과거 모의 평가에서 V2는 사전에 정한 오차 기준에 미달했습니다. 새로 고정한 예측의 21·63거래일 결과가 쌓이면 같은 기준으로 다시 확인합니다.

2026.11.04~06 · 장기 V2 학습 보고서와 진단·편입 원장

운영 검증

이력이 맞지 않을 때 마지막 검증 집계를 유지했습니다

문제 성과 집계와 추천 이력의 식별자가 맞지 않으면 다른 추천의 결과가 붙거나 같은 기록이 두 번 집계될 수 있습니다.

해결 기준 이력과 성과가 같은 기록으로 연결된 경우에만 최신 집계로 교체하도록 했습니다.

바이브코딩으로 해결한 과정 Codex로 입력 이력과 성과의 연결을 확인하고, 전송하는 집계의 접근 권한과 재전송 중복 방지를 검수했습니다.

확인한 결과 과거 이력 43건의 연결을 재확인해야 했던 검수에서는 교체를 보류하고 직전 검증 집계 118건을 유지했습니다.

2026.11.06 11:35 KST · 크릴리오네 모델 구성 및 검증 보고서

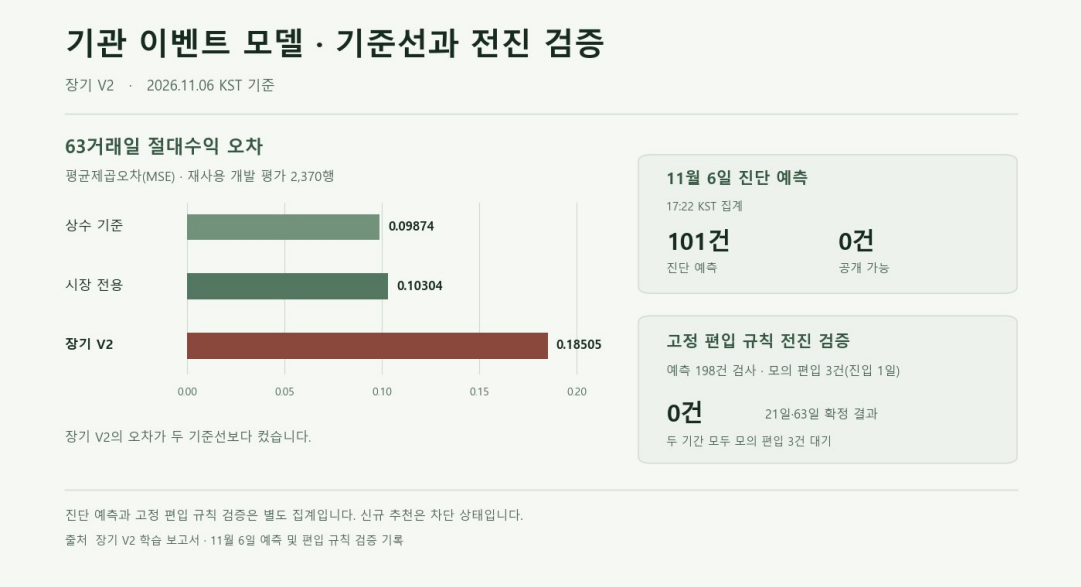
V2 과거 모의 평가 · 2026년 상반기 재사용 자료

기간	수익 방향 정확도	상위 3종목 모의 선별 · 비용 후 수익 양수
21거래일	56.96% (1,350 / 2,370건)	44.00% (33 / 75건)
63거래일	52.03% (1,233 / 2,370건)	37.33% (28 / 75건)

이벤트 모델 · 진단과 공개 편입 검증

추론	**고정 V2와 최신 자료** 21·63일 절대·초과수익과 하방 추정	
모의 편입	**기관 증가 근거와 전망 개선** 같은 기준집단에서 점수 개선 확인	**비용·위험 필터** 왕복 비용 1% 차감 후 절대·초과수익 양수 · 예상 하방 10%/15% 이내
성과 확정	**실제 전진 원장** 21·63일 각각 완료된 결과를 별도로 보존	**공개 심사** 두 기간 확정 진입일 최소 20일 · 평균 순절대·초과수익 양수 · 하방 한도 초과율 25% 이하

진단 예측은 계속 저장합니다. 공개 심사는 별도로 검증한 진입일과 성과를 사용합니다.



이벤트 V2 · 기준선 오차와 초기 전진 검증 사례

V2 학습·모의 평가·전진 검증 기록 (TXT)

논문 등록 과정

제가 확인하고 싶은 것은 기관이 보유량을 바꾸는 순서와 강도를 알면, 가격과 거래량만으로 판단할 때보다 이후 움직임을 더 잘 설명할 수 있는가입니다. **같은 시점과 종목**에서 기관 입력을 쓴 V2와 **시장 전용 모델**을 비교합니다.

예측을 먼저 저장하고 이후 21·63거래일 결과를 연결합니다. 어느 시장 국면에서 기관 정보가 도움이 됐고 어디에서 빗나갔는지, 데이터가 없었던 경우까지 정리해 **논문으로 발전시킬 수 있는지** 검토하고 있습니다.

03 단기 상승 요건 추적 모델

왜 만들었는가

이벤트 V2의 모의 검증을 보면서 전망이 좋더라도 언제 진입할지는 별도로 판단해야 한다고 봤습니다. 그래서 **V2 전망에서 산출한 점수**에 현재 **가격 위치, 거래량과 하방 조건**을 더했습니다. 크릴리오네에 올라오는 ARK·GK 종목을 매 거래일 같은 기준으로 비교합니다.

일일 실행에서 신규 후보를 선별하고 주간 누적을 확인합니다. 후보는 **하루 0~3개, 주간 0~5개**입니다. 조건을 만족하는 종목이 없으면 빈 목록을 그대로 기록합니다.

데이터 수집

같은 게시 시점의 종목 목록에서 중복을 정리하고, 종목과 SPY의 날짜가 맞는 **연속 23개 완료 일봉**을 확인합니다. 최근 3개 봉과 그 앞 20개 봉을 나누면 비교 기준에 현재 거래량이 섞이지 않습니다.

미국 거래소·USD 가격·마감 여부를 확인합니다. 수정주가의 조정계수가 크게 달라진 구간은 기업행사를 확인할 수 있도록 보류합니다. 기관 보유 근거가 있는 경우 원문과 **실제 관측 시점**을 확인합니다.

개장 전 확보한 자료로 가격 조건까지 평가할 수 있는 종목을 매일 확인합니다. 한 실행에서는 종목·시세 식별자 **132개 중 119개**의 가격 조건을 평가했습니다. 나머지는 확보 시각, 일봉 누락, 기업행사 등 보류 사유를 남겼습니다.

데이터 검증

과거에 볼 수 있었던 종목과 가격으로만 비교했습니다

문제 오늘의 종목 목록이나 뒤늦게 받은 일봉으로 과거를 재현하면 당시 판단할 수 없었던 정보를 사용하게 됩니다.

해결 기준 개장 전에 확보한 게시본과 시세를 사용하고, 출처나 가격이 맞지 않으면 보류하도록 정했습니다.

바이브코딩으로 해결한 과정 Codex로 게시본의 버전, 종목·SPY의 거래일, 시세 확보 시각을 검사하는 단계를 만들었습니다. 당시 게시본이 없는 날은 평가 불가로 따로 기록합니다.

확인한 결과 당일 입력과 사후 복원 자료를 분리했습니다. 같은 조건을 다시 검증할 때에도 당시 종목 목록과 평가 가능 여부를 확인할 수 있습니다.

2026.11.05 · 단기 진입 신호 일일 검수 기록

모델 설계

20거래일 고점을 저항선으로 삼고, 그 아래에서 거래량이 쌓이거나 막 돌파한 종목을 찾습니다. 최근 3거래일 평균 거래량은 직전 20거래일 평균의 **1.30배 이상**이어야 합니다. 돌파 뒤 허용 범위는 고점 위 3%입니다.

SPY 대비 상대강도 또는 확인된 기관 보유 증가에 가격 흐름, 최소 거래대금과 하방 7% 조건을 함께 봅니다. 진입 요건을 충족하면 근거 점수는 최대 85점입니다. 같은 진입일의 V2 전망이 21·63거래일 비용·위험 기준을 통과하면, **21거래일 예상 순초과수익**의 순위로 **최대 15점**을 더합니다.

이번 비교에서는 저항선 아래 **접근 폭만 3%와 5%**로 바꿨습니다. 거래량·돌파 상한과 일 0~3건·주 0~5건 기준은 두 규칙에 동일하게 적용합니다. 같은 입력을 다시 실행해도 그날 **처음 확정한 기록**을 유지합니다.

판단 기준

V2 전망과 단기 진입 조건을 함께 판단했습니다

문제 V2의 21·63거래일 전망으로는 오늘의 가격이 이미 과열됐는지, 돌파를 뒷받침하는 거래량이 있는지 알기 어려웠습니다.

해결 기준 가격과 거래량으로 진입을 검토하고, 같은 진입일의 V2 전망이 비용·위험 조건을 통과할 때 순위에 가점을 반영했습니다.

바이브코딩으로 해결한 과정 Codex로 같은 후보군에서 진입 근거만 쓴 목록과 V2 가점을 더한 목록을 만들었습니다. 두 목록에 같은 편입 상한을 적용해 점수가 실제 선별을 바꾸는지 확인합니다.

확인한 결과 첫 비교에서 네 조합 모두 진입 조건을 통과한 후보가 없었습니다. 후보가 없었던 날도 기록하고, 이후 양쪽의 편입과 실제 가격을 비교합니다.

단기 V2 가점 전진 비교 원장

전진 검증

저항선 접근 범위만 넓혀 비교했습니다

문제 저항선과 거래량 조건을 함께 만족하는 후보가 드물었습니다. 거래량 기준만 낮춘 과거 재현에서는 하방 한도를 넘는 사례가 생겼습니다.

해결 기준 거래량 1.30배와 편입 한도는 유지하고 저항선 아래 접근 폭만 3%에서 5%로 넓힌 안을 병행하도록 정했습니다. 공개 추천은 검증을 통과할 때까지 차단했습니다.

바이브코딩으로 해결한 과정 Codex로 같은 개장 전 배치를 두 규칙에 적용하고, 진입일별 첫 유효 결과를 해시와 함께 보존하도록 구현했습니다. 5거래일 순수익과 보유 중 최대 하락폭을 각각 집계합니다.

확인한 결과 같은 개장 전 배치로 두 규칙을 나란히 실행했습니다. 모의 편입이 없었던 날도 결과를 저장하고, 두 규칙의 성과를 비교할 수 있도록 편입 이후의 가격을 계속 모읍니다.

2026.11.05~06 · 「논문 작성 방향 분석」과 단기 저항선 전진 검증 원장

단기 후보 판단 · 진입 근거와 V2 점수

입력	**개장 전 종목·시세** ARK·GK 종목과 완료 일봉	**이벤트 V2 전망** 21·63거래일 수익·하방
판단	**진입 근거 최대 85점** 거래량 1.30배 · 저항선 아래 3% 또는 5%	**V2 가점 최대 15점** 비용·위험 통과 종목을 21일 순초과수익으로 정렬
검증	**가점 적용 전후 비교** 첫 유효 배치 고정 · 일 0~3건, 주 0~5건	**5거래일 성과** 비용 후 +3% 달성 · 하락폭 7% 기준

같은 입력에서 V2 가점이 후보 선택과 성과를 바꾸는지 확인합니다.

단기 모델 규칙 비교와 전진 검증 기록 (TXT)

학습과 검증

기존 모델에서 남긴 단기 급상승·상승 추천 이력으로 먼저 확인했습니다. 2026년 11월 4일에 보존한 크릴리오네 운영본에서, 9월 8일~10월 30일에 추천되고 종료가 확인된 **51건**을 집계했습니다. 두 유형에 겹친 추천은 원장의 한 건으로 계산했습니다.

관찰 중 상승 확인 비율은 **82.4%(42/51건)**였습니다. 원장에 저장된 관찰 최고가가 각 추천의 기준가를 한 번이라도 넘은 경우입니다. 종료 수익률은 기록이 있는 48건 중 **20건(41.7%)**이 양수였습니다. 예를 들어 ILMN은 관찰 중 **최대 +16.42%**였지만 종료 기록은 -0.43%였습니다.

현재는 매 거래일 개장 전 같은 자료로 저항선 접근 폭 3%와 5%를 비교하고, 각각 **V2 가점을 넣은 경우와 뺀 경우**를 기록합니다. 편입 후 3·5·10거래일 가격이 쌓이면 비용 차감 후 수익과 하락폭을 확인합니다. 아래 표는 기존 추천 원장 집계입니다.

비교 실험

AI 가점이 실제 후보를 바꾸는지 확인했습니다

문제 가점을 붙여 점수가 달라져도 후보나 성과가 달라지는지는 따로 확인해야 했습니다.

해결 기준 같은 날에 같은 수집 근거를 적용하고, AI 가점을 넣은 경우와 뺀 경우의 3·5·10거래일 성과를 비교하도록 정했습니다.

바이브코딩으로 해결한 과정 Codex로 두 집단에 같은 비용과 위험 조건을 적용하는 재현을 만들었습니다. 진입일 단위로 후보 구성과 비용 차감 후 수익 차이를 비교했습니다.

확인한 결과 과거 재현의 한 비교에서는 가점이 붙은 종목이 한 개였지만 편입 종목은 바뀌지 않았습니다. 실제 후보 구성과 수익이 달라지는지는 같은 방식으로 계속 확인합니다.

2026.11.05 · 단기 진입 신호 3·5·10거래일 과거 재현 기록

기존 추천 최대 상승률 상위 10종목 · 종목별 최고 1건 · 2026년

종목	추천 유형	추천일	최고가 관찰일	최대 상승률
NET	상승	09.11	09.24	+19.87%
COIN	상승	09.18	09.21	+19.75%
VCYT	상승	09.23	09.24	+17.89%
ILMN	상승	10.02	10.06	+16.42%
NTRA	상승	09.18	09.24	+12.73%
PLTR	상승	09.18	09.24	+10.46%
CBRS	상승	09.18	09.22	+9.82%
SHOP	상승	09.22	09.23	+9.69%
CRCL	상승	09.21	09.21	+8.92%
PACB	단기 급상승	09.22	09.22	+8.82%

크릴리오네 후보 이력 (로그인): https://krillione.com/ai/history

기존 추천 성과 집계 기준과 출처 (TXT)

결과와 운영 현황

V2의 모의 검증을 바탕으로 크릴리오네 웹에 후보를 살펴보는 **연구용 프로토타입**을 게시했습니다. 웹에서 후보와 이력을 확인할 수 있게 하고, 새 단기 규칙의 편입 결과는 따로 기록하고 있습니다.

지금은 진입 조건을 업데이트하면서 같은 개장 전 배치에서 **V2 가점이 후보 순위와 편입을 바꾸는지** 추적합니다. 후보가 생기면 5거래일 비용 후 수익 +3%와 하락폭 7% 이내를 함께 충족하는지 확인합니다. 첫 비교에서 편입이 없어 단기 성과 집계는 계속 진행 중입니다.

크릴리오네 AI 후보 화면: https://krillione.com/ai/candidates

크릴리오네 방문: https://krillione.com/

Instagram @krillione: https://www.instagram.com/krillione/

현재 상태와 다음 과제

· 이벤트 V2는 새 데이터로 63거래일 검증을 진행 중입니다. 단기 후보는 웹 프로토타입과 모의 원장으로 관찰합니다.

· **공개 보유 변화**로 주문·체결·의도를 직접 알 수는 없습니다. 출처와 **공개 시점** 검증을 이어갑니다.

· 시장 전용 기준과 비교해 기관 보유 변화의 설명력을 살피고 논문화 가능성을 검토합니다.

CSI / 실험 설계·데이터

신호를 모으고, 모델의 판단을 확인했습니다.

5인 팀 · 장비·수집·전처리·모델 실험·검증 참여 | 2026년 7~10월
팀 졸업 연구 · 결과 정리 | Python · PyTorch · Intel 5300 · CSI

카메라 없이 집 안의 움직임을 알 수 있을까

집을 지키기 위해 설치한 카메라가 사생활을 드러내는 통로가 되기도 합니다. 2025년 12월 경찰청은 국내 **IP카메라 약 12만 대**를 해킹해 영상을 탈취한 사건을 발표했습니다. 보안 해킹 우려가 폭증하는 가운데 CCTV를 대체할 수단의 필요성이 부각되고 있는 상황입니다. 우리는 집에 이미 설치된 Wi-Fi로 실내 변화를 읽고, **홈 CCTV의 움직임 감지 기능을 영상 촬영 없이** 구현할 수 있는지 실험했습니다.

CSI(Channel State Information, 채널 상태 정보)는 수신기가 Wi-Fi 패킷의 피지컬 데이터를 이용해 추정한 **물리 계층의 채널 신호**입니다. 송수신 링크와 부반송파 그룹마다 신호의 **진폭과 위상**이 얼마나 달라졌는지를 복소수로 기록한 것이 바로 CSI입니다.

수신기에는 직접 도착한 전파와 벽·가구·사람에 **반사되거나 산란·회절된 전파**가 겹쳐 들어옵니다. 사람이 움직이면 이 경로들의 합이 바뀌고, 그 변화가 **시간에 따른 CSI 패턴**으로 나타납니다. 모델이 이 패턴을 학습하면 **화면 없이도 인원이나 보행을 구분**할 수 있지 않을까 하는 가능성을 파악하고자 했습니다.

개인정보보호위원회 IP카메라 해킹 사례와 보안 안내: https://pipc.go.kr/np/cop/bbs/selectBoardArticle.do?bbsId=BS074&mCode=&nttId=12007

Linux 802.11n CSI Tool 기술 자료: https://dhalperi.github.io/linux-80211n-csitool/

Intel 5300과 Linux로 CSI를 수집한 과정

송신·수신 PC에 **Intel Wi-Fi Link 5300 카드와 안테나 3개씩**을 연결했습니다. 안테나 간격과 방향을 매 수집시마다 일정하게 맞추기 위해 **3D 프린팅 거치대**를 사용했습니다. 두 PC가 같은 **2.4 GHz 휴대폰 핫스팟**에 접속하고, 수신 PC는 핫스팟에서 들어오는 Wi-Fi 프레임의 CSI를 측정했습니다.

수신 PC는 **Ubuntu 18.04.6 LTS의 Linux 4.15 커널**에서 수정한 iwlwifi 드라이버와 펌웨어를 사용했습니다. connector_log=1로 CSI 로그를 켠 뒤, 송신 PC의 safe_sender.py가 **512바이트 UDP 패킷을 초당 100개씩** 보내도록 설정했습니다. 수신 PC의 **log_to_file**이 CSI를 DAT 파일에 저장하면, 파일과 실제 CSI 레코드가 함께 기록됐는지 확인했습니다.

수신 PC의 실행기에서 강의실, 장비 배치, **고정 인원·자유 활동 중 수집 방식**, **인원수, 행동, 수집 시간**, 웹캠 사용 여부를 지정했습니다. 필요한 회차에는 참가자의 자리와 행동별 시간·반복 횟수, **빈 방 기준 세션**을 함께 기록했습니다.

빈 방은 0명 기준으로 수집했습니다. 빈 방을 수집하는 이유는 공간에 대한 학습이 필요하기 때문입니다. 인원을 고정한 실험에서는 **앉기·서기·교대 걷기**를 순서대로 안내하기도 했습니다. 별도 자유 활동 실험에서는 **약 40분 동안 출입과 움직임**의 순서를 정하지 않고 기록했습니다.

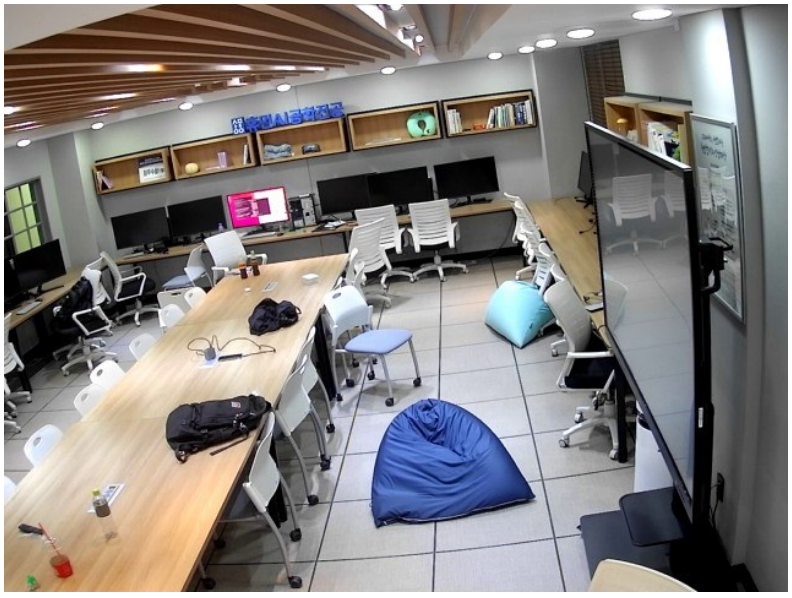
실험 장비와 패킷 흐름

송신	**송신 PC** Intel 5300 · 안테나 3개 · UDP 100패킷/초	
중계	**휴대폰 핫스팟** 2.4 GHz Wi-Fi로 수신 PC에 전달	
기록	**수신 PC** Intel 5300 · 안테나 3개 · Linux CSI Tool	**웹캠** 같은 실험을 촬영해 행동 라벨에 사용

송신 PC는 실험용 트래픽을 만들고, 수신 PC는 핫스팟에서 도착한 프레임의 채널 응답을 기록했습니다.



G304 강의실 · 실제 수집 영상 프레임



G308 강의실 · 실제 수집 영상 프레임

두 강의실에서 모은 CSI와 모델 입력

가구와 장비 배치가 다른 **G304·G308 강의실**에서 반복 수집해, 환경이 바뀌었을 때도 동일한 모델이 준수한 성능을 기록하는지 비교하고자 했습니다. **G304에서는 0~5명, G308에서는 0~4명** 조건을 기록했습니다.

수집 이력 31개 중 데이터가 남은 **원본 DAT 22개**에서 **CSI 레코드 1,417,202개**를 확인했습니다. 파일별 첫 기록부터 마지막 기록까지 합산하면 **약 4시간 53분**입니다. 이 가운데 **고정 인원 학습 후보는 15개 세션**, 약 3시간 46분이었고, 약 40분의 자유 활동 세션을 기록하여 모델 테스트 용으로 분리했습니다.

CSI의 형태는 **수신 안테나 3개 × 송신 공간 스트림 2개 × 부반송파 그룹 30개**였습니다. 학습 후보에서 실제 관측률은 **초당 약 70~87개**였습니다. 초기 인원수 모델은 6개 링크의 수신 안테나 순서를 복원하고 진폭을 보정한 뒤, **100 Hz 시간 격자에서 비중첩 2초 구간**을 만들었습니다.

수집 및 전처리 단계별 데이터 규모

범위	세션·파일	CSI 관측 시간	레코드·입력 구간
기록이 남은 전체 원본	DAT 22개	약 4시간 53분	레코드 1,417,202개
고정 인원 학습 후보	15개 세션	약 3시간 46분	레코드 1,102,726개
자유 활동 시험 후보	1개 세션	약 40분	레코드 190,409개
초기 인원수 모델 입력	14개 세션	전처리 후 구간 선별	2초 구간 5,716개

동시 촬영 영상에서 행동 라벨로

CSI를 받는 동안 웹캠으로 같은 공간을 촬영했고, 원본 영상을 **인원수 및 보행 라벨링**에 사용했습니다.

초 단위 영상 프레임을 검토해 빈 방, 앉아 있거나 제자리에 서 있는 구간은 **정적(S)**, 사람이 분명히 걷는 구간은 **보행(D)**으로 분류했습니다. 가림·암전으로 행동을 확인하기 어려운 구간은 제외했습니다. **영상과 CSI의 시간차**를 추정한 뒤 같은 행동 라벨 안에 온전히 들어오는 **2초 구간 4,344개**를 남겼습니다.

YOLO v8n을 실행해 화면 속 사람을 탐지하고 위치를 표시했습니다. 보행 라벨은 시간대별 영상 검토를 거쳐 확정했고, CSI 모델에는 **정렬한 라벨을 정답으로, Wi-Fi CSI를 입력으로** 사용했습니다.

초기에는 **Person-in-WiFi의 구조를 참고한 U-Net 계열 모델**로 0~5명 분류를 시험했습니다. 같은 수집 회차의 특징을 외우는지 확인하기 위해 **세션 단위**로 데이터를 나눴고, 이후 **인원 구간과 정적·보행을 별도 과제**로 다뤘습니다. 모델 설계·검증에서는 CSI만으로 학습한 모델들이 두 강의실에서 보행을 얼마나 잘 구분했는지 비교합니다.

데이터에서 모델 과제로

과제	정답	모델 입력
인원수 분류	수집 메타데이터의 0~5명	CSI 구간
인원 구간 분류	정해진 인원 범주	CSI 구간의 특징
정적·보행 감지	영상 검토 후 CSI에 맞춘 S / D 라벨	CSI의 진폭·위상 변화와 시간 문맥

CSI / 모델 설계·검증

신호를 모으고, 모델의 판단을 확인했습니다.

5인 팀 · 장비·수집·전처리·모델 실험·검증 참여 | 2026년 7~10월
팀 졸업 연구 · 결과 정리 | Python · PyTorch · Intel 5300 · CSI

01 두 논문에서 두 과제로

Person-in-WiFi와 DensePose From WiFi

Person-in-WiFi (ICCV 2019)는 연속된 CSI 표본을 2차원 특징 지도로 펼친 뒤, 잔차 합성곱과 U-Net을 통해 신체 윤곽과 관절 위치·연결을 추정했습니다. 카메라에서 얻은 정답으로 학습하고, 추론에는 Wi-Fi 신호를 넣었습니다.

DensePose From WiFi는 진폭과 보정된 위상을 따로 인코딩한 뒤 결합해, 신체 24개 영역의 표면 좌표까지 추정했습니다. 같은 배치의 평균 정밀도(AP) 43.5가 학습하지 않은 배치에서는 27.3으로 떨어졌고, 세 명 이상이 동시에 등장하면 개인별 신호를 분리하기 어려웠다고 보고했습니다.

우리는 DensePose From WiFi를 선행 연구로 검토하고, 실제 기준 모델은 **Person-in-WiFi의 공간 변환과 잔차 합성곱·U-Net을 응용**해 처음부터 학습했습니다. 수집 장비에서 나온 6개 링크의 진폭을 5시점씩 읽어 96×96 특징 지도로 변환하고, 단일 U-Net과 전역 평균 풀링 뒤에 0~5명 분류기를 붙였습니다. 2초 구간은 짧은 입력 조각들의 분류 확률을 평균해 판단했습니다.

Person-in-WiFi · Wang 외, ICCV 2019: https://arxiv.org/abs/1904.00276

DensePose From WiFi · Geng 외: https://arxiv.org/html/2301.00250#S4.SS7

같은 신호에서 인원수와 보행을 분리한 이유

초기 U-Net은 원본 녹화가 겹치지 않도록 나눈 14세션·5,716구간에서 **개발 정확도 22.97%**를 기록했습니다. 별도로 남긴 자유 활동 녹화에서는 **66/1,024구간(6.45%)**을 맞혔습니다. 실제 인원이 2~4명이던 자유 활동 평가에서 예측의 56.74%가 5명으로 몰렸습니다.

별도 통계 분류기의 진단에서도 **같은 세션을 섞어 나누면 98.31%, 세션 하나를 통째로 빼면 17.28%**였습니다. 한 번의 녹화에서 잘 맞는 특징이 새 녹화의 인원수까지 설명하지 못했습니다. 강의실, 날짜, 채널, 배치가 인원 조건과 함께 바뀐 회차가 있어 어떤 요인이 오류를 일으켰는지 따로 가려내기 어려웠습니다.

같은 네 사람도 앉아 있을 때와 걸을 때 신호가 달라집니다. 반대로 빈 방과 사람이 가만히 있는 방에서는 둘 다 시간에 따른 변화가 작을 수 있습니다. **몇 명이 있는가와 누군가 걷고 있는가는 서로 다른 질문**입니다. 인원은 방별 진폭 특징으로 범주를 추정하고, 보행은 진폭·위상 변화와 직전 흐름으로 판별하도록 나눴습니다.

02 인원수 범주 모델

강의실별 CSI로 0명·1~2명·3명 이상 분류

발표에 사용한 인원 모델의 출력은 **0명·1~2명·3명 이상**입니다. 각 강의실에 맞춰 모델을 따로 학습했습니다. 2초의 CSI 진폭을 보정하고 64 Hz로 재표본화한 뒤, 6개 링크와 30개 부반송파 그룹에서 **시간 변동·링크별 통계·부반송파별 평균 등 265개 특징**을 계산했습니다. 학습 데이터로 정한 10~90백분위 범위로 특징을 스케일링하고, RBF SVM으로 세 범주의 경계를 학습했습니다.

인원 조건별 자료를 시간순으로 나눠 **앞 80%로 학습하고 뒤 20%를 시험**했으며, 경계 전후 15초는 제외했습니다. G304는 **510/596구간(85.57%)**, G308은 **521/521구간(100%)**을 맞혔습니다. 같은 원본 녹화의 앞뒤가 학습과 시험에 들어갈 수 있으므로 이 수치는 해당 강의실과 수집 조건에서의 분류 결과입니다.

G304에서는 실제 3명 구간 140개 중 **81개(57.86%)**만 3명 이상으로 맞혔습니다. 3명 이상 전체 결과보다 경계에 걸친 3명 조건의 오류가 컸습니다.

아래 그림은 G308에서 고른 세 인원 범주의 수집 영상과 같은 시간대의 실측 CSI입니다. 6번 링크는 0명 예시에서 가장 높았고, 2번 링크는 1~2명 예시에서 가장 높았습니다. **인원 범주가 바뀔 때 링크마다 진폭의 변화 방향이 달랐습니다.** 모델은 30개 부반송파 그룹의 여러 특징을 함께 사용합니다.

인원 범주별 정답 수 / 시험 구간

실제 인원 범주	G304	G308
0명	59 / 59	105 / 105
1~2명	123 / 145	212 / 212
3명 이상	328 / 392	204 / 204
전체	510 / 596 · 85.57%	521 / 521 · 100%

인원 범주 모델의 입력과 출력

입력	**2초 CSI 진폭** 6개 링크 × 30개 부반송파 그룹 · 64 Hz	
특징	**진폭 특징 265개** 전체 변동 · 링크 통계 · 부반송파별 평균	
분류	**G304 전용 RBF SVM** 0명 · 1~2명 · 3명 이상	**G308 전용 RBF SVM** 0명 · 1~2명 · 3명 이상

강의실을 지정해 해당 방에서 학습한 스케일러와 분류기를 사용했습니다.



G308 인원 범주별 수집 장면과 실측 CSI

03 정적·보행 모델

왜 주판단기와 보조판단기를 나눴나

보행은 2초 구간의 경계에 걸치기도 합니다. G304의 문맥 실험에서는 **현재 2초만 보면 보행 8/12구간, 직전 문맥을 더하면 9/12구간**을 찾았습니다. 정적 오경보는 두 경우 모두 3/202구간이었습니다. 현재 구간의 변화가 약해도 직전에 이어지던 보행이 판별의 근거가 될 수 있었습니다.

주판단기는 6개 링크의 진폭·위상이 100 ms 사이에 얼마나 바뀌는지 계산한 뒤, 현재 2초를 MLP와 여러 시간 폭의 1차원 CNN으로 읽습니다. **보조판단기**는 같은 CSI 인코더로 현재와 직전 2초·4초 구간을 각각 읽고, 구간 사이의 특징 차이까지 묶어 최대 6초의 흐름에서 보행을 다시 확인합니다.

직전 문맥을 포함한 판정은 다른 방에서 오경보를 늘리기도 했습니다. G304 V2를 G308에 바로 적용했을 때 주경로와 주·보조 결합은 모두 보행 4/5구간을 찾았지만, **정적 오경보는 22/302에서 39/302로 늘었습니다.** 보조 경로의 보행 후보를 다시 걸러야 했습니다.

V5에서는 기존 두 경로를 고정하고, 두 점수·이력 유무·물리 특징 24개를 입력받는 **482개 파라미터의 보정기**를 학습했습니다. V6에서는 보행 후보를 정적으로 뒤집으려는 상황에서 이력이 부족하거나 경계에 가까우면 **판단 보류(U)**를 반환하는 규칙을 추가했습니다. 같은 V6 모델과 규칙을 두 강의실에 적용했습니다.

V6 정적·보행 판단 흐름

입력	**2초 CSI 변화** 진폭·위상 100 ms 차분 · 100 Hz · 결측 마스크		
판단	**주판단기** 현재 2초 · MLP + 다중 스케일 1D CNN · 128 → 64 → 2	**보조판단기** 현재·직전 2초·4초 · 특징 차이를 묶어 640 → 64 → 2	
재검사	**공통 보정기** 두 점수 + 이력 유무 + 물리 특징 24개 · 27 → 16 → 2		
출력	**S 정적** 분명한 보행 없음	**D 보행** 분명한 실내 이동	**U 판단 보류** 보행 취소 판단의 경계·이력 부족

V6의 보류 규칙은 V5 보정기의 가중치를 유지한 채 추가했습니다.

같은 4인 녹화에서 정적과 보행 비교

그림은 **같은 G304 4인 녹화**에서 골랐습니다. 자리에 머문 6초와 실제로 걸은 6초의 웹캠 장면, 그리고 대표 부반송파 그룹의 6개 링크를 한 눈금에 놓았습니다. 유효 시점의 로그 진폭 차분 RMS는 정적 **0.067**, 보행 **0.121**로, 이 두 예시에서는 보행 구간의 변화가 컸습니다. 표의 성능은 전체 기존 시험 구간에서 계산했습니다.

공통 V6는 G304에서 보행 **9/12구간(75%)**, G308에서 **4/5구간(80%)**을 검출했습니다. G308의 동일한 307개 구간에서 G304 V2를 바로 적용했을 때와 비교하면, 보행 검출 4/5는 유지하면서 **정적 오경보가 39건에서 9건으로** 줄었습니다. G304에서는 비교군이 맞혔던 보행 한 구간을 더 놓쳤습니다.

G304는 날짜를 분리했고, G308은 같은 원본 녹화 안에서 학습과 시험 구간을 나눴습니다. G308은 모두 정적으로 분류해도 전체 정확도가 98.37%인 시험입니다. 그래서 **보행 검출률·정적 오경보율·전체 보류율**을 함께 봤고, 보류한 보행도 검출률의 분모에 넣었습니다.

공통 V6 기존 시험 · 각 구간 2초

평가	G304	G308
보행 검출	9 / 12 · 75.00%	4 / 5 · 80.00%
정적 오경보	0 / 202 · 0.00%	9 / 302 · 2.98%
전체 보류	1 / 214 · 0.47%	8 / 307 · 2.61%



G304 같은 4인 녹화의 정적·보행 장면과 실측 CSI

04 기대와 한계

방과 장비가 바뀌면 무엇이 무너졌나

다른 장비의 데이터에서도 동작하는지 보기 위해 **공개 EHUNAM의 단일 링크 입력**을 기존 모델 형식에 맞췄습니다. 가중치와 정규화를 고정한 상태로 주 수신기 47,510구간을 평가했습니다. 동적 검출은 **7,010/33,098구간(21.18%)**이었고 정적 오경보율은 0.32%, 보류율은 1.67%였습니다.

외부 자료의 동적에는 기계 작동도 포함됩니다. 자체 자료의 동적은 사람의 보행입니다. 링크 수와 장비, 공간, 정답의 의미가 함께 달라졌으므로 검출 저하의 원인을 하나로 특정하기 어렵습니다. **가중치를 그대로 옮기는 방식으로는 안정적인 외부 감지가 어려웠습니다.**

자체 시험에도 놓친 보행이 남았습니다. G304에서 보행을 놓친 한 2초 구간의 대표 링크 묶음 RMS는 0.115로, 검출한 보행 시연의 0.121에 가까웠습니다. G308에서 놓친 보행은 직전의 연속 구간이 없어 보조판단기에 충분한 이력을 줄 수 없었습니다. **변화량이 큰 보행도 놓칠 수 있고, 보조 경로의 도움은 연속 기록을 확보할 수 있는지에 달려 있었습니다.**

카메라 없이 어디까지 감지할 수 있나

이번 실험에서 기대할 수 있는 것은 **공간을 촬영하지 않고 빈 방 여부나 뚜렷한 보행을 확인하는 감지 기능**입니다. 장비와 배치가 정해진 공간에서는 인원 범주를 보조 정보로 쓸 가능성도 봤습니다. 감지 단계에서 영상 입력을 빼면 집 안의 일상을 촬영하는 부담을 줄일 수 있습니다.

가장 아쉬웠던 점은 **같은 녹화에서는 높던 점수가 새 회차에선 크게 떨어졌다는 것**입니다. 2초 구간을 많이 만들어도 서로 같은 방·채널·배치에서 이어진 관측입니다. 인원수마다 독립 녹화가 적은 상태에서는 모델이 사람 수와 함께 나타난 세션의 특징을 외우기 쉬웠습니다. 보행 시험도 G304 12구간, G308 5구간이어서 한 구간의 오답이 검출률을 크게 바꿉니다.

다시 실험한다면 같은 방과 장비에서 **인원수·자리·행동을 번갈아 바꿔 반복 수집**하고, 방이나 채널을 바꿀 때는 다른 조건을 고정해 비교하겠습니다. 새 날짜와 새 방에서 모은 원본 회차를 별도 시험으로 남겨, 정적 오경보와 놓친 보행, 판단 보류를 함께 보겠습니다. 그래야 집을 촬영하지 않고도 필요한 변화를 얼마나 꾸준히 포착하는지 판단할 수 있습니다.

주막 / AI 창작·커뮤니티 기획

AI로 만든 메뉴를, 내 주막에서.

1인 기획·개발 | 2026년 중순 ~ 현재
게임 프로토타입 개발 · AI 서비스 설계 | Unity · C# · Simulation

AI로 나만의 메뉴 만들기

국밥의 그릇과 술병의 모양에도 플레이어의 취향이 담기길 바랐습니다. 자신이 만든 메뉴를 손님에게 팔 때 **창작이 경영의 일부**가 된다고 생각했습니다.

심즈 4·시티즈 스카이라인의 사용자 콘텐츠 커뮤니티를 참고했습니다. **AI 메쉬 생성**으로 제작의 진입 장벽을 낮추고, **제작·판매·공유**를 잇는 서비스를 기획했습니다.



주막 내부와 영업 공간

메뉴 제작 도구

플레이어가 술·국밥·전을 고르고 재료와 그릇을 설명하는 도구를 기획했습니다. 꽃무늬 사발의 국밥처럼 자기 취향에서 제작을 시작합니다.

Meshy로 메쉬와 텍스처를 생성하고, 형태를 확인하며 수정하도록 설계했습니다. 크기·재질·용량을 **게임 규격**에 맞춘 뒤 메뉴명과 분류를 연결해 판매할 계획입니다.

메뉴 제작과 공유

구상	**플레이어의 메뉴** 종류·형태·재료·참고 이미지	
제작	**AI 메쉬 생성** 형태 확인	**텍스처·규격 조정** 표면·크기·용량 확인
적용	**메뉴 등록** 에셋과 메뉴 데이터 연결	**주막 영업** 조리·서빙·판매에 사용
공유	**커뮤니티 게시** 미리보기·제작자·버전	**다른 주막에서 활용** 내려받기·설치·메뉴 적용

기획 단계입니다. AI 제작·게임 적용·공유 **연결이 남아** 있습니다.

Meshy Text to 3D 공식 문서: https://docs.meshy.ai/en/api/text-to-3d

제작에서 공유까지

만든 메뉴를 바로 게시하고 다른 사람의 메뉴를 내려받는 **커뮤니티**를 기획했습니다. 제작에 익숙하지 않은 플레이어도 마음에 드는 메뉴로 주막을 바꿀 수 있습니다.

에셋과 메뉴 설명·식별자·버전을 하나의 콘텐츠로 묶을 계획입니다. 업데이트나 삭제가 기존 **저장**에 미치는 영향도 함께 다룰 과제입니다.

AI 제작 비용

AI 생성 비용을 누가 부담할지 두 경로로 나누었습니다. 이용자 본인의 **Meshy** API를 연결하거나, 서비스가 생성을 중계하고 **제작 비용**을 받는 방식입니다.

작업별 예상 비용과 진행 상태를 알리고 실패·취소·재요청을 구분할 계획입니다. 계정·과금 연결과 결과물의 이용·재배포 조건은 아직 확정하지 않았습니다.

AI 메뉴 제작의 비용 부담 경로 · 검토안

경로	비용 부담	서비스에서 구성할 일
이용자 API 연결	이용자 본인의 Meshy 사용량	계정 연결·작업 상태·사용량 안내
서비스에서 제작 비용 수취	서비스가 생성 비용을 지불하고 제작 비용 회수	작업별 견적·요청 기록·실패 및 재시도 처리

Meshy API 시작 안내: https://docs.meshy.ai/en/api/quick-start

하루의 영업과 성장

차림표·재료·직원·가구를 준비하고 손님을 받는 하루를 구성했습니다. 마감에는 매출과 이탈 손님, 재료비·봉급·유지비를 보고 다음 날의 운영을 바꿉니다.

좌석만 늘려서는 해결되지 않도록 부뚜막·배식대·동선을 연결했습니다. 재정 유예와 84일 영업·최종 **정산**을 구성했으며 장기 균형은 조정 중입니다.

주막의 하루와 성장

준비	**차림표·재료** 가격과 준비량 선택	**직원·공간** 직무와 동선 정리
영업	**맞이·주문** 대기와 손님 상태	**조리·서빙** 주방·배식대·좌석 연결
마감	**장부 확인** 수입·비용·포기 원인	
개선	**다음 영업 준비** 보충·채용·리모델링	

운영의 결과가 다음 날의 준비와 공간 변경으로 이어지도록 구성했습니다.

직접 일하고, 직원에게 맡기기

직원에게 영업을 맡기면서 주모가 E로 조리·운반·대화를 돕도록 구성했습니다. 밀린 주문에 직접 개입하는 선택이 운영 결과에 드러나야 한다고 보았습니다.

직무와 능력·체력·만족도·봉급·휴식을 연결했습니다. 운반 목적지와 작업 중 직무 변경도 처리해 개인의 상태가 전체 영업에 반영되도록 했습니다.

사람이 머무는 주막

함께 앉고 대화하며 다시 주문하는 손님을 구성했습니다. 기다림은 아이콘과 시계로 보여주고, 모임의 합산 **결제**와 가까이서 읽는 생활 대사를 연결했습니다.

주막의 성장은 가구 추가와 주방·좌석의 재배치로 이어집니다. 계절과 밤낮을 다듬고 있으며 역사적 인물·사건은 장기 기획으로 두었습니다.

주막 / 구현 기반·AI 연결 설계

AI로 만든 메뉴를, 내 주막에서.

1인 기획·개발 | 2026년 중순 ~ 현재
게임 프로토타입 개발 · AI 서비스 설계 | Unity · C# · Simulation

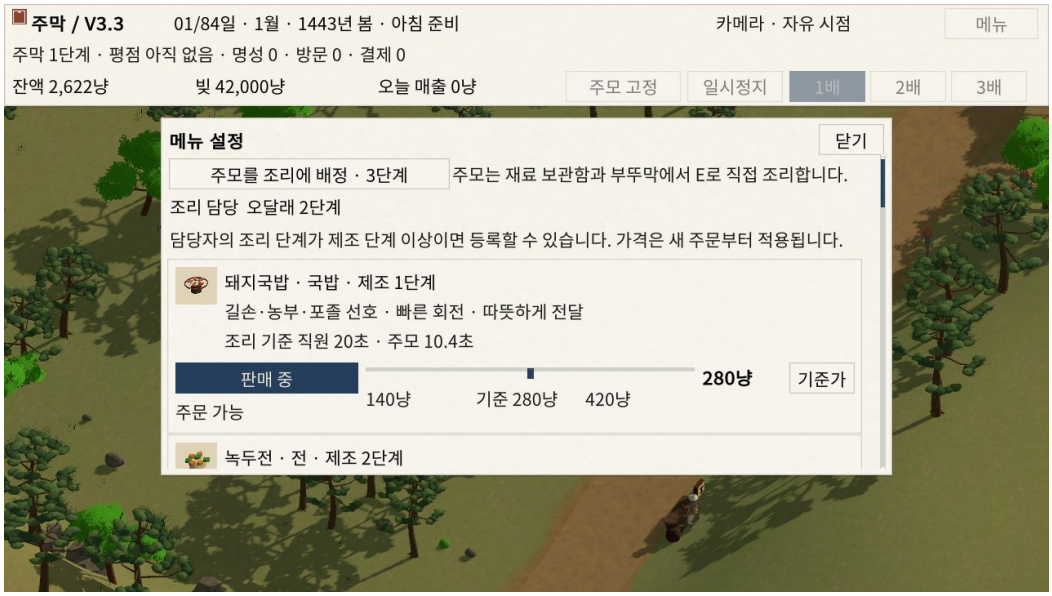
현재 구현과 AI 연결 범위

기본 메뉴의 선택·가격·조리·주문·서빙을 연결했습니다. 재료와 조리 공간, 직원 상태를 함께 다뤄 메뉴가 영업 안에서 작동하도록 만들었습니다.

현재 메뉴 종류는 고정되어 있습니다. **사용자 메뉴**를 받으려면 식별자·에셋·판매 규칙을 읽는 계층이 필요하며, 생성 API·설치·**커뮤니티**·결제는 아직 미연결입니다.

구현 기반과 AI 연결 범위

영역	현재 구성	다음 연결
게임 메뉴	기본 메뉴 선택·가격·조리·서빙	사용자 메뉴 식별자와 판매 규칙 연결
3D 표현	개발자가 적용한 게임 에셋·공간 표현	AI 생성 결과의 검사·게임 규격 변환·불러오기
운영·저장	직원·주문·회계·게임 상태 저장	사용자 콘텐츠 버전·참조 저장
AI·커뮤니티	창작 도구와 공유 흐름 기획	Meshy 연동·게시·다운로드·비용 처리



기본 메뉴·가격·조리 설정

생성물을 게임 메뉴로 연결하기

생성 서버가 **요청 ID**와 진행 상태를 관리하고, 완료한 메쉬는 크기·폴리곤·텍스처·용량을 확인하는 구조를 설계했습니다. 결과를 게임용 콘텐츠로 변환해 배포할 계획입니다.

에셋의 모양과 가격·재료·조리 규칙은 분리합니다. 식별자와 버전으로 주문·**저장**에 연결하고, 파일이 없을 때의 기본 표현과 복원 규칙도 마련하고자 합니다.

생성물 확인→테스트 주막에서 판매→공유·설치 순서로 검증할 계획입니다. 편집 허용 범위와 구체적 수치는 후속 기획입니다.

서비스와 게임의 연결

서비스	**전용 제작 화면** 설명·이미지·메뉴 분류	**생성 작업 관리** 요청 ID·상태·제작 비용
AI 제작	**Meshy API** 메쉬·텍스처 생성	**게임 규격 확인** 크기·용량·재질·배치 기준
콘텐츠	**메뉴 패키지** 식별자·버전·에셋·설명	**공유 자료실** 게시·다운로드·업데이트
게임	**Unity 콘텐츠 적용** 메뉴 데이터와 시각 표현 연결	**현재 운영 기반** 주문·조리·서빙·저장

기획 단계입니다. AI 제작·게임 적용·공유 **연결이 남아** 있습니다.

기획을 조작 규칙으로

어떤 기능을 넣을지보다 실제로 어떻게 조작할지를 정했습니다. 포인터 중심 줌, 가운데 드래그 이동, 오른쪽 드래그 회전과 WASD 주모 이동을 구분했습니다.

화면을 가리는 벽을 처리하고 재고·장부는 필요할 때 열도록 바꿨습니다. **피드백**과 수정 사항은 결정표로 관리했습니다.

주문부터 식사까지

주문 단계와 담당자·설비·자원 예약을 연결했습니다. 직원과 주모가 같은 주문을 맡아도 음식이나 자원을 **중복**으로 가져가지 않도록 구성했습니다.

직무 변경·경로 막힘·재료 부족 시 작업의 유효성을 확인하고 예약을 해제하거나 재배정합니다. 음식 전달 뒤 **결제**와 상 정리까지 연결했습니다.

한 주문이 손님에게 도착하기까지

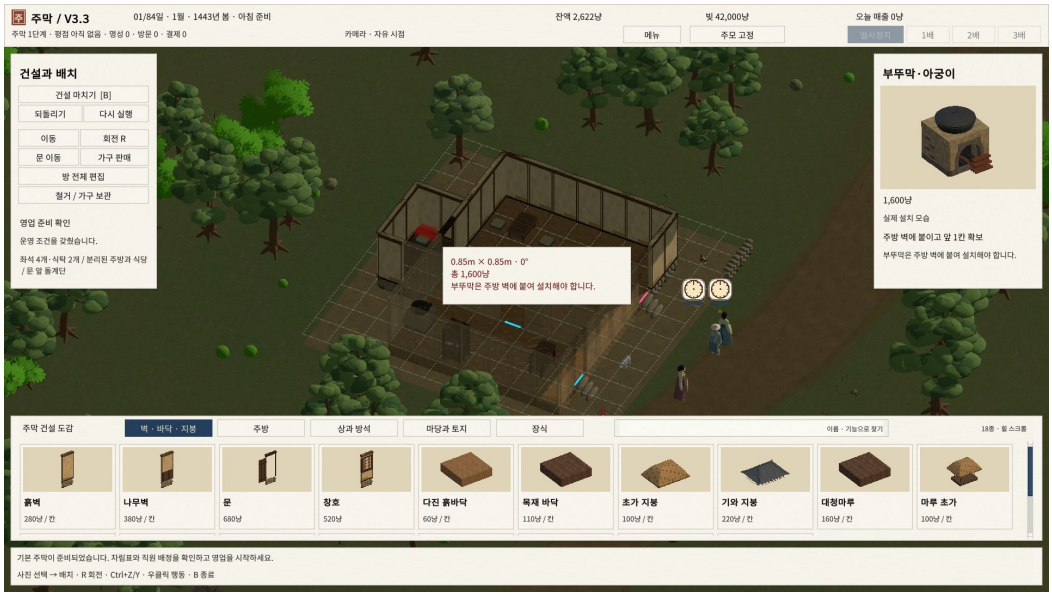
접수	**손님·주문** 메뉴·수량·대기 상태	
준비	**재료·부뚜막 예약** 수량과 작업 담당자 관리	
전달	**조리·배식대** 완성 음식과 인계	**주모·직원 운반** 목적 좌석과 동선
종료	**식사·결제·정리** 좌석을 다시 사용할 수 있는 상태	

작업 취소·변경 시에는 관련 예약과 담당 관계도 함께 정리하는 흐름입니다.

운영을 이어가는 건설

사진 도감에서 가구를 고르고 크기·가격·방향·접근 공간을 확인하게 했습니다. 방과 벽·문·바닥·지붕을 편집하고 큰 가구의 끝부분도 선택할 수 있게 다듬었습니다.

임시 운영 부족은 경고, 겹침과 미소유 땅은 차단으로 나누었습니다. 실패와 Undo/Redo는 비용·보관·방·배치를 함께 복원하고, 개점 시 운영 조건을 다시 검사합니다.



건설 도감과 가구 배치

조선의 생활감과 화면

한지와 목재, 처마·마루·장독으로 조선 주막의 생활감을 구성했습니다. 보유 에셋과 공개 자료를 활용하고 게임에 맞춰 표현을 수정했습니다.

밤낮·계절 속에서도 손님과 통로가 읽히도록 검토했습니다. 글자 크기·입력 감도·키 설정·카메라·가림 처리와 걷기·착석·식사 동작을 함께 다듬었습니다.



가을의 주막과 주변 환경

운영 상태를 저장하기

정산·빚·미납·캠페인 진행을 **저장**에 연결했습니다. 데이터를 검사해 임시 파일에 쓴 뒤 기존 파일을 교체하고, 건설 복원은 현금·가구·방 설정까지 확인했습니다.

자동 규칙 검사와 계단 이동·착석·가구 편집·군중 성능을 나누어 검토했습니다. 장시간 영업과 조작 품질, 전체 성장 균형은 계속 다듬고 있습니다.

현재 구성과 이어갈 개발

구분	현재 구성한 범위	다음 확인
운영	직원·주모·음식·회계 연결	장기 성장과 재정 균형
건설	배치·회전·복원·개점 검사	복합 편집과 사용성
화면·동작	계절·카메라·연속 동작 개선	군중 성능과 최종 검수

현재 상태와 다음 과제

· 기본 메뉴·직원·건설·회계·저장을 연결한 프로토타입입니다. **AI** 제작 도구와 **커뮤니티**는 설계 단계입니다.

· **Meshy** 생성·**사용자 메뉴** 설치·공유·결제는 미연결입니다. API 연결과 제작비 수취 방식을 검토하고 있습니다.

· **게임 규격**·재배포 조건·메뉴 규칙을 정하고, 메뉴 한 개의 제작→판매부터 검증할 계획입니다.

· 장시간 영업·군중 성능·조작·성장 균형은 조정 중입니다. 영상과 체험판은 추후 추가합니다.

주막 / 기획과 개발 판단

AI로 만든 메뉴를, 내 주막에서.

1인 기획·개발 | 2026년 중순 ~ 현재
게임 프로토타입 개발 · AI 서비스 설계 | Unity · C# · Simulation

직접 만드는 플레이어

문제 3D 제작 지식이 없으면 메뉴 창작에 참여하기 어렵습니다.

판단 AI 제작→주막 판매→공유를 게임의 확장 방향으로 정했습니다.

구현 메뉴 운영 기반을 개발하고 도구·콘텐츠·**커뮤니티** 연결을 기획했습니다.

검토 메쉬 외에 메뉴 식별자·조리 규칙·**저장** 참조가 필요함을 확인했습니다.

결과 창작 기능의 범위를 정했습니다. 생성·판매·공유 통합은 **다음 단계**입니다.

외형과 경영 규칙을 분리

문제 외부 메쉬는 규격이 다르고 현재 메뉴 종류는 고정되어 있습니다.

판단 AI 에셋과 메뉴 규칙을 별도 계층으로 연결할 구조를 잡았습니다.

구현 생성 상태·규격 검사·**메뉴 데이터**·버전 참조를 나눴습니다.

검토 현재 주문·**저장** 코드와 새 연결 계층의 경계를 확인했습니다.

결과 기존 영업에 **사용자 메뉴**를 추가할 구현 범위를 정했습니다.

주모가 개입하는 운영

문제 주모와 직원의 역할에 차이가 있어야 운영을 선택할 수 있습니다.

판단 자동 영업을 유지하며 주모가 병목을 돕게 했습니다.

구현 조리·서빙 직무와 E 상호작용·운반·대기를 연결했습니다.

검토 직접 개입과 배정에 따른 주문·음식·동선을 비교했습니다.

결과 직접 일하기와 운영 조정을 함께 선택하게 했습니다.

문과 동선을 영업 조건으로 만들다

문제 건물을 배치해도 주모가 인물 사이에 갇히거나 손님·직원의 길이 끊기는 문제가 있었습니다.

판단 손님은 외부에서 좌석까지, 직원은 외부·주방·식당을 오가며 **각 방의 문을 통과**하도록 요구했습니다.

구현 벽·가구 충돌과 인물끼리의 회피를 나누고, 문 위치를 바꾸면 손님과 직원의 이동 경로를 다시 검사하도록 했습니다.

검토 V3.3 Unity 자동검사 기록 · 2026.10.06

결과 건설 중의 배치 검사와 개점 시의 영업 조건을 나눴습니다. 손님과 직원이 문을 지나 좌석에 도착하고 퇴장하는 **두 PlayMode 사례가 통과**했습니다.

어드벤처 게임 / AI 동료·대화 경험 기획

게임 속 동료와, 내 말로 대화하기.

1인 기획·개발 | 2026년 중순 ~ 현재
게임 프로토타입 개발 · AI 서비스 설계 | Unity · C# · Gameplay

로보를 AI 동료로 만든 이유

탐험에서 도움을 받는 **로보**와 집에서는 이야기를 나누고 싶었습니다. 정해진 선택지 대신 자기 말로 질문할 때 관계가 더 개인적으로 남는다고 생각했습니다.

안전가옥에서 세계관·스토리를 묻고 **개인적인 고민**도 나누는 **GPT-6 Luna** 대화를 기획했습니다. 로보의 이름·말투·표정과 현재 이야기의 맥락을 유지하는 것이 목표입니다.

자유 대화의 활성화와 용량은 게임 속 GPU에 연결했습니다. 탐험의 보상과 동료의 기능, **API 운영비**를 함께 설계했습니다.



소년과 로보의 세이프하우스

세계관 속에서 대화하는 로보

개인적인 고민을 나눌 때도 **로보**의 이름·말투·성격을 유지하도록 기획했습니다. 대화가 관계와 몰입으로 이어지되 필수 진행과 결말을 임의로 바꾸지는 않게 합니다.

최근 대화·**관련 기억**·퀘스트·방문 지역·전투 결과 중 공개된 정보를 문맥으로 구성할 계획입니다. 아직 모르는 사건은 숨기고 공략은 **진행도**에 맞는 힌트로 제공합니다.

텍스트와 표정·몸짓·기계음부터 연결합니다. 실제 한국어 응답에서 캐릭터 유지·정보 한계·지연을 확인할 과제가 남았습니다.

GPT-6 Luna 공식 모델 안내: https://developers.openai.com/api/docs/models/gpt-6-luna

GPU로 여는 AI 대화

탐험에서 낮은 확률로 얻은 **GPU**를 로보에 장착해야 **자유 대화**가 열리도록 기획했습니다. 찾아온 부품으로 동료의 기능을 넓히는 일이 파밍과 귀환의 목적이 됩니다.

GPU는 활성화 권한과 대화 용량을 나타내는 **가상 아이템**입니다. 실제 응답은 GPT API 서버에서 생성하며, 입력과 전달 답변의 토큰으로 사용량을 관리할 계획입니다.

고급 GPU는 매우 드문 전리품이자 유료 상품으로 두고 **API 운영비**를 충당하고자 합니다. 필수 스토리·고정 선택 대사는 분리하며, 용량·확률·가격과 월정액 여부는 미확정입니다.

GPU와 AI 대화

획득	**탐험에서 발견** 낮은 확률의 GPU 파밍	**상점에서 구매** 고급 GPU의 유료 판매 계획
활성화	**로보에 GPU 장착** 등급·잔량에 따른 자유 대화 권한	
대화	**게임의 대화 경험** 세계관·스토리·개인적인 고민	**실제 응답 생성** 운영 서버를 통한 GPT-6 Luna
운영	**게임 용량 정산** 입력·전달 답변의 사용량	**실제 비용 관리** 제공자 청구·기억·재시도 비용

가상 **GPU**는 권한·용량을 표현합니다. 실제 모델·정산·결제는 **미연결**입니다.

다음 접속에도 이어지는 기억

다음 접속마다 처음 만난 동료가 되지 않도록 **장기 기억**을 기획했습니다. PC에 원문·요약·검색 인덱스를 두고 최근 대화와 **관련 기억**만 골라 모델에 전달하는 구조입니다.

삭제는 원문과 관련 요약·검색 기록에 함께 반영할 계획입니다. 필요한 문맥은 외부 서버에 전달되며, 현재의 **세션 이력**과 **장기 기억**은 구분합니다.

실제 GPT 응답을 먼저 연결한 뒤 기억의 검색 품질과 비용을 검증할 예정입니다. **장기 기억**의 형식·검색 방식은 아직 미확정입니다.

다음 모험으로 이어지는 귀환

장비·탄창·식량을 준비해 탐험하고, 더 수색할지 탈출할지 판단하는 흐름을 만들었습니다. 돌아오면 **정산**에서 가져온 물품을 확인합니다.

재료는 제작·수리와 농사·요리·전력·창고·방 확장으로 이어집니다. 현재 하나의 탐험 맵과 거점에서 반복 출발의 연결을 다듬고 있습니다.

모험과 거점의 반복 흐름

준비	**장비·가방·탄창** 목표와 운반 여유 선택	
탐험	**수색·기록 수집** 물품을 찾는 동안의 위험	**교전·회피** 위치·시야·소리 판단
귀환	**탈출·정산** 가져온 물품과 진행 반영	
성장	**제작·수리·생활** 설비·전력·로보 정비	
재출발	**다음 탐험 목적** 필요한 자원과 새 지역	

기능이 각각 존재하는지를 넘어, 다음 탐험을 준비하는 이유가 생기는지를 **기준**으로 기획했습니다.

시야와 소리로 판단하는 전투

고정 탑뷰에서 마우스 방향이 시야와 조준에 이어지도록 했습니다. 전방과 장애물에 따라 적을 인식하고 낮은 자세·엄폐·엿보기를 활용하게 구성했습니다.

총성과 위치, 탄창·회복 자원을 함께 고려하는 전투를 기획했습니다. 수색 중에도 시간이 흐르며 교전·우회·도주를 선택할 수 있게 개발하고 있습니다.

무엇을 가져갈 것인가

격자 가방과 회전 배치, 주머니·장비·탄창 잔탄을 연결했습니다. 판매품과 수리 부품, 탄약·치료품이 같은 공간을 두고 경쟁하도록 기획했습니다.

수색 중 위험이 유지되고 **로보**의 운반·보호 공간도 선택에 영향을 줍니다. 실제 획득 속도와 물품의 가치는 반복 플레이로 조정할 과제입니다.



보급품 수색과 격자 가방

함께 일하고 대화하는 동료

따라오기·대기·운반·조사와 집의 요리·수리를 **로보**의 역할로 구성했습니다. 실제로 함께 일하는 동료여야 대화에서도 함께 지낸 관계가 느껴진다고 보았습니다.

구조·가동·명령·운반·일부 집 작업을 구현했습니다. 이 행동들은 현재 게임 규칙으로 동작하며, GPT 대화는 **GPU** 활성화·용량과 함께 별도로 연결할 계획입니다.

어드벤처 게임 / 대화 구현·AI 서비스 설계

게임 속 동료와, 내 말로 대화하기.

1인 기획·개발 | 2026년 중순 ~ 현재
게임 프로토타입 개발 · AI 서비스 설계 | Unity · C# · Gameplay

대화 화면과 통신 기반

가까운 **로보**에게 말을 걸면 별도 카메라와 렌더 텍스처로 대화 화면을 엽니다. 창을 닫을 때 원래 입력과 화면으로 복원하고 글자 출력·표정·몸짓·기계음 반응을 구성했습니다.

UTF-8 **NDJSON** 이벤트와 **요청 ID**·순서로 수신·표시·완료·취소·오류를 관리합니다. 취소한 응답이 새 대화를 덮지 않게 하고, 실제 표시한 답변의 기록이 끝나야 다음 요청을 보냅니다.

현재 기본 실행은 고정 대화입니다. 합성 통신 검사는 연결 기반을 확인한 것이며 실제 GPT·GPU 차감·**장기 기억**은 아직 **미연결**입니다.

대화 처리 흐름

표현	**Unity 대화 화면** 클로즈업·입력·반응 연출
세션	**응답 상태 관리** 수신·표시·완료·취소·오류
통신	**중계 클라이언트** NDJSON 이벤트·요청·순서 확인
기록	**실제 전달 기록** 표시된 답변·식별자·기록 재시도

UI·세션·통신·**전달 기록**을 구현했습니다. 실제 GPT·정산·**장기 기억**은 미연결입니다.

게임과 GPT를 잇는 서버

서버가 인증·**GPU** 권한·요청 크기를 확인하고 GPT 요청을 중계하는 구조를 설계했습니다. 로보의 성격과 공개된 세계관, 최근 대화·**관련 기억**을 문맥으로 구성할 계획입니다.

GPT-6 Luna 응답은 텍스트·반응·완료 이벤트로 게임에 전달합니다. 현재 세션의 표시·취소·기록을 활용하고, 구매·발급·잔액은 세이브와 별도의 운영 데이터로 관리하려 합니다.

실제 모델·기억·권한·정산 서버는 아직 **미연결**입니다. 통합 뒤 캐릭터 유지·**스포일러**·회상·지연과 끊김·재시도를 확인할 계획입니다.

AI 대화 서비스

게임	**대화 UI·현재 진행** 질문·현재 지역·공개 사건	**기억 검색안** 최근 대화·관련 기록
서버	**인증·GPU 권한** 장착·잔량·요청 한도	**문맥 구성** 로보 성격·공개 설정·기억
모델	**GPT-6 Luna** 캐릭터에 맞는 응답 생성	
전달	**현재 대화 기반** 스트림 표시·표정·취소	**기록·정산 연결** 전달 결과·잔량·다음 기억

모델·문맥·기억·권한·**정산** 서버는 연결할 단계입니다.

대화 용량과 실제 비용

플레이어의 입력·전달 답변과 시스템 지침·기억을 포함한 실제 청구량은 다릅니다. 전달 데이터에 게임·제공자의 입력·출력 토큰 필드를 나누어 마련했습니다.

실제 사용량 수집과 카드 차감은 아직 **미연결**입니다. 발급 출처·등급·잔량·**요청 ID**를 서버에서 관리하고, 중복·부분 전달·취소의 정산 규칙을 연결해야 합니다.

무료 발급량·사용률·문맥과 답변 길이·실패·기억 처리 비용을 측정할 계획입니다. 가격과 손익은 이 사용 데이터를 바탕으로 정하고자 합니다.

로보 AI 기능의 구현 범위

영역	현재 구성	다음 연결
대화 표현	클로즈업·글자 출력·표정·몸짓·기계음 기반	실제 GPT 응답과 반응 연동
중계 통신	요청 식별·이벤트 순서·취소·오류 처리	운영 서버·GPT-6 Luna 연결
대화 기록	세션 이력·실제 표시 답변의 전달 기록	접속을 넘는 기억 저장·검색·삭제
GPU	일부 등급의 로컬 드롭 규칙	활성화 권한·인증된 발급·잔량·차감
비용·판매	사용량 구분 필드·경제 검토 문서	실제 사용량 수집·결제·중복 정산 방지

외형보다 플레이 연결을 우선

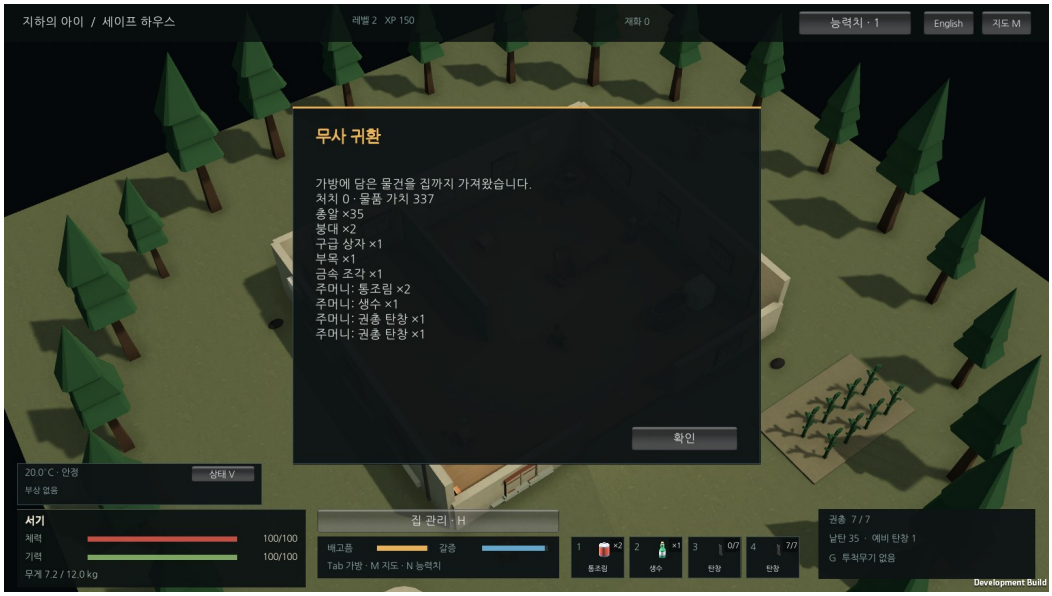
세계관·서사와 카메라·이동·시야·캐릭터를 만든 뒤 개발 우선순위를 바꿨습니다. 외형 수정을 멈추고 한 맵에서 준비·수색·교전·탈출·제작·재출발을 잇는 데 집중했습니다.

필요한 소품은 무료 에셋에서 골라 적용했습니다. 미술 작업이 플레이 기능의 연결을 계속 늦추지 않도록 범위를 정했습니다.

탐험 결과를 거점에 반영하기

가방과 **로보**의 운반 물품을 귀환 **정산**에 반영했습니다. 가져온 재료는 제작·수리와 충전·농사·수확·요리·방 확장에 연결했습니다.

같은 **저장**의 반복 출발과 두 번째 귀환, 제작·수리를 확인했습니다. 준비된 상태의 기능 검사와 정상 시작의 성장 속도·장기 균형은 나누어 검토합니다.



두 번째 탐험 뒤 귀환 정산

명령을 실제 작업으로

조사를 맡기면 **로보**가 접근해 물품을 차례로 공개하고 플레이어가 가져가도록 했습니다. 명령·위험·경로·배터리·부위 상태를 확인하고 집 작업은 모듈·재료·전력을 사용합니다.

상자에 도착해도 손끝이 닿지 않아 조사가 멈췄습니다. 기능은 도착·정지·시야·안전·시간으로 판단하고 팔 동작은 표현으로 분리해 흐름을 복구했습니다.

로보 작업의 진행과 중단

명령	**조사·운반·집 작업** 대상과 필요한 역할 선택	
조건 확인	**접근·시야·안전** 경로와 도착 상태	**자원·모듈** 배터리·전력·재료
실행	**작업 시간·물품 상태** 진행·명령 변경·위험 중단	
반영	**결과와 피드백** 발견 물품·운반·설비 결과	

명령·자원에 따른 게임 규칙으로 동작합니다.

성공과 실패를 저장하는 규칙

귀환·사망·체포·강제 종료를 구분해 휴대품 손실, 보호 공간과 경험치·레벨·해금·퀘스트를 처리했습니다. 가방·장비·**로보**·창고 사이에서 물품 소유가 **중복**되지 않게 구성했습니다.

장비·탄창·설비·**정산**의 **저장**과 복원을 확인했습니다. 사용자 저장과 시험 저장을 분리하고, 사망·중단 뒤 이어하기를 점검했습니다.

에셋을 게임 기능에 맞추기

구급상자 모델·가방 아이콘과 조리 설비 외형을 교체했습니다. 필요한 모델·텍스처만 가져와 충돌·상호작용·**저장** 구조를 유지하고 회수·조리 연결을 확인했습니다.

외형을 유지하며 시야·추적·수색·운반·귀환 오류를 우선 수정했습니다. 정상 시작의 성장 속도와 반복 탐험·전술·장기 균형·전체 서사는 계속 다듬을 과제입니다.

현재 기능과 장기 기획

분야	현재 구현·연결한 범위	다음 개발
모험	수색·교전·탈출·정산	반복 탐험의 밀도와 균형
거점·로보	명령·운반·설비 작업	경로·자원·반복 플레이
서사·대화	기록·고정 대사·진행 기반	전체 서사·실시간 자유 대화

현재 상태와 다음 과제

· 탐험·귀환·거점과 **로보**의 대화 UI·세션·중계 클라이언트를 개발했습니다. 기본 실행은 고정 대화입니다.

· 실제 **GPT-6 Luna**·**장기 기억**·GPU 권한·사용량 정산·결제는 아직 미연결입니다.

· GPU 발견·장착이 **자유 대화**의 활성화 조건입니다. 초기 무료량·**등급별 용량**·확률·가격은 미확정입니다.

· 전체 서사·장기 플레이·균형·교전 성능과 체험판은 개발·검토 중입니다. 영상은 추후 추가합니다.

어드벤처 게임 / 기획과 개발 판단

게임 속 동료와, 내 말로 대화하기.

1인 기획·개발 | 2026년 중순 ~ 현재
게임 프로토타입 개발 · AI 서비스 설계 | Unity · C# · Gameplay

대화와 모험의 보상

문제 자유 대화의 이용량에 따라 **API 운영비**가 누적됩니다.

판단 GPU 발견·장착·용량과 고급 카드 판매를 연결했습니다.

구현 캐릭터·경제·비용 기획과 **사용량** 필드를 나누었습니다.

검토 등급은 용량 차이이며 실제 추론은 GPT 서버가 맡도록 설계했습니다.

결과 파밍·관계·비용을 함께 기획했습니다. 발급·판매·**정산**은 **미연결**입니다.

받은 답변과 본 답변

문제 취소·단절 시 수신한 답변을 모두 읽은 것은 아닙니다.

판단 수신·표시·**전달 기록**과 취소·실패를 나누었습니다.

구현 ID와 **실제 표시**된 텍스트를 기록하고 다음 요청을 제어했습니다.

검토 세션·중계 코드와 합성 통신을 확인했습니다. 실제 GPT 검증은 남아 있습니다.

결과 기억·**정산**에 전달 결과를 연결할 기반을 구현했습니다.

나갈 이유와 돌아올 이유

문제 수집·전투만 반복하면 모험을 이어갈 목적이 약해집니다.

판단 밖에서는 물자·단서를, 집에서는 성장·관계를 얻도록 했습니다.

구현 탐험을 귀환 **정산**과 제작·수리·생활에 연결했습니다.

검토 가져온 물품이 다음 준비와 재출발에 쓰이는지 확인했습니다.

결과 수집·생존·성장과 동료를 하나의 순환으로 정리했습니다.

손끝 때문에 멈춘 조사

문제 로보가 도착해도 손끝 접촉 조건 때문에 조사가 멈췄습니다.

판단 기능의 진행과 팔의 시각 표현을 분리했습니다.

구현 도착·정지·시야·안전·시간을 조사 조건으로 삼았습니다.

검토 접근·조사와 명령 변경·위험에 따른 중단을 확인했습니다.

결과 수색 흐름을 복구했습니다. 전체 지형의 자동 조사는 검증 중입니다.

첫 출정의 완료 기준을 재출정까지 잡다

문제 탐험과 거점 기능을 이어서 플레이하려면 실패·귀환·저장 뒤에도 진행 상태가 맞아야 했습니다.

판단 원본 v1과 기존 저장을 보존하면서 첫 출정·귀환·저장·재출정까지 이어지는 실행본을 요구했습니다.

구현 기존 에셋과 저장을 유지한 적용본에서 수색·전투·탈출·귀환·재출정을 이어서 실행하고, 저장과 재시작 뒤 상태를 비교했습니다.

검토 AppliedV1_08 첫 순환 자동 실행 기록

결과 자동 실행에서 두 번 사망한 뒤 같은 저장으로 세 번째 출정에 생환했습니다. 연구로 늘린 **가방 16→20칸**도 재시작 뒤 보존됐습니다.