NYS
盧玧璇

ENGINEERING NOTES · 2026
NOH YOONSEON

끝까지 종료되는 흐름을 만드는 엔지니어

문제는 제가 정의하고, 실행은 AI와 나누고,
결과는 끝까지 검증합니다.

for 넷마블네오 · 2026 AI 인턴십 AI Game Maker

노윤선 | 동덕여자대학교 컴퓨터학과 · 2027.02 졸업예정

010-7397-9722 · lucy7noh@gmail.com

github.com/lucy7noh · lucynoh.co.kr

* 본 자료는 넷마블네오 공식 자료가 아닌 개인 지원 포트폴리오입니다.

CONTENTS

02 프로필
03 PLAY LOG 게임에서 본 재미
04 IDEA 반응형 몬스터 AI
05 AI와 일하는 다섯 단계
06 LOG-01 AI Agent 종료 조건
07 LOG-02 반복 작업 자동화
08 LOG-03 사용자 반응 운영
09 LOG-04 처음 쓰는 기술 익히기
10 LOG-05 시스템을 상태로 분해
11 LOG-06 배워서 나누기
12 AI 협업 로그
13 인턴십 5개월 계획

PROFILE



노윤선

盧玧璇 · 010-7397-9722 · lucy7noh@gmail.com · lucynoh.co.kr

EDU	동덕여자대학교 컴퓨터학과 · 2023.03 ~ 2027.02 졸업예정 · 인공지능 A0 · 빅데이터처리 A+
AI	LangGraph · LangChain · RAG · Claude / Gemini API · Claude Code · Cursor · LangGraph Studio
BUILD	Python · FastAPI · React · TypeScript · Spring Boot · Docker · Kubernetes
AWARD	가비아CNS 웹사이트 공모전 2위 · 한국경영컨설팅학회 우수상 · 교내 NPC 해커톤 우수상(기획)

ACTIVITY INDEX — AI Game Maker 관련 순

01 Proovy AI Agent
AI 파트 2명 중 1명

02 WordPress LLM 자동화
2h → 10m · 운영 중

03 가비아CNS 공모전
참여형 콘텐츠 · 2위

04 meetwhen-k8s
처음 쓴 K8s · HPA 2→5

05 DB IFC AXIOFLOW
업무를 상태로 분해

06 멋사 백엔드 운영진
Django 독학 · 세션

07 한이음 ICT 멘토링
적응형 문진 흐름

08 LG Almers 9기
147만 건 · 검증 설계

// 끝까지 종료되는 흐름을 만드는 엔지니어

PLAY LOG

협력하고, 상황에 따라 전략을 바꾸는 재미

즐겨 한 게임: 모두의마블 · 배틀그라운드 · 제5인격 / 최근 1년 출시작: 몬길: STAR DIVE (2026.04 출시 · 초반 스토리 · 전투 구간 플레이)

WHY GAMES · 제5인격

생존자로 플레이하며 감시자의 움직임에 맞춰 팀원과 역할을 그때그때 나눠야 했습니다. 같은 맵도 선택에 따라 결과가 달라졌습니다.

SO I BUILD

재미를 소비하는 데서 나아가, 이용자의 선택에 따라 달라지는 경험을 직접 구현하고 싶어졌습니다.

REVIEW · 몬길: STAR DIVE (넷마블)

작품성	원작 몬스터 길들이기의 수집 정체성을 지키며, 언리얼 엔진 5 연출과 서브컬처 감성으로 재해석
상업성	모바일·PC 전 세계 동시 출시로 접점 확대 · 몬스터 포획·합성과 캐릭터 소환이 장기 수집 동기
기술성	3인 파티 실시간 태그 액션을 쉬운 조작으로 풀어, 교체 타이밍으로 전략 차이를 냄
유저 재미	몬스터 행동을 읽고 어떤 캐릭터로 바꿔 대응할지 고르는 순간이 핵심

MY QUESTION

같은 패턴이 반복되면 대응이 "정해진 답"이 되지 않을까? → 다음 장

→ at NETMARBLE NEO

직접 플레이하며 재미가 생기는 순간과 식는 순간을 찾고, 그 지점을 기능 과제로 바꾸겠습니다

IDEA · REACTIVE ENEMY

플레이어의 습관에 반응하는 몬스터를 설계한다면

[제안 · 미구현] 몬길: STAR DIVE를 플레이하며 떠올린 아이디어를, 제가 Agent에서 써 온 상태 · 분기 방식으로 그려 봤습니다.

S1 관찰
플레이어 태그
선택 기록

→

S2 판단
자주 쓰는 대응
패턴 집계

→

S3 전환
행동 우선순위
변경

→

S4 검증
난이도 상한
· 종료 조건

WHY 같은 전투도 매번 다른 판이 되게 해, 대응이 정해진 답처럼 느껴지는 순간을 줄이려는 시도입니다.

RISK 몬스터가 너무 똑똑하면 불공정하게 느껴집니다. 전환 횟수와 난이도에 상한을 두는 것이 핵심입니다.

FROM Proovy의 재시도 상한·Fallback, 한이음의 응답별 질문 분기를 그대로 옮겨 온 설계입니다.

AI에게 맡길 것 / 내가 정할 것

· AI: 행동 패턴 후보 · 대사·연출 초안
· AI: 테스트 플레이 로그 요약

· 나: 전환 조건과 상한
· 나: "재미있다"의 통과 기준과 직접 플레이 판정

→ at NETMARBLE NEO

이런 반응형 적 AI를 작은 프로토타입으로 만들어 직접 플레이하며 재미를 검증하겠습니다

HOW I WORK WITH AI

AI와 일할 때 이 순서를 지킵니다

모두 실제 프로젝트에서 쓴 순서입니다. ref는 근거가 된 기록입니다.

01
상황 정의
무엇이 이상한지 먼저 관찰합니다

재시도가 끝나지 않음

ref: LOG-01

→

02
문제·목표
원인과 "끝난 상태"를 정합니다

종료 조건 부재 · 상한 +Fallback

ref: LOG-01

→

03
작게 쪼개기
검증 가능한 단위로 나눕니다

입력→생성→검증→게시

ref: LOG-02

→

04
AI 협업 세팅
맡길 일과 판단할 일을 나눕니다

초안은 AI, 기준은 나

ref: LOG-02

→

05
실행하며 익히기
처음 쓰는 도구는 만들며 배웁니다

LangGraph 2주 · K8s 첫 배포

ref: LOG-04

AI가 빨리 만들어도, 무엇이 문제인지와 "어디서 끝낼지"는 제가 정합니다.

결과물은 응답이 아니라 실제 동작(게시 완료 · 종료 확인 · 파드 확장)으로 확인합니다.

→ at NETMARBLE NEO

게임 기능 프로토타입도 문제 정의 → 작은 단위 → AI 협업 → 실행 검증 순서로 만들겠습니다

LOG-01 · AI AGENT

✓ VERIFIED

끝나지 않던 AI Agent에 "끝"을 설계했습니다

[2025.12 - 2026.02] [UMC 9기 · 전체 11인 팀, AI 파트 2명 중 1명] [Python · FastAPI · LangGraph · Claude API]

입력 유형 분기 → 변환 문서·이미지 → 이해 Vision LLM → 검색 RAG·pgvector → 판단 Review → 제어 Retry·Fallback

PROBLEM 기준 미달 답변이 끝없이 재시도되며 실행이 끝나지 않았습니다.

CAUSE LangGraph Studio 상태 추적으로 종료 조건 부재를 찾았습니다.

ACTION 재시도 상한·Fallback 추가, 실패 경로까지 실행해 종료 확인.

TEAM API 응답에 검토 결과·재시도 횟수를 남겨 다른 파트도 바로 테스트.

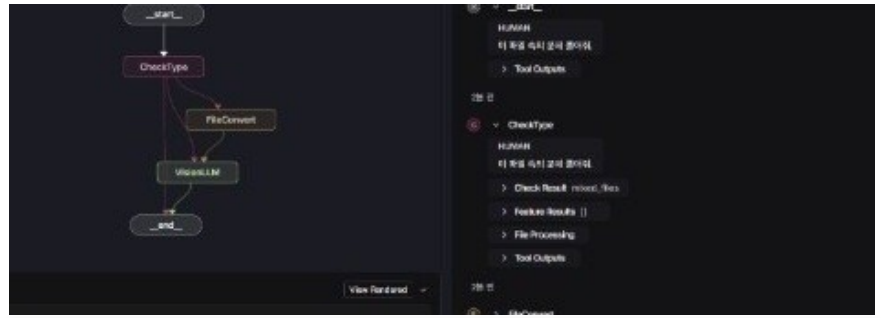


fig.1 LangGraph Studio · 노드별 상태 추적

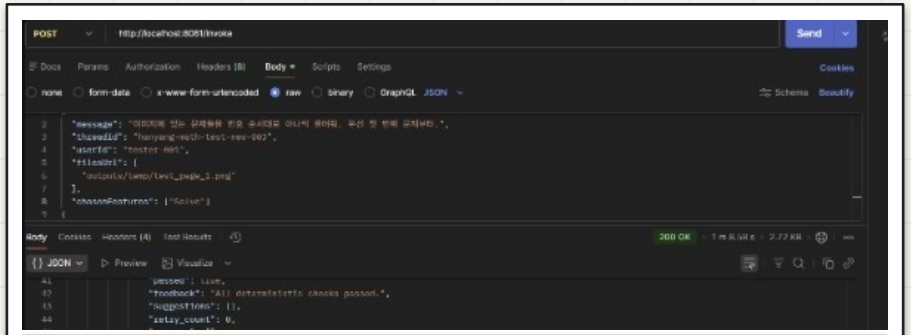


fig.2 Postman · retry_count · feedback 응답

AI에게 맡긴 것

- 문제 풀이·해설 생성
- 이미지·PDF 문제 OCR

내가 한 것

- "검토 통과"를 완료 기준으로
- 종료 조건·실패 경로 설계

→ at NETMARBLE NEO

AI로 만든 게임 기능도 "언제 멈추고 무엇을 통과해야 하는지"를 먼저 정의하고 구현하겠습니다

LOG-02 · AUTOMATION

✓ VERIFIED

2시간 걸리던 반복 작업을 10분으로 줄였습니다

[2025.09 - 운영 중] [개인 프로젝트] [Python · LangChain · Gemini API · WordPress REST API]

AS-IS 글 한 편에 자료 조사부터 게시까지 평균 2시간. LLM 결과는 분량·형식이 매번 달라 다시 고쳐야 했습니다.

TO-BE 입력 → 생성 → 형식 검증 → SEO 파싱 → 자동 게시로 쪼개고, 기준 미달이면 최대 2회 재생성하게 했습니다.

LIVE ISSUE 로컬에선 되던 게시가 운영에서 WAF에 막힘 → 보안은 두고 전용 엔드포인트·인증 경로로 복구했습니다.

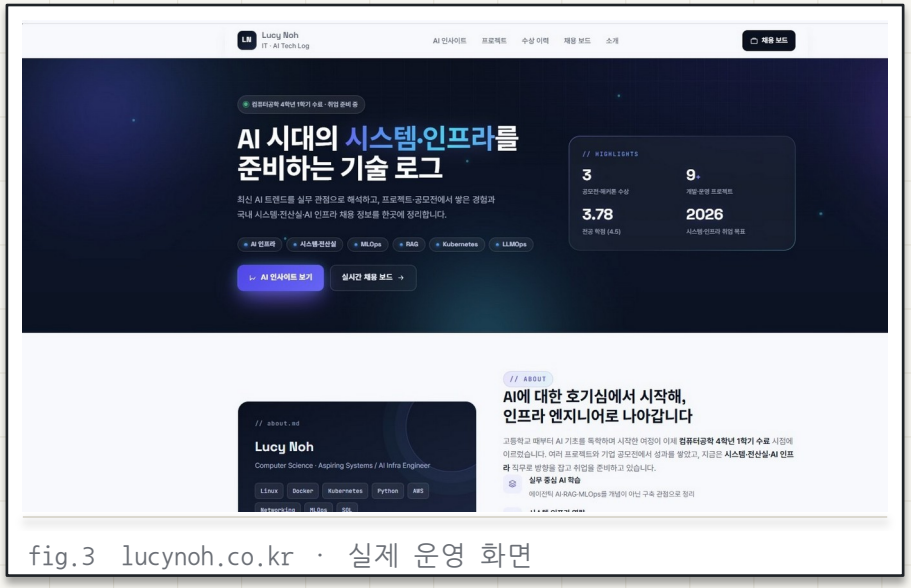


fig.3 lucynoh.co.kr · 실제 운영 화면

AI에게 맡긴 것

- 본문 초안 생성
- 참고자료 OCR · 요약

내가 한 것

- 분량·형식 통과 기준
- 게시 완료까지 검증

METRICS

2h → 10m **retry ≤ 2**

글 1편 발행 시간 · 기준 미달 시 재생성

→ at NETMARBLE NEO

QA·데이터 정리 같은 반복 업무를 AI 자동화로 줄이고, 통과 기준으로 결과를 검증하겠습니다

LOG-03 · USER RESPONSE

✓ VERIFIED

사용자가 머물고 공유하는 콘텐츠를 만들고 운영했습니다

[2025.06 - 2025.08] [가비아CNS 전국 대학생 웹사이트 제작 공모전 · 2인 팀] [React · Vercel]

BUILD 마스코트 Q&A와 전공 성향 테스트처럼 참여하고 결과를 공유하는 콘텐츠를 React로 만들어 배포했습니다.

JUDGE 방문자 수로 순위가 정해지는 대회라, 마감 전 시간을 홍보 80 : 개선 20으로 나누자고 제안해 합의했습니다.

ITERATE 같은 지적이 반복된 버튼 위치·글자 크기·모바일 공유 화면부터 고치며 반응을 다시 확인했습니다.

METRICS

136팀 중 2위
65개 대학 참가

1만 회+
누적 방문

1,125명+
사용자 유입

→ at NETMARBLE NEO

라이브 게임에서도 유저 반응이 반복되는 지점부터 개선하고, 지표로 효과를 다시 확인하겠습니다

LOG-04 · LEARN BY DOING

✓ VERIFIED

처음 쓰는 기술은 만들며 익혔습니다

[2026.07] [개인 프로젝트 meetwhen-k8s · 쿠버네티스 첫 사용] [FastAPI · Docker · Kubernetes · Prometheus]

DEPLOY FastAPI 앱을 Docker 이미지로 만들어 Deployment·Service·Secret·PVC로 배포했습니다.

PROBE liveness·readiness probe로 비정상 컨테이너 자동 재시작과 요청 수신 상태를 확인했습니다.

SCALE HPA 목표 CPU 50%에서 부하를 걸어 파드가 2개에서 5개로 늘어나는 것을 확인했습니다.

ALSO LangGraph는 2주 집중 학습 후 Proovy 워크플로에 적용했고, Django는 독학해 세션을 진행했습니다.

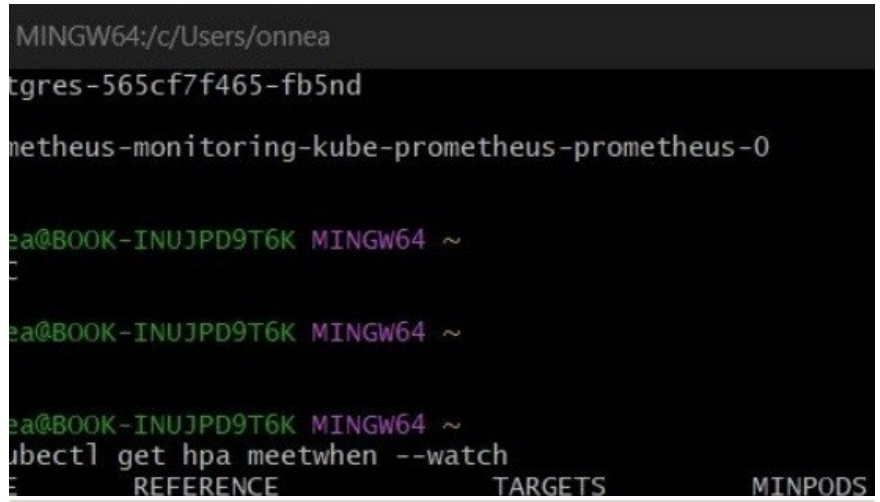


fig.4 kubectl · HPA 파드 확장 확인

METRICS

2 → 5 **2주**

HPA 파드 확장 · LangGraph 학습 후 적용

→ at NETMARBLE NEO

처음 쓰는 엔진·툴도 작은 기능을 직접 만들어 보며 빠르게 익혀 팀 작업에 연결하겠습니다

LOG-05 · DECOMPOSE

✓ VERIFIED

복잡한 업무를 상태 흐름으로 분해했습니다

[2025.09 - 2026.02] [제16회 DB보험금융공모전(DB IFC) · 팀 AXIOFLOW · 미수상] [LangGraph]

SIMULATION · 보험금 지급 처리 기간

전체 처리 기간 · · · · · · · · · 평균 47일

실제 가치를 만드는 시간 · · · 약 6시간 (나머지는 대기 · 재검토)

IDEA — 제조업 Line-Stop 개념을 가져와, 업무를 상태 전이로 모델링하고 멈춘 시간을 비용으로 보는 틀을 제안했습니다.

VERIFY — 생성형 AI의 추론 오류가 그대로 반영되지 않도록 자기 성찰(Self-Reflection) 검증 절차를 설계했습니다.

MY PART — LangGraph 분석 파이프라인 구현, 보험 업무 자료 조사, Line-Stop 이론 조사를 맡았습니다.

→ at NETMARBLE NEO — 게임 시스템도 상태와 흐름으로 분해해, 플레이가 끊기는 지점을 찾아 기획·구현 과제로 만들겠습니다

LOG-06 · TEAM

✓ VERIFIED

배운 것을 팀이 따라올 수 있게 나눴습니다

[2026.02 -] [멋쟁이사자처럼 동덕여대 백엔드 운영진 · 총무] [Spring Boot · Django]

CONTEXT 운영진이 4명에서 2명으로 줄자, 처음 다루는 Django를 독학해 Spring Boot와 함께 약 4개월간 주 2회 세션을 진행했습니다.

PROBLEM 명령어만 설명하자 개발 경험과 운영체제가 다른 부원들이 같은 단계에서 반복해 막혔습니다.

ACTION 단계마다 확인할 것과 정상 결과를 함께 보여주고, Windows·macOS 차이는 직접 재현해 Notion에 정리했습니다.

FOLLOW 현장에서 못 푼 질문은 확인 후 Discord로 공유하고, 반복 질문은 다음 자료에 반영했습니다.

METRICS

4 → 2
운영진 축소 상황

주 2회
약 4개월 세션

8명
백엔드 부원 오류 함께 분석

→ at NETMARBLE NEO

통합 직무 팀에서 기획·개발 경계를 넘어 제가 익힌 AI 활용법을 동료가 쓸 수 있게 공유하겠습니다

AI COLLAB LOG

AI에게 맡긴 것과 제가 판단한 것

실제로 써 본 도구만 적었습니다. 결과는 응답이 아니라 동작으로 확인했습니다.

작업	AI에게 맡김 (도구)	제가 판단	확인한 결과
AI 튜터 Agent	풀이·해설 생성 (Claude API · LangGraph)	검토 통과 기준 · 종료 조건	3가지 경로 모두 종료
블로그 발행	본문 초안 · OCR (Gemini API)	분량·형식 기준 · 재생성 상한	2h → 10m, 실제 게시
상태 추적	노드별 실행 기록 (LangGraph Studio · Langfuse)	어느 노드가 원인인지	종료 조건 부재 발견
코드 작성	구현·수정 보조 (Claude Code · Cursor)	구조 · 리뷰 반영 여부	PR 리뷰 후 병합
코드 리뷰	자동 리뷰 (CodeRabbit)	지적 중 반영할 것 선택	팀 컨벤션 유지

→ at NETMARBLE NEO

인턴십의 AI 도구도 맡길 일과 판단할 일을 나누고, 지시문·기준을 템플릿으로 남겨 팀이 다시 쓰게 하겠습니다

APPLY · 인턴십 5개월

이 순서로 AI와 게임을 만들어 보겠습니다

내부 환경을 모르는 지원자의 계획입니다. 배치 후 멘토와 확인해 다시 조정하겠습니다.

MONTH 1
익히기
팀 도구 · 엔진 · 작업 방식
ref: LOG-04

→

MONTH 2
정의하기
플레이가 끊기는 지점 찾기
ref: LOG-05

→

MONTH 3
만들기
반응형 적 AI 같은 작은 프로토타입
ref: IDEA

→

MONTH 4
검증하기
플레이 반응 · 버그 확인
ref: LOG-03

→

MONTH 5
나누기
방법을 팀 문서로
ref: LOG-06

기획 × 개발
기획 의도를 작은 기능 단위로 쪼개 직접 구현해 보고, 동작하는 화면으로 팀과 의견을 맞춥니다.

AI 협업 템플릿
반복되는 지시와 검증 기준을 템플릿으로 정리해, 같은 품질의 결과를 다시 얻을 수 있게 합니다.

QA · 반복 업무
버그 재현 절차와 로그를 정리하고, 반복되는 확인 작업은 스크립트로 자동화합니다.

→ at NETMARBLE NEO

스스로 과제를 찾아 끝까지 동작하는 결과로 만드는 인턴이 되겠습니다

// 끝까지 종료되는 흐름을 만드는 엔지니어