

PROJECT LIMITLESS

중증 뇌병변 (뇌성마비) 장애인이 AI 와 함께 게임 개발의 장벽을 낮춘 방법

생성형 AI 를 기획 · 프로그래밍 · 디버깅의 개발
파트너로 활용해 Unity 6 기반 2D 턴제
RPG 를 직접 개발하고 있습니다 .

핵심 메시지

AI 는 저 대신 게임을 만든 것이 아니라 ,
제가 게임을 만들 수 있는 방법을
바꾸었습니다 .

접근성을 중요한 설계 원칙으로 삼되 , 무엇보다 재미있는
RPG 를 목표로 개발하고 있습니다 .

현재 구현된 실제 플레이 화면



캐릭터 생성

길 · 직업 · 외형 · 최종 능력치



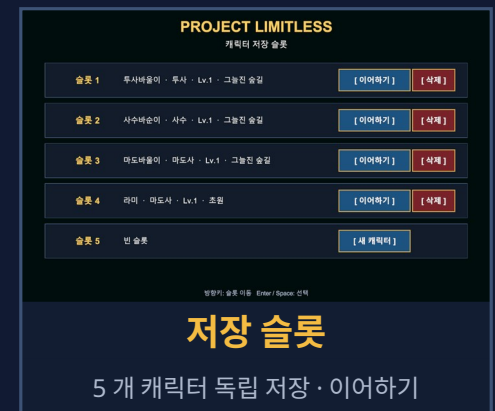
필드 탐험

몬스터 조우와 월드 이동



턴제 전투

반응속도보다 전략 중심



저장 슬롯

5 개 캐릭터 독립 저장 · 이어하기

※ 아래 화면은 Project Limitless 의 Unity Play Mode 실제 캡처입니다 .

Before → AI 협업 → After

텍스트만 읽어도 이해되고, 첨부 이미지로 실제 개발 과정을 증명합니다.

BEFORE

사례 1 저장 시스템 확장

캐릭터 1 명만
저장 가능

여러 캐릭터를 키우기 어려움

AI 협업

제가 요구사항 결정

5 개 캐릭터 독립 저장
손상 슬롯은 서로 분리

단일 저장 → 5 슬롯 → 위치 저장 → 삭제

AFTER



GitHub 증거 | Feature: 단일 슬롯 로컬 저장과 이여하기 → 5 개 캐릭터 저장 슬롯 → 월드 위치 저장과 슬롯 삭제

사례 2 실제 플레이에서 버그 발견 · 수정

광역 스킬 대상 없음
→ 입력 상태 꼬임

게임 조작이 어려워짐

직접 플레이로 문제 발견

상황 재정의
→ AI 와 원인 분석
→ 대상 검사와 복구 로직 추가

AI 결과를 그대로 쓰지 않고 검증



GitHub 증거 | Fix: 광역 스킬 대상 없음 입력 잠금 수정

AI 에게 코드를 부탁하기 전에 , AI 가 일하는 방법부터 설계했습니다 .

단순한 질문 도구가 아니라 , 프로젝트 원칙을 지키는 개발 파트너로 사용했습니다 .

AGENTS.md / CODEX_WORKFLOW.md 핵심 규칙

- ① 프로젝트 문서와 현재 상태를 먼저 확인
- ② 사용자가 정하지 않은 기획은 임의로 변경하지 않기
- ③ 기존 정상 기능과 사용자 수정 이미지를 보호
- ④ 직접 검증하지 못한 것은 검증했다고 말하지 않기

GitHub 저장소의 AGENTS.md 와 CODEX_WORKFLOW.md 에 이 규칙을 명시해 반복 작업의 기준으로 사용합니다 .

나의 AI 협업 흐름

내가 요구사항 결정



AI 가 문서 · 기존 코드 확인



AI 와 구현 / 수정



내가 Unity 에서 플레이 · 검증



문제 발견 시 다시 수정



Git commit + 현재 상태 문서 갱신

결론

AI 는 저 대신 게임을 만들지 않았습니다 .
제가 게임을 만들 수 있는 방법을 바꾸었습니다 .