

프로젝트 제안서



프로젝트 명 : 센서 게임 허브

팀명 / 소속 학과 : 큐보 / 컴퓨터공학과

제출일: 2025.09.10

팀원 명단 (학번, 이름, 역할) : 20201513 안지민

20190635 이재원

20201576 조민혁

지도 교수명 : 이시진 교수님

목차

1. 제안 개요
2. 프로젝트 일반 현황
3. 시장/기술 동향 분석
4. 제안 목표 및 수행 방향
5. 시스템/서비스 개발 계획
6. 사업 관리 계획
7. 지원 및 운영 계획
8. 기대 효과 및 결론
9. 부록 (Appendix)

1. 제안 개요

1.1 제안 배경

현대 게임 산업은 빠르게 성장하고 있지만, 아이디어를 실제 게임으로 구현하는 과정은 여전히 높은 기술적 장벽이 존재합니다. 프로그래밍, 그래픽 디자인, 게임 엔진 활용 능력이 요구되기 때문에 창의적인 아이디어를 가진 사람도 게임을 직접 제작하기는 쉽지 않습니다. 이로 인해 많은 좋은 아이디어들이 실제 게임으로 실현되지 못하고 사라지는 경우가 많습니다.

최근에는 생성형 AI 기술이 발전하면서 코드 작성이나 콘텐츠 제작 과정을 보조할 수 있게 되었고, 이는 게임 개발의 진입 장벽을 낮추는 데 큰 역할을 하고 있습니다. 그러나 무엇보다 중요한 것은 스마트폰에 내장된 센서(자이로스코프, 가속도계 등)를 활용한 몰입감 있는 게임 플레이 경험입니다. 기존의 터치 중심 인터페이스에서 벗어나, 몸의 움직임과 기기의 기울임에 따라 게임이 반응하는 방식은 플레이어에게 훨씬 더 직관적이고 흥미로운 상호작용을 제공합니다.

또한, 오늘날 스마트폰은 이미 전 세계적으로 충분히 보급되어 있어, 누구나 고성능 센서를 갖춘 기기를 손에 쥐고 있습니다. 따라서 별도의 장비 없이도 누구나 쉽게 몰입감 있는 센서 기반 게임을 제작하고 즐길 수 있는 환경을 마련하는 것이 매우 의미 있는 시도라 할 수 있습니다.

저희 프로젝트는 이러한 기술적 흐름을 반영하여, AI를 통한 게임 제작 보조와 스마트폰 센서를 활용한 몰입형 게임 플레이를 결합함으로써, 아이디어가 있는 누구든지 자신만의 독창적인 게임을 손쉽게 구현할 수 있는 환경을 제공하는 것을 목표로 합니다.

1.2 제안 목적

본 프로젝트의 최종 목적은 “누구나 상상한 아이디어를 센서를 활용한 몰입형 게임으로 즉시 구현하고, 바로 플레이할 수 있는 통합 플랫폼을 구축하는 것”입니다.

스마트폰에 기본적으로 내장된 자이로스코프, 가속도계 등의 센서는 단순한 터치 조작을 넘어, 사용자의 움직임을 그대로 게임에 반영할 수 있습니다. 기기를 기울이거나 흔드는 동작만으로도 캐릭터를 조종하거나 이벤트를 발생시킬 수 있기 때문에, 기존의 모바일 게임에서는 느낄 수 없는 현실감과 몰입감을 제공합니다. 저희 팀은 이러한 센서 기술을 적극적으로 활용하여, 누구나 손쉽게 새로운 형태의 게임 플레이를 경험할 수 있도록 하는 데 중점을 두고 있습니다.

또한, 생성형 AI는 사용자가 떠올린 게임 아이디어를 자연어로 설명하면 이를 코드와 리소스로 변환해주는 보조 도구로 활용됩니다. 복잡한 개발 지식 없이도 AI가 기본 구조를 자동으로 구성해주기 때문에, 사용자는 곧바로 센서를 활용한 직관적이고 몰입적인 게임 플레이를 구현할 수 있습니다.

결국 본 프로젝트는 게임 제작의 진입 장벽을 낮추고, 센서를 활용한 새로운 사용자 경험을 널리 확산시키는 것을 목표로 합니다. 이를 통해 비전문가에게는 아이디어를 곧바로 게임으로 구현해 체험할 수 있는 기회를 제공하고, 개발자에게는 센서 기반 게임의 프로토타입을 신속하게 제작할 수 있는 실험 환경을 지원합니다. 궁극적으로는 “누구든 아이디어만 있으면 직접 게임을 만들고 즐길 수 있는 환경”을 마련하여, 게임 제작이 일부 전문가만의 영역이 아니라 학생과 일반 사용자 등 누구에게나 열려 있는 즐거운 경험이 되도록 하는 것이 저희의 목적입니다.

1.3 제안 범위 및 전제조건

제안 범위:

- AI 기반 게임 생성 시스템: 자연어 요구사항을 분석하고, 게임 로직, UI, 센서 연동 코드가 포함된 완전한 웹게임(HTML/CSS/JS)을 생성하는 시스템 개발.
- 실시간 게임 플레이 플랫폼: PC 웹 브라우저에서 게임을 실행하고, 스마트폰을 컨트롤러로 사용하여 실시간으로 센서 데이터를 연동하는 플랫폼 개발.
- 클라이언트-서버 아키텍처: Node.js 기반의 백엔드 서버, 게임 클라이언트(PC), 센서 클라이언트(모바일) 개발.
- AI 지식 베이스(RAG) 구축: 프로젝트 관련 문서 및 코드를 AI가 학습하여 정확한 컨텍스트의 게임을 생성하도록 하는 시스템 구축.

전제조건:

- 사용자는 PC(게임 화면용)와 스마트폰(센서 컨트롤러용)을 모두 보유해야 합니다.
- PC와 스마트폰은 인터넷에 연결되어 있어야 합니다.
- 사용자의 디바이스(PC,스마트폰)는 WebSockets 및 최신 JavaScript를 지원하는 모던 웹 브라우저를 사용해야 합니다.
- 스마트폰의 브라우저는 Device Motion/Orientation API 접근 및 권한 획득이 가능해야 합니다.

1.4 수행 전략

본 프로젝트는 다음과 같은 핵심 전략을 통해 성공적으로 목표를 달성하고자 합니다.

1) RAG(Retrieval-Augmented Generation) 아키텍처 도입

범용 AI 모델의 한계를 극복하기 위해, 프로젝트의 코드베이스, 개발 가이드, 게임 예제 등을 벡터 데이터베이스에 저장합니다. AI는 게임 생성 시 이 데이터베이스를 참조하여, 프로젝트의 고유한 구조(e.g., `SessionSDK.js`)와 스타일에 맞는, 일관되고 완성도 높은 코드를 생성합니다.

2) WebSocket 기반 실시간 통신

PC와 모바일 간의 센서 데이터를 최소한의 지연 시간으로 전달하기 위해 `Socket.IO`를 활용한 실시간 양방향 통신 채널을 구축합니다.

3) 모듈화된 아키텍처 설계

AI 모듈, 세션 관리 모듈, 게임 스캔 모듈 등을 명확히 분리하여 개발 및 유지보수의 효율성을 높입니다. 특히, `SessionSDK.js`라는 클라이언트 라이브러리를 통해 게임과 서버 간의 통신을 표준화합니다.

4) 점진적 기능 개발 (Iterative Development)

1단계로 정적인 센서 게임 플레이 환경을 구축하고, 2단계로 단일 프롬프트 기반의 AI 게임 생성을, 3단계로 대화형 게임 생성 기능을 구현하는 점진적 방식으로 개발 리스크를 최소화합니다.

5) 애자일(Agile) 개발 방법론 적용

애자일 방법론을 기반으로 프로젝트를 진행합니다. 주 단위 혹은 스프린트 단위로 목표를 설정하고, 주기적인 피드백과 개선 과정을 거쳐 빠르게 프로토타입을 검증합니다. 이를 통해 변화하는 요구사항에 유연하게 대응하고, 완성도를 점차 높여가는 방법론을 사용하려고 합

니다.

1.5 팀 역할 및 협조 사항

본 프로젝트는 총 3명의 팀원으로 구성되어 있으며, 각자 맡은 역할에 따라 프로젝트를 공동 수행합니다.

◆ 역할 분담

- 팀장 / 프로젝트 매니저(PM) & 시스템 아키텍트
- 프로젝트 전체 일정 관리 및 진행 상황 점검
- 시스템 아키텍처 설계 및 주요 기술 선택
- 팀원 간 협업 조율 • 백엔드 & AI 담당
- 서버 구축 및 데이터베이스 설계
- AI 모델 연동 및 게임 생성 로직 구현
- API 및 서버 최적화
- 프론트엔드 & UI/UX 담당
- 웹 클라이언트 및 모바일 인터페이스 개발
- 센서 데이터 연동 기능 구현
- 사용자 경험(UX) 및 인터페이스(UI) 디자인

◆ 협조 사항

1. 지도 교수

- 프로젝트 전반적인 방향성 자문
- 기술적 난관 및 문제 해결 피드백 제공
- 주요 마일스톤별 진행 상황 검토

2. 오픈소스 커뮤니티 및 자료 활용

- 공식 문서, GitHub, reddit 등 커뮤니티 자료 적극 참고
- 최신 기술 동향 반영 → 프로젝트 완성도 향상

1.6 제안의 특징 및 차별성

1) '대화형' 센서 게임 생성 AI

단순히 명령을 수행하는 것을 넘어, AI가 사용자에게 역으로 질문하고 대화를 통해 요구사항을 구체화하여 게임을 완성하는 독창적인 방식을 제공합니다.

2) 코드 수준의 결과물 생성

추상적인 게임 로직이 아닌, 즉시 실행 가능한 완전한 `HTML/CSS/JavaScript` 코드를 생성하여 사용자에게 제공하고, 플랫폼 내에서 즉시 플레이할 수 있게 합니다.

3) RAG 기반의 '전문가' AI

본 프로젝트의 아키텍처와 코드 스타일을 학습한 AI가 마치 해당 프로젝트의 전문 개발자처럼 컨텍스트에 맞는 고품질 코드를 생성합니다.

4) 플랫폼리스(Platform-less) 웹 기반 접근

별도의 앱 설치 없이 PC와 모바일의 웹 브라우저만으로 모든 기능(게임 생성 및 플레이)을 이용할 수 있어 접근성이 매우 뛰어납니다.

1.7 기대 효과

교육적 효과

- 프로그래밍 지식이 없는 학생이나 일반인도 게임 제작 과정을 경험하며 컴퓨팅 사고력과 창의력을 증진시킬 수 있습니다.

개발 생산성 향상

- 게임 개발자가 센서 연동 게임의 프로토타입을 수 분 내에 제작하고 테스트해볼 수 있어, 개발 초기 단계의 시간과 비용을 획기적으로 절감할 수 있습니다.

새로운 콘텐츠 생태계

- 사용자가 직접 만든 독창적인 센서 게임들이 플랫폼에 축적되면서, 새로운 형태의 사용자 주도형 콘텐츠 생태계 형성 가능성을 제시합니다.

2. 프로젝트 일반 현황

2.1 팀 소개

◆ 팀원 역할

이재원(팀장)

- 팀장 / 프로젝트 매니저(PM) & 시스템 아키텍트
- 프로젝트 전체 일정 관리 및 진행 상황 점검
- 시스템 아키텍처 설계 및 주요 기술 선택
- 팀원 간 협업 및 업무 조율
- 주요 마일스톤 관리 및 산출물 검토

조민혁(팀원)

- 백엔드 & AI 담당
- 서버 구축 및 데이터베이스 설계
- AI 모델 연동 및 게임 생성 로직 구현
- API 개발 및 서버 성능 최적화
- 시스템 보안 및 확장성 고려

안지민(팀원)

- 프론트엔드 & UI/UX 담당
- 웹 클라이언트 및 모바일 인터페이스 개발
- 센서 데이터 연동 기능 구현
- 사용자 경험(UX) 및 인터페이스(UI) 디자인
- 반응형 UI 및 접근성 고려

2.2 주요 수행 분야 및 기술 역량

주요 수행 분야

- 생성형 AI, 실시간 웹 애플리케이션, 풀스택 웹 개발

보유 기술 역량

- AI/ML: LangChain.js, OpenAI API, Anthropic API를 활용한 RAG 파이프라인 구축 및 프롬프트 엔지니어링. Python 기반의 머신러닝 모델 경험.
- Backend: Node.js, Express.js를 이용한 RESTful API 및 웹 서버 구축. Socket.IO를 이용한 실시간 통신 서버 개발.
- Frontend: JavaScript(ES6+), HTML5, CSS3를 이용한 동적 및 반응형 웹 UI 개발. Canvas API 및 Three.js를 이용한 그래픽 렌더링.
- Database: Supabase를 활용한 벡터 데이터베이스 구축 및 쿼리.
- DevOps: Git, GitHub를 이용한 버전 관리. npm을 이용한 의존성 관리.

2.3 프로젝트 수행 일정 개요

- 총 수행 기간: 약 3개월 (12주)

주요 단계별 일정

1단계 (3주) - 기획 및 핵심 설계

- 요구사항을 구체화하고, 프로젝트의 기술적 기반을 완성하는 단계

2단계 (4주) - 핵심 기능 개발 구축

- 실시간 통신, 센서 데이터 처리 및 API 처리의 기본 구조를 구축하는 가장 집중적인 개발 단계

3단계 (3주) - AI 기능 및 대화형 시스템 구현

- 프로젝트의 핵심 차별점인 '대화형 생성' 기능을 집중적으로 개발하는 단계

4단계 (2주) - 통합 테스트 및 최종 완료

- 개발된 모든 기능들을 통합하여 테스트하고, 프로젝트를 최종 마무리하는 단계

3. 시장/기술 동향 분석

3.1 관련 기술 현황 (국내외 동향)

생성형 AI의 발전

- GitHub Copilot, cursor, claude 등 AI가 코드나 UI를 직접 생성하는 서비스가 상용화되며 개발 패러다임이 변화하고 있습니다. 본 프로젝트는 이러한 코드 생성 기술을 '게임'이라는 특정 도메인에 전문화하고, '센서'라는 독특한 입력 방식을 결합하여 기술적 트렌드를 선도합니다.

Low-Code/No-Code 플랫폼의 확산

- Bubble, Webflow 등 코딩 없이 웹사이트나 앱을 제작하는 플랫폼 시장이 급성장하고 있으며, 이는 개발의 민주화에 대한 높은 시장 수요를 방증합니다. 본 프로젝트는 이러한 노코드 트렌드를 게임 개발 영역으로 확장합니다.

웹 기술의 발전

- WebAssembly, WebGL, WebSockets 등 웹 기술이 고도화되면서, 이제 웹 브라우저는 네이티브 앱에 준하는 고성능 애플리케이션 실행 환경으로 자리 잡았습니다. 이는 별도 설치 없는 웹 기반 게임 플랫폼의 기술적 타당성을 뒷받침합니다.

3.2 벤치마킹 사례 / 유사 서비스 분석

Roblox / ZEPETO

- 사용자가 직접 게임이나 월드를 창작하고 다른 사용자와 공유하는 대표적인 메타버스 플랫폼. 강력한 커뮤니티와 제작 도구를 제공하지만, 학습 곡선이 존재하며 센서를 직접 연동하는 기능은 제한적입니다.

차별점: 본 프로젝트는 별도의 제작 도구 학습 없이 '자연어'만으로 게임을 생성하며, '모바일 센서' 연동에 특화되어 있습니다.

Scratch

- MIT에서 개발한 블록 코딩 기반의 교육용 프로그래밍 플랫폼. 코딩의 원리를 쉽게 배울 수 있지만, 텍스트 기반의 자유로운 창작에는 한계가 있으며, 모바일 센서 연동이 복잡합니다.

차별점: 블록 코딩의 제약을 넘어, 사용자가 상상하는 그대로를 텍스트로 서술하여 게임을 생성하는 높은 자유도를 제공합니다.

3.3 문제점 및 개선 필요성

기존 게임 개발 환경은 높은 진입 장벽을 가지고 있으며, 아이디어를 빠르게 프로토타이핑하기 어렵습니다. 특히 스마트폰 센서를 활용한 게임은 네이티브 앱 개발 지식과 복잡한 연동 과정이 필요하여 웹 기반으로는 시도조차 어려운 실정입니다.

AI 코드 생성 도구들은 범용적으로 설계되어 특정 프레임워크(e.g., SessionSDK.js)에 맞는 코드를 일관되게 생성하지 못하는 문제도 있습니다. 본 프로젝트는 이러한 문제들을 '대화형 AI'와 'RAG 아키텍처'를 통해 해결하고자 합니다.

3.4 SWOT 분석

Strength (강점)

- AI와 센서 연동을 결합한 독창적이고 혁신적인 아이디어.
- RAG 아키텍처를 통해 프로젝트에 특화된 고품질 코드 생성 가능.
- 웹 기반으로 별도 설치가 필요 없어 접근성이 매우 높음.

Weakness (약점)

- OpenAI, Anthropic 등 외부 API에 대한 의존성.
- PC와 스마트폰, 두 개의 기기가 필수적인 사용자 경험.

Opportunity (기회)

- 생성형 AI 및 사용자 제작 콘텐츠(UGC)에 대한 시장의 폭발적인 관심.
- 코딩 교육 및 창의력 증진을 위한 교육용 플랫폼으로의 확장 가능성.
- 새로운 장르의 '센서 게임' 시장 개척 가능성.

Threat (위협)

- LLM 기술의 빠른 발전으로 현재 모델이 단기간에 구식이 될 수 있음.
- 외부 API의 요금 정책 변경 또는 서비스 중단 리스크.
- 유사한 아이디어를 가진 거대 자본의 시장 진입 가능성.

4. 제안 목표 및 수행 방향

4.1 프로젝트 최종 목표

"사용자와의 멀티턴 대화를 통해 의도를 파악하고, 요구사항에 부합하는 센서 연동형 웹게임을 완전한 코드로 생성하여 즉시 플레이 가능한 상태로 제공하는 지능형 웹 플랫폼을 구축한다. 이를 통해 스마트폰의 보급률 증가에 따른 접근성 향상과 편의성을 극대화하고, 몰입감 있는 게임 환경을 제공하여 사용자 경험을 한층 강화한다."

4.2 세부 목표 (단계별)

1단계 (기반 구축)

- 실시간 세션 관리 및 센서 데이터 통신 프레임워크(SessionSDK.js 및 SessionManager.js)를 안정적으로 구축한다.

2단계 (핵심 기능 구현)

- AI 모델(LLM)을 연동하여, 싱글턴으로 기본적인 게임 로직을 생성하는 기능을 구현한다.

3단계 (고도화)

- RAG 파이프라인을 구축하고, 멀티턴을 통해 사용자의 요구사항을 구체화하고 게임을 완성하는 '대화형 생성' 기능을 구현한다.

4단계 (완성 및 배포)

- 생성된 게임을 저장하고, 동적으로 게임 목록에 추가하며, 사용자가 즉시 플레이할 수 있는 전체 워크플로우를 완성하고 테스트한다.

4.3 프로젝트 범위

개발 범위

- 백엔드: Express 기반 API 서버, Socket.IO 기반 실시간 통신 서버
- 프론트엔드: AI 대화형 게임 생성 UI, 동적 게임 허브 페이지, 모바일 센서 클라이언트 페이지
- AI: LangChain.js 기반 RAG 파이프라인, OpenAI/Anthropic LLM 연동 모듈
- DB: Supabase 벡터 DB 스키마 설계 및 연동, Supabase Auth

서비스 범위

- 사용자는 웹을 통해 AI와 대화하여 게임을 생성할 수 있다.
- 생성된 게임은 플랫폼 내에서 즉시 플레이 가능하다.
- 게임 플레이 시 스마트폰을 컨트롤러로 사용할 수 있다.

예외 범위

- 네이티브 모바일 앱(iOS/Android) 개발은 포함하지 않는다.
- 사용자 계정 시스템 및 생성된 게임의 영구 저장/수정 기능은 초기 버전에서 제외한다.
- AI가 생성하는 그래픽/사운드 리소스는 범위에 포함하지 않으며, 기본 제공되는 요소나 텍스트 기반으로 처리한다.

5. 시스템/서비스 개발 계획

5.1 시스템/서비스 개요

본 시스템은 크게 '게임 생성 시스템'과 '게임 플레이 시스템'으로 나뉩니다.

게임 생성 시스템

- 사용자가 'Interactive Game Generator' UI를 통해 AI와 대화하면, 백엔드의 AIGameGenerator가 RAG 아키텍처를 통해 LLM을 호출하여 게임 코드를 생성하고, 이를 public/games 디렉토리에 저장합니다.

게임 플레이 시스템

- 사용자가 Game Hub 페이지에서 특정 게임을 선택하면, PC와 모바일이 SessionManager를 통해 WebSocket으로 연결되어 모바일의 센서 데이터가 PC의 게임 로직에 실시간으로 반영됩니다.

5,2 아키텍처 구성 (H/W, S/W 포함)

H/W 구성

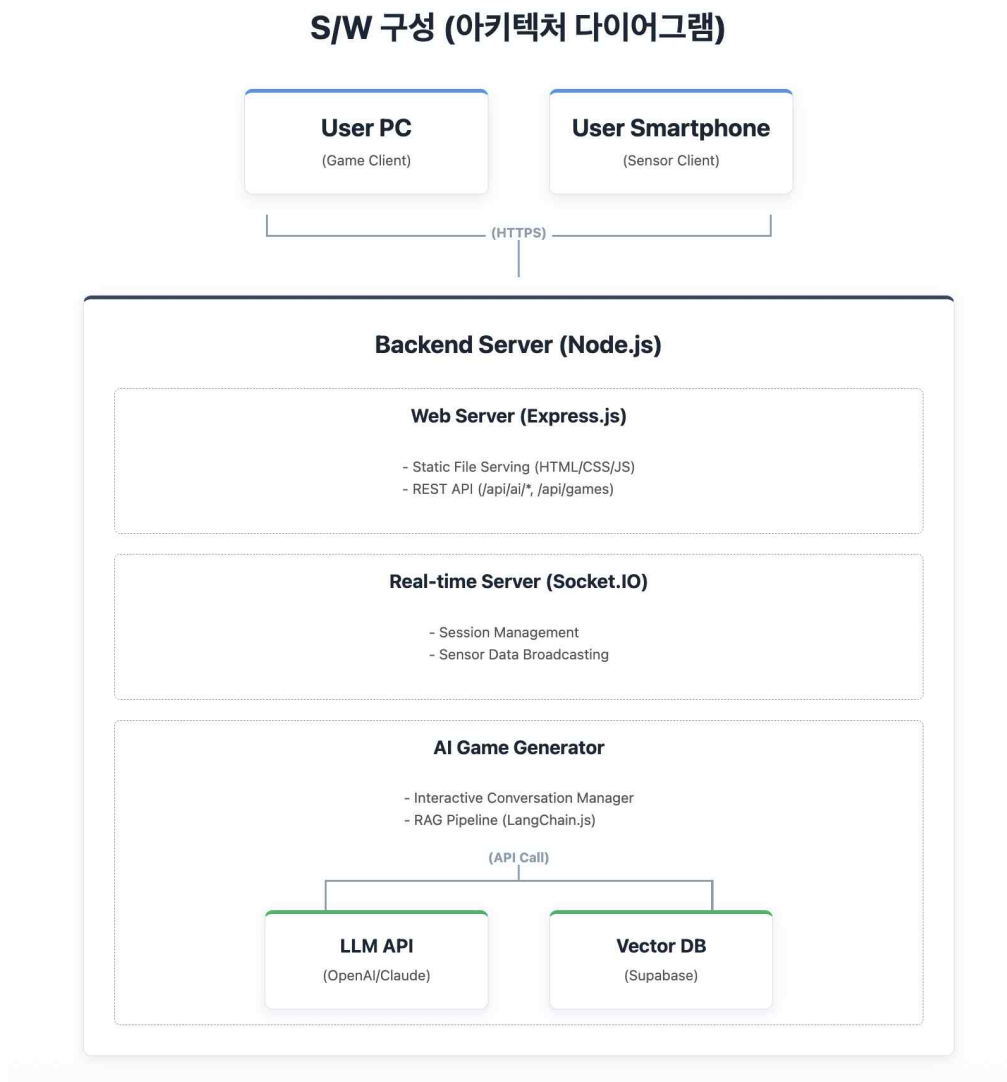
서버

- 클라우드 기반 가상 서버 (render) 또는 로컬 개발 PC

클라이언트

- 웹 브라우저가 설치된 PC 및 자이로/가속도 센서가 내장된 스마트폰

S/W 구성 (아키텍처 다이어그램)



5.3 기능 정의 및 주요 화면

대화형 게임 생성 페이지

- 기능 : AI와 채팅, 빠른 질문 버튼, 생성 진행 상태 표시, 최종 게임 생성 및 다운로드
- 화면 : 채팅 UI, AI 상태 표시등, 입력창, 전송 버튼

게임 허브 페이지

- 기능 ; 플레이 가능한 게임 목록 동적 표시, 게임 통계 표시, 개발자 도구 링크
- 화면 : 게임 카드 그리드, 검색/필터 기능

게임 플레이 페이지

- 기능 : 게임 캔버스 렌더링, 세션 코드 및 QR 코드 표시, 센서 연결 상태 표시
- 화면 : 중앙의 게임 캔버스, 상단의 상태 표시 바

모바일 센서 클라이언트 페이지

- 기능 : 세션 코드 입력, 센서 권한 요청, 센서 데이터 시각화, 연결 상태 표시
- 화면 : 코드 입력 필드, 연결 버튼, 센서 데이터 실시간 그래프

5.4 데이터 구조

Vector DB (Supabase - game_knowledge 테이블)

- id (uuid, PK): 문서의 고유 ID
- content (text): 분할된 문서 또는 코드의 원본 텍스트
- metadata (jsonb): 출처 파일 경로, 문서 타입 등의 메타 정보
- embedding (vector): content를 벡터로 변환한 값 (AI 검색용)

In-Memory Data (서버 메모리) - Session Object

- id (string): 세션 고유 ID
- sessionCode (string): 4자리 연결 코드
- game (object): 연관된 게임 정보
- hostSocket (object): PC 클라이언트의 소켓 정보
- sensors (Map): 연결된 센서(모바일) 클라이언트들의 소켓 정보
- state (string): 'waiting', 'ready', 'playing' 등 세션 상태

5.5 개발 방법론

애자일(Agile) 기반의 반복적 점진적 개발(Iterative and Incremental Development) 방법론을 채택합니다.

- 프로젝트를 2~3주 단위의 스프린트(Sprint)로 나누어 진행합니다.
- 각 스프린트마다 동작하는 작은 단위의 기능을 개발하고 테스트하여, 프로젝트의 불확실성을 줄이고 지속적인 피드백을 통해 완성도를 높입니다.
- 예: Sprint 1: SessionManager 구현 → Sprint 2: SessionSDK 및 기본 통신 테스트 → Sprint 3: LLM 연동 및 단일 프롬프트 생성 구현.

5.6 단계별 개발 계획

1단계 (3주) - 기획 및 핵심 설계

- SessionManager 및 SessionSDK 설계 및 초기 구현
- PC-모바일 간 WebSocket 연결 및 데이터 전송 구조 검증
- 기본 게임 템플릿 및 정적 게임 허브 페이지 설계

2단계 (4주) - 핵심 기능 개발 구축

- AIAssistant 모듈 개발 및 LLM API 연동
- DocumentEmbedder 및 update-embeddings.js 스크립트 구현
- Supabase 벡터 DB 구축 및 1차 RAG 파이프라인 구현
- 단일 프롬프트 기반 게임 생성 로직(AIGameGenerator) 개발

3단계 (3주) - AI 기능 및 대화형 시스템 구현

- InteractiveGameGenerator 모듈 개발
- 대화 흐름 관리 및 요구사항 누적 기능 구현
- 대화형 게임 생성 UI 프론트엔드 개발

4단계 (2주) - 통합 테스트 및 최종 완료

- 전체 시스템 통합 테스트 및 버그 수정
- 성능 최적화 및 안정화
- 최종 배포 및 검증 완료

6. 사업 관리 계획

6.1 프로젝트 일정 (Gantt Chart)

프로젝트 계획: 간트 차트

주요 단계별 일정

1단계 (3주) - 기획 및 핵심 설계

요구사항을 구체화하고, 프로젝트의 기술적 기반을 완성하는 단계

2단계 (4주) - 핵심 기능 개발 구축

실시간 통신, 센서 데이터 처리 및 API 처리의 기본 구조를 구축하는 가장 집중적인 개발 단계

3단계 (3주) - AI 기능 및 대화형 시스템 구현

프로젝트의 핵심 차별점인 '대화형 생성' 기능을 집중적으로 개발하는 단계

4단계 (2주) - 통합 테스트 및 최종 완료

개발된 모든 기능들을 통합하여 테스트하고, 프로젝트를 최종 마무리하는 단계

주차별 진행 계획 (총 12주)

Task	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12
1. 기획 및 설계 (W1-W3)												
요구사항 분석	요구사항 분석											
아키텍처 설계	아키텍처 설계											
2. 핵심 기능 개발 (W4-W7)												
실시간 통신 모듈				실시간 통신 모듈								
기본 게임 템플릿				기본 게임 템플릿								
AI 연동(단일)				AI 연동(단일)								
3. 기능 고도화 (W8-W10)												
RAG 파이프라인								RAG 파이프라인				
대화형 UI 개발								대화형 UI 개발				
대화형 생성 로직								대화형 생성 로직				
4. 통합 및 마무리 (W11-W12)												
통합 테스트											통합 테스트	
안정화/배포											안정화/배포	
최종 보고서											최종 보고서	

6.2 인력 투입 계획 (역할, 담당 업무)

이재원 (팀장 / 프로젝트 매니저 & 시스템 아키텍트)

- 프로젝트 총괄 및 일정, 리스크 관리
- 시스템 아키텍처 설계 및 주요 기술 스택 선정
- 팀원 간 업무 조율 및 협업 리딩
- 주요 마일스톤 관리 및 최종 산출물 검토

조민혁 (팀원 / 백엔드 & AI)

- Node.js, Express 기반 백엔드 서버 구축 및 데이터베이스 설계
- AI 모델 연동, RAG 파이프라인 구축, 게임 생성 로직 구현
- REST API 개발 및 서버 성능 최적화
- 시스템 보안 및 확장성 설계

안지민 (팀원 / 프론트엔드 & UI/UX)

- HTML, CSS, JavaScript 기반 웹 클라이언트 및 모바일 인터페이스 개발
- 사용자 경험(UX) 및 인터페이스(UI) 디자인
- WebSocket을 이용한 센서 데이터 연동 기능 구현
- 반응형 UI 및 웹 접근성 준수

6.3 예산 추정 (필요 자원:장비, 서버, API, 기타 비용)

본 프로젝트는 졸업 작품으로, 최소 비용으로 진행하는 것을 목표로 합니다.

H/W 자원

- 개인 보유 노트북 및 스마트폰 활용 (0원)

S/W 자원

- 서버 호스팅: Render 등 무료 티어 활용 (0원)
- LLM API
OpenAI API: 무료 크레딧 또는 소액 결제 (월 10\$ 내외 예상)
Anthropic API: 무료 크레딧 또는 소액 결제 (월 10\$ 내외 예상)
- Vector DB: Supabase 무료 티어 활용 (0원)

총 예상 비용: 개발 기간 중 약 50,000원 (API 테스트 및 사용료)

6.4 리스크 관리 방안

기술적 리스크

- 외부 API(LLM, DB)의 정책 변경 또는 중단
- 각 API 연동 부분을 추상화 계층으로 분리하여, 문제 발생 시 다른 서비스(e.g., 다른 LLM 모델)로 쉽게 교체할 수 있도록 설계

일정 지연 리스크

- 특정 기능 개발 난이도로 인한 지연
- 애자일 방법론에 따라 핵심 기능(MVP)을 우선 개발하고, 추가 기능은 우선순위를 조정, 주간 단위로 진행 상황을 점검하고 일정 재조정

품질 저하 리스크

- AI가 생성하는 코드의 품질이 낮거나 버그가 많을 경우
- RAG 시스템의 지식 베이스를 지속적으로 보강하고, 프롬프트 엔지니어링을 통해 생성 코드의 품질을 개선, 생성된 코드에 대한 자동 검증(Validation) 로직 강화

7. 지원 및 운영 계획

7.1 테스트 및 검증 방법

단위 테스트 (Unit Test)

- lint 등의 프레임워크를 사용하여 SessionManager, AIAssistant 등 백엔드의 주요 모듈별 핵심 기능의 정합성을 검증

통합 테스트 (Integration Test)

- AI 게임 생성 요청부터 파일 저장까지 이어지는 전체 파이프라인이 정상 동작하는지 검증

E2E 테스트 (End-to-End Test)

- 실제 사용 시나리오에 따라, 사용자가 게임을 생성하고, PC와 모바일로 접속하여 플레이하는 전 과정을 테스트

사용성 테스트 (Usability Test)

- 잠재 사용자(동료 학생 등)에게 시스템을 사용하게 하고, 피드백을 받아 UI/UX를 개선

7.2 사용자 교육 및 매뉴얼 계획

README.md

- 프로젝트의 개요, 기술 스택, 설치 및 실행 방법을 상세히 기술한 README.md 파일을 제공

인터랙티브 가이드

- AI 어시스턴트 자체가 대화형 매뉴얼 역할을 수행. 사용자가 "게임은 어떻게 만들어?"라고 질문하면 AI가 단계별로 안내

개발자 가이드

- /developer-guide 페이지를 통해 주요 기능(SDK 사용법, 디버깅 팁 등)에 대한 링크와 설명을 제공

7.3 유지보수/업데이트 계획

지식 베이스 업데이트

- 프로젝트 코드에 변경 사항이 생기거나 새로운 개발 가이드가 추가될 경우, `scripts/update-embeddings.js`를 실행하여 AI의 지식 베이스를 주기적으로 업데이트

의존성 관리

- `npm audit` 등을 통해 보안 취약점을 정기적으로 확인하고, 라이브러리 버전을 업데이트

버그 수정

- GitHub Issues를 통해 발견된 버그를 관리하고, 수정 사항을 반영하여 버전 업데이트

7.4 기술 이전/확장성 고려 사항

LLM 교체 가능성

- AIAssistant 모듈은 특정 LLM에 종속되지 않도록 설계. 향후 더 뛰어난 성능의 모델(e.g., GPT-5, 차세대 Claude)이 출시될 경우, 해당 모델의 API를 호출하도록 쉽게 교체 가능

서비스 확장성

- 현재는 게임 생성/플레이에 집중하지만, 향후 생성된 게임을 공유하고 평가하는 '커뮤니티 기능', 사용자가 직접 만든 게임으로 수익을 창출하는 '마켓플레이스 기능' 등으로 확장 가능

플랫폼 확장성

- 현재 웹 기반에서 나아가, Electron이나 Tauri를 이용해 데스크톱 앱으로 패키징하거나, React Native / Flutter 등을 통해 모바일 앱으로 확장할 수 있는 기반을 고려하여 설계

8. 기대 효과 및 결론

8.1 프로젝트 수행 후 기대되는 효과

본 프로젝트가 성공적으로 완료되면, 기술적, 사회적으로 다음과 같은 긍정적 효과를 기대할 수 있습니다.

1. 창작의 장벽 파괴

- 코딩 능력이 없는 사람도 상상력만으로 게임을 만들 수 있게 되어, IT 및 게임 분야에 대한 대중의 관심과 참여를 유도할 수 있습니다.

2. 신속한 프로토타이핑

- 아이디어 구상 단계에서 수 분 만에 실제 동작하는 프로토타입을 제작하고 테스트함으로써, 'Fail Fast, Learn Faster' 문화를 게임 개발에 적용할 수 있습니다.

3. AI 융합의 새로운 모델 제시

- 단순한 정보 제공을 넘어, 특정 도메인에 전문화된 AI가 실제 동작하는 결과물을 창출해 내는 성공적인 AI 융합 서비스 모델을 제시합니다.

8.2 차별화 요소 요약

대화형 인터페이스

- 명령이 아닌, 대화를 통해 점진적으로 결과물을 완성

도메인 특화 (RAG)

- 범용 AI가 아닌, 프로젝트를 '학습'한 전문가 AI

End-to-End 자동화

- 아이디어 구상부터 즉시 플레이까지 전 과정이 자동화

센서 연동 특화

- 다른 생성 AI가 다루지 않는, 모바일 센서 연동이라는 독창적인 영역에 집중

8.3 결론 및 향후 발전 방향

본 'Sensor Chatbot' 프로젝트는 생성형 AI와 실시간 웹 기술을 결합하여, 게임 개발의 패러다임을 바꾸는 혁신적인 시도입니다. RAG 아키텍처를 통해 AI의 창의성과 프로젝트의 일관성을 모두 확보하고, 대화형 인터페이스를 통해 사용자 경험을 극대화함으로써 기술적 완성도와 사용성을 모두 만족시키는 것을 목표로 합니다.

향후 발전 방향

- 멀티모달(Multi-modal) 입력 지원
 - 텍스트뿐만 아니라, 사용자가 그린 스케치나 이미지, 음성 명령을 이해하여 게임을 생성하는 기능
- AI 기반 협업 기능
 - 여러 사용자가 함께 AI와 대화하며 하나의 게임을 같이 만들어가는 협업 모드 개발
- 생성된 게임의 자가 발전
 - 사용자의 플레이 데이터를 AI가 학습하여, 게임의 난이도를 자동 조절하거나 새로운 스테이지를 스스로 생성하는 기능

9. 부록 (Appendix)

9.1 참고 문헌 및 자료

- LangChain Official Documentation
<https://js.langchain.com/docs/>
- OpenAI API Reference
<https://platform.openai.com/docs/api-reference>
- Anthropic API Documentation
<https://docs.anthropic.com/claude/reference/getting-started-with-the-api>
- Supabase Documentation
<https://supabase.com/docs>
- Socket.IO Documentation
<https://socket.io/docs/v4/>