

AiFrenz

ISSUE PAPER

01

바이브 코딩과 교육 현장의 변화 실천 사례와 미래 전망

작성자 | 박권 호원중학교 · 정용석 서울신답초등학교
김혜진 증평초등학교 · 한준구 삼양초등학교

발 행 | AiFrenz 학회 산하 교육AI연구회



CC BY 4.0

발행정보

항목	내용
발행처	AiFrenz 학회 산하 교육AI연구회
발행일	2026년 1월
통권	제 1호
ISSN	(발급 예정)
주제	바이브 코딩과 교육 현장의 변화
라이선스	CC BY 4.0
편집	Designed with Claude Code

기고자

Chapter	저자	소속
Chapter 1	박권	호원중학교 기술가정 교사
Chapter 2	정용석	서울신답초등학교 교사
Chapter 3	김혜진	증평초등학교 교사
Chapter 4	한준구	삼양초등학교 교사

초록

본 이슈페이퍼는 ‘바이브 코딩(Vibe Coding)’이 교육 현장에 가져오는 변화와 그 실천 사례를 탐구한다. 바이브 코딩은 AI와 자연어로 대화하며 프로그램을 개발하는 새로운 패러다임으로, 복잡한 프로그래밍 문법 없이도 교육적 의도만으로 도구를 설계하고 구현할 수 있게 한다.

본고에서는 네 가지 현장 사례를 중심으로 분석한다. 첫째, 안티그래비티 IDE와 Gemini 3.0을 활용한 게임 기반 학습(GBL) 도구 개발 사례를 통해 교사의 ‘메이커’ 역할 전환 가능성을 제시한다. 둘째, 학교평가 대시보드와 3D 우주 시뮬레이션 개발을 통해 행정 효율화와 수업 확장의 실제를 보여준다. 셋째, 개념기반 탐구학습에서 교육용 웹앱 활용의

성공과 실패 요인을 분석하여 ‘기술을 위한 기술’이 아닌 ‘교육 목표 달성을 위한 기술’의 중요성을 강조한다. 넷째, Claude Code를 중심으로 터미널 기반 AI 코딩 에이전트의 교육적 활용 방안을 안내한다.

연구 결과, AI 시대 교사는 교육 도구의 단순 소비자에서 능동적 생산자로 전환할 수 있으며, 이를 위해서는 기술적 역량보다 ‘무엇을 만들 것인가’에 대한 교육적 상상력이 더 중요함을 확인하였다.

주제어: 바이브 코딩, 생성형 AI, 교사 역할 변화, 게임 기반 학습, 개념기반 탐구학습, 교육용 웹앱, Claude Code

Abstract

This issue paper explores the transformative impact of ‘Vibe Coding’ on educational practice and presents practical implementation cases. Vibe coding represents a new paradigm where educators develop programs through natural language conversation with AI, enabling tool design and implementation based solely on pedagogical intent without complex programming syntax.

This study analyzes four field cases. First, we demonstrate the potential for teachers to transition into ‘maker’ roles through game-based learning (GBL) tool development using Anti-Gravity IDE and Gemini 3.0. Second, we illustrate administrative efficiency and instructional expansion through school evaluation dashboards and 3D space simulations. Third, we analyze success and failure factors in utilizing educational web applications for concept-based inquiry learning, emphasizing ‘technology for educational goals’ rather than ‘technology for technology’s sake.’ Fourth, we provide guidance on educational applications of terminal-based AI coding agents, focusing on Claude Code.

The findings confirm that in the AI era, teachers can transform from passive consumers to active producers of educational tools, and that pedagogical imagination regarding ‘what to create’ is more crucial than technical competency.

Keywords: Vibe Coding, Generative AI, Teacher Role Transformation, Game-Based Learning, Concept-Based Inquiry Learning, Educational Web Apps, Claude Code

머리말

바이브 코딩과 교육 현장의 변화

지금 우리는 “코딩 시대”에서 “바이브 코딩 시대”로 넘어가는 매우 짧은 다리를 건너는 중이다. 불과 1~2년 전만 해도 생성형 AI는 주어진 질문에 답하고, 글을 요약하고, 코드를 제안하는 수준의 “대화형 도우미”로 이해됐다. 하지만 오늘날의 AI는 단순히 대답만 하지 않는다. 스스로 문제를 정의하고, 필요한 정보를 외부에서 수집하고, 도구를 호출하고, 다른 AI나 사람과 협업하고, 최종적으로는 실제 업무 절차를 끝까지 수행한다.

중요한 변화는 “지능의 위치”보다 “행동의 단위”에 있다. 과거에는 인간이 업무 프로세스를 분할해 시스템에 명령했다. 앞으로는 고수준의 지시만 내리면, AI 에이전트가 여러 하위 에이전트와 도구를 불러 업무를 분해·배정·실행한다. 바이브 코딩은 이러한 변화의 교육 현장 적용 사례이다.

본 이슈페이퍼는 그러한 전환의 교육적 실체를 다룬다. 우리는 “바이브 코딩이 트렌드”라는 것보다는 현장에서 바로 적용 가능한 구조와 한계를 구체적으로 짚는다. 특히 다음 세 가지 질문에 집중한다.

핵심 질문

- ① 바이브 코딩을 활용한 게임 기반 학습(GBL) 도구 개발은 어떻게 가능한가?
- ② 교사가 ‘소비자’에서 ‘생산자(Prosumer)’로 전환하는 실제 사례는 무엇인가?
- ③ 개념기반 탐구학습에서 교육용 웹앱 활용의 성공과 실패 요인은 무엇인가?

이 세 가지 질문에 대한 답을 통해, 독자 여러분은 AI 시대 교사의 역할 변화와 기술 활용의 균형에 대한 통찰을 얻으실 수 있을 것이다.



본문

CHAPTER 1. 안티그래비티 IDE와 바이브 코딩을 활용한 GBL 수업 설계

박권

호원중학교 기술가정 교사

1. 들어가는 말: GBL 수업의 현실적 제약

게임 기반 학습(Game-Based Learning, GBL)은 학생들의 몰입도와 학습 동기를 높이는 효과적인 교수법으로 알려져 있다. 그러나 교육 현장에서 GBL을 적용하는 과정에는 여러 제약이 존재한다. 서버 불안정, 상용 플랫폼의 비용 부담, 그리고 수업 맥락에 맞게 기능을 수정할 수 없는 한계 등이 대표적이다.

2025년 현재, 구글의 차세대 개발 환경인 '안티그래비티(Anti-Gravity)' IDE는 이러한 문제에 대한 하나의 해결책을 제시한다. 자연어 만으로 개발 환경을 설정하고, AI 도구와의 통합을 통해 비전공자도 교육용 앱을 제작할 수 있는 가능성을 열었다.

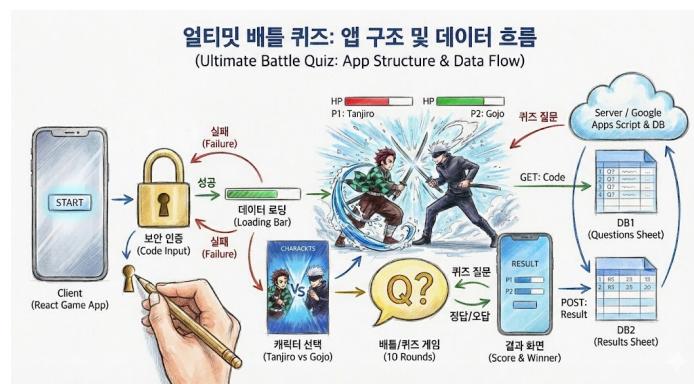


그림 1-1: 안티그래비티 IDE 실행 화면

본고에서는 Gemini 3.0, NanoBanana Pro, 그리고 바이브 코딩(Vibe Coding)을 활용하여 교사가 직접 GBL 도구를 개발한 사례를 소개하고자 한다.

2. GBL의 이상과 현실

2.1 학생 기대 수준과 교육용 도구의 괴리

프로젝트 기반 학습(PBL)에 게임화(Gamification) 요소를 결합하면 학습 효과를 높일 수 있다. 그러나 현재 학생들은 고해상도 그래픽과 즉각적인 피드백에 익숙한 세대다. 기존 교육용 도구로 제작한 게임은 UI의 완성도, 기능의 다양성, 디자인 측면에서 학생들의 기대 수준에 미치지 못하는 경우가 많다.

이러한 기대 격차는 학생들의 학습 몰입도를 저하시키는 요인이 된다.

2.2 교사의 도구 생산자로서의 역할

Kahoot, Quizizz 등 기존 플랫폼은 사용이 편리하지만, 정해진 형식의 제약이 있다. 수업의 특수한 맥락에 맞는 기능을 구현하기 위해서는 교사가 직접 도구를 설계하고 개발하는 ‘메이커(Maker)’ 역할을 수행할 필요가 있다. 생성형 AI의 발전은 이러한 역할 전환을 가능하게 하는 기술적 기반을 제공한다.

3. AI 기반 시각 자료 생성: NanoBanana Pro 활용

교육용 게임 개발에서 시각 자료 확보는 상당한 시간과 비용이 소요되는 작업이다. 저작권 문제가 없는 이미지를 찾거나, 직접 제작해야 하는 부담이 있었다. 이미지 생성 모델 NanoBanana Pro는 이 과정을 단순화한다. 텍스트 프롬프트만으로 게임 개발에 적합한 에셋(Asset)을 생성할 수 있다.

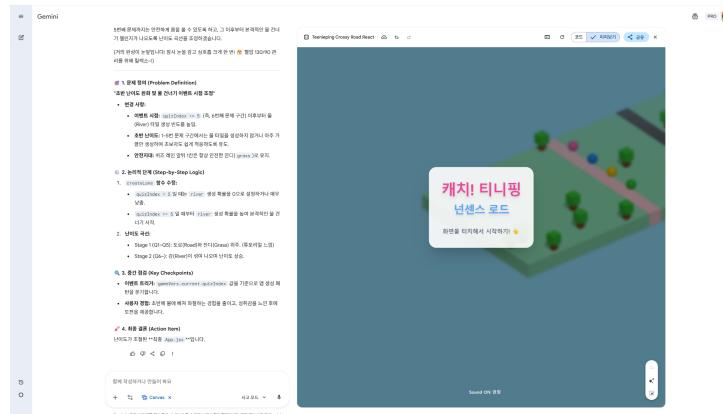


그림 1-2: NanoBanana Pro를 활용한 이미지 생성 예시

Gemini 3.0은 생성된 시각 자료에 인터랙션을 부여하는 역할을 담당한다. 자연어 프롬프트를 통해 애니메이션 효과, UI 컴포넌트, 사용자 인터랙션 등 프론트엔드(UI/UX) 요소를 구현할 수 있다.

4. 바이브 코딩: Gemini Canvas를 활용한 개발 방식

'바이브 코딩(Vibe Coding)'은 AI와 자연어로 대화하며 기능을 구현해 나가는 개발 방식을 의미한다. 전통적인 코딩이 문법(Syntax)에 대한 이해를 전제로 한다면, 바이브 코딩은 개발자의 의도(Intent)를 AI가 해석하여 코드로 변환하는 과정에 초점을 맞춘다.

Gemini Canvas는 코드 에디터와 AI 채팅 인터페이스가 통합된 환경이다. 교사는 원하는 기능을 자연어로 설명하고, AI가 생성한 코드를 검토하며 수정 사항을 다시 자연어로 전달하는 반복적 과정을 통해 개발을 진행한다.

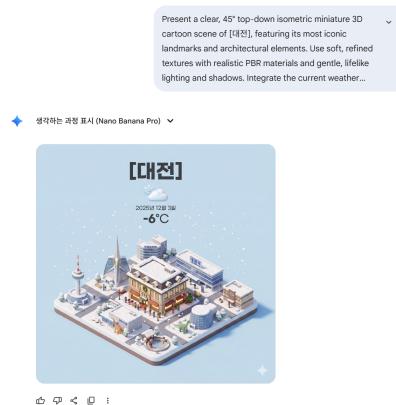


그림 1-3: Gemini Canvas 작업 화면

5. 개발 사례: AI 퀴즈 게임 'Gemini-Quiz'

안티그래비티 IDE 환경에서 'Gemini-Quiz'라는 퀴즈 게임을 개발한 사례를 소개한다. 개발 목표는 다음과 같았다: 수업 주제에 따라 AI가 실시간으로 퀴즈를 생성하고, 학생들의 점수를 실시간 랭킹으로 표시하는 것이다.

데이터 저장의 필요성을 자연어로 전달하자, Gemini는 Firebase 연동 코드를 생성했다. 이를 통해 학생들의 퀴즈 응답과 점수가 클라우드 데이터베이스에 실시간으로 저장되는 구조를 구현할 수 있었다.

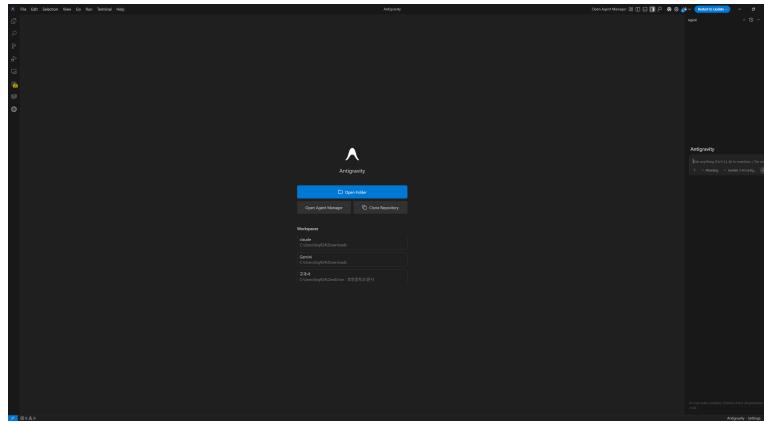


그림 1-4: Full-Stack GBL 아키텍처 다이어그램

6. 맷음말: AIM 프레임워크

본 사례를 통해 교사가 AI 도구를 활용하여 직접 교육용 앱을 개발할 수 있는 가능성을 확인하였다. 이를 “AIM(Approach-Implement-Master)” 프레임워크로 정리할 수 있다.

Approach (접근): 안티그래비티, 바이브 코딩 등을 통해 개발 진입 장벽을 낮춘다.

Implement (구현): Gemini 3.0, NanoBanana Pro 등을 활용하여 교육적 아이디어를 실제 도구로 구현한다.

Master (숙달): 반복적인 개발 경험을 통해 기술과 데이터를 주도적으로 활용하는 역량을 갖춘다.

AI 도구의 발전은 교사가 교육 콘텐츠의 소비자에서 생산자로 전환할 수 있는 환경을 조성하고 있다. 이러한 변화는 수업의 맥락에 최적화된 교육 도구 개발을 가능하게 하며, 궁극적으로 학습 효과의 향상에 기여할 수 있을 것으로 기대된다.

CHAPTER 2. 교육용 앱 개발의 실제와 '바이브 코딩'의 가능성

정용석

서울신답초등학교 교사

1. 들어가는 말: 에듀테크의 주도권 이동

지금까지 에듀테크(EdTech) 시장은 철저히 공급자 중심이었다. 기업이 개발한 앱을 학교가 구매하여 사용하는 구조 속에서, 현장의 미세한 맥락(Context)과 요구사항은 종종 기술적 표준화라는 명목하에 생략되곤 했다. 하지만 생성형 AI와 이를 활용한 코딩 에이전트(Cursor, Claude 등)의 등장은 이러한 공식을 근본적으로 바꾸고 있다.

이제 교사는 복잡한 프로그래밍 문법(Syntax)을 학습하지 않고도, 교육적 의도와 논리(Logic)만으로 자신에게 필요한 도구를 직접 개발할 수 있게 되었다. 이른바 '바이브 코딩(Vibe Coding)'의 시대가 열린 것이다.

2. [사례 1: 행정 효율화] 데이터 기반 학교평가 대시보드

매년 말, 일선 학교는 해당년도 학교평가와 차년도 교육계획 수립을 위해 학생, 학부모, 교원을 대상으로 방대한 설문조사를 실시한다. 그러나 수백 명의 응답 데이터를 엑셀(Excel)로 취합하고, 담당자가 이를 수기로 분석하여 보고서를 작성하는 아날로그적 방식은 교사의 행정 업무를 가중시키는 주원인이었다.

이를 해결하기 위해 개발한 '학교평가 대시보드'는 전문 개발자가 아닌 교사가 AI 코딩 도구를 활용해 엔터프라이즈급 솔루션을 구축한 사례다. "학년별 응답 차이를 도넛 차트로 비교해 달라"는 자연어 프롬프트를 통해, AI는 즉시 실행 가능한 코드를 생성해냈다.



그림 2-1: 학교평가 지표 분석 - 교원, 직원, 학생, 학부모 만족도 조사



그림 2-2: 학생 생활 실태 분석 - 학생 설문조사 결과

3. [사례 2: 수업의 확장] 3D 우주 시뮬레이션 '밤하늘 관찰'

초등 4학년 과학 '지구와 달' 단원은 학생들의 흥미도가 높지만, 수업 운영에는 한계가 명확했다. 도심의 빛 공해와 날씨, 그리고 안전 문제로 인해 실제 밤하늘을 관측하는 활동이 어렵기 때문이다.

이러한 문제를 해결하고자 웹 기반의 '3D 우주 시뮬레이션'을 개발했다. 이 과정에서 AI는 고난도 3D 그래픽 라이브러리(Three.js)를 다루는 전문 엔지니어 역할을 수행했다. 운석의 질량과 속도에 따라 분화구의 깊이가 달라지는 충돌 시뮬레이션, 풍화 작용에 의한 지형 변화 등 복잡한 물리 법칙을 코드로 구현했다.



그림 2-3: 3D 우주 시뮬레이션 화면 - 운석 충돌 및 지형 변화

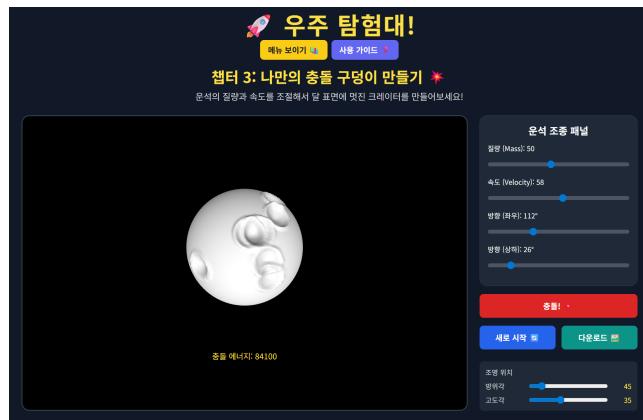


그림 2-4: 3D 우주 시뮬레이션 화면 - 달의 위상 변화 관찰

4. [사례 3: 미래 교육] 3D AI 민주시민 시뮬레이션

사회과의 '지역 사회 문제 해결' 수업은 현장 탐문과 조사가 핵심이지만, 최근 사회적 안전 이슈로 인해 학생들을 학교 밖으로 내보내는 활동이 위축되고 있다. 이에 대한 대안으로 '3D AI 민주시민 시뮬레이션'을 개발했다.

이 사례의 핵심은 생성형 AI(LLM)가 단순한 챗봇을 넘어, 학생과 상호작용하는 '에이전트(Agent)'로 기능한다는 점이다. 가상 마을의 NPC들은 각기 다른 이해관계를 가진다. 환경미화원 NPC는 쓰레기 투기 문제에 분노하고, 상인 NPC는 규제로 인한 매출 하락을 걱정한다.



그림 2-5: 3D 민주시민 시뮬레이션 화면 - NPC와의 대화



그림 2-6: 3D 민주시민 시뮬레이션 화면 - 해결책에 대한 피드백

5. 맷음말

위의 세 가지 사례는 교사가 AI를 도구로 삼아 ‘소비자’에서 ‘생산자’로 전환될 때, 교육 현장이 얼마나 역동적으로 변할 수 있는지를 보여준다. 행정 업무는 효율화되고, 수업의 시공간은 확장되며, 학생들은 AI와 상호작용하며 더 깊이 있는 배움을 경험한다.

‘바이브 코딩’은 기술적 장벽을 허물었다. 이제 중요한 것은 ‘무엇을 코딩할 것인가’에 대한 교육적 상상력이다. 교사가 수업의 기획자이자 도구의 설계자(Architect)로서 AI를 주도적으로 활용할 때, 공교육의 디지털 전환은 비로소 완성될 것이다.

CHAPTER 3. AI 시대, 바이브코딩 및 교육용 웹앱 활용을 통한 개념기반 탐구학습

김혜진

증평초등학교 교사

1. AI 시대, 왜 개념기반 교육인가?

1.1 인간의 능동적 추론과 AI의 학습 방식

마코프 블랭킷(Markov Blanket) 이론에 따르면, 인간의 뇌는 단순히 정보를 수동적으로 받아들이는 것이 아니라 능동적 추론(Active Inference)을 통해 세상을 이해한다. 시각 정보가 유입되면 과거의 경험을 선별하고, 원인을 추론하며, 예측 오류가 발생하면 추론을 업데이트하는 과정을 반복한다.

놀랍게도 현대 AI도 이와 유사한 방식으로 학습한다. 하지만 여기서 중요한 질문이 제기된다: “학생들에게 지식을 많이 넣어주는 것만으로는 불충분하다면, 무엇을 가르쳐야 하는가?” 답은 명확하다. 지식을 연결할 수 있는 ‘개념’이다.

1.2 2022 개정 교육과정: 2차원에서 3차원으로

2022 개정 교육과정은 기존의 2차원 교육(지식+기능)에서 3차원 교육(지식+기능+이해)로의 전환을 핵심으로 한다. 여기서 ‘개념’은 2차원과 3차원을 연결하는 브릿지 역할을 한다. 개념은 시간을 초월하고 보편적이며 추상적이라는 특성을 갖는다.

2. 개념기반 탐구학습의 실제: 증평초 수업 사례

“사회의 변화와 문화의 다양성” 주제로 10차시를 체계적으로 설계했다. 1차시에서는 형태, 기능, 인과관계, 변화, 연결성, 관점, 책임의 7가지 렌즈로 탐구질문을 생성하고 마인드맵으로 개념을 연결한다. 2-5차시는 저출산·고령화, 정보화, 세계화를 집중 탐구하며 피쉬본 다이어그램, 신호등 기법, 관점의 원 등 사고 전략을 활용한다.

3. 교육용 웹앱의 필요성: 왜 디지털 도구인가?

개념기반 탐구학습은 사고 시각화, 실시간 피드백, 협업과 공유, 과정 기록, 결과물 전시를 요구하며, 전통적 교실 환경에서 이를 동시에 수행하기 어렵다. 교육용 웹앱은 이러한 요구를

효과적으로 지원한다.

4. 성공 사례: 교육용 웹앱이 만든 변화

4.1 바이브코딩 통계청 API 연동 그래프 도구

2-3차시 저출산·고령화 탐구에서 교사가 바이브코딩으로 제작한 실시간 데이터 시각화 웹앱을 활용했다. 통계청 공공 API를 연동하여 실제 출산율 데이터를 불러와 학생들이 연도별 출산율 변화를 선 그래프로 직접 그리고, 65세 이상 노인 인구 비율 그래프와 비교하며 인과관계를 파악했다.

한 학생은 “저출산이 숫자가 아니라 우리 동네 학교가 없어질 수도 있다는 의미”라며 추상적 통계를 구체적 현실로 이해했다. 또 다른 학생은 “그래프가 계속 내려가면 할머니 할아버지를 돌볼 사람이 줄어들 것 같다”며 인과관계 개념을 자발적으로 표현했다.



그림 3-1: 바이브 코딩으로 제작한 합계출산율 변화탐색기



그림 3-2: 학생들의 성찰일지 - 출산율

4.2 바이브코딩 키오스크 시뮬레이션 앱

3차시 정보화 사회 탐구에서 교사가 바이브코딩으로 제작한 키오스크 시뮬레이션 앱을 활용하여 학생들이 '관점'이라는 추상 개념을 몸으로 체험하게 했다. 실제 키오스크 인터페이스를 재현하되, "노인 체험 모드"에서는 글씨 크기 축소, 짧은 제한시간, 버튼 반응 지연 등으로 의도적 어려움을 설정했다.

학생 A의 소감: "처음에는 키오스크가 진짜 편한데 왜 안 쓰냐고 생각했어요. 근데 노인 모드로 해보니 글씨도 안 보이고 시간도 부족해서 화가 났어요. 할머니가 왜 키오스크를 어려워하시는지 알 것 같아요."

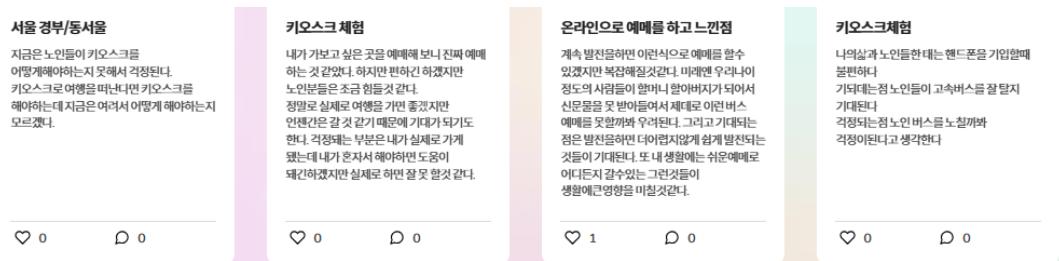


그림 3-3: 바이브코딩으로 제작한 키오스크 시뮬레이션 앱



그림 3-4: 학생들의 성찰일지 - 키오스크

5. 실패 사례: 바이브코딩 교육용 웹앱의 함정

6-7차시 "문화의 다양성" 조사를 위해 교사가 바이브코딩으로 "세계 문화 룰렛" 웹앱을 제작했다. 학생들은 처음 10분간 "우와! 신기해요!"라며 열광했지만, 20분 후에는 "선생님, 다 봤는데 이제 뭐 해요?"라며 흥미를 잃었다.

핵심 교훈

- 기술의 "신기함"과 게이미피케이션의 외재적 동기는 일시적 흥미만 유발할 뿐, 개념 이해로 연결되지 않으면 학습 효과는 미미하다.
- 화려한 앱보다 "학습 목표 → 사고 전략 → 개념 연결"이 명확한 단순한 도구가 더 효과적이다.

6. 결론: 균형 잡힌 기술 활용의 지혜

AI 시대에 개념기반 교육은 필수적이며, 개념은 지식의 연결고리이자 전이 가능한 사고의 기반이다. 교육용 웹앱은 사고 시각화, 협업, 창작 활동에 효과적이나, 보안·안정성·복잡한 로직은 전문가의 영역으로 남아야 한다.

"기술을 위한 기술"이 아니라 "교육 목표를 달성하기 위한 기술"이어야 한다.

성공의 기준은 "얼마나 멋진 앱을 만들었는가"가 아니라 "학생의 내면화와 액션을 얼마나 촉진했는가"이다. AI 시대에도 변하지 않을 교육의 핵심은 개념을 연결하고, 이해를 깊게 하며, 학생들이 스스로 생각하고 행동하도록 돋는 것이다.

CHAPTER 4. Claude Code를 활용한 바이브 코딩 실전 가이드

한준구

교육AI연구회 회장

1. 들어가는 말: 왜 Claude Code인가?

앞선 챕터에서 살펴본 것처럼, 안티그래비티 IDE와 Gemini Canvas는 웹 브라우저 기반의 직관적인 개발 환경을 제공한다. 그러나 교사가 바이브 코딩을 본격적으로 활용하고자 할 때, 또 다른 강력한 도구가 있다. 바로 Anthropic이 개발한 Claude Code이다.

Claude Code는 터미널(명령줄) 기반의 AI 코딩 에이전트로, 다음과 같은 차별화된 강점을 갖는다:

첫째, 로컬 파일 시스템과의 완전한 통합이다. 웹 기반 도구와 달리 교사의 컴퓨터에 있는 모든 파일을 직접 읽고, 수정하고, 생성할 수 있다. 수업 자료가 담긴 폴더 전체를 맥락으로 활용하여 일관성 있는 교육 콘텐츠를 제작할 수 있다.

둘째, 자동화와 배치 처리가 가능하다. 30개 학급의 개별화된 학습지를 한 번에 생성하거나, 학기 전체 수업 자료를 체계적으로 구조화하는 작업을 자동으로 수행할 수 있다.

셋째, 확장 가능한 Skills 시스템을 제공한다. 반복적으로 사용하는 워크플로우를 ‘스킬(Skill)’로 저장하여 재사용할 수 있으며, 다른 교사들과 공유할 수도 있다.

본 챕터에서는 교사가 Claude Code를 효과적으로 활용하기 위해 알아야 할 핵심 기능들을 실전 중심으로 안내한다.

2. CLI 환경의 이해

(1) CLI란 무엇인가?

CLI(Command Line Interface, 명령줄 인터페이스)는 텍스트 명령어를 입력하여 컴퓨터와 상호작용하는 방식이다. 우리가 일상적으로 사용하는 마우스 클릭 방식의 GUI(그래픽 사용자 인터페이스)와 달리, CLI는 키보드로 명령어를 직접 입력한다.

CLI가 교사에게 낯설 수 있지만, Claude Code와 같은 AI 도구를 활용하면 복잡한 명령어를 직접 외울 필요 없이 자연어로 원하는 작업을 요청할 수 있다. AI가 적절한 명령어를 생성하고 실행해주기 때문이다.

(2) 터미널 실행 방법

Windows:

- 시작 메뉴에서 “PowerShell” 또는 “명령 프롬프트” 검색
- 또는 Win + R → cmd 입력 → Enter

macOS:

- Spotlight 검색(Cmd + Space) → “터미널” 입력
- 또는 응용 프로그램 → 유틸리티 → 터미널

Linux:

- Ctrl + Alt + T 단축키
- 또는 응용 프로그램 메뉴에서 “Terminal” 검색

3. Claude Code 설치하기

(1) 사전 준비: Node.js 설치

Claude Code는 Node.js 환경에서 실행된다. 먼저 Node.js가 설치되어 있는지 확인한다:

```
node --version
```

버전 번호가 표시되면 이미 설치된 것이다. 만약 “command not found” 오류가 나타나면 Node.js를 설치해야 한다.

Node.js 설치 방법:

- 공식 웹사이트(<https://nodejs.org>)에서 LTS 버전 다운로드
- 설치 파일 실행 후 안내에 따라 설치 완료



그림 4-1: Node.js 공식 웹사이트

(2) Claude Code 설치

터미널에서 다음 명령어를 입력하여 Claude Code를 설치한다:

```
npm install -g @anthropic-ai/clause-code
```

설치가 완료되면 다음 명령어로 정상 설치를 확인한다:

```
clause --version
```

(3) Claude Code 시작하기

설치가 완료되면 원하는 작업 폴더로 이동한 후 Claude Code를 실행한다:

```
cd ~/Documents/수업자료
clause
```

처음 실행 시 Anthropic 계정 로그인이 필요하다. 브라우저가 자동으로 열리면 로그인을 완료한다.



그림 4-2: Claude Code 로그인 화면

로그인 후 터미널에 › 프롬프트가 나타나면 준비 완료이다. 이제 자연어로 원하는 작업을 요청할 수 있다:

- › 이 폴더에 있는 파일 목록을 보여줘
- › 새로운 수업 계획서 템플릿을 만들어줘
- › 학습지.docx 파일의 내용을 요약해줘

4. 권한 설정과 워크플로우 최적화

(1) Claude Code의 권한 시스템 이해

Claude Code는 사용자의 파일을 보호하기 위해 3단계 권한 시스템을 운영한다:

작업 유형	예시	승인 필요
읽기 전용	파일 읽기, 검색	불필요
명령 실행	터미널 명령어	매번 승인
파일 수정	편집, 생성, 삭제	세션 내 1회

기본 설정에서는 파일 수정이나 명령 실행 시마다 사용자의 승인을 요청한다. 이는 안전한 사용을 보장하지만, 대량의 자료를 생성하거나 자동화 워크플로우를 구축할 때는 작업 흐름을 방해할 수 있다.

(2) 자동화를 위한 권한 우회: --dangerously-skip-permissions

반복적인 승인 과정을 생략하고 싶을 때, 다음 옵션을 사용할 수 있다:

```
claude --dangerously-skip-permissions
```

이 옵션을 활성화하면 모든 권한 확인이 자동으로 승인되어, AI가 끊김 없이 작업을 수행한다.

교육 현장 활용 사례

- 수업 자료 대량 생성: 30개 학급별 맞춤 학습지를 중단 없이 처리
- 프로젝트 초기 설정: 새 학기 수업 폴더 구조와 템플릿 파일을 한 번에 생성
- CI/CD 파이프라인: GitHub Actions와 연동하여 교육 콘텐츠 자동 배포

주의사항

1. 신뢰할 수 있는 환경에서만 사용: 직접 작성하거나 검증된 프롬프트만 실행
2. 중요 파일 백업: 작업 전 원본 자료 백업 권장
3. 격리된 폴더에서 작업: 시스템 파일이나 다른 프로젝트에 영향을 주지 않도록

전체 권한을 우회하지 않고도 작업 효율을 높일 수 있는 더 안전한 대안들이 있다:

```
# 파일 수정만 자동 승인 (명령 실행은 확인)
```

```
claude --permission-mode acceptEdits
```

```
# 특정 도구만 허용
```

```
claude --allowedTools "Read,Write>Edit"
```

```
# 특정 명령어 패턴만 허용
```

```
claude --allowedTools "Bash(npm run build:*)"
```

5. 컨텍스트 관리: 효율적인 작업 흐름

교사의 업무 특성상, 수업 준비 중 다른 업무로 전환해야 하는 경우가 빈번하다. Claude Code는 이러한 상황에 대응할 수 있는 강력한 컨텍스트 관리 명령어들을 제공한다.

(1) /resume: 중단된 작업 이어가기

```
/resume
```

이 명령어를 입력하면 최근 작업 세션 목록이 표시되고, 원하는 세션을 선택하여 이전 작업을 그대로 이어갈 수 있다. 모든 대화 내용, 파일 수정 이력, 작업 맥락이 복원된다.

활용 시나리오

- 수업 자료 제작 중 회의 참석 후 복귀
- 전날 작업하던 프로젝트 다음날 계속
- 여러 프로젝트를 번갈아가며 진행

(2) /compact: 토큰 절약과 컨텍스트 압축

```
/compact
```

장시간 작업 시 대화 내용이 누적되면 AI의 컨텍스트 한도에 가까워진다. /compact 명령어는 지금까지의 대화를 핵심 내용만 남기고 압축하여, 작업을 계속할 수 있게 해준다.

상태	메시지 수	토큰 사용량
압축 전	50개	~15,000
압축 후	2개 (요약)	~2,500

특정 영역에 집중하여 압축할 수도 있다:

```
/compact "학습지 템플릿 작업에 집중"
```

(3) /clear: 새로운 주제로 전환

```
/clear
```

완전히 새로운 작업을 시작할 때 사용한다. 이전 대화 이력이 삭제되어 깨끗한 상태에서 시작할 수 있다. 단, 이미 수정된 파일들은 그대로 유지된다.

(4) /export: 작업 기록 저장 및 공유

```
/export chapter4_작업기록.md
```

현재까지의 대화 내용을 마크다운 파일로 저장한다. 문서화, 동료 교사에게 작업 과정 전달, AI 활용 수업의 예시 자료로 활용할 수 있다.

(5) /context: 현재 상태 확인

```
/context
```

현재 토큰 사용량과 컨텍스트 상태를 시각적으로 보여준다. 70% 이상이면 /compact를 고려하고, 90% 이상이면 /compact가 필수이며, 완전히 새 주제라면 /clear 후 시작하는 것이 좋다.

6. Skills 시스템: 맞춤형 AI 워크플로우 구축

(1) Skills의 개념

Skills는 Claude에게 특정 분야의 전문 지식을 가르치는 마크다운 파일이다. 일반적인 프롬프트와 달리, Skills는 다음과 같은 특징을 갖는다:

1. 자동 발견: 사용자의 요청이 Skill의 설명과 일치하면 자동으로 활성화
2. 구조화된 지식: 복잡한 워크플로우를 단계별로 정의
3. 재사용 가능: 한 번 만들면 반복 사용
4. 공유 가능: 다른 교사나 팀과 공유

(2) Skill의 저장 위치

```
~/.claude/skills/      # 개인 Skills (모든 프로젝트에서 사용)  
.claude/skills/       # 프로젝트 Skills (해당 폴더에서만 사용)
```

(3) Skill 파일 구조

Skill은 SKILL.md라는 파일명으로 저장되며, 다음 구조를 따른다:

name: lesson-plan-generator
description: 학교 현장에서 사용하는 교수·학습 과정안 양식으로 수업 지도안을 생성합니다.

--

교수·학습 과정안 생성기

사용 시점

- 새 단원 수업 준비 시
- 공개수업/연구수업 지도안 작성 시
- 교육실습생 수업 시연 준비 시

입력 정보

1. 교과목, 학교급/학년, 단원명
2. 배움주제, 차시 정보

생성 단계

1. 성취기준 및 핵심역량 분석
2. 배움목표 설정
3. 교실수업 개선전략 (활동형태, 적용모형)
4. 배움활동 설계 (도입-전개-정리)
5. 평가 계획 및 판서 계획 수립

배움활동 구분기호

☆배움안내 ◎배움상황설명 ◇일반자료 ※유의점

위 스킬을 실행하면 다음과 같이 학교 현장에서 사용하는 교수·학습 과정안이 자동으로 생성된다:

그림 4-3: lesson-plan-generator 스킬 실행 결과

(4) /plugin: 플러그인 설치 및 관리

Skills를 패키지 형태로 배포하는 것이 플러그인이다. 다음 명령어로 관리할 수 있다:

```
/plugin install plugin-name  # 플러그인 설치  
/plugin list                 # 설치된 플러그인 목록  
/plugin remove plugin-name   # 플러그인 제거
```

7. 교육용 커스텀 스키ル 레포지토리 소개

바이브 코딩 환경에서 즉시 활용할 수 있는 교육용 스킬 레포지토리들을 소개한다. 이 도구들은 Claude Code의 Skills 시스템과 연동하여 교사의 콘텐츠 제작을 자동화한다.

① ppt-yt-gen: PPT를 영상으로 자동 변환

GitHub: <https://github.com/jkf87/ppt-yt-gen>

이 도구는 프레젠테이션 슬라이드를 나레이션과 자막이 포함된 영상으로 자동 변환한다. 학생들을 위한 플립러닝(Flipped Learning) 영상을 제작하고 싶지만, 직접 녹화하고 편집할 시간이 부족한 교사에게 유용하다.

핵심 기능

- 슬라이드별 이미지 자동 렌더링 (1920×1080)
 - AI 음성 클로닝을 통한 나레이션 생성
 - 자동 자막 생성 및 동기화
 - FFmpeg 기반 고품질 MP4 출력

② antigravity-workflow-ppt: 학술 스타일 PPT 자동 생성

GitHub: <https://github.com/jkf87/antigravity-workflow-ppt>

Google Gemini API를 활용하여 학술적 스타일의 이미지가 포함된 프레젠테이션을 자동 생성한다. 연구 발표, 학술 세미나, 전문적인 교육 자료 제작 시 유용하다.

핵심 기능

- 주제 기반 슬라이드 구조 자동 설계
- 학술 논문 스타일의 다이어그램/도표 생성
- IEEE/CVPR 스타일 시각 자료 지원
- JSON 기반 슬라이드 정의

③ doc-research-generator: 연구 문서 자동 생성

GitHub: <https://github.com/jkf87/doc-research-generator>

키워드나 주제를 입력하면 웹 리서치, 시각화, 문서 작성은 자동으로 수행하여 완성된 보고서를 생성한다.

핵심 기능

- 자동화된 웹 리서치 및 자료 수집
- 데이터 분석 및 시각화 (차트, 다이어그램)
- 다국어 보고서 지원 (한글/영문)
- Word/PPT 형식 출력

세 가지 도구를 연계한 통합 워크플로우를 Claude Code Skill로 구성하면, "기후변화에 대한 수업 영상 만들어줘"라는 한 문장으로 리서치부터 영상 제작까지 자동 수행된다.

8. 맷음말: 교사를 위한 AI 도구 활용 로드맵

Claude Code는 터미널이라는 다소 낯선 인터페이스를 사용하지만, 그만큼 강력한 자동화와 확장성을 제공한다. 본 챕터에서 다룬 내용을 단계별로 정리하면 다음과 같다:

1단계: 기본 사용 - Claude Code 설치 및 기본 대화, 간단한 파일 생성 및 수정 요청

2단계: 효율화 - /resume, /compact 등 컨텍스트 명령어 활용, --permission-mode 옵션으로 워크플로우 최적화

3단계: 자동화 - Skills 제작 및 적용, 커스텀 레포지토리 활용, 반복 작업의 자동화 구축

4단계: 공유와 협업 - 제작한 Skills를 동료 교사와 공유, 학교/교육청 단위의 Skills 라이브러리 구축

앞선 챕터들에서 살펴본 안티그래비티 IDE, Gemini Canvas와 함께 Claude Code를 활용한다면, 교사는 더욱 풍부한 바이브 코딩 도구 생태계를 갖추게 된다. 중요한 것은 도구 자체가 아니라, 교육적 목표를 달성하기 위해 적절한 도구를 선택하는 안목이다.

바이브 코딩의 시대, 교사는 이제 상상하는 것을 직접 만들 수 있다.



맺음말

본 이슈페이퍼에서 다룬 네 편의 기고문은 공통된 메시지를 전달한다: AI 시대, 교사는 더 이상 기술의 단순 소비자가 아니라 교육 도구의 능동적 생산자가 될 수 있다는 것이다.

바이브 코딩의 등장은 프로그래밍이라는 기술적 장벽을 낮추었다. 이제 교사는 복잡한 문법(Syntax)을 암기하지 않아도, 교육적 의도와 논리(Logic)만으로 자신의 수업에 최적화된 도구를 설계하고 구현할 수 있다. 안티그래비티 IDE, Gemini Canvas, Claude Code 등의 도구들은 이러한 가능성을 현실로 만들어주고 있다.

그러나 기술 만능주의에 대한 경계의 목소리도 함께 전달한다. "기술을 위한 기술"이 아니라 "교육 목표를 달성하기 위한 기술"이어야 한다. 화려한 앱보다 '학습 목표 → 사고 전략 → 개념 연결'이 명확한 단순한 도구가 더 효과적일 수 있다.

교육의 본질은 변하지 않는다. 학생들이 개념을 연결하고, 깊이 이해하며, 스스로 생각하고 행동하도록 돋는 것이 교사의 역할이다. AI는 이 역할을 대체하는 것이 아니라, 교사가 더 효과적으로 이 역할을 수행할 수 있도록 지원하는 도구이다.

본 이슈페이퍼가 현장 선생님들께 작은 영감이 되기를 바라며, 앞으로도 교육AI연구회는 교육과 기술의 균형 잡힌 발전을 위해 노력하겠다. 기고해 주신 박권, 정용석, 김혜진, 한준구 선생님께 깊은 감사의 말씀을 드린다.

AiFRenz 교육AI연구회 일동

IV

참고문헌

학술 문헌

- Erickson, H. L., Lanning, L. A., & French, R. (2017). 생각하는 교실을 위한 개념기반 교육과정 및 수업 (온정덕 & 윤지영, Trans.). 학지사. (Original work published 2017)
- 위정현 and 김태연. (2010). G러닝(온라인게임 기반 학습) 콘텐츠의 학습효과 분석: 초등학생의 학업성취도에 미치는 영향. PREVIEW : 디지털영상학술지, 7(1), 67-82.
- 백영균 and 정용석. (2004). 게임기반학습에서 학습자의 게임능력 및 학습능력이 논리적사고력에 미치는 효과. 교육정보미디어연구, 10(4), 119-140.

정책 문서

- 교육부. (2022). *2022 개정 교육과정 총론*. 교육부 고시 제2022-33호.
- 교육부. (2024). *디지털 기반 교육혁신 방안*. 교육부.
- International Baccalaureate Organization. (2023). *PYP: From principles into practice*. IBO.

웹 자료 및 기술 문서

- Google. (2025). *Anti-Gravity IDE: Agentic Development Platform*. <https://labs.google/>
- Google DeepMind. (2025). *Gemini 3.0 Technical Report*. <https://deepmind.google/>



“학생들이 무엇을 배워야 하는가”에
답하는 것은 여전히 교사의 몫이다.

AI 도구들은 그 답을 실현하는 과정을
더욱 효율적이고 창의적으로 만들어줄 뿐이다.

AiFrenz

교육AI연구회