구글 안티그라비티(Google Antigravity) 기반 자율형 멀티모달 3D 시나리오 생성 및 인터랙션 웹 파일럿(Web Pilot) 상세 설계 보고서

1. 서론: 공간 컴퓨팅과 에이전트 기반 개발(Agentic Development)의 융합

1.1 배경 및 목적

현대 소프트웨어 공학은 단순히 코드를 작성하는 단계를 넘어, 복잡한 인공지능 모델들을 조율(Orchestration)하고 이들을 통해 새로운 실시간 콘텐츠를 생성해내는 '생성형 엔지니어링' 시대로 진입하고 있습니다. 귀하께서 제안하신 "이미지와 프롬프트를 기반으로 시나리오를 자동 생성하고, 이에 부합하는 3D 공간과 인터랙션을 구축하는 AI 모델"은 생성형 AI 기술의 정점에 있는 과제입니다. 이는 시각적 정보(Image)를 서사적 구조(Scenario)로 변환하고, 다시 이를 공간적 실체(3D Space)와 행위적 논리(Interaction)로 재구성하는 고도의 멀티모달 파이프라인을 요구합니다.

본 보고서는 구글의 최신 에이전트 중심 통합 개발 환경(IDE)인 **안티그라비티(Antigravity)**를 활용하여, 이러한 복잡한 시스템을 구축하기 위한 상세 설계안을 제시합니다. 우리는 이 시스템을 **"웹 파일럿(Web Pilot): 자율형 공간 서사 엔진(Autonomous Spatial Narrative Engine)"**이라 명명하며, 이는 사용자의 추상적인 의도를 구체적인 3D 웹 경험으로 항해(Pilot)해주는 지능형 에이전트를 의미합니다.[1]

1.2 기술적 난제와 해결 방안

이 프로젝트의 핵심 난제는 서로 다른 모달리티(Modality) 간의 의미론적 손실 없는 변환입니다. 정지된 2D 이미지에서 4차원적인 시간과 인과율을 가진 시나리오를 추출하고, 이를 다시 물리적 제약을 가진 3D 공간으로 매핑하는 과정은 기존의 절차적 생성(Procedural Generation) 방식으로는 불가능합니다.

이를 해결하기 위해 본 설계는 구글의 제미나이 3 프로(Gemini 3 Pro) 모델의 '심층 사고(Deep Think)' 기능과 안티그라비티의 미션 컨트롤(Mission Control) 아키텍처를 결합합니다.[1] 제미나이 3는 이미지 내의 객체뿐만 아니라 그 객체가 내포한 '행동 유도성(Affordance)'을 추론하여 인터랙션 로직을 생성하며, 안티그라비티의 자율 에이전트들은 이러한 로직이 실제 웹 환경(Three.js/R3F)에서 오류 없이 작동하도록 코드를 구현, 테스트, 검증하는 역할을 수행합니다.[1]

---

2. 개발 환경 아키텍처**:** 구글 안티그라비티**(Antigravity)** 생태계 구축

안티그라비티는 단순한 코드 에디터가 아닌, 자율 에이전트들을 관리하고 운용하는 플랫폼입니다. 웹 파일럿 시스템의 복잡성을 고려할 때, 인간 개발자는 '코더(Coder)'가 아닌 '아키텍트(Architect)'로서 에이전트 군단을 지휘해야 합니다.

2.1 미션 컨트롤**(Mission Control)** 기반 워크플로우 설계

안티그라비티의 인터페이스는 크게 **에디터(Editor)**와 **에이전트 매니저(Agent Manager)**로 이원화되어 있습니다.[1] 본 프로젝트에서는 에이전트 매니저를 통해 다수의 특화된 에이전트를 병렬로 운용하는 전략을 채택합니다.

표 **1.** 웹 파일럿 개발을 위한 안티그라비티 에이전트 역할 분담

에이전트 **ID**	역할 **(Persona)**	주요 임무 및 책임	활용 모델	생성 아티팩트**(Artifacts)**
Architect-01	시스템 설계자	전체 프로젝트 구조 설계, Next.js 라우팅 정의, API 스키마(Zod) 확정	Gemini 3 Pro (High Thinking)	태스크 리스트(Task List), 구현 계획서(Implementation Plan) [1]
Visual-Core	3D 그래픽스 엔지니어	Three.js/R3F 컴포넌트 구현, 쉐이더(Shader) 코딩, GLB 로더 최적화	Gemini 3 Pro / Claude 3.5 Sonnet	코드 변경사항(Diffs), UI 스크린샷 [5]
Logic-Weaver	게임 로직 설계자	XState 기반 인터랙션 상태 머신 생성, 시나리오 분기 로직 구현	Gemini 3 Pro (Deep Think)	상태 머신 다이어그램, 유닛 테스트 코드 [6]
QA-Pilot	테스트 및 검증	브라우저 내 자율 주행	Gemini 3 Flash (Low Latency)	브라우저 녹화 영상(Browser

		테스트, WebGL 성능(FPS) 모니터링, 오류 디버깅		Recordings), 테스트 로그 [1]

이러한 분업화는 안티그라비티의 **"비동기 에이전트 실행(Asynchronous Agent Execution)"** 기능을 통해 가능합니다. 예를 들어, Visual-Core 에이전트가 3D 모델 로딩 최적화를 수행하는 동안, Logic-Weaver 에이전트는 시나리오 생성 알고리즘을 작성할 수 있습니다.[2]

2.2 신뢰성 확보를 위한 아티팩트**(Artifacts)** 중심 개발

생성형 AI를 이용한 개발의 가장 큰 위험은 '환각(Hallucination)'에 의한 잘못된 코드 생성입니다. 안티그라비티는 이를 방지하기 위해 아티팩트라는 검증 가능한 산출물을 제공합니다.[2]

- 구현 계획**(Implementation Plan):** 코드를 작성하기 전, 에이전트는 "어떻게 시나리오 엔진을 구현할 것인가"에 대한 상세 계획을 마크다운 문서로 생성합니다. 개발자는 이 계획을 검토하고, "시나리오의 다양성을 위해 Temperature 파라미터를 0.7로 조정하라"와 같은 피드백을 댓글(Comment) 형태로 남길 수 있습니다.[7]
- 브라우저 레코딩**(Browser Recordings):** 3D 웹 환경은 정적 테스트(Unit Test)만으로는 검증이 어렵습니다. QA-Pilot 에이전트는 실제 헤드리스 브라우저(Headless Browser)를 띄워 생성된 3D 공간을 돌아다니며, 벽을 뚫고 지나가지는 않는지(충돌 처리 확인), 클릭 이벤트가 정상 작동하는지 확인하고 그 과정을 영상으로 기록하여 개발자에게 보고합니다.[1]

2.3 워크스페이스 규칙**(.antigravity/rules.md)** 설정

성공적인 에이전트 협업을 위해 프로젝트 루트에 명시적인 규칙을 설정해야 합니다.[1] 본 프로젝트를 위한 핵심 규칙은 다음과 같습니다:

1. **3D** 좌표계 표준화**:** "모든 Three.js 객체 배치는 Y-up 좌표계를 따르며, 바닥면은 y=0으로 고정한다. 객체 간의 겹침(Overlapping)을 방지하기 위해 Bounding Box 계산을 필수적으로 수행하라."
2. 상태 관리 원칙**:** "모든 인터랙션 로직은 XState 머신으로 정의되어야 하며, 불확실한 if-else 분기 대신 명시적인 상태 전이(State Transition)를 사용하라."
3. 리소스 관리**:** "생성된 3D 자산(GLB/Texture)은 반드시 비동기적으로 로드되어야 하며, Suspense와 Fallback 컴포넌트를 사용하여 사용자 경험을 저해하지 않도록 한다."

---

3. 인지 엔진**(Cognitive Engine):** 멀티모달 시나리오 생성 파이프라인

사용자가 업로드한 이미지를 단순한 배경이 아닌, 살아있는 이야기의 무대로 변환하기 위해서는 고도화된 인지 엔진이 필요합니다. 이 엔진은 **제미나이 3 프로(Gemini 3 Pro)**를 핵심 두뇌로 사용합니다.

3.1 시각적 추론과 내러티브 추출 (Visual-to-Narrative)

이미지 한 장에서 시나리오를 생성하는 과정은 단순한 캡셔닝(Captioning)을 넘어섭니다. 이는 이미지 속에 숨겨진 인과관계와 잠재된 서사를 발굴하는 과정입니다.

프로세스 흐름:

1. 이미지 분석 **(Semiotics Analysis):** 제미나이 3의 멀티모달 기능을 활용하여 이미지 내의 객체(Object), 조명(Lighting), 분위기(Mood), 그리고 특이점(Anomaly)을 분석합니다. 예를 들어, 평범한 서재 이미지에서 "바닥에 떨어진 책"이라는 특이점을 발견하면, 이를 "누군가 급하게 떠난 흔적"으로 해석합니다.[4]
2. 프롬프트 융합 **(Prompt Fusion):** 사용자의 입력 프롬프트(예: "미스터리 추리물")와 이미지 분석 결과를 결합합니다.
3. 심층 사고 **(Deep Think):** 제미나이 3의 thinking_level="high" 설정을 통해 시나리오의 개연성을 확보합니다.[3] 모델은 내부적으로 여러 가지 시나리오 분기를 시뮬레이션해보고, 가장 논리적이고 흥미로운 서사를 선택합니다.

출력 데이터 구조 **(Scenario JSON Schema):**

JSON

```
{
  "scenario_title": "잊혀진 서재의 비밀",
  "genre": "Mystery",
  "atmosphere": "Tense, Gloomy",
  "narrative_arc": {
    "introduction": "당신은 오래된 저택의 서재에 들어섰습니다. 공기는 차갑고, 바닥에는 책 한 권이 떨어져 있습니다.",
    "climax": "떨어진 책 사이에서 숨겨진 금고의 비밀번호가 적힌 쪽지를 발견합니다.",
    "resolution": "금고를 열어 가문의 비밀을 밝혀냅니다."
  },
  "key_elements": [
    {"object": "fallen_book", "role": "clue", "description": "가죽 장정의 낡은 책"},
    {"object": "desk_lamp", "role": "interaction", "description": "깜빡거리는 램프"}
  ]
}
```

3.2 의미론적 장면 그래프 (Semantic Scene Graph) 생성

시나리오는 공간을 필요로 합니다. 텍스트로 생성된 시나리오를 3D 공간으로 변환하기 위해, 중간 단계인 **장면 그래프(Scene Graph)**를 생성합니다. 이는 공간 내 객체들의 관계와 위치를 정의하는 구조화된 데이터입니다.[10]

장면 그래프의 역할:

- 공간적 관계 정의: "책상은 방의 중앙에 있다", "램프는 책상 위에 있다"와 같은 상대적 위치 정보를 정의합니다. LLM은 절대 좌표(x, y, z)를 다루는 데 약점이 있으므로, 이러한 의미론적 관계(Semantic Relationship)를 생성하고, 이를 후술할 **레이아웃 리졸버(Layout Resolver)**가 실제 좌표로 변환하도록 설계합니다.[12]
- 생성 **API** 트리거: 각 노드(객체)는 3D 생성 API(Blockade Labs, Tripo3D)에 보낼 프롬프트를 포함합니다.

---

4. 생성형 3D 파이프라인 (Generative 3D Pipeline) 구축

안티그라비티 에이전트는 정의된 장면 그래프를 바탕으로 실제 3D 자산들을 생성하고 배치하는 파이프라인을 자동으로 코딩하고 연결합니다.

4.1 환경 생성: Blockade Labs Skybox API 통합

공간의 전체적인 분위기를 결정하는 배경(Environment)은 **Blockade Labs**의 **Skybox AI**를 활용합니다.[13]

구현 전략:

- 프롬프트 엔지니어링 자동화: 제미나이 3가 시나리오의 분위기(Atmosphere)를 바탕으로 Skybox API에 최적화된 프롬프트를 생성합니다. (예: "Interior of a Victorian library, stormy night outside window, volumetric lighting, 8k resolution, realistic style").
- 깊이 맵(Depth Map) 활용: 단순한 360도 이미지는 평면적입니다. Skybox API에서 제공하는 **Depth Map**을 함께 요청하여, Three.js 환경에서 입체적인 포그(Fog) 효과나 파티클 효과가 배경 깊이에 맞춰 렌더링되도록 합니다.[13]
- 리믹스(Remix) 기능 구현: 사용자가 시나리오 진행 중 "불을 켠다"는 행동을 하면, 동일한 구조(Structure)를 유지하되 조명만 밝게 변경된 스카이박스를 생성하기 위해 remix 기능을 활용하는 로직을 구현합니다.[13]

안티그라비티 에이전트 작업 지시:

> "Architect-01, BlockadeService.ts 모듈을 작성하라. 이 모듈은 Skybox API의 generate 및 export 엔드포인트를 처리하고, 생성 상태를 폴링(Polling)하며, 완료 시 텍스처 URL을 반환해야 한다. 오류 발생 시 지수 백오프(Exponential Backoff) 재시도 로직을 포함하라."

4.2 객체 생성: Tripo3D 및 Meshy AI 통합

시나리오의 핵심 아이템(단서, 도구 등)은 **Tripo3D** 또는 **Meshy** API를 통해 3D 메시(Mesh)로 생성됩니다.[15]

지연 시간(Latency) 극복 전략:
3D 모델 생성은 수 초에서 수 분이 소요됩니다. 웹 파일럿의 실시간성을 보장하기 위해 점진적 로딩(Progressive Loading) 전략을 사용합니다.

1. 플레이스홀더(Placeholder): 객체가 생성되는 동안, 해당 위치에 "홀로그램 쉐이더"가 적용된 기본 도형(Cube, Sphere)을 먼저 배치하여 사용자가 상호작용할 수 있게 합니다.
2. 백그라운드 스트리밍: GLB 파일 생성이 완료되면, 안티그라비티가 작성한 AssetLoader가 자동으로 플레이스홀더를 실제 모델로 교체(Hot-swap)하고, 등장 애니메이션을 재생합니다.[15]

4.3 자동 레이아웃 해결 엔진 (Auto-Layout Resolver)

LLM이 생성한 "책상은 방 중앙에, 램프는 책상 위에"라는 추상적 지시를 3D 좌표로 변환하는 엔진입니다.

알고리즘 설계:

- 제약 조건 충족(Constraint Satisfaction): 각 객체의 Bounding Box 크기를 고려하여 겹치지 않게 배치합니다.
- 레이캐스팅(Raycasting) 기반 접지: 객체가 공중에 떠 있지 않도록, 위에서 아래로 가상의 레이(Ray)를 쏘아 바닥면(또는 테이블 윗면)의 Y좌표를 찾아 배치합니다.[17]
- 이 엔진은 TypeScript로 작성되며, 안티그라비티의 **Logic-Weaver** 에이전트가 핵심 알고리즘을 구현하고, **QA-Pilot**이 시뮬레이션을 통해 충돌 여부를 검증합니다.

---

5. 자율 인터랙션 및 행동 합성 (Interaction & Behavior Synthesis)

단순히 공간을 보여주는 것을 넘어, 사용자의 행동에 반응하는 **"인터랙션(Interaction)"**이 웹 파일럿의 핵심입니다.

5.1 행동 유도성(Affordance) 기반 인터랙션 생성

제미나이 3는 이미지 속 객체가 어떤 행동을 유발할 수 있는지(Affordance)를 분석합니다.

- 의자: 앉기(Sit), 밀기(Push), 조사하기(Inspect).
- 문: 열기(Open), 잠그기(Lock), 노크하기(Knock).
- 책: 읽기(Read), 줍기(Pick up), 던지기(Throw).

이러한 행동 가능성은 사용자의 프롬프트 시나리오에 맞춰 필터링됩니다. 예를 들어 "탈출 게임" 시나리오라면, 문에 대한 인터랙션은 "열기"가 아니라 "잠금 장치 확인하기"가

우선순위를 갖게 됩니다.

5.2 XState 기반 상태 머신(State Machine) 자동 생성

복잡한 인터랙션 로직을 if-else문으로 관리하면 코드가 난잡해지고 버그가 발생하기 쉽습니다. 따라서 **XState** 라이브러리를 사용하여 결정론적(Deterministic) 상태 머신을 구축합니다.[6]

자동 생성 프로세스:

1. 상태 정의: 제미나이 3가 시나리오 흐름에 따라 객체의 상태를 정의합니다 (예: 금고 - Closed, Locked, Open).
2. 전이 조건(Transition Logic): 상태 간의 이동 조건을 정의합니다 (예: Locked -> Closed (조건: 비밀번호 입력 성공)).
3. 코드 생성: 안티그라비티 에이전트가 이 정의를 바탕으로 XState 머신 코드를 생성합니다.

코드 예시 (에이전트 생성 결과물):

TypeScript

```
const safeMachine = createMachine({
  id: 'safe',
  initial: 'locked',
  context: { retries: 0 },
  states: {
    locked: {
      on: { UNLOCK: { target: 'closed', cond: 'checkPassword' } }
    },
    closed: {
      on: { OPEN: 'open' }
    },
    open: {
      type: 'final'
    }
  }
});
```

5.3 런타임 내러티브 관리 (Dungeon Master Agent)

웹 파일럿 실행 중, 사용자의 행동에 대한 서술적 피드백은 실시간 AI가 담당합니다. 사용자가 "책을 줍는다"는 행동을 하면, 클라이언트 측의 **Gemini 3 Flash** 모델이 즉각적으로 "당신은 먼지 쌓인 책을 집어 들었습니다. 책 표지에는 이상한 문양이 새겨져 있습니다."라는 텍스트를

생성하여 UI에 표시합니다.[3]

---

6. 웹 파일럿 클라이언트 런타임 아키텍처

최종 사용자가 경험하게 될 웹 애플리케이션의 기술적 구조입니다.

6.1 프론트엔드 스택: Next.js + React Three Fiber (R3F)

안티그라비티는 **Next.js** 환경에서 **React Three Fiber(R3F)**를 사용하는 프로젝트를 스캐폴딩(Scaffolding)합니다. R3F는 Three.js의 명령형(Imperative) 코드를 리액트의 선언형(Declarative) 컴포넌트로 변환해주어, 생성형 AI가 코드를 작성하고 관리하기에 훨씬 유리합니다.[3]

- **Scene Composition:** 생성된 각 객체는 <Suspense>로 감싸진 R3F 컴포넌트로 변환되어 비동기적으로 씬에 추가됩니다.
- **Post-processing:** 제미나이 3가 시나리오 분위기에 맞는 쉐이더 코드를 작성하여 <EffectComposer>에 주입합니다 (예: 공포 분위기 -> 노이즈 효과, 비네팅 추가).[19]

6.2 성능 최적화 (Performance Optimization)

웹 환경에서의 3D 경험은 최적화가 필수적입니다. 안티그라비티의 **QA-Pilot** 에이전트는 다음 최적화 작업을 자동으로 수행합니다:

- 텍스처 압축: 모든 텍스처를 WebP 또는 KTX2 포맷으로 자동 변환.
- 형상 압축: GLB 파일에 Draco 압축 적용.
- **LOD (Level of Detail):** 카메라 거리에 따라 모델의 폴리곤 수를 조절하는 로직 구현.

---

7. 안티그라비티 활용 구현 로드맵 (Step-by-Step)

개발자가 안티그라비티를 사용하여 이 시스템을 실제로 구축하는 구체적인 단계입니다.

단계 1: 프로젝트 초기화 및 에이전트 설정

- 명령: 안티그라비티 에디터에서 Cmd+L을 눌러 에이전트 패널을 열고 입력: "새로운 Next.js + R3F 프로젝트를 생성해. 프로젝트 이름은 'WebPilot-Engine'이야. .antigravity/rules.md에 3D 개발 규칙을 추가해줘."
- 에이전트 행동: 필요한 패키지(three, @react-three/drei, xstate, firebase)를 설치하고 폴더 구조를 잡습니다.[1]

단계 2: 생성형 API 연동 모듈 구현

- 명령: "Blockade Labs와 Tripo3D API를 연동하는 서비스 레이어를 작성해. 각 API의 응답 스키마를 Zod로 정의하고, 에러 처리를 포함해."

- 에이전트 행동**:** src/services/ 폴더에 API 클라이언트 코드를 작성하고, 테스트 파일을 생성합니다.

단계 **3:** 시나리오-공간 변환 로직 구현

- 명령**:** "Gemini 3 Pro를 사용하여 이미지를 분석하고 Scene Graph JSON을 반환하는 서버리스 함수를 만들어. 그리고 이 JSON을 받아 R3F 씬으로 렌더링하는 SceneGenerator 컴포넌트를 구현해."
- 에이전트 행동**:** Next.js API 라우트(pages/api/generate-scene.ts)를 작성하고, 프롬프트 엔지니어링을 수행합니다.

단계 **4:** 인터랙션 시스템 및 **UI** 구축

- 명령**:** "XState를 사용하여 아이템 습득 및 문 열기 로직을 관리하는 상태 머신을 만들고, 이를 UI 오버레이와 연동해."
- 에이전트 행동**:** 상태 머신 훅(useMachine)을 구현하고, 3D 캔버스 위에 HTML UI 레이어를 덮어씌웁니다.

단계 **5:** 자율 테스트 및 배포

- 명령**:** "QA 에이전트를 실행해. 생성된 씬에서 캐릭터가 이동 가능한지, 클릭 이벤트가 발생하는지 확인하고 브라우저 녹화 영상을 보여줘."
- 에이전트 행동**:** Playwright 등을 이용한 E2E 테스트를 수행하고 결과를 아티팩트로 제출합니다.[5]

---

8. 결론 및 향후 전망

본 보고서는 구글 안티그라비티라는 혁신적인 도구를 통해, 단 한 장의 이미지와 텍스트만으로 살아있는 3D 웹 경험을 창조하는 **"웹 파일럿"**의 설계도를 제시했습니다. 이는 기존에 수십 명의 개발자와 아티스트가 필요했던 작업을 단 한 명의 '아키텍트'와 AI 에이전트 팀이 수행할 수 있게 만드는 패러다임의 전환입니다.

향후 **3D** 가우시안 스플래팅**(Gaussian Splatting)** 기술이 웹 표준에 통합되면, 메시 생성 단계를 건너뛰고 사진에서 바로 3D 씬을 실시간으로 합성하는 것이 가능해질 것입니다.[21] 안티그라비티의 유연한 에이전트 구조는 이러한 기술적 진보를 즉각적으로 파이프라인에 통합할 수 있는 확장성을 제공합니다. 이제 개발자는 코드를 타이핑하는 것이 아니라, 세계를 설계하고 지휘하는 역할을 맡게 될 것입니다.

데이터 테이블 및 비교 분석

표 **2.** 생성형 모델 선정 매트릭스

구성 요소	추천 모델/API	선정 근거	안티그라비티 에이전트 활용
인지 코어 **(Cognitive Core)**	**Gemini 3 Pro**	최고 수준의 추론 능력(1501 Elo), 멀티모달 입력 처리, 심층 사고(Deep Think)를 통한 논리적 시나리오 구성.[8]	시스템 오케스트레이터; 장면 그래프 및 상태 머신 설계.
실시간 로직 **(Runtime Logic)**	**Gemini 3 Flash**	낮은 지연 시간, 높은 처리량, NPC 대화 및 실시간 스크립트 처리에 적합.[3]	런타임 실행; 웹 파일럿의 채팅 인터페이스 구동.
환경 생성 **(Environment)**	**Blockade Labs Skybox**	360도 일관성 우수; '리믹스' 기능을 통한 스타일 일관성 유지 및 조명 변경 가능.[13]	API 키 관리 및 리믹스 로직 구현 에이전트.
3D 자산 **(3D Assets)**	**Tripo3D / Meshy**	빠른 추론 속도; 이미지 기반 3D 생성(Image-to-3D) 지원으로 씬 구성 속도 향상.[16]	폴링(Polling) 로직 및 GLB 최적화 파이프라인 구현.
코드 생성 **(Code Gen)**	**Gemini 3 (via Antigravity)**	쉐이더 코딩을 위한 'Vibe Coding' 능력; Three.js/R3F 문법에 대한 깊은 이해.[19]	애플리케이션 소스 코드 및 유닛 테스트 작성.

표 **3.** 개발 패러다임 비교**:** 전통적 방식 **vs.** 안티그라비티

특징	전통적 **IDE (VS Code)**	에이전트형 **IDE (Antigravity)**	**"웹 파일럿"** 프로젝트에 미치는 영향
작업 단위	파일 및 코드 라인	태스크 및 아티팩트	"로더 코드 작성"이 아닌 "로딩 시스템

	(Files & LOC)	(Tasks & Artifacts)	구축"이라는 상위 목표에 집중 가능.
AI 상호작용	자동완성 / 채팅 창 (Autocomplete / Chat)	자율 에이전트 매니저 (Autonomous Agent Manager)	개발자가 로직을 구상하는 동안 에이전트가 백그라운드에서 레이아웃 버그를 수정.
테스트 방식	수동 / 스크립트 작성	브라우저-인-더-루프 에이전트 (Browser-in-the-Loop)	에이전트가 시각적으로 3D 씬을 돌아다니며 렌더링 및 성능을 검증.
컨텍스트	파일 기반	워크스페이스 및 프로젝트 기반	에이전트가 3D 엔진의 전체 아키텍처를 이해하고 리팩토링 제안.
워크플로우	동기식 (사용자 입력 대기)	비동기식 (위임 및 병렬 실행)	50개의 자산 생성 프롬프트 작성을 에이전트에 위임하고 병렬 처리 가능.

참고 자료

1. Getting Started with Google Antigravity, 1월 6, 2026에 액세스, https://codelabs.developers.google.com/getting-started-google-antigravity
2. Build with Google Antigravity, our new agentic development platform, 1월 6, 2026에 액세스, https://developers.googleblog.com/build-with-google-antigravity-our-new-agentic-development-platform/
3. Gemini 3 Developer Guide | Gemini API - Google AI for Developers, 1월 6, 2026에 액세스, https://ai.google.dev/gemini-api/docs/gemini-3
4. Gemini 3 - Google DeepMind, 1월 6, 2026에 액세스, https://deepmind.google/models/gemini/
5. Tutorial : Getting Started with Google Antigravity | by Romin Irani - Medium, 1월 6, 2026에 액세스, https://medium.com/google-cloud/tutorial-getting-started-with-google-antigravi

ty-b5cc74c103c2
6. XState - Stately.ai, 1월 6, 2026에 액세스, https://stately.ai/docs/xstate
7. Introducing Google Antigravity, a New Era in AI-Assisted Software Development, 1월 6, 2026에 액세스, https://antigravity.google/blog/introducing-google-antigravity
8. Gemini 3 Explained: Google's Most Advanced Agentic AI Model With Deep Reasoning, 1월 6, 2026에 액세스, https://www.sculptsoft.com/gemini-3-explained-advanced-agentic-ai-model/
9. Building AI Agents with Google Gemini 3 and Open Source Frameworks, 1월 6, 2026에 액세스, https://developers.googleblog.com/building-ai-agents-with-google-gemini-3-and-open-source-frameworks/
10. Can Large Language Models Understand and Generate Scene Graphs? A Benchmark and Empirical Study - ACL Anthology, 1월 6, 2026에 액세스, https://aclanthology.org/2025.acl-long.1036.pdf
11. Planner3D: LLM-enhanced graph prior meets 3D indoor scene explicit regularization - arXiv, 1월 6, 2026에 액세스, https://arxiv.org/html/2403.12848v2
12. SceneWiz3D: Towards Text-guided 3D Scene Composition - Qihang ZHANG, 1월 6, 2026에 액세스, https://zqh0253.github.io/SceneWiz3D/
13. Skyboxes | Blockade Labs API Documentation, 1월 6, 2026에 액세스, https://api-documentation.blockadelabs.com/api/skybox.html
14. Roadmap | Skybox AI - Blockade Labs, 1월 6, 2026에 액세스, https://www.blockadelabs.com/roadmap
15. Meshy AI - The #1 AI 3D Model Generator, 1월 6, 2026에 액세스, https://www.meshy.ai/
16. Tripo API - Tripo AI, 1월 6, 2026에 액세스, https://platform.tripo3d.ai/
17. Does threejs allow opening a box as a way of interaction or is it only interactions with outside, meaning push the box, move it around... - Reddit, 1월 6, 2026에 액세스, https://www.reddit.com/r/threejs/comments/18z2lb2/does_threejs_allow_opening_a_box_as_a_way_of/
18. XState in React: Look Ma, no useState or useEffect! - frontend undefined, 1월 6, 2026에 액세스, https://www.frontendundefined.com/posts/monthly/xstate-in-react/
19. Hands-on with Gemini 3.0: Building 3D Universes with a Single Prompt and “Keep Developing”, 1월 6, 2026에 액세스, https://sewox.medium.com/hands-on-with-gemini-3-0-building-3d-universes-with-a-single-prompt-and-keep-developing-1a606a4fb2c0
20. Google Antigravity Tutorial: Build a Finance Risk Dashboard - DataCamp, 1월 6, 2026에 액세스, https://www.datacamp.com/tutorial/google-antigravity-tutorial
21. Revolutionizing Web Experiences with AI-Generated 3D Scenes - DEV Community, 1월 6, 2026에 액세스, https://dev.to/yitzi/revolutionizing-web-experiences-with-ai-generated-3d-scenes-3oj