

**SceneForge AI: MCP, A Multimodal Agent for Generative 3D Art and Scripting in Blender** project plan



Ver 1.3

QKV

Table of Content

1. 프로젝트 제목 5

2. 필요성 및 배경 5

3. 목표 5

4. 팀별 역할 5

5. 요구사항 분석 및 정의 5

5.1 자료 조사 5

5.2 경쟁사 벤치마킹 5

5.3 요구사항 정의 5

6. 서비스 설계(예시) 6

6.1 서비스 개념 및 구조도 6

6.2 UX 설계 6

6.3 데이터 설계 6

6.4 프로그램 모듈 구조, 모델 및 알고리즘 설계 7

7. 테스트 계획 7

8. 구현 7

9. 결과 8

10. 감사의 글 8

11. 레퍼런스 8

# 프로젝트 제목

SceneForge AI: MCP, A Multimodal Agent for Generative 3D Art and Scripting in Blender

# 필요성 및 배경

Blender는 매우 강력한 3D 모델링 도구이지만, 비전문가가 사용하기에는 진입 장벽이 높은 편입니다. 최근에는 인공지능과 자연어 처리 기술의 발달로, 사용자의 텍스트 명령을 코드로 자동 변환하는 시스템이 주목받고 있습니다. 이에 따라 누구나 복잡한 조작 없이 텍스트로 3D 작업을 수행할 수 있는 인터페이스에 대한 수요가 커지고 있습니다.

본 프로젝트는 이러한 흐름에 발맞추어, 사용자가 웹에서 입력한 자연어 명령을 Python 코드로 변환한 후, Blender 내부에서 이를 자동 실행할 수 있도록 하는 **AI 기반 3D 명령 시스템**을 개발하고자 합니다. 이를 통해 Blender의 접근성을 높이고, 향후 음성 명령, AR/VR 등 다양한 인터페이스로 확장할 수 있는 기반을 마련하고자 합니다.

# 목표

1. 자연어 기반 3D 제어 서비스를 통해 비전문가 대상의 3D 콘텐츠 제작 시장을 공략
   * 복잡한 UI 조작 없이 텍스트만으로 Blender를 제어할 수 있도록 하여, 일반 사용자나 비개발자도 3D 작업을 쉽게 수행할 수 있는 사용 경험을 제공합니다.
2. 기존 Blender 활용 환경의 높은 학습 비용과 자동화 한계를 개선하여 3D 제작 생산성을 획기적으로 향상시킵니다.
   * 사용자의 의도를 코드로 직접 전환하는 프로세스를 자동화함으로써 반복 작업의 부담을 줄이고, 창작 과정에 집중할 수 있는 환경을 조성합니다.

# 팀별 역할

|  |  |  |
| --- | --- | --- |
| 이름 | 역할 | 산출물 |
| 윤동의 | 조장, 프로젝트 기획, Agent 개발 | RAG 기반 Agent 모듈 |
| 이세훈 | Text to 3D.obj 모델 fine-tuning | Text to 3D.obj 모듈, 기획서 |
| 박준형 | Text to Image 모델 fine-tuning | Text to Image 모듈, 2page논문 |
| 이주엽 | MCP 클라이언트 구현 | MCP 클라이언트 모듈 |

# 요구사항 분석 및 정의

## 자료 조사

본 프로젝트는 자연어 기반으로 Blender 명령을 실행하는 MCP(Model Command Proxy) 서버를 구축하는 것을 목표로 합니다. 이를 위해 현재 GitHub 상에서 공개된 [Claude와 연동되는 Blender MCP 서버 프로젝트](https://github.com/ahujasid/blender-mcp) 사례를 조사하였습니다.

해당 사례는 MCP 서버를 통해 자연어 명령을 Claude로 전달하고, Claude가 반환한 Python 스크립트를 Blender에서 직접 실행하는 구조를 채택하고 있습니다. 이 아키텍처는 다음과 같은 시사점을 줍니다:

MCP 서버는 LLM 모델과 Blender 간의 중재자 역할을 하며, 요청 → 코드 생성 → 실행의 흐름을 책임진다. LLM과의 연동을 통해 고정된 명령어가 아닌 열린 자연어 인터페이스를 구현할 수 있다. Blender 내부에서 직접 LLM이 생성한 코드를 실행할 수 있도록 안전한 애드온 구조가 필요하다.

이와 같은 구조는 사용자의 접근성과 직관성을 크게 높이며, 향후 음성 명령 기반의 확장성도 확보할 수 있다는 점에서 본 프로젝트의 방향성과 일치합니다.

## 경쟁사 벤치마킹

6.1에서 조사한 **Claude + Blender MCP 연동 프로젝트**는 본 프로젝트와 유사한 구조를 가지고 있으나 기능적 확장 항목을 목표로 벤치마킹 및 차별화를 시도합니다.

| **항목** | **설명** |
| --- | --- |
| **기술문서 임제종 처리** | **Blender Python API 및 스크립트 예제 문서를 임베딩하여 Vector DB에 저장함. 이를 통해 다양한 함수 호출 및 구조에 대한 의미 기반 검색이 가능해짐.** |
| **Texture** | **Text-to-Texture 생성 기능을 추가하여, 감성적/추상적 텍스트에 대해 이미지 텍스처 + 텍스트를 함께 전달함으로써 더 정밀하고 풍부한 3D 생성이 가능하도록 설계** |
| **Mesh** | **Text-to-Mesh 생성 기능을 추가하여, 기본 도형 조합을 넘어서는 자유로운 3D 형상 생성을 지원합니다.** |
|  |  |
|  |  |
|  |  |

## 요구사항 정의

주 사용자(고객)는 다음과 같다.

1. 3D 모델링에 대한 기본 개념은 있으나, Blender 스크립팅에 익숙하지 않은 비전문가

2. 빠르게 프로토타이핑을 하고자 하는 기획자 또는 창작자

요구사항은 다음과 같다.

1. 사용자가 **텍스트 입력을 통해 Blender에 3D 명령을 요청**할 수 있어야 한다.

2. MCP 서버는 **사용자의 요청을 적절한 Python 코드로 변환**하여 전달해야 한다.

3. Blender는 **외부에서 전달된 코드를 안전하게 실행하고, 결과를 뷰포트에 반영**해야 한다.

# 서비스 설계(예시)

## 텍스트, 의자, 가구, 스크린샷이(가) 표시된 사진 AI가 생성한 콘텐츠는 부정확할 수 있습니다.서비스 개념 및 구조도

사용자의 **자연어 명령을 Blender 파이썬 스크립트로 자동 변환**하여, **웹 브라우저 상에서 고급 3D 작업을 직관적으로 수행**할 수 있게 하는 **AI 기반 모델링 지원 시스템**입니다. 핵심은 **MCP(Model Command Proxy)** 서버 구조로, Flask 웹 서버 → FastAPI 명령 처리기 → LLM → Blender 애드온 순으로 연결되어, 입력된 명령어가 **즉시 코드로 생성·실행**되고, 결과가 **3D 뷰포트에 실시간 시각화**됩니다. 또한 “숲 느낌”과 같은 **감성적 키워드**는 이미지로 변환되어, 자연어와 함께 LLM에 전달됨으로써 **멀티모달 명령 생성**이 가능해집니다.  
이를 통해 사용자는 **감성적이고 창의적인 3D 표현을 손쉽게 구현**할 수 있습니다.

## UX 설계.

텍스트, 스크린샷, 멀티미디어 소프트웨어, 소프트웨어이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

## 데이터 설계

텍스트, 스크린샷, 양초, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.총 **1,761페이지 분량의 Blender Python 기술 문서**를 수집하여, 총 \*\*9,429개의 문서 단위(doc chunk)\*\*로 분할하고 임베딩을 수행하였습니다. 임베딩에는 all-MiniLM-L6-v2 모델을 사용하였으며, 결과는 FAISS 벡터 데이터베이스에 저장하였습니다. 사용자의 자연어 요청에 대해 유사 문서를 정밀하게 검색하고, 고급 함수까지 반영된 문맥 기반 응답을 통해 **정확하고 세밀한 코드 생성이 가능하도록** RAG 기반 설계를 적용하였습니다.

## 프로그램 모듈 구조

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

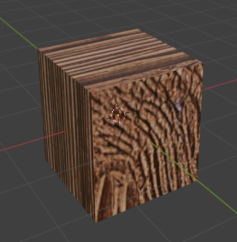
AI가 생성한 콘텐츠는 부정확할 수 있습니다.해당 서비스는 사용자의 자연어 프롬프트를 입력받아, LLM이 요청의 의도를 분석한 후 다섯 가지 기능 중 하나를 선택해 실행합니다. 기능에는 일반 코드 생성(rag\_normal), 삭제 명령(rag\_del), 메시 생성(rag\_mesh), 텍스처 생성(rag\_texture), 기본 명령(normal)이 포함됩니다. 메시 생성 요청 시 LLaMA 기반 text-to-3D 모델이, 텍스처 요청 시 Stable Diffusion 기반 text-to-image 모델이 활용되며, 생성 결과는 후처리 과정을 거쳐 Blender에서 실행 가능한 형태로 정제됩니다. 최종적으로 Gradio 챗봇 UI를 통해 결과가 제공되며, 사용자는 이를 확인하고 Blender MCP 서버에서 직접 실행할 수 있습니다.

# 테스트 계획

기존 프로젝트(AS-IS)와 신규 프로젝트(TO-BE)가 동일한 자연어 요청(prompt)에 대해 Blender에서 실행 가능한 Python 스크립트를 얼마나 정확하고 효율적으로 생성하는지를 비교·평가합니다. 특히 TO-BE 모델의 개선 효과를 계량적으로 증명하고자 한다.

# 구현

Prompt: 디테일한 나무 재질의 큐브 생성해줘 (texture 반영)

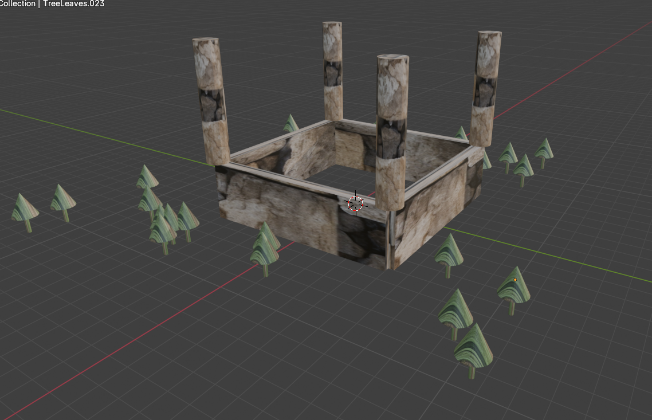
상자이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다. (AS-IS) (TO-BE)

봉투, 문구용품, 디자인이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.스크린샷, 디자인이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다. Prompt: 디테일한 자동차 모델 생성해줘 (mesh 반영)

Prompt: 판타지 영화에 나오는 중세 성을 만들어줘. 성 주위에는 나무가 많이 심어져있어. 성의 재질은 돌로 만들어줘 (texture, mesh 반영)

스크린샷, 3D 모델링이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

# 결과

Stable Diffusion 기반 텍스처 생성과 Text-to-Mesh를 활용한 3D 객체 생성은 기존의 단순한 색상 매핑이나 primitive 중심 모델링을 넘어, 시각적 표현력과 구성의 자유도를 크게 향상시켰다. 또한 의미 기반 문서 검색을 통해 정확하고 정교한 파이썬 코드를 자동 생성할 수 있게 하였다.

향후에는 음성 명령 처리, 실시간 피드백, 생성 품질 향상을 통해 더욱 직관적인 대화형 3D 모델링 플랫폼으로 확장될 수 있을 것으로 기대됩니다.

# 감사의 글

본 과제의 수행에 있어 귀중한 통찰과 실질적인 조언을 아끼지 않으신 LLM 관련 강의를 진행해주신 강사님께 깊은 감사의 말씀을 드립니다.  
또한, Blender 기반 MCP 시스템 구현에 있어 선례를 남겨 주신 GitHub 사용자 [@ahujasid](https://github.com/ahujasid)님의 오픈소스 프로젝트는 본 기획의 방향성과 구조 설계에 있어 큰 영감을 주었습니다. 진심으로 감사드립니다.

# 레퍼런스

본 과제 할 때 다음 도구 및 자료들을 참고하였다.

1. OpenAI GPT
2. Visily
3. Github Copilot
4. <https://huggingface.co/Zhengyi/LLaMA-Mesh>
5. <https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-v1-5>
6. <https://github.com/ahujasid/blender-mcp>
7. <https://docs.blender.org/api/current/index.html>
8. <https://github.com/CompVis/stable-diffusion>
9. [https://daddynkidsmakers.blogspot.com](https://daddynkidsmakers.blogspot.com/)