

자료구조 Term Project 제안서

팀명	뭐하지	학년-반	2학년 4반
과제명	KUBOS		
<p>1. 개요</p> <ul style="list-style-type: none"> ➤ 수준 1: 본 프로젝트는 초심자부터 전문가까지 모두 사용할 수 있는 통합 3D 웹 에디터, KUBOS를 개발하는 것을 목표로 함 ➤ 수준 1: Three.js와 Next.js 기반의 웹 환경에서 3D 모델링, 셰이더 <i>그래프</i> 및 코드 편집, 웹 임베딩 기능을 통합적으로 제공한다 ➤ 수준 1: 사용자는 단계별 튜토리얼과 직관적인 UI를 통해 3D 그래픽 제작의 쉬운 학습과 실습이 가능하다 <ul style="list-style-type: none"> - 수준 2: 초심자도 학습 절벽 없이 3D 그래픽 원리를 이해하고 활용 가능 - 수준 2: 전문가도 고급 셰이더 편집과 웹 임베딩 기능을 활용하여 효율적으로 콘텐츠 제작 가능 <p>2. 연구개발 목표 및 내용</p> <ul style="list-style-type: none"> ➤ 수준 1: 3D 그래픽 입문자와 전문가 모두가 사용 가능한 웹 기반 3D 에디터를 개발 ➤ 수준 1: 세부 목표로는 기초 메이커 환경 구축, FSM 기반 튜토리얼 구현, 셰이더 편집 기능, 웹 임베딩 기능 통합이 포함되어 개발 ➤ 수준 1: 이를 통해 학습과 제작, 배포를 하나의 환경에서 수행할 수 있는 효율적인 개발 플랫폼을 제공할 예정 <ul style="list-style-type: none"> - 수준 2: 입문자는 튜토리얼 기반 학습을 통해 3D 그래픽 기초를 직관적으로 습득가능 - 수준 2: 전문가와 개발자는 고급 기능을 활용하여 효율적으로 고품질 콘텐츠를 제작하고 공유 가능 <p>3. 기대 효과</p> <ul style="list-style-type: none"> ➤ 수준 1: 프로젝트 완료 시 3D 콘텐츠 제작의 기술적, 심리적 진입 장벽 감소 ➤ 수준 1: 국내 웹 3D 생태계 활성화와 함께 국산 3D 웹 에디터의 기술 경쟁력을 확보 가능 ➤ 수준 1: 교육, 디자인, 게임, 메타버스 등 다양한 분야에서 창의적인 아이디어 구현을 지원 <ul style="list-style-type: none"> - 수준 2: 학습자는 직관적인 튜토리얼과 실습으로 3D 그래픽 기술을 체계적으로 학습 가능 			
<p>위와 같이 Term Project 제안서를 제출합니다.</p> <div style="text-align: right; margin-top: 20px;"> 2025년 9월 30일 팀장: 신유준 (인) </div>			
담당교사 귀하			

목 차

1. 연구개발 배경 및 중요성 (Background and Significance)
 - 가. 국내외 관련 연구개발 동향
 - 나. 선행 연구개발의 문제점
 - 다. 연구개발 과제의 필요성 및 중요성
 - 라. 기대효과
2. 연구개발 목표 (Specific Aims)
 - 가. 최종 목표
 - 나. 세부과제별 연구개발 목표
 - 다. 기술개발 목표
 - 라. 목표 대상
3. 연구개발 내용 및 방법 (Research Design and Method)
 - 가. 연구개발 내용
 - 나. 연구개발 방법
 - 다. 연구개발 일정
4. 활용 기자재 및 재료 (Equipment Required)
 - 가. 개요
 - 나. 활용계획 및 기대효과
5. 참여연구원 현황 및 역할분담 (The Role of Researchers)
 - 가. 참여연구원 현황
 - 나. 프로젝트 수행 조직
 - 다. 참여연구원 간 역할분담
6. 참고문헌 (Reference)

표

- [표 1] 선행 연구개발의 문제점 및 개선 목표(요약)
[표 2] 세부과제별 목표(요약)
[표 3] 세부과제별 연구개발 일정
[표 4] 참여연구원 현황
[표 5] 참여연구원 간 역할분담 내역

그림

- [그림 1] 전체 연구과제의 구조도
[그림 2] 프로젝트 수행 조직도

프로젝트 제안서

1. 연구개발 배경 및 중요성 (Background and Significance)

가. 국내외 관련 연구개발 동향

본 프로젝트는 전문가용과 입문자용으로 양극화된 3D 웹 에디터 시장의 한계를 극복하고, Three.js 기술을 기반으로 초심자부터 전문가까지 모두 활용할 수 있는 통합 솔루션을 개발하는 것을 목표로 한다. 이를 위해 사용자가 3D 그래픽의 원리를 자연스럽게 학습할 수 있는 대화형 튜토리얼, 전문가 수준의 결과물을 제작할 수 있는 커스텀 셰이더 편집 기능, 제작 결과물을 웹에 손쉽게 게시할 수 있는 간편한 웹 임베딩 기능을 핵심으로 구현하고자 한다.

국내외 시장 동향을 살펴보면, 해외 3D 웹 에디터 시장은 기술 발전과 함께 사용자의 접근성과 전문성을 동시에 충족시키는 방향으로 빠르게 진화하고 있다. 대표적으로 Spline은 Figma와 유사한 직관적 인터페이스를 통해 3D 디자인 경험이 없는 사용자도 손쉽게 인터랙티브 3D 콘텐츠를 제작할 수 있도록 하여 시장의 저변을 확대하고 있다. 반면 PlayCanvas와 Babylon.js Editor는 단순한 모델링을 넘어 웹 기반 게임 개발, 물리 엔진 연동, 복잡한 씬 구성 및 스크립팅을 지원하는 고기능 플랫폼으로 자리 잡았다. 특히 Babylon.js의 노드 기반 셰이더 에디터는 코딩 없이도 고품질 재질을 시각적으로 구현할 수 있어, 아티스트와 개발자 간 협업 효율성을 극대화하며 전문가의 복잡하고 세밀한 요구사항까지 충족시키고 있다. 이러한 흐름은 사용자의 숙련도에 따라 단계적으로 깊이 있는 기능을 제공하는 통합 환경이 글로벌 표준으로 자리 잡고 있음을 보여준다.

반면 국내 시장은 범용 3D 웹 에디터 개발보다는 이커머스 3D 상품 뷰어, 부동산 및 인테리어 분야의 가상 쇼룸, 교육용 실감형 콘텐츠 등 특정 산업에 종속된 기능으로 웹 3D 기술을 활용하는 데 집중하고 있다. 이로 인해 국내 기술로 개발된 완성도 높은 범용 3D 웹 에디터가 부족하며, 한국어 교육 콘텐츠 부재와 국내 웹 환경에 최적화된 기능 개발의 필요성이 동시에 존재한다. 이러한 배경에서 본 프로그램 개발의 필요성이 제기되었다.

나. 선행 연구개발의 문제점

현재 3D 웹 에디터 시장은 전문가용 고기능 툴과 기능이 제한적인 입문자용 툴로 명확히 양극화되어 있다. 이러한 시장 구조는 입문자가 초급 툴의 기능적 한계에 부딪혔을 때, 전문가용 툴로 넘어가기 위한 가파른 학습 곡선 앞에서 성장을 포기하게 만드는 '학습 절벽' 현상을 야기한다. 이는 3D 웹 생태계 확장을 저해하는 주요 원인이 된다.

동시에 전문가는 간단한 시각화나 프로토타이핑 작업에도 불필요하게 무겁고 복잡한 툴을 사용해야 하는 비효율을 겪는다. 여러 툴을 오가야 하는 워크플로우 단절은 전체 개발 생산성을 저하시킨다. 또한 대부분의 고기능 툴이 해외에서 개발되어 공식 한국어 문서나 커뮤니티 지원이 부족

하므로, 국내 사용자는 정보 접근성에 제약을 받고 문제 해결에 어려움을 겪으며 시장에서 소외되는 근본적인 문제점을 안고 있다.

다. 연구개발 과제의 필요성 및 중요성

이러한 시장의 명확한 문제점을 해결하기 위해, 초심자부터 전문가까지의 성장 경로를 하나의 툴 안에서 끊임 없이 지원하는 통합 3D 웹 에디터 개발은 기술적·산업적으로 매우 중요하고 시급하다. 본 연구개발은 사용자가 3D 그래픽의 기본 원리를 배우고 그 결과물을 즉시 프로젝트에 활용할 수 있도록 체계적인 대화형 튜토리얼을 툴 내에 유기적으로 통합하여 학습과 제작의 선순환 구조를 구축하고자 한다. 이를 통해 지식 격차를 해소하고 잠재적 창작자를 생태계로 유입시키는 핵심 동력을 마련할 것이다.

또한 숙련도에 따라 ‘커스텀 셰이더(GLSL) 직접 편집’과 같은 전문가용 기능을 점진적으로 활용하게 함으로써 사용자가 툴을 이탈하지 않고 전문가로 성장할 수 있는 지속 가능한 환경을 제공하고, 플랫폼의 충성도를 극대화할 수 있다. 마지막으로 제작 결과물을 클릭 몇 번으로 웹사이트에 삽입할 수 있는 간편한 웹 임베딩 기능을 통해, 3D 콘텐츠가 소수 전문가의 전유물이 아닌 누구나 쉽게 공유하고 활용할 수 있는 대중적 매체로 자리 잡도록 기여할 것이다.

라. 기대효과

본 연구개발 과제가 성공적으로 완료되면, 기술적으로 3D 콘텐츠 제작의 진입 장벽을 혁신적으로 낮추어 국내 개발자 및 디자이너의 저변을 확대하고, 창의적 아이디어가 기술적 한계 없이 구현될 수 있는 환경을 조성할 것이다. 이는 국내 웹 3D 기술 생태계를 질적·양적으로 활성화하는 중요한 기반이 될 것이다.

경제·산업적 측면에서는 고가 외산 툴 의존도를 낮추어 기술 종속성을 해소하고, 국산 범용 에디터 부재로 인한 시장 공백을 선점하여 초기 시장에서 경쟁 우위를 확보할 수 있다. 또한 메타버스, 인터랙티브 광고, 실감형 교육 등 급성장하는 차세대 웹 산업 분야에서 국내 기업이 콘텐츠 경쟁력을 확보할 수 있는 핵심 인프라를 제공함으로써, 관련 산업의 동반 성장을 견인하고 글로벌 시장으로 확장할 수 있는 중요한 교두보 역할을 수행할 것으로 기대된다.

2. 연구개발 목표 (Specific Aims)

가. 최종 목표

[표 1] 선행 연구개발의 문제점 및 개선 목표(요약)

항목 번호	선행 연구개발의 문제점	개선 목표 및 관련 세부과제
----------	--------------	-----------------

1	높은 진입 장벽 및 학습 절벽 현상	초심자도 쉽게 배울 수 있는 끊임 없는 통합 환경 제공. -> 사용자가 툴 내에서 3D 툴의 원리를 체계적으로 학습하고 즉시 실습할 수 있는 기능 구현.
2	복잡한 툴 사용으로 인한 비효율	하나의 툴 안에서 학습, 제작, 배포가 모두 가능한 효율적인 개발 환경 조성. -> 제작된 3D 결과물을 클릭 몇 번으로 웹사이트나 포트폴리오에 손쉽게 게시하고 공유하는 기능 구현. 전문가 수준의 세밀한 재질 표현과 시각 효과를 직접 제어할 수 있는 고급 기능 제공.
3	외국 언어 의존도 과도	순수 한국인을 위한 툴은 없음 -> 한국어로 특화 되어있는 툴 제작

나. 세부과제별 연구개발 목표

[표 2] 세부과제별 목표(요약)

세부과제	목표
세부과제명 1	기초 메이커 툴 잡기
세부과제명 2	튜토리얼 과정(FSM) 구성 및 구현
세부과제명 3	셰이더 기능 구현
세부과제명 4	웹 임베딩 기능 및 통합 + 백엔드 연동

다. 기술개발 목표

본 프로젝트는 사용자의 접근성을 최우선으로 고려하여 웹 기반으로 개발될 예정이며, 3D 기능

구현을 위해 Three.js[4] 라이브러리를 활용할 계획이다. Three.js는 웹 환경에서 복잡한 3D 그래픽을 효율적으로 구현할 수 있는 도구로, 다양한 모델링과 애니메이션 기능을 제공한다. 그러나 3D 렌더링은 일반적으로 연산량이 많고 브라우저 환경에서는 성능 부담이 발생할 수 있으므로, 이러한 문제를 최소화하고 원활한 실행 환경을 제공하기 위해 SSR(Server-Side Rendering) 기능을 지원하는 Next.js를 함께 적용할 예정이다. SSR을 활용하면 초기 페이지 로딩 시 서버에서 렌더링을 수행하여 클라이언트의 부담을 줄이고, 사용자 경험을 향상시킬 수 있다. 이를 통해 웹 기반 3D 환경에서도 지연이나 렉 없이 안정적인 사용이 가능하도록 설계할 예정이다.

첫 번째로, 프로젝트의 튜토리얼 과정은 FSM[1]패턴을 기반으로 한 이벤트 처리 시스템을 활용하여 구현될 예정이다. 튜토리얼 시스템은 사용자가 단계별로 기능을 익힐 수 있도록 체계적으로 구성되며, 상태, 이벤트, 전환, 규칙이라는 네 가지 핵심 요소로 이루어진다. 상태는 튜토리얼의 현재 진행 단계를 의미하며, 사용자가 어느 위치에서 어떤 작업을 수행하고 있는지를 나타낸다. 이벤트는 사용자가 에디터 내에서 수행하는 구체적인 행동으로, 버튼 클릭, 모델 선택, 값 입력 등 다양한 상호작용을 포함한다. 전환은 특정 상태에서 이벤트가 발생했을 때 다음 단계로 이동하는 과정으로 정의되며, 튜토리얼의 흐름을 제어하는 중요한 요소이다. 규칙은 각 상태를 완료하기 위해 필요한 조건이나 이벤트를 결정하는 기준으로 작용한다. 이러한 FSM 기반 설계를 통해 튜토리얼의 각 단계가 명확하게 정의되고, 사용자의 입력에 따라 동적으로 진행될 수 있도록 구현될 예정이다. 본 FSM 알고리즘은 JavaScript의 Object 구조를 활용하여 직관적이고 효율적으로 구현할 계획이다.

두 번째로, 프로젝트에는 셰이더[2][3] 기능이 포함될 예정이다. 이를 통해 사용자는 웹 브라우저 환경에서도 전문가 수준의 실시간 그래픽 작업을 수행할 수 있게 된다. 셰이더 기능 구현을 위해 React의 상태 관리 기능을 활용하여, 동적으로 유니폼 UI를 생성하는 기법을 적용할 계획이다. 구체적으로, 셰이더 변수의 타입에 따라 슬라이더, 컬러 피커, 체크박스 등 적절한 제어 UI를 자동으로 생성하며, 생성된 UI와 셰이더 간 데이터가 양방향으로 동기화되도록 설계된다. 이를 통해 사용자는 UI를 통해 값의 변화를 즉시 확인할 수 있으며, 셰이더 코드를 직접 수정하지 않더라도 다양한 실험과 조절이 가능하다. 또한, 이러한 구조는 새로운 셰이더 변수를 추가하거나 수정할 때도 자동으로 UI가 반영되도록 설계되어 확장성과 유지보수성을 높인다.

마지막으로, 프로젝트는 웹 임베딩[5] 기능을 제공하여, 사용자가 자신만의 웹 환경에서 에디터의 결과물을 쉽게 활용할 수 있도록 구성될 예정이다. 사용자는 에디터 내 코드 스니펫을 통해 지금까지 작성된 셰이더와 3D 모델, 설정 값 등을 포함한 코드를 복사할 수 있으며, 이를 자신의 웹사이트에 삽입함으로써 즉시 결과를 확인할 수 있다. 이러한 기능은 프로젝트에서 생성한 그래픽과 모델을 재사용하거나, 외부 웹 페이지에서 시각적으로 통합하는 과정을 간소화하며, 사용자의 편의성을 크게 향상시킨다. 이를 통해 웹 기반 3D 에디터가 단순한 개발 도구를 넘어, 실질적인 그래픽 제작과 배포 환경으로 활용될 수 있도록 설계된다.

라. 목표 대상

본 프로젝트의 목표 대상은 크게 두 가지로 나눌 수 있다. 첫 번째 대상은 3D 그래픽 및 웹 개발 분야의 학습자이며, 두 번째 대상은 현업 웹 개발자 및 콘텐츠 제작자이다. 먼저, 3D 그래픽 및 웹 개발 분야의 학습자에는 관련 전공을 이수하는 대학생, 직업 교육을 받는 학습자, 혹은 독학으로 기술을 습득하는 개인 학습자 등이 포함된다. 이들은 기존의 복잡한 3D 도구, 예를 들어 블렌더와 같은 전문 툴을 활용할 경우 학습 과정에서 높은 난이도와 진입 장벽으로 인해 어려움을 겪는 경우가 많다. 반대로, 지나치게 단순화된 도구를 사용할 경우에는 전문가가 전문적인 작업을 수행하기 어렵다는 문제점도 존재한다. 본 프로젝트에서 개발하는 통합 에디터는 이러한 문제를 해결하기 위해 체계적이고 대화형 튜토리얼을 제공한다. 이를 통해 학습자는 Three.js와 셰이더의 기본 원리를 시각적으로 이해하고, 학습 결과를 즉시 3D 뷰포트에서 확인함으로써 직관적인 지식을 습득할 수 있다. 결과적으로 3D 기술의 진입 장벽이 낮아지고, 학습 동기와 성취도가 크게 향상되는 효과를 기대할 수 있다. 다음으로, 현업 웹 개발자 및 콘텐츠 제작자는 웹사이트에 3D 요소를 추가하고자 할 때, 간단한 기능을 구현하기 위해 다수의 Three.js 코드 라인을 작성하거나, 복잡한 GLSL 코드를 직접 작성해야 하는 비효율적인 상황에 직면할 수 있다. 본 에디터는 이러한 문제를 해결하기 위해 개발되었으며, 개발자는 에디터를 활용하여 아이디어를 신속하게 프로토타이핑하고, 시각 효과를 빠르게 개발할 수 있다. 또한 완성된 결과물을 간편한 임베딩 기능을 통해 자신의 웹 프로젝트에 즉시 통합할 수 있으므로, 개발 생산성을 높이고 작업 효율을 극대화할 수 있는 장점을 제공한다. 나아가, 본 프로젝트의 산출물은 디자인 및 시각 예술 분야의 학생과 컴퓨터 그래픽 교육자에게도 유용하게 활용될 수 있다. 코딩에 익숙하지 않은 디자이너들은 직관적인 제어 UI를 활용하여 자신의 시각적 아이디어를 직접 구현해 보는 경험을 얻을 수 있으며, 교육자들은 복잡하고 추상적인 셰이더 렌더링 과정을 시각적으로 시연할 수 있는 교육 도구로 활용하여 강의를 보다 효과적이고 이해하기 쉽게 진행할 수 있을 것으로 예상된다.

3. 연구개발 내용 및 방법 (Research Design and Method)

가. 연구개발 내용



[그림 1] 전체 연구과제의 구조도

세부과제 1 - 기초 메이커 툴 잡기

초기 환경은 Next.js와 TypeScript를 기반으로 구축하여 타입 안정성과 개발 생산성을 확보하고, 3D 렌더링은 react-three-fiber와 drei 라이브러리를 통해 Three.js를 제어하는 방식으로 구현합니다. UI 레이아웃은 React 컴포넌트 단위로 설계하여 전체 화면을 코드 에디터, 3D 뷰포트, 속성 패널의 3단 구조로 분리하여 개발합니다. 중앙 3D 뷰포트 컴포넌트 내부에는 react-three-fiber의 <Canvas>를 배치하고, 그 안에 기본적인 <mesh> 객체와 조명, 그리고 사용자가 씬을 쉽게 조작할 수 있도록 drei의 <OrbitControls> 같은 카메라 컨트롤러를 포함시켜 기본 3D 씬을 렌더링합니다.

세부과제 2 - 튜토리얼 과정(FSM) 구성 및 구현

튜토리얼 기능은 유한 상태 머신(FSM) 패턴을 기반으로 구현합니다. 먼저, 각 튜토리얼 단계를 목표, 지시문, 완료 규칙 등을 포함하는 JSON 객체 배열로 명확하게 정의하여 데이터 구조를 설계합니다. React의 useState 훅을 사용하여 현재 진행 중인 튜토리얼의 상태를 관리하고, 이 상태값에 따라 튜토리얼 UI 컴포넌트는 해당 단계의 지시문을 동적으로 렌더링하며 관련된 UI 요소를 시각적으로 강조합니다. 사용자의 행동은 에디터 전역에서 발생하는 이벤트 핸들러를 통해 감지되며, 이 이벤트가 현재 단계의 완료 규칙을 충족시키는지 실시간으로 검증하여 상태를 다음 단계로 전환시켜 튜토리얼을 자연스럽게 진행합니다.

세부과제 3 - 셰이더 기능 구현

전문가 수준의 셰이더 편집을 위해 Monaco Editor를 React 컴포넌트로 통합하여 GLSL 구문 강조 및 자동 완성 기능을 제공합니다. 사용자가 코드를 수정하면, 디바운스(기법을 적용한 이벤트 핸들러가 이를 감지하여 불필요한 재 컴파일을 방지하고 안정적으로 THREE.ShaderMaterial을 실시간으로 업데이트합니다. 또한, 셰이더의 유니폼(Uniform) 변수들은 React의 useState로 관리되는 객체 배열을 기반으로, 타입에 맞는 UI 컨트롤(슬라이더, 컬러 피커 등)을 동적으로 생성합니다. 사용자가 이 컨트롤을 조작하면 상태가 변경되고, 이는 즉시 3D 뷰포트의 머티리얼에 반영되어 직관적인 편집 경험을 제공합니다.

본 과제의 핵심 목표는 사용자가 제작한 3D 씬을 다른 웹사이트에 즉시 삽입할 수 있는 HTML 임베드 코드를 생성하여 반환하는 것입니다.

이 기능을 구현하기 위해, 먼저 현재 씬을 구성하는 모든 상태 정보(셰이더 코드, 유니폼 값 등)를 하나의 데이터(JSON)로 변환합니다. 이 데이터는 백엔드 서버에 전송되어 고유 ID와 함께 데이터 베이스에 저장됩니다. 최종적으로, 이 고유 ID를 포함한 <iframe> 태그가 사용자에게 제공되어, 복사-붙여넣기만으로 자신의 창작물을 외부 프로젝트에 손쉽게 통합하고 공유할 수 있게 합니다. 또한, 서버에 저장된 데이터는 사용자가 나중에 자신의 작업을 다시 불러와 수정하고 관리할 수 있는 기반이 됩니다.

나. 연구개발 방법

1) 요구분석 방법

연구개발 프로젝트의 요구사항 분석은 국내외 시장 동향 분석, 사용자 설문조사, 온라인 커뮤니티 의견 수집, 전문가 자문 등을 통해 체계적으로 진행되었다. 우선 최근 발표된 산업 동향 보고서와 실무 가이드, 그리고 웹3D 관련 기술 세미나의 객관적 자료를 기반으로 국내 사용자들이 겪는 주요 문제들이 확인되었다. 이러한 자료에 따르면 국내 3D 웹 에디터 시장은 범용 툴의 부재, 복잡한 외산 3D 툴의 기술적·언어적 장벽, 그리고 공식적인 한국어 지원 부족으로 인한 불편함이 지속적으로 지적되고 있다.

또한 실제 사용자들의 요구사항을 확인하기 위하여 개발자, 디자이너, 웹 콘텐츠 제작자 등을 대상으로 설문조사와 온라인 커뮤니티 의견 조사를 병행하였다. 그 결과, 국내 사용자들은 3D 툴의 높은 복잡성, 영어 기반 UI 및 매뉴얼에 대한 부담, 전문가용과 초심자용 툴 간의 기능 격차, 결과물을 웹 환경에 쉽게 게시하기 어려운 점, 그리고 공식적인 한국어 문서 부족 등의 문제를 경험하는 것으로 나타났다.

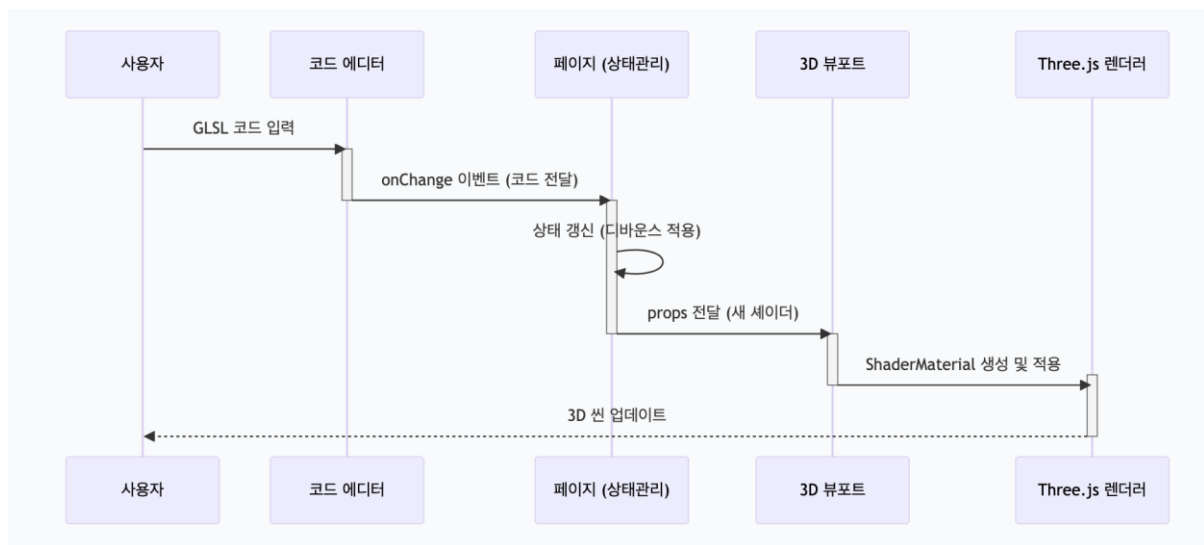
추가적으로 웹3D 기술 현업 전문가, 교육 및 이커머스 분야 전문가, UI·UX 연구자와의 심층

인터뷰를 통해 산업 및 교육 현장에서 나타나는 기능적 한계, 협업의 비효율성, 기술 적응의 어려움 등을 확인하였다. 특히 국내 시장은 이커머스, 가상 쇼룸, 실감형 콘텐츠 등 특정 산업 중심으로만 웹3D 기술이 활용되어 왔으며, 범용 3D 웹 에디터의 필요성이 크다는 의견이 지배적이었다.

이와 같이 연구개발 착수 단계에서 산업 동향 자료 분석, 사용자 설문 및 커뮤니티 의견 수집, 전문가 자문을 종합적으로 반영함으로써 실제 사용자가 겪는 불편을 객관적으로 도출할 수 있었으며, 이는 본 프로젝트의 개발 방향과 세부 기능 목표를 설계하는 중요한 근거가 되었다.

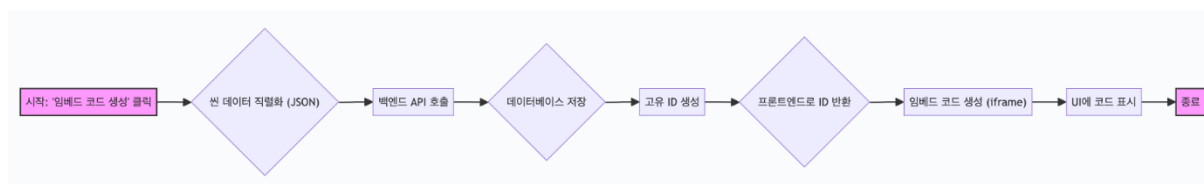
2) 설계

1. 실시간 셰이더 편집 다이어그램



본 에디터의 핵심 기능인 '실시간 셰이더 편집' 과정에서 발생하는 객체 간 상호작용을 순서 다이어그램으로 나타낸다. 사용자가 코드 에디터에 GLSL 코드 기반을 입력하면, onChange 이벤트가 상위 페이지 컴포넌트로 전달된다. 상위 페이지는 디바운스 로직을 통해 불필요한 업데이트를 방지한 후, 셰이더 상태를 갱신하여 뷰포트 컴포넌트를 리렌더링한다. 뷰포트는 변경된 셰이더 코드를 props로 전달받아 새로운 ShaderMaterial을 생성하고, Three.js 렌더러는 이를 3D 모델에 적용하여 최종적으로 업데이트된 화면을 사용자에게 보여준다. 이 과정을 통해 사용자의 입력과 시각적 피드백 사이의 상호작용이 실시간으로 이루어진다.

2. 웹 임베딩 코드 생성



'웹 임베딩 코드 생성' 기능의 전체적인 작업 흐름을 액티비티 다이어그램으로 설계했다. 사용자가 '임베드 코드 생성' 버튼을 클릭하는 것으로 프로세스가 시작된다. 프론트엔드는 현재 3D 씬의 모든 상태를 JSON 객체로 직렬화하여 백엔드 API로 전송한다. 서버는 이 데이터를 데이터베이스

에 저장하고, 해당 썸을 식별할 고유 ID를 생성하여 다시 프론트엔드로 반환한다. 프론트엔드는 이 ID를 사용하여 최종적으로 <iframe src=".../embed/{ID}"> 형태의 HTML 코드를 생성하고, 이를 UI에 표시하여 사용자가 복사할 수 있도록 제공하며 작업 흐름이 종료된다.

3) 구현

개발 제작에 있어서 IDE는 VSC를 사용할 예정이고, NextJS, react-three-fiber, drei, Monaco Editor등을 프론트엔드 개발할때 사용될 예정이다.

4) 실험

본 과제 결과물의 실효성을 검증하기 위해, 차후 3D 그래픽 입문자와 현직 웹 개발자 두 그룹을 대상으로 사용성 테스트를 진행할 계획이다. 입문자에게는 튜토리얼 기반 과제를 부여하여 기존 툴 대비 학습 용이성을 과업 완료 시간과 만족도로 평가한다. 현직 개발자에게는 실무 중심의 셰이더 제작 과제를 제시하고, 심층 인터뷰를 통해 기존 개발 환경 대비 작업 효율성과 생산성 향상 정도를 평가할 것이다. 이 다각적인 평가를 통해 본 에디터가 목표로 하는 가치를 객관적으로 입증하고자 한다.

다. 연구개발 일정

[표 3] 세부과제별 연구개발 일정

세 부 과 제	연구 개발 일정							비 고
	9월 2/2	10월 1/2	10월 2/2	11월 1/2	11월 2/2	12월 1/2	12월 2/2	
3D 렌더링 엔진								
모델링 툴								
셰이더 그래프 툴								
튜토리얼								
임베딩								

4. 활용 기자재 및 재료 (Equipment Required)

가. 개요

본 연구에서는 서버 환경을 직접 구축하고 운영하며 웹 기반 시스템을 실습할 수 있도록 소형 서버 장비를 활용한다. 이를 위해 라즈베리파이 5(16GB 모델)를 중심 장비로 선정하였으며, 장시간 안정적인 운영을 위해 액티브 쿨러를 장착하고, 운영체제와 프로젝트 데이터를 저장하기 위해 내구성이 높은 마이크로 SD카드를 사용하였다. 이러한 기자재 구성은 프로젝트 수행에 필요한 최소한의 서버 환경을 제공하며, 실제 서버 운영 과정을 구현할 수 있도록 지원한다.

나. 활용계획 및 기대효과

라즈베리파이 5는 본 프로젝트에서 웹 서버와 데이터 처리 환경을 구현하는 핵심 장비로 활용된다. 이를 통해 서버 설치, 구성, 운영 과정을 직접 수행할 수 있으며, 실제 환경에서 발생할 수 있는 문제에 대응할 수 있다. 액티브 쿨러는 라즈베리파이 5가 장시간 작동할 경우 발생할 수 있는 과열 문제를 방지하여 안정적인 성능을 유지하도록 한다. 마이크로 SD카드는 서버 운영에 필요한 운영체제와 데이터를 저장하는 역할을 수행하며, 반복적인 데이터 기록에도 안정성을 제공함으로써 프로젝트 수행 중 발생할 수 있는 데이터 손실을 최소화한다.

이와 같이 본 프로젝트는 라즈베리파이 5와 관련 기자재를 활용하여 실제와 유사한 서버 환경을 구축하고 운영하는 과정을 가능하게 한다. 이를 통해 프로젝트 수행에 필요한 서버 환경을 안정적으로 제공하고, 장비 구성과 운영 과정에서 발생할 수 있는 다양한 기술적 문제를 해결할 수 있도록 지원한다. 또한 이러한 환경은 향후 유사한 프로젝트나 연구 과제 수행 시 응용할 수 있는 기반을 제공할 것으로 기대된다.

5. 참여연구원 현황 및 역할분담 (The Role of Researchers)

가. 참여연구원 현황

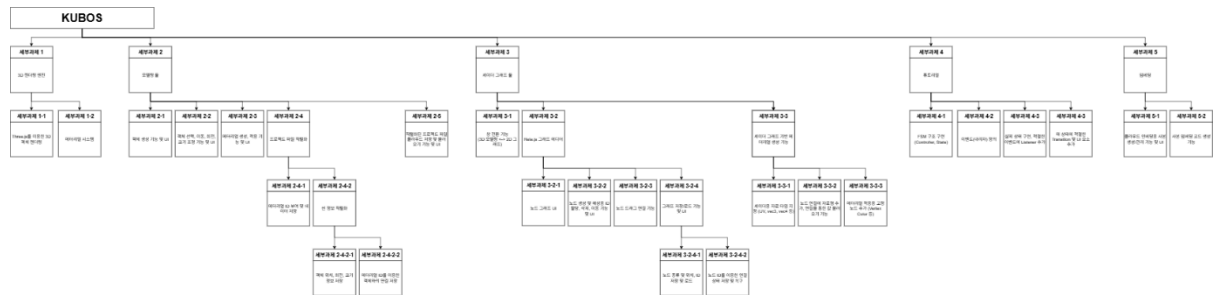
[표 4] 참여연구원 현황

학과	학번	이름	학교 이메일 주소
소프트웨어	20412	신유준	24sunrin090@sunrint.hs.kr
소프트웨어	20414	원현서	24sunrin111@sunrint.hs.kr
소프트웨어	20418	장현원	24sunrin076@sunrint.hs.kr

나. 프로젝트 수행 조직도

<https://drive.google.com/file/d/1znw54q9tttEgdloWuP7rwgpscaDiEWiQ/view?usp=>

sharing



[그림 2] 프로젝트 수행 조직도

다. 참여연구원 간 역할분담 내역

[표 5] 참여연구원 간 역할분담 내역

연구원	역할분담 세부내역	비고
신유준	<ul style="list-style-type: none"> - Next.js를 이용한 3D 렌더링 엔진, 모델링 툴 및 메타리얼 시스템 제작 - Rete.js를 이용한 노드 그래프 시스템 제작, 이를 이용한 셰이더 작성용 비주얼 프로그래밍 툴 제작 (A.K.A 셰이더 그래프) - 셰이더 그래프를 메타리얼 시스템에 통합해 3D 렌더링 엔진에 적용 - 셰이더 그래프 => GLSL 코드 인터프리터 제작 - 클라우드 저장 시스템과 연동된 3D 웹 임베딩 시스템 제작 - 자료 조사 및 문서 작성 	없음
원현서	<ul style="list-style-type: none"> - 3D 모델, 메타리얼 및 셰이더 그래프 파일의 클라우드 저장 시스템 제작 - 자료 조사 및 문서 작성 	없음
장현원	<ul style="list-style-type: none"> - 프론트엔드 UI 디자인 - 자료 조사 및 문서 작성 	없음

6. 참고문헌 (Reference)

- [1] D. Harel, "Statecharts: A visual formalism for complex systems," *Science of Computer Programming*, vol. 8, no. 3, pp. 231-274, 1987.
- [2] The Khronos Group, "The OpenGL Shading Language (GLSL) Specification, Version 4.60," 2018. [Online]. Available: . (Accessed: Sep. 30, 2025).
- [3] S. L. Tanimoto, "A perspective on the evolution of liveness in programming," in *Proceedings of the 2013 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2013, pp. 183-186.
- [4] R. Cabello et al., "Three.js – JavaScript 3D Library." [Online]. Available: . (Accessed: Sep. 30, 2025).
- [5] D. Crockford, "The JSON Data Interchange Format," RFC 8259, Dec. 2017. [Online]. Available: . (Accessed: Sep. 30, 2025).