


# 캡스톤디자인 중간보고서

제 목	국문		웹 기반 메타버스 구축 플랫폼 구현	
	영문		Implementation of a web-based metaverse building platform	
진 행 상 황	중요 마일스톤	1. 기능 요구사항에 대한 중요 마일스톤 1.1. Region 생성 및 삭제 및 씬(Scene) 연결 - 프로토타입 1.2. 씬(Scene) 생성/읽기/삭제 - 프로토타입 및 360도 이미지 없이 디폴트 배경 씬 생성에 대한 소스코드 추가 1.3. 씬(Scene) 편집 - 내부 GUI button을 통한 3D Object 생성 소스코드 작성 생성된 3D Object DB 연결 및 저장 소스코드 작성 3D Object Position/Rotate/Scale 조정 소스코드 작성 1.4. Region 공유 - link 공유를 위한 button 소스코드 작성 2. 성능 요구사항에 대한 중요 마일스톤 2.1. 웹 페이지 로딩 시간 - 웹 페이지 로딩 시간 테스트 2.2. 동시 사용자 접속 수 - 알파테스트를 통한 동시 사용자 수 테스트 2.3. 데이터 형식 오류 응답 시간 - 플랫폼에서 발생할 수 있는 오류들에 대한 예외처리 3. 인터페이스 요구사항에 대한 중요 마일스톤 3.1. 3D Object GUI - 3D Object 사용시의 필요 기능 정리 3.2. 오류 웹페이지 안내 - 사용자 입장에서 발생할 수 있는 오류들에 대한 오류 테스트 4. 데이터 요구사항에 대한 중요 마일스톤 4.1. 데이터베이스 구축 및 저장 - 데이터베이스 서버 구축 및 오픈, 연결 소스코드 작성 4.2. 데이터 수정 - 저장 데이터 불러오기 및 수정 소스코드 작성 5. 테스트 요구사항에 대한 중요 마일스톤 5.1. 테스트 계획 수립 - 플랫폼을 활용한 테스트 수업 계획 작성 5.2. 테스트 수행 - 계획에 맞춘 테스트 수업 진행 5.3. 단위 테스트 - 각 기능에 대한 테스트 수행 5.4. 알파 테스트 - 해당 플랫폼을 활용한 수업 진행 후 테스트 결과 취합		
		진행 상황	1.1 Region 생성 및 삭제 및 씬(Scene) 연결 - 프로토타입으로 완료 1.2 씬(Scene) 생성/읽기/삭제 - 프로토타입 및 360도 이미지 없이 디폴트 배경 씬 생성에 대한 소스코드 추가 완료 1.2.1 내부 GUI button을 통한 3D Object 생성 소스코드 작성 완료 1.2.2 생성된 3D Object DB 연결 및 저장 소스코드 작성 완료 1.2.3 3D Object Position 조정 소스코드 작성 완료 4.1 데이터베이스 구축 및 저장 - 데이터베이스 서버 구축 및 오픈, 연결 소스코드 완료 5.1 테스트 계획 수립 - 플랫폼을 활용한 테스트 수업 계획 완료 5.2 테스트 수행 - 계획에 맞춘 테스트 수업 진행 1, 2차 완료	
	산출물		요구사항 정의서(별첨 1), 중간보고서(별첨 2)	
팀 구성원	학년	학 번	이 름	연락처(전화번호/이메일)
	4	20191792	최진아	010-3268-5527/20191792@edu.hanbat.ac.kr
	4	20191750	이혜진	010-3589-8597/20191750@edu.hanbat.ac.kr
	4	20191740	유선아	010-5036-0149/20191740@edu.hanbat.ac.kr
컴퓨터공학과와 프로젝트 관리규정에 따라 다음과 같이 요구사항 정의서와 중간보고서를 제출합니다				
2022 년 5 월 6 일				
책임자 : 최진아 				
지도교수 : 최창범 (인)				

[별첨1]

프로젝트명 : 웹 기반 메타버스 구축 플랫폼 구현

# 소프트웨어 요구사항 정의서

Version 1.0

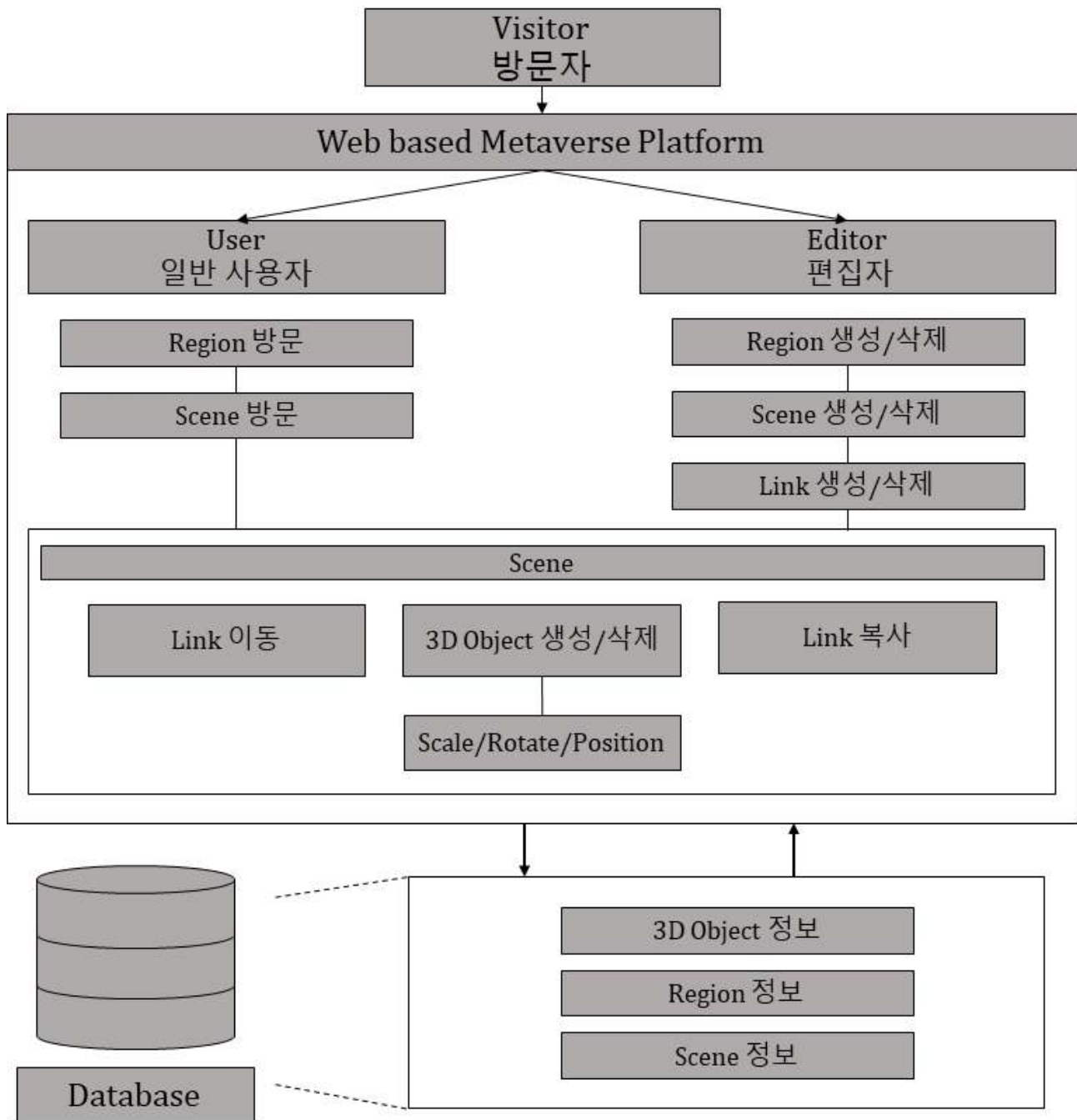
개발 팀원 명(팀리더):최진아  
이혜진  
유선아

대표 연락처:010-3268-5527  
e-mail: 20191792@edu.hanbat.ac.kr

## 목차

1. 개요
2. 시스템 장비 구성요구사항
3. 기능 요구사항
4. 성능 요구사항
5. 인터페이스 요구사항
6. 데이터 요구사항
7. 테스트 요구사항
8. 보안 요구사항
9. 품질 요구사항
10. 제약 사항
11. 프로젝트 관리 요구사항

## 1. 시스템 개요



## 2. 시스템 장비 구성요구사항

요구 사항 명칭	고유 번호	요구사항 정의 및 세부내용	산출 정보	관련 요구 사항
통신 서버	ECR-01	<p>정의 : 실시간 데이터 전송을 위한 TCP-IP 통신서버</p> <ul style="list-style-type: none"> <li>• 해당 라즈베리파이4를 웹 페이지를 운영하기 위한 서버로 사용함</li> <li>• 수량 : 1EA</li> <li>• 라즈베리파이4 2GB</li> <li>• Broadcom BCM2711, 쿼드 코어 Cortex-A72 (ARM v8) 64 비트 SoC @ 1.5GHz</li> <li>• 메모리 : 2GB</li> <li>• 연결성 : 2.4GHz 및 5.0GHz IEEE 802.11b / g / n / ac 무선 LAN, Bluetooth 5.0, BLE</li> </ul>		
O/S 소프트웨어	ECR-02	<p>정의 : O/S 소프트웨어</p> <ul style="list-style-type: none"> <li>• 인터넷에 접속이 가능한 모든 소프트웨어 지원 가능(Window, Linux, Max OS, Ubuntu 등)</li> </ul>		
DB 소프트웨어	ECR-03	<p>정의 : DB 소프트웨어</p> <ul style="list-style-type: none"> <li>• 소프트웨어 : MongoDB</li> <li>• 기능 : DB 관리</li> <li>• 성능 및 특징 : 수평적 확장이 가능한 Document 지향 데이터베이스</li> </ul>		

### 3. 기능 요구사항

요구 사항 명칭	고유 번호	요구사항 정의 및 세부내용	산출 정보	관련 요구 사항
Region 생성 및 삭제	SFR-01	정의 : 사용자만의 Region을 생성하여 Region 내의 원하는 씬을 구성함 <ul style="list-style-type: none"> <li>12 bytes binary 숫자에 대한 MongoDB 24자리 hex 문자열 ID를 자동으로 부여하여 Region을 생성함</li> <li>추가(ADD), 삭제(DELETE) 버튼을 눌러 생성한 Region 내의 각 씬을 추가하거나 삭제하고 자율적으로 구성함</li> </ul>		
씬(Scene) 생성/읽기/삭제	SFR-02	정의 : 360° 4K 이미지를 배경으로 가상공간인 Scene을 생성/삭제함 <ul style="list-style-type: none"> <li>사용자가 원하는 이미지를 씬의 배경으로 하여 Scene을 생성(Create)하여 DB에 저장함</li> <li>DB에 저장된 Scene을 불러와 읽음(Read)</li> <li>Scene에 대해 수정(Update)을 가한 뒤 수정된 데이터를 DB에 반영함</li> <li>DB에 저장되어있는 Scene 데이터를 삭제&gt;Delete)함</li> </ul>		
씬(Scene) 편집	SFR-03	정의 : 사용자가 생성한 씬을 주어진 오브젝트를 이용하여 꾸밀 수 있음 <ul style="list-style-type: none"> <li>3D Object의 CRUD 기능을 구현함</li> <li>3D Object를 생성(Create)하여 DB에 저장함</li> <li>DB에 저장된 3D Object를 불러와 읽음(Read)</li> <li>3D Object에 대해 수정(Update)을 가한 뒤 수정된 데이터를 DB에 반영함</li> <li>DB에 저장되어있는 데이터를 삭제&gt;Delete)함</li> </ul>		
씬(Scene) 연결	SFR-04	정의 : 각 씬들을 link 오브젝트를 통해 연결할 수 있음 <ul style="list-style-type: none"> <li>상하좌우에 link 오브젝트를 생성할 수 있음</li> <li>각 link를 scene 이름을 통해 연결함</li> </ul>		
3D object Repository 구축	SFR-05	정의 : 사용자가 사용할 수 있는 3D Object를 불러옴 <ul style="list-style-type: none"> <li>상자(box), 원기둥(cylinder) 등 3D Object를 불러옴</li> </ul>		
Region 공유	SFR-06	정의 : 생성한 Region을 URL로 공유하여 다수의 사용자가 접속하여 봄 단축 아이콘을 통해 URL을 공유함		

#### 4. 성능 요구사항

요구 사항 명칭	고유 번호	요구사항 정의 및 세부내용	산출 정보	관련 요구 사항
웹 페이지 로딩 시간	PER-01	<ul style="list-style-type: none"><li>• 플랫폼 이용 사용자의 화면에 가상 세계가 로딩되는 시간</li><li>• 각 가상 세계 내에 배치된 3D object들을 포함한 전체 웹 페이지가 사용자가 요구한 시점으로부터 10초 이내에 완벽하게 로딩되어야 함</li></ul>		
동시 사용자 접속 수	PER-02	<ul style="list-style-type: none"><li>• 웹페이지 당 동시 사용자 수 50명 이상을 지원해야 하고 성능이 저하되지 않아야 함</li></ul>		
데이터 형식 오류 응답 시간	PER-03	<ul style="list-style-type: none"><li>• 사용자로부터 발생한 모든 오류에 대해서 5초 이내에 오류 웹페이지 또는 메시지를 전송해야 함</li></ul>		

## 5. 인터페이스 요구사항

요구 사항 명칭	고유 번호	요구사항 정의 및 세부내용	산출 정보	관련 요구 사항
3D Object GUI	SIR-01	정의 : 사용자가 3D Object와 관련된 기능 사용 시의 인터페이스 <ul style="list-style-type: none"> <li>• 사용자는 구체적인 사용법 없이도 인터페이스만을 보고 3D Object를 생성, 추가, 편집할 수 있어야 함</li> </ul>		
사용자/편집자 구분	SIR-02	정의 : 사용자/편집자 구분 <ul style="list-style-type: none"> <li>• 사용자는 새로운 가상세계를 생성 또는 삭제하고 싶을 경우 편집자 모드를 통해서 원하는 대로 편집할 수 있어야 함</li> </ul>		
오류 웹 페이지 안내	SIR-03	정의 : 사용자의 잘못된 입력으로 인한 오류 발생 시의 오류 웹 페이지 <ul style="list-style-type: none"> <li>• 해당 웹 플랫폼을 사용하면서 사용자가 잘못된 입력을 할 경우 발생한 오류에 관한 설명을 포함한 오류 웹 페이지를 사용자에게 제시해야 함</li> </ul>		



## 6. 데이터 요구사항

요구사항 명칭	고유 번호	요구사항 정의 및 세부내용	산출 정보	관련 요구사항
데이터베이스 구축	DAR-01	<p>정의 : 사용자가 서비스를 이용하면서 생성된 데이터를 보관하기 위한 데이터베이스를 구축함</p> <ul style="list-style-type: none"> <li>내부 데이터베이스는 MongoDB를 사용함</li> <li>데이터베이스는 관리자 외에 접근할 수 없음</li> </ul>		
데이터 저장	DAR-02	<p>정의 : 사용자가 서비스를 이용하면서 생성된 데이터를 내부 데이터베이스에 저장함</p> <ul style="list-style-type: none"> <li>데이터베이스에 저장된 정보는 사용자가 삭제하지 않는 이상 보관함</li> </ul>		
데이터 수정	DAR-03	<p>정의 : 사용자가 서비스를 이용하면서 생성된 데이터를 수정할 때 내부 데이터베이스에도 반영되어야 함</p> <ul style="list-style-type: none"> <li>데이터베이스에 저장된 정보는 사용자가 수정하는 즉시 반영되어야 함</li> </ul>		

## 7. 테스트 요구사항

요구사항 명칭	고유 번호	요구사항 정의 및 세부내용	산출 정보	관련 요구사항
테스트 계획 수립	TER-01	<p>정의 : 요구사항이 제대로 반영되었는지를 테스트하기 위해서 테스트 계획을 수립함</p> <ul style="list-style-type: none"> <li>테스트 단계별 수행할 테스트와 목표, 활동, 보완 절차를 수립함</li> <li>테스트 대상, 기간, 성능 목표, 범위, 방법, 시나리오, 정량화된 측정지표 등 상세 계획을 수립함</li> </ul>		
테스트 수행	TER-02	<p>정의 : 테스트 계획을 바탕으로 테스트를 실시함</p> <ul style="list-style-type: none"> <li>설계된 단위에 맞추어서 수시로 단위 테스트를 진행함</li> <li>개발된 시스템이 시나리오에 맞게 완벽하게 수행되는지에 맞춰서 알파 테스트를 진행함</li> </ul>		
단위 테스트	TER-03	<p>정의 : 단위 테스트 계획과 예상결과를 포함하는 계획서 작성</p> <ul style="list-style-type: none"> <li>개발 단계에 따라 개발자 수준에서 단위 테스트를 수행함</li> </ul>		
알파 테스트	TER-04	<p>정의 : 알파 테스트 계획과 예상결과를 포함하는 계획서 작성</p> <ul style="list-style-type: none"> <li>시스템의 기능이 정상적으로 동작하는지 확인하기 위해서 계획 및 상세 시나리오를 작성하고 테스트를 수행함</li> </ul>		

## 8. 보안 요구사항

요구 사항 명칭	고유 번호	요구사항 정의 및 세부내용	산출 정보	관련 요구 사항
장비 보안 요구사항	SER-01	<p>정의 : 시스템을 운영하는 통신 서버의 물리적 · 시스템적 보안에 대한 요구사항</p> <ul style="list-style-type: none"> <li>통신 서버에 물리적으로 접근할 수 없도록 분리된 공간에 위치해야 함</li> <li>통신 서버에 접속하기 한 계정의 아이디와 패스워드는 관리자 외 알 수 없도록 관리해야 함</li> </ul>		
네트워크 보안 요구사항	SER-02	<p>정의 : 네트워크 접근 통제, 네트워크 장비의 취약성 및 구성 설정에 대한 보안 요구사항 등 통신을 위해 사용하는 장비 및 접근과 관련한 요구사항</p> <ul style="list-style-type: none"> <li>유선 네트워크를 사용하여 무선 네트워크 사용으로 발생할 수 있는 취약점을 예방함</li> </ul>		

## 9. 품질 요구사항

요구 사항 명칭	고유 번호	요구사항 정의 및 세부내용	산출 정보	관련 요구 사항
장애 복구 대책	QUR-01	<ul style="list-style-type: none"> <li>장애가 발생하지 않도록 시스템을 구축하여야 하며 장애 발생 시 신속하게 조치하여야 한다.</li> <li>시스템의 최적 운용 방안 및 응급처리 방안 등 상세한 장애대책을 제출하여야 한다.</li> <li>단계별 장애를 분류하여 체계적이고 효과적인 백업방안과 복구지침을 마련하여야 한다.</li> </ul>		
웹 호환성	QUR-02	<ul style="list-style-type: none"> <li>특정 브라우저에 적용되는 기술은 가급적 배제하고, 웹브라우저에 대한 호환성이 확보될 수 있도록 설계함</li> </ul>		
정보제공	QUR-03	<ul style="list-style-type: none"> <li>시스템은 온라인 에러 메시지 기능을 제공해야 함</li> <li>시스템은 콘텐츠의 모양이나 배치를 논리적으로 이해하기 쉽게 구성하여 설계·개발해야 함</li> <li>온라인 서식과 관련된 콘텐츠는 입력항목에 대한 설명을 설계시 포함해야 함</li> </ul>		

## 10. 제약 사항

요구 사항 명칭	고유 번호	요구사항 정의 및 세부내용	산출 정보	관련 요구 사항
프레임워크	COR-01	<ul style="list-style-type: none"> <li>JavaScript, Aframe, Node.js 등에서 기본으로 제공되는 라이브러리 사용</li> </ul>		
프로그래밍 언어	COR-02	<ul style="list-style-type: none"> <li>프론트엔드에는 JavaScript, HTML, CSS, Pug, Aframe을 사용함</li> <li>백엔드에는 Node.js, JavaScript를 사용함</li> </ul>		
개발 방법론	COR-03	<ul style="list-style-type: none"> <li>알파 테스트와 애자일 방법론 사용</li> <li>알파 테스트에서 받은 피드백 기반으로 애자일 방법론을 적용함</li> <li>일정 주기로 실시하는 알파 테스트마다 기능들을 추가하며 프로토타입을 제작함</li> </ul>		
업무 모듈화 및 자원 활용 방안	COR-04	<ul style="list-style-type: none"> <li>유연성, 확장성을 확보할 수 있도록 모듈화 개발전략을 반영함</li> <li>현재 보유하여 활용 가능한 H/W, S/W를 최대한 재활용하며, 추가 도입이 필요한 솔루션의 경우 본 용역에 포함하여 수행함</li> </ul>		
시스템 설계	COR-05	<ul style="list-style-type: none"> <li>기존 구축되어 있는 현행 시스템 구조 및 전체 표준화 호환성, 시스템 분산설계, 데이터 유형, 프로세스 환경 유형, 사용자 유형, 시스템 토폴로지가 고려되어 구조 설계가 이루어져야함</li> </ul>		

## 11. 프로젝트 관리 요구사항

요구 사항 명칭	고유 번호	요구사항 정의 및 세부내용	산출 정보	관련 요구 사항
프로젝트 수행 조직	PMR-01	<ul style="list-style-type: none"> <li>프론트엔드 개발, 백엔드 개발은 데이터베이스의 데이터를 이용해 가상공간의 입출력을 처리함</li> <li>데이터베이스 개발 및 관리는 데이터베이스를 개발하고 데이터를 관리함</li> </ul>		
프로젝트 일정 계획	PMR-02	<ul style="list-style-type: none"> <li>주기적으로 알파테스트를 진행하며 개발을 진행함</li> </ul>		
단계별 산출물	PMR-03	<ul style="list-style-type: none"> <li>계획서, 중간보고서, 개발 및 커스터마이징 결과물 확인함</li> </ul>		

[별첨2]

## 중간보고서

1. 요구사항 정의서에 명시된 기능에 대하여 현재까지 분석, 설계, 구현(소스코드 작성) 및 테스트한 내용을 기술하시오.

- 소스 코드

1) vr\_item.pug : VR Scene에서의 UI와 오브젝트의 동작 관리

```
15 script(src='https://rawgit.com/rdub80/aframe-gui/master/dist/aframe-gui.min.js')
16 script(src='https://unpkg.com/aframe-event-set-component@5.0.0/dist/aframe-event-set-component.min.js')
17 script(src='https://unpkg.com/aframe-environment-component@1.3.1/dist/aframe-environment-component.min.js')
18 script(src='https://cdn.rawgit.com/donmccurdy/aframe-extras/v4.1.2/dist/aframe-extras.min.js')
19 script(src='https://unpkg.com/super-hands@3.0.3/dist/super-hands.min.js')
```

그림 1 UI와 오브젝트의 동작 관련 오픈소스

```
69 a-camera(look-controls-enabled='true')
70   a-entity(id="marker" position="0 0 -5" rotation="" scale="" visible="")
71
72   a-entity(id="dropdownContainer" position="0.0 0.7 -1.0" scale='0.15 0.15 0.15')
73     a-gui-flex-container(flex-direction="row" justify-content="flexStart" align-items="normal"
74       component-padding="0.1" opacity="0" width="2.5" height="0.75" position="0 0 0" rotation="0 0 0")
75       a-gui-button.clickable(id="mydropdown" width="2.5" height="0.75" onclick="openDropdown"
76         value="OBJECT" font-family="" font-size="0.25" visible="true" animation_rotateIn="property:
77         rotation; to: 180 0 0; dur: 1000; startEvents:opendropdown" animation_visibleIn="property:
78         visible; to: false; delay: 500; startEvents:opendropdown" animation_rotateOut="property:
79         rotation; to: 0 0 0; dur: 1000; startEvents:closedropdown" animation_visibleOut="property:
80         visible; to: true; delay: 500; startEvents:closedropdown")
81       a-gui-flex-container(id="myoptions" flex-direction="column" justify-content="flexStart"
82         align-items="normal" component-padding="0.1" opacity="0" width="2.5" height="0.75" position="0 0 0"
83         rotation="0 0 0" scale="1 0.001 0.001" visible="false" animation_positionIn="property: position; to:
84         0 0.375 0.15; dur: 1000; startEvents:opendropdown" animation_scaleIn="property: scale; to: 1 1 1;
85         dur: 1000; startEvents:opendropdown" animation_visibleIn="property: visible; to: true; dur: 100;
86         startEvents:opendropdown" animation_positionOut="property: position; to: 0 0 -0.15 dur: 1000;
87         startEvents:closedropdown" animation_scaleOut="property: scale; to: 1 0.001 0.001; dur: 1000;
88         startEvents:closedropdown" animation_visibleOut="property: visible; to: false; dur: 100; delay: 500;
89         startEvents:closedropdown")
90         a-gui-button.clickable(id="box" width="2.5" height="0.75" pressed-gui-button="" font-size="0.3"
91           value="BOX")
92         a-gui-button.clickable(id="sphere" width="2.5" height="0.75" pressed-gui-button="" font-size="0.3"
93           value="SPHERE")
94         a-gui-button.clickable(id="cylinder" width="2.5" height="0.75" pressed-gui-button="" font-size="0.
95         3" value="CYLINDER")
96         a-gui-button.clickable(height="0.5" onclick="closeDropdown" value="x")
```

그림 2 Dropbox가 열렸을 때, BOX, SPHERE, CYLINDER 메뉴를 보여줌

```

49  // - Object Make
50  a-assets
51      a-mixin(id='box' geometry='primitive: box; width: 0.5; height: 0.5; depth: 0.5' hoverable='' grabbable=''
        stretchable='' draggable='' droppable='' shadow='' event-set__dragdrop='_event: drag-drop;'
        event-set__hoveron='_event: hover-start; material.opacity: 0.7; transparent: true'
        event-set__hoveroff='_event: hover-end; material.opacity: 1; transparent: false'
        event-set__dragon='_event: dragover-start; material.wireframe: false' event-set__dragoff='_event:
        dragover-end; material.wireframe: false')
52
53      a-mixin(id='sphere' geometry='primitive: sphere; width: 0.5; height: 0.5; depth: 0.5' hoverable=''
        grabbable='' stretchable='' draggable='' droppable='' shadow='' event-set__dragdrop='_event: drag-drop;'
        event-set__hoveron='_event: hover-start; material.opacity: 0.7; transparent: true'
        event-set__hoveroff='_event: hover-end; material.opacity: 1; transparent: false'
        event-set__dragon='_event: dragover-start; material.wireframe: false' event-set__dragoff='_event:
        dragover-end; material.wireframe: false')
54
55      a-mixin(id='cylinder' geometry='primitive: cylinder; width: 0.5; height: 0.5; depth: 0.5' hoverable=''
        grabbable='' stretchable='' draggable='' droppable='' shadow='' event-set__dragdrop='_event: drag-drop;'
        event-set__hoveron='_event: hover-start; material.opacity: 0.7; transparent: true'
        event-set__hoveroff='_event: hover-end; material.opacity: 1; transparent: false'
        event-set__dragon='_event: dragover-start; material.wireframe: false' event-set__dragoff='_event:
        dragover-end; material.wireframe: false')

```

그림 3 차례대로 box, sphere, cylinder 오브젝트 생성의 뼈대

```

56  a-entity(id='objects')
57      // - 반투명
58      //a-entity(class='box' mixin='box' position='0 1 -1.25' material='color: red')
59      //a-entity(class='sphere' mixin='sphere' position='2 1 -1.25' material='color: blue')

```

그림 4 Dropbox의 메뉴로 오브젝트 생성 시 해당 오브젝트 정보가 저장될 공간

```

62  // - Object Move
63  a-entity(wasd-controls='' position='0 1 1' capture-mouse='' raycaster='objects: .box' cursor='rayOrigin:mouse'
        static-body='shape: sphere; sphereRadius: 0.001' super-hands='colliderEvent: raycaster-intersection;
        \colliderEventProperty: els;\colliderEndEvent:raycaster-intersection-cleared;\colliderEndEventProperty:
        clearedEls;')
64
65  a-entity(wasd-controls='' position='0 1 1' capture-mouse='' raycaster='objects: .sphere'
        cursor='rayOrigin:mouse' static-body='shape: sphere; sphereRadius: 0.001' super-hands='colliderEvent:
        raycaster-intersection;\colliderEventProperty: els;\colliderEndEvent:raycaster-intersection-cleared;
        \colliderEndEventProperty: clearedEls;')
66
67  a-entity(wasd-controls='' position='0 1 1' capture-mouse='' raycaster='objects: .cylinder'
        cursor='rayOrigin:mouse' static-body='shape: sphere; sphereRadius: 0.001' super-hands='colliderEvent:
        raycaster-intersection;\colliderEventProperty: els;\colliderEndEvent:raycaster-intersection-cleared;
        \colliderEndEventProperty: clearedEls;')

```

그림 5 차례대로 box, sphere, cylinder 오브젝트가 마우스로 드래그될 때의 동작 처리

```

67  //test entity
68  if objectlist
69      a-entity(id='boxtest' loc='box1' mixin='box' position=objectlist.pos.x+ ' +objectlist.pos.y+ ' +objectlist.pos.z rotation=objectlist.pos.yaw+ ' +objectlist.pos.pitch+ '
        +objectlist.pos.roll color='red' material='color: '+objectlist.pos.color)

```

그림 6 데이터베이스에서 정보 불러와서 화면에 띄우기



## 2) dropdown-event.js : Dropbox UI 조작 관련

```
1 window.closeDropdown = function() {  
2     console.log("closedropdown");  
3     var mydropdown = document.getElementById("mydropdown");  
4     mydropdown.emit('closedropdown');  
5     var myoptions = document.getElementById("myoptions");  
6     myoptions.emit('closedropdown');  
7  
8 }  
9  
10 window.openDropdown = function() {  
11     console.log("opendropdown");  
12     var mydropdown = document.getElementById("mydropdown");  
13     mydropdown.emit('opendropdown');  
14     var myoptions = document.getElementById("myoptions");  
15     myoptions.emit('opendropdown');  
16  
17 }
```

그림 7 Dropbox를 열 때와 닫을 때의 동작 처리

### 3) pressed-gui-button.js : Dropbox의 각 메뉴 클릭에 따른 동작 (오브젝트 생성) 처리

```
1 AFRAME.registerComponent('pressed-gui-button', {
2
3   init: function() {
4     //var sceneEl = document.querySelector('a-scene');
5     var objectEl = document.querySelector('#objects');
6     var markerEl = document.querySelector('#marker');
7     var boxIndex = 0;
8     var sphereIndex = 0;
9     var cylinderIndex = 0;
10
11     // Add boxe when spacebar is pressed.
12     this.el.addEventListener('click', function (e) {
13       //if (e.keyCode !== 32) return;
14       //console.log(this.id);
15
16       // 각 오브젝트 유형에 따른 설정 지정
17       switch(this.id) {
18         case "box":
19           var newEl = document.createElement('a-box');
20           newEl.setAttribute('color', 'red');
21
22           newEl.setAttribute('class', this.id);
23           newEl.setAttribute('id', this.id + boxIndex);
24           boxIndex++;
25
26           newEl.setAttribute('mixin', this.id);
27
28           break;
29         case "sphere":
30           var newEl = document.createElement('a-sphere');
31           newEl.setAttribute('color', 'blue');
32
33           newEl.setAttribute('class', this.id);
34           newEl.setAttribute('id', this.id + sphereIndex);
35           sphereIndex++;
36
37           newEl.setAttribute('mixin', this.id);
38
39           break;
40         case "cylinder":
41           var newEl = document.createElement('a-cylinder');
42           newEl.setAttribute('color', 'yellow');
43
44           newEl.setAttribute('class', this.id);
45           newEl.setAttribute('id', this.id + cylinderIndex);
46           cylinderIndex++;
47
48           newEl.setAttribute('mixin', this.id);
49
50           break;
51         default: break;
52       }
53
54       // 오브젝트 생성
55       //sceneEl.appendChild(newEl);
56       objectEl.appendChild(newEl);
57
58       // 각 오브젝트가 클릭이 되도록 리스너 달기
59       newEl.classList.add('clickable');
60       newEl.addEventListener('click', handleClickEvent);
61
62       // 오브젝트가 생성되었을 때의 위치 수정
63       var position = markerEl.object3D.getWorldPosition();
64       //position.y = 0.5;
65       newEl.setAttribute('position', position);
66     });
67   }
68 });
69
70 // 해당 오브젝트가 맞는지 확인하는 용
71 function handleClickEvent (event)
72 {
73   var obj = event.target;
74
75   console.log(obj.id);
76   console.dir(obj);
77   // console.log(obj.getAttribute("position"));
78 }
```

그림 10 오브젝트 처리

- 오브젝트의 색 지정 : box 오브젝트는 빨간색, sphere 오브젝트는 파란색, cylinder 오브젝트는 노란색
- 오브젝트의 ID 지정 : 오브젝트 유형 + 생성된 순서 (예: box1, sphere4 등)
- 각 오브젝트가 마우스로 드래그될 때 이동처리 및 클릭 리스너 달기
- 오브젝트 생성 및 생성되었을 때의 위치 수정

#### 4) vrModel.js : mongoDB schema

```
1 var mongoose = require('mongoose');
2 var Schema = mongoose.Schema;
3
4 var vrSchema = new Schema({
5   region_id: {type:mongoose.Schema.ObjectId},
6   scene_name:String,
7   image_file: String,
8   left_name: String,
9   up_name: String,
10  right_name: String,
11  down_name: String,
12  links:{left: {name:String, x:Number, y:Number, z:Number, yaw:Number, pitch:Number, roll:Number},
13           up:   {name:String, x:Number, y:Number, z:Number, yaw:Number, pitch:Number, roll:Number},
14           right:{name:String, x:Number, y:Number, z:Number, yaw:Number, pitch:Number, roll:Number},
15           down: {name:String, x:Number, y:Number, z:Number, yaw:Number, pitch:Number, roll:Number},
16           default: false},
17  boxtest:{pos: {x:Number, y:Number, z:Number, yaw:Number, pitch:Number, roll:Number, id:String, loc:String, mixin:String, color:String},
18            default: false},
19
20 });
21
22 module.exports = mongoose.model('vrItem', vrSchema);
```

그림 9 MongoDB의 기본적인 schema 정의

#### 5) vr.js : mongoDB로 저장

```
136 if("boxtest" in req.body)
137 {
138   console.log('boxtest');
139   item.boxtest.pos.x = req.body.boxtest[0]['x']; //position
140   item.boxtest.pos.y = req.body.boxtest[0]['y'];
141   item.boxtest.pos.z = req.body.boxtest[0]['z'];
142
143   item.boxtest.pos.yaw = req.body.boxtest[1]['x']; //rotation
144   item.boxtest.pos.pitch = req.body.boxtest[1]['y'];
145   item.boxtest.pos.roll = req.body.boxtest[1]['z'];
146
147   item.boxtest.pos.id = req.body.boxtest[2]; //id
148   item.boxtest.pos.loc = req.body.boxtest[3]; //loc
149   item.boxtest.pos.mixin = req.body.boxtest[4]; //mixin
150   item.boxtest.pos.color = req.body.boxtest[5]; //color
```

그림 10 MongoDB에 요소별 value 저장

- 구현 결과

1) Dropbox UI 동작

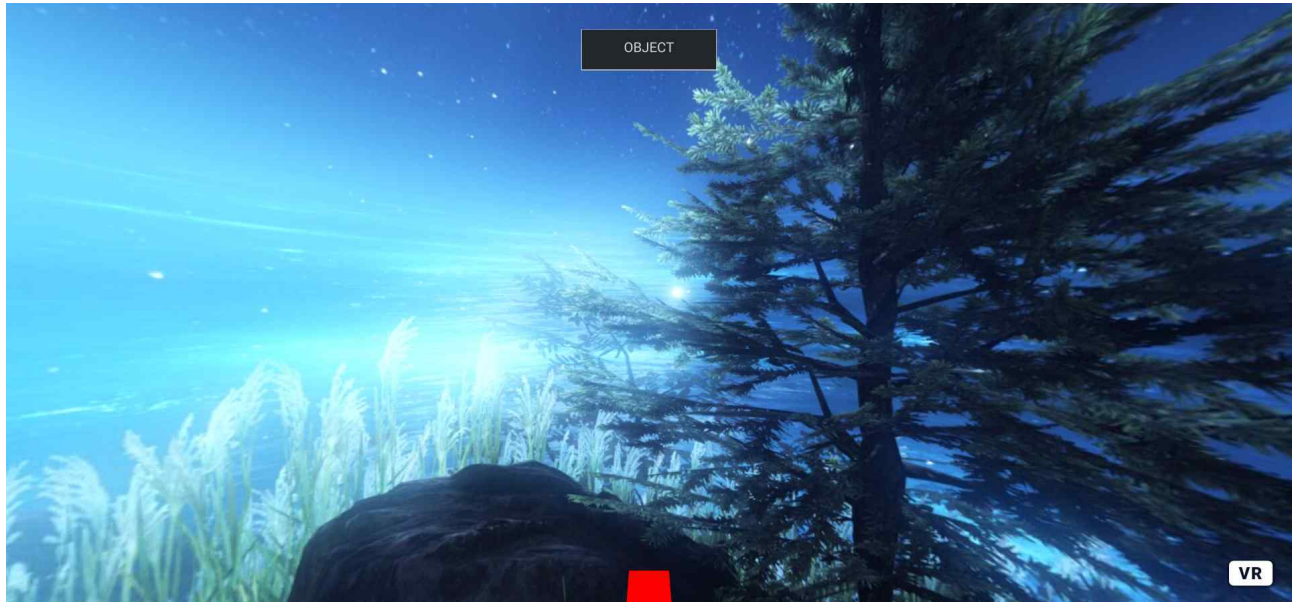


그림 11 Dropbox 닫힘 상태

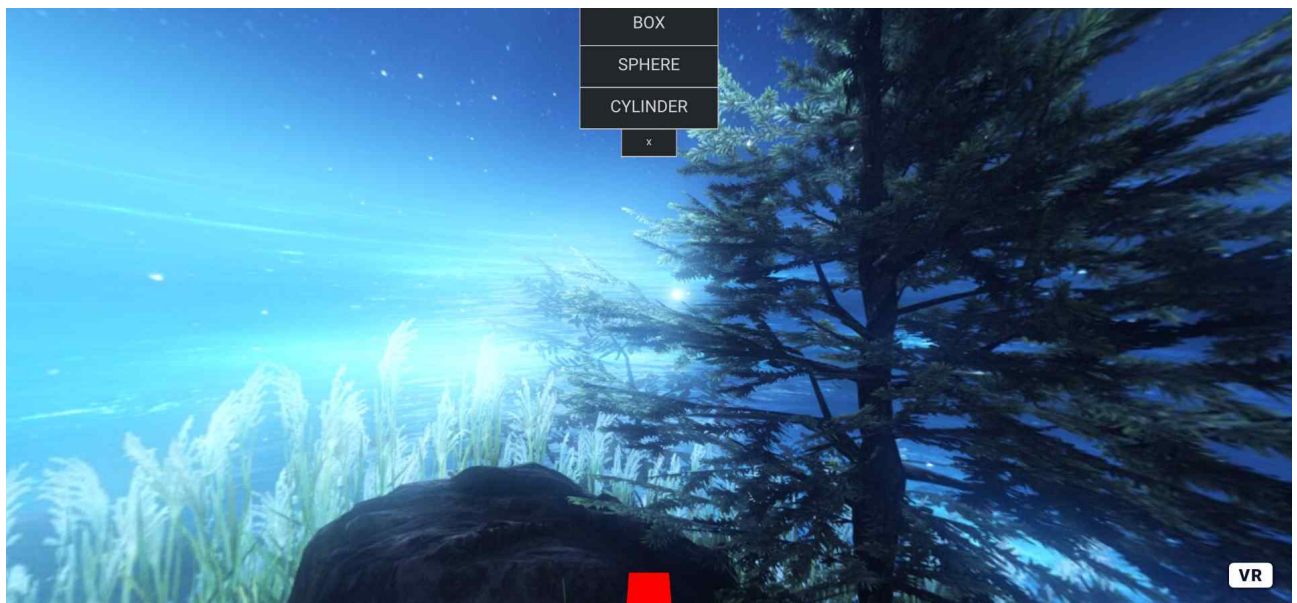


그림 12 Dropbox 열림 상태



## 2) Dropbox의 메뉴 아이템으로 오브젝트 (box, sphere, cylinder) 생성

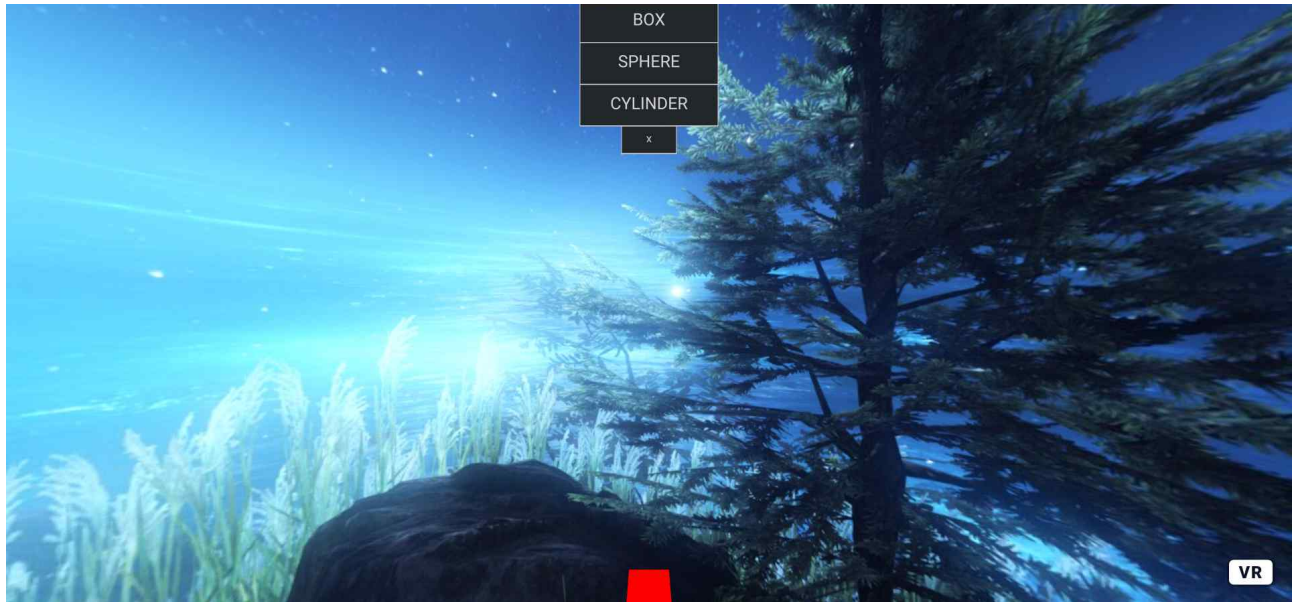


그림 13 오브젝트 생성 전

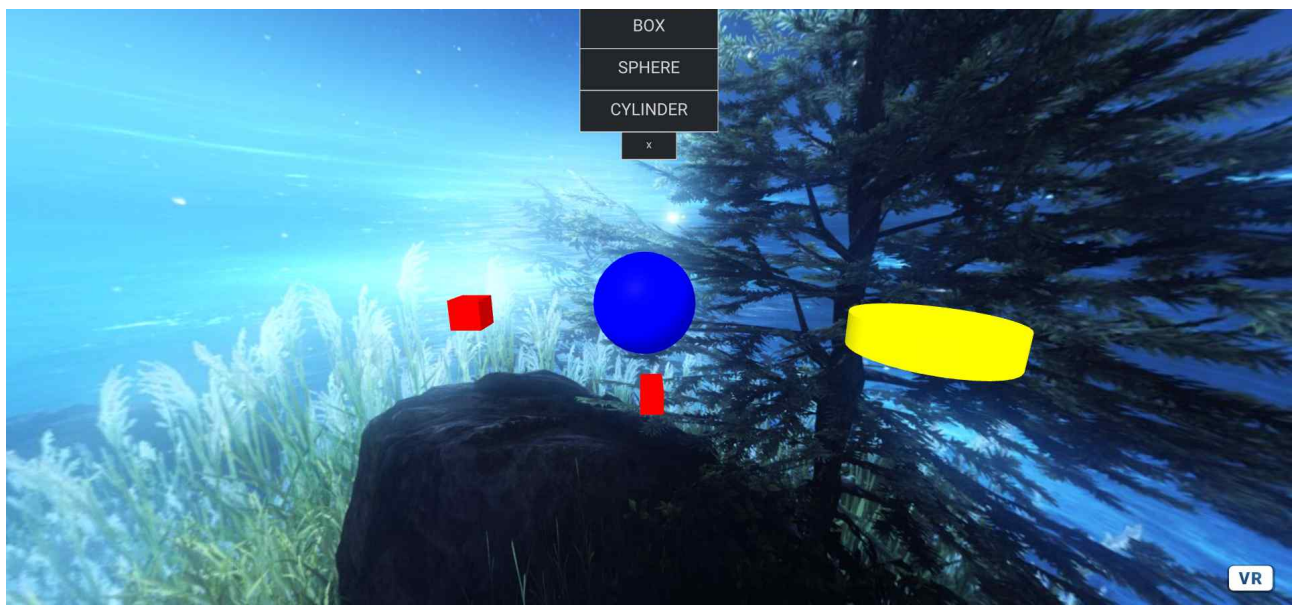


그림 14 box, sphere, cylinder 오브젝트 생성 후

### 3) 오브젝트 이동

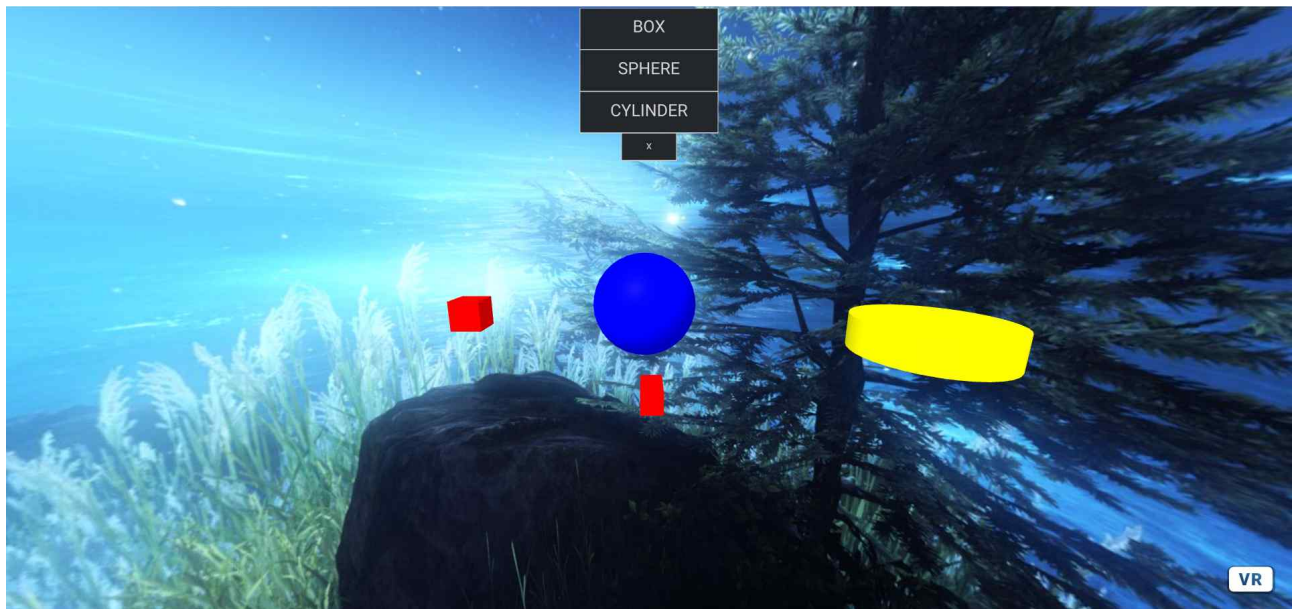


그림 15 드래그로 오브젝트 이동 전

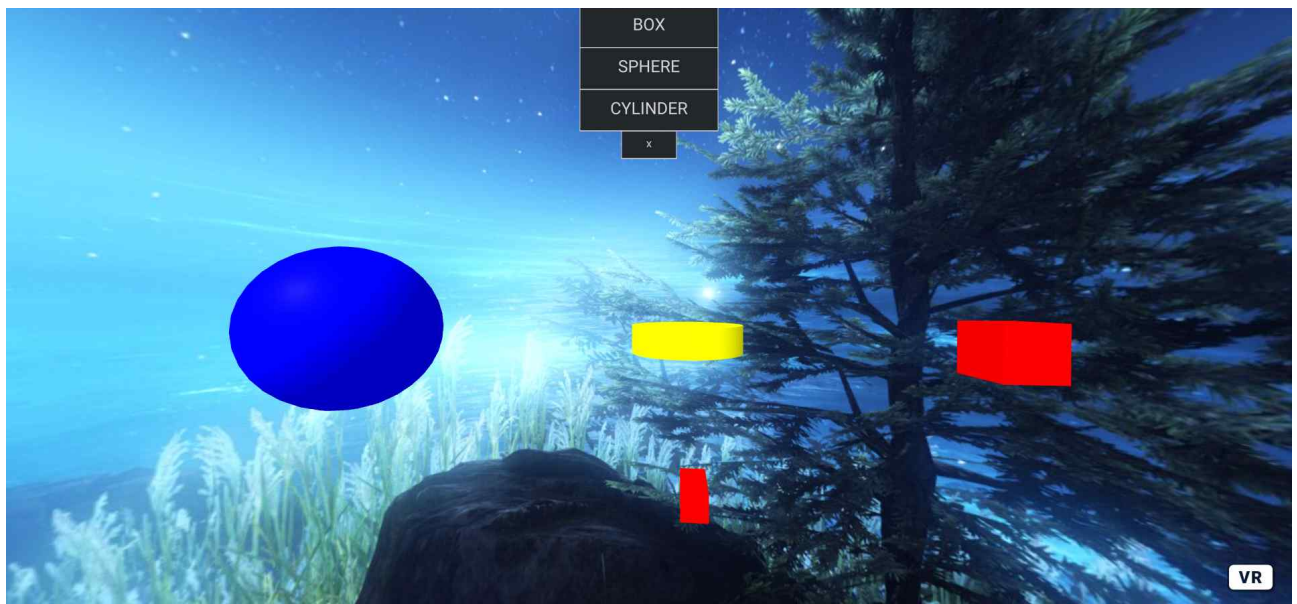


그림 16 드래그로 오브젝트 이동 후



#### 4) MongoDB에 데이터 저장

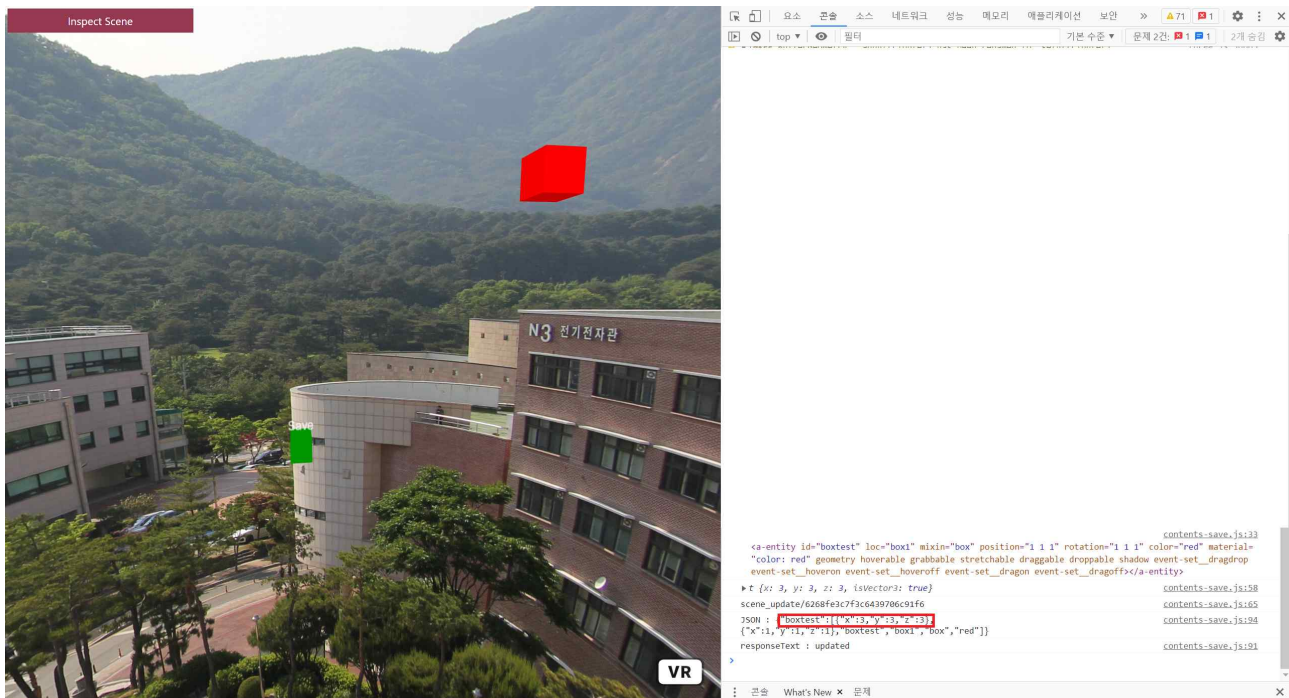


그림 19 웹 페이지에서 object의 위치를 변경한 모습(새로고침 후에도 반영됨)

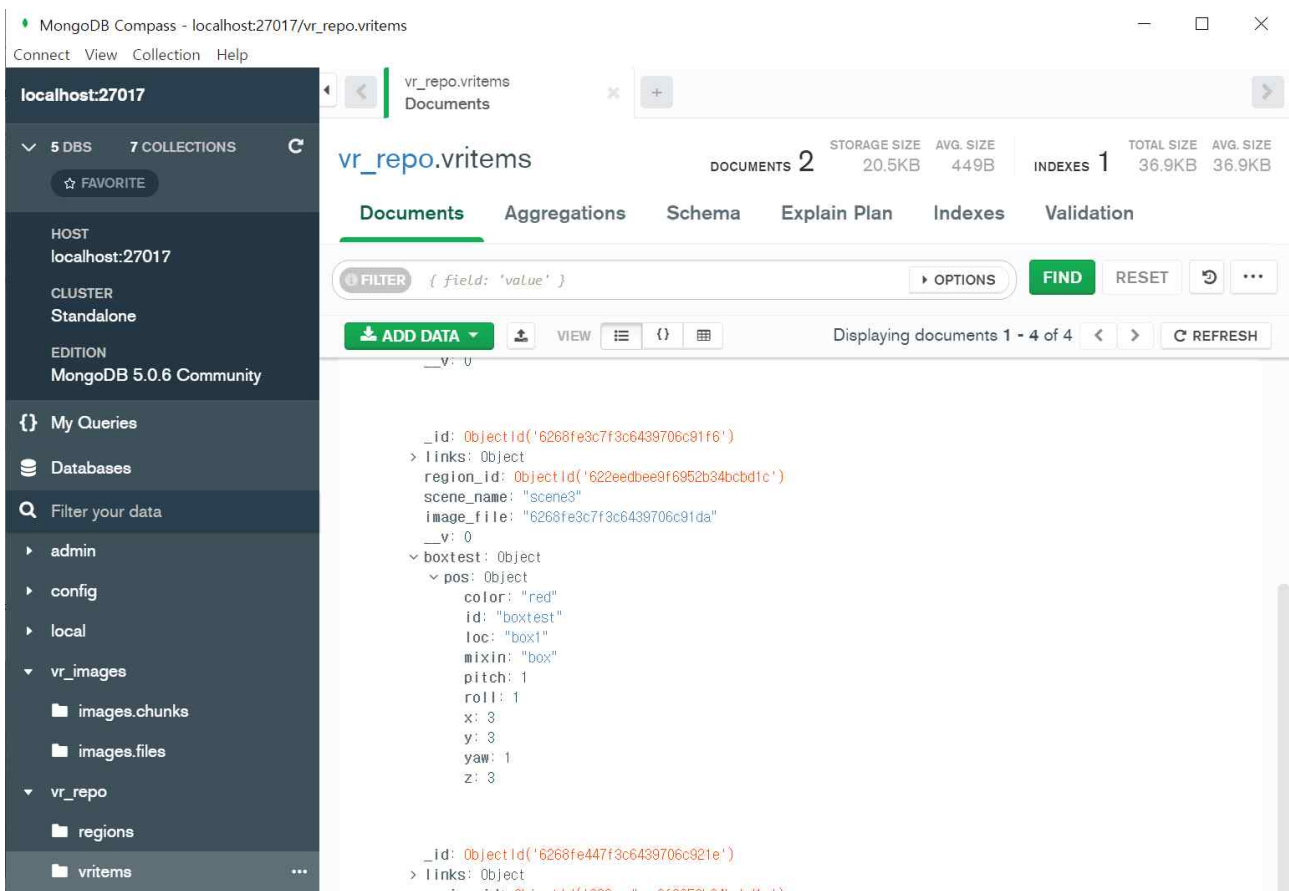


그림 20 MongoDB에 위치가 반영된 모습

2. 프로젝트 수행을 위해 적용된 추진전략, 수행 방법의 결과를 작성하고, 만일 적용과정에서 문제점이 도출되었다면 그 문제를 분석하고 해결방안을 기술하시오.

### 1) 기능 구현 과정

Dropbox 메뉴를 통해 오브젝트 생성을 한 후, 제대로 생성되었는지 확인과 드래그로 오브젝트를 이동하기 위해서는 클릭 이벤트 리스너(Click event listener)를 등록해야 했습니다. addEventListener를 통해 클릭 이벤트 시 handleClickEvent 함수를 등록했는데, 마우스로 클릭해봐도 지정된 리스너 함수가 제대로 동작하지 않는 문제가 발생했습니다. F12 개발자 모드로 확인해 본 결과 오브젝트 자체는 잘 생성되었지만, 오브젝트에 클릭 이벤트 리스너 등록이 안되었다는 것을 알게 되었습니다. 팀원과 의논해보고, Google 통해 검색을 해 보면서 클릭 리스너가 제대로 동작하기 위해서는 클래스에 clickable이 있어야 한다는 것을 깨달았습니다. classList.add('clickable')를 추가하여 해당 기능 구현을 마무리 지을 수 있었습니다.

### 2) 팀원의 책임 및 역할 분배

프로젝트를 성공적으로 수행하기 위해서 팀원 각자 자신이 수행할 수 있는 부분을 나누어서 분배했습니다. 프로젝트에서 크게 백엔드, 데이터베이스, 테스트 관리 3가지가 가장 중요한 요소라고 생각하여 각 파트별로 책임자를 나누었습니다. 최진아 학생은 프로젝트 팀장으로서 백엔드를 개발하고, 이혜진 학생은 데이터베이스 개발 및 관리를 맡고, 유선아 학생은 백엔드 개발과 애플리케이션 테스트 관리의 역할을 맡았습니다. 하지만 적은 인원으로 모든 영역을 개발할 때, 각자가 맡은 역할만 수행하는 것은 프로젝트 진행 속도에 무리가 있다고 판단하였습니다. 따라서 한 사람씩 맡은 파트를 책임자로 정하고, 책임자 중심으로 다른 학생들이 개발을 지원하는 방식으로 프로젝트를 운영했습니다. 예를 들어, 최진아 학생이 백엔드 개발의 책임자가 되어 백엔드 개발에서 문제가 발생하거나 개발이 지연되는 경우 이혜진, 유선아 학생이 백엔드 개발에 함께 참여하여 개발의 속도와 질을 높였습니다. 각 파트마다 책임자가 존재하기 때문에 의견 충돌이 있거나 문제가 발생한 경우 책임자를 중심으로 의사 결정을 할 수 있었고, 프로젝트를 수행하면서 팀원들의 책임과 역할을 명확하게 분배하여 프로젝트를 수행하고 있습니다.

### 3) 현재까지의 프로젝트 수행 내용 및 앞으로의 계획

프로젝트를 일정은 아래 표와 같습니다.

	1월	2월	3월	4월	5월	6월
프로젝트 계획 수립						
프로젝트 준비						
프로젝트 수행(개발)						
프로토타입에 대한 연구 논문 작성						
알파 테스트 및 유지보수						
프로젝트 마무리						

1월에 프로젝트 계획 수립을 시작으로 프로젝트 준비 과정에서는 팀원들과 매일 만나 프로젝트 수행을 위해 필요한 언어(HTML, CSS, JavaScript) 학습, 프로토타입 분석 등 추가적



인 학습을 진행했습니다. 2월 말부터 프로젝트 수행을 위해 기능적 요구사항을 세우고 실질적인 개발을 시작했습니다. 동시에 프로토타입을 가지고 2차례의 알파 테스트(썸뿔여자고등학교, 유성여자고등학교)를 진행하면서 기능적 요구사항에 대한 추가적인 의견을 수렴하고, 이 과정을 연구 논문으로 작성하였습니다. 작성한 연구 논문은 'FRONTIERS IN EDUCATION 2022'에 'Non-face-to-face Career Exploration Program utilizing Web-based Metaverse Hands-on Contents'의 제목으로 제출하였습니다. 연구 논문은 비대면 상황에서 진로 탐색 프로그램을 운영하기 위하여 웹 기반의 메타버스(Web-based Metaverse)를 활용하여 학생들의 공학 계열로의 흥미를 증가시키고 실습 중심의 학습이 학습의 성취도가 향상된 점을 중심으로 작성하였습니다. 5월 중 Paper Revision Requirements에 대한 공지와 7월 중 Peer Review를 통해 논문 게재 여부가 발표됩니다. 연구 논문과 함께 기능적 요구사항을 만족시키기 위한 기능 중심의 프로젝트 개발도 수행하고 있습니다.

프로젝트 개발이 끝난 후에는 2차례의 알파 테스트를 통해 요구사항을 만족하는지 검토하고 테스트 과정을 문서화하여 유지보수 계획을 세울 예정입니다. 첫 번째 알파 테스트는 3주간 우리 학과 1학년 학생들을 대상으로 공학설계입문 수업에서 진행하고, 두 번째 알파 테스트는 유성여자고등학교 1학년 학생 20명을 대상으로 진행됩니다.

## 캡스톤 디자인 | 중간보고서 채점표

평가도구	평 가 항 목	평 가 점 수				
		1	2	3	4	5
중간 보고서 및 실행 결과	1. 요구사항 정의서(기능, 성능, 인터페이스 등)가 구체적으로 작성되었는가?					
	2. 요구분석, 설계 산출물(모델, 프로토타입 등)의 내용이 충실한가?					
	3. 설계 및 구현 문제를 위해 적용한 이론, 문제해결 방법이 제시되었으며 그 적용이 적합한가?					
	4. 구현된 소프트웨어(또는 이와 동등한 하드웨어 시스템)가 버그없이 실행되었는가?					
	5. 구현된 소프트웨어(또는 이와 동등한 하드웨어 시스템)의 성능 요구사항은 충족되었는가?					
도구활용	6. 설계 및 구현을 위해 도구가 적절히 활용되었는가?					
	7. 도구의 활용수준(능숙도)은 프로젝트 수행에 적합한가?					
팀원의 업무 및 역할	8. 팀원의 업무분담에 따른 역할 및 협력이 충실히 이루어졌는가? (평가자에 의한 질의)					
	9. 프로젝트 중간 진척상황에 대해 팀원이 충분히 인지하고 있는가?(평가자에 의한 질의)					
합계						
*검토 의견(최종완료 때까지 보완해야할 점에 대해 작성 요망) <div style="height: 150px; border: 1px solid black; margin-top: 10px;"></div>						
심사위원(소속):		(이름)			(인)	