

Campus Tour w. Meta FG

Software Design Specification



Team 8

Leader 권 준 희
Member 박 수 현
Member 한 정 우
Member 김 장 훈
Member 김 태 현
Member 김 휘 원
Member 박 성 원

목 차

1. Preface	8
1.1 Readership	8
1.2 Scope	8
1.3 Objective	8
1.4 Document Structure	8
2. Introduction	9
2.1 Objectives	9
2.2 Applied Diagram	9
2.2.1 UML	10
2.2.2 Use Case Diagram	10
2.2.3 Process Diagram	10
2.2.4 Sequence Diagram	10
2.2.5 Class Diagram	11
2.3 Applied Tools	11
2.3.1 Microsoft PowerPoint	11
2.4 Project Scope	11
2.5 References	11
3. System Architecture – Overall	12
3.1 Objectives	12
3.2 System Organization	12
3.2.1 Use Case diagram	13
3.2.2 Process diagram	13
3.2.2.1 Process diagram of 사용자 관리	13
3.2.2.2 Process diagram of 캠퍼스 투어	14
4. System Architecture – Modelling	14
4.1 Objectives	15
4.2 Subcomponents	15
4.2.1 전체 캠퍼스	15

4.2.2 삼성 학술 정보관	16
4.2.3 반도체관	16
4.2.5 제2 공학관	17
4.2.6 산학협력 센터	18
4.2.7 캐릭터	19
4.2.7.1 FG 유저	20
4.2.7.2 새내기 유저	21
5. System Architecture – Backend	21
5.1 Objectives	22
5.2 Overall Architecture	22
5.3 Subcomponents	22
5.3.1 포탈	22
5.3.1.1 Class diagram	23
5.3.1.2 Sequence diagram	23
5.3.2 텔레포트	24
5.3.2.1 Class diagram	24
5.3.2.2 Sequence diagram	25
5.3.3 전광판	26
5.3.3.1 Class diagram	26
5.3.3.2 Sequence Diagram	27
5.3.4 미니맵	28
5.3.4.1 Class diagram	28
5.3.4.2 Sequence diagram	29
6. Testing Plan	30
6.1 Objectives	30
6.2 Testing Policy	30
6.2.1 개발 테스트(Development Testing)	30
6.2.1.1 Performance	30
6.2.1.2 Reliability	31
6.2.2 배포 테스트 (Release Testing)	31
6.2.3 사용자 테스트 (User Testing)	32
6.2.4 Testing Case	32

7. Development Environment	32
7.1 Objectives	32
7.2 Frontend Environment	32
7.2.1 Unity	32
7.2.2 VRChat SDK3	33
7.3 Backend Environment	34
7.2.3 Udon	34
7.2.3 Github	34
7.4 Constraints	35
7.4.1 Memory Constraints	35
7.4.2 Design Constraints	36
7.5 Assumptions and Dependencies	36
8. Supporting Information	36
8.1 Software Design Specification	36
8.2 Document History	36

List of Figures

- [Figure 1] Overall System Organization (12)
- [Figure 2] Use Case Diagram (13)
- [Figure 3] Process Diagram of 사용자 관리 (14)
- [Figure 4] Process diagram of 캠퍼스 투어 (14)
- [Figure 5] 전체 캠퍼스 평면도 (15)
- [Figure 6] 삼성 학술 정보관 평면도 (16)
- [Figure 7] 반도체관 평면도 (17)
- [Figure 8] 제2 공학관 평면도 (18)
- [Figure 9] 산학협력 센터 평면도 (19)
- [Figure 10] 기본 캐릭터 예시 (20)
- [Figure 11] FG 캐릭터 식별 표시 예시 (20)
- [Figure 12] FG 캐릭터 식별 표시 예시 (21)
- [Figure 13] Class Diagram of 포탈 (23)
- [Figure 14] Sequence Diagram of 포탈 (24)
- [Figure 15] Class Diagram of 텔레포트 (25)
- [Figure 16] Sequence Diagram of 텔레포트 (26)
- [Figure 17] Class Diagram of 전광판 (27)

[Figure 18] Sequence Diagram of 전광판 (28)

[Figure 19] Class Diagram of 미니맵 (29)

[Figure 20] Sequence Diagram of 미니맵 (30)

[Figure 21] Software Release Lifecycle (31)

[Figure 22] Unity (33)

[Figure 23] VRChat (33)

[Figure 24] Udon (34)

[Figure 25] Github (35)

List of Tables

- [Table 1] 전체 캠퍼스 정보 (15)
- [Table 2] 삼성 학술 정보관 건물 정보 (16)
- [Table 3] 반도체관 건물 정보 (17)
- [Table 4] 제2 공학관 건물 정보 (18)
- [Table 5] 산학협력 센터 건물 정보 (19)
- [Table 6] FG 캐릭터 (21)
- [Table 7] 새내기 유저 (21)
- [Table 8] NPC 캐릭터 (21)
- [Table 9] Document History (37)

1. Preface

이 문서는 성균관대학교 Freshman Guide(이하 FG) 단체가 진행하던, 신입생들을 대상으로 하는 캠퍼스 투어를 메타버스 환경으로 구현하는 “Campus Tour w. Meta FG”에 대한 Design Specification을 기술하기 위해 작성된 디자인 명세서입니다.

1.1 Readership

이 문서는 크게 10개의 Chapter 와 Chapter 내 세부적인 subsections 으로 작성되었습니다. 이 문서의 주요 독자는 2021년 2학기 성균관대학교 소프트웨어공학 수업의 Team8 입니다. 또한 소프트웨어공학 수업의 교수님, TA, 다른 Team 의 학생들 역시 이 문서의 주요 독자가 될 수 있습니다. 그 외 개발자, 테스터 등 모든 이해관계자도 독자층에 포함됩니다.

1.2 Scope

문서의 10개 Chapter 는 어플리케이션의 소프트웨어 엔지니어링을 위해 필요한 시스템 아키텍처 등 시스템 디자인 설계 및 구현 관련 전반적인 내용을 포함합니다.

1.3 Objective

이 문서의 주요 목적은 “Campus Tour w. Meta FG” 어플리케이션 개발과 실행에 필요한 소프트웨어 아키텍처 디자인 구조 및 아키텍처 설계 관련 정보를 제공하는 것입니다. 문서의 내용은 객체 지향 소프트웨어 디자인 및 개발을 전제로 합니다. 전체적인 시스템 아키텍처 구조의 주요 서브 컴포넌트 구성요소들을 여러 UML 다이어그램 기반으로 설명합니다. 시스템 아키텍처와 더불어 테스트, 개발 계획에 대한 정보를 전달하는 것이 문서의 목적입니다.

1.4 Document Structure

1. Preface: 문서의 예상 독자층, 범위 정의와 더불어 전체적 목표, 문서 구조를 다룹니다.
2. Introduction: 시스템 디자인에 활용할 다이어그램, 도구 및 시스템 목표를 설명합니다.
3. Overall System Architecture: 시스템의 전체적인 아키텍처 구조를 use case, process 다이어그램으로 표현합니다.
4. System Architecture – Modelling: 모델링에 관한 아키텍처 구조를 설명합니다
5. System Architecture – Backend: 시스템의 각 주요 기능에 대한 아키텍처 구조를 class diagram, sequence diagram 으로 표현합니다.
6. Testing Plan: 시스템 테스트 계획을 설명합니다.
7. Development Plan: 시스템 개발에 활용할 개발 도구 및 개발 관련 이슈를 설명합니다.
8. Supporting Information: 문서 작성 기준 및 문서 작성 기록 정보입니다.

2. Introduction

이 프로젝트의 목적은 성균관대학교 Freshman Guide(이하 FG) 단체가 진행하는 신입생 대상 캠퍼스 투어를 메타버스 환경을 통하여 비대면으로 진행할 수 있는 VRChat 기반 플랫폼을 설계하고 개발하는 것이다. 이 플랫폼을 통해 성균관대학교 FG는 다수의 신입생 및 재학생들을 대상으로 메타버스 공간 내에서 성균관대학교 캠퍼스 소개를 진행할 수 있고 성균관대학교 신입생 및 재학생들은 캠퍼스 투어 및 학우들과 소통을 할 수 있다. 시스템은 사용자의 요구사항에 맞추어 성균관대 캠퍼스 World 구현, 아바타 이동 기능, 보이스 채팅 기능, 캠퍼스 소개를 위한 멀티미디어 기능 등을 제공한다. 이 설계 문서는 프로젝트 구현에 사용되거나 사용되도록 의도된 설계를 제시하며 설명된 설계는 프로젝트를 위해 미리 작성된 소프트웨어 요구 사항 문서에 지정된 요구 사항을 따른다.

2.1 Objectives

이 장에서는 설계 단계에서 이 프로젝트에 적용된 다양한 도구 및 도표에 관한 설명을 진행한다.

2.2 Applied Diagram

2.2.1 UML

UML은 Unified Modeling Language의 약어로, 시스템 개발 과정에서 개발자 간의 원활한 의사소통을 위해 표준화한 모델링 언어다. 소프트웨어 개발자가 비즈니스 모델링 및 기타 비소프트웨어 시스템뿐만 아니라 소프트웨어 시스템의 아티팩트들을 지정, 시각화, 구성 및 문서화를 용이하게 하기 위해 개발되었다. UML은 현재 가장 널리 사용되는 모델링 기법으로 객체 지향 소프트웨어와 소프트웨어 개발 프로세스에서 매우 중요한 부분을 차지한다. UML은 주로 소프트웨어 프로젝트 설계를 표현하는데 있어 그래픽 표기법을 사용하며 이를 통해 프로젝트 진행에 있어 아키텍처 설계를 탐색, 검증하고 팀원들 간의 의사소통을 원활히 하는데 있어서 매우 큰 도움이 된다.

2.2.2 Use Case Diagram

시스템과 사용자의 상호작용을 다이어그램으로 표현한 것으로 사용자의 관점에서 시스템의 서비스, 기능 및 그와 관련된 외부 요소를 보여주는 것이다. 사용자가 시스템 내부에 있는 기능 중에 어떤 기능을 사용할 수 있는지 나타내며 Use Case Diagram을 통해 고객과 개발자가 요구사항에 대한 의견을 조율할 수 있다.

2.2.3 Process Diagram

Flow Diagram이라고도 불리며 프로세스 내의 활동 순서를 모델링하는데 사용된다. 세부적으로는, 프로세스의 활동 순서뿐만 아니라 프로세스 참가자, 프로세스 중에 교환되는 정보 및 트리거 이벤트 또한 나타낸다. 또한 일련의 활동 내에 존재하는 다양한 점검, 선택 및 조정을 상세하게 설명할 수 있다.

2.2.4 Sequence Diagram

Sequence Diagram은 행위자가 어떠한 순서로 어떤 객체들과 어떻게 상호작용하는지를 표현하는 다이어그램으로 현재의 시스템이 어떠한 시나리오를 따라 동작하는지를 나타내는데 장점을 가지고 있다.

2.2.5 Class Diagram

시스템의 클래스, 클래스의 속성, 동작 방식, 객체 간 관계를 표시함으로써 시스템의 구조를 기술하는 정적 구조 다이어그램의 일종으로 객체 지향 언어를 직접 매피ング할 수 있는 유일한 UML 다이어그램이기 때문에 객체 지향 시스템 모델링에 널리 사용된다.

2.3 Applied Tools

2.3.1 Microsoft PowerPoint

텍스트 및 그림 도면을 지원하는 툴로, 다양한 형태의 도형을 그릴 수 있어 다이어그램을 그리는데 매우 편리하다. 또한, 기본적으로 문서 작업 도구이기 때문에 그래픽적 요소인 도형과 텍스트를 함께 작업하기가 용이하여 다이어그램 작업에 큰 도움이 된다.

2.4 Project Scope

이 프로젝트는 COVID-19로 인해 새내기들을 상대로 기존의 캠퍼스 투어를 진행하던 FG와 입학 후 캠퍼스를 직접 방문해보지 못한 신입생들 및 재학생들을 대상으로 메타버스 공간 내에서 가상 캠퍼스 투어 서비스를 제공할 것이다. 성균관대학교 자연과학 캠퍼스의 주요 건물들을 모델링한 가상 월드를 제공할 것이며 더욱 더 사실감 있는 투어를 위해 멀티미디어 기능을 통해 사진, 영상 및 음성 재생을 통해 생생한 투어가 가능하도록 서비스를 제공할 예정이다. 또한 음성 채팅 기능을 통해 다른 학우들과 소통할 수 있는 기회를 제공한다.

2.5 References

- 2020 SKKUSE Team 1, 'https://github.com/skkuse/2020spring_41class_team1'
- Ian Sommerville, 'Software Engineering 10th Edition'
- IEEE STANDARD 1016: Software Design Specification

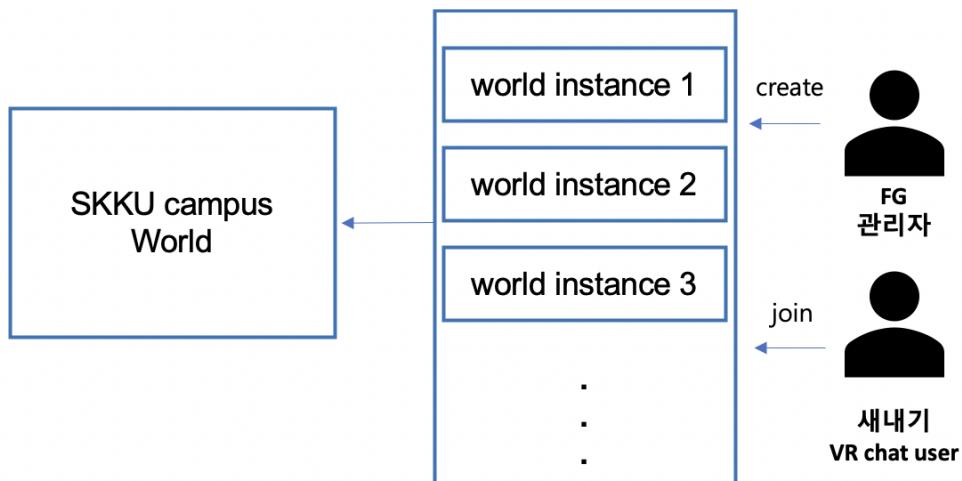
3. System Architecture - Overall

3.1 Objectives

전체적인 시스템 구조를 usecase diagram, process diagram 으로 나타낸다.

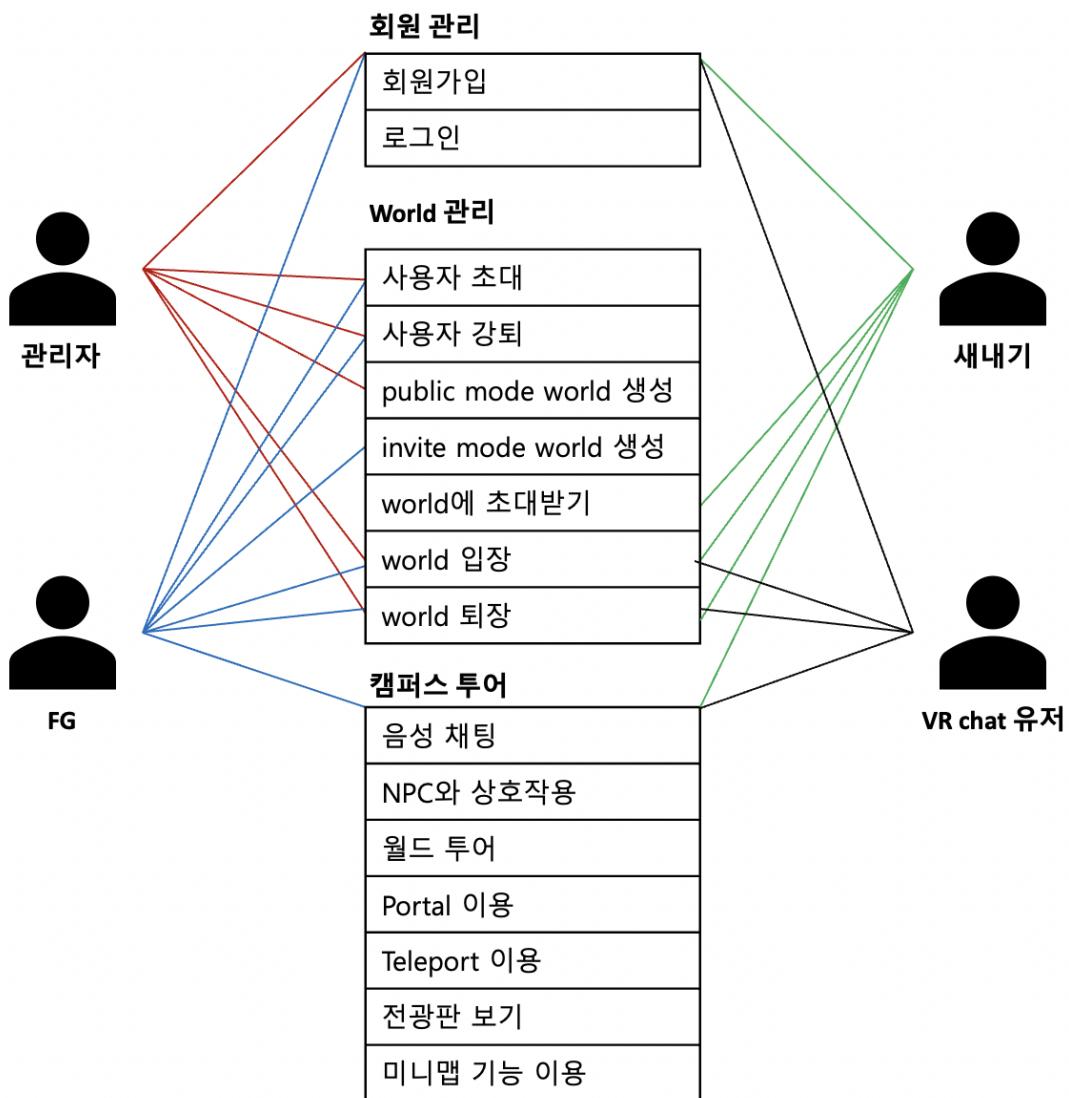
3.2 System Organization

본 Campus Tour w. Meta FG 시스템은 VRchat 환경을 기반으로 구현되기 때문에 VRchat에서 제공되는 다수의 기능 (회원 관리 기능, 음성 채팅 기능 등)을 활용한다. 따라서 별도의 데이터베이스는 사용하지 않으며, VRchat 서버에서 시스템의 데이터를 처리한다. world 인스턴스는 여러개 생성될 수 있으며, 각 인스턴스에서 개별적으로 투어가 진행된다. 사용자는 생성된 world 인스턴스에 접속한 후 SKKU world에서 캠퍼스 투어를 진행할 수 있으며, 음성 채팅, NPC 기능, 미니맵 기능, portal 및 teleport 기능, 전광판 기능 등을 이용할 수 있다.



[Figure 1] Overall System Organization

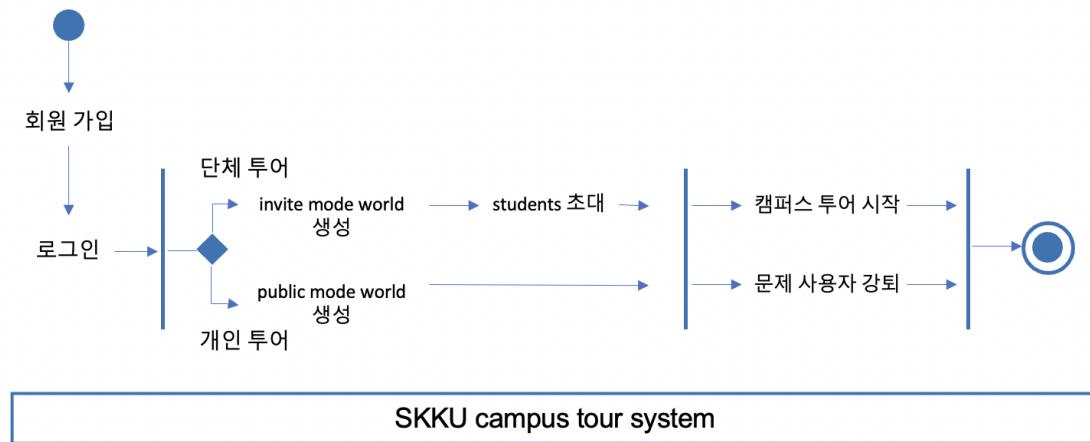
3.2.1 Use Case diagram



[Figure 2] Use Case Diagram

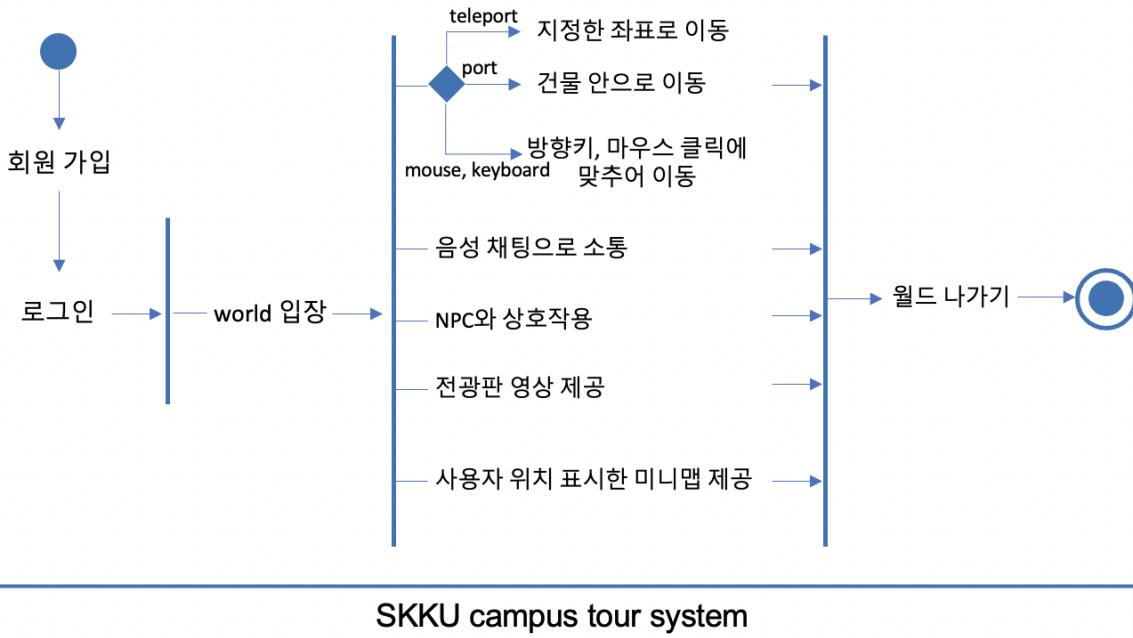
3.2.2 Process diagram

3.2.2.1 Process diagram of 사용자 관리



[Figure 3] Process Diagram of 사용자 관리

3.2.2.2 Process diagram of 캠퍼스 투어



[Figure 4] Process diagram of 캠퍼스 투어

4. System Architecture – Modelling

4.1 Objectives

Chapter 4는 프로그램 상의 건물 및 주변 환경 모델링을 위한 여러 컴포넌트들에 대한 디자인 세부 사항들에 대해 기술합니다.

4.2 Subcomponents

4.2.1 전체 캠퍼스



[Figure 5] 전체 캠퍼스 평면도

명칭	전체 캠퍼스
둘레	2,376.79 m
면적	363,011.9 제곱미터
입장 가능	-

[Table 1] 전체 캠퍼스 정보

4.2.2 삼성 학술 정보관



[Figure 6] 삼성 학술 정보관 평면도

명칭(건물 번호)	삼성 학술 정보관 (48)
둘레	429.13 m
가로	131.21 m
세로	86.2 m
높이	42.89 m
면적	$10,975.28 m^2$
입장 가능	가능

[Table 2] 삼성 학술 정보관 건물 정보

4.2.3 반도체관



[Figure 7] 반도체관 평면도

명칭(건물 번호)	반도체관 (40)
둘레	321.96 m
가로	133.9 m
세로	27.08 m
높이	51 m
면적	$3,626.012 m^2$
입장 가능	불가능

[Table 3] 반도체관 건물 정보

4.2.5 제2 공학관



[Figure 8] 제2 공학관 평면도

명칭(건물 번호)	제2 공학관 (25, 26, 27)
둘레	732 m
가로	116.54 m
세로	118.47 m
높이	40 m
면적	$9,936.26 m^2$
입장 가능	불가능

[Table 4] 제2 공학관 건물 정보

4.2.6 산학협력 센터

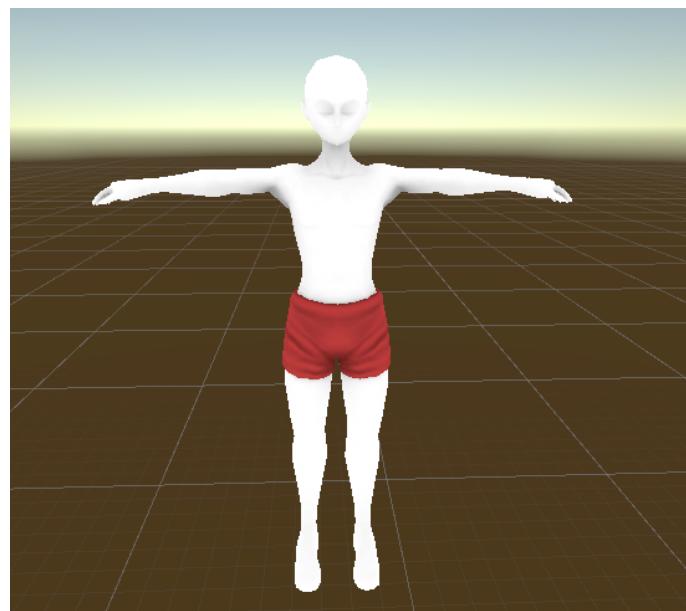


[Figure 9] 산학협력 센터 평면도

명칭(건물 번호)	산학협력 센터 (85)
둘레	393.16 m
가로	78.81 m
세로	92.92 m
높이	40 m
면적	$5,179.28 m^2$
입장 가능	불가능

[Table 5] 산학협력 센터 건물 정보

4.2.7 캐릭터



[Figure 10] 기본 캐릭터 예시

4.2.7.1 FG 유저



[Figure 11] FG 캐릭터 식별 표시 예시

명칭	FG(Freshman Guide) 유저
역할	새내기 유저 통솔 음성 채팅 기능을 이용해 자유롭게 설명
역할 수행 시 식별자	“FG” 문구가 적혀 있는 연초록 상의

[Table 6] FG 캐릭터

4.2.7.2 새내기 유저

명칭	새내기(Freshman) 유저
역할	FG의 초대에 승낙해 FG를 뒤따르며 설명 경청 자유롭게 필드 이동 가능 NPC를 클릭해 설명 경청 가능 입장이 가능한 건물에 입장 해 자유로운 소통 및 활동 가능

[Table 7] 새내기 유저

4.2.8.3 NPC



[Figure 12] FG 캐릭터 식별 표시 예시

명칭	NPC 캐릭터
역할	유저가 아닌 캐릭터 클릭 시, 해당 역할에 맞는 설명 음성 진행
역할 수행 시 식별자	캐릭터 위에 떠있는 전구 아이콘

[Table 8] NPC 캐릭터

5. System Architecture – Backend

5.1 Objectives

이 챕터에서는 포탈, 텔레포트, 미니맵, 전광판 시스템에 대한 구조를 설명한다.

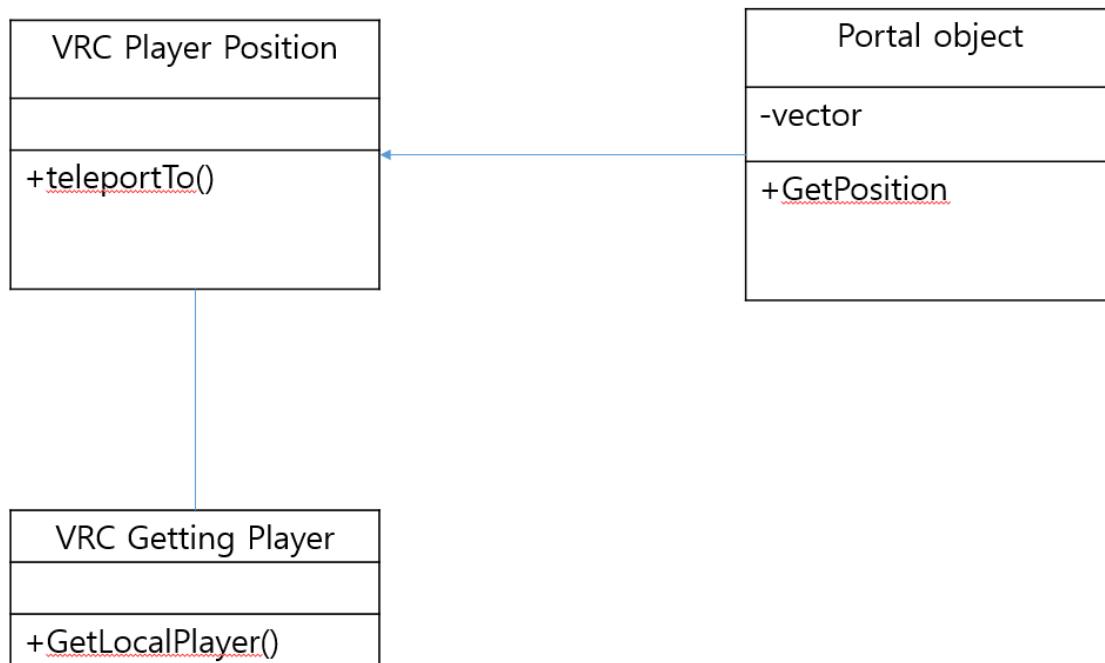
5.2 Overall Architecture

포탈과 전광판은 vrchat내부의 3D 모델과 유저의 상호작용, 미니맵은 유저와 unity엔진과의 상호작용, 텔레포트는 vrchat ui와 유저의 상호작용으로 이루어져있다. 위의 상호작용들은 Vrchat sdk인 udon을 베이스로 하고 있다.

5.3 Subcomponents

5.3.1 포탈

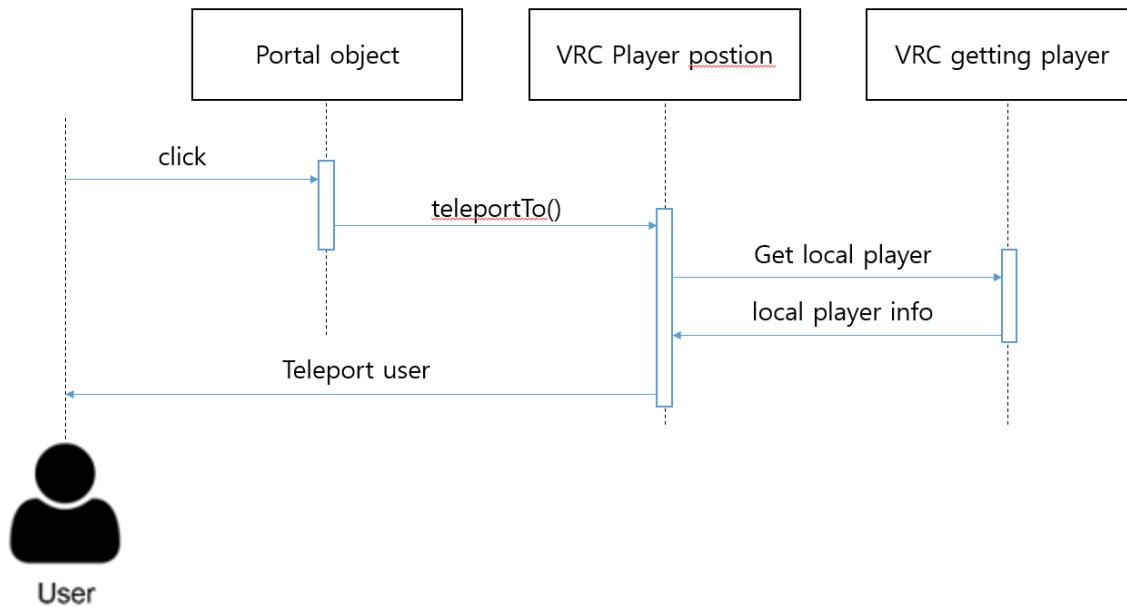
5.3.1.1 Class diagram



[Figure 13] Class Diagram of 포탈

portal object는 목적지의 vector값을 가지고 있다. VRC Player position 클래스는 플레이어를 teleportTo()의 인자에 해당하는 위치로 옮겨주는 역할을 한다. VRC Getting Player는 Udon 스크립트를 실행중인 플레이어 인스턴스를 찾는 역할을 한다.

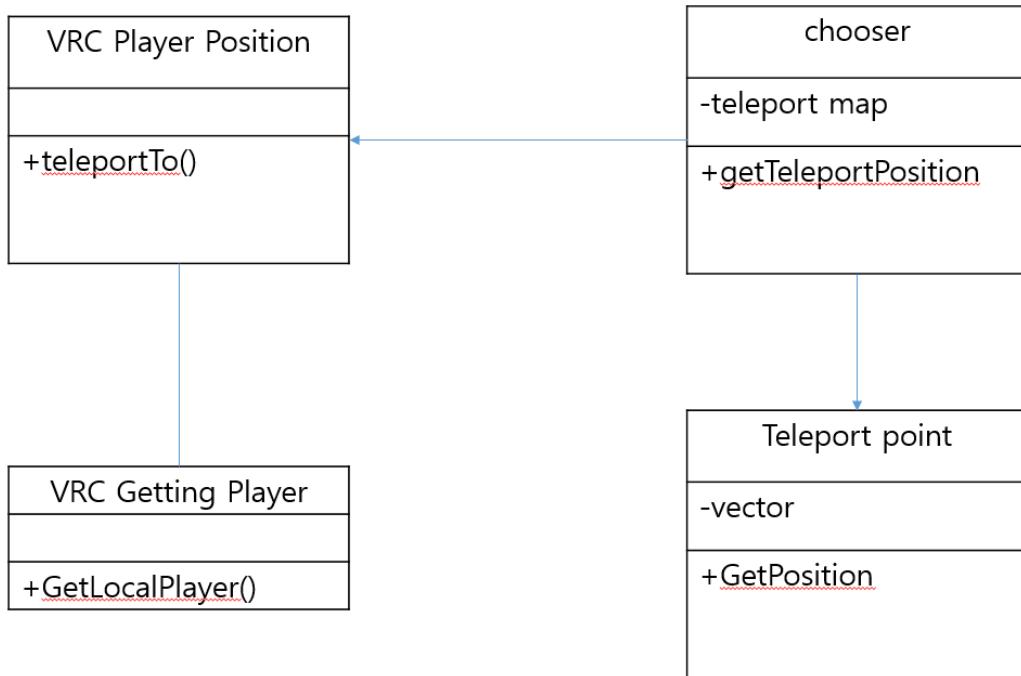
5.3.1.2 Sequence diagram



[Figure 14] Sequence Diagram of 포탈

5.3.2 텔레포트

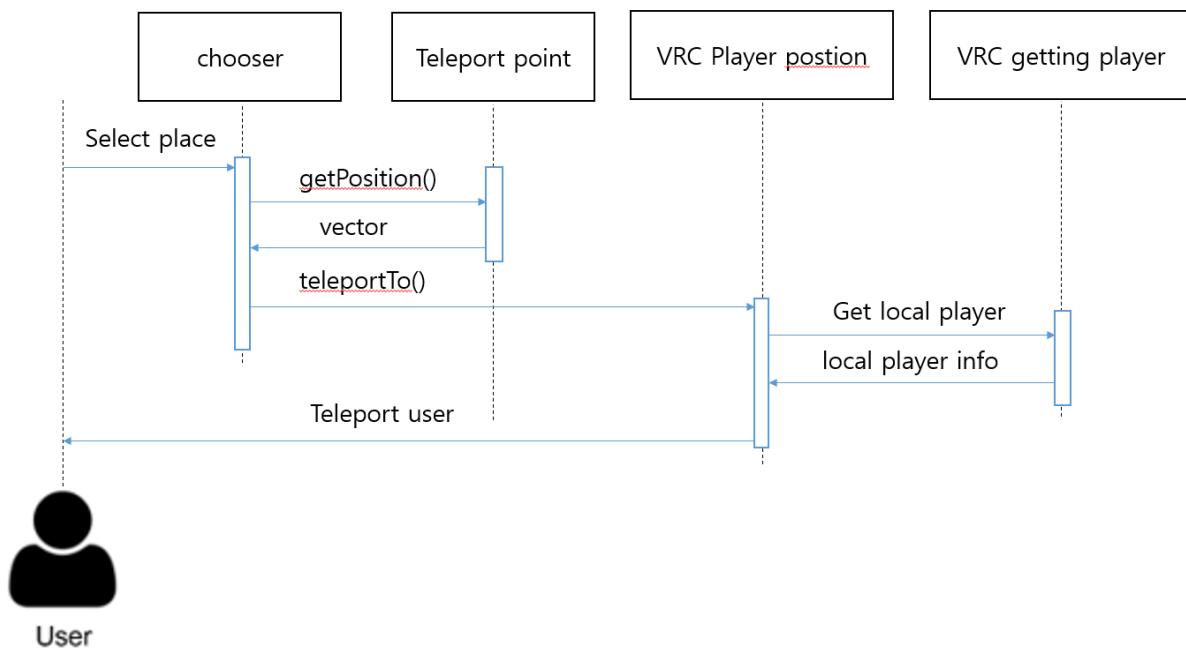
5.3.2.1 Class diagram



[Figure 15] Class Diagram of 텔레포트

chooser는 텔레포트할 장소들의 map을 가지고 있다. Teleport point는 자신의 장소를 반환하는 getPosition을 가지고 있다.

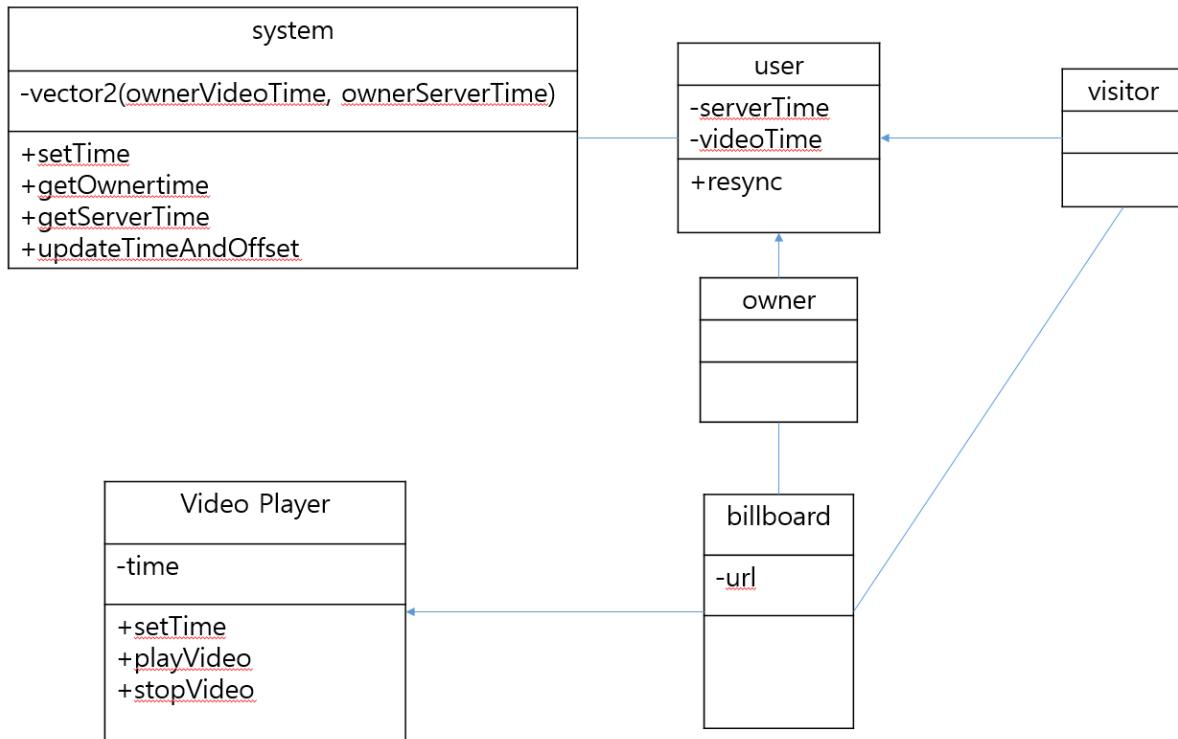
5.3.2.2 Sequence diagram



[Figure 16] Sequence Diagram of 텔레포트

5.3.3 전광판

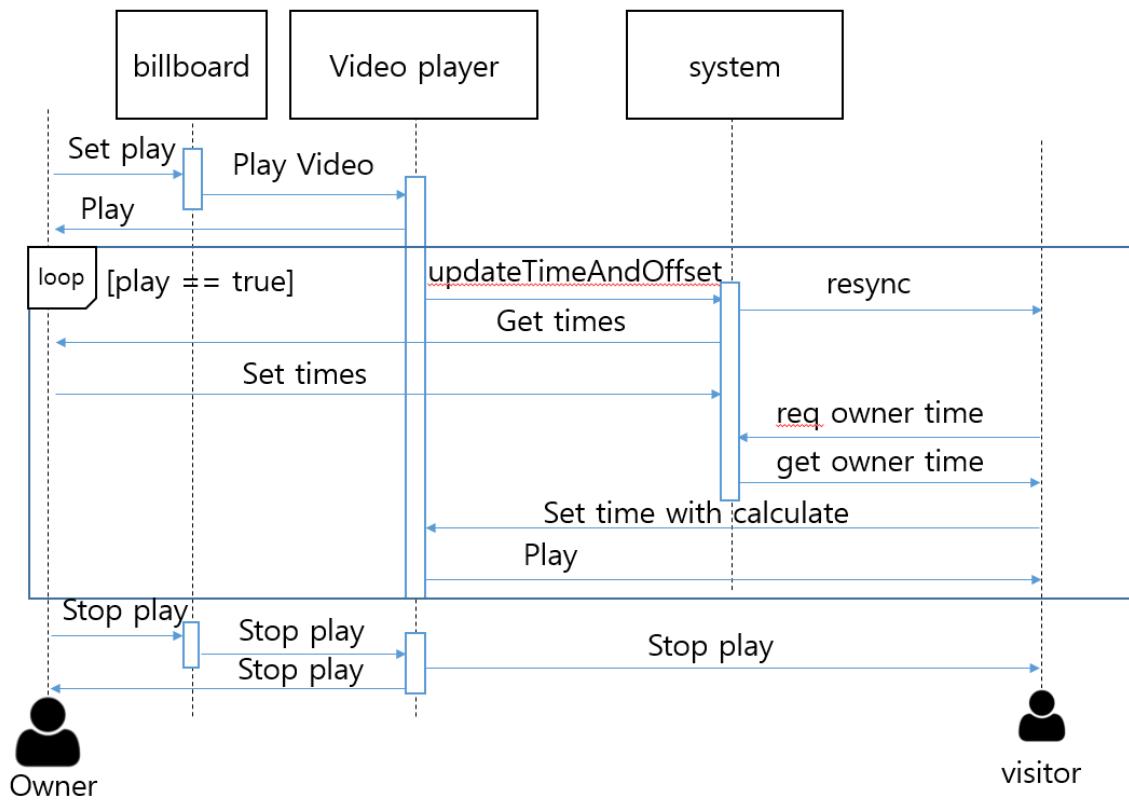
5.3.3.1 Class diagram



[Figure 17] Class Diagram of 전광판

유저를 상속받는 owner, visitor가 있다. System은 owner의 비디오 재생 시간, 서버 시간을 가지고 있다. 서버는 updateTimeAndOffset이라는 이벤트를 계속 실행한다. 유저의 resync이벤트는 updateTimeAndOffset에 의해 촉발되고 visitor에게만 이벤트가 전달된다. Video player는 비디오의 특정 시점으로 이동하는 setTime과 playVideo, stopVideo가 있다.

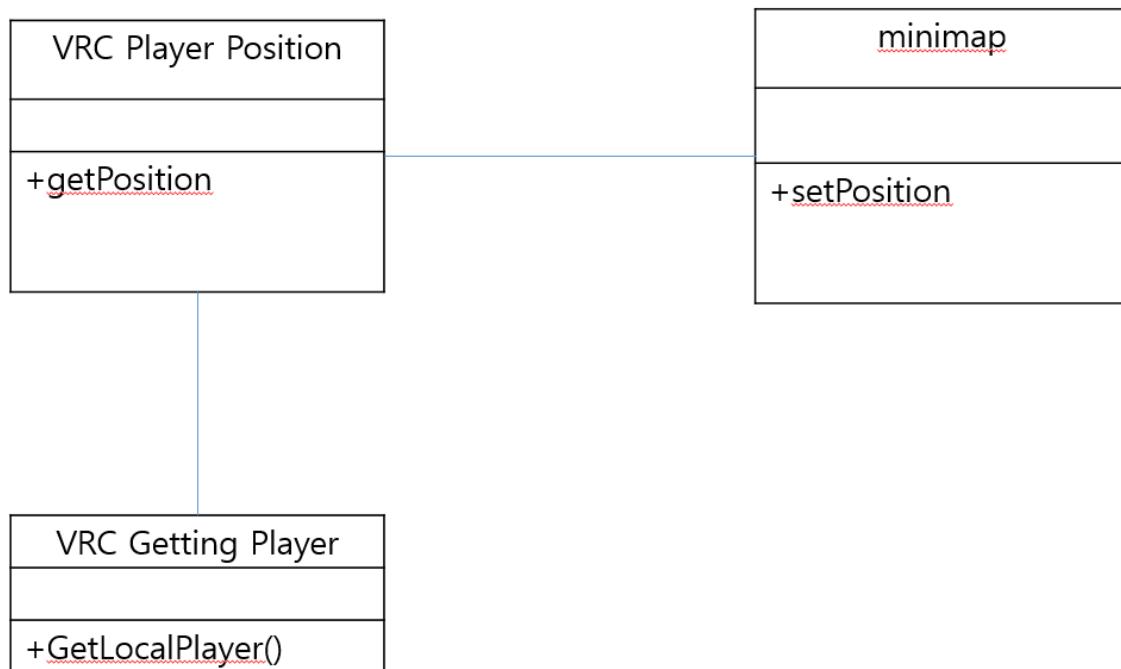
5.3.3.2 Sequence Diagram



[Figure 18] Sequence Diagram of 전광판

5.3.4 디자인 맵

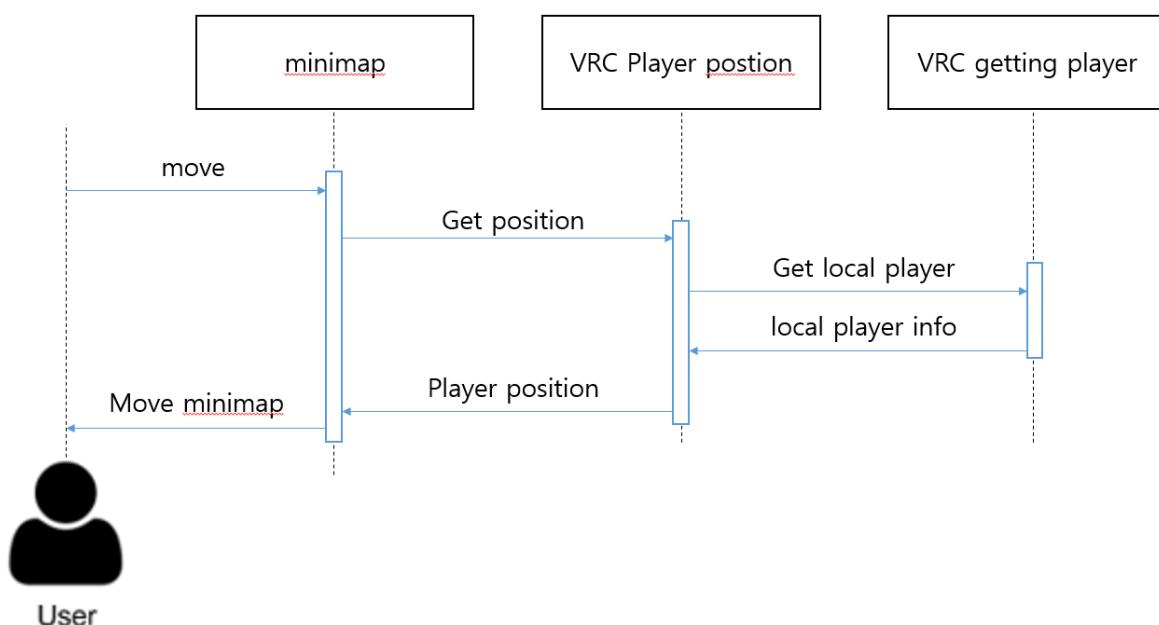
5.3.4.1 Class diagram



[Figure 19] Class Diagram of 미니맵

미니맵은 유저가 움직임에 따라 setPosition을 적절히 실행한다.

5.3.4.2 Sequence diagram



[Figure 20] Sequence Diagram of 미니맵

6. Testing Plan

6.1 Objectives

이 섹션에서는 개발 테스트, 배포 테스트 및 사용자 테스트의 세가지 주요 테스트 계획에 대해서 서술합니다. 각 테스트는 프로젝트 진행 단계에서 일어날 수 있는 잠재적인 오류를 줄임으로써 프로젝트 마지막 단계에서 해당 오류가 문제를 일으켜 프로젝트 비용이 상승하게 되는 상황을 방지할 수 있습니다.

6.2 Testing Policy

6.2.1 개발 테스트(Development Testing)

개발 테스트는 개발단계에서 일어날 수 있는 숨겨져 있는 오류들을 미리 확인할 수 있는 기회를 제공함으로써 프로젝트 마지막 단계에서 앞 단계로 재진입하지 않을 수 있도록 하는 역할을 해줍니다. 개발이 활발하게 이루어지고 있는 과정에서 수행되는 테스트이기 때문에, 테스트 도중에 충돌과 오류가 빈번하게 일어날 수 있다는 특징이 있습니다. 개발 테스트에서는 주로 소프트웨어의 각 컴포넌트가 의도한 대로 작동을 하는지, 각 컴포넌트들이 서로 상호작용을 할 때 충돌없이 상호작용이 이루어지는지를 확인할 수 있다.

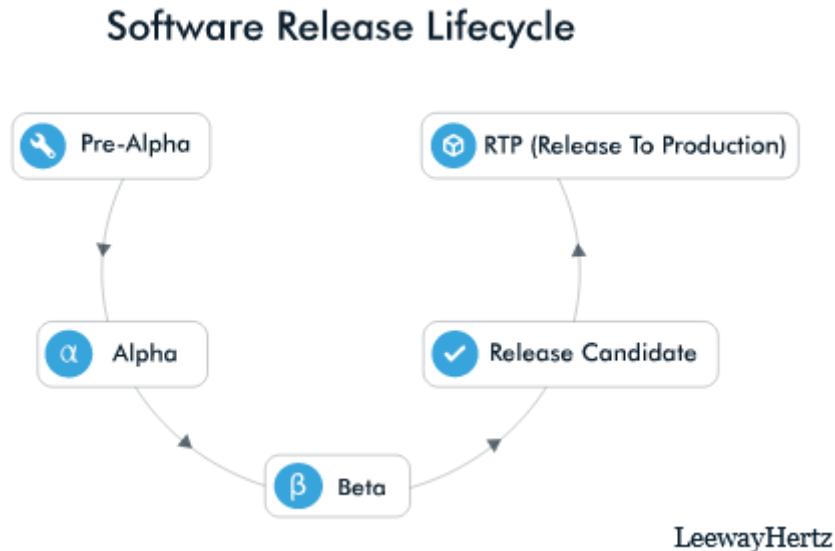
6.2.1.1 Performance

저희 소프트웨어에서는 캐릭터가 월드 내의 다른 개체들(의자, 건물, 버튼 등)과 상호작용하는 기능들이 성능에 가장 큰 영향을 끼칩니다. 버튼을 통해서 오디오와 비디오 컨텐츠에 접근하기 때문에 해당 미디어들에 빠르게 접근할 수 있도록 최적화가 필요합니다.

6.2.1.2 Reliability

저희 소프트웨어는 오류, 고장없이 서비스를 제공해야 합니다. 예를 들어 캐릭터가 특정 건물에 포탈을 통해서 진입하려고 할 때, 캐릭터와 월드가 의도한 대로 이동이 이루어지는지를 확인해야합니다. 작은 오류들이 쌓여서 또 다른 오류 상황을 만들어낼 수 있기 때문에, 사소한 오류라도 체크하고 개선해야 합니다.

6.2.2 배포 테스트 (Release Testing)



[Figure 21] Software Release Lifecycle

소프트웨어는 결국 최종적으로 실제 사용 환경에 배포되어서 사용될 것이기 때문에, 실제 환경에서 문제없이 작동되는지 테스트하는 것이 중요합니다. 좋은 소프트웨어라도 출시 과정에 있어서 어떤 장애물이 있다면, 사용자들은 해당 소프트웨어를 경험해보기도 전에 안좋은 인상을 갖게 될 가능성이 있습니다. 그렇기 때문에 배포 과정에서 문제를 일으키지 않도록 미리 테스트하는 것이 중요합니다. 배포 테스트가 이루어지려면, 테스트 이전에 우리의 소프트웨어가 작동할 환경을 준비하는 것이 선행되어야 합니다. 배포 테스트는 보통 Pre-Alpha, Alpha, Beta 테스트를 거치고, Release Candidate로 정해지면 배포되게 됩니다. Pre-Alpha 단계에서는 각각의 컴포넌트가 제대로 구현되었는지를 확인합니다. pre-alpha 단계에서 발견된 버그들을 alpha test 단계에서 수정하게 됩니다. 해당 버그들을 수정한 후에야 소프트웨어가 feature

complete 단계에 이르렀다고 말할 수 있게 됩니다. Beta testing 단계에서는 performance나 crashing issue들을 다루게 됩니다.

6.2.3 사용자 테스트 (User Testing)

사용자 테스트에서는 사용자 테스트를 진행하기 위한 일어날 수 있는 시나리오와 현실적인 상황을 설정합니다. 저희 프로그램에서는 캐릭터가 예상 범위에 없는 행동을 했을 때, 어떻게 처리해야 할지에 대한 시나리오를 설정할 수 있습니다. 그러한 테스트를 위해서 실제 캠퍼스 투어를 하는 것처럼 가정해서 실험을 진행한 후, use case를 수집할 수 있습니다. 실제 투어 진행을 20명 단위로 진행한다고 가정하고, 실험을 진행하게 됩니다.

6.2.4 Testing Case

테스트 케이스는 어플리케이션의 대표적인 기능별로 테스트를 진행하게 됩니다. 저희 소프트웨어의 기능은 크게 이동, 미디어 재생, 월드 내의 맵 이동, 캐릭터간의 상호 작용, 미니맵 작동 여부 등등이 있습니다. 각 기능별로 성능, 오류 발생 여부를 테스트하게 됩니다.

7. Development Environment

7.1 Objectives

본 챕터에서는 System Development를 위해 필요한 개발 환경과 언어에 대해 기술한다. 크게 Frontend Environment와 Backend Environment로 나눠 개발 환경 또는 언어를 제시하고 개발에 적용되는 Constraints, Assumptions, Dependencies를 기술한다.

7.2 Frontend Environment

7.2.1 Unity



[Figure 22] Unity

Unity는 3D 및 2D Video Game의 Development Environment를 제공하는 Game Engine이자, 3D Animation과 Visualization of Architecture, VR 등 Interactive 컨텐츠 제작을 위한 통합 제작 도구이다. 본 프로젝트에서는 VRChat SDK3와 함께 Avatar를 생성하고 World를 구성하는 역할을 한다.

7.2.2 VRChat SDK3



[Figure 23] VRChat

VRChat은 3D Character Model로 구현된 다른 Player들과 상호작용하는 대규모 다중 사용자 온라인 가상 현실 소셜 서비스이다.

사용하는 Package는 Avatar, World 두 개를 사용하는데 Avatar의 경우 VRCSDK3-Avatar를 사용하며 Avatar 생성을 위한 많은 기능들이 존재하는 Framework이라고 생각하면 된다.

World의 경우 VRCSDK3-World를 사용하며 World 내의 동작을 구현하는 VRChat Udon이 함께

제공된다. 본 Project에서는 World와 Avatar를 다 구성하지만 Avatar와 World의 Model을 구현하는데 사용되어진다.

7.3 Backend Environment

7.2.3 Udon



[Figure 24] Udon

VRChat Udon은 VRChat에서 자체적으로 개발한 프로그래밍 언어이다. Node와 Wire(noodles)를 이용하여 입출력을 연결하는 Built-in Visual Programming interface인 VRChat Udon을 통해 더욱 안전하고 효율적이고 간편하도록 설계되어진 언어이다. Trigger와 Action을 통해 동작을 구성하고 복제할 수도 있다. 또한 Udon은 VRChat Client와 Unity에서 모두 실행돼 쉽게 Testing과 Debugging을 할 수 있다. 본 프로젝트에서는 World내 기능 구현과 Avatar와의 Interaction과 관련한 기능을 구현하는데 사용되어진다.

7.2.3 Github



[Figure 25] Github

Github는 Git을 이용한 소프트웨어 개발 및 버전 관리를 위해 만들어졌다. Git의 분산 제어 및 소스 코드 관리(SCM) 기능을 제공한다. 접근 제어와 버그 추적, 기능 요청, 작업 관리, 지속적인 통합 등 Management를 향상시켜준다.

7.4 Constraints

기본적으로 Constraints는 다음과 같다. 추가적으로 Memory와 Design Constraints가 존재한다.

- Unity 2019.4.31f1을 사용하여 VRChat World를 구현한다.
- 실제 캠퍼스의 모습을 기반으로 World를 구현한다.
- 단체 투어의 경우 성균관대학교 학생들로 입장은 제한하기 위해 World를 Invite 모드로 설정한다.
- 개인 투어의 경우 불건전한 대화 등을 방지하기 위한 방장의 관리가 필요하다.
- VR 환경과 PC 환경에서 모두 Test를 진행한다.
- 실시간 음성 채팅을 위한 마이크 사용이 필요하다.
- 실시간 음성 채팅을 통한 소통이 중요하므로, 인터넷 연결 환경이 좋은 곳에서 시스템이 이용되어야 한다.

7.4.1 Memory Constraints

해당 프로그램은 최소 4GB 이상의 RAM이 구성된 Desktop에서 실행되어야 하며, 설치와 실행을 위해서 최소 2GB의 저장 공간이 요구된다.

7.4.2 Design Constraints

시스템은 Desktop과 Windows 환경을 통해서 이용할 수 있어야한다. VRChat이 지원하는 Hardware의 최저사양을 만족해야 한다. VR 기기가 없어도 시스템을 원활히 이용할 수 있어야한다. 시스템은 또한 VRChat Platform 위에서 개발 돼야한다. 시스템의 모든 변수들 및 함수들의 이름은 파스칼 케이스(Pascal case) 네이밍 컨벤션을 따른다.

7.5 Assumptions and Dependencies

본 문서는 구현할 메타버스 스마트 캠퍼스 투어 소프트웨어는 최소 Windows 7 이상의 OS에서 VRChat이 사용하고 있는 Unity 2019.4.31f1 Version에서 개발된다는 가정하에 작성되었다. Windows 이외의 OS 또는 SW의 Version 이외에서는 적용되지 않을 수 있다.

8. Supporting Information

8.1 Software Design Specification

이 디자인 명세서 문서는 IEEE 표준안 (IEEE Std. 1012-2012, IEEE Standard for Software Verification & Validation)에 따라 작성되었습니다.

8.2 Document History

Date	Version	Description	Writer
2021/11/12	0.1	양식 작성	한정우
2021/11/13	1.0	1 작성, 8 작성	한정우
2021/11/14	1.1	2 작성	박성원

2021/11/14	1.2	3 작성	김휘원
2021/11/15	1.3	4 작성	박수현
2021/11/15	1.4	5 작성	김태현
2021/11/16	1.5	6 작성	권준희
2021/11/17	1.6	7 작성	김장훈
2021/11/18	1.7	최종본 양식 수합	한정우

[Table 9] Document History