

토 대 리

- VR 가상 토론 시스템 -

○ 제정일자 : 2017년 01월 05일

○ 문서버전 : Ver 1.0○ 팀명 : 토대리 제작 위원회





문 서 승 인 정 보

프로젝트 명	토대리
TASK 명	VR 가상 토론 시스템
문 서 명	프로젝트 완료보고서
발행 년 월일	2017년 01월 05일

구 분	성 명	서 명	일 자
작 성 자	임지훈		2017.01.05
작 성 자	강승리		2017.01.05
작 성 자	손민수		2017.01.05
Technical Leader	강승리		2017.01.05
Team Leader	임지훈		2017.01.05
프로젝트관리자	문상환		2017.01.05



문 서 이 력 정 보

Ver.	Page	작성일자	변경사항	작성자	승인자	승인일자
1.0	110	2017.01.05	제 정	손민수		
1.1	243	2017.01.07	수 정	손민수		
2.0	262	2017.010	수 정	손민수		



1.	프	로젝트 배경	6
	A.	솔루션 개발의 필요성	6
	В.	신기술 개발 시 파급효과 및 활용방안	6
	C.	프로젝트 개발 목적 및 당위성	6
	D.	프로젝트 개발 분야	6
2.	<u>팀</u> 음	원 소개 및 역할	7
3.	아카	키텍처	10
	A.	프로젝트 진행 일정	11
5.	<u> </u>	로젝트 진행 보고	15
	A.	개인별 개발 일정	15
	В.	팀 일일 진행	123
	C.	회의록	137
6.	<u> </u>	로젝트 개발 내용	180
	A.	Source Code	180
	В.	사용 주요 기술	217
7.	<u>상</u> /	네 개발 내용	218
	A.	시스템 아키텍처	218
	В.	시스템 설계 및 구현	219
		i 기능정의리스트	219



		ii. UI 구현	222
	C.	개발환경 설치	240
8.	사용	용자 매뉴얼	255
9.	프로	르젝트 <u>마무리</u>	256
	A.	기대효과	256
	В.	문제점	256
	C.	개선방안	256
	D.	어려웠던 점	257
	E.	참고문헌 및 논문	259
	F.	참고사이트	259
	_	FIOLES A 7L	200



1. 프로젝트 배경

A. 솔루션 개발의 필요성

현재 한국 교육 시스템은 주입식 교육제도이다. 비록 지금 교육 정책이 토론 수업의 분량을 늘리고 있으나 아직 딱딱한 방식으로 인하여 학생들의 흥미를 돋구기에는 힘든 점이 발견된다. 또한 토론 수업 시 책상 등 토론 수업 환경을 조성하는 데 있어 소요되는 시간 및 책상 등 옮기는 물리적인 체력 소모가 유발된다. 이에 따라 학교, 가정 등에 보급된 PC를 이용하고 저렴하고 쉽게 구할 수 있는 VR을 이용하여 가상현실에서 토론을 즐길 수 있는 환경을 제공한다.

B. 신기술 개발 시 파급효과 및 활용방안

기존 토론 수업 시간에 학생들의 흥미를 유발하고 물리적, 시간적 낭비를 줄일 수 있다.

C. 프로젝트 개발 목적 및 당위성

가상 토론 장소를 제공하여 원거리에 있는 사용자들이 공간에 제약 받지 않고 어디서든 참여가 가능하여 토론이 실생활에 정착할 것이다.

D. 프로젝트 개발 분야

VR 가상 토론 시스템은 기존 웹 혹은 게임 형식의 채팅 시스템이 아닌 VR을 이용한 가상 현실 토론 시스템이다. 가상 현실이란 컴퓨터 등을 사용한 인공적인 기술로 실제와 유사하지만 실제가 아닌 어떤 특정한 환경이나 상황 혹은 그 기술 자체를 의미한다. 최근 떠오르고 있는 VR기기는 누구나 손쉽게 가상현실에 접근이 가능하다.

개발된 "VR 가상 토론 시스템"은 3D Game(Client), IOCP Server, VR의 접목으로 이루어 진다. Client에서 User의 행동, Animation, Voice Data는 Byte Stream으로 변환하여 Server에 전송하게된다. Server의 model이 IOCP model이기 때문에 다른 Client에게 안정적으로 Data를 전송할수 있다. 결과적으로 Server와 Client의 안정성을 높이는 결과를 초래한다.

Client Service를 이용하기 위해서는 계정이 필요하다. 계정 등록 시 Text Data 뿐만 아니라 Image file을 Server에 전송하여 Database에 저장을 한다. 이 Image file은 Client 내 Stage에서 자신의 Profile Image를 띄우기 위한 수단 및 자신을 친구 추가 하였을 때 사용되는 Image 이다. 또한 이 Image file은 Database Server에 Id로 file name이 결정된다. 다시 Server에서 Cleint에게 Send 해도 그대로 적용된다. 이 Image는 친구 정보, 내정보에 등록된다.



2. <u>팀원 소개 및 역할</u>

책임	직무	사진	이름	역할 설명
Leader	팀장		임지훈	팀원 의견 조율방향 결정기술 결정일정 관리기획서 관리회의 주제 및 회의록 정리
	Technical	POR SUKVINIA 40	강승리	기술 서포트 및 의견 조율 서포
	Leader	200 000 000 000 000 000 000 000 000 000		트, 팀장 부재 시 팀장 역할
UX	정	PART STATE OF THE	강승리	UX 시퀀스 설계 및 구현
	부		임지훈	UX 시퀀스 설계 및 구현



	•		_	
UI, 모델링	정	DATE OF THE CONTRACT OF THE CO	강승리	UI 모델링 및 배경 및 맵 제작 (Unity_3D)
	부		손민수	UI 모델링 및 배경 및 맵 제작 (Unity_3D)
Voice 통신	정		손민수	Server - Client 음성 통신 및 음 성 서버 구축(UDP)
	부	Bell address of	강승리	Server - Client 음성 통신 및 음 성 서버 구축(UDP)
c/s 통신	정		임지훈	Server – Client 통신 및 서버구축 (TCP)

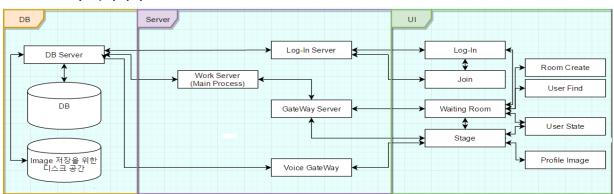


	부	BER SOLVER OF	강승리	Server - Client 통신 및 서버구축 (TCP)
DB	정		임지훈	DB 설계 및 DB 서버 구축
	부		손민수	DB 설계 및 DB 서버 구축
VR	정		손민수	Trinus VR, Client Connect
	부		임지훈	Trinus VR, Client Connect

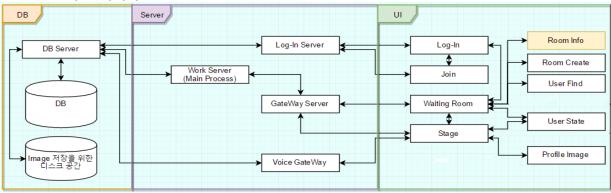


3. 아키텍처

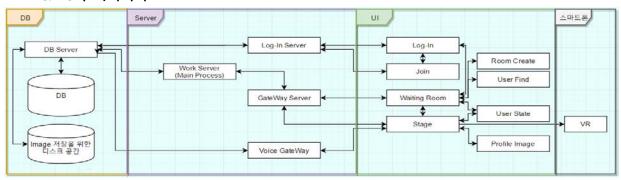
A. 1차 아키텍처



B. 2차 아키텍처



C. 3차 아키텍처





4. 프로젝트 개발 일정

A. 프로젝트 진행 일정

일정 완료	\rightarrow
공휴일	

1) 프로젝트 기획 /분석 /설계

어므브아	월 21 22 23 24 주제선정	11	윌					
· 급구 문약	ᅙᆒ	21	22	23	24	25	26	i
	보다 일 21 2 절 주제선정 Image: First Section of the content of the							
교문제는 기침	팀원 역할 선정	Ш	1					
프로젝트 기획	일정 예측		I	1				
입무 문아일주제선정 팀원 역할 선정 일정 예측 기획서 작성프로젝트 분석기능 정의	기획서 작성		Л	1				
프로젝트 분석	기능 정의			Г	1			
프로젝트 설계	S/W UI				1			

2) **11월**

i. UI, Modeling, UX

नुग		11월								
뎰	21	22	23	24	25	26	27	28	29	30
LogIn										
Join										
Wating Room										1
Stage										1
User State										
Profile Image										
Debug										



ii. Server/Database

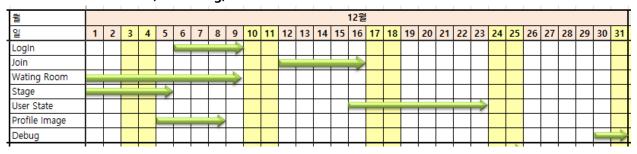
월		11월								
일	21	22	23	24	25	26	27	28	29	30
Login Server										
GateWay Server										
Work Server										
Database										
DB Server										1
Voice GateWay										1
Debug										

iii. Function

				11	윌				
21	22	23	24	25	26	27	28	29	30
		Ш							1
	21	21 22	21 22 23	21 22 23 24		11월 21 22 23 24 25 26			11월 21 22 23 24 25 <mark>26 27 28 29</mark>

3) **2순위**

i. UI, Modeling, UX





ii. Server/Database

쥘																12월															
일	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
LogIn Server																						Ш			$^{\uparrow}$						
GateWay Server																				Ш		1									
Work Server																				J											
Database																															
DB Server					\Rightarrow																										
Voice GateWay																												1			
Debug																										Л					1

iii. Function

쥘																12월															
일	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Voice Network																												1			
Trinus VR																															1

4) 3순위

- ···													
UI, Modeling	, UX												
월							1월						
뎰	1	2	3	4	5	6	7	8	9	10	11	12	13
LogIn													
Join													
Wating Room													
Stage													
User State													Г
Profile Image													
Debug									1				



ii. Server/Database

뒬							1월						
일	1	2	3	4	5	6	7	8	9	10	11	12	13
Login Server													
GateWay Server													
Work Server													
Database													
DB Server													
Voice GateWay													
Debug									4				

iii. Function

g a		1월											
일	1	2	3	4	5	6	7	8	9	10	11	12	13
Voice Network													
Trinus VR													

iv. Wrap-up

쥘							1월						
일	1	2	3	4	5	6	7	8	9	10	11	12	13
Wrap-up	Ш											1	
Documentation						Л				1			



5. 프로젝트 진행 보고

A. 개인별 개발 일정

i. 임지훈

	개발일지
날짜	2016-11-22 작성시간 2016-11-22 9:15
이름	임지훈
개발과제	1. 팀 주제 회의 및 선정 2. 기획서 작성 3. 데모데이 박람회 관람
내용	1. 팀 주제 회의 및 선정 팀 주제의 목표를 정확하게 정하고, 프로젝트의 명칭, 목적, 이점, 필요성, 수행이유등 회의 2. 기획서 작성 앞서 회의한 내용을 토대로 기획서 부분 작성 3. 데모데이 박람회 관람 아이디어 구상 및 구체화를 위한 데모데이 박람회 관람



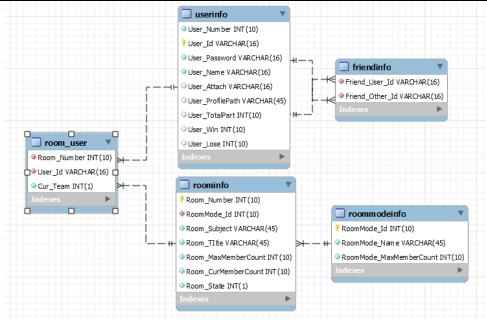
	개발일지
날짜	2016-11-23 작성시간 2016-11-23 9:15
이름	임지훈
개발과제	1. 기획서 회의 및 작성 2. MySQL 설치 및 연동
내용	1. 기획서 회의 및 작성 팀원간 역할 및 할일 회의 및 결정. 프로젝트 1 차, 2 차, 3 차 목표 회의 및 설정. 서버 아키텍처 회의 및 설계 1 차, 2 차, 3 차 에 따른 시스템 아키텍처 회의 및 설계 UI 순서도 회의 및 설계, 각화면에 대한 UI 배치 구상화 및 회의 팀일정, 개인일정 회의 및 작성 2. MySQL 설치 및 연동 MySQL홈페이지에서 각종 필요한 툴 다운로드 C++에서 MySQL 연동후 버전 확인 테스트



	개발일지												
날짜	2016-11-24 작성시간 2016-11-24 9:15												
이름	임지훈												
	1. 기능정의서 작성												
	2. DB 설계												
개발과제	3. DB Table 정의서 작성												
	4. SW_UI 구상도 작성												
	5. Protocol 설계도 작성												
	1. 기능정의서 작성 :												
	팀 회의 후, 개발할 기능들을 구상. 개박하 기능들을 부르별로 그류하												
	개발한 기능들을 분류별로 그룹화. 기능들에 대해 여화 부단												
	기능들에 대해 역할 분담. 기능들에 대한 우선순위 결정												
	기능들에 대한 우선순위 결정.												
	2. DB 설계 :												
	서버에 저장이 될 데이터들을 분류.												
	데이터들을 그룹화												
	Table 을 구성하고 정규화 확인.												
내용	Table 간 관계 설정.												
	UserInfo, FriendInfo, LogInInfo, RoomInfo, RoomModeInfo, Room_User Table 구성												
	3. DB Table 정의서 작성												
	설계한 DB Table 들을 문서화.												
	4. SW_UI 구상도 작성												
	필요한 화면 분류 (로그인화면, 회원가입화면, 로비화면, 내상태창, 투표창, 결과창등등)												
	각 화면에 대한 UI를 구상 후 이미지화												
	5. Protocol 설계도 작성												
	고정길이 헤더 + 가변길이 헤더로 패킷 구성												



	개발일지										
날짜	2016-11-25 작성시간 2016-11-25 9:15										
이름	임지훈										
	1. DB 구축										
711 11 71 71	2. C++ 과 MySQL 의 연동										
개발과제	3. DB 서버 구축 시작										
	1. DB 구축										
	Workbench 로 DB 와 Table 구성 및 Attribute 설정										
내용	2. C++ 과 MySQL 의 연동 C++ 코드로 MySQL 에 Direct 로 연결성공 C++ 코드로 MySQL 에 ODBC 를 이용하여 연결 성공 C++ 코드로 INSERT, DELETE 테스트 및 Foreign Key 의 CASCADE 작동 테스트										
	3. DB 서버 구축 시작										
	DB 서버를 MySQL 에 Direct 로 연결하여 구축 시작										



페이지 18 / 261



			개발일지
날짜	2016-11-28	작성시간	2016-11-28 9:15
이름	임지훈		
개발과제	1. DB 서버 투 2. DB 서버 대		
내용	2. DB 서버 다	르그인부터 ² 바무리	각 기능에 대한 쿼리문 작성 DB 서버 완료



	개발일지			
날짜	2016-11-29 작성시간 2016-11-29 9:15			
이름	임지훈			
개발과제	1. DB Server 구축 완료. 2. Unity UI 와의 통신완료.			
내용	1. DB Server 구축 완료. IOCP 를 이용한 DB Server 구축 완료. 유저가입, 로그인, 방생성, 방들어가기, 방목록받아오기 등등 구현완료. 2. Unity UI 와의 통신완료. Test 용 Unity UI 구성 완료 및 구현. Unity UI 와 DB Server 간의 통신 및 DB 업데이트 확인 완료			

```
case CMD_JOIN:
    ret_val = Join(split_buf);
    break;
case CMD_LOGIN:
    ret_val = Login(split_buf);
    break;
case CMD_LOGOUT:
    ret_val = LogOut(split_buf);
    break;
case CMD_CREATE_ROOM:
    ret_val = CreateRoom(split_buf);
    break;
case CMD_ENTER_ROOM:
    ret_val = EnterRoom(split_buf);
    break;
case CMD_LEAVE_ROOM:
    ret_val = LeaveRoom(split_buf);
    break;
case CMD_START_ROOM:
    ret_val = StartRoom(split_buf);
    break;
case CMD_GET_ROOM_LIST:
    ret_val = GetRoomList(retBuf);
    break;
```



개발일지					
 날짜	2016-11-30	작성시간 2016-11-30 9:15			
이름	임지훈				
개발과제	WorkServer Unity Anima	와 Unity 통신 테스트 및 구상 tor 구상			
내용	1. WorkServer 와 Unity 통신 테스트 및 구상 WorkServer 에 IOCP 모델 적용. Unity 의 Stage 안에서 케릭터들끼리 통신방법 구상 및 테스트 Animation 효과의 통신방법 구상 2. Unity Animator 구상 Skeleton 및 Avatar 적용 케릭터 구성 및 Animator 적용 통신방법 테스트 및 구상				



	개발일지				
날짜	2016-12-01 작성시간 2016-12-01 9:15				
이름	임지훈				
개발과제	1. Unity Character 와 WorkServer 의 통신 테스트 마무리 2. 다중접속을 위한 WorkServer 구축				
내용	1. Unity Character 와 WorkServer 의 통신 테스트 마무리 Unity Character (Script) 와 WorkServer 의 통신 테스트 및 메카님 애니메이터의 변수 송수신 확인 2. 다중접속을 위한 WorkServer 구축 다중접속을 위한 IOCP 모델 적용 및 디버깅 IOCP 의 GetQueuedCompletionStatus() 의 여러가지 상황에 따른 예외처리 적용중 Client 가 어디로 갈지 모르기 때문에				



개발일지				
날짜	2016-12-02	작성시간	2016-12-02 9:15	
이름	임지훈			
개발과제	2. 케릭터와 W	orkServer ਪ	Server 구축 완료 라의 통신 및 패킷(건트와 WorkServer	
내용	다중접속을 2. 케릭터와 W 하나의 케릭 각 이벤트에 3. 다중접속된	위해 TCP 통 forkServer 9 터와 WorkS 따른 패킷/ 각 클라이S 릭터(클라C	라의 통신 및 패킷(Server 와 통신. 생성. 언트와 WorkServer 기언트)와 WorkSer	· 완료.



개발일지				
날짜	2016-12-05	작성시간	2016-12-05 9:15	
이름	임지훈			
개발과제	1. 케릭터 위치			
내용	따라서 60fp	패킷을 60 s 로 보내지	fps 로 송수신 하면 클라이언트가 너무 느리고 버벅거림. 않더라도 상대방 케릭터가 자연스럽게 움직이는 작업 필요 	



개발일지			
	게 ㄹ ㄹ 시		
날짜	2016-12-06 작성시간 2016-12-06 9:15		
이름	임지훈		
	1. 캐릭터 위치 동기화		
	2. 애니메이션 클립 작업		
	3. 음성통신		
개발과제 개발과제			
게르되세			
	1. 캐릭터 위치 동기화		
	끊기는 캐릭터 이동해결중. 자문구함 TCP 로해도 로컬지역에서 두개의 케릭터만		
	송수신하기 때문에 느리면 안됨. 그래서 다시 처음부터 코딩해보기로함 하기전에 캐릭터 동작 네트워크도 같이 병행하기 위해 애니메이션 클립작업 수행		
	기획다 중작 대트워크로 같이 항황아기 위에 에디메이션 클립학립 구행 2. 애니메이션 클립 작업		
	기 기 기 기		
	일어서기 동작)		
	3. 음성통신		
내용	클립작업중 음성통신이 너무 딜레이 되기 때문에 음성통신부터 해결하기 위해 임시		
	투입.		



개발일지				
<u></u> 날짜	2016-12-07 작성시간 2016-12-07 9:15			
이름	임지훈			
개발과제	1. 음성통신 협업			
내용	1. 음성통신 협업 Unity 에서 C# Script 로 작업. Microphone 클래스와 AudioClip 클래스로 음성데이터 입력. Float 배열로 저장(AudioClip::GetData) Float 배열을 Byte 배열로 변환. Socket 클래스의 TCP 통신을 사용하여 송신. 다른 쪽에서 Byte 배열을 수신. Byte 배열을 Float 배열로 변환. Float 배열을 AudioSource 클래스와 AudioClip 클래스로 음성데이터 출력. (AudioClip::SetData)			



	개발일지					
날짜	2016-12-08 작성시간 2016-12-08 9:15					
이름	2016-12-08 작성시간 2016-12-08 9:15 임지훈					
-	1. 음성통신 협업					
	2. 캐릭터 위치 동기화 재시작					
개발과제						
	1. 음성통신 협업					
	음성 통신 중 끊김과 버벅임을 해결하기 위해 끊김이나 버벅임이 발생 할 시 오디오					
	재생을 0.1 초씩 멈춤으로 해결.					
	처음에 Thread 로 Recv 를 대기하고 있었으나, 공유변수 참조로 데이터가 중간중간					
	사라지는 현상 발생.					
	공유변수에 Critical Section 으로 lock 을 걸었으나, Unity 멈춤 (이유 못찾음. Unity 에서는 Thread 보다 CoRoutine 을 더 지향한다고 나옴.)					
	Thread 를 버리고 CoRoutine 에 Non-Blocking 소켓으로 작업.					
	송수신되는 속도는 살짝 줄었지만 데이터가 사라지지 않고 잘 넘어가는 것 확인.					
	팀원 민수에게 해당 소스 투척					
내용	2. 캐릭터 위치 동기화 재시작					
	2. 계곡되 위치 8기회 제치곡 음성통신 협업 하기 전에 위치 동기화를 시키고 있었으나, 실패.					
	다시 진행하기 위해 자료조사.(책, 인터넷) 여러 가지 방법과 문제에 대해 인식.					
	위치정보 송신부터 재시작.					



	개발일지				
날짜	2016-12-09 작성시간 2016-12-09 9:15				
이름	임지훈				
	1. 캐릭터 좌표 동기화				
	2. 캐릭터 애니메이션 통신				
개발과제					
기리되시					
	1. 캐릭터 좌표 동기화				
	캐릭터 좌표 동기화 다중접속 테스트 완료.				
	2. 캐릭터 애니메이션 통신 캐릭터 애니메이션 동작 시 다른 클라이언트에게도 똑같이 애니메이션 통신.				
	게 하다 에다테이는 중국 사 다른 필디어인트에게도 구들이 에다테이는 중단.				
내용					



	개발일지
날짜	2016-12-12 작성시간 2016-12-12 9:15
이름	임지훈
개발과제	1. 애니메이션 통신
내용	1. 애니메이션 통신 필요한 애니메이션 클립 작업. 워크서버와 다중 클라이언트간의 애니메이션 통신 테스트 완료. 2. 음성과 좌표이동 애니메이션 통신 클라이언트 결합 애니메이션 통신클라이언트와 좌표이동 통신 클라이언트 결합 완료. 음성과 다른 클라이언트 결합중.



	개발일지				
날짜	2016-12-13 작성시간 2016-12-13 9:15				
이름	임지훈				
개발과제	1. 각 클라이언트 필요 정보 전달 2. 서버 보완 3. 팀원 지원				
내용	1. 각 클라이언트 필요 정보 전달 팀원과 합심하여 클라이언트 네트워크 구현. 2. 서버 보완 클라이언트에서 전송하는 패킷 처리부분 수정 및 디버깅 및 부족한 부분 추가. 3. 팀원 지원 팀원의 네트워크 작업을 위한 서포트 및 지원.				



개발일지					
날짜	2016-12-14 작성시간 2016-12-14 9:15				
이름	임지훈				
개발과제	 서버에서 클라이언트로 사진 송, 수신 패킷 설계 사진 송, 수신 알고리즘 작성 서버 보완 저녁에 랜섬웨어 출현. 				
내용	1. 서버에서 클라이언트로 사진 송, 수신 패킷 설계 서버에서 클라이언트로 사진 송, 수신 패킷 설계 CMD (2): LENGTH (4): Len_ld(1): ld(Len_ld): Data() 2. 사진 송, 수신 알고리즘 작성 기존 패킷과의 헤더가 달라서 따로 송수신 작업 작성 3. 서버 보완 서버를 수정하면서 디버그 수정 4. 저녁에 랜섬웨어 출현.				



개발일지				
	<u> — </u>			
날짜	2016-12-15 작성시간 2016-12-15 9:15			
이름	임지훈			
	1. 랜섬웨어로 사라진 파일 확인 및 백업본 확인.			
	2. 로비까지 코드 및 디자인 합치기			
	3. 방 정보 송수신 및 클라이언트 방 정보 처리 작업.			
개발과제	4. 팀원 지원			
	4 과서에이크 지기자 파이 참이 미 배어비 참이			
	1. 랜섬웨어로 사라진 파일 확인 및 백업본 확인.			
	랜섬웨어로 클라이언트 위치동기화, 음성데이터 송수신 증발. 위치동기화는 아예 사라졌고, 음성데이터는 예전에 시작한지 몇 시간 된 코드 발견			
	위사용기와는 이에 사다였고, 음성대이나는 에션에 사막인지 및 사진 전 고드 글선 			
	2. 로비까지 코드 및 디자인 합치기			
	우선 내일 발표를 위한 로비까지의 코드 및 디자인을 합친다.			
	우면 내는 근표를 하면 모이까지의 모드 못 되자면을 합면하.			
	3. 방 정보 송수신 및 클라이언트 방 정보 처리 작업.			
	클라이언트에서 방 정보 송수신 및 방 정보 출력.			
내용	서버에서 방 정보 송수신 및 방 정보 출력.			
0	4. 팀원 지원			
	팀원의 원활한 클라이언트 네트워크 작업을 위한 서포트 및 지원.			



개발일지			
날짜	2016-12-16 작성시간 2016-12-16 9:15		
이름	임지훈		
개발과제	1. 클라이언트 방 출력 2. 방 정보 마우스호버 시 나오게 적용 3. 로비까지 코드 및 디자인 수정 및 보완		
내용	1. 클라이언트 방 출력 클라이언트에서 방 next prev refrash 버튼 추가, 버튼 이동 시 씬 이동. 버튼 호버 시 로비에 방 정보 출력 2. 방 정보 마우스호버 시 나오게 적용 방 정보를 마우스호버 시 마우스 옆에서 나오게 적용 3. 로비까지 코드 및 디자인 수정 및 보완 로비까지의 코드 재 통합 및 디버깅 및 수정, 보완, 시연, 발표준비		



개발일지			
날짜	2016-12-19 작성시간 2016-12-19 9:15		
이름	임지훈		
개발과제	1. 로비와 씬 코드 통합 및 디버깅. 2. 애니메이션 작업.		
내용	1. 로비와 씬 코드 통합 및 디버깅. 로비와 Stage 코드 통합 및 디버깅. 코드 통합 뒤 재분배 후 각 팀원마다 InStage 작업 시작 2. 애니메이션 작업. 케릭터 선발. 케릭터 뼈 리깅.		



개발일지				
<u></u> 날짜	2016-12-20 작성시간 2016-12-20 9:15			
이름	임지훈			
, =	1. 애니메이션 작업.			
	2. 서버와의 통신 베이스			
개발과제				
	1. 애니메이션 작업.			
	케릭터 뼈 리깅. 애니메이션 검색 및 선출.			
	Animator 설계 및 작업.			
	캐릭터 이동 적용.			
	2. 서버와의 통신 베이스			
	Animator 송수신을 위한 송수신 프로토콜 설계. 패킷 설계.			
내용				



개발일지				
 날짜	2016-12-21 작성시간 2016-12-21 9:15			
이름	임지훈			
	1. 서버와의 애니메이션 통신.			
	2. 서버와의 위치 통신 및 보정.			
개발과제				
	1. 서버와의 애니메이션 통신.			
	설계된 패킷에 따른 클라이언트와 서버에 함수 코딩. 및 적용. 설계된 프로토콜 대로 서버와 다중 클라이언트간의 애니메이션 송수신.			
내용	2. 서버와의 위치 통신 및 보정. 서버와 클라이언트간의 위치통신 및 보정 프로토콜 설계. 앞에 애니메이션 통신과 합하여 키 동기 방식 채택. 서버에 모든 캐릭터들의 키 상태 값을 0.05~0.1 초마다 재송신.(컴퓨터 사양에 대한 FixedUpdate 로 하면 느려질 수 도 있기 때문에 FixedUpdate -> Update 로 변경). 서버에서 모든 캐릭터의 키 상태 값을 각 클라이언트로 전송. 각 클라이언트에서 각 캐릭터마다 해당하는 플래그 값 적용. 이와 동시에 0.05~0.1 초마다 자기좌표 서버에 전송. 서버에서 모든 캐릭터의 위치정보도 계속 보관. 20 번에 한번씩 서버에서 클라이언트로 위치정보 전송 후 오차가 심할 시 위치되 및 방향보정			



	개발일지			
날짜	2016-12-22 작성시간 2016-12-22 9:15			
이름	임지훈			
개발과제	1. 서버와 클라이언트간의 키 방식 통신 및 위치정보 보정 추가 2. 보이스 통신 작업 진행			
내용	1. 서버와 클라이언트간의 키 방식 통신 및 위치정보 보정 추가 서버와 클라이언트간의 키 방식 통신 중 클라이언트에서 CoRoutine + NonBlocking 방식과, Blocking 방식의 속도 차 고려 일단 두 방식 모두 개발 2. 보이스 통신 작업 진행 서버와 클라이언트간의 단일 보이스 통신 성공. 클라이언트에서 수신 받은 서버의 보이스 데이터와 Audio 실행 시점과 자연스러운 싱크 맞추기 필요.			



			개발일지	
날짜	2016-12-23	작성시간	2016-12-23 9:15	
이름	임지훈	•		
개발과제		1. 멀티 음성통신 2. 프로젝트 통합 및 디버깅		
내용	이제 캐릭 음성 패킷 업데이트 리스너는 2. 프로젝트	선 싱크를 ! 터가 생성될 을 아이디 ! 본인캐릭터야 통합 및 다	때마다 오디오를 추가 결로 받아서 아이디에 맞는 캐릭터오디오에 음성데이터 추가	



개발일지				
2016-12-26 작성시간 2016-12-26 9:15				
임지훈				
1. 프로젝트 통합 및 디버깅				
2. 서버 코드	정리 및 디	버깅		
1. 프로젝트	 통합 및 디	버깅		
기능 통합	및 프로젝트	트 버그부분 디버깅.		
서버 정리	및 오류부분	분 디버깅.		
	임지훈 1. 프로젝트 2. 서버 코드 1. 프로젝트 기능 통합 2. 서버 코드			



	개발일지
날짜	2016-12-27 작성시간 2016-12-27 9:15
이름	임지훈
개발과제	1. 서버 디버깅 및 음성통신 디버깅
내용	1. 서버 디버깅 및 음성통신 디버깅 서버와 클라이언트간 음성통신 디버깅 및 오류 수정. 음성통신이 클라이언트간 CPU 성능 때문에 느린 CPU 에서는 끊김 현상 발생.



	개발일지
 날짜	2016-12-28 작성시간 2016-12-28 9:15
이름	임지훈
개발과제	1. 클라이언트간 음성통신간에 끊김 현상 해결
내용	1. 클라이언트간 음성통신간에 끊김 현상 해결 클라이언트간 성능 차에 의해 음성통신이 끊김. 우선 서버를 WIFI 에서 랜선으로 변경. 음성통신 딜레이를 0.09 초에서 0.5 초로 변경. 여러 디버깅 후 데이터 송수신횟수가 클라이언트에 부하를 주는 것 같아서 작은 데이터를 여러 번 송수신 하는 것 보다 큰 데이터를 한번에 송수신하는 것으로 해결.



개발일지			
날짜	2016-12-29	작성시간	2016-12-29 9:15
이름	임지훈		
개발과제	1. 캐릭터의 상태 동기화 2. 서버 디버깅		
내용	서버에서 싱 해준다.	입장하면 :태를 수시를 모를 경우	이미 입장한 사람들의 캐릭터의 상태가 초기화된 상태이므로, 로 저장하고, 방에 입장할 때마다 캐릭터의 상태를 동기화 를 대비해서 3초에 한번씩 동기화를 해준다. 코드 추가



개발일지			
날짜	2016-12-30 작성시간 2016-12-30 9:15		
이름	임지훈		
	1. 합치기 작업.		
개발과제	2. 통합 디버깅. 3. Character 앉기 모션 위치 조정.		
	1. 합치기 작업. 개발한 내용 병합 및 재분배		
	2. 통합 디버깅. 개발된 위치까지의 디버깅 및 오류 조정		
	3. Character 앉기 모션 위치 조정. Character 앉기 모션 시 캐릭터의 위치가 변하는 현상 조정.		
내용			



개발일지				
날짜	2016-01-02 작성시간 2016-01-02 9:15			
이름	임지훈			
개발과제	1. 캐릭터 애니메이션 재작업. (앉기, 손들기) 2. 음성테스트 3. 결과 전적 저장확인 4. 추가 통합 5. 통합 디버깅 6. 발표 준비			
내용	1. 캐릭터 애니메이션 재작업. (앉기, 손들기) Blender 를 이용하여 애니메이션 위치 보정 후 재작업. 2. 음성테스트 음성이 깔끔하게 넘어가는지 확인 및 화면 UI와 연동. 3. 결과 전적 저장확인 DB에 전적 저장 결과 확인. 및 디버깅 수정. 4. 추가 통합 디버깅한 뒤 프로젝트 재통합. 5. 통합 디버깅 다시 한번 시뮬레이션 돌리면서 디버깅.			



개발일지				
날짜	2016-01-03 작성시간 2016-01-03 9:15			
이름	임지훈			
개발과제	 발표준비 통합 디버깅 비디오스크라이브 수정. 			
내용	1. 발표준비 2 차 발표 준비. 2. 통합 디버깅 프로젝트 시연 전에 다시 한번 테스트 및 디버깅. 3. 비디오스크라이브 수정. 비디오스크라이브로 자기발표부분 재조정.			



			개발일지	
날짜	2016-01-04 작성시간 2016-01-04 9:15			
이름	임지훈			
개발과제	1. 서버와 클라이언트 부분 디버깅.			
내용	IOCP 모델(하는 부분에, 원형 Queu 쓰레드가 다해 됐을 두 번 같은 패킷(부분에서 에는 따라서 검으로 수정. 이에 따라 현상은 사라	조 전을 대기를 지하는 다음을 다 보냈다. 지수 보생 전송하는 다음을 다면 전송하는 다면 전송하는 다면 전성 보생. 지수 지수를 다 지수를 다 보냈다. 지수 지수 지수를 다 지수를 다 되었다. 지수 지수 지수를 다 지수를 다 되었다.	WorkServer 디버깅. 및 문제 발견. end()의 Completion Queue 를 받았을 때, 검사를 하고 재전송 탱. Send 버퍼를 구성하여 보냈으나 Completion Queue 를 받은 서 보내기 요청을 한번 더한 소켓에 대해 검사를 할때 같은	



	개발일지			
날짜	2016-01-05 작성시간 2016-01-05 9:15			
이름	임지훈			
개발과제	1. 서버 재 디버깅 및 완성. 2. TrinusVR 병합 및 수정 보완.			
내용	1. 서버 재 디버깅 및 잔버그 잡고 완성. 2. TrinusVR 병합 및 수정 보완. 민수가 작업한 TrinusVR 병합 및 부족한 부분 수정 및 보완.			



	개발일지		
날짜	2016-01-06 작성시간 2016-01-06 9:15		
이름	임지훈		
개발과제	 클라이언트 재 통합. 통합 디버깅. 문서작성. 		
내용	1. 클라이언트 재 통합. 승리형이 완성한 Music 기능과 다국어 기능 추가 통합. 연결 안된 부분 및 오류 부분 수정 및 보완. 2. 통합 디버깅. 모든 프로젝트 통합 후, 버그 발견 및 버그 수정. 방 만들고 나와서 방 생성이 안됨. 방 들어갈 때 두 번 클릭해야 들어가짐. TrinusVR 등등. 3. 문서작성. 프로젝트 완료보고서 작성 및 보조.		



개발일지						
날짜	2016-01-09 작성시간 2016-01-09 9:15					
이름	임지훈					
개발과제	1. 클라이언트 디버깅. 2. Trinus VR 디버깅. 3. 2 차 발표 준비					
내용	1. 클라이언트 디버깅. 발표에 앞서 클라이언트 통합적인 디버깅. 및 잔 버그 수정. 2. Trinus VR 디버깅. Trinus VR 실행 시 화면 해상도 변하는 부분과 여러가지 필요한 사항 디버깅 후 수정. 3. 2 차 발표 준비 오후 마지막 발표를 위한 발표준비 및 연습.					



	개발일지							
날짜	2016-01-10 작성시간 2016-01-10 9:15							
이름	임지훈							
개발과제	1. 프로젝트 완료보고서 수정 및 완료. 2. 시연영상 제작. 3. 이력서 포트폴리오 작성. 4. 미흡한 각종 문저 작업.							
내용	1. 프로젝트 완료보고서 수정 및 완료. 프로젝트 완료보고서 중 개발탭의 소스코드 설명 추가. 2. 시연영상 제작. 팀원과 함께 마지막발표 시연시 잘못될 경우를 대비해 시연영상 제작. 3. 이력서 포트폴리오 작성. 개인 이력서 및 포트폴리오 작성. 4. 미흡한 각종 문저 작업.							



ii. 강승리

개발일지						
날짜	2016-11-21 작성시간 2016-11-21 AM 11:06					
이름	강승리					
개발과제	1. 팀명 결정2. 팀장, 서기 결정3. 프로젝트명 결정4.역할 분담5.프로젝트 아이디어 회의6.모델링 의논					
· · · · · · · · · · · · · · · · · · ·	아이디어 참고를 위한 스마트업 아이디어공모전 견학 1. 팀명 결정 - 팀명 후보 : 1)셋이서 한조					



	개발일지								
날짜	2016-11-22	작성시간	2016-11-22	PM 20:30					
이름	강승리								
개발과제	 기획서 회의 및 작성 모델링 테마 결정 모델링 방법, 디자인 결 	정							
내용	서버 아키텍처 회의 및 1차, 2차, 3차 에 따른 UI 순서도 회의 및 설 팀일정, 개인일정 회의 2. 국회, 도서관, 카페의	차 목표 회의 및 설정. 보설계 로 시스템 아키텍처 회의 및 설계 계, 각화면에 대한 UI 배치 구상화 및 작성							



개발일지							
날짜	2016-11-23	작성시간	2016-11-23	PM	20 : 06		
이름	강승리						
개발과제	모델링 설계 카페, 도서관 스테이지 모 모델링프로그램 스터디	델링 기본적인 뼈대 및 틀 완성					
내용		티로 넘길때 마테리얼 적용문제 발성 로그램및 유니티 마테리얼 공부.	생)				



	개발일지							
날짜	2016-11-24 PM 20:11							
이름	강승리							
개발과제	모델링 설계 카페, 도서관 스테이지 모델링 모델링프로그램 스터디							
내용	모델링 완성 마테리얼적용문제점 해결 (DAE 파일을 3DMAX 프로그램으로 륱확장자로 변경) 조명 및 내부구조 연출							



	개발일지							
날짜	2016-11-25	작성시간	2016-11-25	PM	20 : 10			
이름	강승리							
개발과제	모델링 설계 카페, 도서관 스테이지 완성 모델링프로그램 스터디							
내용	모델링 완성 조명 및 내부구조 완성 (조명이 화려하기때문에 무거워짐) 최적화 내용 찾기 최적화 완료 (라이트매핑을 황용하여 모델을 미리	리 베이크해서	서 최적화 완성)				



개발일지							
날짜	2016-11-25	작성시간	2016-11-25 PN	И 20:10			
이름	강승리	•					
개발과제	모델링 설계 카페, 도서관 스테이지 완성 3 번째 스테이지 국회 모델링						
내용	카페, 도서관 스테이지 완성 최적화 시도 3 번째 스테이지 국회 모델링 완성	(마테리얼 <i>, </i>	조명효과 미적용)				



	개발일지							
날짜	2016-11-29	작성시간	2016-11-29	PM	20 : 22			
이름	강승리							
개발과제	최적화 시도 (3 번째 스테이지 국회 모델링 완성)							
내용	도서관, 카페 , 국회 3 마테리얼 적용완료 조명효과 적용완료 국회 최적화시도 (컴	3개의 스테이지 완성 퓨터 연산이 느려 하루 통	통채로 빌드)					



	개발일지						
날짜	2016-11-30	작성시간	2016-11-30 PM 20:11				
이름	강승리						
개발과제	모데링된 스테이지 확인 스테이지 마무리작업 UI 작업 (대략적인 배치도 및 UX 작성및 구상)						
내용	모델링 완성 마테리얼적용문제 (DAE 파일을 3D 조명 및 내부구	MAX 프로그	L램으로 륱확장자로 변경)				



	개발일지								
날짜	2016-12-01	작성시간	2016-12-01	PM 20:	28				
이름	강승리								
개발과제	스테이지 조명효과 개선 및 변 UI 작업 (대략적인 배치도 및								
내용	스테이지 라이트매핑 완성 씬 이동시 스테이지 조명 효과 (조명 효과 변경 및 개수 조 UI 대략적인 UI 배치 구성 DB와 내용맞추기			∥해결)					



		개발일	실지		
날짜	2016-12-02	작성시간	2016-12-02	PM	20 : 22
이름	강승리				
개발과제	UI 배치 개선 내용추가 UX 제작, 스테이지 디버그및	문제점 체크	L		
내용	UI 배치 개선 완료 스테이지 문제점 체크 완료 등 로그인화면 회원가입 ux 완료 로비 ux (60% 완료)		지 약간 버벅임	있음	(학원 개발환경기준)



	개발일지				
날짜	2016-12-05	작성시간	2016-12-05	PM	20 : 39
이름	강승리				
개발과제	UX 작성 내용추가 UX 제작, 스테이지 디버그및	문제점 체 <i>=</i>	3		
내용	UI 배치 개선 완료 UI 및 UX 기능적인 부분만 구성 가완성 (DB와 연동은 아직 연결하지않은 상태) 로비 기능 완성 로그인씬 화면 개선 (씬 추가로드로 2 개의씬을 합쳐본결과 상당한 버벅거림발견) 카페 스테이지 활용하여 모델링 복사, 자전하는 큐브 만들어 카메라배치후 카페 스테이지기준으로 자전하는 배경완성, 터레인추가 하여 지형 구성				



	개발일지				
날짜	2016-12-06	작성시간	2016-12-06	PM	20:00
이름	강승리				
개발과제	로그인씬 , 회원가입 기능 완 UI 디자인 및 이미지 편집	성			
내용	로그인씬 , 회원가입 기능 코 UI 디자인 및 이미지 편집 포토샵을 활용하여 이미지 편 로그인, 회원가입, 로비 UI 및 [명일 부터 네트워크 공부]	!집 및 맞춤·			



	개발일지			
날짜	2016-12-07	작성시간 2016-12-07 PM 20:40		
이름	강승리			
개발과제	네트워크 스터디 [참고문헌 - 반다이남코 현역프로그래머가 알려주는 유니티 네트워크 프로그래밍] 1~2 장 이해하기 3 장 네트워크 TCP 예제 연습 4 장 채팅프로그램 제작			
내용	병가로 인한 조퇴			



개발일지			
날짜	2016-12-08 PM 20:38		
이름	강승리		
개발과제	네트워크 스터디 [참고문헌 - 반다이남코 현역프로그래머가 알려주는 유니티 네트워크 프로그래밍] 1~2 장 이해하기 3 장 네트워크 TCP 예제 연습 4 장 채팅프로그램 제작		
내용	네트 워크 스터디 TCP 네트워크 스터디및 네트워크 이해 TCP 서버 클라이언트 예제 작성 유니티환경에서 실행 테스트 로그인부분 연결 확인 세부조정및 로그인 , 회원가입 UI 병합(정확히는 씬을 하나로 다시만듬) 씬 이동시 서버 로그아웃했다가 다시 로그인하는모습을 확인 (해결방안이 없는지 다시확인)		



	개발일지				
날짜	2016-12-09	작성시간	2016-12-09	PM 20:38	
이름	강승리	강승리			
개발과제	네트워크 스터디 참고문헌 [c# 5.0 프로그래밍] TCP 네트워크부분 스터디 UX 스크립트수정 (1 개의 스크립트에 로비 , 로그인 , 회원가입씬의 정보가 다있었기때문에) 스크립트 분할 (로그인&회원가입 씬, 로비씬 따로 제작) 네트워크 스크립트 분리 (싱글톤이용) -UX 스크립트부분에 네트워크 코드들이 합쳐져있어 분리작업				
내용	네트워크 TCP 부분 스터디 UX 스크립트 분리 완료 [로그인 & 회원가입 , 로비씬] 및 조정 버튼 부분 디자인 수정 네트워크 스크립트 분리 (따로 네트워크 스크립트 제작 send, receive emd 부분적 함수화) - 서버에 접속은 되지만 데이터전송문제 발생 [싱글톤에 들어있던 스크립트의 Start 부분에 문제가 있었음] - 수정완료				



	개발일지			
날짜	2016-12-12	작성시간 2016-12-12 PM 20:14		
이름	강승리			
개발과제	네트워크 스크립트 분리 (싱글 -UX 스크립트부분에 네트워크 Ui 및 UX 수정	글톤이용) . 코드들이 합쳐져있어 분리작업		
내용	UX 스크립트 분리 수정 버튼 부분 디자인 수정 네트워크 스크립트 분리 (따로 네트워크 스크립트 제 - 서버에 접속은 되지만 데이	작 send, receive emd 부분적 함수화) 터전송문제 발생		



	개발일지				
날짜	2016-12-13	작성시간	2016-12-13	PM	20 : 40
이름	강승리				
개발과제	네트워크 스크립트 수정				
내용	로그인 부분에서 로비 넘어 데이터 저장및 이동 문제 빌 프리팹을 이용하여 문제 해 ²	·생			



	개발일지
날짜	2016-12-14 작성시간 2016-12-14 PM 21:14
이름	강승리
개발과제	랜섬웨어 피해 복구
내용	랜섬웨어발견 . 작업 올스톱 감염되지않은 파일 가려내어 백업



		개발일지
날짜	2016-12-15	작성시간 2016-12-09 PM 21:06
이름	강승리	
개발과제	랜섬웨어 피해 복구	
내용	백업해서 살린 데이터 적음 컴퓨터 포맷 개발 환경 다시 조정 유니티 설치 비주얼스튜디오	설치



		개발일지
날짜	2016-12-16	작성시간 2016-12-16 PM 20:55
이름	강승리	
개발과제	랜섬웨어 피해 복구 중간발표 준비	
내용	중간 발표용 ppt 작성 로그인 로비 UI, UX 복구 네트워크 부분적 미복구 중간발표.	



	개발일지			
날짜	2016-12-19	작성시간 2016-12-19 PM 20:40		
이름	강승리			
개발과제	랜섬웨어 피해 복구 완료 로비 네트워크 스크립트 수정 인게임 UI제작			
내용	로비부분 네트워크 스크립트 인게임 [스테이지] UI 디자인 스테이지 UI 기능 스크립트 제	민 및 버튼 기능 구성		



	개발일지			
날짜	2016-12-20	작성시간 2016-12-20 PM 21:02		
이름	강승리			
개발과제	스테이지 UI 기능 스크립트 원	악 성		
내용	UI 버튼 이미지 조정 및 기능· UI 스크립트 기능 완료 , 스크 스테이지 네트워크 스크립트	1립트 제작완료		



		개발임	일지		
날짜	2016-12-21	작성시간	2016-12-21	PM	20 : 40
이름	강승리				
개발과제	UI 네트워크 스크립트 제작				
내용	인게임 네트워크 스크립트 제 (아직 세부적인 디버깅 , 오루		음)		



	개발일지					
날짜	2016-12-22	작성시간 20	016-12-22	PM	20 : 18	
이름	강승리					
개발과제	UI 네트워크 스크립트 제작 (인게임 네트워크 스크립트	가완성)				
내용	인게임 네트워크 스크립트 가 (아직 세부적인 디버깅 , 오루)			



	개발일지						
날짜	2016-12-26	작성시간	2016-12-26	PM 21:	10		
이름	강승리						
개발과제	인게임 네트워크 스크립트 가	·완성					
내용	인게임 네트워크 스크립트 가세부적인 디버깅작업 실시, 오류수정완료 준비하기 버튼 활성화를 위한	오류 발견 (준비상태에서	준비해제	상태 바뀌는 오류)		



	개발일지						
날짜	2016-12-27	작성시간 2016-12-27 PM 21:28					
이름	강승리						
개발과제	인게임 네트워크 스크립트 가 일본어 버전 만들기	·완성					
내용	스테이지 구조 변경 준비상태 활성화를 위한 세부 로그인 , 회원가입 , 로비 씬의						



	개발일지						
날짜	2016-12-28	작성시간	2016-12-28	PM	21 : 12		
이름	강승리						
개발과제	인게임 네트워크 스크립트 수 일본어 버전 만들기	정					
내용	일본어 버전 만들기 준비상태 활성화 작업 실시 중 로비씬 로그인씬 일본어 작업						



	개발일지					
날짜	2016-12-29	작성시간 2016-12-29 PM 21:00				
이름	강승리					
개발과제	인게임 네트워크 스크립트 기일본어 버전 만들기 준비 활성화를 위한 조건 및					
내용	준비상태 활성화 완료 일본어 버전 만들기 완료 네트워크 스크립트 가완성 (다른 인원이 만든 것과 합쳐 봐야 함)				



	개발일지						
날짜	2016-12-30	작성시간	2016-12-30	PM 21:28			
이름	강승리						
개발과제	UI 백지화 고급스러운 느낌으 일본어 버전 만들기(UI에 맞						
내용	UI 이미지 만들기 버튼 및 UI 작업 구상 (골드톤의 고급스러운 분위기 버튼이미지 일본어 버젼 작성		결정)				



	개발일지						
날짜	2017.01.02	작성시간	2017-01-02	PM	21 : 13		
이름	강승리						
개발과제	Ui 작업 완료 일본어 스크립트 완료						
내용	일본어 스크립트 수정 및 완료 (이슈부분 및 자잘한 곳까지 UI개편 완료						



	개발일지					
날짜	2017-01-03	작성시간 2017-01-03 PM 18:05				
이름	강승리					
개발과제	2 차발표 준비 프리젠테이션 준비 (비디오스크라이브를 이용한 시연 영상 촬영	동영상 프리젠테이션)				
내용	비디오스크라이브 를 이용한 (문자를 이미지화 시켜 작성 시연영상 촬영 2 차발표					



		개발일지
날짜	2017-01-04	작성시간 2017-01-04 PM 21:08
이름	강승리	
개발과제	2차 발표 후 부족했던 부분통합 디버깅	및 앞으로 더 해야 할 작업들 논의
내용	- 처음 시작하기에 앞서 퍼포 보여줌] 통합디버깅 - VR 모드 중 시점이 하늘을 - 다국어 모드중 결여된 부분 언어에 따른 배경 음악필요성	및 수정해야 할 부분 발견 [오타 및 불필요한 부분제거



	개발일지					
날짜	2017-01-05 작성시간 2017-01-05 PM 19 : 45					
이름	강승리					
개발과제	통합디버깅 배경음악 편집 및 효과음 편집 [음향 증폭 , 노이즈 제거 , 구간편집] 최종발표 퍼포먼스 구체적 방안 논의 다국어모드 가완성					
내용	최종발표 퍼포먼스 [각국의 정상들이 모여 북 핵 문제를 논의 하는데 장소의 제한을 해학적으로 표현] - 처음 시작하기에 앞서 퍼포먼스 부족 [3~4분 짜리 간이 연극으로 토대리의 필요성을 보여줌] UI 편집 - UI 이미지 편집 및 기능 정리 [음악관련 UI 추가] 다국어 모드 & 인게임 [부분 수정 및 인게임 부분 영어 버젼에서 음악 재생 및 선택 부분 실험 .(성공)] 통합디버깅 - VR 모드중 시점이 하늘을 향해 시작이 되는 부분을 발견 [수정 중]					



개발일지					
날짜	2017-01-06 작성시간 2017-01-06 PM 17 : 45				
이름	강승리				
	통합디버깅				
개발과제	최종발표 퍼포먼스 구체적 방안 논의				
	다국어모드 완성				
내용	다국어 모드 완성 최종발표 퍼포먼스 [각국의 정상들이 모여 북 핵 문제를 논의 하는데 장소의 해학적으로 표현] - 처음 시작하기에 앞서 퍼포먼스 부족 [3~4 분 짜리 간이 연극으로 토대리으보여줌] - 시나리오완성 (처음 퍼포먼스부분) - 시연 시나리오 구상 중 UI 편집 - UI 이미지 편집 및 기능 정리 [음악관연 UI 추가] 다국어 모드 & 인게임 [다른 언어의 캔버스에 통합 중에 함수연결 문제 발상통합디버깅 - VR 모드 중 시점이 하늘을 향해 시작이 되는 부분을 발견 [해결 & 빌드 (통구상 중)]	필요성을			



		개발일지
날짜	2017-01-09	작성시간 2017-01-09 PM 21 : 45
이름	강승리	•
개발과제	통합디버깅 최종발표 퍼포먼스 구체적 방 발표 시나리오 완성 시연영상 촬영 및 자막 삽입	
내용		인부분에 가면 버벅이는 현상 발견] 네 끼임현상 발견 [캐릭터 매쉬 삭제로 해결] 집 수정 [자막까지 완료]



		개발일지
날짜	2017-01-10	작성시간 2017-01-10 PM 14 : 05
이름	강승리	
개발과제	시연영상 재촬영 및 자막 삽원 최종보고서 작성 , 수정 최종발표때까지 스케쥴 및 발	
내용	시연영상 재촬영 , 편집 최종보고서 수정및 작성 최종발표시 발표 방안 재설정	d[퍼포먼스 방법, 발표의 시간, 발표 진행 순서등]



iii. 손민수

개발일지		
날짜	2016-11-22 작성시간 2016-11-22 11:11	
이름	손민수	
개발과제	1. 팀명 결정 2. 팀장, 서기 결정 3. 프로젝트명 결정 4.역할 분담	
내용	1. 팀명 결정 - 팀명 후보:1)셋이서 한조 2)보온병 3)필라맨트 4)OOO 제작위원회 최종결정: 토대리 제작위원회 2. 팀장 결정 팀장:임지훈 서기: 손민수 3. 프로젝트명 결정 프로젝트명 결정 프로젝트 명: 토대리(VR 토론 시스템) 4. 역할 분담 강승리:UI, 모델링 임지훈: Network, Database	



개발일지			
날짜	2016-11-23	작성시간	2016-11-23 11:05
이름	손민수		
개발과제	Voice Network 자료조사 및	Example Tes	st
내용	일반 Chat 에 관한 자료는 [이용한 방식이나 Unet 을 이용	많았으나 Vo 룡한 방식밖 erver 와 Clie	및 Example 을 Test 했습니다. ice Chat 에 관한 자료는 유료 에셋을 에 없어서 일단 Unet 을 이용한 방식의 nt 는 돌아가나 Voice 가 넘어가지 않아



		날일지	
날짜	2016-11-24	작성시간	2016-11-24 10:12
이름	손민수	•	
개발과제	Voice Network C#		
내용	일반 Chat 에 관한 자료는 이용한 방식이나 Unet 을 이	많았으나 Vo 용한 방식빆 erver 와 Clie	및 Example 을 Test 했습니다. Dice Chat 에 관한 자료는 유료 에셋을 함에 없어서 일단 Unet 을 이용한 방식의 ent 는 돌아가나 Voice 가 넘어가지 않아



	개빝	날일지	
날짜	2016-11-25	작성시간	2016-11-25 10:20
이름	손민수		
개발과제	Voice Network C#		
내용	문제인지는 모르겠으나 음질 매우 안좋았습니다. Photon 사람들의 댓글을 보면 성능이 그리고 DircectSound 기반음질은 마찬가지로 안좋았으구현 할 수 있는 확률을 높	이 Direct S Cloud 기년 괜찮다고 여 G711Audio 보나 외부에서 하는 예제인	mple 을 테스트 하였습니다. 기기의 ound 나 Unet 을 이용한 음질과 같이 반의 Voice Chat Example 을 사용한 하는데 저음질의 원인을 못 찾았습니다. 사용 예제를 상세 분석 하였습니다.서 지원하는 서버가 아닌 직접 서버를 만큼 분석하였습니다. 서버에서는 별도에서 보내준 데이터를 다른 Client 들에게



	개'	발일지	
날짜	2016-11-28	작성시간	2016-11-28 10:05
이름	손민수		
개발과제	Voice Network C#		
내용	sound 등) 사용하는 방법: 내용은 없었으나 외부 dll	을 조사 했습 을 사용할	버구축을 하고 Unity 에서 외부 dll(Direct 습니다. 조사 결과 Direct sound 에 관한 순 있지만 모든 dll 들이 돌아가는 것은 Step 으로 시간 내 최상의 결과를 낼 수



	개발일지
날짜	2016-11-29 작성시간 2016-11-29 9:13
이름	손민수
개발과제	Voice Network C#
내용	Unity 에서 DirectX, Direct Sound 기반으로 Client 개발을 진행했습니다. 애석하게도 Unity 에서는 DirectX 와 그 내부 기능들을 사용 못하는 환경이어서 클라이언트 전면 수정에 착수했습니다. Unet 기반으로 개발된 클라이언트를 수정하여 C# Server 와 통신을 시도해봤습니다만 연결이 되지 않아 여러 방면으로 수정 -> 테스트 했습니다.



	개발일지
날짜	2016-11-30 작성시간 2016-11-30 10:01
이름	손민수
개발과제	Voice Network C#
내용	C# Tcp Server 와 Unity Client 간 소켓 통신을 시도하였습니다. 소켓 통신 자체는 연결이 잘 되었으나 Client 에서 Server 로 데이터를 넘길 때 Server 에서 데이터 자체가 안들어왔습니다. Client 에서 데이터를 전송을 못하는 지, Client 에서 Server 로 데이터는 보내나 받지를 못하는 지 원인 분석을 하였습니다.



	개발일지
날짜	2016-12-01 작성시간 2016-12-01 10:20
이름	손민수
개발과제	Voice Network C#
내용	여러 방법의 Voice Chat 을 Test 했으나 결과가 영 좋지 못했습니다. Server 와 Client 통신할 때 음성을 Bytestream 에 맞춰서 다시 Test 했지만 음성 값이 클라이언트에서 서버로 넘어가지를 못했습니다. 지속적인 자료조사와 분석을 하면서 계속 진행하겠습니다.



	개발일지
날짜	2016-12-02 작성시간 2016-12-02 10:43
이름	손민수
개발과제	Voice Network C#
내용	강사님께 37 기 프로젝트 소스예제를 받아서 음성을 어떻게 처리하였고 전송했는지 분석을 하려고 했으나 37 기 결과보고서에 적혀있던 부분이 없어서 음성부분만 지웠다라는 결론을 내고 빠르게 타 예제 등 분석을 시도했습니다. 단순히 Capy & paste 만 해서는 안되기에 분석을 하고 program 의 흐름을 쭉 따라가봤는데 예제들의 코드들이 심오하게 얽혀있어 어려움에 Open Source 사용 방법이 전부 영어여서 영어공부에 대한 필요성을 한번 더 느꼈습니다.



	개발일지
날짜	2016-12-05 작성시간 2016-12-05 10:01
이름	손민수
개발과제	Voice Network C#
내용	금요일 및 주말 간 음성을 Bytestream 으로 변환하여 전송하는 작업을 시도했습니다. 변환은 되었으나 전송과정에서 코드 문제인지 넘어가지 않아 많은 애로사항을 겪었습니다. 많은 예제를 보면서 공부 했으나 코드 자체가 난잡하게 되어있거나 코덱을 사용하여 처리하는 코드가 많았습니다. 강사님께서 code project 예제 몇 개를 보내주셨습니다만 실행결과 작동이 다안되어서 매우 안타까웠습니다. 일단 UDP 연결 및 값 send, recieve 부분을 재작성하고 테스팅 할 예정입니다.



		7U H F O I	TI
		개발일	^
날짜	2016-12-06	작성시간	2016-12-06 10:09
이름	손민수		
개발과제	Voice Network	C#	
내용	적용 중에 있습 [®] 것으로 보고 있습	니다. 데모의 습니다. 어제 지훈이형이	11 Codec 을 이용한 C# 데모를 유니티에 흐름은 파악 하였기에 작업이 수월 할 강사님께서 음성 파트를 혼자 말고 같이 같이 투입 하였습니다. 생각보다 빨리 진행



개발일지					
날짜	2016-12-07 작성시간 2016-12-07 10:27				
이름	손민수				
개발과제	Voice Network C#				
내용	NAudio.dll 을 이용하여 G7211 Codec 을 이용한 UDP 음성 채팅을 구현하고 있습니다. Naudio 공식 홈페이지에 있는 dll 버전과 Unity 와 호환이 안되어 웹 검색 하여 호환 되는 dll 을 받아서 작성 하였습니다. Unity 에 맞게 적용시키고 디버깅 해본 결과 WaveOut.Play()와 waveIn.StartRecording() 부분에서 알 수 없는 오류가 발생하였습니다. 원인을 못찾고 살짝 수정 하고 실행 했을 땐 Awake 와 start 가 두 번 도는 것을 확인하게 되었습니다. 그로 인해서 소켓이 두 번 사용하게 되고 프로그램이 멈추게 되었습니다. 디버깅 시에는 종료되지 않고 데이터가 Codec 으로 들어가 연산이 되는 것을 확인 하였습니다.				



개발일지					
날짜	2016-12-08 작성시간 2016-12-08 9:29				
이름	손민수				
개발과제	Voice Network C#, GateWay Server				
내용	TCP 기반의 음성 채팅 클라이언트를 UDP 로 변경 후 UDP 서버를 구축하여 송수신 하였습니다. UDP 서버를 잘못 짰는지 Send 와 Receive 는 되는 것을 확인 하였지만 음성으로 변환되지 않았습니다. 예상으로는 각 클라이언트에서 한번에 보내기 때문에 충돌이 일어난 듯 싶습니다. 한편 Gateway Server 구축을 착수 했습니다. Packet 에 대해 무지하였기 때문에 Packet 에 대해 공부하였고 이전 mini, semi project 에 적용 시켰던 것들을 보며 공부 했습니다.				



개발일지					
날짜	2016-12-09	작성시간	2016-12-09 9:50		
이름	손민수				
개발과제	Voice Network	C#, GateWa	ay Server		
내용	송수신 하였습니 확인 하였지만 을 한번에 보내기 구축을 착수 했습	다. UDP 서 음성으로 변 때문에 충¦ 습니다. Pac	아이언트를 UDP 로 변경 후 UDP 서버를 구축하여 버를 잘못 짰는지 Send 와 Receive 는 되는 것을 환되지 않았습니다. 예상으로는 각 클라이언트에서 돌이 일어난 듯 싶습니다. 한편 Gateway Server eket 에 대해 무지하였기 때문에 Packet 에 대해 project에 적용 시켰던 것들을 보며 공부했습니다.		



개발일지				
날짜	2016-12-12 작성시간 2016-12-12 8:31			
이름	손민수			
개발과제	Voice Network C#, GateWay Server			
내용	IOCP 기반의 Voice Server 를 구축하였습니다. 아직 패킷(해더 + 데이터)는 미 구축 상태입니다. 통신할 때 한 개의 클라이언트가 음성 데이터를 보내면 수신하는 클라이언트들은 마이크를 끈 상태에만 잡음 없이 음성이 깔끔하게 나왔습니다. 다만 청자 pc A 와 B 라고 했을 때 화자 pc 에서 이야기를 하면 약 0.5 초 뒤에 A pc 에 들리고 A pc 에서 들린 후 1 초정도 뒤에 B pc 에서 들렸습니다. 16 일 1 차 발표를 앞두고 있어서 음성 부분은 잠시 미루고 Join 부분의 사진을 전송하여 Disk 에 저장하고 해당 저장 경로를 Database 에 저장하는 부분을 진행 할 것입니다.			



개발일지				
날짜	2016-12-13	작성시간	2016-12-13 8:41	
이름	손민수			
개발과제	Client Lobby Net	work		
내용	진행하였습니다. Join Part 에서 ³ 저장하고 넘겨주· 전송되는 우려가 소속을 보내주고	회원가입 시 는 기능입니 있어서 On Server 에서 Lobby 에	/oice Chat 을 잠시 뒤로 미루고 Client Lobby 작업을 프로필 사진을 등록하게 되는데 사진 파일을 Server 에 다. Text 와 한번에 보내주면 좋지만 실패 할 경우 사진이 ne Button One Click 으로 초기 ID, Password, Nick Name, 더 Header 를 보내주면 성공 값이면 바로 사진이 전송되게 서 MyInfo 에 내 사진을 Server 에서 수신하여 받아오는	



			개발일지
날짜	2016-12-14	작성시간	2016-12-14 8:41
이름	손민수		
개발과제	Client Lobby Ne	twork	
내용	폴더에 랜섬웨어	l 감염된 시 고 본 proje	완료 후 친구 정보에 이미지 작업 도중 개인 PC 공유 나실을 파악하여 포맷을 진행 했습니다. Voice Chat Test ect 는 감염이 안된 것으로 파악되어 나머지 Main Project



			개발일지
날짜	2016-12-15	작성시간	2016-12-15 9:22
이름	손민수		
개발과제	Client Stage Ne	twork	
내용	Stage UI 를 팀 추가했습니다.	원(강승리)고	과 작업하였습니다. 각종 버튼과 필요한 기능에 대한 UI 를



개발일지				
날짜	2016-12-16 작성시간 2016-12-16 9:08			
이름	손민수			
개발과제	1 차 발표 준비			
내용	16(금)에 있을 1 차 발표 자료를 만들었습니다. Stage 의 UI 배치를 수정 하기로 회으때 결정이 되어 발표 자료 만들기 전에 필요한 UI를 제작하였습니다.	븨		



개발일지					
날짜	2016-12-19 작성시간 2016-12-19 9:30				
이름	손민수				
개발과제	Stage UI				
내용	주말간 Stage 에 필요로한 UI 작업을 했습니다. 이사 준비하느라 다소 작업 한 디자인이 마음에 안드는 부분이 있었으나 UX 작업 하면서 수정해 나가는 방향으로 진행 할 예정입니다.				



개발일지					
날짜	2016-12-20 작성시간 2016-12-20 9:22				
이름	손민수				
개발과제	Stage UI				
내용	Stage UI 및 UX 작업을 계속 진행 하였습니다. 오늘 방 계약 및 이사로 인해서 많은 작업은 못할 듯 싶으나 할 수 있는 작업은 최대한 진행하겠습니다.				



개발일지					
날짜	2016-12-21	작성시간	2016-12-21 10:17		
이름	손민수				
개발과제	Stage UX, Net	work			
내용	WindowForm 5	으로 수정하	ditorWindow 로 개발되어 있는데 빌드 시 오류가 생겨 计였습니다. 이사 후 몸이 안좋아서 일찍 들어갔기에 Openfile 만 e Network 부분을 개발할 예정입니다.		



			개	발일	지				
날짜	2016-12-22	작성시간	2016-12	-22 9:58					
이름	손민수								
개발과제	Stage UX, Ne	twork							
내용	Stage 내부 왼쪽자리로 ^을 내부 투표 기	음겼습니다.					위해	강승리	팀원



	개발일지							
날짜	2016-12-23 작성시간 2016-12-23 10:40							
이름	손민수							
개발과제	Stage UX, Network							
내용	강승리 팀원이 작성한 code를 설명 받고 분석하여 그 style에 맞게 코드를 수정하였고 계속 진행중에 있습니다. 오늘(23 일) 강승리 팀원의 부재(사유 : 독감판정)로 인해 팀장의 지시 받아 지시에 따라 진행 하도록 하겠습니다.							



			개肯	발일지							
날짜	2016-12-26	작성/	시간 201	6-12-26 10:24							
이름	손민수										
개발과제	Stage UI, Network										
내용	짜잘한 버그 처리히 예정입니다.	·였고	합치기까지	하였습니다.	음성	Button	작업	및	통합	Debug	할



개발일지								
날짜	2016-12-27	작성시간	2016-12-27 9:44					
이름	손민수							
개발과제	Stage UI, UX, Netw	ork						
내용	문제점으로는 myin 아니라 sin(프로젝트 미리 만들어진 R	fo, friend in E test 할 [oom 에 한 Characte	빼서 통합 Testing 후 몇 개의 문제점을 발견하였습니다. Info 의 image 저장 경로가 실행파일 위치에서 생기는 것이 때 지정한 경로)위치로 저장되는 문제점과 Testing 용으로 접속 할 시 Client 가 뻗어버리는 현상이 발견되어 er 위에 Cube(발언권 요청 시 Host 가 발언권 권한을 줄 수 중입니다.					



	개발일지								
날짜	2016-12-28	작성시간	2016-12-28 10:00						
이름	손민수								
개발과제	Stage UI, UX, Netw	ork							
내용	성공하였습니다. 발	언권 부여 E utton 위에	성하고 각 Button 의 parant ID 값을 받아오는 데 Button 의 위치를 수정하여 옆에 Mute Button 을 생성하고 profile Image 를 띄울 예정입니다. 완료 후 합치기 작업을						



	개발일지								
날짜	2016-12-29 작성시간 2016-12-29 9:10								
이름	손민수								
개발과제	Stage UI, UX, Trinus VR								
내용	발언권 부여 및 mute Button(Cube)기능을 추가하였고 Instage profile image 를 cube 에 띄우는 시도를 하였습니다. 오늘 cube 에 image 를 띄우고 통합 작업 및 Trinus VR 연동할 예정입니다.								



	개발일지
날짜	2016-12-30 작성시간 2016-12-30 10:10
이름	손민수
개발과제	Stage UI, UX, Trinus VR
내용	cube 에 image 를 띄우고 통합 작업 및 Trinus VR 연동을 실시 하였습니다. Trinus VR 에서 회전각도 일부 고정을 시도할 예정이며 UI 수정이 필요로한 부분을 고칠 예정입니다.



개발일지								
날짜	2017-01-02	작성시간	2017-01-02 10:39					
이름	손민수							
개발과제	Stage UI, 2 차 발:	표준비						
내용	회전각도 제한시	키는 부분을	부분 수정(외국어 Part와 일부 버튼 디자인)하였고 Trinus VR 을 연구하였으나 실패하였습니다. 또한 주말간 발표 준비를 부분 수정 및 발표 자료 제작 예정입니다.					



	개발일지								
날짜	2017-01-03 작성시간 2017-01-03 10:22								
이름	손민수								
개발과제	2 차 발표준비								
내용	MS PowerPoint 와 Video Scribe 를 이용한 2 차 발표 준비를 진행하였습니다. 또한 발표 시간이 15:00 으로 수정되었기에 만든 발표 자료에 자료를 조금 첨삭 하여 발표 준비를 진행 할 예정입니다. 시연준비도 할 예정입니다.								



	개발일지
날짜	2017-01-04 작성시간 2017-01-04 10:40
이름	손민수
개발과제	Trinus VR
내용	어제 2 차 발표 준비 및 발표하였습니다. 오늘은 Trinus VR Headtracking 및 Click Event 를 할 예정입니다.



	개발일지							
날짜	2017-01-05	작성시간	2017-01-05 9:25					
이름	손민수							
개발과제	Trinus VR, 보고	1서 작성						
내용	띄우기.	Pointer 고경	을 진행하였습니다. 1 차적으로 Mouse Pointer 를 Application 에 정. 3 차로는 고정된 Pointer 가 Button 에 3 초간 호버되면 클릭 정입니다.					



フル 出FO! エ!								
			개발일지					
날짜	2017-01-06	작성시간	2017-01-06 9:42					
이름	손민수							
개발과제	보고서 작성							
내용			1 차 제출인 프로젝트 00에 제출 예정입니다.	: 완료	보고서를	어제와	마찬가지로	



	개발일지		
날짜	2017-01-09 작성시간 2017-01-09 9:32		
이름	<u> </u>		
개발과제	보고서 작성, 발표준비		
내용	2 차 발표 자료를 토대로 첨삭을 하며 주말 간 수정을 하였고 오늘도 최종 수정 예정입니다. 또한 결과 보고서 수정 및 추가 해야할 부분을 작성할 예정입니다. 시연에 앞서 Program 이 안정적인지 버그는 없는지 통합 디버깅도 할 것입니다.		



	개발일지		
날짜	2017-01-10 작성시간 2017-01-10 9:10		
이름	손민수		
개발과제	보고서 작성, 발표준비		
내용	2 차 발표 자료를 토대로 첨삭을 하며 주말 간 수정을 하였고 오늘도 최종 수정 예정입니다. 또한 결과 보고서 수정 및 추가 해야할 부분을 작성할 예정입니다. 시연에 앞서 Program 이 안정적인지 버그는 없는지 통합 디버깅도 할 것입니다.		



B. 팀 일일 진행

	2016 년 11 월 4 주차(11 월 21 일 ~ 11 월 25 일)			
	11 월 21 일 월요일			
담당자	업무 이름	업무 내용		
		1. 팀명 결정		
		2. 팀장, 서기 결정		
고트		3. 프로젝트명 결정		
공통		4.역할 분담		
		5.프로젝트 아이디어 회의		
		6.모델링 의논		
임지훈				
강승리				
손민수				
	11 월 22 일 화요일			
공통		1. 기획서 회의 및 작성		
임지훈	Database 환경설정	1. MySQL 설치 및 연동		
강승리	UI Modeling	1. 모델링 테마 결정		
004	of Modelling	2. 모델링 방법, 디자인 결정		
소민수	Voice Network	1. Voice Network 자료조사 및		
	Voice Network	Example Test		
	11 월 23 일 수요일			
공통		1. 기능정의서 작성		
		1. Database table 설계		
임지훈	Database Server 설계	2. table 정의서 작성		
		3. 포로토콜 설계 가이드라인 작성		
		4. Software UI 구조설계서 작성		
강승리	Stage UI Modeling	1. stage 모델링 설계 완성		



손민수	Voice Network	1. voice chat관련 자료 조사 및example 구동 및 분석2. 작동방안 구상(Example 구동 결과 :음성은 넘어가나 잡음 및 증폭이 심함)
	11 월 24 일 목요일	!
공통		1. 팀 내 회의 및 기술결정
		1. DB 구축
임지훈	Database Server 개발 착수	2. C++ 과 MySQL 의 연동
		3. DB 서버 구축 시작
강승리	Stage UI	1. stage(도서관) 완성
004	Stage Oi	2. stage(cafe) 시작(30%)
		1. Photon Cloud 기반 Voice Chat
소미스	Voice Chat Example 분석	Example Test => 타 예제와 음질 같음
	voice chat Example E ¬	2. DirectSound 기반 G711Audio 사용
		예제 상세 분석
	11 월 25 일 금요일	
공통		1. 팀 내 회의 및 기술결정
		1. DB 구축(IOCP Model 을 이용하여
임지훈	Database Server 구축	DB 서버 구축)
		2. DB 서버에 적용할 Query 문 작성
강승리	Ctoro III	1. Stage(Café) 모델링 완료 (성능
904	Stage UI	향상을 위한 최적화 까지 완료)
손민수	Voice Chat Network	1. 음성통신의 저품질 음질 해결
	2016 년 11 월 5 주차(11 월 28 일	~ 12 월 02 일)
11 월 28 일 월요일		
공통		1. 팀 내 회의 및 기술결정
	Database Comiser 75	1. Database Server 완료
임지훈	Database Server 구축	2. Unity UI 와 통신 완료
	Client UI	3. UI 작업 진행
강승리	Stage UI	1. Stage(국회) 90%완료



손민수	Voice Chat Network	1. Direct Sound 기반 클라이언트 개발 실패(사유 : Unity 에서 지원 안함) Unet 기반 클라이언트 응용 진행	
	11 월 29 일 화요일	<u>!</u>	
공통		1. 필요성 부족과 시간 부족으로 Github를 이용한 코드공유 프로젝트 적용 실패	
임지훈	Client, Server 간 Socket Network	1. WorkServer 와 Unity 통신 테스트 및 구상 2. Unity Animator 구상	
강승리	Stage UI	1. Stage(국회)완료 2. 모든 Stage 최적화 완료	
손민수	Voice Chat Network	1. Client <-> Server 통신 =>결과 : 소켓통신은 되나 음성 데이터 안넘어감	
	11 월 30 일 수요일		
공통		1. 팀 내 회의 및 기술결정	
임지훈	Client, Server 간 Socket Network	1. Unity Character 와 WorkServer 의 통신 테스트 마무리 2. 다중접속을 위한 WorkServer 구축	
강승리	Stage UI	1. Stage Baking 진행 2. Stage 최적화 정기적으로 진행	
손민수	Voice Chat Network	1. Voice Client <-> Server 통신테스트 -> 소켓 통신만 되고 음성처리 값 전달 X 원인 분석	
	12월 01일 목요일		



공통		1. 회원가입시 표시할 UI 종류와 DB 칼럼 비교 2. 방생성시 표시할 UI 종류와 DB 칼럼 비교 3. 로비에서 표시할 UI 종류와 DB 칼럼 비교
임지훈	WorkServer 구축 Client Animation 과 Server Network	1. 다중접속을 위한 WorkServer 구축 완료 2. 케릭터와 WorkServer 와의 통신 및 패킷생성 3. 다중접속된 각 클라이언트와 WorkServer 와의 통신
강승리	Client UI 및 UX	1. 로그인창 UI 작업과 기본적인 UX 틀 구축 2. 회원가입창 UI 작업과 기본적인 UX 틀 구축 3. 로비 UI 작업과 기본적인 UX 틀 구축
손민수	Voice Chat Example 확인	1. 37 기 Example 분석 -> Voice 부분만 없음. UDP Client, Server 통신 확인
	12 월 02 일 금요일	
공통		1. 팀 내 회의 및 기술결정
임지훈	Character Network	1. 캐릭터 이동 동기화
강승리	Client UI 및 UX	1. UI 틀 제작 완료 2. UX 60%완료(로그인, 로비, 회원가입)
손민수	Voice Chat Network	1. 아날로그 음성 Bytestream 변환 성공 2. UDP 환경 전송 테스트
2016 년 12 월 1 주차(12 월 05 일 ~ 12 월 09 일)		
	12 월 05 일 월요일	
공통		1. 팀 내 회의 및 기술결정



임지훈	Character Animation Network Voice Chat Network	1. 캐릭터 위치 동기화(진행-> PM 미팅후 보류) 2. 애니메이션 클립 받음 3. Voice Chat Part 투입 1. UI, UX 일정 부분 완료(Database
강승리	Client UI 및 UX	연동 X), 씬병합 -> 엄청 버벅거림 => 라이트 지우고 새로 병합 -> 버벅임 감소
손민수	Voice Chat Network	1. Voice Chat 구현 및 테스트 -> 실패, NAudio 이용 Voice Chat 구현 진행중
	12 월 06 일 화요일	
공통		1. 팀 내 회의 및 기술결정
임지훈	Voice Chat Network	1. 음성통신 협업
강승리	Client UI 및 UX	1. UI, UX 완료 (방 번호 부여 어려움 -> Server 에서 부여하기로 결정)
손민수	Voice Chat Network	1. G722 Codec 기반 음성채팅 구현 진행 중(Awake, start 두 번 돌아서 수정요망)
	12 월 07 일 수요일	!
공통		1. 팀 내 회의 및 기술결정
임지훈	Voice Chat Network Character Animation Network	1. 음성통신 협업 2. 캐릭터 위치 동기화 재시작
강승리		1. 아침부터 심한 두통으로 조퇴
손민수	Voice Chat Network Gateway Server 구축	1. TCP 기반 음성 채팅 -> UDP 변경 후 테스트 2. Gateway Server 개발 착수
	12 월 08 일 목요일	
공통		1. 팀 내 회의 및 기술결정
임지훈	Character Animation Network	1. 애니메이션 통신



		2. 캐릭터간 좌표이동 통신
강승리	Client, Login Server 간 Network	1. Network -> login Server 연동
손민수	Voice Chat Gateway Server 구축	1. TCP Server 2 개 구축(Voice Gateway) 문제점: 서버 1: Client 에서 Receive 하지 못함 서버 2: Send 못함, Receive 잘 못함
	12 월 09 일 금요일	1
공통		1. 팀 내 회의 및 기술결정
임지훈	Animation Network Voice Chat Network 검토	1. Stage 애니메이션 통신 2. 음성 Part 는 일부 수정 후 사용 가능성 파악
강승리	Client, Server 간 Network	1. Network packet 을 포함한 Data 전송 (단 Database Sever 제외)
손민수	Voice Chat Server 구축	1. IOCP Server 구축, 음성 Data 통신(문제점 : 속도, 음질)
	2016 년 12 월 2 주차(12 월 12 일	~ 12 월 16 일)
	12 월 12 일 월요일	
공통		1. 팀 내 회의 및 기술결정
임지훈	Client, Server 간 통신 Server 점검	1. 각 클라이언트 필요 정보 전달 2. 서버 보완
강승리	Lobby UI Client, Work Server 간 통신	1. Lobby 친구목록 2. 방만들기 Network 진행중
손민수	Join Scene, Login Server 간 통신	1. 회원가입 Network 보완 및 프로필 사진 전송 2. 내정보 사진 정보 진행중



	12 월 13 일 화요일		
공통		1. 팀 내 회의 및 기술결정	
		1. 클라이언트에 프로필 사진	
임지훈	Server -> Client File send	전송(본인, 친구)	
		2. 서버 보완	
강승리	Client -> Server Network	1. 방생성, 친구목록 Network 작업 계속 진행	
손민수	Client 기능	1. 개인 프로필 사진 출력	
	12 월 14 일 수요일		
공통		1. 어제 랜섬웨어로 인한 고통	
임지훈	1 차 통합	1. 로비까지 코드 및 디자인 합치기	
강승리	Stage UI	1. 팀원(손민수)와 Stage UI 작업	
손민수	Stage UI	1. 팀원(강승리)와 Stage UI 작업	
	12 월 15 일 목요일		
공통		1. 팀 내 회의 및 기술결정	
		1. 클라이언트 방 출력	
임지훈	Client 기능	2. 방정보 마우스 호버 시 나오게 적용	
	Client design	3. 로비까지 코드 및 디자인 수정 및	
		보완	
강승리	Stage UX	1. Stage UX 작업	
손민수	Stage UX	1. Stage UI 수정 및 보완	
	12 월 16 일 금요일	[
공통		1. 발표 준비	
임지훈	2 차 통합	1. 로비와 씬 코드 통합 및 디버깅.	
ㅁ끼푼	Character Animation	2. 애니메이션 작업.	
강승리	Stage UX	1. Stage UX 작업	
손민수	Stage UX	1. Stage UI 작업	
	2016 년 12 월 4 주차(12 월 19 일 ~ 12 월 23 일)		



	12 월 19 일 월요일		
공통		1. 팀 내 회의 및 기술결정	
이지층	Character Animation	1. 애니메이션 작업.	
임지훈	Client Animation, Server 간 통신	2. 서버와의 통신 베이스	
강승리	UX Network	1. UX, Network 작업	
손민수	Client UI, UX	1. UI,UX 작업	
	12 월 20 일 화요일	<u> </u>	
공통		1. 팀 내 회의 및 기술결정	
이지층	Server, Client Animation 간 통신	1. 서버와의 애니메이션 통신.	
임지훈	Server 안정화 작업	2. 서버와의 위치 통신 및 보정.	
강승리	Client UI	1. UI 완료, Network 작업	
004	Client Network	T. Of E.m., Network 7 a	
소민수	 Client 기능 수정	1. Openfile 오류 부분 수정	
	enent 10 10	2. 오후 이사	
	12 월 21 일 수요일		
공통		1. 팀 내 회의 및 기술결정	
	Client, Server 간 통신 규약 조정	1. 서버와 클라이언트간의 키 방식	
임지훈	Client Character 위치 동기화	통신 및 위치정보 보정 추가	
	Voice Chat Network	2. 보이스 통신 작업 진행	
강승리	Stage Network	1. Stage Network 작업(거시적 ->	
00-	Stage Network	미시적)	
손민수	Stage Network	1. Stage Network 작업 및 UI 다듬기	
	Client UI	2. 저녁 독감으로 조퇴	
	12 월 22 일 목요일		
공통		1. 팀 내 회의 및 기술결정	
	Voice Chat Network	1. 멀티 음성통신	
임지훈	3 차 통합	2. 프로젝트 통합 및 디버깅	
	Client 기능(투표) Server 작업	3. 투표 기능 서버작업	
강승리	Client 기능(투표), Server 통신	1. 투표 기능 Network 통신 작업	
손민수	Client 기능(투표), Server 통신	1. 투표 기능 Network 통신 작업	



12 월 23 일 금요일		
공통		1. 프로젝트 통합 및 디버깅.
임지훈	4 차 통합 통합 디버깅	1. 프로젝트 통합 및 디버깅.
강승리		1. 독감으로 인한 병결
손민수	4 차 통합 통합 디버깅	1. 프로젝트 통합 및 디버깅.
	2016 년 12 월 5 주차(12 월 26 일	~ 12 월 30 일)
	12 월 26 일 월요일	
공통		1. 팀 내 회의 및 기술결정
임지훈	Server 점검 및 디버깅	1. 서버정리 및 디버깅.
ㅁ끼푼	Voice Chat Network 디버깅	2. 음성통신 디버깅.
강승리	Client 디버깅	1. 클라이언트 디버깅.
손민수	Stage UI 수정	1. Stage 내에서 동작할 UI 3D Button 구현
	12 월 27 일 화요일	1
공통		1. 팀 내 회의 및 기술결정
임지훈	Voice Chat Network	1. 음성통신간 끊김 현상 해결
강승리	Stage 트리거	1. Stage 내에 트리거 생성.
	Client Language	2. 한글 Text -> 일본어 Text 로 변경 중
손민수	Stage UI 수정	1. Stage 내에서 상대방에게 발언권을 줄 Button 생성
12 월 28 일 수요일		
공통		1. 팀 내 회의 및 기술결정
임지훈	Client Character 동기화	1. 캐릭터 상태 및 위치 동기화



강승리 Stage UI 작업 1.Stage UI 조정 및 동기화 1. 발언권 및 mute Button 생성 2만수 Stage UI 수정			
1. 발언권 및 mute Button 생성			
L 프린구 L Stade UL 구성			
2. Stage User profile Image 띄유	기		
12 월 29 일 목요일			
공통 1. 프로젝트 재 통합 및 디버	깅		
임지훈 5 차 통합 1. Client 통합			
Client UI 수정 1. UI mapping 3 전			
5 차 통합 2.Client 통합			
소미스 Client 기능 1. Character 위 cube 에 사진 띄	우기		
손민수 Trinus VR 연동			
12월 30일 금요일			
공통 1. 프로젝트 재 통합 및 디버	깅		
Animation 조정 1. 캐릭터 간 앉기 모션 위치 조	:정		
입지후 디버깅 2. 통합 디버깅			
Database save data 확인 3. Voice Chat Test 4. 결과 전적 저장확인			
기 시 기 Client UI 보완 1. Stage Light mapping			
강승리 Language UI 작업 2. 외국어 기능 UI 작업			
Language UI 작업 1. 외국어 기능 UI 작업			
손민수 발표 준비 2. 2 차 발표 준비			
2017 년 01 월 1 주차(01 월 02 일 ~ 01 월 06 일)			
01월 02일 월요일	01 월 02 일 월요일		
	깅		
	S		



	elula	. = 4
임지훈	디버깅	1. 통합 디버깅
	발표준비	2. 2 차 발표 준비
	Client UI 보완	1. Stage Light mapping
강승리	발표준비	2.2차 발표 준비
손민수	디버깅	1. 통합 디버깅
'	발표준비	2. 2 차 발표 준비
	01 월 03 일 화요일	1
¬ =		
공통		1. 프로젝트 재 통합 및 디버깅
01-1-1	디버깅	1. 통합 디버깅
임지훈	문서작성	2. 문서 작성
강승리	디버깅	1. 통합 디버깅
	문서작성	2. 문서 작성
ا ک ۱۱۱ ک	디버깅	1. 통합 디버깅
손민수	문서작성	2. 문서 작성
	01 월 04 일 수요일	
공통		1. 프로젝트 재 통합 및 디버깅
임지훈	Server 점검	1. WorkServer 보완
		Workserver
	Client music	1. Background Music 설정
강승리	Client UI	2. UI 보완
손민수	Trinus VR	1. Trinus VR Headtracking
_		
	01 월 05 일 목요일	
고트		 1 교리제도 제 토침 미 디비기
공통		1. 프로젝트 재 통합 및 디버깅



임지훈		디버깅 및 유지보수	1. Server 전체적 보완 및 유지보수					
강승리 Client music 2서작성 1. Background Music 설정 및 비보완, 2. 보고서 작성 오민수 Trinus VR 2서작성 1. Trinus VR 기능 보완 2. 보고서 작성 O1월 06일 금요일 공통 1. 프로젝트 재 통합 및 디버깅 임지훈 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 강승리 문서작성 다버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관민수 문서작성 다버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 오민수 모시작성 다버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 공통 1. 프로젝트 재 통합 및 디버깅 임지훈 1. 프로젝트 재 통합 및 디버깅 강승리 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 라너깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관시작성 다버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관인수 문서작성 다버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 소민수 문서작성 다버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정	임지훈							
상하기 문서작성 2. 보고서 작성 유민수 Trinus VR 가능 보완 2. 보고서 작성 01월 06일 금요일 공통 1. 프로젝트 재 통합 및 디버깅 임지훈 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 강승리 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 소민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 2017년 01월 2주차(01월 09일 ~ 01월 10일) 01월 09일 월요일 공통 1. 프로젝트 재 통합 및 디버깅 임지훈 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 강승리 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관인수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 논민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정		문서작성	3. 보고서 작성					
환연수성 2. 보고서 작성 환연수성 1. Trinus VR 기능 보완 2. 보고서 작성 O1월 06일 금요일 공통 1. 프로젝트 재 통합 및 디버깅 임지훈 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 라이리 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 O1월 09일 참으일 공통 1. 프로젝트 재 통합 및 디버깅 임지훈 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 라이리 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정	강승리	Client music	1. Background Music 설정 및 UI 보완,					
전시작성 2. 보고서 작성 01월 06일 금요일 공통 1. 프로젝트 재 통합 및 디버깅 입지훈 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관리수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 2017 년 01월 2주차(01월 09일 ~ 01월 10일) 고통 1. 프로젝트 재 통합 및 디버깅 임지훈 보서작성 기버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 임지훈 보서작성 기버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관이리 문서작성 기버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관민수 문서작성 기버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관민수 문서작성 기버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관민수 문서작성 기버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정	00-1	문서작성	2. 보고서 작성					
문서작성 2. 보고서 작성 01월 06일 금요일 공통 1. 프로젝트 재 통합 및 디버깅 임지훈 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 강승리 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 2017년 01월 2주차(01월 09일 ~ 01월 10일) 조통 1. 프로젝트 재 통합 및 디버깅 임지훈 보서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 라비깅 보시작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관이구 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정	소미스	Trinus VR	1. Trinus VR 기능 보완					
공통 1. 프로젝트 재 통합 및 디버깅 임지호 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 강승리 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 소민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 2017년 01월 2 주차(01월 09일 ~ 01월 10일) 이1월 09일 월요일 공통 1. 프로젝트 재 통합 및 디버깅 임지훈 무서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 강승리 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 소민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 소민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정	순인구 	문서작성	2. 보고서 작성					
임지호 문서작성 다버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 강승리 문서작성 다버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 소민수 문서작성 다버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 2017 년 01월 2 주차(01월 09일 ~ 01월 10일) 지료 의료 기계		01 월 06 일 금요일						
임시	공통		1. 프로젝트 재 통합 및 디버깅					
임시								
강승리 문서작성 다버강 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 소민수 문서작성 다버강 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 2017년 01월 2주차(01월 09일 ~ 01월 10일) 01월 09일월요일 공통 1. 프로젝트 재 통합 및 디버강 임지훈 문서작성 다버강 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 강승리 문서작성 다버강 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 소민수 문서작성 다버강 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 소민수 문서작성 다버강 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정	임지후	문서작성						
강승리 디버깅 2. 프로그램 버그 수정 오민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 2017년 01월 2주차(01월 09일 ~ 01월 10일) 01월 09일월요일 공통 1. 프로젝트 재 통합 및 디버깅 임지훈 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 강승리 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 소민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 소민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 소민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정		디버깅	2. 프로그램 버그 수정					
오민수 문서작성 다버경 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 2017 년 01월 2 주차(01월 09일 ~ 01월 10일) 이1월 09일 월요일 공통 1. 프로젝트 재 통합 및 디버깅 임지훈 문서작성 다버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 강승리 문서작성 다버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 소민수 문서작성 다버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 소민수 문서작성 다버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정	가스리	문서작성	1. 발표 자료 및 보고서 보완					
준민수 디버깅 2. 프로그램 버그 수정 2017 년 01월 2 주차(01월 09일 ~ 01월 10일) 이1월 09일 월요일 공통 1. 프로젝트 재 통합 및 디버깅 임지훈 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 강승리 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 소민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 그버깅 무서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정	004	디버깅	2. 프로그램 버그 수정					
그 변경 2. 프로그램 버그 수정 2. 프로젝트 재 통합 및 디버깅 2. 프로젝트 재 통합 및 디버깅 2. 프로그램 버그 수정 3. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 3. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 3. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 3. 발표 자료 및 보고서 보완 3. 프로그램 버크 수정 3. 발표 자료 및 보고서 보완 3. 프로그램 버크 수정 3. 발표 자료 및 보고서 보완 3. 프로그램 버크 수정 3. 발표 자료 및 보고서 보완 3. 프로그램 버크 수정 3. 발표 자료 및 보고서 보완 3. 프로그램 버크 수정 3. 발표 자료 및 보고서 보완 3. 프로그램 버크 수정 3. 발표 자료 및 보고서 보완 3. 프로그램 버크 수정 3. 발표 자료 및 보고서 보완 3. 프로그램 버크 수정 3. 발표 자료 및 보고서 보완 3. 프로그램 버크 수정 3. 발표 자료 및 보고서 보완 3. 프로그램 버크 수정 3. 발표 3. 플로그램 버크 수정 3. 발표	소미스	문서작성	1. 발표 자료 및 보고서 보완					
01월 09일월요일 공통 1. 프로젝트 재 통합 및 디버깅 임지훈 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 강승리 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 그 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 2. 프로그램 버그 수정	순인구 	디버깅	2. 프로그램 버그 수정					
공통 1. 프로젝트 재 통합 및 디버깅 임지훈 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 강승리 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 소민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 그러나 기 그 발표 자료 및 보고서 보완 2. 프로그램 버그 수정		2017 년 01 월 2 주차(01 월 09 일 ~ 01 월 10 일)						
임지훈 문서작성 다버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 강승리 문서작성 다버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관민수 문서작성 다버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 그러구 무서작성 다버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정		01 월 09 일 월요일						
임시운 디버깅 2. 프로그램 버그 수정 강승리 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정	공통		1. 프로젝트 재 통합 및 디버깅					
강승리 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정 관민수 디버깅 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정	이지호	문서작성	1. 발표 자료 및 보고서 보완					
강승리 디버깅 2. 프로그램 버그 수정 손민수 문서작성 디버깅 1. 발표 자료 및 보고서 보완 2. 프로그램 버그 수정	임시운	디버깅	2. 프로그램 버그 수정					
손민수 문서작성 1. 발표 자료 및 보고서 보완 디버깅 2. 프로그램 버그 주정	고	문서작성	1. 발표 자료 및 보고서 보완					
^{손민수} 디버깅 2. 프로그램 버그 수정	상승리	디버깅	2. 프로그램 버그 수정					
니버강 2. 프로그램 버그 주성	소민소	문서작성	1. 발표 자료 및 보고서 보완					
01 월 10 일 화요일	논민구 	디버깅	2. 프로그램 버그 수정					



공통		1. 프로젝트 재 통합 및 디버깅		
임지훈	문서작성 디버깅	1. 발표 자료 및 보고서 보완, 영상촬영 2. 프로그램 버그 수정		
강승리	문서작성 디버깅	1. 발표 자료 및 보고서 보완, 영상촬영 2. 프로그램 버그 수정		
손민수	문서작성 디버깅	1. 발표 자료 및 보고서 보완, 영상촬영 2. 프로그램 버그 수정		
	01 월 11 일 수요일	1		
공통		1. 팀 내 회의 및 발표 준비		
임지훈	발표 준비	1. 최종 발표 준비 및 발표 연습		
강승리	발표 준비	1. 최종 발표 준비 및 발표 연습		
손민수	발표 준비	1. 최종 발표 준비 및 발표 연습		
	01 월 12 일 목요일			
공통		1. 팀 내 회의 및 발표 준비		
임지훈	발표 준비	1. 최종 발표 준비 및 발표 연습		
강승리	발표 준비	1. 최종 발표 준비 및 발표 연습		
손민수	발표 준비	1. 최종 발표 준비 및 발표 연습		
01 월 13 일 금요일				



공통		1. 최종 발표
임지훈	최종 발표	1. 최종 발표
강승리	최종 발표	1. 최종 발표
손민수	최종 발표	1. 최종 발표



C. 회의록

	회의록				
날짜	2016.11.21	시간	오전 11시 04분		
참석자	임지훈, 강승	임지훈, 강승리, 손민수			
안건	1. 팀명 결정 2. 팀장, 서기 3. 프로젝트 4.역할 분담	기 결정 명 결정			
내용	2. 팀장 결정 팀장 서기 3. 프로젝트 프로젝트 4. 역할 분덤 강경	: 1)셋이서 2)보온 3)필리 4)OOG : 토대리 제 당 : 임지훈 I : 손민수 명 결정 : 명 : 토대리 당 : 이다.	남병 H맨트 O 제작위원회 작위원회 리(VR 토론 시스템)		



			회의록
날짜	2016.11.22	시간	오전 09시 15분
참석자	임지훈, 강성	승리, 손민수	
안건	1. 기획서 직	작성	
	2. 일정계획	표 작성	
내용		1) 업무 정, 2) 아키텍처 표 작성 1) 전체 일	



	회의록				
날짜	2016.11.23 시	간	오전 11 시 04 분		
참석자	임지훈, 강승리,	손민수			
	1. 작일 업무 결	· 보과			
안건	2. 금일 개별 업	부			
	3. 기능정의서 3	3. 기능정의서 작성			
	임지훈	심 : stage È : Data	e 모델링 base 환경 구성(My-Sql) e chat 관련 자료 조사 및 example 구동(결과 : 실패)		
내용	임지훈	: 정)모 ^년 : 정)Da	델링(국회의사당) / 부) Voice Network tabase 설계 및 table 생성, Test / 부) Voice Network # code 로 Voice Network 생성 및 Unity 로 연동 / 부) Database		
			는 문서여서 과거 기수 문서를 참조하여 작성 ·여 작성		



	회의록				
날짜	2016.11.2	4 시간	오전 09시 15분		
참석자	임지훈, 경	당승리, 손민수			
	1. 작일 (업무 결과			
안건	2. 금일 7	배별 업무			
	1. 작일 6		ㅁ데귀 서계 이 너		
		강등리 : stag	e 모델링 설계 완성		
	작성,	임지훈 : Dat	abase table 설계, table 정의서 작성, 포로토콜 설계 가이드라인		
	10,	So	ftware UI 구조설계서 작성		
			e chat 관련 자료 조사 및 example 구동 및 분석, 작동방안 구상 cample 구동 결과 : 음성은 넘어가나 잡음 및 증폭이 심함)		
내용		공 통 : 기능	정의서 작성		
	2. 금일 7	ㅐ별 업무			
	:	강승리 Stage	모델링 2개 완성		
	,	임지훈 : Datab	oase Table 생성, Database Server 연동 및 구축		
	:	손민수 : Netwo	ork 스터디(C# <-> Unity), Voice Network sample 구축		
	-	공 통:팀별	미팅 후 PM advice 참고하여 문서 수정 및 보완		



음성인식 API) 아바타 얼굴 텍스처링은 시간이 다소 오래 걸리므로 맨 마지막에 2. 진행 방향 제시 VR 기능 시 트리너스 VR 을 쓸 것인지 Unity 에서 Android Applic 할 것인지 1) 배포생각을 안할 시 트리너스 VR 사용, 배포 시 트리너스 VIII 배포가 힘든 점 인지 -> Android Application 으로 개발(단점 : object 가 많 폰이 감당 불가) Android 개발 시 Unitypro 에서만 소켓통신이 지원 되므로 Banketwork 처리				회의록
함석자 임지훈, 강승리, 손민수 1. 서류 검토 2. 진행 방향 제시 미팅 일시: 11:40 1. 서류 검토 개발 3 단계 part 도 구현한다는 전제조건. 따라서 3 단계에 작성했던 음성인식도 구현 지시(Google 혹음성인식 API) 아바타 얼굴 텍스처링은 시간이 다소 오래 걸리므로 맨 마지막에 2. 진행 방향 제시 VR 기능 시 트리너스 VR 을 쓸 것인지 Unity 에서 Android Applic 할 것인지 1) 배포생각을 안할 시 트리너스 VR 사용, 배포 시 트리너스 VI 배포가 힘든 점 인지 -> Android Application 으로 개발(단점: object 가 많은 이 감당 불가) Android 개발 시 Unitypro 에서만 소켓통신이 지원 되므로 Banket Network 처리	날짜	2016.11.24	시간	오전 14시 20분
안건 2. 진행 방향 제시 미팅 일시: 11:40 1. 서류 검토	참석자		–	
1. 서류 검토 개발 3 단계 part 도 구현한다는 전제조건. 따라서 3 단계에 작성했던 음성인식도 구현 지시(Google 혹음성인식 API) 아바타 얼굴 텍스처링은 시간이 다소 오래 걸리므로 맨 마지막에 2. 진행 방향 제시 VR 기능 시 트리너스 VR 을 쓸 것인지 Unity 에서 Android Applic 할 것인지 1) 배포생각을 안할 시 트리너스 VR 사용, 배포 시 트리너스 VI 배포가 힘든 점 인지 -> Android Application 으로 개발(단점 : object 가 많은 감당 불가) Android 개발 시 Unitypro 에서만 소켓통신이 지원 되므로 Banketwork 처리	안건			
결정 및 개발	내용	1. 서류 검토 개발 따라 음성인식 AF 아타 2. 진행 방향 VR 할 것인지 1) 배포가 힘든 폰이 감당 불 Network 처 결정 및 개발	를 3 단계 pa라서 3 단기이 하다 얼굴 턴 기능 시 트 개시 기능 시 트 거 -> # 불가) Android 기리 => 최초 및	볚에 작성했던 음성인식도 구현 지시(Google 혹은 Micro Soft



		회의록	
날짜	2016.11.25 시간	오전 09 시 14 분	
참석자	임지훈, 강승리, 손민수		
	1. 작일 업무 결과		
안건	2. 금일 개별 업무		
	1. 작일 업무 결과		
	강승리 : stage	e(도서관) 완성, stage(cafe) 시작(30%)	
	임지훈 : Dat 데이터 입출력 및	abase table 생성 완료, c++기반에서 Database table 접근 및	
	수전	정 test 성공, Database Server 개발 시작	
 내용	손민수 : Photon Cloud 기반 Voice Chat Example Test => 타 예제와 음질 같음 DirectSound 기반 G711Audio 사용 예제 상세 분석		
"0			
	2. 금일 개별 업무		
	강승리 Stage(Cafe) 개발 (완료 시 국회의사당 Stage 개발)	
	임지훈 :Databa	ise Server 개발 계속 진행(model : IOCP)	
	손민수 : Direct	Sound 기반 서버 개발	



	회의록			
날짜	2016.11.28 시간 오전 09 시 14 분			
참석자	임지훈, 강승리, 손민수			
	1. 주말 업무 결과			
안건	2. 금일 개별 업무			
	1. 주말 업무 결과			
	강승리 : 금요일 작업(도서관 stage)작업 부분 소실되어 복구 및 완성, 각 stage 랜더링 무게 등 최적화 필요 부분 최적화 완료, Stage(국회의사당)관련 모델 자료조사			
	임지훈 : Database Server IOCP 모델 70% 완료(쿼리문 작성 진행 중)			
내용	손민수 : C# 기반 단순 TCP 비동기 Server 구축, Unity 내 외부 dll 사용법 조사			
	2. 금일 개별 업무			
	강승리 Stage(국회의사당) 개발			
	임지훈 :Database Server 개발 계속 진행(model : IOCP)			
	손민수 : 단순 Server 에 음성 전송 메서드 추가, 기존 Voice Chat 클라이언트와 테스트			



	회의록				
날짜	2016.11.29 시간	오전 09시 44분			
참석자	 임지훈, 강승리, 손민	수			
-1	1. 작일 업무 결과				
안건	2. 금일 개별 업무				
	1. 작일 업무 결과				
	강승리 : St	age(국회) 90%완료			
	임지훈 : Da	ntabase Server 완료, Unity UI 와 통신 완료, UI 작업 진행			
		rect Sound 기반 클라이언트 개발 실패(사유 : Unity 에서 지원 안함) Unet 기반 클라이언트 응용 진행			
내용	2. 금일 개별 업무				
	강승리 : Sta	ge 완료 후 UI 작업 진행			
	임지훈 : Wo	rk Server 개발 착수, 케릭터 개발 병행			
	손민수 : Voi	ce Client 작업 계속 진행, Gateway Server 개발 착수			
	공 통 : Gitl	nub 환경 구축			



	회의록				
날짜	2016.11.30 시간	오전 09시 20분			
참석자	임지훈, 강승리, 손민수				
	1. 작일 업무 결과				
안건	2. 금일 개별 업무				
	1. 작일 업무 결과				
	공 통 : GitH	ub 연결 실패 (개발 시간으로 인한 작업시간 부족)			
	강승리 : Stage(국회)완료, 모든 Stage 최적화 완료				
	임지훈 : Work Server 구축 진행, 캐릭터 애니메이터 작업				
	손민수 : Client <-> Server 통신 =>결과 : 소켓통신은 되나 음성 데 안넘어감				
내용	2. 금일 개별 업무				
	강승리 : Stage	PC Test 후 UI 작업 진행, Network Study			
	임지훈 : Work	Server <-> 캐릭터 통신			
	손민수 : Voice 시 UI 작업	Client & Sever 음성 데이터 전송 불가 원인 분석 및 수정, 완료			



회의록				
날짜	2016.12.01 시간	오전 09시 23분		
참석자	임지훈, 강승리, 손민수			
안건	 작일 업무 보고 금일 업무 내용 			
내용	임지훈 : Char 손민수 : Voic 전달 X 원 2. 금일 업무 내용 강승리 : UI 디	ge Baking 진행, Stage 최적화 정기적으로 진행 racter <-> Server 통신 , Work Server 진행 te Client <-> Server 통신테스트 -> 소켓 통신만 되고 음성처리 값인 분석 자인 후 UX 작업 착수 Server 작업 계속 진행		
	손민수 : 37 기	project 기술 참조 및 분석, Client 개발 착수		



회의록				
날짜	2016.12.02	시간	오전 09시 10분	
참석자	임지훈, 강승	하리, 손민수		
안건	1. 작일 업두 2. 금일 업두 3. 프로그램	그 내용		
내용	임지 손민 2. 금일 업두 강해 임기 손[3. 프로그램	·리 : Light n 훈 : Work \$ 수 : 37 기 E - 내용 - 내용 이훈 : 다중? 민수 : 음성 기능 논의	napping, UI UX 와 연계한 작업(Login, Join, Lobby) Server 구축 완료, 패킷 정해서 캐릭터 <-> Work Server 통신 xample 분석 -> Voice 부분만 없음. UDP Client, Server 통신 확인 UI 연계한 작업 계속 진행 접속 Test, 패킷 정해서 각 상황 별 Test, 애니메이터 작업 -> Bytestream, Bytestram ->음성 Convert 작업 ble column 항목 : 방모드, 주제, 제목, 최대 인원수	



	회의록				
날짜	2016.12.05	시간	오전 09 시 15 분		
참석자	임지훈, 강	승리, 손민수			
	1. 주말 업	무 보고			
안건	2. 금일 업	무 내용			
	1. 주말 업	무 보고			
	강승리 : UI 틀 제작 완료, UX 60%완료(로그인, 로비, 회원가입)				
	임지훈 : 캐릭터 이동 동기화				
	ĩ	손민수 : 아날	로그 음성 Bytestream 변환 성공, UDP 환경 전송 테스트		
내용	2. 금일 업	무 내용			
	강	'승리 : UX 가	완성, Ingame UI 구성		
	임지훈 : 캐릭!		더 이동 동기화(현재 너무 버벅거려서 추후 데드레커닝 적용 예상)		
	손	:민수 : UDP s	end, recive 부분 재설정 및 테스트		



	회의록				
날짜	2016.12.05	시간	오후 17시 20분		
참석자	PM : 문상환 강사님 임지훈, 강승리, 손민수				
	1. 진행사항	보고			
인건 	2. 애로사항 및 향 후 일정				
	1. 진행사항	보고			
	디자인은 어느정도 되었고 각 캐릭터간 움직임 동기화 진행중. 심각히 버벅거림. 음성채팅 부분은 예제는 많으나 많이 꼬여있어 함부로 사용 불가능 더 많은 자료조사 및 테스팅 필요. 타 팀과 다르게 일정대로 개발 진행중.				
내용	2. 애로사항 및 향 후 일정				
	완료시 팀원		어려운 점으로 인해 팀장(임지훈)이 음성채팅 기능 개발 참여. Database 및 캐릭터 동기화 부분 참여. 팀원(강승리) 네트워크		
	알 수 없는 문제	데 발생>	간 눈에 보일 정도의 버벅임 때문에 서버 문제인지 PC 문제인지 많은 데이터도 아니어서 서버/TCP 문제는 아닌 것으로 판단,		
	음성채팅 완 후	·료 추후 확인	요망		



회의록				
날짜	2016.12.06 시간	오전 09시 12분		
참석자	임지훈, 강승리, 손민수			
	1. 작일 업무 보고			
안건	2. 금일 업무 내용			
	1. 작일 업무 보고			
		JX 일정 부분 완료(Database 연동 X), 씬병합 -> 엄청 버벅거림 > 라이트 지우고 새로 병합 -> 버벅임 감소		
		터 위치 동기화(진행-> PM 미팅 후 보류), 애니메이션 클립 받음 oice Chat Part 투입		
내용	손민수 : Void 진행중	ce Chat 구현 및 테스트 -> 실패, NAudio 이용 Voice Chat 구현		
	2. 금일 업무 내용			
	강승리 : UI 디	자인 후 Network 투입(예정)		
	임지훈 : 음성	대팅 마무리		
	손민수 : G.722	Codec 기반 음성 데이터 압축 및 Voice Chat 구현		



	회의록				
날짜	2016.12.07	시간	오전 09시 14분		
참석자	임지훈, 강승	리, 손민수			
	1. 작일 업무	보고			
안건	2. 금일 업무	내용			
	3. 특별교육				
	1. 작일 업무	보고			
	강승리 : UI, UX 완료 (방 번호 부여 어려움 -> Server 에서 부여하기로 결정)				
	임지훈 : TCP 기반 음성 채팅(Bytestream 변환) 구현(데이터 압축 요망)				
	손민수 : G722 Codec 기반 음성채팅 구현 진행 중(Awake, start 두번 돌아서 수정요망)				
	2. 금일 업무 내용				
내용	강승	: Netwo	rk 예제 기반으로 스터디(오전), Login 데이터 송수신(오후)		
	임지	훈 : 작일 ?	작성 코드 보완, Codec 적용		
	손민	!수 : G.722	Codec 기반 프로그램 작성 및 Vocie Chat 과 합치기		
	피교	자 : 임지훈 육자 : 강승i 내용 : Netw	리, 손민수 ork 통신 간 Packet 간단한 이론 및 전송 및 수신 방법		



		회의록	
날짜	2016.12.08 시간	오전 09 시 14 분	
참석자	임지훈, 강승리, 손민수		
	1. 작일 업무 보고		
안건	2. 금일 업무 내용		
	1. 작일 업무 보고		
	강승리 : 병:	조퇴	
	임지훈 : TCI	P 기반 음성 채팅(Bytestream 변환) 보완, 캐릭터 이동 송수신 작업	
내용	손민수 : TC 착수	CP 기반 음성 채팅 -> UDP 변경 후 테스트, Gateway Server 개발	
"0	2. 금일 업무 내용		
	강승리 : Netv	work 예제 기반으로 스터디(오전), Login 데이터 송수신(오후)	
	임지훈 : 캐릭	니터 이동 송수신 작업	
	손민수 : Gate	eway Server <-> Client Packet 송수신	



			회의록		
날짜	2016.12.09	시간	오전 09 시 10 분		
참석자	임지훈, 강승	임지훈, 강승리, 손민수			
	1. 작일 업두	구 보고			
안건	2. 금일 업두	무 내용			
	1. 작일 업무	무 보고			
	깅	t승리 : Netv	vork -> login Server 연동		
	임지훈 : 애니메이션 통신, 캐릭터간 좌표이동 통신				
	손		Server 2 개 구축(Voice Gateway) 제점 : 서버 1: Client 에서 Receive 하지 못함 서버 2 : Send 못함, Receive 잘 못함		
내용	2. 금일 업두	무 내용			
	강: 없어보임		ork 적용에 따른 UX 수정, 이동간 일시 로그아웃 현상 -> 타 게임도 같은증상. 문제		
	임7	지훈 : Voice	Gateway 마무리, workServer 마무리		
	손[민수 : Voice	Gateway 마무리, Database Server 마무리		



회의록				
날짜	2016.12.12 시간 오전 08시 57분			
참석자	임지훈, 강승리, 손민수			
	1. 주말 업무 보고			
안건	2. 개발 방향성			
	3. 금일 업무 내용			
내용	1. 주말 업무 보고 강승리: Network packet을 포함한 Data 전송 (단 Database Sever 제외) 임지훈: Stage 애니메이션 통신, 음성 Part는 일부 수정 후 사용 가능성 파악 손민수: IOCP Server 구축, 음성 Data 통신(문제점: 속도, 음질) 2. 개발 방향성: 16일 1차 발표에 앞서 현재 진행 속도가 조금 더딘 경향이 있어 12일(월) 로비 Part 완성 및 13(화)~14(수) Stage 개발의 진행 필요성. Voice Chat 기능에 시간을 너무 잡혀 있기 때문에 Voice Chat 기능을 1차 발표 후 진행. 로비에서 방 목록 2~3초 자동 갱신 및 목록 갱신 버튼을 이용하여 방 목록 갱신 표현			
	3. 금일 업무 내용 강승리: Login 성공, 실패 유무 확인(Database), 방 정보 Packet -> 파싱 후 방 정보를 버튼에 뿌려주기, 방 목록은 지역변수로 저장 임지훈: 세부 보완사항 보완 및 각 팀원 개발 지원 손민수: Join Part 에서 사진 등록 기능(Disk 에 저장 후 경로 Database 에 저장) 이후 유저찾기, 친구목록, 자기정보 순으로 로비 기능 투입			



	회의록				
날짜	2016.12.13	시간	오전 09시 03분		
참석자	임지훈, 강승리	믜, 손민수			
	1. 작일 업무	보고			
안건	2. 개발 방향성	성			
	3. 금일 업무	내용			
	1. 작일 업무	보고			
	강승	s리 : Lobby	친구목록, 방만들기 Network 진행중		
	임지훈 : 각 클라이언트 필요 정보 전달, 서버 보완				
	손민 진행중	빈수 : 회원	가입 Network 보완 및 프로필 사진 전송, 내정보 사진 정보		
내용	2. 개발 방향성	Stag	bby 가 끝날 것으로 예상되어 Stage 작업으로 진행될 예정. e 새부적 내용은 Lobby 작업 완료 후 회의 시간에 결정 방장인지 아닌지 구분, 투표창, 상태, 프로필 등		
	3. 금일 업무 강승	., -	친구목록 Network 작업 계속 진행		
	임지	훈 : 클라이	언트에 프로필 사진 전송(본인, 친구), 서버 보완		
	손민수 :개인 프로필 사진 출력, 친구 프로필 사진 출력 공 통 : 완료 후 합치기				



	회의록				
날짜	2016.12.14	시간	오전 09시 13분		
참석자	임지훈, 강승	리, 손민수			
안건	 작일 업무 개발 방향 금일 업무 	성			
	5. ㅁ글 답ㅜ 1. 작일 업무				
	강: 임:	승리 : 방생· 지훈 : 클라 ⁽	성, 친구목록 Network 작업 계속 진행 이언트에 프로필 사진 전송(본인, 친구), 서버 보완 프로필 사진 출력		
2. 개발 방향성 : 로비 합치고 목요일까지 발표 준비에 최선을 기할 것.			합치고 목요일까지 발표 준비에 최선을 기할 것. 섬웨어 감염되었다고 좌절하지말고 앞으로의 일을 향해 최선을		
			논민수)와 Stage UI 작업		
			·지 코드 및 디자인 합치기 승리)와 Stage UI 작업		



회의록						
날짜	2016.12.15 시간	오전 09 시 12 분				
참석자	임지훈, 강승리, 손민=	<u>ት</u>				
안건	1. 작일 업무 보고 2. 금일 업무 내용 3. 발표준비					
내용	3. 발표준비 1. 작일 업무 보고 강승리 : 팀원(손민수)와 Stage UI 작업 임지훈 : 로비까지 코드 및 디자인 합치기 손민수 :팀원(강승리)와 Stage UI 작업 2. 금일 업무 내용 강승리 : Stage UX 작업 임지훈 : 로비까지 코드 및 디자인 수정 및 보완 손민수 :Stage UI 수정 및 보완 공 통 : 16 일 발표 준비 3. 발표준비 1 차 발표(16 일)에 앞서 각 발표 내용은 동기(임지훈), 시연 및 영상(강승리),					



참석자 임지훈, 2 1. 작일 원 3. 발표준 1. 작일 원 2. 금일 원 4. 대용	회의록						
1. 작일 C 2. 금일 C 3. 발표준 1. 작일 C 4. 금일 C 4. 금일 C	날짜 2016	2016.12.16 시간 오전 09 시 16 분					
인건 2. 금일 C 3. 발표준 1. 작일 C 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	참석자 임지	훈, 강승리	l, 손민수				
2. 금일 일 내용	안건 2. 금	1. 작일 업무 보고 2. 금일 업무 내용 3. 발표준비					
3. 발표준 1 차 별	2. 금 내용 3. 벨	1. 작일 업무 보고 강승리 : Stage UX 작업 임지훈 : 로비까지 코드 및 디자인 수정 및 보완, 애니메이션 손민수 :팀원(강승리)와 Stage UI 작업 2. 금일 업무 내용 강승리 : 발표 준비 및 UI, UX 작업 임지훈 : 발표 준비 및 애니메이션 작업 손민수 : 발표 준비 및 UI,UX 작업 공 통 : 16 일 발표 준비					



	회의록					
날짜	2016.12.19 시간 오전 09 시 14 분					
참석자	임지훈, 강승리	의, 손민수				
안건	1. 작일 업무 보고 2. 금일 업무 내용					
내용	임지 손면 2. 금일 업무 강승 임지	리 : Stage L 기훈 : 로비끼 민수 : Stage 내용 리 : UX, Net	IVN 코드 및 디자인 수정 및 보완, 애니메이션 UI 작업 twork 작업			



회의록					
날짜	2016.12.20	시간	오전 09 시 10 분		
참석자	임지훈, 강승	리, 손민수			
	1. 작일 업무	보고			
안건	2. 금일 업무	내용			
내용	임; 손· 2. 금일 업무 강: 임;	승리 : UX, N 지훈 : 애니 ^다 민수 : UI,U 내용 승리 : UX, N	letwork 작업 메이션 및 Voice 작업		



회의록						
날짜	2016.12.20	시간	오전 18시 17분			
참석자	PM : 문상환 강사님 임지훈, 강승리, 손민수					
	1. 진행사항	보고				
안건	2. PM advice	e				
	3. 향 후 진로					
	1. 진행사항 보고 1차 발표 직전 랜섬웨어로 인해 미숙한 발표와 결과에 대한 방향 및 advice를 토대로 진행 하고 있음. Stage 내부 애니메이션 작업 및 음성 채팅 복구 및 보완 조치를 진행 하고 있으며 빠른 시일 내로 복구 및 Project 결과물을 낼 수 있을 것임을 보고					
내용	2. PM Advice					
	너무 느슨한 것도 안좋고 너무 타이트한 것도 안좋지만 적당히 타이트하게 진행하면 좋은 결과물 을 예상할 수 있다고 하심. 겨울철 건강 유의 하라고 하심					
3. 향 후 진로						
	프로젝트	. 후반부라	PM 께서 각자의 진로에 대해 다시 한번 확인 하심.			



회의록				
2016.12.21	시간	오전 09시 25분		
임지훈, 강승리, 손민수				
1. 작일 업두	- 보고			
2. 금일 업두	그 내용			
1. 작일 업무 보고				
강승리 : UI 완료, Network 작업(거시적)				
임	지훈 : 애니	메이션, 캐릭터 이동 완료 및 Voice 복구 및 구축		
손	·민수 : Op	enfile 오류 부분 수정		
2. 금일 업무 내용				
강	·승리 : Stag	e Network 작업(거시적 -> 미시적)		
임	지훈 : Voice	e Netwrok 복구 및 구축		
손민수 : Stage Network 작업 및 UI 다듬기				
	임지훈, 강승 1. 작일 업두 2. 금일 업두 3 임 소 2. 금일 업두 강 임	1. 작일 업무 보고 2. 금일 업무 보고 3가승리 : UI 완임지훈 : 애니 소민수 : Ope 2. 금일 업무 내용 강승리 : Stage		



회의록					
날짜	2016.12.22	시간	오전 09 시 14 분		
참석자	임지훈, 강승	하리, 손민수			
안건	1. 작일 업무 보고 2. 금일 업무 내용				
	1. 작일 업두	로 보고			
	강	·승리 : Stag	e Network 작업(거시적 -> 미시적)		
	임	지훈 : Voice	e Netwrok 복구 및 구축		
	손	·민수 : Sta	ge Network 작업 및 UI 다듬기		
내용	2. 금일 업무 내용				
	강 -	·승리 : 투표	기능 Network 통신 작업		
	임	지훈 : Mu	lti Voice Network 작업		
	손	·민수 : 투표	표 기능 Network 통신 작업		



	회의록					
날짜	2016.12.23	시간	오전 10시 55분			
참석자	임지훈, 손민	임지훈, 손민수				
	1. 작일 업무	무 보고				
안건	2. 금일 업무	무 내용				
	3. PM 전달	사항 전파				
	1. 작일 업두	무 보고				
	강승리 : 투표 기능 Network 통신 작업(오전 유선보고) 임지훈 : Multi Voice Network 작업 손민수 : 투표 기능 Network 통신 작업					
	2. 금일 업무 내용					
내용	강승리 : 독감으로 인한 병결 임지훈 : 완성 후 합치기 손민수 : 완성 후 합치기					
	3. PM 전달사항 전파					
	2)	1.3 발표 [©] 아픈사람은 ex) 팀원 c	I 무조건 강사님께 카톡남기기 II 목표치 까지 보여줄 것 II 팀장이 파악 후 쉬게 할 경우 강사님께 카톡남기기 POOO 아파서 쉬게하겠습니다. IF 문서 꼭 SVN 에 올릴 것			



	회의록				
날짜	2016.12.26	시간	오전 10시 55분		
참석자	강승리, 임지훈, 손민수				
안건	1. 금일 업무 내용				
내용	1. 금일 업무 내용 강승리 : Client Debug 임지훈 : Client Debug 에 따른 Server Debug & Voice Chat Test 손민수 : In Stage UI (3D Button) & Debug				



	회의록				
날짜	2016.12.27 시간 오전 09시 10분				
참석자	강승리, 임지훈, 손민수				
	1. 작일 업두	무 보고			
안건	2. 금일 업두	무 내용			
	1. 작일 업무 보고				
	강승리 : Client Debug				
	임	지훈 : Clier	nt Debug 에 따른 Server Debug & Voice Chat Test		
	손	·민수 : In St	age UI (3D Button) & Debug		
내용	2. 금일 업무 내용				
	강	승리 : Stag	e model 조정, 트리거 씌우기		
	임	지훈 : Voic	e Chat 끊김현상 수정		
	손	·민수 : 발언	권 부여기능 Button 기능 추가		



	회의록					
날짜	2016.12.28 시간 오전 09 시 10 분					
참석자	강승리, 임지훈, 손민수					
	1. 작일 업무	무 보고				
안건	2. 금일 업두	무 내용				
	3. 남은 기능					
1. 작일 업무 보고 강승리 : Stage model 조정, 트리거 씌우기, 외국어 기능(진행중 임지훈 : Voice Chat 끊김현상 수정 손민수 : 발언권 부여기능 Button 기능 추가						
내용	임	승리 : UI 조 지훈 : 상태	정 및 동기화 동기화 및 캐릭터 위치 동기화 권 및 mute Button 생성 및 In Stage profile Image 띄우기			
	3. 남은 기능 1) 발언권 및 mute Button 2) In Stage profile Image 3) 외국어 기능 4) 상태 동기화(입장 시 캐릭터 모션, 위치) 5) VR 6) 웹 파싱 7) 음성인식					



	회의록					
날짜	2016.12.29	시간	오전 10시 50분			
참석자	강승리, 임지훈, 손민수					
	1. 작일 업두	무 보고				
안건	2. 금일 업두	무 내용				
	1. 작일 업무	무 보고				
	강	승리 : UI 조	정 및 동기화			
	임	지훈 : 상태	동기화 및 캐릭터 위치 동기화			
내용	손민수 : 발언권 및 mute Button 생성 및 In Stage profile Image 띄우기					
네 성	2. 금일 업두	무 내용				
	강	·승리 : UI m	apping, 합치기			
	임	지훈 : 합치	기 작업			
	손	:민수 : Cube	e profile Image 띄우기, Trinus VR 연동			



	회의록					
날짜	2016.12.30	시간	오전 10 시 30 분			
참석자	강승리, 임지	훈, 손민수				
안건	1. 작일 업두 2. 금일 업두					
	1. 작일 업두	구 보고				
	강승리 : UI mapping, 합치기					
	임지훈 : 합치기 작업					
내용	손	·민수 : Cube	e profile Image 띄우기, Trinus VR 연동			
	2. 금일 업무 내용					
	강승리 : Light mapping					
	임	지훈 : Char	acter 앉기모션 위치 조정 ,통합 디버깅			
	손	·민수 : UI 디	자인 수정, Trinus VR 연동			



회의록				
날짜	2017.1.02	시간	오전 10시 00분	
참석자	강승리, 임	지훈, 손민수	·	
안건	1. 주말 업무 보고 2. 금일 업무 내용			
내용	2. 금일 업·	강승리 : Ligh 임지훈 : Cha 진 논민수 : 외국 무 내용 강승리 : Ligh	at mapping, 외국어 기능 UI 작업 racter 앉기모션 위치 조정 ,통합 디버깅 라버그 잡기, 음성테스트, 결과 전적 저장확인 라어 기능 UI 작업, 발표 준비 라 mapping, 추가 통합, 발표 준비 나 통합, 통합 디버깅, 발표 준비	
	í	손민수 : 추기	ト 통합, 발표 준비	



	회의록					
날짜	2017.1.03	시간	오전 10시 07분			
참석자	강승리, 임	지훈, 손민수	:			
안건	1. 작일 업 2. 금일 업					
	1. 작일 업	무 보고				
	-	강승리 : Light mapping, 추가 통합, 발표 준비				
	임지훈 : 추가 통합, 통합 디버깅, 발표 준비					
	ĩ	손민수 : 추기	ト통합, 발표 준비			
내용	2. 금일 업	무 내용				
	- 7	강승리 : 통협	합 디버깅, 문서 작성			
	C	임지훈 : 통협	합 디버깅, 문서 작성			
	ĩ	손민수 : 통협	합 디버깅, 문서 작성			



	회의록					
날짜	2017.1.04	시간	오전 10시 30분			
참석자	강승리, 임지훈, 손민수					
	1. 작일 업					
안건	2. 금일 업무 내용					
	1. 작일 업	1. 작일 업무 보고				
	강승리 :통합 디버깅, 문서 작성					
	임지훈 : 통합 디버깅, 문서 작성					
	손민수 : 통합 디버깅, 문서 작성					
내용						
	2. 금일 업무 내용					
	강승리 : Backgraound Music 설정 및 UI 보완					
	임지훈 : WorkServer 보완					
	ί	논민수 : Trin	us VR Headtracking 및 click Event			



	회의록						
날짜	2017.1.05	시간	오전 09 시 10 분				
참석자	강승리, 임	지훈, 손민수					
안건		1. 작일 업무 보고 2. 금일 업무 내용					
	1. 작일 업	무 보고					
	-	강승리 : Backgraound Music 설정 및 UI 보완					
	Ç	임지훈 : Wo	kServer 보완				
내용	τ	손민수 : Trin	us VR Headtracking 및 click Event				
	2. 금일 업	무 내용					
	3	강승리 : Bac	kgraound Music 설정 및 UI 보완, 보고서 작성				
	Ç	임지훈 : Serv	ver 전체적 보완 및 유지보수, 보고서 작성				
	ĩ	손민수 : Trin	us VR Headtracking 및 click Event, 보고서 작성				



	회의록					
날짜	2017.1.06	시간	오전 09시 10분			
참석자	강승리, 임	지훈, 손민수				
안건	1. 작일 업무 보고 2. 금일 업무 내용					
내용	2. 금일 업·	강승리 : Bac 임지훈 : Serv 손민수 : Trin 무 내용 강승리 : 완료 임지훈 : 완료	kground Music 설정 및 UI 보완, 보고서 작성 ver 전체적 보완 및 유지보수, Trinus VR 기능 보완 보고서 작성 us VR 기능 보완, 보고서 작성 로 보고서 작성 로 보고서 작성 및 통합 디버깅 로 보고서 작성			



	회의록				
날짜	2017.1.09	시간	오전 09시 25분		
참석자	강승리, 임	지훈, 손민수	:		
	1. 주말 업	무 보고			
안건	2. 금일 업무 내용				
	1. 주말 업	무 보고			
	7	강승리 : 발표	표 자료 보완 및 버그 수정		
	Ç	임지훈 : 발표	표 자료 보완 및 버그 수정		
내용	ť	손민수 : 발표	표 자료 보완 및 버그 수정		
	2. 금일 업	무 내용			
	3	강승리 : 발표	E 자료, 프로젝트 보고서 보완 및 버그 수정		
	Ç	임지훈 : 발표	E 자료, 프로젝트 보고서 보완 및 버그 수정		
	í	손민수 : 발표	E 자료, 프로젝트 보고서 보완 및 버그 수정		



회의록				
날짜	2017.1.10	시간	오전 09시 10분	
참석자	강승리, 임지훈, 손민수			
	1. 작일 업	무 보고		
안건	2. 금일 업	무 내용		
	3. 발표 연 [.]	습 결과 및	보완 사항	
	1. 작일 업	무 보고		
	7	강승리 : 발표	표 자료, 프로젝트 보고서 보완 및 버그 수정	
	Ç	임지훈 : 발표	표 자료, 프로젝트 보고서 보완 및 버그 수정	
	í	손민수 : 발표	표 자료, 프로젝트 보고서 보완 및 버그 수정	
내용	2. 금일 업	무 내용		
-110	3	강승리 : 개팀	별 문서 작성 및 팀 보고서 작성, 시연 영상 수정	
	Ç	임지훈 : 개팀	별 문서 작성 및 팀 보고서 작성, 시연 영상 수정	
	Í	손민수 : 개팀	별 문서 작성 및 팀 보고서 작성, 시연 영상 수정	
	3. 발표 연습 결과 및 보완 사항 발표 연습을 통해 많이 나아진 점은 있으나 자잘한 환경 설정, Trinus VR 연결 및			
	화면	시점 수정	필요. 발표 시간 약 39분 -> 단축 요망	



	회의록					
날짜	2017.1.11	시간	오전 09시 20분			
참석자	강승리, 임	강승리, 임지훈, 손민수				
안건	1. 작일 업 2. 금일 업					
내용	2. 금일 업	강승리 : 개발 임지훈 : 개발 소민수 : 개발 무 내용 강승리 : 최종	별 문서 작성 및 팀 보고서 작성, 시연 영상 수정 별 문서 작성 및 팀 보고서 작성, 시연 영상 수정 별 문서 작성 및 팀 보고서 작성, 시연 영상 수정 등 발표 준비 및 발표 연습 등 발표 준비 및 발표 연습			



	회의록					
날짜	2017.1.12	시간	오전 09시 11분			
참석자	강승리, 임	지훈, 손민수				
안건	1. 작일 업· 2. 금일 업·					
	1. 작일 업	무 보고				
	ā	강승리 : 최종	등 발표 준비 및 발표 연습			
	Ç	임지훈 : 최종	등 발표 준비 및 발표 연습			
내용	τ	손민수 : 최충	등 발표 준비 및 발표 연습			
	2. 금일 업	무 내용				
	-	강승리 : 최종	등 발표 준비 및 발표 연습			
	Ç	임지훈 : 최경	통 발표 준비 및 발표 연습			
	ĩ	소민수 : 최경	통 발표 준비 및 발표 연습			



회의록					
날짜	2017.1.13	시간	오전 09시 10분		
참석자	강승리, 임지훈, 손민수				
안건	1. 작일 업· 2. 금일 업·				
내용	2. 금일 업·	강승리 : 최종 임지훈 : 최종 손민수 : 최종	등 발표		



6. 프로젝트 개발 내용

A. Source Code

Server

네트워크 통신을 위한 패킷의 CMD (2byte)

```
('0' << 8) | 'J'
    #define CMD_JOIN
                                                       ('0' << 8) | 'L'
    #define CMD_LOGIN
    #define CMD_LOGOUT
                                                       ('1' << 8) | 'L'
                                                       ('R' << 8) | 'C'
    #define CMD_CREATE_ROOM
                                                       ('R' << 8) | 'E'
    #define CMD ENTER ROOM
    #define CMD_LEAVE_ROOM
                                                       ('R' << 8) | 'L'
                                                       ('R' << 8) | 'S'
    #define CMD_START_ROOM
    #define CMD_GET_ROOM_LIST
                                                       ('L' << 8) | 'R'
                                                       ('I' << 8) | 'F'
    #define CMD_FRIEND_FIND_ID
                                                      ('N' << 8) | 'F'
    #define CMD_FRIEND_FIND_NAME
                                                      ('|' << 8) | '$'
    #define CMD_SAVE_PROFILE_IMAGE
                                                       ('|' << 8) | 'L'
    #define CMD_LOAD_PROFILE_IMAGE
                                                       ('|' << 8) | 'U'
    #define CMD_GET_USER_INFO
                                                       ('A' << 8) | 'F'
    #define CMD_FRIEND_ADD
                                                       ('F' << 8) | 'G'
    #define CMD_GET_FRIEND_LIST
                                                       ('I' << 8) | 'K'
    #define CMD_KEY_PACKET
                                                       CMD_KEY_PACKET
    #define CMD_GET_USER_LIST_INROOM
                                                       ('0' << 8) | 'F'
    #define CMD_FRIEND_ADD_SUCCESS
                                                       ('9' << 8) | 'F'
    #define CMD_FRIEND_ADD_FAIL
    #define CMD_JOIN_SUCCESS
                                                       ('S' << 8) | 'J'
                                                       ('S' << 8) | 'L'
    #define CMD_LOGIN_SUCCESS
                                                       ('F' << 8) | 'J'
    #define CMD_JOIN_FAIL
                                                       ('F' << 8) | 'L'
    #define CMD_LOGIN_FAIL
                                                       ('D' << 8) | 'S'
    #define CMD_DB_SUCCESS
                                                       ('E' << 8) | 'E'
    #define CMD DB FAIL
                                                       ('3' << 8) | '0'
    #define CMD_NOT_FOUND_CMD
                                                       ('0' << 8) | 'R'
    #define CMD_STAGE_RESULT
                                                       ('S' << 8) | 'C'
#define CMD_CONNECT_STAGE
#define CMD_LOGED_IN
                                              ('1' << 8) | '1'
                                              ('S' << 8) | 'S'
#define CMD_POS_STATUS_PACKET
                                                       ('S' << 8) | 'G'
#define CMD_GET_POS_STATUS_PACKET
```

페이지 180 / 261



```
#define CMD_SET_VOICE_PACKET
                                             ('V' << 8) | 'S'
                                             ('V' << 8) | 'G'
   #define CMD_GET_VOICE_PACKET
                                                    ('V' << 8) | 'V'
   #define CMD_START_VOICE
                                                    ('S' << 8) | 'V'
   #define CMD_VOTE_START
                                                    ('A' << 8) | 'V'
   #define CMD_VOTE_AGREE
                                                    ('D' << 8) | 'V'
   #define CMD_VOTE_DISAGREE
                                                    ('N' << 8) | 'V'
   #define CMD_VOTE_NULLITY
                                           ('R' << 8) | 'V'
   #define CMD_GET_VOTE_RESULT
   IOCP모델
   각 쓰레드마다 자신만의 메모리를 정적 할당.
   char ThreadBuf[CNT THREAD][SIZE THREAD BUF];
연결이 끊긴 소켓의 메모리를 해제할 때,
다른 쓰레드는 해당 소켓에 대한 다른 작업을 하지 않아야 하기 때문에 다른 쓰레드들은
처리작업을 수행하지 않아야 하기 위한 Critical Section
   CRITICAL SECTION thread CS[CNT THREAD];
   Completion Queue 에 쌓인 Recv, Send 정보를 처리하는 함수
   void DataProcess(char *threadBuf, HANDLE hcp, DWORD cbTransferred, SOCKETINFO* ptr, LPOVERLAPPED
   lpResultOverlapped, int threadNum)
       int retval = 0;
       unsigned short cmd = 0;
       if (&ptr->overlappedRecv == lpResultOverlapped)
       {
               IOCP_RecvViewMessage(ptr, cbTransferred, threadNum);
               Main_WorkServerLogic(threadBuf, hcp, cbTransferred, ptr, threadNum);
              Req_WSARecv(ptr, hcp, threadNum);
       }
       else
       {
               ptr->Lock();
```



```
ptr->sentbytes += cbTransferred;
            if (ptr->willSendbytes < ptr->sentbytes)
            {
                    // 데이터 더 보내기
                    Req_WSASend(ptr);
            }
            else
            {
                    ptr->willSendbytes = SIZE_BUF;
                    ptr->sentbytes = SIZE_BUF;
            ptr->UnLock();
    }
}
모든 쓰레드들이 동작할 함수. (연결끊김, 데이터처리를 한다.)
unsigned int CALLBACK WorkerThread(LPV0ID arg)
    HANDLE hcp = (HANDLE)arg;
    int retval;
    static int threadCnt = 0;
    int ThreadNum = threadCnt++;
    while (1)
    {
            DWORD cbTransferred;
            LPOVERLAPPED IpResultOverlapped;
            SOCKETINFO** pptr = NULL;
            retval = GetQueuedCompletionStatus(
                    hcp, &cbTransferred,
                    (LPDWORD)&pptr,
                    &lpResultOverlapped, INFINITE);
            //printf("cbTransferred = %d\m", cbTransferred);
            if ((retval == false || (cbTransferred == 0 /*&& lpResultOverlapped ==
&(*pptr)->overlappedRecv*/)) && IpResultOverlapped != NULL)
                    Enter_thread_CS();
                    if ((*pptr) != NULL)
                    {
                             IOCP_ViewMessage((*pptr)->sock, "Closed!!");
                             if (retval == false && pptr != NULL)
                             {
                                     DWORD temp1, temp2;
                                     WSAGetOverlappedResult((*pptr)->sock,
                                             IpResultOverlapped, &temp1, FALSE, &temp2);
```



```
::printf("\t");
                                     ViewError(WSAGetLastError());
                            CloseSocketInfo(hcp, ThreadBuf[ThreadNum], (*pptr), ThreadNum);
                             (*pptr) = NULL;
                    Leave_thread_CS();
                    //DisConnect_pptr(hcp, ThreadNum, pptr);
                    continue;
            }
            else if (lpResultOverlapped == NULL)
                    ::printf("Out OverlappedIO* is Null\n");
                    continue;
            }
            Enter_thread_CS(ThreadNum);
            if((*pptr) != NULL)
                    DataProcess(ThreadBuf[ThreadNum], hcp, cbTransferred, (*pptr),
lpResultOverlapped, ThreadNum);
            Leave_thread_CS(ThreadNum);
    }
    return 0;
메인쓰레드에서 돌아가는 함수. (소켓의 연결을 기다리고 받는다.)
void Run_IOCP_Model(SOCKET &listenSock, HANDLE &hcp)
   SOCKADDR_IN clientaddr;
    int addrlen = sizeof(clientaddr);
   SOCKET client sock;
   ViewMessage("IOCP : IOCP_Model Run!!!");
   while (1)
            client_sock = accept(listenSock, (SOCKADDR*)&clientaddr, &addrlen);
            if (client_sock == INVALID_SOCKET)
            {
                    ViewMessage("IOCP : accept() error");
                    continue;
            10CP_ViewMessage(client_sock, "Connected!!");
                                   페이지 183 / 261
```



```
// 소켓 정보 구조체 할당
           SOCKETINFO **pptr = new SOCKETINFO*();
           *pptr = new SOCKETINFO(client_sock, pptr);
           if (*pptr == NULL)
                   ViewMessage("IOCP: [오류] 메모리가 부족합니다");
                   break:
           }
           // 1번 인자와 2번 인자를 결합하고, 3번 인자를 주고받는다.
           HANDLE hResult = CreateloCompletionPort((HANDLE)client_sock, hcp, (DWORD)pptr, 0);
           if (hResult == NULL)
                   return;
           Req_WSARecv(*pptr);
    }
}
DBServer Main
DBServer 에서 MySQL로 하나의 쓰레드만 접속 가능하게 하기 위한 Critical Section
CRITICAL_SECTION DB_CS;
DB Server 에서 Completion Queue를 통해 들어오는 Recv 데이터를 처리하기 위한 Main 함수
int MAIN_DB(char *Packet, unsigned short Len_Pack, char *SendBuf, unsigned short *Len_SendBuf)
    int ret_val = 0;
   char* split_buf[20];
   ParseRecvBuf(Packet, split_buf);
   unsigned short cmd = *(unsigned short*)Packet;
   EnterCriticalSection(&DB_CS);
   switch (cmd)
    {
   case CMD_JOIN:
           ret_val = Join(split_buf, SendBuf, Len_SendBuf);
           break;
   case CMD_LOGIN:
           ret_val = Login(split_buf, SendBuf, Len_SendBuf);
```



```
break;
    case CMD_LOGOUT:
             ret_val = LogOut(split_buf, SendBuf, Len_SendBuf);
            break;
    case CMD_CREATE_ROOM:
            ret_val = CreateRoom(split_buf, SendBuf, Len_SendBuf);
            break;
    case CMD_ENTER_ROOM:
             ret_val = EnterRoom(split_buf, SendBuf, Len_SendBuf);
            break;
    case CMD_LEAVE_ROOM:
            ret_val = LeaveRoom(split_buf, SendBuf, Len_SendBuf);
            break:
    case CMD START ROOM:
            ret_val = StartRoom(split_buf, SendBuf, Len_SendBuf);
            break;
    case CMD_GET_ROOM_LIST:
            ret_val = GetRoomList(SendBuf, Len_SendBuf);
            break;
    case CMD_GET_USER_INFO:
    case CMD_FRIEND_FIND_ID:
            ret_val = GetUserInfo(*(unsigned short *)Packet, split_buf, SendBuf, Len_SendBuf);
            break;
    case CMD FRIEND FIND NAME:
            ret_val = GetUserInfoByName(*(unsigned short *)Packet, split_buf, SendBuf,
Len_SendBuf);
            break;
    case CMD_GET_FRIEND_LIST:
            ret_val = GetFriendList(split_buf, SendBuf, Len_SendBuf);
            break;
    case CMD_FRIEND_ADD:
            ret_val = AddFriend(split_buf, SendBuf, Len_SendBuf);
            break;
    case CMD_GET_USER_LIST_INROOM:
            ret_val = GetUserListInRoom(split_buf, SendBuf, Len_SendBuf);
            break;
    case CMD VOTE AGREE:
            AgreeInStage(split_buf, SendBuf, Len_SendBuf);
            break;
    case CMD_VOTE_DISAGREE:
            DisagreeInStage(split_buf, SendBuf, Len_SendBuf);
            break;
    case CMD_GET_VOTE_RESULT:
            GetResult(split_buf, SendBuf, Len_SendBuf);
            break;
    case CMD_LOGED_IN:
```



```
LogedIn(split_buf, SendBuf, Len_SendBuf);
           break;
    default:
           ret_val = FAIL_CMD;
           MakeSendPacket_CMD(SendBuf, CMD_NOT_FOUND_CMD, Len_SendBuf);
    }
   LeaveCriticalSection(&DB_CS);
    return EXSIST_SEND_BUF;
}
WorkServer Main
유저의 정보를 아이디와 소켓, 각 게임상 데이터들을 연결시켜줄 Map 구조 변수.
삽입. 삭제보다 Search 동작이 훨씬 많으므로. 시간을 단축시키기 위한 Map 구조로 변수 선언.
두개의 변수는 같은 데이터를 가르키고 ld or SocketPtr 로 데이터에 접근할 수 있도록 선언
std::map<string, UserData*> mapUserInfo_InRoom_id;
std::map<SOCKETINFO*, UserData*> mapUserInfo_InRoom_ptr;
CRITICAL_SECTION UserInfo_CS;
Work Server 에서 Completion Queue를 통해 들어오는 Recv 데이터를 처리하기 위한 Main 함수.
void Main_WorkServerLogic(char *threadBuf, HANDLE hcp, unsigned int cbTransferred, SOCKETINFO *ptr,
int threadNum)
{
   ptr->recievedbytes += cbTransferred;
   unsigned short length = *(unsigned short *)&ptr->recvBuf[2];
   while (ptr->recievedbytes >= SIZE HEADER && length <= ptr->recievedbytes)
           switch (*(unsigned short *)&ptr->recvBuf)
           case CMD_L0G0UT:
           case CMD_GET_ROOM_LIST:
           case CMD_VOTE_AGREE:
           case CMD_VOTE_DISAGREE:
           case CMD_GET_VOTE_RESULT:
           case CMD_GET_USER_INFO:
           case CMD_FRIEND_FIND_ID:
           case CMD_FRIEND_FIND_NAME:
           case CMD_FRIEND_ADD:
```

페이지 186 / 261



```
case CMD_GET_FRIEND_LIST:
             case CMD_STAGE_RESULT:
                     Req_DB(ptr->recvBuf, length, threadBuf, SIZE_THREAD_BUF, threadNum);
                     SendBuf_ptr(ptr, hcp, threadBuf, *(TYPE_HEADER_LENGTH *)(threadBuf +
SIZE_HEADER_CMD), threadNum);
                     break;
             case CMD_START_ROOM:
                     Req_DB(ptr->recvBuf, length, threadBuf, SIZE_THREAD_BUF, threadNum);
                     SendBuf_ptr(ptr, hcp, threadBuf, *(TYPE_HEADER_LENGTH *)(threadBuf +
SIZE_HEADER_CMD), threadNum);
                     Scatter_CMD_InRoom(threadBuf, hcp, ptr, threadNum, CMD_START_ROOM);
                     break;
             case CMD_LOAD_PROFILE_IMAGE:
                     Req_DB(ptr->recvBuf, length, threadBuf, SIZE_THREAD_BUF, threadNum);
                     SendBuf_ptr_img(ptr, hcp, threadBuf, *(int *)(threadBuf + SIZE_HEADER_CMD),
threadNum);
                     break:
             case CMD_CREATE_ROOM:
                     CreateRoom(threadBuf, hcp, ptr, threadNum, length);
                     break;
             case CMD LEAVE ROOM:
                     LeaveRoom(threadBuf, hcp, ptr, threadNum);
                     break;
             case CMD ENTER ROOM:
                     EnterRoom(threadBuf, hcp, ptr, threadNum, length);
             case CMD_LOGED_IN:
                     LogedIn(threadBuf, hcp, ptr, threadNum);
                     break:
             case CMD_KEY_PACKET:
                     //PrintPacket(ptr, ptr->recvBuf, *(TYPE_HEADER_LENGTH *)(ptr->recvBuf +
SIZE_HEADER_LENGTH), "Recv");
                     TradeKeyPacket(threadBuf, hcp, ptr, threadNum, length);
                     //printf("_____ %s : ", mapUserInfo_InRoom_ptr[ptr]->Id);
                     //for (int i = 0; i < 8; i++)
                     //{
                              printf("%d ", (mapUserInfo_InRoom_ptr[ptr]->Curkey >> i) & 1);
                     //
                     //}
                     //printf("\n");
                     break;
             case CMD_POS_STATUS_PACKET:
                     Set_Pos_Status_Packet(ptr);
                     break;
             case CMD GET POS STATUS PACKET:
                     Get_Pos_Status_Packet(threadBuf, hcp, ptr, threadNum);
                     break:
```



VoiceServer Main

VoiceServer 는 방대한 데이터를 처리해야 하기 때문에 WorkServer 나 DBServer 와 패킷구조가 다르다.

따라서 CMD byte수와 CMD 타입, 패킷길이의 byte수와 패킷길이의 타입 등을 전처리기에서 선언.

```
#define SIZE_HEADER_VOICE_CMD
#define TYPE_HEADER_VOICE_CMD
                                      unsigned short
#define SIZE_HEADER_VOICE_LENGTH
#define TYPE_HEADER_VOICE_LENGTH
                                      int
#define SIZE_HEADER_VOICE
                                              SIZE_HEADER_VOICE_CMD + SIZE_HEADER_VOICE_LENGTH
#define VOICE_FREQUENCY
                                                       44100
#define SIZE_VOICE_BUFFER_FLOAT
                                              VOICE FREQUENCY / 2
#define SIZE_VOICE_BUFFER
                                              SIZE_VOICE_BUFFER_FLOAT * 4
                                     (SIZE_VOICE_BUFFER_FLOAT * 1000) / VOICE_FREQUENCY
#define TIME_VOICE_BUFFER_MS
```

여기서도 Id 와 Socket Ptr, 음성데이터부분을 연결할 Map 구조 변수 선언.

삽입, 삭제보다 Search 동작이 훨씬 많으므로, 시간을 단축시키기 위한 Map 구조로 변수 선언.

여기도 역시 같은 객체를 가르키고 ld 와 Ptr 로 접근을 할 수 있다.

```
std::map<SOCKETINFO*, VoiceData*> mapVoice_ptr;
```



```
std::map<string, VoiceData*> mapVoice_id;
Voice Server 에서 Completion Queue를 통해 들어오는 Recv 데이터를 처리하기 위한 Main 함수.
void Main_VoiceServerLogic(char *threadBuf, HANDLE hcp, unsigned int cbTransferred, SOCKETINFO *ptr,
int threadNum)
{
    ptr->recievedbytes += cbTransferred;
    TYPE_HEADER_VOICE_LENGTH length = *(TYPE_HEADER_VOICE_LENGTH *)&ptr->recvBuf[2];
    while (ptr->recievedbytes >= SIZE_HEADER_VOICE && length <=</pre>
(TYPE_HEADER_VOICE_LENGTH)ptr->recievedbytes)
            switch (*(TYPE_HEADER_VOICE_CMD *)&ptr->recvBuf)
            case CMD_START_V01CE:
                     StartVoice(ptr, threadNum);
                     break;
            case CMD_SET_VOICE_PACKET:
                     SetVoiceData(ptr, threadNum);
                     break:
             case CMD_GET_VOICE_PACKET:
                     Send_Voice_Data(ptr, threadNum, threadBuf);
             case CMD_V01CE_GIVE_T0_SAY:
                     Send_CMD_To_Other_Voice(threadBuf, hcp, ptr, threadNum,
CMD_V0ICE_GIVE_T0_SAY);
                     break;
            case CMD_V01CE_STOP_T0_SAY:
                     Send_CMD_To_Other_Voice(threadBuf, hcp, ptr, threadNum,
CMD_V01CE_STOP_T0_SAY);
                     break;
            default:
                     break:
            ptr->recievedbytes = ptr->recievedbytes - (unsigned int)length;
            memcpy(ptr->recvBuf, ptr->recvBuf + length, ptr->recievedbytes);
            length = *(TYPE_HEADER_VOICE_LENGTH *)&ptr->recvBuf[2];
    }
}
```



Client

```
Singleton 부분
```

Client 에서 데이터를 일괄적으로 관리하기 위해서 Singleton 모델링 적용. 또한 DonDestroy 이기 때문에 Client의 모든씬에서 공통적으로 사용할 간단한 함수포함.

```
public class csSingleton : MonoBehaviour
{
   public string WorkIP = "14.32.7.30";
   public int WorkPort = 9000;
   public string VoiceIP = "14.32.7.30";
   public int VoicePort = 9500;
   public string DBIP = "14.32.7.30";
   public int DBPort = 8000;
   public Socket WorkSock;
   public string id = "";
   public int cur_room = 0;
   public string name_room_mode = "3_Lib1";
   public const float POSITION_REVISION_VALUE = 3f;
   public const float ROTATION_REVISION_VALUE = 10f;
   public int ReadyUserCnt_InRoom { get; set; }
   public int language_int = 0;
    public string nick;
   public string regiment;
   public int join_count;
   public int passed_count;
   public int reject_count;
   public int Master_type = 0;
   public int Total_Agree = 0;
   public int Total_disagree = 0;
   public int Total_nullity = 0;
   public string total_result;
   public bool bEnd_Room = false;
   public bool bStart_Room = false;
   public bool bVote_Start = false;
   public bool start_talk_popup = false;
   public bool bMyTriggerExe = false;
```



```
public bool bReady = false;
    public void InitStageValue()
            //Master_type = 0;
            bEnd_Room = false;
            bStart_Room = false;
            bVote_Start = false;
            start_talk_popup = false;
            ReadyUserCnt_InRoom = 0;
    }
    public string Path_Image = Directory.GetCurrentDirectory() + "/pro_img/";
    public GameObject Camera_InStage;
    public static csSingleton singleton = null;
    // Use this for initialization
    void Awake ()
    {
            if(singleton != null)
                    Destroy(singleton.gameObject);
            Application.runInBackground = true;
            Path_Image = Directory.GetCurrentDirectory() + "/pro_img/";
            Application.runInBackground = true;
            DontDestroyOnLoad(gameObject);
            singleton = this;
    private void OnApplicationQuit()
            WorkSock.Close();
            WorkSock = null;
    void Update()
            //탭키 누르면 InputField 커서 이동
            if(Input.GetKeyDown(KeyCode.Tab))
            {
                    EventSystem system =
GameObject.Find("EventSystem").GetComponent<EventSystem>();
                    Selectable next =
system.currentSelectedGameObject.GetComponent<Selectable>().FindSelectableOnDown();
                    if(next != null)
```



```
InputField inputfield = next.GetComponent<InputField>();
                           if(inputfield != null)
                                   inputfield.OnPointerClick(new PointerEventData(system));
                   }
           }
   }
}
Login_Join Recv
서버와 연결하여 Login 및 Join 에서 Recv 받은 데이터를 처리할 함수.
하나의 Socket 이기 때문에 Recv 버퍼는 하나이어야 손실되는 패킷이 없다.
public string receive()
   {
       receiveData = "";
       int bytes = stream.Read(data, 0, data.Length);
       receive_cmd = System.Text.Encoding.Default.GetString(data, 0, 2);
       receicve_len = BitConverter.ToUInt16(data, 2);//총길이
       cmd_switch();
       Debug.Log("rcv cmd: " + receive_cmd);//현재cmd로 바꿔놓았음
       return receiveData;
   }
   public void cmd_switch()
       switch (receive_cmd)
           case "LS":// 로그인 성공
              Login_IDscan = 1;
              break;
           case "LF"://로그인 실패
              Login_IDscan = 0;
              break;
           case "JS"://회원가입 성공
              Join_finsh = true;
              break;
```



```
case "JF":// 회원가입 실패
               Join finsh = false;
              break:
           case "UI":// 유저정보요청 커멘드
              int index_len = 5;
               int user_nick_len = (int)BitConverter.ToUInt16(data, index_len);//닉네님길이
               index len += 1;
              user_nick = System.Text.Encoding.Default.GetString(data, index_len,
(int)user_nick_len);//닉네임 스트링
               index_len += user_nick.Length + 1;
               int user_regiment_len = (int)BitConverter.ToUInt16(data, index_len);
               index len += 1;
               user_regiment = System.Text.Encoding.Default.GetString(data, index_len,
user_regiment_len);
               index_len += (user_regiment_len + 1);
               //int join_len = (int)BitConverter.ToInt32(data, index_len);
               user_join_count = (int)BitConverter.ToInt32(data, index_len);
               index_len += 4;
              user_passed_count = (int)BitConverter.ToInt32(data, index_len);
               index_len += 4;
              user_reject_count = (int)BitConverter.ToInt32(data, index_len);
              break;
       }
   }
C# - Packet 변환
패킷을 보낼 때 보편적으로 많이 쓰이는 함수. C# string 변수를 byte 배열의 원하는 위치에 변환해준다.
ushort Combine_Str(byte[] dstbuf, string str, ushort index_len)// 보내기전 바꿔주는 함수
   {
       Debug.Log("Called Combine_Str()");
           dstbuf[index_len++] = (byte)Encoding.UTF8.GetByteCount(str);
       Array.Copy(Encoding.UTF8.GetBytes(str), 0, dstbuf, index_len, dstbuf[index_len -
1]);//아이디를 컨버팅하고 0부터 버퍼에 5위치부터 메세지길이만큼채워넣어라
       index_len += dstbuf[index_len - 1];
       return index_len;
   }
```



```
이미지 로드 부분
    윈도우 DLL 사용하여 Popup을 띄우고 사진경로를 받을 수 있게 구현.
using System.Windows.Forms;
public class OpenFile
   public static string path;
   public static void Apply()
       OpenFileDialog ofd = new OpenFileDialog();
       ofd.Filter = "jpg Files | *.jpg";
       ofd.Title = "사진을 선택하세요";
       if (ofd.ShowDialog() == DialogResult.OK)
           path = ofd.InitialDirectory + ofd.FileName;
       }
   }
}
회원가입 부분
회원가입 버튼을 누르면 패킷을 생성하여 Server 로 Send 를 한다.
public void Join_ok_button()
       ushort len = 0;
       Join_input_ID = GameObject.Find("Join_input_id").GetComponent<InputField>().text;
       Join_input_PW = GameObject.Find("Join_input_PW").GetComponent<InputField>().text;
       Join_NickName = GameObject.Find("Join_input_nickname").GetComponent<InputField>().text;
       Join_Regiment = GameObject.Find("Join_input_regiment").GetComponent<InputField>().text;
       Debug.Log("input_ID : " + Join_input_ID);
       Debug.Log("input_PW : " + Join_input_PW);
       Debug.Log("input_Nick : " + Join_NickName);
       Debug.Log("input_Regiment : " + Join_Regiment);
       //커멘드 , 아이디.,패스워드, 닉네님, 소속, 총길이
       send_to_buffer = ST.Join_Get_packet("J0", Join_input_ID, Join_input_PW, Join_NickName,
```



```
Join_Regiment, ref len);
       ST.send_to_Login_server(send_to_buffer, len);// 회원정보보내는 부분
       Array.Clear(send_to_buffer, 0, send_to_buffer.Length);//배열 초기화
           ST.Join_finsh = true;
           recv_data = ST.recv_server();
           Join finsh = ST.Join finsh;
           if(Join_finsh == true)
                  ////사진 전송 //////
                  Debug.Log(path.ToString());
                   /*여기다 패스 받아서 보내주기*/
                  ST.Send_img(path, Join_input_ID);
                   string cmd = ST.receive_imgresult();
                  if(cmd == "SD")
                  {
                          Debug.Log("파일 전송 성공!");
                          Join_complete.SetActive(true);
                  }
                  else
                  {
                          Join_error.SetActive(true);
                          Debug.Log("전송 실패");
                  }
           }
           else
           {
                  Join_error.SetActive(true);
           }
   }
회원가입 패킷을 만들어서 리턴해 주는 함수.
public byte[] Join_Get_packet(string send_cmd, string send_id, string send_pw, string send_nick,
string regiment, ref ushort index_len)
   {
      Debug.Log("Called Join_Get_packet()");
       // 커맨드 , 총길이 , 아이디길이, 아이디,패스워드길이 ,패스워드, 닉네님, 소속
       Array.Copy(Encoding.Default.GetBytes(send_cmd), 0, send_buffer, 0, 2);// 커맨드를
인코딩하는데 처음부터 버퍼의 0번째위치부터 2개만큼채움
       index_len = 4;
```

페이지 195 / 261



```
index_len = Combine_Str(send_buffer, send_id, index_len);
        index_len = Combine_Str(send_buffer, send_pw, index_len);
        index_len = Combine_Str(send_buffer, send_nick, index_len);
        index_len = Combine_Str(send_buffer, regiment, index_len);
       Array.Copy(BitConverter.GetBytes(index_len), 0, send_buffer, 2, 2);// 인트형을 언사인드
쇼트로만들었으니 2바이트가 된다
       return send buffer;
   }
회원가입 시 Window DLL 로 받아온 사진의 경로로 사진을 binary 형식으로 열어 서버로 전송 해주는 함
수.
public void Send_img(string path, string U_id)
    {
            try
            {
                    FileStream fs = new FileStream(path, FileMode.Open, FileAccess.Read);
                    FileInfo fi = new FileInfo(path);
                    Debug.Log(string.Format("[서버-파일전송] : 파일명 : {0}, 크기 : {1} byte",
fi.Name, fi.Length));
                    long IngFileLength = fi.Length;
                    BinaryReader br = new BinaryReader(fs);
                    byte[] imageData = new byte[1024 * 1024 * 4];
                    byte[] SendData = new byte[1024 * 1024 * 4 + 100];
                    br.Read(imageData, 0, imageData.Length);
                    Array.Copy(System.Text.Encoding.UTF8.GetBytes("SI"), 0, SendData, 0, 2);
                    Array.Copy(BitConverter.GetBytes(IngFileLength + 7 + U_id.Length), 0,
SendData, 2, 4);
                    Array.Copy(BitConverter.GetBytes((byte)U_id.Length), 0, SendData, 6, 1);
                    Array.Copy(System.Text.Encoding.UTF8.GetBytes(U_id), 0, SendData, 7,
U id.Lenath);
                    Array.Copy(imageData, 0, SendData, 7 + U_id.Length, IngFileLength);
                    Mysend(SendData, (int)IngFileLength + 7 + U_id.Length);
            }
            catch (SocketException ex)
```



```
{
                   Debug.Log("전송 오류 : " + ex.ToString());
           }
   }
서버에 회원정보부터 사진까지 모두 잘 송신이 됬다면 회원가입 완료를 판단할 함수.
public string receive_imgresult()
           receiveData = "";
           int bytes = stream.Read(data, 0, data.Length);
           while (bytes < 4)</pre>
                   bytes += stream.Read(data, bytes, data.Length - bytes);
           string cmd = System.Text.Encoding.Default.GetString(data, 0, 2);
           Debug.Log("rcv : " + cmd);
           if (cmd == "SD")
           {
                   Debug.Log("사진 전송 성공");
                   return cmd;
           }
           else
           {
                   Debug.Log("실패");
                   return cmd;
           }
    }
Lobby
Recv
Lobby Scene 에서 서버에서 받은 Recv Data 를 처리할 함수.
Login, Join 과는 다른 소켓 사용.
public string receive()
                                페이지 197 / 261
```



```
{
    receiveData = "";
        while(len_recv < SIZE_HEADER || len_recv < BitConverter.ToUInt16(data, 2))</pre>
                len_recv += csSingleton.singleton.WorkSock.Receive(data, 0, data.Length, 0);
        }
        int len_pack = BitConverter.ToUInt16(data, 2);
        while(len_recv >= SIZE_HEADER && len_recv >= BitConverter.ToUInt16(data, 2))
        {
                len_pack = BitConverter.ToUInt16(data, 2);
                cmd_switch();
                len_recv -= len_pack;
                Array.Copy(data, len_pack, data, 0, len_recv);
   Debug.Log("rcv cmd: " + receive_cmd);//현재cmd로 바꿔놓았음
    return receiveData;
}
public void cmd_switch()
        receive_cmd = System.Text.Encoding.Default.GetString(data, 0, 2);
        switch (receive_cmd)
       case "LS":// 로그인 성공
           Login_IDscan = 1;
           break;
       case "LF"://로그인 실패
           Login_IDscan = 0;
           break;
       case "JS"://회원가입 성공
           Join_finsh = true;
           break;
       case "JF":// 회원가입 실패
           Join finsh = false;
           break;
       case "UI":// 유저정보요청 커멘드
           int index_len = 4;
           int user_ID_len = (int)data[index_len];//아이디길이
           index_len += 1;
```

페이지 198 / 261



```
string user_ID = System.Text.Encoding.UTF8.GetString(data, index_len,
(int)user_ID_Ien);//아이디 스트링
               index_len += user_ID_len;
               int user_nick_len = (int)data[index_len];//닉네님길이
               index_len += 1;
               user_nick = System.Text.Encoding.UTF8.GetString(data, index_len,
(int)user_nick_len);//닉네임 스트링
               index_len += user_nick_len;
               int user_regiment_len = (int)data[index_len];//소속길이
               index len += 1;
               user_regiment = System.Text.Encoding.UTF8.GetString(data, index_len,
user_regiment_len);//소속스트링
               index_len += (user_regiment_len);
               //int join_len = (int)BitConverter.ToInt32(data, index_len);
               user_join_count = (int)BitConverter.ToInt32(data, index_len);
               index_len += 4;
               user_passed_count = (int)BitConverter.ToInt32(data, index_len);
               index len += 4;
               user_reject_count = (int)BitConverter.ToInt32(data, index_len);
               break;
           case "FI"://아이디로찾기
               index_len = 4;
               int find_ID_len = (int)data[index_len];//아이디길이
               index_len += 1;
               find_ID = System.Text.Encoding.Default.GetString(data, index_len,
(int)find_ID_Ien);//아이디스트링
               index_len += find_ID.Length;
               int find_nick_len = (int)data[index_len];//닉네님길이
               index_len += 1;
               find_nick = System.Text.Encoding.Default.GetString(data, index_len,
(int)find_nick_len);//닉네임 스트링
               index_len += find_nick.Length;
               int find_regiment_len = (int)data[index_len];//소속길이
               index_len += 1;
               find_regiment = System.Text.Encoding.Default.GetString(data, index_len,
```



```
find_regiment_len);//소속스트링
               index_len += find_regiment.Length;
               find_join_count = (int)BitConverter.ToInt32(data, index_len);//참여횟수
               index_len += 4;
               find_passed_count = (int)BitConverter.ToInt32(data, index_len);// 가결횟수
               index_len += 4;
               find_reject_count = (int)BitConverter.ToInt32(data, index_len);//부결횟수
               find_error = false;
               break:
           case "FN":// 닉네임으로찾기
               index_len = 4;
               find_ID_len = (int)data[index_len];//아이디길이
               index_len += 1;
               find_ID = System.Text.Encoding.Default.GetString(data, index_len,
(int)find_ID_len);//아이디스트링
               index_len += find_ID.Length;
               find_nick_len = (int)data[index_len];//닉네님길이
               index len += 1;
               find_nick = System.Text.Encoding.Default.GetString(data, index_len,
(int)find_nick_len);//닉네임 스트링
               index len += find nick.Length;
               find_regiment_len = (int)data[index_len];//소속 길이
               index_len += 1;
               find_regiment = System.Text.Encoding.Default.GetString(data, index_len,
find_regiment_len);//소속 스트링
               index_len += find_regiment.Length;
               //int join_len = (int)BitConverter.ToInt32(data, index_len);
               find_join_count = (int)BitConverter.ToInt32(data, index_len);//참여횟수
               index_len += 4;
               find_passed_count = (int)BitConverter.ToInt32(data, index_len);//가결 횟수
               index_len += 4;
               find_reject_count = (int)BitConverter.ToInt32(data, index_len);//부결횟수
               find_error = false;
               break:
```



```
case "ED"://에러디비
               find_error = true;
               break;
           case "F0":// 친구아이디보내기 성공
               friend_add_success = true;
               break;
           case "F9"://친구아이디보내기 실패
               friend_add_success = false;
               break;
           case "GF"://친구목록 성공시
               index_len = 4;
               int friend_name_len = 0;
               int big_len = (int)BitConverter.ToUInt16(data,2);// 총길이 2바이트만 읽음
               for (int i = 0; index_len < big_len; i++)</pre>
                   friend_name_len = (int)data[index_len];//아이디길이
                   index_len += 1;
                   friend_names[i] = Encoding.Default.GetString(data, index_len,
(int)friend_name_len);//친구 아이디 1개 저장
                   index_len += friend_name_len;
               break;
           case "CR":
                            csSingleton.singleton.cur_room = BitConverter.ToUInt16(data,
SIZE_HEADER);
                            csSingleton.singleton.Master_type = 1;
    GameObject.Find("systemUI").GetComponent<UImanager_Robby>().SceneChange();
                            break;
                    case "ER":
                            csSingleton.singleton.Master_type = 2;
    GameObject.Find("systemUI").GetComponent<UImanager_Robby>().SceneChange();
                            break;
            }
   }
```



```
자기정보 받아오기
Lobby 에서 자기정보를 서버에 Packet 을 보내 자기정보를 얻어온다.
public void Robby_Myinfo_btn()
   {
       Robby Coll DB("UI");
       //ID = ST.user_ID;
       nick_name = ST.user_nick;
       regiment = ST.user_regiment;
       Join_count_int = ST.user_join_count;
       Passed_count_int = ST.user_passed_count;
       reject_count_int = ST.user_reject_count;
       Debug.Log("ID"+ csSingleton.singleton.id);
       Debug.Log("nick" + nick_name);
       Debug.Log("regiment" + regiment);
       Debug.Log("join_count" + Join_count_int);
       Debug.Log("passed" + Passed_count_int);
       Debug.Log("reject" + reject_count_int);
       Robby_Myinfo.SetActive(true);
            Image_Load_via_Net(csSingleton.singleton.id);
            GameObject.Find("My_ID_text").GetComponent<Text>().text = csSingleton.singleton.id;
       GameObject.Find("My_nick_text").GetComponent<Text>().text = nick_name;
       GameObject.Find("My_Regiment_text").GetComponent<Text>().text = regiment;
       GameObject.Find("My_passed_count").GetComponent<Text>().text =
Passed_count_int.ToString();
       GameObject.Find("My_reject_count").GetComponent<Text>().text =
reject_count_int.ToString();
       GameObject.Find("My_join_count").GetComponent<Text>().text = Join_count_int.ToString();
   }
Server 에서 Image 를 로드 받는 함수.
Image 로드 패킷을 만들어 보내고 Recv 를 받는다.
```

void Image_Load_via_Net(string id)



```
{
            ST.request_img(id);
            string cmd = ST.recv_img();
            if (cmd == "ED")
            {
                    file_Name = null;
                    Debug.Log("파일 수신 실패");
            }
            else
            {
                    file_Name = cmd;
                    Debug.Log("파일 수신 성공 // 파일이름 : {0}" + file_Name);
            }
            ////////내정보 이미지 불러오기/////////
            Image i = GameObject.Find("Myimage").GetComponent<Image>();
            Sprite s;
            if (file_Name != null)
                    s = IMG2Sprite.LoadNewSprite(csSingleton.singleton.Path_Image + file_Name);
                    i.sprite = s;
            }
    }
//친구 혹은 개인 프로필 사진 받는 부분
   byte[] imageData = new byte[1024 * 1024 * 4];
   public string recv_img()
    {
            receiveData = "";
            int bytes = csSingleton.singleton.WorkSock.Receive(imageData, 0, imageData.Length, 0);
            while (bytes < 2)</pre>
                    bytes += csSingleton.singleton.WorkSock.Receive(imageData, bytes,
imageData.Length - bytes, 0); //전체 길이 - cmd 길이??!!
            string cmd = System.Text.Encoding.Default.GetString(imageData, 0, 2);
            Debug.Log("rcv : " + cmd);
            if (cmd == "LI")
            {
                    while (bytes < 6)
                            bytes += csSingleton.singleton.WorkSock.Receive(imageData, bytes,
imageData.Length - bytes, 0); //전체 길이 - cmd 길이??!!
```



```
int packet_len = System.BitConverter.ToInt32(imageData, 2);
                    Debug.Log(packet_len.ToString());
                   while (bytes < packet_len)</pre>
                           bytes += csSingleton.singleton.WorkSock.Receive(imageData, bytes,
imageData.Length - bytes, 0); //전체 길이 - cmd 길이??!!
                   Debug.Log("사진 수신 성공");
                   byte id_len = imageData[6];
                   string id = System.Text.Encoding.UTF8.GetString(imageData, 7, id_len);
                    string PATH = csSingleton.singleton.Path_Image + id + ".jpg";
                    FileStream fs = new FileStream(PATH, FileMode.Create, FileAccess.ReadWrite);
                   FileInfo fi = new FileInfo(PATH);
                   Debug.Log(string.Format("[서버-파일수신] : 파일명 : {0}, 크기 : {1} byte",
fi.Name, fi.Length));
                    long IngFileLength = fi.Length;
                    BinaryWriter bw = new BinaryWriter(fs);
                    bw.Write(imageData, 7 + id_len, packet_len - (7 + id_len));
                    fs.Close();
                    Debug.Log("사진 수신 성공");
                    return fi.Name;
            }
            else
            {
                   Debug.Log("사진 수신 실패");
                    return cmd;
            }
    }
Image -> Sprite 변환
Server 에서 받은 jpg 파일을 Unity 에서 사용할 Sprite 로 동적 변환 해줄 클래스.
using System.10;
public class IMG2Sprite : MonoBehaviour
                                  페이지 204 / 261
```



```
{
    // This script loads a PNG or JPEG image from disk and returns it as a Sprite
    // Drop it on any GameObject/Camera in your scene (singleton implementation)
    //
    // Usage from any other script:
    // MySprite = IMG2Sprite.instance.LoadNewSprite(FilePath, [PixelsPerUnit (optional)])
    private static IMG2Sprite _instance;
    public static IMG2Sprite instance
        get
            //If _instance hasn't been set yet, we grab it from the scene!
            //This will only happen the first time this reference is used.
            if (_instance == null)
                _instance = GameObject.FindObjectOfType<IMG2Sprite>();
            return _instance;
        }
    }
    public static Sprite LoadNewSprite(string FilePath, float PixelsPerUnit = 100.0f)
        // Load a PNG or JPG image from disk to a Texture2D, assign this texture to a new sprite and
return its reference
        Sprite NewSprite = new Sprite();
        Texture2D SpriteTexture = LoadTexture(FilePath);
            if(SpriteTexture != null)
                     NewSprite = Sprite.Create(SpriteTexture, new Rect(0, 0, SpriteTexture.width,
SpriteTexture.height), new Vector2(0, 0), PixeIsPerUnit);
        return NewSprite;
    }
    public static Texture2D LoadTexture(string FilePath)
        // Load a PNG or JPG file from disk to a Texture2D
        // Returns null if load fails
        Texture2D Tex2D;
        byte[] FileData;
```



```
if (File.Exists(FilePath))
          FileData = File.ReadAllBytes(FilePath);
          Tex2D = new Texture2D(2, 2);  // Create new "empty" texture
          if (Tex2D.LoadImage(FileData))
                                              // Load the imagedata into the texture (size
is set automatically)
                                        // If data = readable -> return texture
              return Tex2D;
       }
                                   // Return null if load failed
       return null;
   }
}
C# - Packet 변환 [CMD : LENGTH : ID]
Client 에서 자주 사용하는 패킷을 만들어주는 함수.
public byte[] Get_cmd_id_packet(string send_cmd, string ID, ref ushort len)
   {
       Debug.Log("Get_cmd_id_packet called)");
       // 커맨드 , 총길이
       Array.Copy(Encoding.Default.GetBytes(send_cmd), 0, send_buffer, 0, 2);// 커맨드를
인코딩하는데 처음부터 버퍼의 0번째위치부터 2개만큼채움
       len = 4;
       len = Combine_Str(send_buffer, ID, len);
       Array.Copy(BitConverter.GetBytes(Ien), 0, send_buffer, 2, 2);// 인트형을
언사인드쇼트로만들었으니 2바이트가 된다
       return send_buffer;
   }
유저찾기
해당 유저의 Id 를 Server 로 보내 해당 Id 의 정보를 받을 수 있는 함수.
public void Robby_find_IDFind_btn()
       find_ID = GameObject.Find("Find_ID").GetComponent<InputField>().text;
       ushort len = 0;
```

페이지 206 / 261



```
send_buffer = ST.Get_cmd_id_packet("FI", find_ID, ref len);
       ST.send_to_Login_server(send_buffer, len);
       ST.receive();
       Array.Clear(send_buffer, 0, send_buffer.Length);//배열 초기화
        find_info = !ST.find_error;
       if (find_info)
       {
           find_ID = ST.find_ID;
           find_nick = ST.find_nick;
           find_regiment = ST.find_regiment;
           find_join_count_int = ST.find_join_count;
            find_passed_count_int = ST.find_passed_count;
            find_reject_count_int = ST.find_reject_count;
           Robby_find_ok.SetActive(true);
                    Image_Load_via_Net(find_ID);
                    GameObject.Find("Find_ID_text").GetComponent<Text>().text = find_ID;
           GameObject.Find("Find_nick_text").GetComponent<Text>().text = find_nick;
           GameObject.Find("Find_Regiment_text").GetComponent<Text>().text = find_regiment;
           GameObject.Find("Find_passed_count").GetComponent<Text>().text =
find_join_count_int.ToString();
           GameObject.Find("Find_reject_count").GetComponent<Text>().text =
find passed count int.ToString();
           GameObject.Find("Find_join_count").GetComponent<Text>().text =
find_reject_count_int.ToString();
       }
       else
       {
           Robby_find_error.SetActive(true);
       }
       Debug.Log("findID : " + find_ID);
방정보 버튼 생성
방 버튼을 Prev 와 Next 버튼을 통해 다음 방정보를 보여주기 위해 방버튼을 6개씩 Panel로 관리할 변수
List<GameObject> IRoomPagePanel = new List<GameObject>();
모든 방 버튼을 관리할 변수.
List<Button> IRoomBtn = new List<Button>();
                                   페이지 207 / 261
```

etComponent<Text>().text,



```
그리고 서버에서 방정보를 받아올 변수.
   RoomInfo[] arrRoomInfo = new RoomInfo[MAX ROOMPAGE * MAX ROOM];
방 번호와 방정보변수의 인덱스가 다르기 때문에 필요할 때마다 모든 방정보를 검색하지 않게
방번호와 방정보변수의 인덱스를 Mapping 시켜줄 변수 생성
   int[] mapRoomIdx = new int[MAX ROOM MAP];
   int RoomCnt = 0;
   int CurRoomPage = 0;
   방 버튼과 패널을 Lobby Scene 으로 넘어 올때 맥시멈 까지 생성. 각 방버튼에 대한 여러가지
   동작 이벤트도 동적으로 생성 및 각버튼과 연결 해준다.
   void Start()
       {
           spriteStage[1] = Resources.Load<Sprite>("Image/3_Lib");
           spriteStage[2] = Resources.Load<Sprite>("Image/4_Cafe");
           spriteStage[3] = Resources.Load<Sprite>("Image/5_Pali");
           GameObject canvas lobby = GameObject.Find("Canvas Lobby");
               for (int i = 0; i < MAX_ROOMPAGE; i++)</pre>
               {
                       GameObject RoomPage = Instantiate(PanelRoomPagePref);
                       RoomPage.name = "panelRoomPage_" + i;
       RoomPage.GetComponent<RectTransform>().SetParent(canvas_lobby.GetComponent<RectTransform>()
   );
                       RoomPage.GetComponent<RectTransform>().anchorMin = new
   Vector2(panelRoomPageAnchor[0], panelRoomPageAnchor[1]);
                       RoomPage.GetComponent<RectTransform>().anchorMax = new
   Vector2(panelRoomPageAnchor[2], panelRoomPageAnchor[3]);
                       RoomPage.GetComponent<RectTransform>().offsetMin = new Vector2(0, 0);
                       RoomPage.GetComponent<RectTransform>().offsetMax = new Vector2(0, 0);
                       RoomPage.GetComponent<RectTransform>().SetSiblingIndex(10);
                       IRoomPagePanel.Add(RoomPage);
                       for (int j = 0; j < MAX_ROOM; j++)
                       {
                              Button Btn = Instantiate(BtnRoomPref).GetComponent<Button>();
                              Btn.name = "btnRoom_" + ((i * MAX_ROOM) + j);
                              #region Event Script
                              Btn.onClick.AddListener(delegate ()
                              {
       this.ButtonClicked(Btn.transform.FindChild("Panel").FindChild("pNum").FindChild("txtNum").G
```



```
Btn.transform.FindChild("Panel").FindChild("pRoomModeName").FindChild("txtRoomModeName").GetComp
onent<Text>().text);
                          });
    EventTrigger.Entry entry = new EventTrigger.Entry();
                          entry.eventID = EventTriggerType.PointerEnter;
                          entry.callback.AddListener(delegate(BaseEventData eventData)
                          {
    this.ButtonOver(Btn.transform.FindChild("Panel").FindChild("pNum").FindChild("txtNum").GetC
omponent<Text>().text, eventData);
                          Btn.GetComponent<EventTrigger>().triggers.Add(entry);
    entry = new EventTrigger.Entry();
                          entry.eventID = EventTriggerType.PointerExit;
                          entry.callback.AddListener(ButtonExit);
                          Btn.GetComponent<EventTrigger>().triggers.Add(entry);
                          #endregion
   Btn.GetComponent<RectTransform>().SetParent(RoomPage.GetComponent<RectTransform>());
                          Btn.GetComponent<RectTransform>().anchorMin = new
Vector2(btnRoomAnchor[j,0], btnRoomAnchor[j,1]);
                          Btn.GetComponent<RectTransform>().anchorMax = new
Vector2(btnRoomAnchor[j,2], btnRoomAnchor[j,3]);
                          Btn.GetComponent<RectTransform>().offsetMin = new Vector2(0, 0);
                          Btn.GetComponent<RectTransform>().offsetMax = new Vector2(0, 0);
                          Btn.gameObject.SetActive(false);
                          IRoomBtn.Add(Btn);
                   RoomPage.SetActive(false);
           IRoomPagePanel[0].SetActive(true);
           StartCoroutine(CoUpdate_Room_sec(new WaitForSeconds(3f)));
    }
```

Stage



```
Stage_script.cs
   Stage 에서 UI에 관하여 상태를 항상 검사할 작업들을 Update에 작업.
   void Update()// 업데이트
       if (csSingleton.singleton.bEnd_Room)
          over_count();// 토론이 끝났다면 카운트 팝업뜸
                  stage_result_vote();// 결과팝업
      if (over_count_start) { Over_count_start(); }
      if (voice_count_down)// 보이스 카운트 다운이 트루가되면 카운트 함수 실행
      {
          count_down(voice_count_down);
          Debug.Log("count_down called");
      }
      if (csSingleton.singleton.bVote_Start) { ing_vote_popup(); }// 무효표 판정
   }
발언권 부분
#region 발언권
   const int TIME_SAY = 10;
   bool voice_count_down = false;
   float time_count = TIME_SAY;
   public void request_voice_btn()
          if(GameObject.Find("ID_" +
csSingleton.singleton.id).GetComponent<csAnimatorController_Mechanim>().status.bSitting
          && !GameObject.Find("ID_" +
csSingleton.singleton.id).GetComponent<csAnimatorController_Mechanim>().status.bQuestioning)
                  csKeyController.SpecialKeyDown(KeyFlag.Jump);
   }
```



```
public void Give_to_voice()// 발언 주는 부분
      //해당유저위에 버튼을 눌럿을시 유저의 닉네임을 보내 활성화?
      // 서버로 커맨드만 보내고 타이머 팝업띄우는 함수를 따로만들자
      Gain_voice();
   }
   public void Start_Voice_Cnt_Master()
          time_count = TIME_SAY;
          Timer_Popup.SetActive(true);// 60초 타이머 팝업 띄움
          voice_count_down = true; //1초마다 깍이게
   }
   public void Stop_Voice_Cnt_Master()
          Timer_Popup.SetActive(false);// 60초 타이머 팝업 띄움
          voice_count_down = false; //1초마다 깍이게
          time_count = TIME_SAY;
   }
   public void Gain_voice()// gain = 얻다
   {
          if(type_num != 1)
          {
                  time_count = TIME_SAY;
                  Timer_Popup.SetActive(true);// 60초 타이머 팝업 띄움
                  voice_count_down = true; //1초마다 깍이게
//if(발언시간 < 0){ Mute_voice(); }
   public void Stop_voice()
          if(type_num != 1)
          {
                  Timer_Popup.SetActive(false);// 60초 타이머 팝업 띄움
                                        //1초마다 깍이게
                  voice_count_down = false;
                  time_count = TIME_SAY;
          }
   }
   public void Mute_voice()// 발언권박탈
```



```
Timer_Popup.SetActive(false);
   }
   void count_down(bool count_down)// 발언시간 카운트다운
       if (count_down)
          if (time_count > 0)
              time_count -= Time.deltaTime; // 1초마다 깍이고
              GameObject.Find("Timer_txt").GetComponent<Text>().text =
((int)time_count).ToString();// 타이머 텍스트에 넣는다
          else if (time_count <= 0)// 시간을 모두 소진시 네트워크로 알려줘야함
              voice_count_down = false; // 더이상 함수에 못들어가게 하고
              time_count = TIME_SAY; // 시간을 리셋
              Timer_Popup.SetActive(false);
   GameObject.Find("AudioManager").GetComponent<csMicNetwork>()._Process_MicStop();
       }
   }
   #endregion
csNetwork_stage
Recv부분
Stage Scene 에서 Server 로부터 Recv 받은 데이터를 처리할 함수.
동작에 관련한 함수.
void RecvProcess()
   {
           int idx_start = 0;
           int idx_end = RecvLength;
           int Len_Packet = 0;
           string cmd = "";
                               페이지 212 / 261
```



```
if(RecvLength >= SIZE_HEADER)
            {
                     Len_Packet = BitConverter.ToUInt16(RecvBuf, SIZE_HEADER_CMD);
                     if(Encoding.UTF8.GetString(RecvBuf, idx_start, SIZE_HEADER_CMD) ==
CMD_LOAD_PROFILE_IMAGE)
                             Len_Packet = BitConverter.ToInt32(RecvBuf, SIZE_HEADER_CMD +
idx start);
                     //Debug.Log("____[Recv : " + RecvBuf[0] + "" + RecvBuf[1] + "] [Len_Packet
= " + Len_Packet + "] [RecvLength = " + RecvLength + "] [TotalRecv=" +
csSingleton.singleton.Debug_Len_Recv + "]");
            }
            cnt while = 0;
            while(idx_end - idx_start >= SIZE_HEADER && idx_end - idx_start >= Len_Packet)
            {
                     //cnt_while++;
                     //if(MAX_COUNT_WHILE < cnt_while)</pre>
                     //{
                     //
                             csSingleton.singleton.WorkSock.Close();
                     //
                             Debug.Log("_____Recv Break [RecvLength=" + RecvLength + "]
[idx_start=" + idx_start + "] [idx_end=" + idx_end + "] [TotalRecv=" +
csSingleton.singleton.Debug_Len_Recv + "]");
                     //
                             break;
                     //}
                     cmd = Encoding.UTF8.GetString(RecvBuf, idx_start, SIZE_HEADER_CMD);
                     switch(cmd)
                     {
                              case CMD_LOAD_PROFILE_IMAGE:
                                      Len_Packet = recv_img(RecvBuf, idx_start);
                                      break;
                              case CMD_START_ROOM:
                                      _ProcessStartRoom();
                                      break;
                              case CMD_KEY_PACKET:
                                      _ProcessKeyPacket(idx_start, Len_Packet);
                                      break;
                              case CMD_ENTER_ROOM:
                                      _ProcessEnterRoom(idx_start);
                                      break;
                              case CMD LEAVE ROOM:
                                      _ProcessLeaveRoom(idx_start);
                                      break;
                                    페이지 213 / 261
```



```
case CMD_GET_POS_STATUS_PACKET:
                                      _ProcessGetPosStatusPacket(idx_start, Len_Packet);
                                      break;
                             default:
                                      UIRecvProcess(RecvBuf, idx_start, idx_end, cmd);
                     idx_start += Len_Packet;
                     if(idx_end - idx_start >= csConstValue.Voice.SIZE_HEADER_VOICE)
                     {
                             Len_Packet = BitConverter.ToUInt16(RecvBuf, SIZE_HEADER_CMD +
idx_start);
                             if(Encoding.UTF8.GetString(RecvBuf, idx_start, SIZE_HEADER_CMD) ==
CMD_LOAD_PROFILE_IMAGE)
                                      Len_Packet = BitConverter.ToInt32(RecvBuf, SIZE_HEADER_CMD +
idx_start);
                             //Debug.Log("____[Recv : " + RecvBuf[idx_start] + "" + RecvBuf[1 +
idx_start] + "] [Len_Packet = " + Len_Packet + "] [idx_start = " + idx_start + "] [RecvLength = "
+ RecvLength + "] [TotalRecv=" + csSingleton.singleton.Debug_Len_Recv + "]");
                     }
            RecvLength -= idx_start;
            Array.Copy(RecvBuf, idx_start, RecvBuf, 0, idx_end - idx_start);
    }
비처리에 관한 함수가 여기 들어간다.
void UIRecvProcess(byte[] RecvBuf, int idx_start, int idx_end, string CMD)
    {
            int idx = 0;
            idx = idx_start;
            string str = Encoding.UTF8.GetString(RecvBuf, idx + 1, RecvBuf[idx]);
            idx += 1 + RecvBuf[idx];
            switch(CMD)
                     case success_cmd:
                             Debug.Log("success_cmd" + CMD);
                             break;
                     case error_cmd:
                                    페이지 214 / 261
```



```
Debug.Log("ERROR_cmd" + CMD);
                         break;
                case vote_start:
                        Debug.Log("success_cmd" + CMD);
GameObject.Find("Ulmanager").GetComponent<Stage_script>().vote_ing_SetTrue();
                        break;
                case vote_agree:
                        Debug.Log("success_cmd" + CMD);
                        break;
                case vote_disagree:
                        Debug.Log("success_cmd" + CMD);
                        break;
                //case vote_nullity:
                      Debug.Log("success_cmd" + CMD);
                //
                //
                      break;
                case vote_result:
                        {
                                 // cmd 총길이 길이[1] 찬성 길이 [1]반대 결과길이 결과
                                 int agree = 0;
                                 int disagree = 0;
                                 int index_len = idx_start + 4;
                                 int nullity = 0;
                                 string result;
                                 agree = RecvBuf[index_len];
                                 index_len += 1;
                                 disagree = RecvBuf[index_len];
                                 index_len += 1;
                                 nullity = RecvBuf[index_len];
                                 if(agree > disagree + nullity)
                                         result = "가결!";
                                 else
                                         result = "부결";
                                 csSingleton.singleton.Total_Agree = agree; // 찬성 저장
                                 csSingleton.singleton.Total_disagree = disagree;// 반대
                                 csSingleton.singleton.Total_nullity = nullity;// 무효 저장
                                 csSingleton.singleton.total_result = result;
                                 csSingleton.singleton.bEnd_Room = true;
```

저장





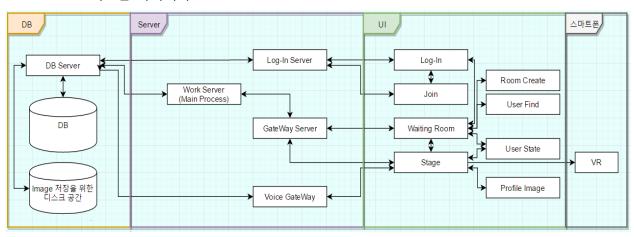
B. 사용 주요 기술

#	이름	사용처			
1	C++	IOCP Server 제작			
		① Client 에서 전달 받은 Voice data 각 Client에 전달			
		② Client 에서 입력 받은 정보 Database에 전달			
		③ Client Animation 좌표 각 Client에 전달 및 동기화			
2	C#	Client 제작			
		① Unity 내부 code 작성			
3	Unity	Client 표준 개발 도구			
		① Client 개발			
4	MySQL	Database 개발 및 제어			
		① Database Table 제작			
		② Database Data Input/Output			
		3 Client profile image save			
		④ Client state save			
5	Blender	Client UI 제작			
		① Client map 제작			
6	SketchUp	Client UI 제작			
		① Client map 제작			
7	3D Max	Client UI 제작			
		① Client map 제작			
8	PhotoShop	Client UI 제작			
		① Client Button 제작			
9	Illustration	Client UI 제작			
		① Client Button 제작			
10	TrinusVR	Virtual reality 도구			
		① Smart phone을 VR 기기 전환 툴			
		② PC화면 Smart phone으로 VR 영상 송신			



7. <u>상세 개발 내용</u>

A. 시스템 아키텍처





B. 시스템 설계 및 구현

i. 기능정의리스트

1. Client

대분류	중간항목	상세항목	항목번호	설명	담당자	순위
Client	준비	Unity 환경 설정	CL-RD-01	Unity 개발 환경을 설정하고 모델링 툴을 선별 한다.	강승리	1
		SketchUp 과 Unity 연동	CL-RD-01	SketchUp 에서 모델링을 하여 Unity에서 마음대로 디자인하고 출력할 수 있게끔 한다.	강승리	1
	모델링	Stage Modeling	CL-MD-01	국회, 카페, 도서관, 비정상회담 맵을 모델링한다.	강승리	1
		Character Modeling	CL-MD-02	Stage 안에서 보여질 케릭터를 모델링 한다.	강승리	1
		Image Texture	CL-MD-03	프로필 사진에 얼굴이 있으면 케릭터 얼굴에 텍스쳐링을 할 수 있도록 구현한다.	강승리	3
	Ui	로그인 화면 구성	CL-UI-01	로그인 화면을 구성 및 구현한다.	강승리	1
		회원가입 화면 구성	CL-UI-02	회원가입 화면을 구성 및 구현한다.	강승리	1
		로비 화면 구성	CL-UI-03	로비 화면을 구성 및 구현한다.	강승리	1
		유저 상태 화면 구성	CL-UI-04	유저 상태 화면을 구성 및 구현한다.	강승리	1
		프로필사진 창 구성	CL-UI-05	프로필 사진 창을 구성 및 구현한다.	강승리	1
		투표 창 구성	CL-UI-06	투표 창을 구성 및 구현한다.	강승리	1
		결과 창 구성	CL-UI-07	결과 창을 구성 및 구현한다.	강승리	1
		상대방정보 창 구성	CL-UI-08	상대방 정보 창을 구성 및 구현한다.	강승리	1
		영어 UI 구성	CL-UI-09	각 UI의 text를 영어로 바꿀 수 있는 버튼과 영어 텍스트로 번역 및 구현한다.	강승리	3



		1	T	,	
	일본어 UI 구성	CL-UI-10	각 UI의 text를 일본어로 바꿀	강승리	3
			수 있는 버튼과 일본어		
			텍스트로 번역 및 구현한다.		
	방정보 UI 구성	CL-UI-11	마우스가 방위에 있으면	강승리	2
			마우스를 따라다니면서		
			방정보를 보여주는 창을 구성		
			및 구현한다.		
출력	벽 트리너스 VR	CL-IO-01	방안에 들어가면 VR로 뿌려줄	강승리	1
	설정		트리너스VR를 설정하고		
			구현한다.		
UX	Unity Voice 통신	CL-UX-01	음성을 받아 송신할 수 있는	손민수	1
	· 구축		방법을 설계하고 구축한다.		
	Unity Scene	CL-UX-02	로그인화면 - 회원가입화면 -	손민수	1
	아키텍처 설계		로비화면 – Stage 로 Scene		
			년 남어가는 구조를 설계 한다.		
	Unity Scene	CL-UX-03	설계된 Scene 아키텍처를	손민수	1
	아키텍처 구축		구축한다.		
	Unity UI	CL-UX-04	구현된 UI가 동작할	손민수	1
	아키텍처 설계		아키텍처를 설계한다.		
	Unity UI	CL-UX-05	설계된 UI 아키텍처대로 Unity	손민수	1
	아키텍처 구축		Client에서 구축한다.		
	Unity 통신 구축	CL-UX-06	Unity에서 서버로의 통신	손민수	1
			방법을 설계하고 구축한다.		
	Character	CL-UX-07	케릭터 이동 및 앉기, 손들기를	손민수	1
	애니메이션 구현		애니메이션으로 구현 및 설계		
			구축한다.		
	마우스 호버	CL-UX-08	로비 화면에서 마우스가	손민수	2
	구현		방위에 있을 때 이벤트를		
			발생시키는 방법을 구축한다.		
VR	Trinus VR 환경	CL-VR-01	Trinus VR 환경을 구축한다.	손민수	3
	구축				
	Trinus Vr, Client	CL-VR-02	Client를 Trinus VR을 이용하여	손민수	3
	연동		사용할 수 있게 한다.		



2. Server

대분류	중간항목	상세항목	항목번호	설명	담당자	순위
Server	DB	UserInfo Table 설계	SV-DB-01	유저정보를 관리하고 저장할 Table 을 설계한다.	임지훈	1
		RoomInfo Table 설계	SV-DB-02	방정보를 관리하고 저장할 Table 을 설계하고 유저정보와 합쳐 정규화를 진행한다.	임지훈	1
		UserInfo Table 생성	SV-DB-03	유저정보 Table 을 Workbench에서 생성한다.	임지훈	1
		RoomInfo Table 생성	SV-DB-04	방정보 Table 을 Workbench에서 생성한다.	임지훈	1
	Disk	lmage File 저장할 Disk 설정	SV-DS-01	Image File을 저장할 Disk 공간을 지정하고 저장 방법을 구축한다.	임지훈	1
		TCP를 이용한 파일전송 구축	SV-DS-02	TCP를 이용하여 Image 파일을 송수신할 방법을 구축하고 구현한다.	임지훈	1
	Voice Network	Voice 통신 설계	SV-VN-01	음성 통신 방법을 익히고 설계한다.	손민수	1
		Voice GateWay 구축	SV-VN-02	음성 통신을 관리할 Voice GateWay를 설계하고 구축한다.	손민수	1
		음성 분석 구현	SV-VN-03	음성분석을 구현하여 사용자가 음성만으로 토론에서 손을 들고 투표를 하는 UX와 연동한다.	손민수	3
	서버 구축	DB 서버 구축	SV-BD-01	DB 서버를 설계하고 구축한다.	임지훈	1
		WorkServer 구축	SV-BD-02	WorkServer 를 설계하고 구축한다.	임지훈	1
		GateWay 구축	SV-BD-03	사용자가 Server에 바로 접근을 하지 못하도록 GateWay를 설계 및 구축한다.	임지훈	1
		LogIn Server 구축	SV-BD-04	사용자가 가장 처음 접속할 서버인 LogIn Server를 설계 및 구축한다.	임지훈	1
		서버 통합	SV-BD-05	구축한 서버들을 통합하고 Client와 통신한다.	임지훈	1
		Work Server Manager 구현	SV-BD-06	다중 WorkServer를 관리할 프로그램을 설계 및 구현한다.	임지훈	3
		Multy Work Server 구현	SV-BD-07	다중 Work Server를 설계 및 구현한다.	임지훈	3



ii. UI 구현

● 전체 UI 레이아웃

구성	UI
Login Page Client 초기 접속 시 보여지는 화면 ID, Password 입력 후 Login하면 lobby 로 이동하고 join을 누르면 join 화면으로 이동함	TOGETI Input your ID Input your PASSHORD Input y

Join Page

Client를 이용하기 위한 User Join 페이지





Lobby Page

Lobby Page의 전체적인 화면



친구 찾기

아이디나 닉네임으로 친구를 찾을 수 있음





검색 친구 창

검색해서 찾은 친구를 추가 할 수 있음



친구 창 및 정보

친구 창에서 아이디 클릭 시 정보를 볼수 있음.





방 설명

로비에서 타 유저가 생성한 방에 마우 스 호버 하면 방 설명이 나옴



방 만들기

방을 생성할 수 있음





이슈 정보

화제가 된 이슈를 볼 수 있음



친구 창 및 정보

친구 창에서 아이디 클릭 시 정보를 볼수 있음.





Stage(Café)

본격적으로 토론을 할 수 있는 Stage.



Stage(Library)

본격적으로 토론을 할 수 있는 Stage.





Stage(Parliament)

본격적으로 토론을 할 수 있는 Stage.



Selcet Music

Stage 내부에서 대기시간 동안 배경음 악을 선택해서 들을 수 있음.





투표 창

토론 종료 직전 Master가 투표를 등록 하여 투표하는 창..



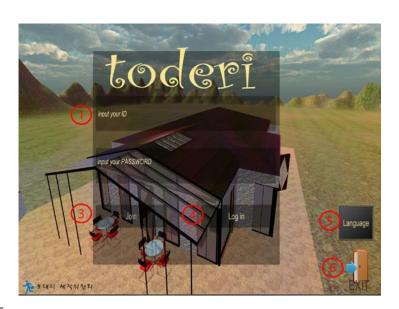
투표 결과 창

투표 종료 후 결과를 알려주는 창.



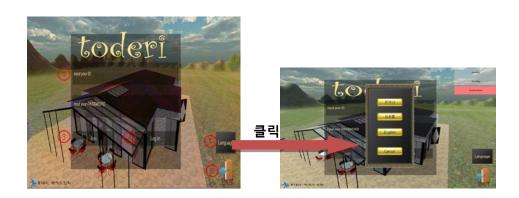


- 세부 기능 UI 레이아웃
 - 1) Login Page
 - i. 정의 및 구성
 - Program 실행 시 최초로 뜨는 화면(1 차 UI)
 - 1 번: Input ID
 - 2 번: Input Password
 - 3 번:Join Button
 - 4 번:Login Button
 - 5 번:Language Change Button
 - 6 번:Exit Button



- ii. 기능
 - Language Change Button : 클릭 시 한국어, 영어, 일본어를 선택







2) Join Page

- i. 정의 및 구성
 - Program 을 이용하기 위한 User 등록 Page 이다.
 - 1 번: Input ID
 - 2 번: Input Password
 - 3 번: Input Nicname
 - 4 번: Input affiliation
 - 5 번: Route Image file
 - 6 번: Search Image file
 - 7 번: 완료 Button
 - 8 번: 뒤로가기 Button



ii. 기능

- 이미지 찾기 Button 을 클릭하면 OpenFileDialog 가 열리면서 Image 를 선택할 수 있다.



페이지 232 / 261



- 3) Lobby Page
- i. 정의 및 구성
 - InGame(Stage)로 들어가기 전 화면으로 대기화면이다.
 - 각 언어모드 마다 BGM(Background Music)이 달라진다.
 - 1 번: Create room Button
 - 2 번: Logout Button
 - 3 번: Room Button
 - 4 번: Search friend Button
 - 5 번: Issue & Topic Button
 - 6 번: Friend list Button
 - 7 번: My information Button
 - 8 번: Refresh Button



ii. 기능

- 매 초마다 방 정보를 갱신해주며 방의 테두리가 붉은색일 경우 토론이 진행 중인 방임을 확인 할 수 있고 금색 테두리인 경우 토론 대기 중인 방임을 확인 할 수 있다.
- 방만들기 Button 을 클릭하면 방만들기 popup 이 표시되어 방을 만들 수 있다.





- 친구 찾기 Button 을 클릭하면 친구 찾기 popup 이 표시되며 찾은 친구를 친구목록에 등록할 수 있다.

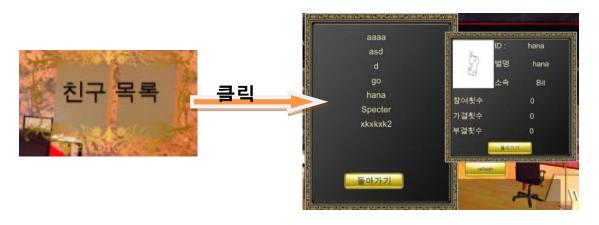


- 이슈 Button 을 클릭하면 선정한 이슈 및 화제를 popup 으로 보여준다.





- 친구목록 Button 을 클릭하면 등록한 친구들의 목록을 확인할 수 있으며 해당 ID 를 클릭 시 친구 정보 popup 이 활성화 되어 친구의 정보를 확인 할 수 있다. 또한 처음 클릭된 ID 일 경우 Profile Image 를 Server 에서 자동으로 Download 되어 저장되고 정보 창에 등록이 된다..

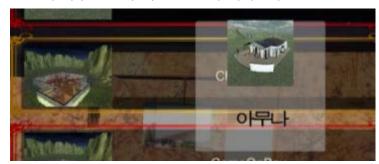


- 내정보 보기 Button 을 클릭하면 내 정보 popup 이 활성화 되어 User(자신)의 정보를





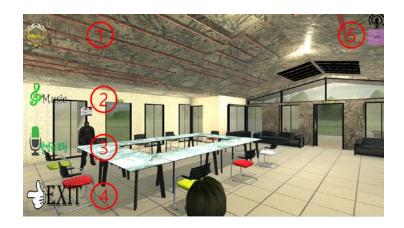
- Refresh Button 클릭 시 진행 중 혹은 대기 중인 방을 갱신 해준다.
- 방 Button 에 마우스 호버 시 방 정보가 나타난다.





4) Stage Page

- i. 정의 및 구성
 - 1. 토론을 할 수 있는 Stage page 이다.
 - 2. 1 번: Menu Button.
 - 3. 2 번: Select Music Button
 - 4. 3 번: 발언권 요청 Button
 - 5. 4 번: EXIT Button
 - 6. 5 번: VR mode connect/disconnect Button
 - 7. 6 번: 투표시작 Button







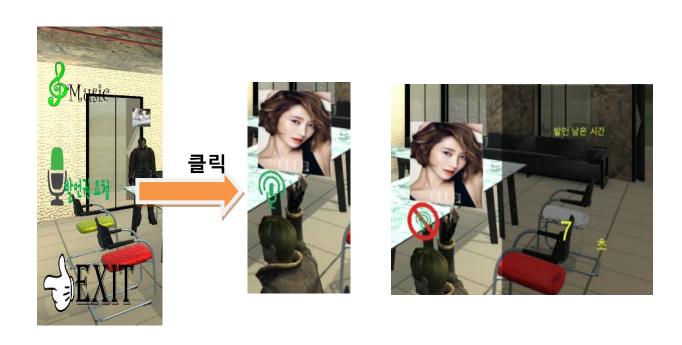
ii. 기능

- Select Music Button 을 클릭 시 배경 음악 선택 창이 나온다.



- 발언권 요청 Button 클릭(혹은 Spacebar key)시 발언권 요청 표시가 뜨고 발언권 부여 받을 시 발언권 회수 Button 이 활성화 된다.





- Master 가 투표시작 Button 클릭 시 투표 창이 활성화 되고 투표가 다끝나거나 일정 시간 도래 시 투표 결과 창이 뜬다.



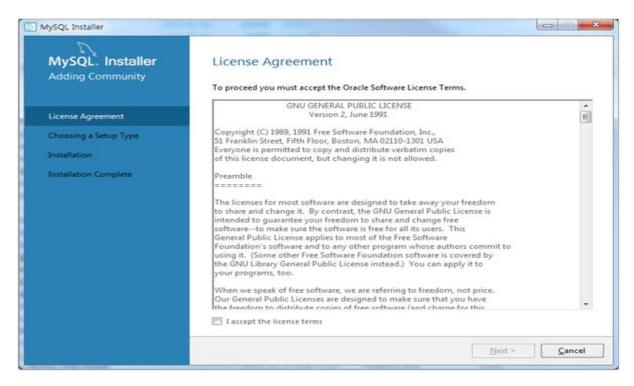
페이지 239 / 261



C. 개발환경 설치

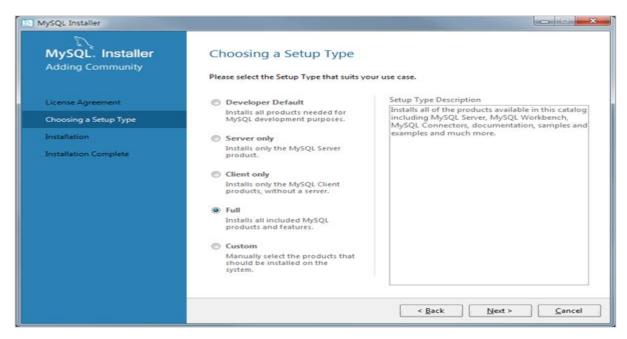
i. MySQL 설치

다운로드 받은 파일을 실행후 아래와 같은 절차로 진행합니다.



1. 처음 실행을 하면 위와 같은 화면이 실행됩니다. 라이센스 동의 체크박스를 클릭후 Next 버튼을 클릭합니다.

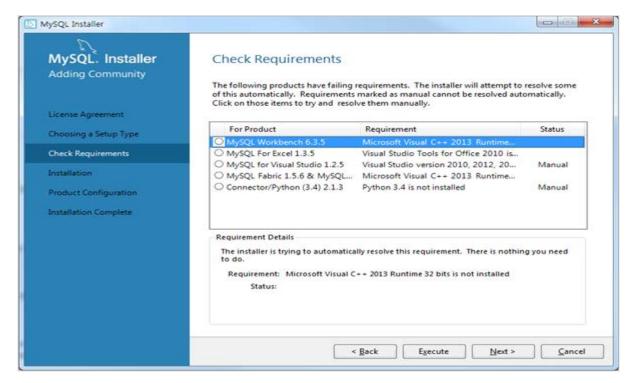




2. 사용자가 원하는 설치 방법을 선택할 수 있습니다.

처음 사용하시는 분은 Full (전체설치)를 선택하시는것이 가장 무난하므로 Full (전체설치) 선택후 Next 버튼을 클릭합니다.





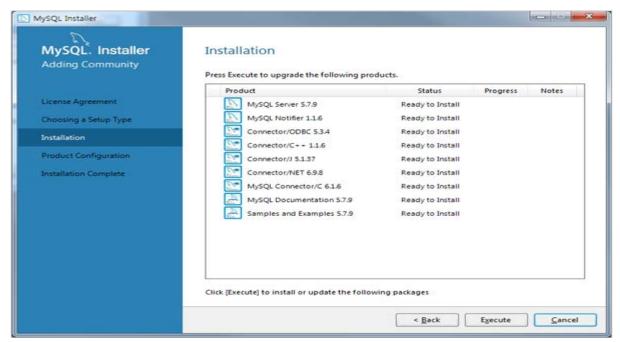
3. 프로그램 설치 요구조건에 충족치 못하는 프로그램 리스트를 나열합니다.

예를 들어 MySQL Workbench 6.3.5 프로그램을 설치하기 위해선 Virtual C++ 2013 Runtime 이 사전에 철치되어야 설치가 가능하다는 요구사항을 보여주고 있습니다.

Execute 버튼을 클릭하면 요구조건에 필요한 프로그램을 설치합니다.

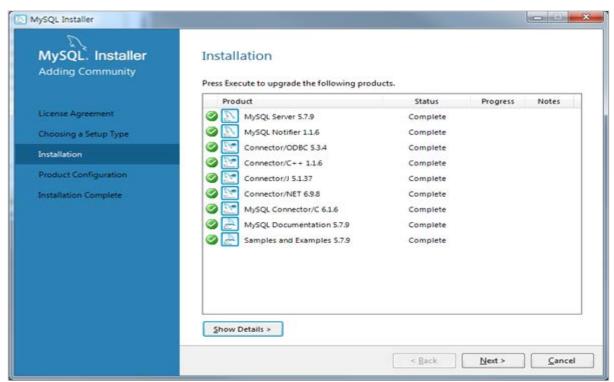
저는 위에 보이는 프로그램들은 크게 중요하지 않아 무시하고 (설치제외) Next 버튼을 클릭하여 다음화면으로 이동합니다.



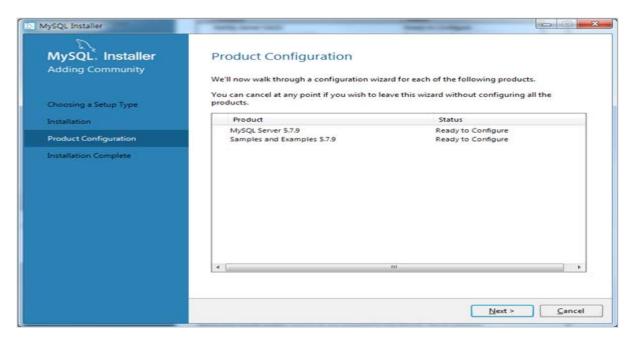


4. 최종적으로 PC에 설치되는 프로그램 리스트를 나열합니다. Execute 버튼을 클릭하면 설치를 시작합니다.





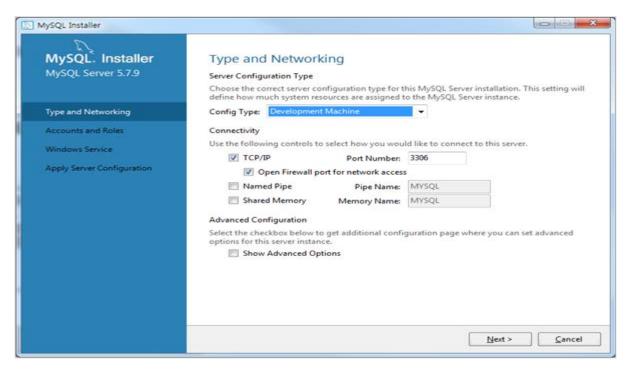
5. 모든항목에 대해 정상적으로 설치가 완료되었습니다. Next 버튼을 클릭하여 다음화면으로 이동합니다.



페이지 244 / 261



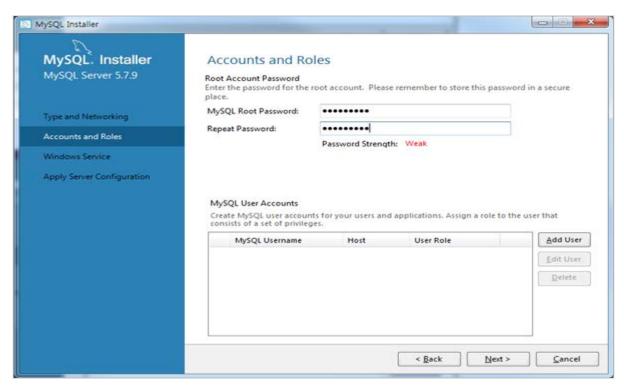
6. 프로그램 설치가 끝난후 MySQL Server 에 대한 환경설정을 해야합니다. Next 버튼을 클릭하여 다음 화면으로 이동합니다.



7. 서버의 사용목적에 따라 필요한 타입을 선택합니다.

공부목적으로 사용할 용도이므로 Development Machine 을 선택하고, 포트는 기본포트인 3306을 선택후 Next 버튼을 클릭하여 다음화면으로 이동합니다.

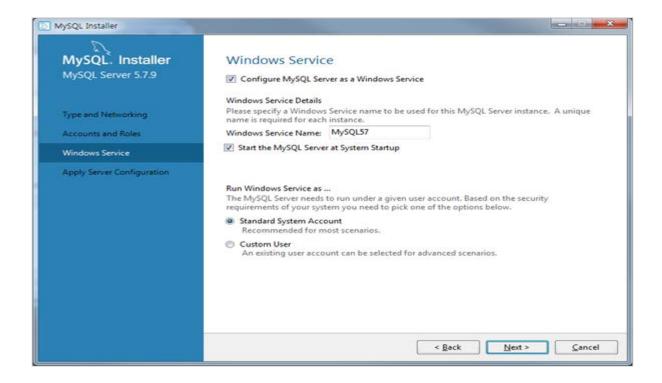




8. 최고관리자(Root) 패스워드를 지정합니다.

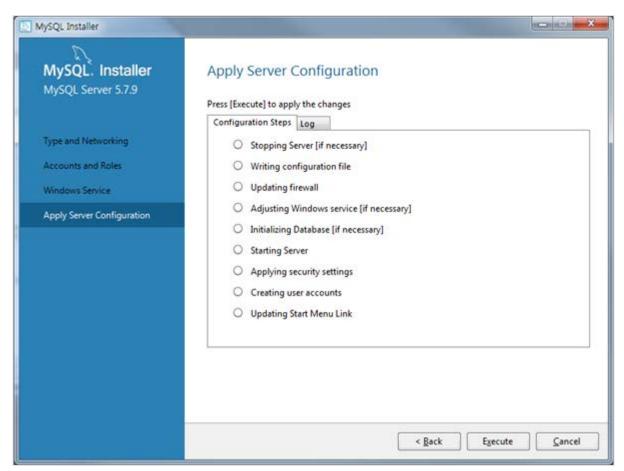
공부목적이므로 쉽게 지정해도 무방하니 알기 쉬운 패스워드를 설정후 Next 버튼을 클릭하여 다음화면으로 이동합니다.





9. 윈도우가 시작시 자동실행옵션을 지정할 수 있는 화면입니다. 원하는 값을 선택 후 Next 버튼을 클릭하여 다음화면으로 이동합니다.

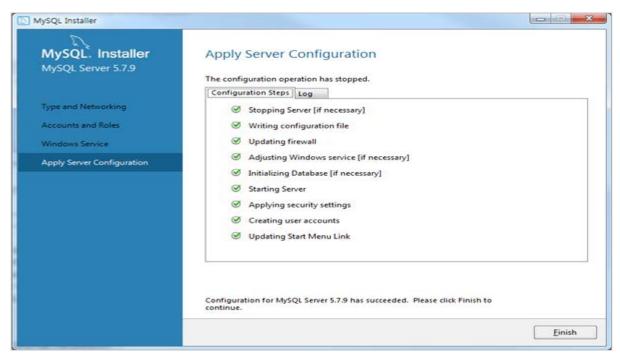




10. 최종적으로 환경설정을 적용하고, MySQL 서버를 기동하기 위한 화면입니다.

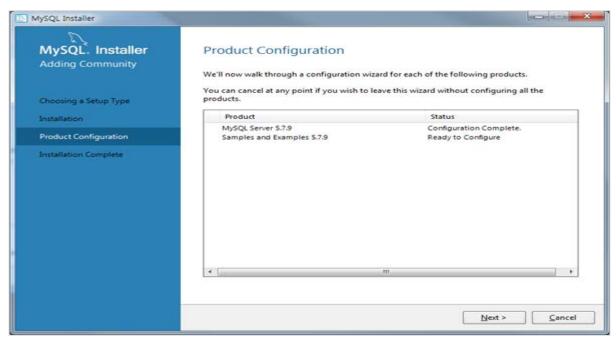
Execute 버튼을 클릭하여 MySQL 서버를 실행합니다.



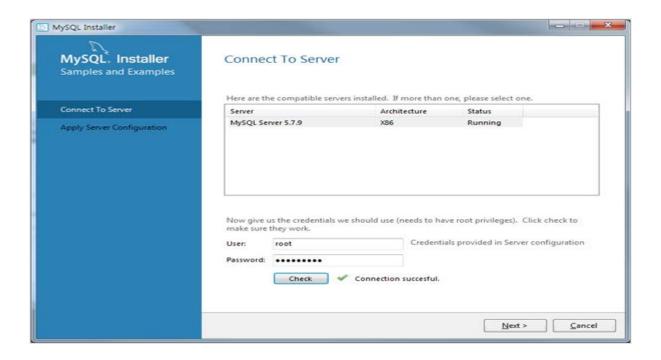


11. MySQL 서버 환경설정이 적용되어 서버가 정상적으로 실행되었습니다. Finish 버튼을 클릭하여 서버 설정화면을 종료합니다.





12. 서버가 정상적으로 실행되었으므로 접속이 가능한지 테스트를 하기 위한 단계만 남아있습니다. 서버접속 테스트를 진행하기위해 Next 버튼을 클릭하여 다음화면으로 이동합니다.

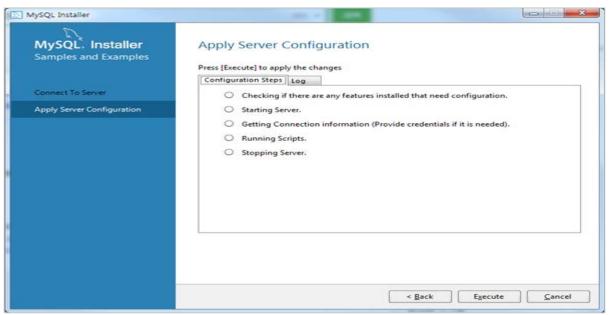


페이지 250 / 261



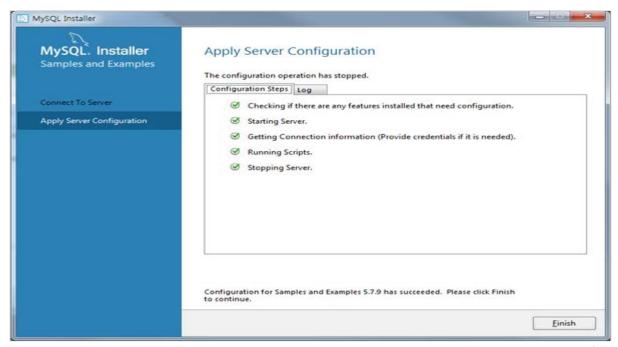
13. Status 를 보면 MySQL 서버가 가동중인것을 확인할 수 있습니다. Check 버튼을 클릭하면 Root 권한으로 성공적으로 연결 테스트가 되는것을 확인 할 수 있습니다.

Next 버튼을 클릭하여 다음화면으로 이동합니다.

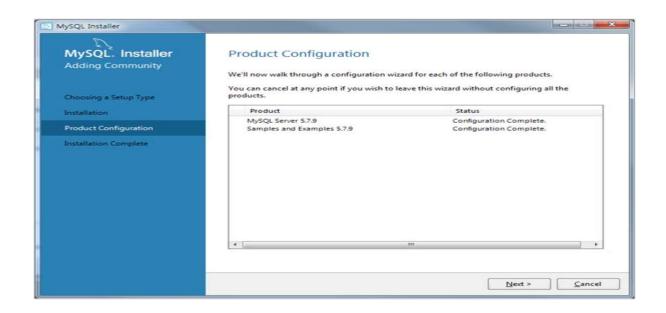


14. 이제 최종단계로 Execute 버튼을 클릭하면 서버를 시작/종료하여 정상적으로 동작하는지 체크합니다.

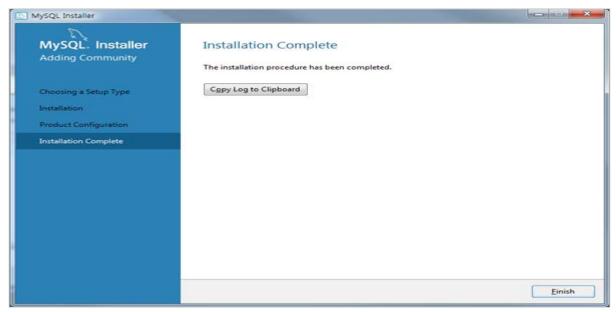




15. 프로그램 기동테스트가 정상적으로 진행되었습니다. Finish 버튼을 클릭하여 테스트를 종료합니다.







16. 이제 모든 설치와 테스트는 종료되었습니다. Next / Finish 버튼을 클릭하여 설치를 종료합니다.

ii. Trinus VR 설치

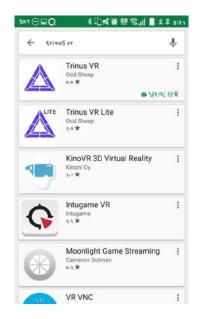
1. Google Play에 접속한다.



페이지 253 / 261



2.Trinus VR을 검색한다.



3. Trinus VR 혹은 Trinus VR Lite를 다운 받아 설치한다. (단 Lite Ver는 15분만 사용가능, 유료 Ver는 약 1만원)



8. 사용자 매뉴얼

- 1. 압축 파일을 해제한다.
- 2. Exe file을 실행한다.
- 3. 처음 이용자는 회원 가입을 한다.
- 4. 로그인을 한다.
- 5. 유저를 검색하여 친구를 친구 추가를 하여 정보를 보거나 자신의 정보를 볼 수 있다.
- 6. 방을 들어가거나 방을 생성하여 다른 유저를 기다린다.
- 7. 타 유저가 방에 접속 할 시 준비가 완료 되는대로 토론을 시작한다.
 - A. Wsad는 방향키 z는 앉기, shift는 달리기, space bar는 점프 혹은 손들기 혹은 버튼을 이용한다.
 - B. 기호에 맞는 Background Music을 들을 수 있다.
- 8. 발언을 하고싶은 사람은 손을 들어 발언권을 요청하고 방장은 해당 유 저의 마이크 버튼을 눌러 발언권을 부여하는 방식으로 토론을 진행한 다.
 - A. 이 때 발언권은 60초 이며 발언권 회수가 가능하다.
- 9. 토론이 종료될 즈음 방장의 선택에 의해 투표를 진행하고 결과를 확인 할 수 있다.
- 10. 투표가 끝나면 일정시간이 지나면 로비로 강제로 이동된다.



9. 프로젝트 마무리

A. 기대효과

- i. 현 교육부에서 고교 학점제를 추진과 동시에 토론식 수업을 확대 하기로 하였다. (2016.12.22 Edaily 발췌) 발표에 앞서 서울시 교육청에서는 서울 중·고교에 학생 참여 형 토론 수업을 도입하기로 하였다.(2016.10.18 News1 발췌) 하지만 기존 교육 방식은 주입식 교육 및 토론 수업 비중이 학기에 1회 밖에 안하고 학생들의 흥미도 또한 '매우 흥미 있다.'의 비중이 초등학생 : 30.30%, 중학생 : 17.40%, 고등학생 : 12% 밖에 안되기 때문에(2016.06.21 머니투데이 발췌) 갑작스레 토론 수업만 늘린다고 해서 학생들의 흥미를 바로 끌 수 있지 않다. 이에 따라 현재 뜨고 있는 VR을 이용하여 토론수업에 접목 시켜 학생들에게 토론수업에 대한 흥미를 이끌어 낼 수 있다.
- ii. 토론수업을 할 때 공간적 시간적 제약이 있다. 학교에서 토론수업을 할 때 토론 형태의 책상 배치를 따로 해야 하고 배치 하는 동안 시간이 흘러간다. 그렇게 되면 토론할 수 있는 시간이 줄어 들고 수업이 끝났을 때 다시 책상을 정리하는데 시간이 소비된다. 하지만 토대리는 각 학교에 컴퓨터 실을 이동하여 실행만 하면 된다는 간편한이점이 있다. 또한 Trinus VR을 이용하기 때문에 고가의 VR 장비 필요 없이 Google Cardboad VR과 같은 저렴한 Smart Phone VR 디바이스를 이용하여 저렴한 비용으로이용할 수 있다. 그리고 각종 토론 대회에서 지리적으로 멀거나 기상이변으로 인해 오프라인 대회장에 늦어지거나 못가는 경우가 있는데 토대리를 이용하면 지리와 기상같은 자연적 조건에 큰 제약을 받지 않기 때문에 범용적으로 사용할 수 있다.

B. 문제점

- ii. 공모전 참가를 하고 싶었으나 연말이라 공모전이 없어 참여하지 못하였다.
- iii. 웹파싱 기능을 추가 하려고 했으나 시간이 촉박하여 추가하지 못하였다.
- iv. 음성 데이터를 Codec으로 압축 시켰어야 했으나 Unity에서 Direct Sound를 지원하지 못하여 압축시키지 못하였다.

C. 개선방안

- ii. 웹파싱을 추가하여 최근 사회적 이슈 및 화제를 실시간으로 업데이트 해줄 수 있다.
- iii. Codec을 적용시켜 음성 데이터의 크기를 줄여 Server에 부하를 줄일 수 있다.



D. 어려웠던 점

가) 문제점

a. Modeling

- i. 스케치업, 3DMAX에서 만든 모델링을 FBX파일로 추출하여 유니 티에 적용 시킬때 텍스쳐 적용방법과 FBX추출시 일부 모델링이 제대로 추출 되지 않는 문제 가 발생.
- ii. 모델링 완성 텍스쳐 완성 후 사실적인 묘사 방법을 모르는 일이 발생.
- iii. 모델링과 라이트 설치 후 상당한 버벅임과 프레임 저하.
- iv. 캐릭터모델링 기술부족.

b. Server

- i. Multithread 환경에서 임계구역 설정 실패로 예상하지 못한 에 러가 종종 발생.
- ii. 캐릭터 이동 시 네트워크 싱크가 맞물리지 않는 문제점 발생.
- iii. 음성 싱크가 맞물리지 않는 문제점 발생 처음에 0.1초 주기로 음성 데이터를 byte배열로 변환하여 서버와 송수신을 하고, 서버 에서는 음성 데이터를 받으면 해당 방안에 클라이언트들에게 음 성데이터를 송신하였으나, 클라이언트가 구동되는 CPU 사양에 따라 음성이 끊기는 현상이 발생.

c. Voice

i. Unity에서 사용하는 .Net Version은 2.0인데 반해 Visual Studio 자체 .Net은 6.0이라 너무 기능이 제한적이다. 그래서 Direct Sound 사용불가.

d. VR

- i. PC와 Smart phone사이 TrinusVR 연동이 매우 불량.
- ii. 연동 후 시점이 멋대로 움직임. Unity내부에서 고정 불가.

나) 해결방법

a. Modeling



- i. 설정에서 모든 모델 삼각형화 , 양면 내보내기 등 설정을 학습하 여 해결.
- ii. 노멀맵, 라이트맵, 유니티 쉐이더를 학습, 이용해 더욱 사실적인 묘사완성.
- iii. 모델링단계로 다시 돌아가 최대한의 삼각형의 개수를 최소화. 라이트 매핑을 이용하여 빛의 연산부분을 미리 베이크 하여 해 결. 부분적으로 움직임이 필요한 부분만 리얼타임 연산을 이용하 여 그림자 구현.
- iv. 3D모델링 사이트의 무료(허가) 배표 자료 이용 리깅과 애니메이 션 제작.

b. Server

- i. 클라이언트가 비정상 종료가 될 시 서버에서 해당 소켓을 접근 하면서 소캣 삭제로 해결
- ii. 애니매이션과 캐릭터 이동을 플래그화 시켜 서버와 동기화.(키입력 동기화 방식) 미묘한 차이로 위치 차이가 발생할 수 있으므로 서버에서 1초마다 각 캐릭터 위치 및 방향, 애니매이션 상태 값을 클라이언트에 뿌려주고, 오차가 자주 발생하는 클라이언트를 수정
- iii. 음성 송수신 주기를 0.5초로 늘려서 각 패킷 송수신 마다 소요 되는 시간을 줄이고, 서버에서는 데이터를 바로 뿌리지 않고 받 은 시간과 함께 데이터를 저장해 놓는 방안을 생각, 클라이언트 에서 다른 클라이언트의 음성을 요청 할 시, 서버에서 음성 데이 터가 저장된 시간을 비교, 분석하여 활성화 된 데이터들만 송신.

c. Voice

i. Codec으로 압축 시키지 못하고 ByteStram으로 음성 데이터를 서버에 전송(Photon Cloud 사용 X)

d. VR

- i. Trinus VR application을 수차례 지웠다 깔았고, Smart phone 재 부팅을 수차례 진행
- ii. Trinus UI 부분에서 강제로 고정

페이지 258 / 261



E. 참고문헌 및 논문

- 가) 뇌를 자극하는 C# 프로그래밍 박상현, 한빛미디어
- 나) 뇌를 자극하는 TCP/IP 운상배, 한빛미디어
- 다) 유니티 개발자를 위한 C#으로 온라인 게임 서버 만들기 이석현, 한빛미디어
- 라) 반다이 남코 현역 프로그래머가 알려주는 유니티 네트워크 프로그래밍 가와다마사 토시, 길벗
- 마) 절대강좌! 유니티5 이재현, 위키북스

F. 참고사이트

- 가) www.google.com
- 나) www.stackoverflow.com
- 다) Café.naver.com/unityube
- 라) Msdn.com
- 마) Unity.com



G. 팀원 별 소감	
팀원	소감
임지훈	C, C++, Java일부분에서만 경험이 있었던 저로써는 이번에 비트캠프를 다니면서 리눅스와 DB와 서버, 3D엔진인 Unity, 아두이노, 라즈베리파이, OpenCV 등등 새로운 것들을 접하면서 자신감도 많이 생겼고, 설사 모르는 툴이나 언어라도 시간을 좀만 가지면 알 수 있다는 자신감이 생겼고, 팀장까지 맡게 되었습니다. 팀장을 하게 되어 조금 부담스러웠지만 너무나 값진 경험이었으며 프로그래밍 기술과 외적, 내적으로 매우 성장하게 되었습니다. 그리고 눈에 보이지 않는 팀장의 어려움을 잘 알게 되었습니다. 다른 분들께도 꼭 이 교육과정과 저희를 가르치신 문상환 강사님을 소개해 주고 싶습니다. 부족한 팀장이지만 잘 믿고 따라준 저희 팀인 강승리형, 손민수에게 감사인사를 드립니다. 비록 강사와 매니저가 많이 바뀌고, 그만둔 학우들도 있지만 이 교육을 받는 동안 함께 고생한 학우분들과 강사님과 매니저님에게 감사 인사드립니다. 다음에 꼭 다시 만나고 싶습니다.
강승리	6 개월 전만해도 printf 도 모르는 상태였던 자신이 프로그래밍 공부를 하면서 이런 프로젝트를 할 수 있었던점에 대해서 자신에게 무척 놀라움을 느꼈습니다. 프로그래밍 공부를 하면서 힘들다, 포기하고 싶다 라는마음이들며 정신적으로 마음이 꺾이는 일이 많았습니다.하지만 하나, 둘 알아 가면서 프로그래밍에 대한 재미를알아가며 성공 했을 때의 성취감이 지금의 저를 지지해준버팀목이였다고 생각합니다. 프로젝트로 유니티를 들어가면서 처음에는 "쉽다" 라는말을 들으며 만만할 줄 알았는데, 역시 프로그래밍에 쉬운것은 없다는 것을 다시한번 체험하고 느끼게되었습니다. 프로젝트를 진행하고 찾아 보고 알아가며 자신감이 조금씩생기는 자신을 발견 하였고, 아직도 저는 함부로



"쉽다"라는 이야기는 하지 못합니다. 하지만 중요한 것은 배움에는 끝이없고 자만심을 가지면 언젠가는 벽을 만난다는 마음가짐을 갖게되었습니다.

자신이 부족해도 하나 둘 배워가며 성장 하는 경험이란 또다시 얻기힘든 귀한 체험을 하였습니다.

또한 함께 고생 해준 팀원 과의 소통과 원활한 대화 회의 등 프로그래밍만큼 사람의 커뮤니케이션이 중요하다는 것, 그리고 할 수 없을 것 같아도 하나 둘 해쳐 나아가면 무엇이든 할 수 있다는 '마음의 힘'을 다시한번 느끼게 되었습니다.

손민수



많이 미숙하고 순수 C#만 파다가 Unity도 접하게 되어 매우 새로웠습니다. 물론 하면서 많이 부들부들하고 욕도 많이하고 화도 많이 냈지만 형들이 많이 도와주고 다독여줘서 매우 큰 힘이 되었습니다. 27살 막내어리광(당시 26살) 받아주느라 많이 고생하셨고 죄송스러웠습니다. 특히 음성채팅 매우 욕나오고 집에서 작업할 때 혼자 쇼도 많이 했지만 그래도 잘 풀려서 매우다행이라 생각합니다.

공모전 나가고 싶었는데 연말이라는 특수 조건 때문에 못나가 매우 아쉬웠지만 나중에 기회가 되면 나갔으면 합니다.

의욕만 앞서고 정작 재대로 하지도 못한 제 자신이 부끄러웠지만 비트에서 정말 많은 것을 배우고 갑니다. 6개월간 하루 10시간 이상 히노애락을 같이한 팀원 및 비트 동기들에게 매우 감사하고 그동안 거처지나간 강사님들과 메니저님 또 문상환 강사님과 문성룡 메니저님께 감사의 인사를 드립니다. 형들은 싫겠지만 언제한번 다시 program을 만들어 봤으면 합니다. 사회나가서 동기들과 강사님, 메니저님 만나서 술 한잔 기울일 때 안주 하나 더 생겨서 더더욱 좋습니다. 다음에 만날땐 다들 성공 한 모습으로 봤으면 합니다.