**작업 진행 보고서**  
네트워크 게임 프로그래밍

7팀 : 김상혁, 진현서, 조성원

**조성원**

**11월 7일 목요일**

<Packet>

플레이어 위치 , 색상 , 크기 정보 구조체 패킷화 작업

struct ClientInfoPacket

**11월 8일 금요일**

<Client>

뷰어 클라이언트 객체 변수 및 함수 정의 및 작성

ViewerPlayer.h , ViewerPlayer.cpp 파일 코드 작성

**11월 9일 토요일**

<Client>

ViewerPlayer 가 모든 클라이언트의 화면에 적용이 되는지 테스트

**11월 14일 목요일**

<Client>

뷰어 클라이언트를 생성 , 검색 , 삭제 하는 함수 작성

void CreateClientPlayer(int ClientID);

ViewerPlayer\* FindClientPlayer(int ClientID);

void RemoveClientPlayer(int ClientID);

**11월 15일 금요일**

<Client>

내 클라이언트의 접속을 끊는 함수 작성

void DisConnectClient();

접속을 끊은 클라이언트가 접속을 끊었다는 패킷을 전송하면 내 클라이언트가 반영하는 함수작성

void DisConnectClientInfo(int ClientID, bool isMaster);

**11월 17일 일요일**

<Client>

내 플레이어의 위치, 색깔, 크기 정보가 변경되었을 때 다른 클라이언트들에게 알려주는 함수 작성

void SendPlayerInfo();

<Client>

다른 클라이언트의 위치, 색깔, 크기 정보가 변경되었을 때 뷰어플레이어 객체를 업데이트 해주는 함수 작성

void ClientInfoData(int ClientID , const ClientInfoPacket& cInfo);

**11월 19일 화요일**

<Client>

뷰어 플레이어의 X 좌표의 위치를 업데이트하는 함수 작성

void updateViewerPosX(ViewerPlayer\* pViewer , float PosX);

**11월 21일 목요일**

<Client>

다른 클라이언트의 위치가 바뀌면 그 클라이언트 ID 를 가지는 뷰어 플레이어의 색과 크기를 변경하는 함수를 작성함

void updateViewerColor(ViewerPlayer\* pViewer , float r , float g , float b);

void updateViewerScale(ViewerPlayer\* pViewer , float x , float y);

인게임 시작 전 ViewerPlayer 객체들을 생성하는 함수 작성

void InitViewerPlayer(int MyClientID);

<Server>

어떤 플레이어의 정보가 변경되었을 때 서버에서 다른 모든 클라이언트에게 그 정보를 쏴주는 함수 작성

void PlayerInfo(SOCKET socket , int clientID , const ClientInfoPacket& Info);

**11월 22일 금요일**

<Client>

뷰어 플레이어 캐릭터의 모델을 마법사 모델로 변경

스테이지 인덱스 오류 수정

인게임에 hp 스크린 추가

**11월 23일 토요일**

<Test>

플레이어 위치 동기화 테스트 및 남은 진행 상황 공유

**11월 28일 목요일**

<Test>

클라이언트 간 벽 속도 동기화 테스트를 통해 버그를 발견하고 서로 공유

**12월 1일 일요일**

<Test>

마스터 클라이언트 위임 문제 버그 발견 및 수정

**12월 2일 월요일**

<Test>

클라이언트 종료 시에 나머지 클라이언트 간 위치 , 벽 동기화가 잘 이루어지는 지 테스트

클라이언트 종료 시에 버그를 발견하고 공유

**12월 3일 화요일**

<Test>

팀원들과 모여서 여러가지 상황을 테스트

**12월 4일 수요일**

Progress Report 작성

**김상혁**

Clinet ,Server 프레임 워크 완성 작업 완료

Client - PacketDecode() - 프로토콜에 따라 패킷해석및 로직실행

SendConnect() - 접속시 서버에게 아이디 부여패킷 요청

RecvConnect() - 서버에게서 아이디 부여 받음

SendStartGame() -서버에 레디 패킷 전송

RecvStartGame() - 서버로 부터 3명이 접속하여 레디한 상태로 게임 시작 패킷 수신 및 로직실행

SendChattingData() - 서버에게 메세지 패킷 전송

RecvChattingData() - 서버로부터 다른클라이언트에게서 받은 메세지 수신

Server -PacketDecode() - 프로토콜에 따라 패킷해석및 로직실행

SendConnect() - 클라이언트에게 ID 부여작업 패킷전송

RecvConnect() - 클라이언트로 부터 ID 부여작업 요청받음.

RecvSendStartGame() - 클라이언트로부터 레디 상태 입력받아 3명 만족시 씬전환 패킷전송

RecvMathcingCancle() - 클라이언트로부터 Mathcing 취소 요청 처리

RecvSendChattingData() - 클라이언트로부터 채팅 메시지 입력받아 브로드캐스팅

11/17 ~11/30

벽동기화작업 / 클라이언트 나갔을때 에 대한 처리

게임 끝날시 다시 게임 재시작할수있게 수정

12/03

리소스 계속해서 로드하는 버그수정

**진현서**

**11월 14일~11월 16일**

**함수 추가**

void BlockCreate();

void BlockCreateReceive();

void HpUpdate(int hp)

커맨드 구분용 CMDCODE 3개 추가.

BlockDataRecv,

BlockCollision,

HPSync,

패킷 송수신 구조체 2개 추가.

struct BlockCreateInfo

struct HPInfo

**11월 17일~11월 29일**

클라이언트 작업

Client::BlockCreate() 마스터클라이언트의 블록정보 전송 함수 작업.

BlockCreateReceive() 게스트 클라이언트 블록정보 수신 및 반영 함수 작업,

Client::HpUpdate() 수정 및 HP 정보 수신 함수 작업.

Client::BlockCollision() 충돌시 충돌 이벤트 전송 함수 작업.

서버 작업

case ENUM::BlockDataRecv:

MakeBlockSend(socket);

break;

case ENUM::BlockCollision:

BlockCollision(socket);

블록 데이터 처리 작업 추가

충돌 이벤트 처리 작업 추가

**11월 20일**

충돌 로직에서 클라이언트가 Wall 의 EmptyIdx를 초기화 하는 시점에 대한 버그가 발생하여 해당 버그를 픽스

**11월 24일**

Server 측 코드에서 HP가 증가해야하는데, 감소하는 로직으로 구성되어 있어 해당 로직 수정

**11월 25일**

마스터 클라이언트가 게임을 플레이하던 중 종료할 시 남아있는 클라이언트중 새로운 마스터 클라이언트를 선정하여 마스터 클라이언트의 권한을 위임하고, 로직을 지속적으로 정상 작동하도록 수정하는 작업.

**11월 28일**

마스터 클라이언트를 선정하는 과정에서 발생한 버그 로직을 수정.

**11월 29일**

클라이언트가 접속을 종료하였을 때 Viewer객체가 Delete 되는 로직을, 객체 재사용성을 위하여 Rendering Disable 방식으로 전환하는 작업

Server에서 \_readyCount를 0으로 초기화 하는 시점을 게임이 매칭되어 시작하는 시점으로 로직 이동,

**12월 01**일

Client가 접속을 종료하였을 때 Socket을 보관하는 ConcurrentQueue에서 socket 정보를 지우는 과정에서 발생한 로직 오류를 수정함.

**12월 03일**

게임이 종료되고 재시작될 때 발생되는 Init 정보들을 Reset 함수로 분리하여 Reset을 통해 필요한 정보들만 초기화 하는 작업 완료.

ViewerPlayer에서 렌더링 정보가 초기화 될 수 있도록 작업 수정 완료

재시작 시 HP바 동기화가 되지 않는 버그 수정 완료.