**Project Progress Report**

2017180024 유재우

2017180027 이 도

2017180028 이도영

목차

1. 문서 개요………………………………………………………………………………………………………………………………………….3

1.1. 기획서 개요…………………………………………………………………………………………………………………..…3

1.2. 개발 환경…………………………………………………………………………………………………………………………4

2. 게임 개요…………………………………………………………………………………………………………………………………….……5

2.1. Perfect Shape 개요………………………………………………………………………………………………..…………5

3. Game Play….………………………………………………………………………………………………………………………………………6

3.1. 게임 플레이……….………………………………………………………………………………………………………...….6

4. Level Design………………………………………………………………………………………………………………………………………7

4.1. High - Level Design ………………………………………………………………………………………………….……..7

4.2 Logine Flow Chart……………………………………………………………………………………………………………..8

4.3 Game Flow Chart………………………………………………………………………………………………………………9

4.4 Code Flow Description…………………………………………………………………………………………………….10

4.5 Low – Level Design………………………………………………………………………………………………………….11

4.5.1 Protocol………………………………………………………………………………………………………....11

4.5.2 함수………………………………………………………………………………………………………...........16

5. 개인별 역할 분담 및 개발일정……………………………………………………………………………………………………..20

5.1. 개인별 역할 분담………………………………………………………………………………………………………….20

5.2. 개발 일정………………………………………………………………………………………………………………………21

6. History……………………………………………………………………………………………………………………………………………..18

1. 문서 개요

1.1 기획서 개요

**Project Progress Report**

**네트워크 게임 프로그래밍 최종 산출물 report**

1.2 개발 환경

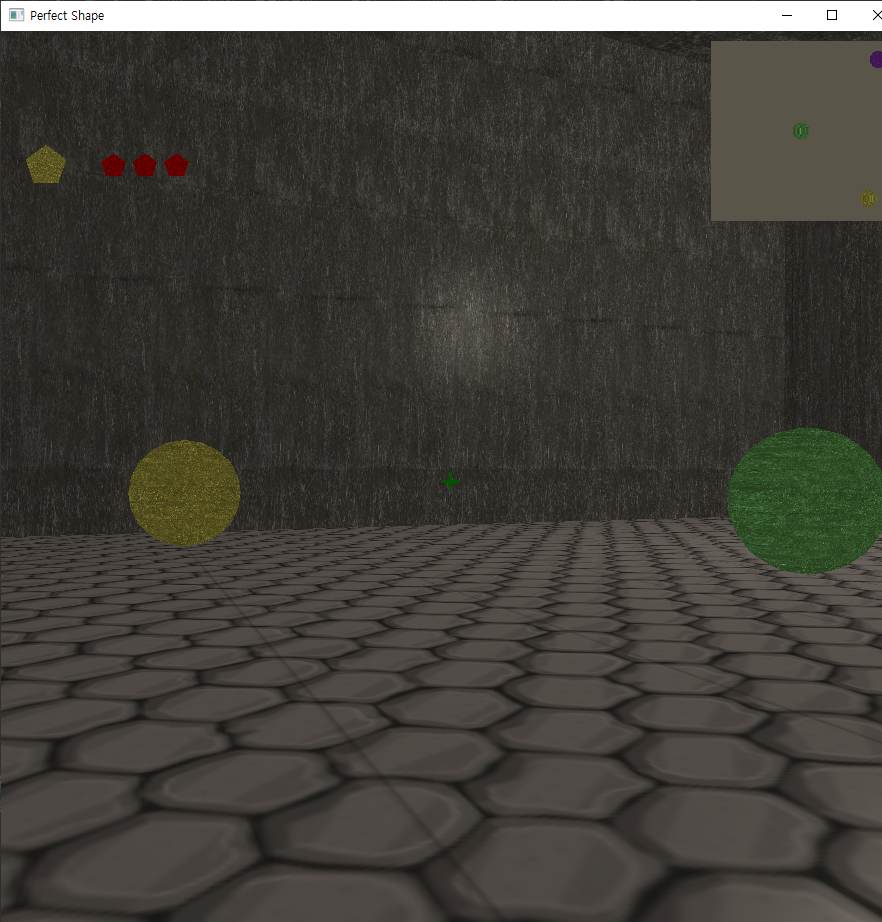
* OS : Windows 10
* IDE : Visual Studio 2019
* API : Win32 API / Windows Socket API
* 네트워크 IO모델 : 다중 쓰레드 모델
* 언어 : C / C++ (Open GL)

2. 게임 개요

2.1 Perfect Shape 개요

- 게임제목: Perfect Shape

- 게임컨셉: 플레이어3명에서 캐릭터를 조종하여 적을 쓰러트리며 다음 방으로 진행하며 지속적으로 살아남는 1인칭 슈팅게임



<게임화면>

3. Game Play

3.1 게임플레이

- WSAD(상하좌우) 키보드를 통해 입력을 받아 플레이어를 움직일 수 있음.

- 마우스 왼쪽 커서를 통해 적을 공격

- 적에게 충돌하거나 적의 총알에 충돌할 경우 데미지를 입음.

- 게임의 흐름

1. 스테이지 시작

2. 모든 적 처치

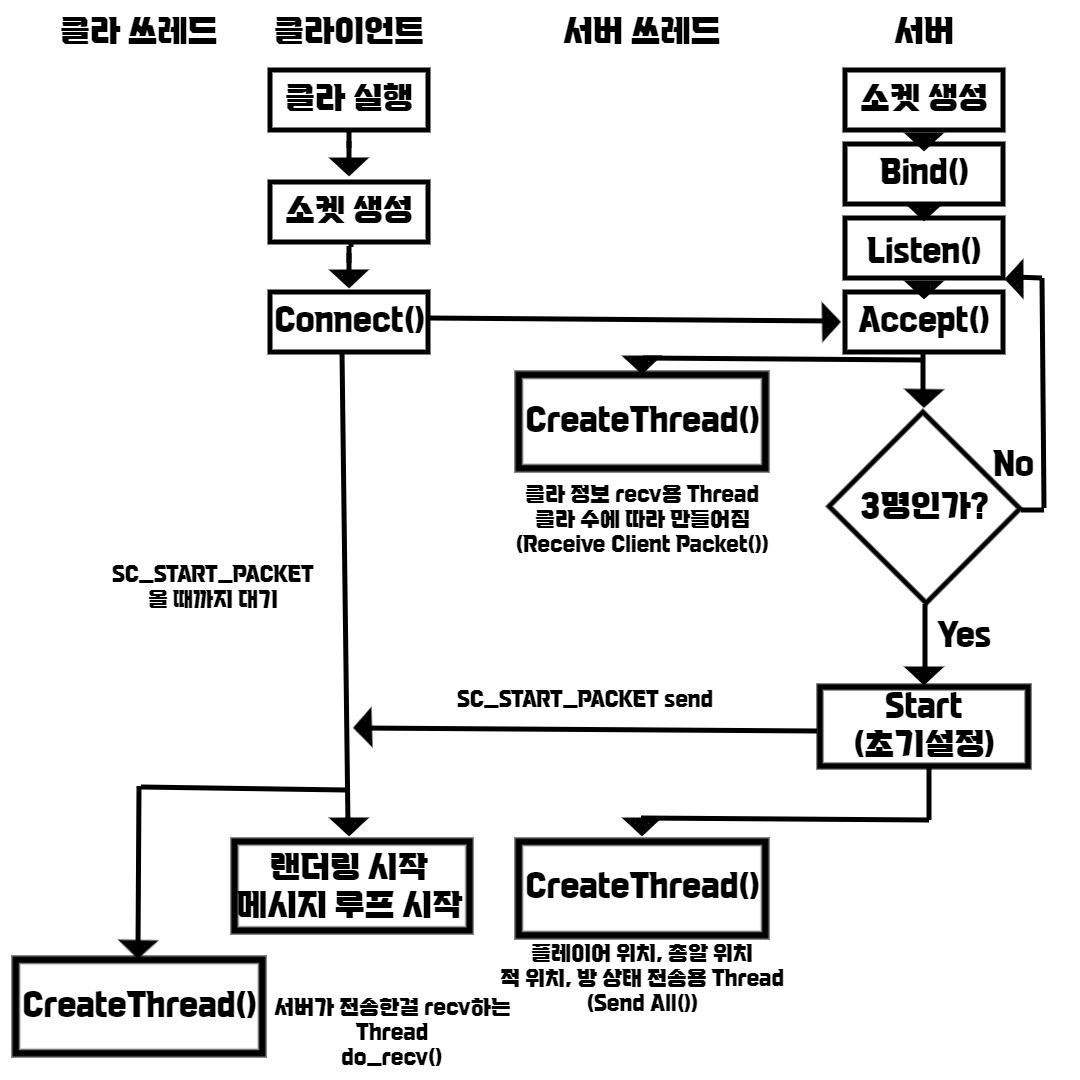
3. 다음 방 이동 (다음 스테이지 시작)

4. Level Design

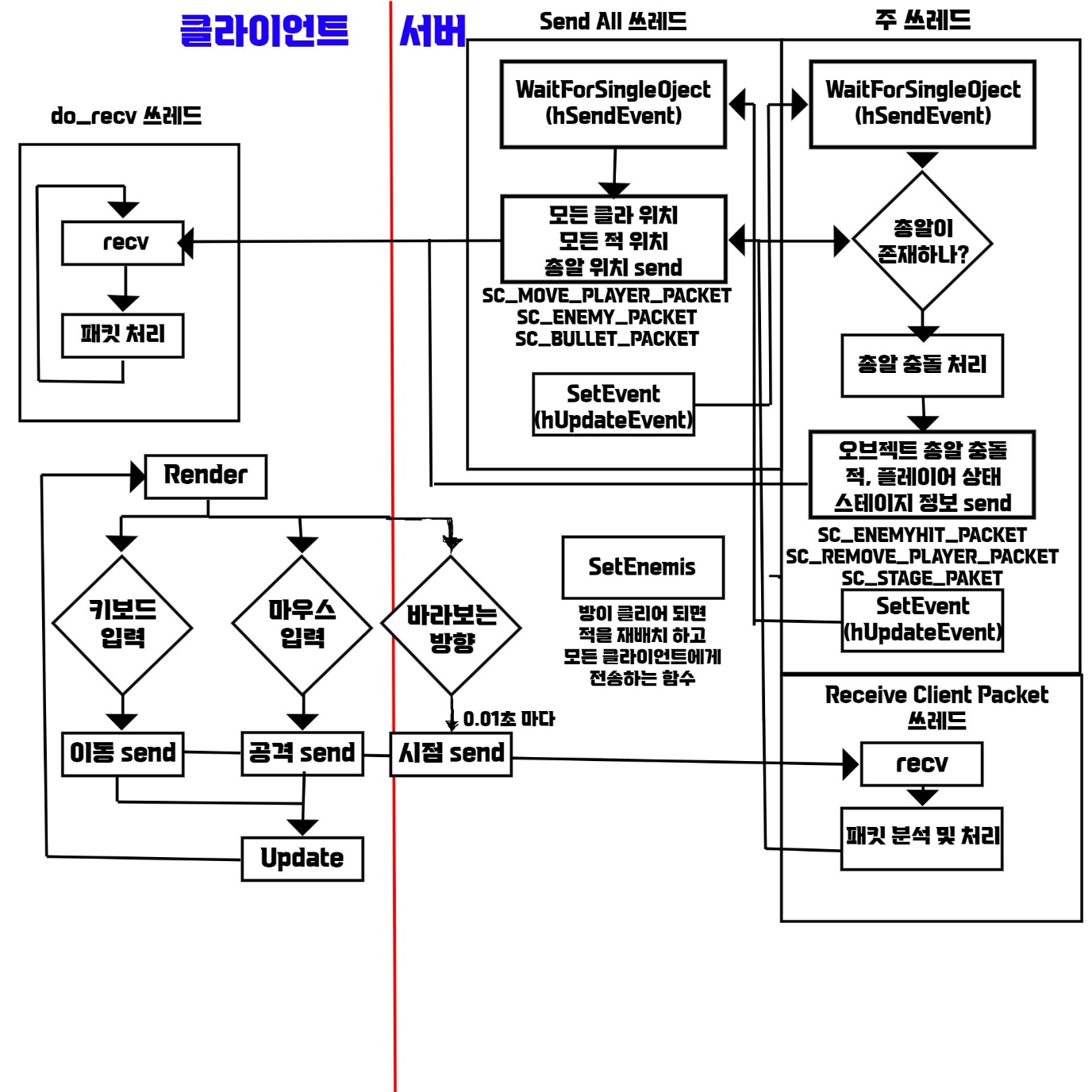
4.1 High – Level Design



4.2 Login Flow Chart



4.3 Game Flow Chart

****

4.4 Code Flow Description

* + 접속
    - ~~클라이언트 실행 시 로그인 GUI 생성 서버로 접속 요청을 보냄~~
    - 서버는 접속요청을 받고 accept가 되면 해당 클라이언트를 위한 쓰레드를 만듬.
    - 그 후 해당 클라이언트에게 id를 부여하고 id와 함께 접속이 되었다는 의미의 패킷을 보내줌.
    - 게임 시작할 때에 3인이 접속할 때까지 대기한 후 모든 플레이어가 접속시 서버에서 다른 플레이어의 ID정보를 담은 패킷을 보내준다.
    - ~~이미 접속한 다른 클라이언트가 있다면 새로 접속한 클라이언트에게 이미 접속해 있는 클라이언트의 ID정보를 담은 패킷을 보내고, 이미 접속해 있는 클라이언트에게는 새로 접속한 클라이언트의 ID정보를 담은 패킷을 보내줌.~~
    - 서버는 3명의 클라이언트가 접속하면 게임시작을 위한 정보 초기화 (플레이어의 위치)를 해주고 게임이 시작되었다는 패킷을 클라이언트에게 보내줌.
    - 게임을 시작하면 send를 해주는 쓰레드를 만들고 메인 쓰레드는 플레이어와 적의위치, 공격, 충돌 처리를 해줌.
  + 게임 시작
    - 게임 시작 시 서버에서 게임이 시작되었다는 패킷을 전송하고 클라이언트에서 수신 받으면 게임 GUI 렌더링을 해줌
    - 움직임이나 공격을 위한 입력이 있다면 서버에 해당 키를 send()해주고, 서버는 알맞은 처리를 해줌
    - 서버에서 대부분 계산처리 작업을 진행하기에 충돌처리를 서버에서 진행할 예정 성능적인 부분은 3인 게임에서 큰 차이는 없을 거라 생각하였음.
    - 동기화방식은 이벤트 방식을 사용할 예정 크리티컬 섹션에 비해서 성능적인 이점을 취할 수 있기에 채택함.
    - 스테이지가 클리어 되면 문이 다 열리고 다음 방으로 진입할 시 모든 클라이언트에게 적군의 정보를 넘겨줌.

4.5 Low – Level Design

4.5.1 Protocol

- 클라이언트에서 플레이어 로그인 시 보내는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

- 클라이언트에서 플레이어 이동시 보내는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

-클라이언트에서 키 입력 시 보내는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

-클라이언트에서 플레이어 마우스 클릭 시 보내는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

-클라이언트에서 플레이어의 방향을 보내는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

- 처음 접속했을 때 접속한 클라이언트 아이디 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

- 접속한 클라이언트의 정보 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

- 게임시작시 알림 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

- 플레이어 없애는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

- 플레이어 이동시 위치 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

-플레이어 적에게 피격 시 보내는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

- 적 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

-적 몬스터 젠 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

-적 피격 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

- 총알 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

- 총알 피격 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

- 스테이지 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

-플레이어 피격 시 조명 변화 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

-플레이어 사망 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

상수

- 클라이언트와 서버 송수신간 보내는 패킷 타입

텍스트이(가) 표시된 사진

자동 생성된 설명

송수신 시 보내는 패킷 타입을 정의하여 서버에서 어떤 패킷을 받았는지 알 수 있게 한다.

기타

텍스트이(가) 표시된 사진

자동 생성된 설명

- 포트번호, 사이즈, 정의

텍스트이(가) 표시된 사진

자동 생성된 설명4.5.2 함수

**클라이언트**

* int NetInit()
  + **서버 접속함수**
* void NetCleanup()
  + **서버 Disconnect시 closesocket(); WSACleanup(); 호출 함수**
* ~~int send\_move\_packet(float x, float z)~~
* int send\_keyboard\_packet(int direction)
  + **키 입력 송신 함수**
* int send\_direction\_packet(float dx, float dz,float degree)
  + **플레이어의 방향 송신 함수**
* int send\_attack\_packet(float dx, float dy, float dz)
  + **플레이어 attack 송신 함수**
* DWORD WINAPI do\_recv()
  + 서버가 전송한 패킷을 받고 패킷을 처리하는 쓰레드 함수

**서버**

* DWORD WINAPI Receive\_Client\_Packet(LPVOID player)
  + **클라이언트 전송 패킷 수신 함수**
* DWORD WINAPI SendAll(LPVOID msg)
  + **서버데이터 송신 함수**
* void send\_login\_packet(SOCKET\* c\_socket, short c\_id)
  + **클라이언트가 접속하면 접속확인과 id를 보내는 함수**
* void send\_add\_packet(SOCKET\* c\_socket, short c\_id)
  + **다른 클라이언트의 접속 전송 함수**
* void send\_move\_packet(SOCKET\* c\_socket, short c\_id)
  + **클라이언트의 움직임 전송 함수**
* void send\_bullet\_packet(SOCKET\* c\_socket, short b\_id)
  + **총알의 위치 값 전송 함수**
* void send\_bulletHit\_packet(SOCKET\* c\_socket, short b\_id)
  + **총알 충돌 시 패킷 전송 함수**
* void send\_remove\_packet(SOCKET\* c\_socket, short c\_id)
  + **죽은 플레이어 삭제 함수**
* void send\_enemy\_packet(SOCKET\* c\_socket, int e\_id)
  + **enemy 정보 전송 패킷 전송 함수**
* void send\_enemyHit\_packet(SOCKET\* c\_socket, short id)
  + **enemy 충돌 시 패킷 전송 함수**
* void send\_hit\_packet(SOCKET\* c\_socket, short c\_id)
  + **플레이어 피격 시 패킷 전송 함수**
* void send\_hitend\_packet(SOCKET\* c\_socket, short c\_id)
  + **플레이어 피격시 조명 변화를 위한 패킷 전송함수**
* void send\_dead\_packet(SOCKET\* c\_socket, short c\_id)
  + **플레이어 사망시 player state 전송 함수**
* void send\_GenRandEnemy\_packet(SOCKET\* c\_socket, int e\_id)
  + **enemy 젠 함수**
* void send\_start\_packet(SOCKET\* c\_socket)
  + **게임 start시 패킷 전송 함수**
* void gameStart()
  + **게임시작시 game\_state 변경 함수**
* void GenRandEnemy(int clear\_num)
  + **enemy 위치 정보 get 함수**
* void CalculateEnemyDirection(int id)
  + **enemy player 쫓아가는 함수**
* void InitEnemyBullet()
  + **enemy 총알 get 함수**
* void Disconnect(SOCKET\* c\_socket, short c\_id)
  + **Disconnect 시 패킷 전송 함수**
* bool IsCollision\_PE(Enemy en, Player pl)
  + **플레이어와 적이 충돌했는지 검사하는 함수**
* void Player\_KnockBack(short id)
  + **플레이어 충돌 시 밀리는 함수**
* void Player\_Check\_Unbeatable(short id, chrono::system\_clock::time\_point start\_time)
  + **충돌 이후 플레이어가 밀리면 무적시간을 부여하는 함수.**
* bool Player\_Check\_Touch\_Wall(short id)
  + **벽 충돌 함수**

5. 개인별 역할 분담 및 개발 일정

5.1 개인별 역할 분담(위치수정)

1. 이도영

- 일정 조율 및 문서 작성

- void send\_login\_packet() : 클라이언트가 접속하면 접속확인과 id를 보내는 함수

- void send\_add\_packet() : 다른 클라이언트의 접속 전송 함수

- void send\_remove\_packet() : 죽은 플레이어 삭제 함수

- void Disconnect() : 플레이어 연결 종료 함수

- Player\_Dead\_state() : 플레이어가 사망시 렌더링을 종료 하는 함수

2. 유재우

- 게임로직 수정

- 각종 렌더링 추가

- void send\_move\_packet() : (클라이언트) 키 입력(W, A, S, D) 송신 함수

- void send\_attack\_packet() : 마우스 좌(attack) 클릭 송신 함수

- void send\_move\_packet() : (서버)클라이언트의 움직임 전송 함수

- void send\_bullet\_packet() : 총알의 위치 값 전송 함수

- int NetInit(), NetCleanup() : 서버 접속 함수

- do\_recv() : (클라)서버가 전송한 패킷 수신 함수

3. 이도

- bool player\_collide() : 플레이어 충돌 판단 함수

- bool enemy\_collide() : 적 충돌 판단 함수

- DWORD WINAPI Send\_all(): (서버)데이터 송신 함수.

- void process\_packet(); 패킷 재 조립 함수

- void GameStart() : 게임이 시작 및 정보 초기화 함수

- 벽 충돌 체크 및 충돌 체크에 관련된 함수 및 버그 오류 수정

( 적 충돌시 조명 변화, 충돌시 무적시간 추가, 벽 적 충돌체크간 오류 수정)

5.2 개발 일정(위치 수정)

|  |  |  |  |
| --- | --- | --- | --- |
|  | **유재우** | **이도** | **이도영** |
| 10/22(토) | 전체 기획 회의  코드분석, 클라이언트 추가 사항, Git 생성 및 clone, 게임 플로우 작성 | | |
| 10/26(수) | 코드 리뷰 원래 게임 변경 Break Down -> Perfect Shape | | |
| 10/29(토) | 코드 리뷰(서버와 클라에 들어갈 코드 나눔) protocol 생각 | | |
| 10/30(일) | High Level Design | | |
| 11/1(화) | Low Level, Game Flow Chart, Code Flow Description, 문서 작성 | | |
| 11/2(수) | High Level Design 추가일정 작성 및 문서 최종 점검 | | |
| 11/3(목) | 문서 제출 | | |
| 11/4(금) |  |  | ~~Send\_login\_pakcet()~~ |
| 11/5(토) | NetInit(), NetCleanup() 작성 |  |  |
| 11/6(일) |  | ~~Send\_all()(서버) 데이터 수신 송신 함수~~ | Send\_login\_pakcet()  서버 main 틀 작성 |
| 11/7(월) |  |  |  |
| 11/8(화) | do\_recv() 작성 | Send\_all() 틀 작성 | Send\_add\_packet() 작성 |
| 11/9(수) | 클라 서버 연결 확인 | | |
| 11/10(목) |  |  |  |
| 11/11(금) | 일정 점검 부족한 부분 수정 | | |
| Do\_recv() 수정, 클라이언트 오류 수정 | | |
| 11/12(토) | ~~void send\_move\_packet() : (클라이언트) 키 입력(W, A, S, D) 송신 함수~~  ~~void send\_move\_packet() : (서버)클라이언트의 움직임 전송 함수~~ |  |  |
| 11/13(일) |  | ~~void process\_packet(); 패킷 재 조립 함수~~ |  |
| 11/14(월) |  |  |  |
| 11/15(화) | ~~게임로직수정~~ |  | ~~void send\_remove\_packet()죽은 플레이어 삭제 함수~~ |
| 11/16(수) |  | ~~void GameStart() : 게임이 시작 및 정보 초기화 함수~~ | void send\_remove\_packet() 작성 |
| 11/17(목) | void send\_move\_packet() : (클라이언트) 키 입력(W, A, S, D) 송신 함수 |  | ~~void send\_start\_packet() : 게임이 시작했다는 정보를 보내는 함수~~ |
| 11/18(금) | 일정 점검 부족한 부분 수정 | | |
| 11/19(토) | 게임로직수정,  void send\_move\_packet() : (서버)클라이언트의 움직임 전송 함수 | ~~bool player\_collide() : 플레이어 충돌 판단 함수~~  void process\_packet(); 패킷 재 조립 함수 |  |
| 11/20(일) | void send\_attack\_packet() : 마우스 좌(attack) 클릭 송신 함수  send\_look\_packet() 추가 | 클라이언트 recv 스레드 툴 |  |
| 11/21(월) |  | Send\_move 작성  Gamestart 패킷 완성 | void Disconnect() : 플레이어 연결 종료 함수 |
| 11/22(화) | 게임로직수정  클라 – 서버 통신중 렌더링 환경 구축 |  |  |
| 11/23(수) |  | ~~bool player\_collide() : 플레이어 충돌 판단 함수~~ |  |
| 11/24(목) |  |  |  |
| 11/25(금) | 일정 점검 부족한 부분 수정  다중 연결 이후 순간이동 및 위치 오류 | | |
| 11/26(토) | ~~void send\_bullet\_packet() : 총알의 위치 값 전송 함수~~  movet test 디버깅  너무 많은 데이터를 주고 받아 데이터가 손실됨 | ~~bool enemy\_collide() : 적 충돌 판단 함수~~ |  |
| 11/27(일) |  |  | 검증 및 history 추가 |
| 11/28(월) |  |  |  |
| 11/29(화) | void send\_bullet\_packet() : 총알의 위치 값 전송 함수 |  |  |
| 11/30(수) |  | ~~bool enemy\_collide() : 적 충돌 판단 함수~~  bool player\_collide() : 플레이어 충돌 판단 함수 |  |
| 리모트 클라이언트 오류 검증 및 해결 | | |
| 12/1(목) | 총알 렌더링 환경 구축 |  | 4차 검증 및 history 추가 |
| 12/2(금) | 일정 점검 부족한 부분 수정 | | |
| 12/3(토) |  | bool enemy\_collide() : 적 충돌 판단 함수 |  |
| 12/4(일) | 검증 및 수정 | | |
| 플레이어 체력 렌더링 추가  적 피격 파티클 추가 | 플레이어 적군 충돌 체크 |  |
| 12/5(월) | 플레이어 움직임 로직 변경 |  |  |
| 12/6(화) | 검증 및 수정 | | |
|  | 피격시 조명 변경 프로토콜 정리  플레이어 적 충돌체크 |  |
| 12/7(수) | 충돌체크 보완 및 수정  적중 충알 제거  적 총알 발사 | 벽 충돌 체크 추가  무적 추가  Flow chart 수정 | Protocol.h 추가 및 수정  Player Dead state 작성  및 검증 |
| 12/8(목) | 최종 검증 | | |

6. HISTORY

2022 10/22 14:30

텍스트, 모니터, 전화, 휴대폰이(가) 표시된 사진

자동 생성된 설명

<Discord를 통한 비 대면 회의>

* 코드분석
* 클라이언트 추가 사항
* Git 생성 및 clone
* 게임 플로우 작성

2022 10/26

넷겜플 수업시간 코드 분석

* 게임 변경 Break Down -> Perfect Shape

1029회의 Discord를 통한 온라인 회의



1. 주석

2. 서버 클라 코드 나누기

3. protocol 생각하기

맵 -> 전부?

* + Server :: Collide
  + Server :: State -> 열리는지 닫히는지
  + Server :: Open\_door
  + Server :: pos
* Enemy -> ADD -> id 값만 받아오면 된다. -> 각 클라 뿌려준다.
  + Enemy protocol 만들어야 함.
    - 이동 -> 플레이어 따라간다.
    - Type -> attack
    - State -> 체력
    - EnemyBulletPos
    - EnemyBulletNum
* Player
  + Protocol
    - Key\_input
    - PlayerBulletPos
    - PlayerBulletNum -> 배열로 함. 20발

행렬 -> protocol

10/30 일 discord를 이용한 온라인 회의

텍스트, 모니터, 전자기기, 화면이(가) 표시된 사진

자동 생성된 설명

회의 내용

1. High Level Design



11/01 월 discord를 이용한 온라인 회의

텍스트, 주차장, 검은색, 스크린샷이(가) 표시된 사진

자동 생성된 설명

1. Low level

2. Level Design에 따른 코드 흐름도 작성

역할 분담

(서버) 플레이 관련 전송

(서버) 쓰레드 생성 –

(서버) 스레드간 동기화(Event)

(서버) 플레이어 총알 충돌처리

(서버) 클라이언트 연결 –

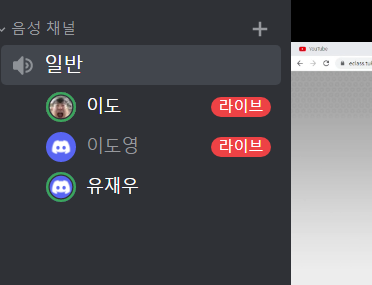
(서버) 게임시작과 접속 관련 전송

(클라이언트) 메인 함수 세팅

(클라이언트) 총알 구현

(클라이언트) 네트워크 연결 및 전송 함수

11/02 화 discord를 이용한 온라인 회의



1. Game Flow Chart

2. Login Flow Chart

3. 개발일정

2022/11/11

1차검증

검증내용

* 클라이언트와 서버가 연결이 되었는지 확인
* 오류 수정

클라이언트와 서버 두 곳 모두 실행이 안되는 오류가 발생

발생원인

서버: 코드를 분할 할 때에 include가 겹치게 되어 오류 발생

#pragma once는 헤더의 중복을 막아주는 것 함수의 예외 였음을 발견

error\_display(), error\_quit() 함수 수정 후 컴파일 성공

클라: switch case 부분에서 오류발생 아직 생성하지 않은 함수와 사용하지 않은 변수가 겹치게 되어 오류 발생 주석 처리 후 컴파일 성공

2차 검증

검증내용

클라이언트의 연결이 확인된 후 데이터가 보내지는지 검증

데이터 보낸 후 disconnect 까지 검증

오류 발생 여부 x

3차 검증 11/25

검증 내용

프로토타입을 만들어 타 컴퓨터와의 통신을 통한 플레이어 움직임 체크

오류 내용:

서버에서 데이터를 받는 것은 보이나 다른 컴퓨터로 다시 되돌려 보내는 것에 오류가 발생

11/26 10:30 재 회의

서버가 클라이언트의 정보를 받는 것은 되나 다시 클라이언트로 재 전송할 때에 오류 발생

생각한 방안에 오류가 발견되지 않아

너무 많은 데이터를 보내어 클라이언트 간에 데이터를 훔치는 경우가 있다 생각하여 sleep을 추가

Sleep 추가이후 두 오류 모두 해결

4차 검증 11/30

오류내용

리모트 서버로 실행 시 플레이어 이동하는 경우 순간이동 하는 오류가 발생함.

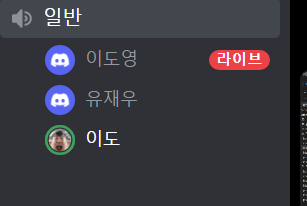
오류 수정

이는 client가recv받는 단계에서 오류가 있었으며

wsabuf 변수내의 예전 값을 제대로 지워주지 않아서 오류가 발생하였음

오류수정 memset(); 함수를 이용한 wsabuf를 초기화하여 오류 수정 완료

12/07 게임 검증



오류 발생 X

보완 및 추가

충돌체크, 총알 제거, 벽에 대한 충돌 체크, 피격 후 무적 시간, 필요 없는 파일 삭제, 프로토콜 수정