**네트워크 게임 프로그래밍  
프로젝트 추진 계획서**

2017180041 최원재  
2017180040 정한솔  
2017182023 원윤식

**목차**

1. 애플리케이션 기획

2. 하이레벨 디자인

3. 로우레벨 디자인

4. 팀원 별 역할 분담

5. 개발 환경

6. 개발 일정

**애플리케이션 기획**

산, 실외, 자연이(가) 표시된 사진

자동 생성된 설명

**| 게임 설명**본 게임은 3명이서 플레이하는 실시간 슈팅 게임이다. 3명의 플레이어가 접속하기 전까지는 대기하다가 플레이어가 모두 접속하면 게임이 시작된다. 게임이 시작되면 각 플레이어는 서로 다른 지정된 위치에서 스폰되며 3개의 목숨을 갖는다. 플레이어는 다른 플레이어의 총알을 맞으면 목숨 1개를 잃고, 남은 목숨이 있는 경우 처음 스폰된 위치에서 리스폰된다. 게임은 최종적으로 1명이 남을 때까지 진행되며 해당 플레이어는 승리하게 된다.

**| 조작법**플레이어는 ↑↓←→키를 이용해 전후좌우로 캐릭터를 움직일 수 있고 X키로 총알을 발사할 수 있다. 총알은 1초에 한 번씩 발사 가능하며 발사한 총알은 1초 동안 날아간다.

**하이레벨 디자인**

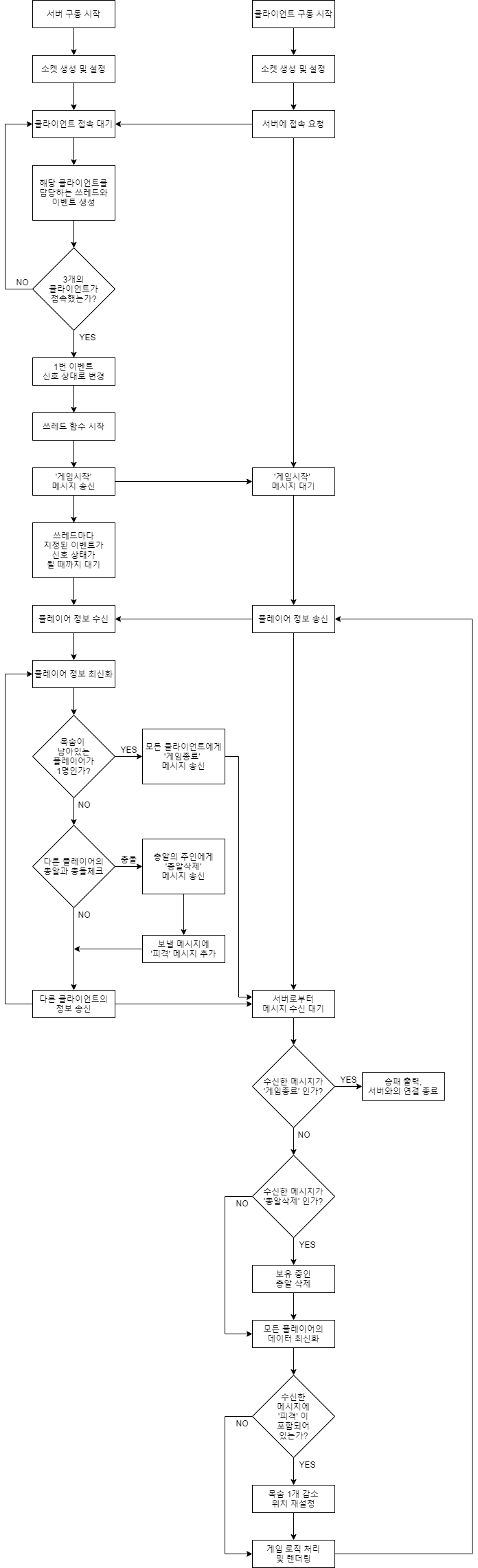
**| 목표 및 진행 방향**기존 Direct3D 기반으로 이루어진 게임 프레임워크에 TCP를 사용하여 클라이언트 환경을 구축하고, TCP 서버를 제작한다. 클라이언트의 접속과 데이터 송수신 테스트를 위한 클라이언트 – 서버 프로토타입을 제작한 뒤, 게임 프레임워크에 프로토타입을 적용하는 방식으로 진행한다.

**| 프로토타입 작동 방식**서버, 클라이언트 프로토타입이 작동하는 방식은 다음과 같다.  
- 서버에서 3개의 클라이언트가 연결되길 기다린다.  
- 서버는 클라이언트가 연결 될 때마다 해당 클라이언트와 연결된 소켓을 저장한다.  
- 서버는 연결된 클라이언트에게 클라이언트별로 미리 지정된 위치의 좌표를 송신한다.  
- 클라이언트는 좌표를 수신해 플레이어의 위치를 해당 좌표로 옮긴다.  
- 클라이언트는 서버로부터 ‘게임시작’ 메시지를 기다린다.  
- 서버와 3개의 클라이언트가 모두 연결되었다면 3개의 쓰레드를 생성해 쓰레드 함수를 실행한다.  
- 쓰레드 함수는 파라미터로 ‘n번째로 접속한 플레이어의 소켓’ 과 ‘n’ 을 받는다.  
- 쓰레드 함수는 처음에 클라이언트에게 ‘게임시작’ 메시지를 전송한다.  
- 서버는 ‘게임종료’ 메시지를 받을 때까지 클라이언트로부터 ‘플레이어 정보 구조체’를 수신 받고 수신 받은 ‘플레이어 정보 구조체’ 를 다시 클라이언트로 송신한다.  
- 서버와 클라이언트에서 수신 받은 ‘플레이어 정보 구조체’ 를 콘솔창에 출력해서 잘 작동하는 지 확인한다.

**| 게임 프레임워크에서의 추가 작업**게임 프레임워크에서는 위의 프로토타입에서 다음과 같은 작업들이 추가된다.  
- 서버는 클라이언트로부터 수신한 플레이어 정보를 이용하여 플레이어가 총알에 피격됐는지를 계산하고, 계산 결과를 클라이언트에 전송한다.  
- 클라이언트는 서버로부터 피격 당했다는 정보를 받으면 목숨을 하나 감소시키고 플레이어의 위치를 지정된 위치로 옮긴다. 만약 목숨이 모두 소진됐다면 관전 모드로 전환되어, 다른 플레이어들이 게임을 진행하는 것을 관전할 수 있다.

- 클라이언트는 서버로부터 받은 정보를 이용하여 본인과 다른 플레이어들을 렌더링한다.  
- 클라이언트는 렌더링 작업과 네트워크 송수신 작업을 동시에 진행하기 위해 네트워크 송수신용 쓰레드를 하나 만들어 사용한다.

- 서버는 목숨이 남아있는 플레이어가 1명이라면, 모든 플레이어에게 ‘게임종료’ 메시지를 보낸다.  
- 클라이언트는 ‘게임종료’ 메시지를 받은 후 승패 여부를 메시지 박스로 출력한 뒤, 클라이언트를 종료한다.



**로우레벨 디자인**

**| 공통  
| 패킷**  
0b00001 : GAME\_START (게임 시작)  
0b00010 : GAME\_OVER (게임 종료)  
0b00100 : PLAYER\_UPDATE (플레이어 정보 업데이트)  
0b01000 : PLAYER\_HIT (총알 피격)  
0b10000 : BULLET\_DELETED (총알 삭제)

**| 플레이어 정보 구조체(struct PlayerData)**XMFLOAT3 position // 위치  
XMFLOAT3 rotation // 회전 정보  
BOOL hasBullet // 총알 유무  
XMFLOAT3 bulletPosition // 총알 위치  
INT life // 목숨 수  
생성자 함수 PlayerData(),   
PlaterData(XMFLOAT3 Position, XMFLOAT3 rotate,   
int life, bool hasBullet, XMFLOAT3 bulletPosition)

**| 클라이언트**  
**| 쓰레드 함수**DWORD WINAPI ~~ProcessServerData~~TransportData(LPVOID arg)  
- 파라미터로는 ~~아무것도 받지 않는다.~~ Client의 SOCKET을 인자로 받는다.  
- 인자로 받은 소켓은 패킷 타입을 받기 위한 통신 소켓으로 사용한다.  
- 이 함수는 서버와의 연결이 완료된 상태에서 호출해야 한다.  
- 이 쓰레드 함수는 받은 패킷 타입에 따라 분기하여 필요한 수신 함수를 호출한다.

**| 송신 함수**void ~~CVehiclePlayer::~~ SendPlayerInfo(const SOCKET& sock)  
- 매 프레임마다 호출된다. PLAYER\_UPDATE 패킷과 플레이어 정보 구조체를 송신한다.

**| 수신 함수**void ~~CVehiclePlayer::~~RecvGameStart(const SOCKET& sock)  
- GAME\_START 패킷이 수신됐을 때 호출된다. 스폰 위치를 수신하고 저장한 뒤 플레이어를 스폰 위치로 옮긴다.

void ~~CVehiclePlayer::~~RecvGameEnd(const SOCKET& sock)  
- GAME\_OVER 패킷이 수신됐을 때 호출된다. 승패 여부를 수신하고 메시지 박스로 알려준다. 이후 서버와의 연결을 끊는다.

void ~~CGameFramework::~~RecvPlayerInfo(const SOCKET& sock)- PLAYER\_UPDATE 패킷이 수신됐을 때 호출된다. 다른 2명의 플레이어의 구조체 정보를 수신하고 저장한다. PLAYER\_HIT 패킷이 포함되어 있을 경우에는 체력을 1깎고 스폰 위치로 이동시킨다.

void ~~CVehiclePlayer::~~RecvBulletDeleted() @@@나중에 함수화 함@@@- BULLET\_DELETED 패킷이 수신됐을 때 호출된다. 보유 중인 총알을 삭제한다.

**| 서버**

**| 쓰레드 함수**DWORD WINAPI ProcessClientData(LPVOID arg)  
- 파라미터로는 INT id, SOCKET sock를 받는다.  
- 해당 id의 클라이언트의 데이터 송수신을 담당하는 함수이다.  
- 이 쓰레드 함수는 받은 패킷 타입에 따라 분기하여 필요한 수신 함수를 호출한다.  
- 쓰레드 동기화는 이벤트 객체를 사용한다. 1, 2, 3번 쓰레드와 1, 2, 3번 이벤트가 있고 각 쓰레드는 자기 번호에 맞는 이벤트가 신호 상태가 될 때까지 대기한다. 만약 1번 이벤트가 신호 상태가 됬다면 1번 쓰레드의 작업을 진행하고 끝난 뒤 1번 이벤트를 비신호 상태로 만들고 2번 이벤트를 신호 상태로 변경한다. 이는 2, 3번 쓰레드에도 동일하게 적용된다.

**| ThreadFuncParam 구조체**- int id, SOCKET sock을 구조체 멤버변수로 가진다.   
- 송수신함수의 인자로 사용한다.

**| 송신 함수**void SendGameStart(ThreadFuncParam\* param)  
- 모든 클라이언트에게 GAME\_START 패킷과 함께 각 클라이언트마다 지정된 좌표를 송신한다.

void SendGameOver(ThreadFuncParam\* param)  
- 모든 클라이언트에게 GAME\_OVER 패킷과 함께 각 클라이언트에게 승패 여부를 송신한다.

void SendPlayerInfo(~~INT id~~ ThreadFuncParam\* param)  
- 해당 id를 제외한 클라이언트에게 PLAYER\_UPDATE 패킷과 해당 id의 플레이어 정보 구조체를 송신한다. 총알에 피격당한 클라이언트에게는 PLAYER\_HIT 패킷도 같이 송신한다.

void SendBulletDeleted(~~INT id~~ ThreadFuncParam\* param)  
- 해당 id의 클라이언트에게 BULLET\_DELETED 패킷을 송신한다.

**| 수신 함수**void RecvPlayerInfo(~~INT id~~ ~~const SOCKET& sock~~ ThreadFuncParam\* param)  
- PLAYER\_UPDATE 패킷이 수신됐을 때 호출되는 함수이다. 플레이어 정보 구조체를 수신하고 ~~자신이 담당하는 id의~~ 클라이언트 정보를 최신화한다.

**| 그 외의 함수**BoundingOrientedBox GetBoundingBox(INT id)  
- 해당 id의 클라이언트 정보를 이용해서 바운딩 박스를 만든 후 반환한다.

BOOL BulletCollisionCheck(~~INT id~~ XMFLOAT3 playerPosition, XMFLOAT3 playerRotate, XMFLOAT3 BulletPosition)  
- 해당 id의 바운딩 박스와 다른 플레이어의 총알 위치 간의 충돌을 체크한다. 만약 충돌했을 경우, 총알의 주인에겐 BULLET\_DELETED 패킷을 송신하고 TRUE를 반환한다. 그 외에는 FALSE를 반환한다.

BOOL ~~GameOverCheck~~IsGameOver()  
- 플레이어 생존 유무를 확인해서 1명만 생존했다면 모든 클라이언트에게 GAME\_OVER 패킷을 전송하고 TRUE를 반환한다. 그 외에는 FALSE를 반환한다.

**팀원 별 역할 분담 및 개발 환경**

**| 팀원 별 역할 분담**클라이언트 송신 함수, 서버 수신 함수 : 원윤식  
클라이언트 수신 함수, 서버 송신 함수 : 최원재  
소켓 설정 및 프레임워크 적용: 정한솔

**| 개발 환경**Visual Studio 2019  
Github

**개발일정** 원윤식

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 11/14 | 11/15 | 11/16 | 11/17 | 11/18 | 11/19 | 11/20 |
|  | 서버 프로젝트 생성,  서버 소켓 생성 및 설정,  서버 필요한 함수 선언 |  | 서버 쓰레드  함수 작성 |  | 클라 SendPlayerInfo  서버 RecvPlayerInfo  함수 작성 |  |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 11/21 | 11/22 | 11/23 | 11/24 | 11/25 | 11/26 | 11/27 |
|  | 서버 쓰레드 함수의  쓰레드 동기화 구현 |  |  |  | 외부 네트워크에서의  접속을 위한 보안 설정 |  |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 11/28 | 11/29 | 11/30 | 12/01 | 12/02 | 12/03 | 12/04 |
| 플레이 및  버그 테스트 | 자기 역할 내 버그 수정 |  |  |  |  | 최종 검수 및  릴리즈 |
|  |  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 12/05 | 12/06 | 12/07 |  |
|  |  |  |
|  |  |  |

최원재

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 11/14 | 11/15 | 11/16 | 11/17 | 11/18 | 11/19 | 11/20 |
|  | 클라 쓰레드  함수 작성 |  | 서버 SendGameStart  클라 RecvGameStart  함수 작성 |  | 서버 SendPlayerInfo 클라 RecvPlayerInfo  함수 작성 |  |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 11/21 | 11/22 | 11/23 | 11/24 | 11/25 | 11/26 | 11/27 |
|  | 서버 SendBulletDeleted  클라 RecvBulletDeleted  함수 작성 | 서버 SendGameOver  클라 RecvGameOver  함수 작성 |  |  |  |  |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 11/28 | 11/29 | 11/30 | 12/01 | 12/02 | 12/03 | 12/04 |
| 플레이 및  버그 테스트 | 자기 역할 내 버그 수정 |  |  |  |  | 최종 검수 및  릴리즈 |
|  |  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 12/05 | 12/06 | 12/07 |  |
|  |  |  |
|  |  |  |

정한솔

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 11/14 | 11/15 | 11/16 | 11/17 | 11/18 | 11/19 | 11/20 |
|  | 서버, 클라이언트  플레이어 정보 구조체,  패킷 선언 |  |  |  | 게임프레임워크  임계 영역 설정 | 서버  GetPlayerInfo  GetBoundingBox  함수 작성 |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 11/21 | 11/22 | 11/23 | 11/24 | 11/25 | 11/26 | 11/27 |
| 서버 BulletCollisionCheck  함수 작성 |  | 서버 GameOverCheck  함수 작성 |  |  | 수신한 정보를 이용하여  다른 플레이어 렌더링 | |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 11/28 | 11/29 | 11/30 | 12/01 | 12/02 | 12/03 | 12/04 |
| 플레이 및  버그 테스트 | 자기 역할 내 버그 수정 |  |  |  |  | 최종 검수 및 릴리즈 |
|  |  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 12/05 | 12/06 | 12/07 |  |
|  |  |  |
|  |  |  |