typedef struct pos **좌표를 위한 구조체**

{

float xpos;

float ypos;

float zpos;

}Pos;

typedef struct player\_info **빌딩 구조체에 맴버변수로 쓸 PInfo**

{

Pos \*pos;

int hp;

int state;

}PInfo;

typedef struct bullet **총알은 좌표를 가지고 있음**

{

Pos \*pPos;

}Bullet;

typedef struct shield **방패는 체력과 좌표를 가지고 있음**

{

int hp;

Pos \*pPos;

}Sheild;

struct Building //나 **상대방에게 보낼 내 정보가 담긴 구조체**

{

int iBuildingState; //0 - 단방향, 1 - 사방, 2 - 폭탄

Bullet \*pBullet;

PInfo \*pInfo;

Sheild \*pShield;

};

struct Enemy **상대방으로부터 받을 정보의 구조체**

{

int iBuildingState; //0 - 단방향, 1 - 사방, 2 - 폭탄

Bullet \*pBullet;

PInfo \*pInfo;

Sheild \*pShield;

};

class GameFramework //내것은 스스로 그리고, 상대정보는 받아서 그림

{

public:

SOCKET server\_socket; //서버랑 통신하기 위한 소켓

SOCKET cleint\_socket;

public:

Scene \* m\_pScene;

Enemy \* m\_pEnemy;

public:

int SendToOpponent(SOCKET socket, Building \*building, int len, 0);

int RecvFromOpponent(SOCKET socket, Enemy \* enemy, int len, 0);

void SetOpponentData() { m\_pScene->SetOpponetData(m\_pEnemy);

**RecvFromOpponent() 함수로 받은 enemy 정보를 Scene에 넘겨주고 Scene에서 게임 플레이를 위한 업데이트를 진행**

}

public:

void MouseInput(int button, int state, int x, int y) {} //클릭 확인

void Update(float elapsedTime) { m\_pScene->Update(); }

bool IsGameOver(); // 게임종료 확인

};

class Scene

{

public:

Enemy \*m\_pEnemyInfo;

public:

bool IsCollide(Bullet &bullet, Building & building) { /\*충돌로직\*/ } //총알과 빌딩이 충돌하면 true

void SetOpponetData(Enemy \*enemy) { m\_pEnemyInfo = enemy; }

void DrawMyData(); // 내꺼 포탑과 총알 정보 그림

void DrawOpponetData(Enemy \*enemy) {} //받아온 상대방 포탑과 총알 정보 그림

void Update(float elapsedTime) { } //내정보 업데이트, /\*총알 좌표 업데이트, 충돌체크 확인, 충돌하면 체력 감소, 체력 0이면 삭제\*/ /\*쉴드 체력 관리\*/

bool BuildObject(); //처음 배치 어떻게 할것인지, 배치 끝나면 데이터 송수신 시작

};