

네트워크 게임 프로그래밍

-Term 프로젝트 추진 계획서-

2018180044 최해성

2018182031 이지형

2019184036 홍명진

1. 애플리케이션 기획

A. 게임소개

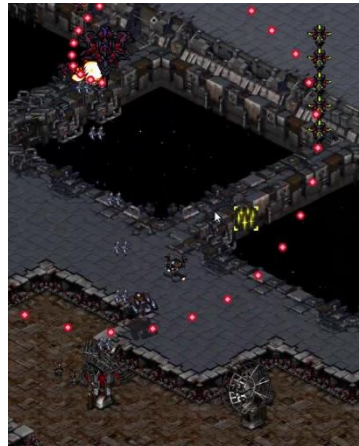
게임: 스타크래프트 이미지를 사용한 갤러그

게임 컨셉: 2인의 플레이어가 비행기 모양의 플레이어 오브젝트를 조작하면서 아이템을 먹고 총알을 발사해 적을 쓰러뜨리면서 스테이지를 진행하고 보스몬스터를 잡으면 클리어한다.

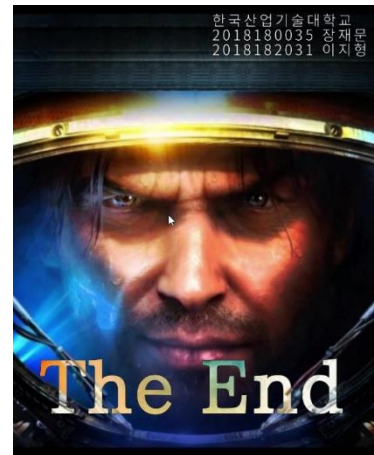
타이틀



스테이지



클리어



B. 조작방법

이동: W/A/S/D

공격: Enter Space

C. 몬스터 이동 패턴

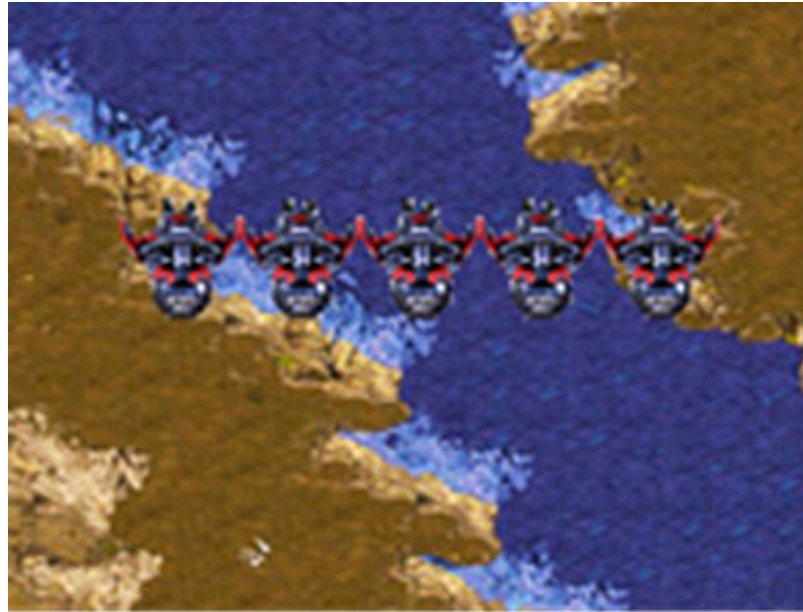
- 교차형-



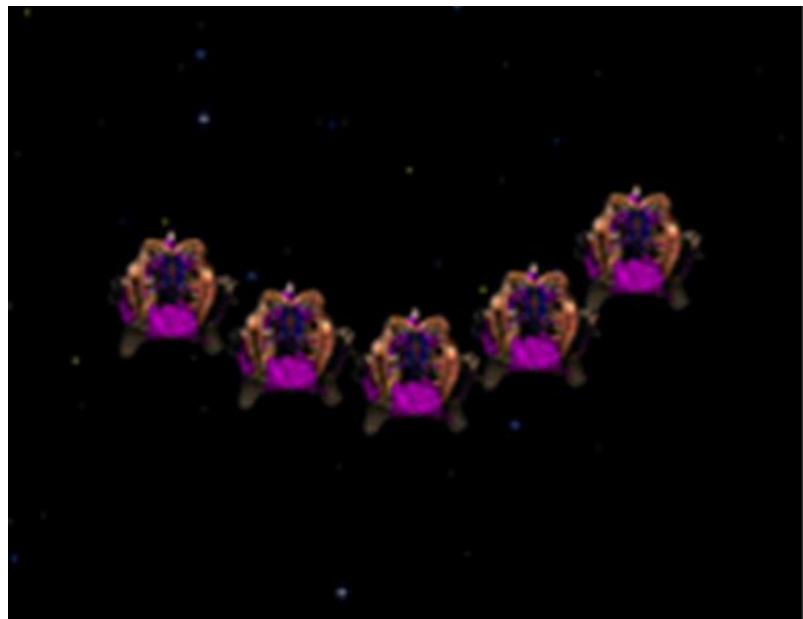
- 직선형 - 수직



● 직선형 — 수평



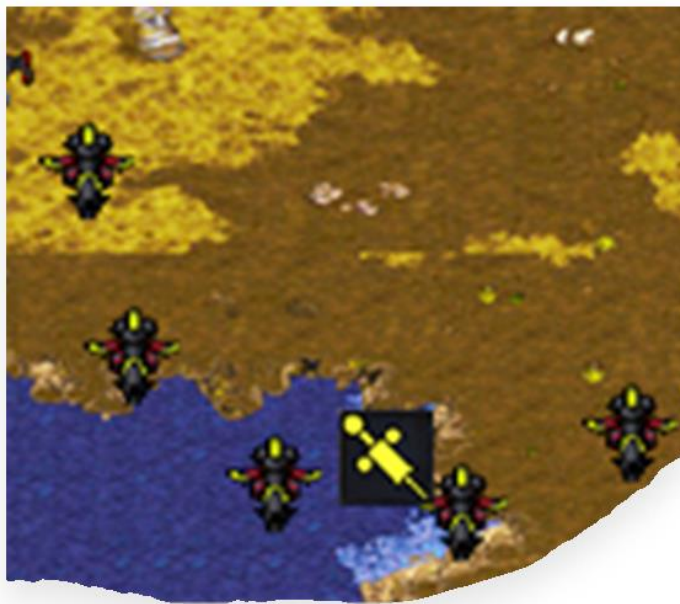
● 유턴형



- 계단형



- 고리형

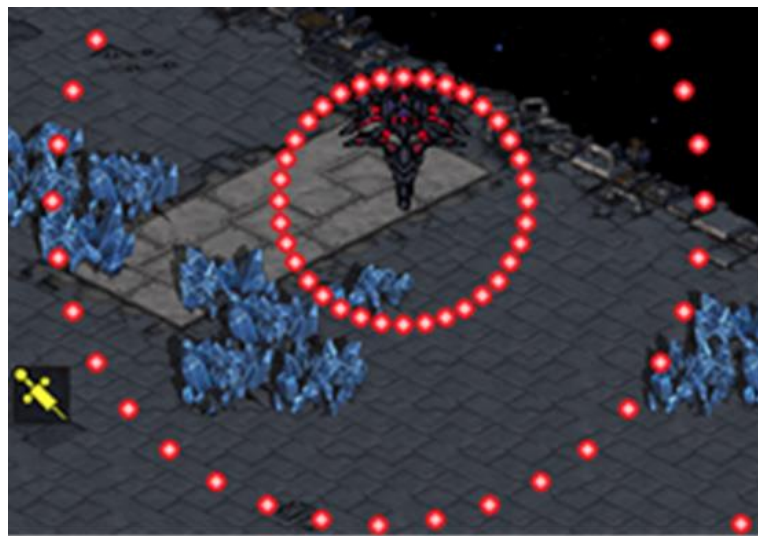


- 특수형: 트랙터



D. 보스패턴

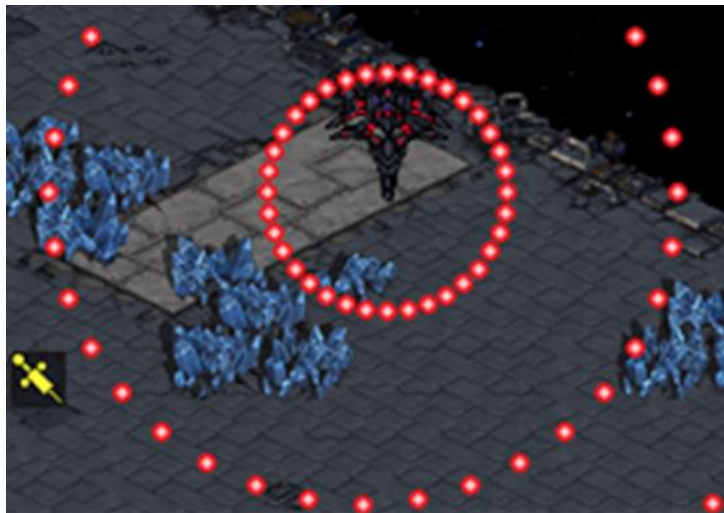
- 분사형



- 소용돌이형



- 흩뿌리기



- 흘뿌리기 2



- 나선형



E. 서버

1. 클라이언트들이 서로의 플레이어 객체의 위치를 공유할 수 있게 한다.
2. 총알 & 몬스터의 충돌을 확인하고 몬스터의 상태를 업데이트 한다.
3. 몬스터들과 총알들의 위치를 클라이언트끼리 같게 한다.
4. 플레이어 & 몬스터의 충돌을 확인한다.
5. 플레이어 & 아이템의 충돌을 확인한다
6. 게임의 상태를 클라이언트끼리 같게 한다.
7. 아이템을 생성한다.

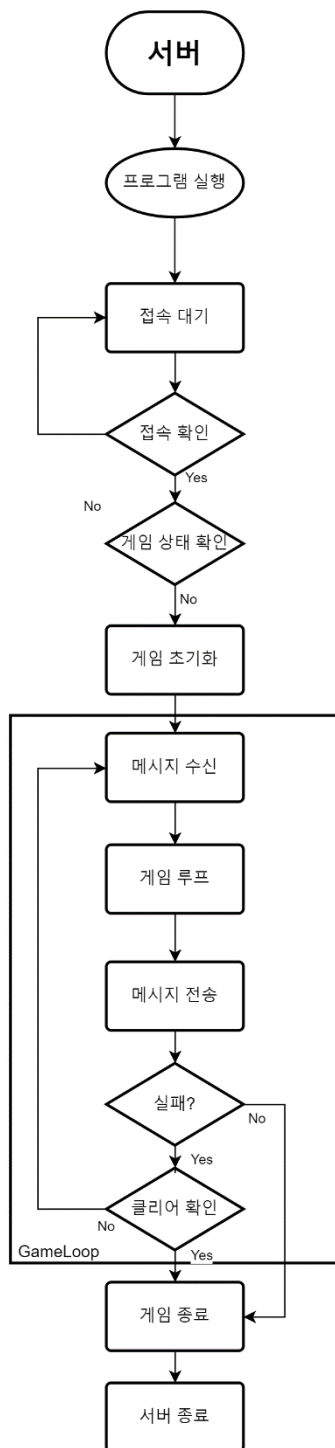
F. 클라이언트

1. 플레이어 상태를 업데이트 한다.
2. 클라이언트 사용자의 키 입력을 받아 처리한다.
3. 사운드를 재생한다.
4. 서버로부터 오는 메시지를 처리한다.
5. 서버로 플레이어 객체의 위치정보와 플레이어 총알정보를 전달한다.
6. 화면을 렌더링한다.

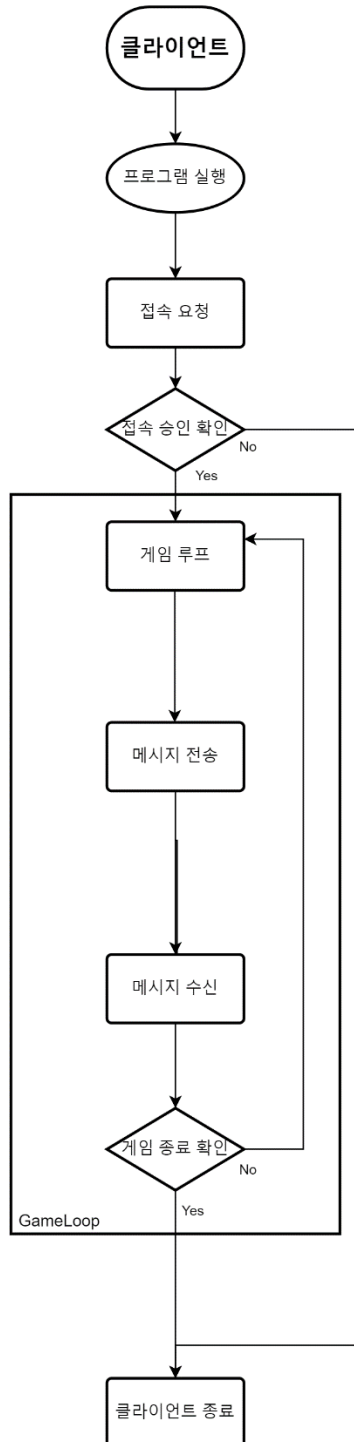
2. High Level 디자인

A. 서버에서의 Data Flow

서버는 프로그램이 실행되면 윈도우 소켓을 초기화 하고, 대기 소켓을 생성해 클라이언트의 접속을 대기하다가 클라이언트가 접속 요청을 하면 현재 게임 상태를 확인해서 게임 초기화를 합니다. 진행중인 게임이 없으면 새로운 게임을 시작하고 이미 진행중인 게임이 있다면 새로운 플레이어를 진행중인 게임에 첨가시킵니다. 서버는 클라이언트로부터 메시지를 받으면 받은 데이터를 이용해 게임을 업데이트하고 업데이트 된 데이터를 클라이언트로 전송합니다. 모든 플레이어가 사망해 게임이 종료되거나 보스 몬스터를 잡아 게임이 클리어 되면 서버를 종료합니다.

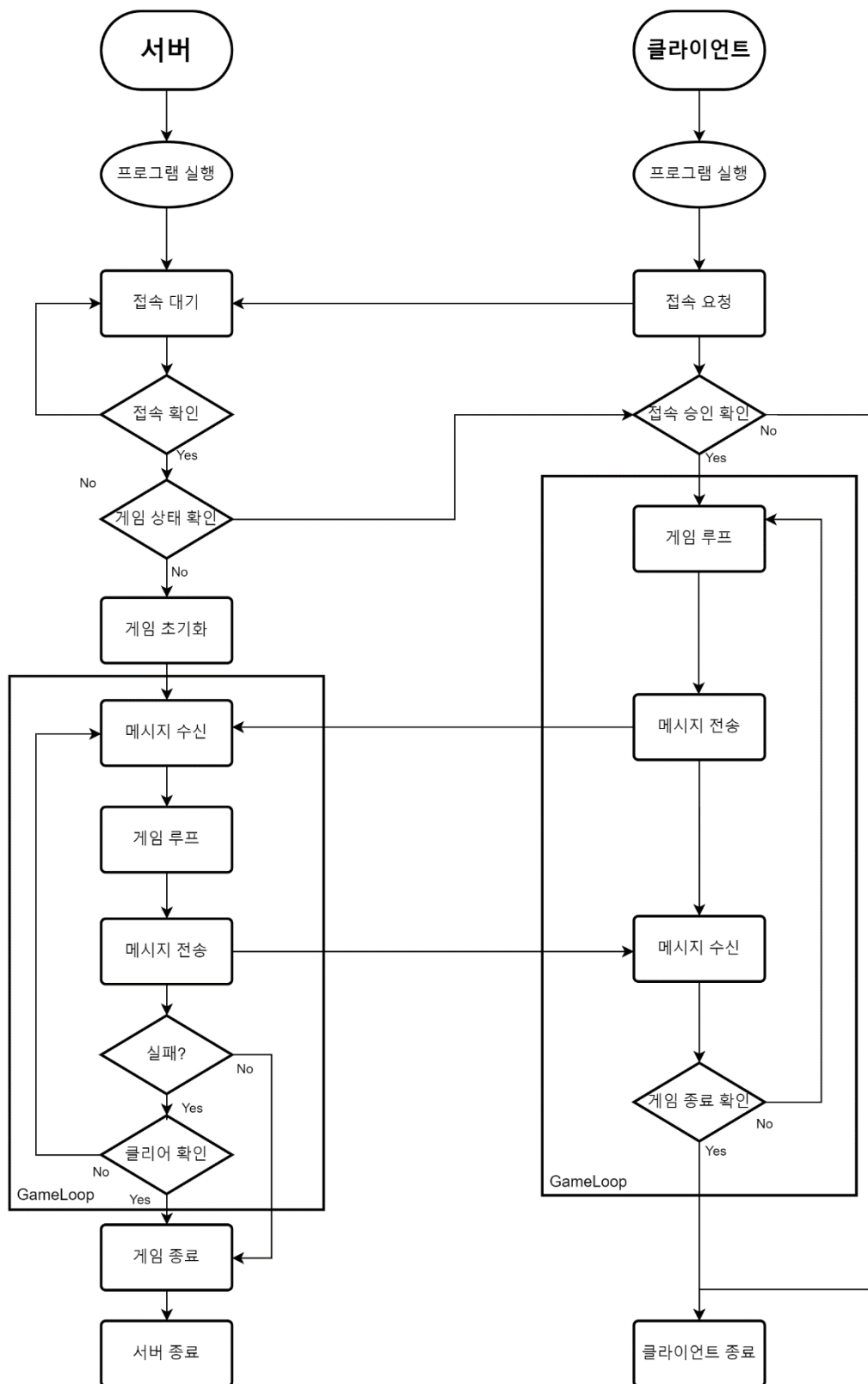


B. 클라이언트에서의 Data Flow



클라이언트는 프로그램이 실행되면 윈도우 소켓을 초기화하고 소켓을 생성해 서버에 접속을 요청합니다. 서버로부터 접속 승인 메시지를 받아 접속이 승인되었다면 게임루프를 실행하고 접속이 승인되지 않았다면 프로그램을 종료합니다. 접속을 승인받아 게임루프에 진입한 클라이언트는 서버에 클라이언트의 정보를 담은 메시지를 보내고 서버로부터 결과 메시지를 받아 게임 종료를 확인해 게임이 종료되었을 경우 클라이언트를 종료하고 게임이 종료되지 않았을 경우 다시 게임루프를 돌게 됩니다.

C. 서버 & 클라이언트에서의 Data Flow



D. 서버 모듈

모듈 이름	기능
CCore	전반적인 프로그램의 흐름을 관리합니다.
CTimer	프로그램의 시간을 동기화합니다.
CSceneManager	씬을 관리합니다.
CScene	씬의 세부정보를 관리합니다.
CObjectManger	게임에 필요한 객체들을 관리합니다.
CObject	게임에 들어가는 객체들의 세부정보를 관리합니다.
CItem	게임 내 아이템들을 관리합니다.
CBullet	게임 내 총알들을 관리합니다.
CNetworkDevice	클라이언트와 메시지를 주고 받으며 메시지를 해석합니다.
CMessageDispatcher	메시지에 따라 객체들에게 메시지를 전달합니다.
Telegram	메시지의 세부정보를 관리합니다.
CLocator	중개자입니다. 플레이어와 네트워크 디바이스를 중개합니다.
CPath	몬스터의 이동 경로를 저장 및 시뮬레이션 합니다.
CMonster	몬스터의 세부정보를 관리합니다.
CBoss	보스몬스터의 세부정보를 관리합니다.

E. 클라이언트 모듈

모듈 이름	기능
CCore	전반적인 프로그램의 흐름을 관리합니다.
CTimer	프로그램의 시간을 동기화합니다.
CSceneManager	프로그램의 시간을 동기화합니다.
CScene	씬을 관리합니다.
CObjectManger	씬의 세부정보를 관리합니다.
CObject	게임에 필요한 객체들을 관리합니다.
CPlayer	플레이어의 세부정보를 관리합니다.
CMonster	몬스터의 세부정보를 관리합니다.
CBoss	보스몬스터의 세부정보를 관리합니다.
CTractor	트랙터의 세부정보를 관리합니다.
CItem	아이템의 세부정보를 관리합니다.
CBullet	총알의 세부정보를 관리합니다.
CSoundManager	게임 내 사운드를 관리합니다.
CNetworkDevice	서버와 메시지를 주고 받으며 메시지를 해석합니다.
CMessageDispatcher	메시지에 따라 객체들에게 메시지를 전달합니다.
Telegram	메시지의 세부정보를 관리합니다.

3. Low Level 디자인

A. 서버 모듈

CCore		
함수 이름	리턴 타입	기능
Init	bool	CTimer, CSceneManager, CObjectManage, CNetworkDevice 를 초기화합니다.
Collision	void	CSceneManager 의 Collision 함수를 호출한다.
Update	int	CSceneManager 의 Update 함수를 호출한다.
LateUpdate	int	CSceneManager 의 LateUpdate 함수를 호출한다.
Logic	void	클라이언트로부터 메시지를 받고 처리한 후 타이머를 업데이트하고 게임 루프를 실행한 뒤 클라이언트로 메시지를 보낸다.
SetPlayerHandle	void	플레이어의 쓰레드 아이디를 셋한다.
SnapshotRun	Void	썬 전체의 스냅샷을 메시지로 만들어 클라이언트로 전송한다

CTimer		
함수 이름	리턴 타입	기능
Init	bool	타이머를 초기화한다.
Update	void	프레임 타임을 계산한다
GetDeltaTime	float	프레임 타임을 반환한다.
GetTime	LONGLONG	타이머의 현재 퍼모먼스 카운터의 QuadPart 를 반환한다.

CSceneManager		
함수 이름	리턴 타입	기능
Init	bool	로딩 CScene, 스테이지 CScene, 클리어 CScene, 사망 CScene,

		CPlayer 를 생성하고 초기화한다.
Update	int	현재 CScene 에 대한 Update 함수를 호출한다.
LateUpdate	ini	현재 CScene 에 대한 LateUpdate 함수를 호출한다.
Collsion	Void	현재 CScene 에 대한 Collsion 함수를 호출한다.
GetPlayer1	CPlayer*	1 번 플레이어의 포인터를 반환한다.
GetPlayer2	CPlayer*	2 번 플레이어의 포인터를 반환한다.
GetItemlistFromSceneType	std::list<CItem*>*	지정된 씬의 아이템 리스트를 반환한다.
GetBulletListFromSceneType	std::list<CBullet*>*	지정된 씬의 볼릿 리스트를 반환한다.
GetCurrentSceneType	SCENE_TYPE	현재 활성화된 씬 타입을 반환한다.
HandleMessage	bool	받은 메시지를 처리한다.

CScene		
함수 이름	리턴 타입	기능
Init	bool	ItemList 를 생성하고 ItemList 와 CScene 을 초기화한다.
Update	int	아이템을 생성하고 오브젝트를 Update 해준다.
LateUpdate	int	몬스터를 삭제하고 플레이어의 LateUpdate 를 호출한다.

GetCurScene MaxDistance	int	스테이지 끝단의 거리를 가져온다.
UpdateMaxDistance	void	현재 거리를 업데이트한다.
GetItemList	std::list<CItem*>*	씬의 아이템 리스트의 포인터를 반환한다.
GetBulletList	CBulletList*	씬의 블릿 리스트의 포인터를 반환한다
UpdateBulletListWithBoss	void	보스의 블릿 리스트의 블릿들을 씬의 블릿 리스트로 이동시킨다.

CObjectManager		
함수 이름	리턴 타입	기능
RegisterObject	void	객체를 오브젝트 매니저에 등록한다.
GetObjectFromID	CObject*	ID 값을 받아 객체의 주소를 반환한다.
RemoveObject	void	객체를 오브젝트 매니저에서 삭제한다.

CObject		
함수 이름	리턴 타입	기능
Init	bool	오브젝트의 이미지를 로드하고 오브젝트를 초기화해줍니다.
Update	void	오브젝트를 업데이트합니다.
LateUpdate	void	오브젝트를 후처리합니다.
Collision	bool	오브젝트의 충돌을 확인합니다.
Changeimg		
GetPos	POSITION	오브젝트 이미지의 왼쪽 위 좌표를 리턴해줍니다.
GetVector	POSITION	오브젝트의 진행 방향을 리턴해줍니다.
GetSize	_SIZE	오브젝트의 크기를 리턴해줍니다.
GetLife	bool	오브젝트의 생존 여부를 리턴해줍니다.
GetHP	float	오브젝트의 체력값을 리턴해줍니다.
SetPos	void	오브젝트의 위치를 설정합니다.
SetVector	void	오브젝트의 진행 방향을 설정합니다.
SetSize	void	오브젝트의 사이즈를 설정합니다.
SetLife	void	오브젝트의 생존 여부를 설정합니다.
SetHP	void	오브젝트의 체력값을 설정합니다.

CItem

함수 이름	리턴 타입	기능
GetEnbale	bool	아이템 유효 여부를 확인합니다.
SetEnalbeFalse	void	아이템을 비활성화합니다.
GetType	ITEM_TYPE	아이템 타입을 반환합니다.
SetEffect	void	아이템의 효과를 플레이어에게 적용한다.

CBullet		
함수 이름	리턴 타입	기능
GetEnbale	bool	아이템 유효 여부를 확인합니다.
SetEnalbeFalse	void	아이템을 비활성화합니다.

CNetworkDevice		
함수 이름	리턴 타입	기능
SendToNetwork	void	CMessageDispatcher로부터 메시지를 전달받아 가공해서 클라이언트로 전송한다.
RecvByNetwork	void	클라이언트로부터 메시지를 받아 해석해서 CMessageDispatcher로 전달한다.
GetTelegram	void	네트워크 디바이스의 적재된 메시지를 CMessageDispatcher의 큐에 담는다.
AddMessage	void	메시지를 디바이스의 전송 큐에 저장한다.

CMessageDispatcher		
함수 이름	리턴 타입	기능
GetMessages	Telegram*	메시지 큐의 데이터를 꺼내서 리턴해준다.
Discharge	void	메시지를 특정 객체에게 전달해준다.
DispatchMessage	void	메시지 큐에 메시지를 담는다.

CLocator		
함수 이름	리턴 타입	기능
SetNetworkDevice	void	스레드 ID와 네트워크 디바이스의 페어를 std::set에 담는다
SetNetworkPtrWithHandle	void	스레드 ID와 짝지어진 네트워크 디바이스를 교체한다.

GetNetworkDevice	CNetworkDevice*	스레드 ID 와 짝지어진 네트워크 디바이스 객체를 반환한다.
SetPlayer	void	n 번 플레이어의 객체를 저장한다.
GetPlayer	CPlayer*	n 번 플레이어의 객체를 반환한다.

CPath		
함수 이름	리턴 타입	기능
AddPoint	void	경로를 추가한다.
SetTension	void	텐션 값을 설정한다.
Update	void	현재 경로의 인덱스와 t 값을 업데이트한다.
GetPathEnd	bool	경로의 끝에 도달했는지에 대한 여부를 반환한다.
GetNextPos	POSITION*	현재 인덱스와 t 값에 따른 카디널 스플라인 결과를 반환한다.
CardinalSpline	POSITION	카디널 스플라인을 하여 위치를 반환한다.

CMonster		
함수 이름	리턴 타입	기능
CreateBullet	void	블릿을 생성하여 블릿 리스트에 적재한다.

CBoss		
함수 이름	리턴 타입	기능
CreateBullet	void	지정된 패턴에 대한 블릿을 생성하여 블릿 리스트에 적재한다.

B. 클라이언트 모듈

CCore		
함수 이름	리턴 타입	기능
Init	bool	CTimer, CSceneManager, CSoundManager, CObjectManage,

		CNetworkDevice 를 초기화하고 서버로 메시지를 보냅니다.
Input	void	CSceneManager 의 Input 함수를 호출한다.
Update	int	CSceneManager 의 Update 함수를 호출한다.
LateUpdate	int	CSceneManager 의 LateUpdate 함수를 호출한다.
Logic	void	타이머를 업데이트하고 게임 루프를 실행하고 렌더링하고 서버로 메시지를 전송하고 다시 받는다.
SendSnapShot	void	클라이언트의 플레이어 정보, 플레이어 총알 정보를 서버로 전송하기 위한 메시지를 적재합니다.

CTimer		
함수 이름	리턴 타입	기능
Init	bool	타이머를 초기화한다.
Update	void	프레임 타임을 계산한다
GetDeltaTime	float	프레임 타임을 반환한다.

CSceneManager		
함수 이름	리턴 타입	기능
Init	bool	필요한 이미지들을 로드하고, 로딩 CScene, 스테이지 CScene, 클리어 CScene, 사망 CScene, CPlayer 를 생성하고 초기화한다.
Input	void	현재 CScene 의 Input 함수를 호출합니다.
Update	int	현재 CScene 에 대한 Update 함수를 호출한다.
LateUpdate	ini	현재 CScene 에 대한 LateUpdate 함수를 호출한다.
Render	void	현재 CScene 의 Render 함수를 호출합니다.

CScene		
함수 이름	리턴 타입	기능

Init	bool	배경이미지를 로드하고 ItemList 를 생성하고 ItemList 와 CScene 을 초기화한다.
imgLT_Move_Auto	void	배경이미지를 스크롤링합니다.
Update	int	아이템을 생성하고 오브젝트를 Update 해준다.
LateUpdate	int	몬스터를 삭제하고 플레이어의 LateUpdate 를 호출한다.
GetCurScene MaxDistance	int	스테이지 끝단의 거리를 가져온다.
Render	void	렌더링을 한다.
UpdateMaxDistance	void	현재 거리를 업데이트한다.
Input	void	사용자의 입력을 처리한다.
ReleaseObjects	void	플레이어 총알, 몬스터, 아이템, 몬스터 총알을 삭제한다.
AddObject	void	오브젝트 타입정보와 위치정보를 받아 오브젝트를 추가한다.

CObjectManager		
함수 이름	리턴 타입	기능
RegisterObject	void	객체를 오브젝트 매니저에 등록한다.
GetObjectFromID	CObject*	ID 값을 받아 객체의 주소를 반환한다.
RemoveObject	void	객체를 오브젝트 매니저에서 삭제한다.

CObject		
함수 이름	리턴 타입	기능
Init	bool	오브젝트의 이미지를 로드하고 오브젝트를 초기화해줍니다.
Update	void	오브젝트를 업데이트합니다.
LateUpdate	void	오브젝트를 후처리합니다.
Collision	bool	오브젝트의 충돌을 확인합니다.
Changeimg		
GetPos	POSITION	오브젝트 이미지의 왼쪽 위 좌표를 리턴해줍니다.
GetVector	POSITION	오브젝트의 진행 방향을 리턴해줍니다.
GetSize	_SIZE	오브젝트의 크기를 리턴해줍니다.
GetLife	bool	오브젝트의 생존 여부를 리턴해줍니다.
GetHP	float	오브젝트의 체력값을 리턴해줍니다.

SetPos	void	오브젝트의 위치를 설정합니다.
SetVector	void	오브젝트의 진행 방향을 설정합니다.
SetSize	void	오브젝트의 사이즈를 설정합니다.
SetLife	void	오브젝트의 생존 여부를 설정합니다.
SetHP	void	오브젝트의 체력값을 설정합니다.
HandleMessage	void	메시지를 처리합니다.
GetState	OBJECT_STATE	오브젝트의 상태를 리턴해줍니다.
RegisterObject	void	오브젝트의 ID 를 받아 해당 ID 를 저장하고 CObjectManager 에 등록합니다.

CPlayer		
함수 이름	리턴 타입	기능
SetHitTractorBeam	void	m_bHit_TractorBeam(트랙터 공격 파격 여부)를 설정합니다.
GetHitTractorBeam	void	m_bHit_TractorBeam(트랙터 공격 파격 여부)를 리턴해줍니다.
GetmyBulletList	CBulletList*	플레이어 총알 리스트를 리턴해줍니다.
IsPlayerDie	bool	플레이어 사망여부를 리턴해줍니다.
AddBullet	void	총알의 숫자를 확인하는 변수를 +1 해줍니다.
AddAttack	void	플레이어의 공격력을 증가시키기위한 함수입니다.
SetINVINIBILITY	void	플레이어의 무적상태를 표현하는 m_bINVINIBILITY 를 설정합니다.
GetINVINIBILITY	bool	플레이어의 무적상태를 표현하는 m_bINVINIBILITY 를 리턴합니다.
SetHPPrototype	void	
GetSpeed	float	플레이어 속도를 리턴합니다.
SetSpeed	void	플레이어 속도를 설정합니다.
GetMoveDistance	float	플레이어의 이동 거리를 계산해 리턴합니다.
SetType	void	플레이어의 Type 을 설정해줍니다.
GetMyType	OBJECT_TYPE	플레이어의 Type 을 리턴합니다.
RenderBulletList	void	플레이어 총알을 모두 렌더링합니다.

CMonster		
함수 이름	리턴 타입	기능
GetIsDie	bool	Monster 생존 여부를 확인합니다.

RenderExplode	void	폭발 애니메이션을 렌더링합니다.
Run	void	CMonster 의 Update, LateUpdate, Render 를 호출합니다.
GetState	MONSTER_STATE	몬스터의 상태를 리턴합니다.
GetFireDelay	int	총알 발사 딜레이 시간을 리턴합니다.

CItem		
함수 이름	리턴 타입	기능
GetEnbale	bool	아이템 유효 여부를 확인합니다.
SetEnalbeFalse	void	아이템을 비활성화합니다.
GetType	ITEM_TYPE	아이템 타입을 반환합니다.

CBullet		
함수 이름	리턴 타입	기능
GetEnbale	bool	아이템 유효 여부를 확인합니다.
SetEnalbeFalse	void	아이템을 비활성화합니다.

CSoundManager		
함수 이름	리턴 타입	기능
playSound	bool	배경음악을 재생합니다.
stopSound	bool	배경음악 재생을 중지합니다.
PlaySound_Bossbgm	bool	보스 배경음악을 재생합니다.
playSound_Effect	bool	효과음을 재생합니다.
playSound_Effect_Explode	bool	폭발 효과음을 재생합니다.
HandleMessage	void	메시지를 처리합니다.

CNetworkDevice		
함수 이름	리턴 타입	기능
SendToNetwork	void	Message 들을 형식을 변환해 서버로 보냅니다.
RecvByNetwork	void	서버로부터 메시지를 받아 해석합니다.
AddMessage	void	메시지 큐에 메시지를 추가합니다.

ConnectNetwork	bool	원속 초기화와 소켓 생성 및 클라이언트와 네트워크 연결을 합니다.
----------------	------	--------------------------------------

CMessageDispatcher		
함수 이름	리턴 타입	기능
Discharge	void	메시지를 특정 객체에게 전달해준다.
DispatchMessage	void	메시지 큐에 메시지를 담는다.

C. 메시지 구조체

Msg_type	sender	receiver	extrainfo
msg_objectCreate	server/client	objectID	ObjectType, position
msg_objectMove	server/client	objectID	position
msg_playerHit	server	objectID	X
msg_objectChangeState	server	objectID	ObjectState
msg_playSound	server	objectID	int
msg_changeScene	server	objectID	int
msg_clientReady	client	objectID	X

D. 패킷 구조 (고정 + 가변 길이 데이터 전송 방식)

메시지 타입 별 개수(28 바이트) + 메시지 데이터(n 바이트)

메시지 데이터의 크기 $n = (\text{msg_objectCreate의 개수}) * 24 + 4 +$
 $(\text{msg_objectMove의 개수}) * 20 + 4 +$
 $(\text{msg_playerHit의 개수}) * 4 + 16 +$
 $(\text{msg_objectChangeState의 개수}) * 8 + 20 +$
 $(\text{msg_playSound의 개수}) * 8 + 20 +$
 $(\text{msg_changeScene의 개수}) * 8 + 20 +$
 $(\text{msg_clientReady의 개수}) * 4 + 16$

E. 패킷 송수신 방식

CNetworkDevice 는 각 메시지 타입별로 Telegram 배열을 지닌다.

CNetworkDevice::AddMessage() 를 호출하여 네트워크 디바이스에 메시지를 전달하여 저장한다.

CNetworkDevice::SendToNetwork() 가 호출되면 네트워크 디바이스는 먼저 각 메시지 타입 별 메시지의 개수를 int 형 배열에 저장한다.

메시지의 종류는 서버와 클라이언트가 모두 알고 있으므로 고정길이 패킷으로 전달이 가능하므로 이렇게 만든 int 형 배열을 고정길이 패킷으로 먼저 송신한다.

그리고 각 메시지 타입별로 나뉘어서 저장된 배열에서 Telegram 을 하나씩 꺼내

[receiver, extrainfo] 순으로 직렬화하여 가변길이 패킷 버퍼에 순차적으로 넣는다.

이때, 메시지타입의 값이 낮은 순부터 넣는다.

반대로 CNetworkDevice::RecvByNetwork() 가 호출되면 받아들인 고정길이 패킷을 int 형 배열로 받아 각 인덱스를 순차적으로 접근하여 메시지 타입별 메시지 개수를 저장한다.

이후 가변길이 패킷을 받아 이중 반복문을 돌린다.

첫번째 반복은 int 형 배열의 크기만큼 반복하며 두번째 반복은 int 형 배열의 현재 i 번째 원소의 값만큼 반복한다.

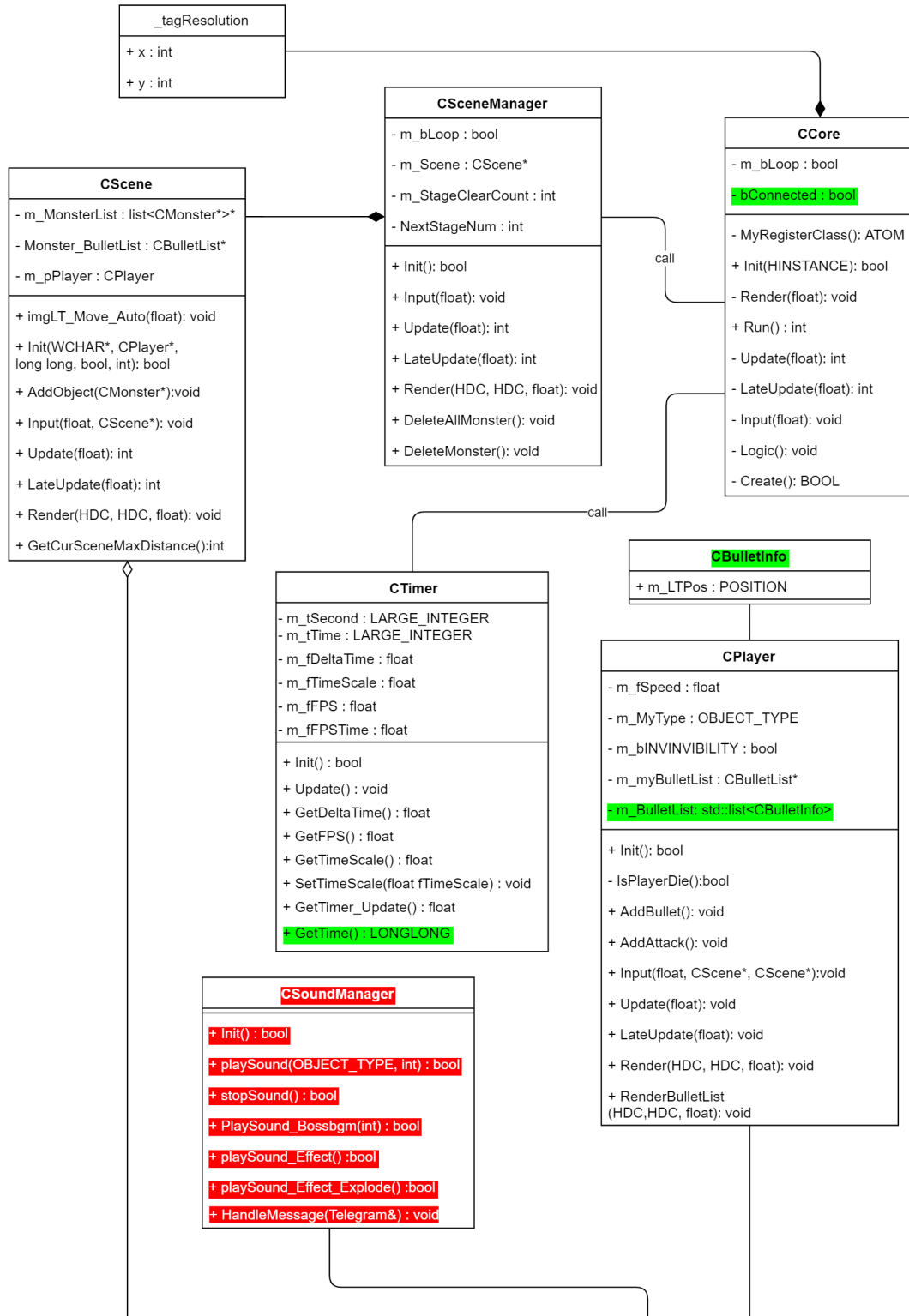
i 번째 원소에는 메시지타입별 메시지의 개수가 저장되어 있다.

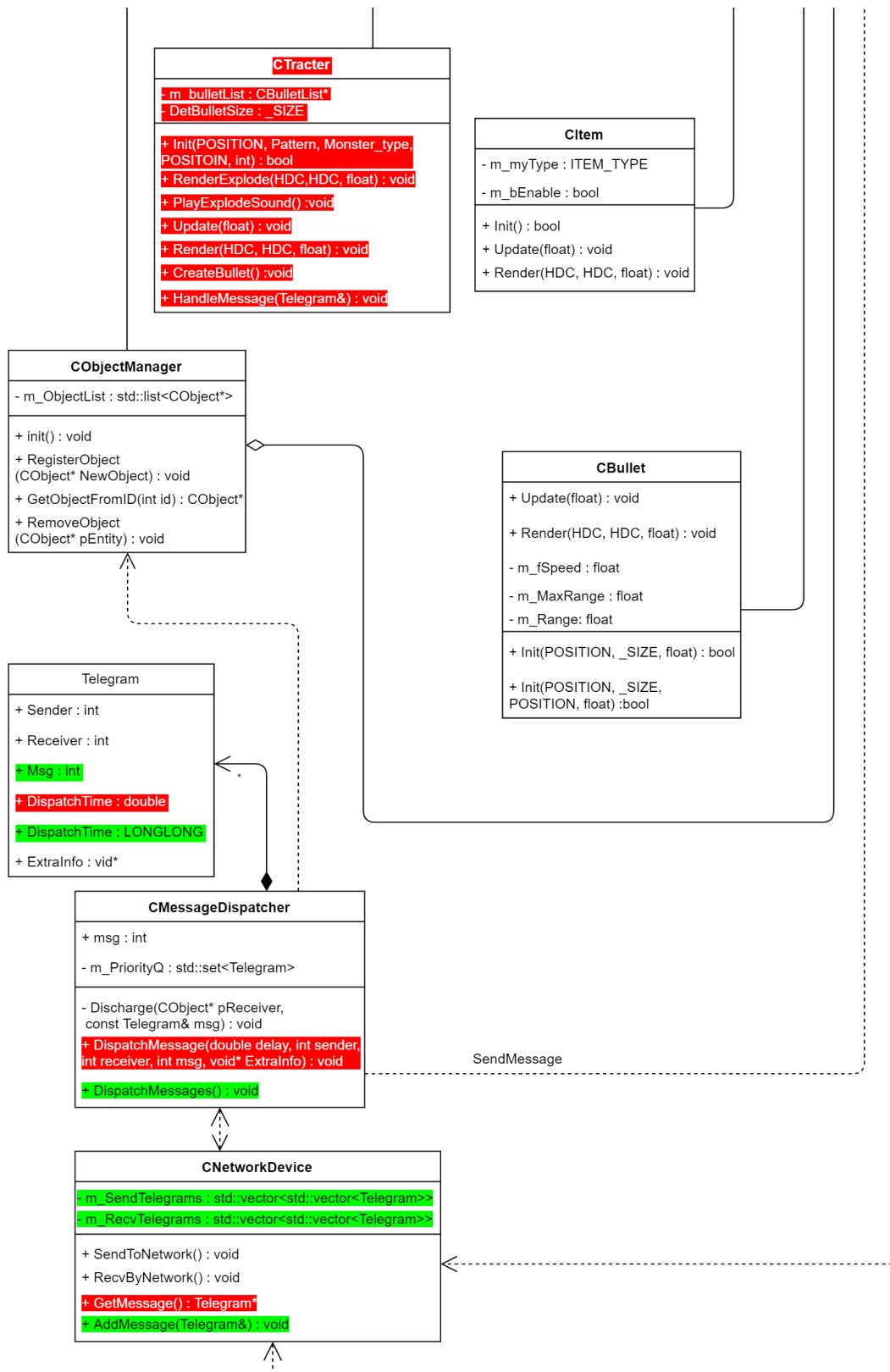
이 루프를 반복하면서 각각의 메시지 타입 별로 미리 계산해 놓은 메시지별 바이트 수만큼 패킷을 읽어 Telegram 객체를 만들어 저장한다.

이후, CMessageDispatcher 에서 CNetworkDevice 로부터 메시지를 넘겨받아 게임 내 객체에게 메시지를 전달한다.

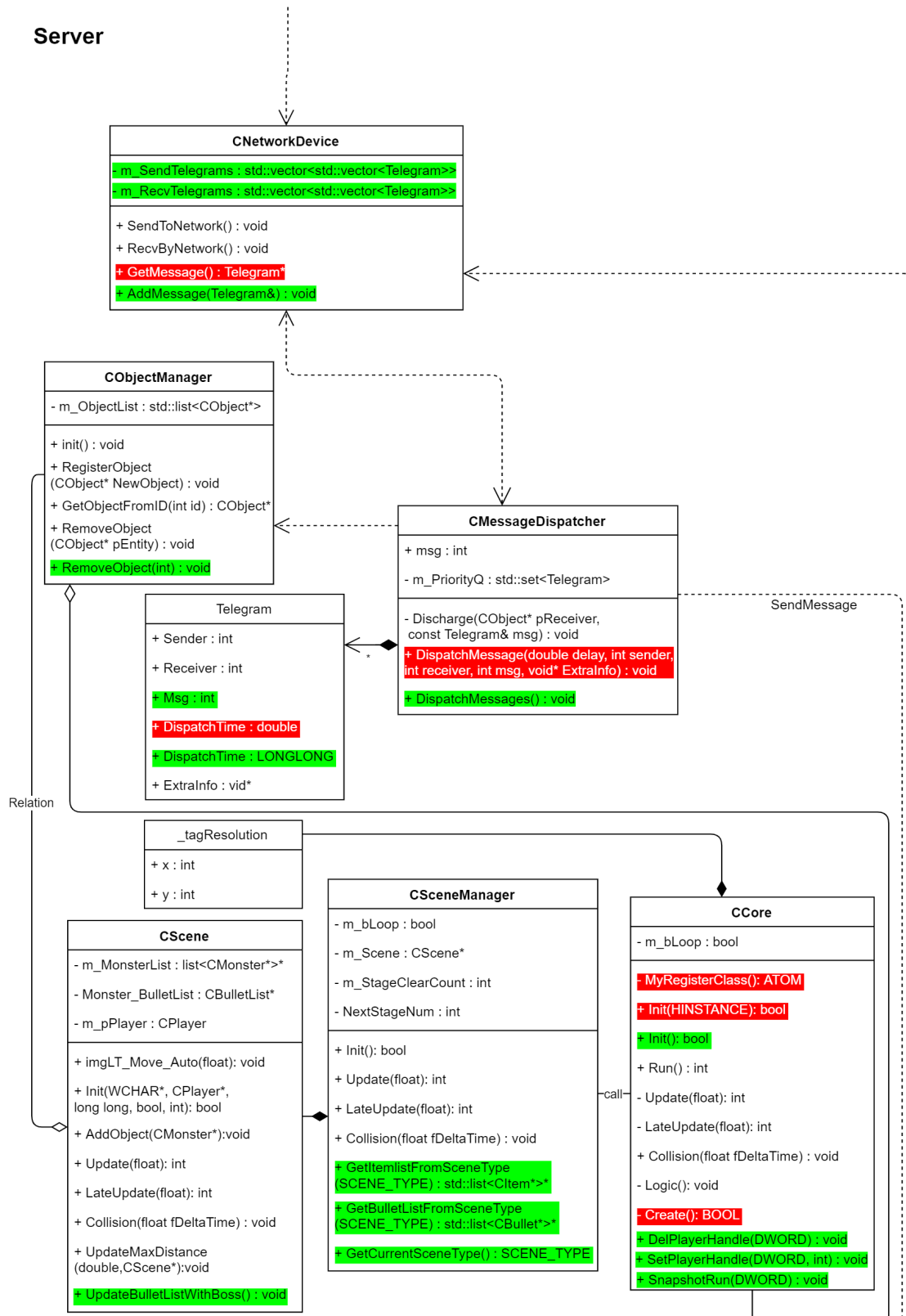
F. UML Class Diagram

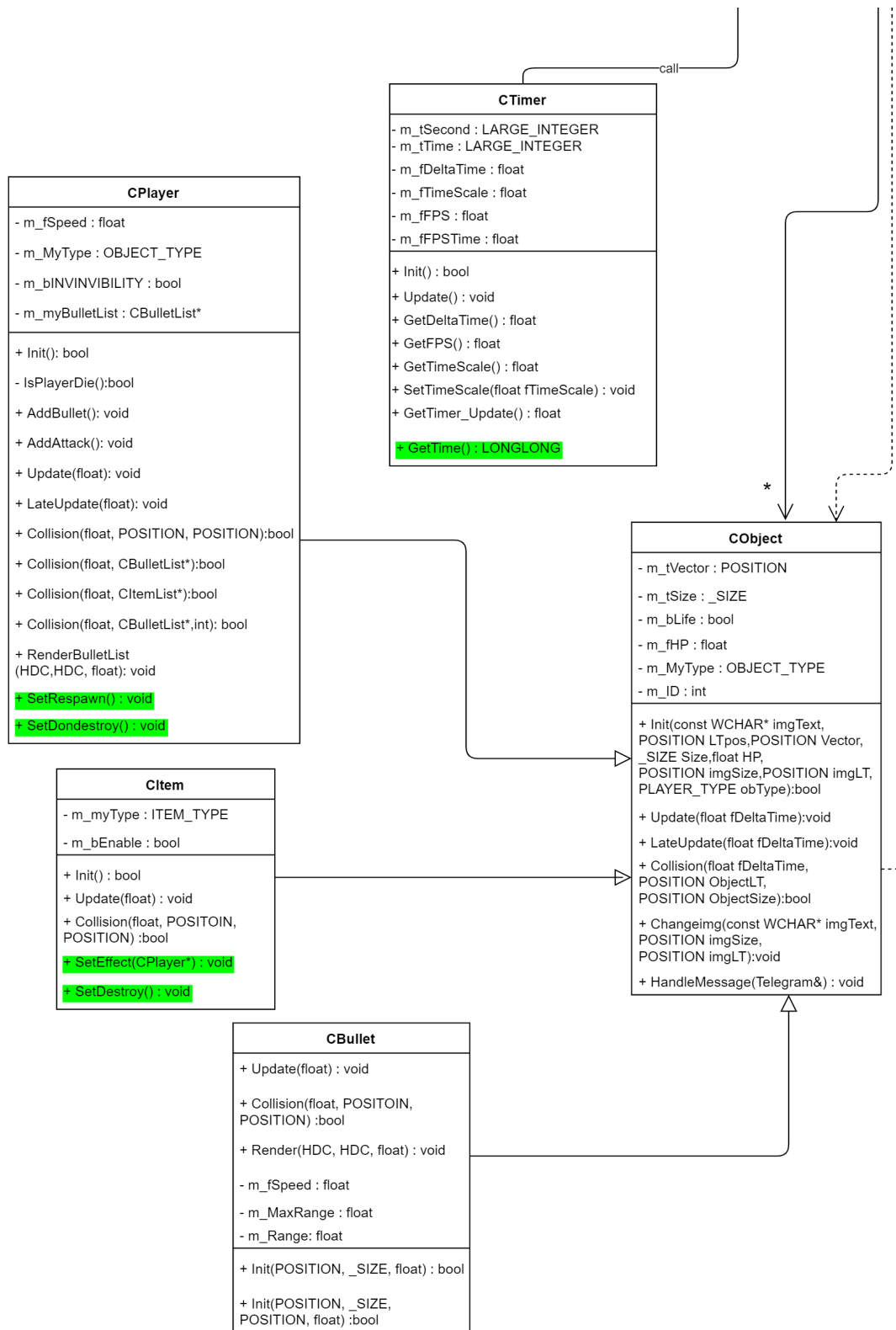
Client





Server







4. 역할분담

A. 서버

구현내용	이지형	최해성	홍명진
서버 프레임워크 구현	✓	✓	✓
씬 전환	✓		
몬스터 생성 및 시뮬레이션	✓		
아이템 생성 및 시뮬레이션		✓	
총알 생성 및 시뮬레이션			✓
충돌처리	✓	✓	
CObjectManager 구현		✓	
CObject 구조 및 상속 구조 변경			✓
CMessageDispatcher 구현	✓	✓	✓
CNetworkDevice 구현	✓	✓	✓
플레이어 사망 및 게임 종료 구현		✓	

B. 클라이언트

구현내용	이지형	최해성	홍명진
프레임워크 수정	✓	✓	✓
플레이어 입력 처리			✓
CObjectManager 구현		✓	
CObject 구조 및 상속 구조 변경			✓
CMessageDispatcher 구현	✓	✓	✓
CNetworkDevice 구현	✓	✓	✓
사운드 재생 방식 변경	✓		

5. 개발환경

- 개발도구: VisualStudio
- 운영체제: Windows 10 x86
- 통신 프로토콜: TCP/IP
- 외부 라이브러리: FMOD

- 관리 툴: GitHub

6. 일정

날짜	이지형	최해성	홍명진
11/5	기존 코드 분석	기존 코드 분석	기존 코드 분석
11/6	기존 코드 분석	기존 코드 분석	기존 코드 분석
11/7			
11/8	서버 CCore 구현	클라 CCore 구현	CObject 상속 구조 변경
11/9	RecvByNetwork()	CObjectManager 구현	SendToNetwork()
11/10			
11/11	Discharge()	DispatchMessage()	AddMessage()
11/12	Monster::HandleMessage()	SceneManager::HandleMessage()	Player::HandleMessage()
11/13			
11/14	Tracter::HandleMessage() Boss::HandleMessage()	Item::HandleMessage() SoundManager::HandleMessage()	Bullet::HandleMessage() 서버 내부 테스트
11/15	Boss::HandleMessage() CScene::HandleMessage() msg_ClientReady 추가	SoundManager::HandleMessage() Client 에서 테스트 메시지 보내고 받게하기	서버 내부 테스트 Client 메인화면 렌더링하게하기
11/16	서버&클라이언트 외부 테스트	클라이언트 내부 테스트	서버 CSceneManager
11/17			
11/18			
11/19	서버 CObject 구현	서버 CScene 구현	서버 CBullet 구현
11/20	서버 CTimer 구현	서버 CItem 구현 ItemList 를 STL 로 전환	BulletList 를 STL 로 전환
11/21	스핀 전환 수정 CardinalSpline()	서버 CItem 버그 수정 클라이언트한테 Item 띄우기	Monster Bullet 구현
11/22	실행 테스트 MonsterPattern 1,2 구현 MonsterPattern 1,2 구현 MonsterPattern 5 구현	서버 몬스터 구현 ObjectID 동기화 실행 테스트	실행 테스트 서버 CBullet 구현 Monster Bullet 구현
11/23	작업물 Merge 후 실행테스트	임계영역 오류 해결 작업물 Merge 후 실행테스트	작업물 Merge 후 실행테스트

11/24	MonsterPattern 1,2 구현	ItemList 를 STL 로 전환	Player 입력처리
11/25	MonsterPattern 3,4 구현	CScene Item 구현	Player Bullet 구현
11/26	MonsterPattern 5,6 구현		
11/27	CCore::SnapshotInit() 구현	플레이어 & 총알 충돌 Player 입력처리	Boss Pattern 1, 2 구현 Player ID 서버-클라 동기화
11/28	아이템 & 플레이어 충돌	플레이어 사망 구현	Boss Pattern 3, 4 구현
11/29	몬스터 & 총알 충돌 Merge 후 오류 해결	서버 GameOver Scene 구현 Player ID 서버-클라 동기화 클라-클라 Player 동기화 Merge 후 오류 해결	Boss Pattern 5 구현 Merge 후 오류 해결
11/30	몬스터 삭제 알고리즘 Client, 통신 스레드 분리	클라이언트 GameOver Scene 구현 Client 네이글 알고리즘 끄기	Player Bullet 생성 구현
12/01	CSoundManager 수정 임계영역 수정	임계영역 수정	임계영역 수정
12/02	보스 생성 몬스터 스폰 수정	Item 생성 수정	Player 총알 발사 개수 구현
12/03	버그 수정 및 동기화	버그 수정 및 동기화 ObjectManager 수정	버그 수정 및 동기화 Object 충돌처리 코드 추가
12/04	버그 수정 및 동기화	버그 수정 및 동기화	버그 수정 및 동기화
12/05	버그 수정 및 동기화 작업 Merge 및 버그 수정	버그 수정 및 동기화 작업 Merge 및 버그 수정	버그 수정 및 동기화 작업 Merge 및 버그 수정
12/06	버그 수정 및 동기화 총알 시뮬레이션 적용 클라이언트 프레임 락	버그 수정 및 동기화 플레이어 아이템 충돌 구현 아이템 Garbage Collect 구현	버그 수정 및 동기화
12/07	버그 수정 및 동기화	버그 수정 및 동기화 아이템 생성 Term 변경 아이템 충돌 버그 수정	버그 수정 및 동기화
12/08	보스 총알 패턴 구현	플레이어 아이템 상호작용 구현 플레이어 총알 충돌 구현	맵 스크롤링 구현
12/09	총알 Garbage Collect 구현	몬스터 State 렌더링 구현	몬스터 State 렌더링 구현
12/10	버그 수정 및 최종 커밋	버그 수정 및 최종 커밋	버그 수정 및 최종 커밋