네트워크 게임 프로그래밍 추진서

2016182024 윤선규

2016182026 이동수

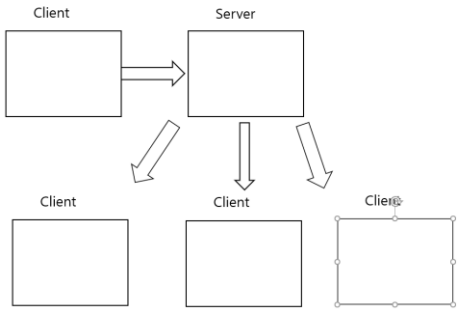
2017180015 서동현

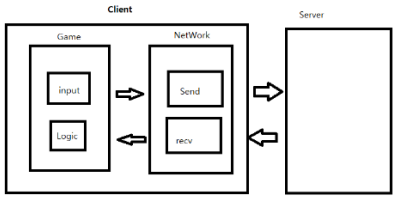
애플리케이션 기획

상대방과 채팅으로 대화를 나누며 상대방과 충돌을 해서 시비를 걸거나.

타격하여 장난을 칠 수 있는 애플리케이션입니다.

High Level





클라이언트의 로직은 서버에 의해서만 변경 가능합니다.

Low Level .

구현될 함수들의 수도코드

```
void CALLBACK send_callback(DWORD errflag, DWORD num_bytes, LPWSAOVERLAPPED send_over, DWORD flag);
// overlapped io 사용, send 콜백함수. 별일 하지 않는다. send_over delete 해준다.

void CALLBACK recv_callback(DWORD errflag, DWORD num_bytes, LPWSAOVERLAPPED recv_over, DWORD flag)
{
        process_packet();
}
// overlapped io 사용, recv 콜백함수.
// process_packet 함수를 호출해서 패킷을 처리한다.


// 콜백함수에 쓸 OVERLAPPED 구조체의 변형.
struct EXP_OVER
{
        WSAOVERLAPPED _wsa_over{};
        Int     _s_id;
        WSABUF  _wsa_buf;
        char    _send_msg[BUFSIZE];
};
```

```
// 서버가 사용하는 클래스. 각각의 SESSION은 클라이언트하나를 의미한다.
class SESSION
{
public:
	void do_recv()
	{
		WSARecv(_socket, &_recv_wasbuf, &_recv_over, recv_callback);
	}

	void do_send(int sender_id, int num_bytes, char* msg)
	{
		EXP_OVER* ex_over = new EXP_OVER{};
		WSASend(_socket, &ex_over->_wsa_buf, &ex_over->_wsa_over, send_callback);
	}

private:
	int		_id;
	SOCKET	_socket;
	WSABUF	_recv_wasbuf;
	char	_recv_buf[MAX_PACKET_SIZE]{};
	WSABUF	_send_wasbuf;
	char	_send_buf[MAX_PACKET_SIZE]{};
	WSAOVERLAPPED _recv_over{};
};

// recv 한 패킷을 처리하는 함수.
// 서버는 패킷 처리 후 모든 session 에 적절한 패킷을 send 한다.
void process_packet(int client_id, unsigned char* packet_start)
{
	unsigned char packet_type = packet_start[1];
	SESSION& client = session[client_id];

	switch (packet_type)
	{
	case CS_PACKET_LOGIN:
	{
		// to do
	}
	break;
	case CS_PACKET_MOVE:
	{
		// to do
	}
	break;
	}
}
```

네트워크 기능
로그인 / 로그아웃 / 채팅 / 타격 / 이동 / 모델변경

패킷
```
template<class T>
struct packet_base
{
	int8 size = sizeof(T);
	PAKCET_TYPE packet_type;
};
#define PACKET(name) struct name : packet_base<name>
```

```
//=============== LOG_IN =================

// => 이 이름으로 로그인 할래

PACKET(cs_try_login)
{
	char name[MAX_NAME_SIZE];
};

// => ok 너의 id는 이거야

PACKET(sc_ok_login)
{
	int8 login_id;
};

// 맵에 누구누구가 어디어디에 있는지 알려주어야 함
// PACKET(sc_new_charator) * numofchractor; to new player

PACKET(sc_new_charator)
{
	int8 login_id;
	char name[MAX_NAME_SIZE];
	float x;
	float y;
};

PACKET(cs_make_map_done)
{

};

// 다른 클라이언트들 한테 새 캐릭터 로그인알림
// PACKET(sc_new_charator); to old players
```

```
//============= LOG_OUT ===========

// => logout (esc 누름)

PACKET(sc_logout)
{
        int8 logout_id;
};

// => recv 의 리턴값이 0 인 경우 또는 PACKET(sc_logout) 패킷이 왔을 경우.

PACKET(cs_logout)
{
        int8 logout_id;
};

// => 맵에서 플레이어 제거.
```

```
//=============== MOVE_INPUT =================

// 이동연잔자들, 비트연산으로 press,unpress 설정
enum class MOVE_DIR : int8
{
        FORWARD = 1 << 0,
        BACK = 1 << 1,
        LEFT = 1 << 2,
        RIGHT = 1 << 3
};

// => input 이 변화했

PACKET(cs_moved)
{
        MOVE_DIR arrow_key;
};

// => 다른플레이어들한테 전송

PACKET(sc_moved)
{
        int8 mover_id;
        MOVE_DIR arrow_key;
};
```

```
//=============== CHATTING =================

// => 채팅 enter

PACKET(cs_chat)
{
        char chat[30];
        char padding = 'WO';
};

// => 다른 플레이어들한테 전송

PACKET(sc_chat)
{
        int8 chatter_id;
        char chat[30];
        char padding = 'WO';
};
```

```
//============= 타격 ===========

// => 플레이어의 타격시도

PACKET(sc_try_attack)
{
        int8 defender_id;
};

// => 서버에서 위치와 접속여부를 토대로 검증

PACKET(cs_attacked)
{
        int8 attacker_id;
        int8 defender_id;
};

// => attacket 의 공격 이펙트 발동
// => defender 의 피격 이펙트 발동
```

```
//=================== MODEL_CHANGE ========================

// => 변경할래

PACKET(cs_model_change)
{
        int8 model_id;
};

// => ok, 모든 플레이어들한테 전송

PACKET(sc_model_change)
{
        int8 id;
        int8 model_id;
};

// => 적용.
```

일정 및 역할 분담

윤선규 :: Overlaped_IO 프레임워크 제작, 클라이언트 구현 // logout , 타격 패킷로직 구현
서동현 :: 로그인, 모델변경 패킷로직 구현
이동수 :: 이동, 채팅 패킷로직 구현

일	월	화	수	목	금	토
미리 윤선규 - 프레임워크제작	11/1 윤선규 - 클라이언트 연결 서동현 - 패킷 전송에 대한 연구	11/2 윤선규 - 클라이언트 연결 서동현 - 로그인 구현	11/3 이동수 - 이동구현 윤선규 - 게임로직 (모델변경, 타격, 로그인/아웃	11/4 윤선규 - 로그아웃 구현	11/5 이동수 - 이동구현 윤선규 - 타격 구현	11/6 윤선규 - 채팅 창 구현
11/7	11/8	11/9 서동현 - 모델변경을 위한 아이디어 고민 - 모델변경 패킷 구현	11/10	11/11	11/12 이동수 - 이동구현	11/13 이동수 - 이동구현 서동현 - 모델변경 패킷 구현
	11/15 이동수 - 채팅구현 서동현 - 모델 변경 패킷 구현	11/16 이동수 - 채팅구현		11/18 서동현 - 모델변경 패킷 구현		11/23 이동수 - 채팅구현