텀 프로젝트 추진 계획서

네트워크 게임 프로그래밍

10 팀

2019184102 박병준

2017180011 박정환

2018180022 방영규

목차

1. 게임 소개

* 게임 개요
* 조작 방법

1. 개발 환경
2. High Level

* Client
* Server

1. Low Level

* Client
* Server

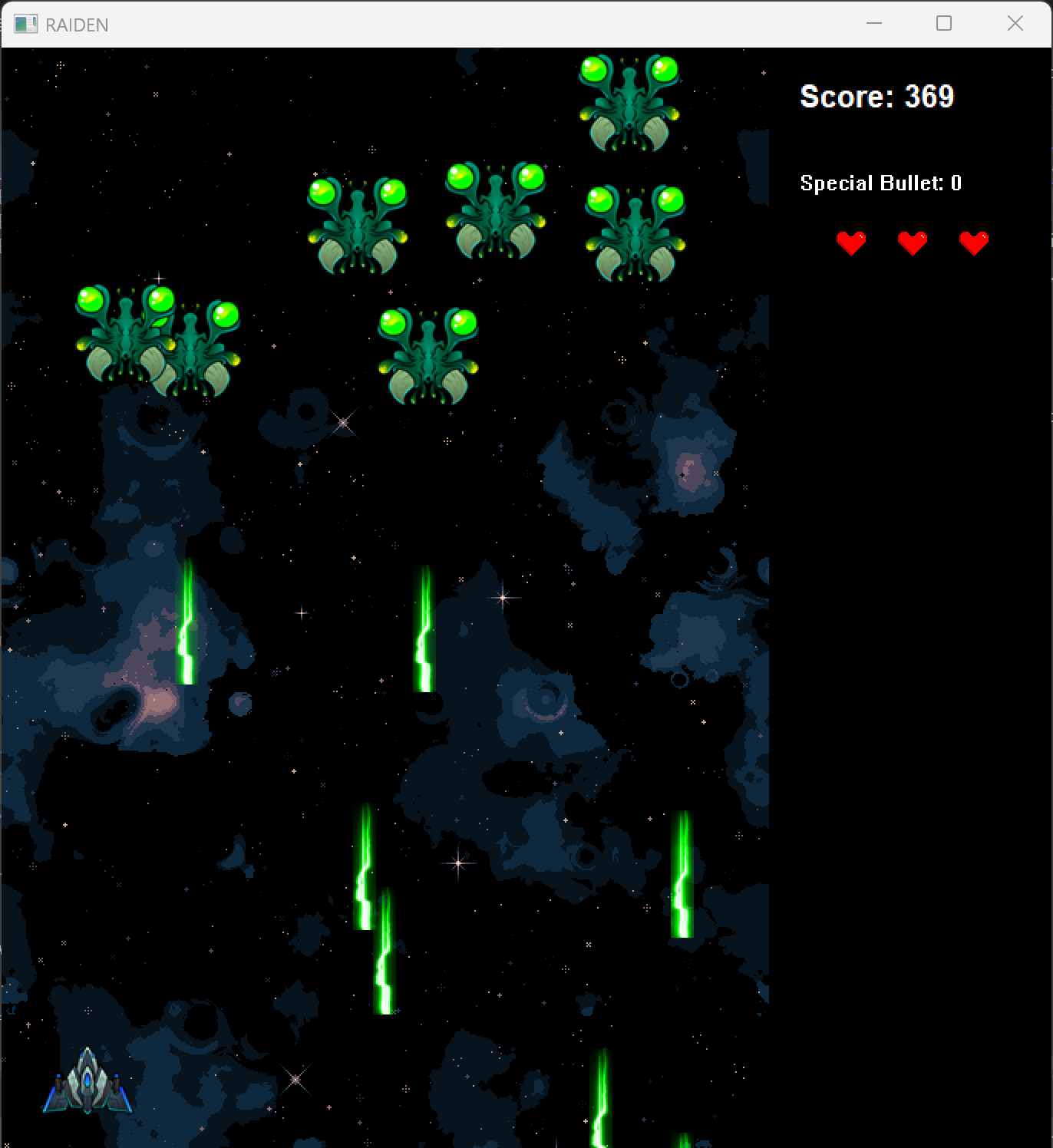
1. 개발 일정

* 역할 분담
* 캘린더

1. 게임 소개

* 게임 개요

클라이언트 프로그램 출처: 박정환 학생의 윈도우 프로그래밍 프로젝트. 라이덴 모작 게임.



네트워크 게임 방향: 2인 멀티플레이 게임

게임을 실행하면 다른 플레이어와 매칭이 될 때까지 대기한다.

매칭이 성공하고 게임이 시작되면 각자 본인의 전투기를 조작하여 게임을 플레이 한다.

플레이어는 스페이스바 키를 입력하여 총알을 발사한다.

무작위 위치에 출현하는 적을 총알을 발사하여 제거한다.

일정 점수에 도달할 때마다 환경이 변화한다. (플레이어 공격 강화, 다른 적 출현, 보스 출현)

플레이어가 적의 공격을 맞고 사망하거나 보스를 처치하면 게임이 종료된다.

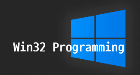
* 조작 방법

상하좌우 방향키 – 이동

스페이스바 – 공격

시프트 + 스페이스바 – 특수 공격

1. 개발 환경

Visual Studio TCP / IP Windows API Github Desktop

1. High Level

* Client

스크린샷, 텍스트, 흑백, 디자인이(가) 표시된 사진

자동 생성된 설명

스크린샷, 텍스트, 흑백, 디자인이(가) 표시된 사진

자동 생성된 설명

클라이언트는 main, Player Thread, Enemy Thread, Recv Thread 총 4개의 Thread로 구성.

main

객체 렌더링과 서버와의 연결을 담당하는 역할.

Enemy Thread

적의 위치 정보를 서버에서 수신하는 역할.

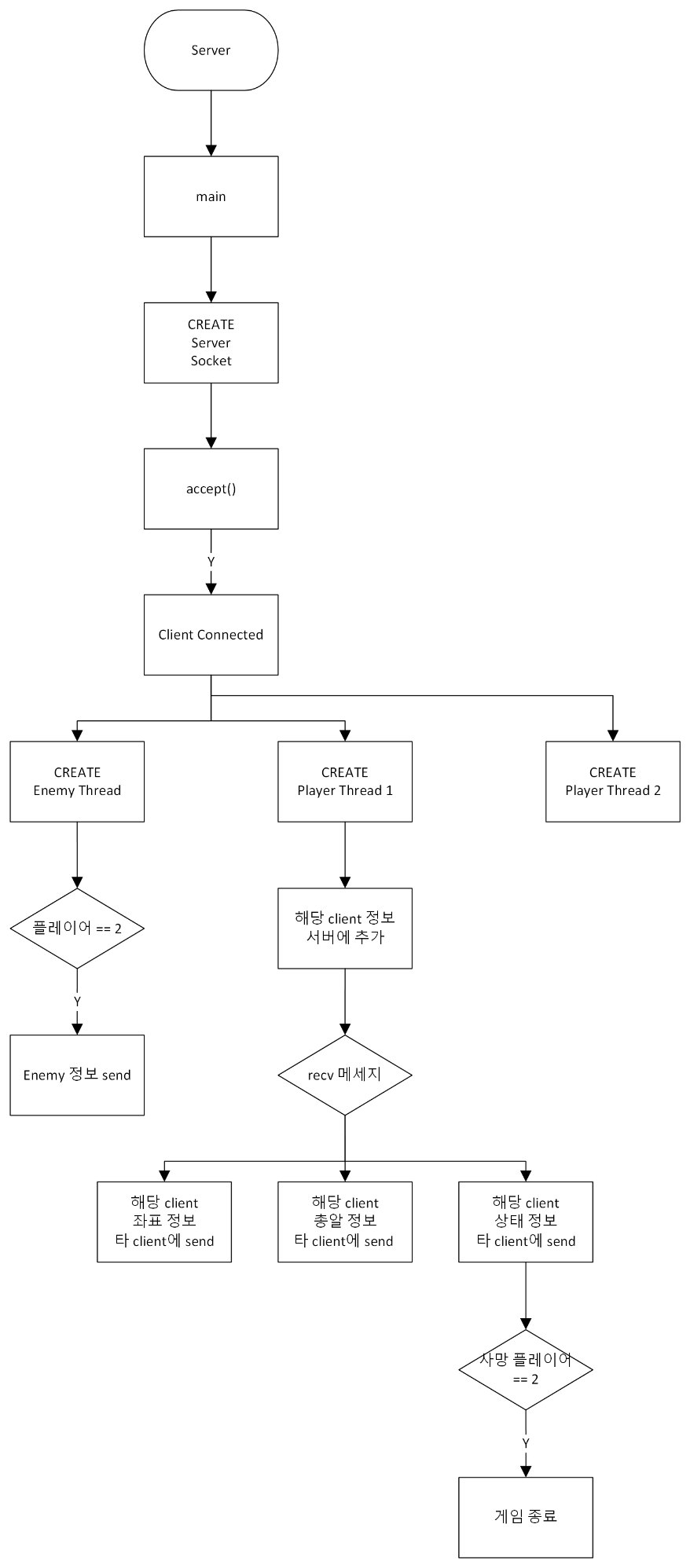
Player Thread

플레이어의 이동, 공격, 사망 정보를 서버에 전달하는 역할.

Recv Thread

적의 정보와 타 클라이언트의 정보를 서버에서 수신하는 역할.

* Server



서버는 main, Enemy Thread, Client\_1 Thread, Client\_2 Thread 총 4개의 Thread로 구성

main

클라이언트와 accept하여 서버와 클라이언트를 연결하는 역할.

2명의 플레이어가 연결될 때까지 대기한다.

Enemy Thread

클라이언트에게 무작위의 위치에 생성되는 적 위치를 전송하는 역할.

Client Thread

클라이언트에 대한 정보를 관리하고 데이터의 갱신이 필요하면 클라이언트에게 데이터를 전송하는 역할.

1. Low Level

* Client
* 전역 변수

unordered\_map<USHORT, CEnemy> g\_enemy

서버로부터 받은 적의 정보를 관리하는데 사용되는 전역 변수.

* 구조체

struct Vec2 {float x; float y; bool dir; };

2D 공간의 벡터 또는 좌표를 표현하기 위한 구조체. 위치와 방향을 나타내는 데 사용

struct CS\_MOVE\_PLAYER {BYTE playerID; Vec2 playerPos; };

플레이어의 이동 정보를 서버와 클라이언트 간에 전달하기 위한 구조체. 플레이어 ID, 현재 위치, 이동 방향을 포함.

struct CS\_PLAYER\_ATTACK {BYTE playerID; Vec2 bulletPos; };

플레이어의 공격 정보를 서버와 클라이언트 간에 전달하기 위한 구조체. 플레이어 ID, 총알의 초기 위치와 방향을 포함.

struct CS\_BULLET\_HIT ENEMY {BYTE playerID; USHORT enemyID; };

플레이어의 총알과 적의 충돌 정보를 저장하는 구조체. 충돌한 총알을 발사한 플레이어의 ID와 적의 ID를 포함.

struct CS\_PLAYER\_DEAD {BYTE playerID; };

플레이어가 사망했을 때 정보를 전달하는 구조체. 사망한 플레이어의 ID를 포함.

* 함수

DWORD WINAPI SendPlayerInfo(LPVOID lpParam)

클라이언트가 서버로 플레이어의 정보를 전송하는 함수. 플레이어의 이동과 공격, 상태 정보를 전송하는 역할.

DWORD WINAPI RecvEnemyInfo(LPVOID lpParam)

서버에서 적의 정보를 수신하여 업데이트하는 함수. 적의 위치나 상태 변화를 수신하여 실시간으로 반영하는 역할.

DWORD WINAPI RecvThread(LPVOID lpParam)

서버로부터 다른 클라이언트의 데이터를 수신하는 함수. 다른 클라이언트의 위치정보, 총알정보, 상태정보를 수신하고 갱신하는 역할.

void PlayerMove(CS\_MOVE\_PLAYER& player)

플레이어의 이동을 처리하는 함수. 플레이어의 새로운 위치와 방향을 설정하고, 이를 서버에 전송하여 이동 상태를 업데이트.

void PlayerAttack()

플레이어의 공격을 처리하는 함수. 새로운 총알을 생성하고 서버에 해당 정보를 전송.

bool IsPlayerDead(const int live)

플레이어의 남은 생명 수를 확인하는 함수. 클라이언트 내에서 생명 수를 관리하며, 생명 수가 0 이하일 경우 true를 반환하고, 플레이어가 사망했음을 서버에 전송.

* Server
* 전역 변수

unordered\_map<USHORT, SESSION> g\_client

클라이언트를 아이디에 따른 소켓으로 통신하기 위해 unordered\_map으로 클라이언트 소켓을 관리해줄 전역변수.

unorderd\_map<USHORT, CEnemy> g\_enemy

SendEnemyInfo에서 생성한 정보를 아이디에 대해서 관리해줄 적 전용 전역변수.

HANDLE client\_1\_done\_event

Client\_1 Thread가 작업을 완료했을 때 Client\_2 Thread가 실행될 수 있도록 신호를 전달하는 이벤트.

HANDLE client\_2\_done\_event

Client\_2 Thread가 작업을 완료했을 때 Client\_1 Thread가 다시 실행될 수 있도록 신호를 전달하는 이벤트.

* 구조체

struct SEND\_PLAYER\_MOVE {BYTE playerID; Vec2 playerPos; };

플레이어 이동 정보를 보내기 위한 패킷.

플레이어 ID, 현재 위치, 이동 방향을 포함.

struct SEND\_PLAYER\_DEAD {BYTE playerID; };

플레이어 사망 시 그 플레이어를 삭제 요청하기 위한 패킷.

사망한 플레이어의 ID를 포함.

struct SEND\_PLAYERPOS {BYTE playerID; Vec2 playerpos; };

플레이어 시작 위치를 보내기 위한 패킷.

플레이어 ID, 현재 위치를 포함.

struct SEND\_ENEMYPOS {BYTE playerID; USHORT enemyID; };

클라이언트에게 적 생성 위치를 보내기 위한 패킷.

적 ID, 플레이어 ID를 포함.

* 함수

DWORD WINAPI SendEnemyPos (LPVOID lpParam)

서버에서 생성된 적의 위치를 연결된 클라이언트에게 전송하는 역할.

DWORD WINAPI ClientThread (LPVOID lpParam)

클라이언트에게 데이터를 수신 받고 다른 클라이언트에게 전송. 플레이어의 이동, 사망, 총알, 시작 위치 정보를 송수신하는 역할.

void SendMovePlayer, RecvMovePlayer(const CS\_MOVE\_PLAYER& packet)

플레이어 이동 정보를 주고받아 다른 플레이어의 이동을 동기화하는 역할.

void SendRemovePlayer, RecvRemovePlayer(USHORT cid)

플레이어의 사망 정보를 주고받아 플레이어의 사망을 동기화하는 역할.

void SendPlayerBullet, RecvPlayerBullet(BYTE playerID, Vec2 bulletPos)

플레이어 총알 정보를 주고받아 플레이어의 총알을 동기화하는 역할.

void SendInitPos(BYTE playerID, Vec2 playerPos)

플레이어들의 시작 위치를 전송하는 역할.

1. 개발 일정

* 역할 분담

|  |  |  |
| --- | --- | --- |
| 이름 | 클라이언트 | 서버 |
| 방영규 | PlayerMove()  IsPlayerDead()  Send\_PlayerInfo() | SendPlayerBullet()  recvPlayerBullet() |
| 박정환 | Recv\_EnermyInfo()  PlayerThread() | Send\_EnemyPos()  recvMovePlayer()  ClientThread() |
| 박병준 | RecvThread()  PlayerAttack() | SendRemovePlayer()  recvRemovePlayer()  SendInitPos()  SendMovePlayer() |

* 캘린더

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 월 | 화 | 수 | 목 | 금 | 토 | 일 |
|  |  | 10/30 | 31 | 11/1 | 2 | 3 |
|  |  |  |  |  |
| 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  |  |  |  | PlayerMove(방영규) | |  |
| PlayerAttack(박병준) | |
| Send\_EnemyPos(박정환) | |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|  | Send\_PlayerInfo(방영규) | |  |  |  |  |
| SendMovePlayer(박병준) | |
| Recv\_EnemyInfo(박정환) | |
| 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| SendPlayerBullet(방영규) | |  |  | IsPlayerDead(방영규) |  |  |
|  | SendRemovePlayer(박병준) | | RecvRemovePlayer(박병준) | |
|  | Recv\_EnemyInfo(박정환) | |  |  |
| 25 | 26 | 27 | 28 | 29 | 30 | 12/1 |
|  | RecvPlayerBullet(방영규) | |  |  |  |  |
|  |  | SendInitPos(박병준) |
|  | RecvMovePlayer(박정환) | |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| PlayerThread(방영규) | |  | 프로그램 테스트 및 버그 체크 | | 버그 수정 |  |
| RecvThread(박병준) | |
| ClientThread(박정환) | |
| 9 | 10 | 11 | 12 |  |  |  |
|  |  |  |  |  |  |  |