**네트워크 게임 프로그래밍**

**2019년 2학기 텀 프로젝트**

**기획서**

이 름 학 번

이경섭 2014180031

김태화 2014182011

**목차**

1. **어플리케이션 기획**
2. 게임 컨셉
3. 조작법
4. 개발 환경
5. **High-Level Design**
6. **Low-Level Design**
7. **팀원 별 역할 분담**
8. **개발 일정**
9. **어플리케이션 기획**
   1. 게임 컨셉

[](https://blog.naver.com/zlemr0622/90187467135)

제목: The Binding of Isaac  
장르: 2인 어드벤처 게임  
플랫폼: PC  
소개: 두명의 플레이어가 같이 보스를 처치하는 게임

2. 게임 조작법

- W, A, S, D키: 캐릭터 이동  
 - 방향키: 눈물(총알) 발사

3. 스테이지 구성

보스 방

- 기본 방: 두명의 클라이언트가 접속할 때까지 기다렸다가 전부 접속하면 게임 시작 가능  
 - 보스 방: 보스 몬스터가 존재하고 전투를 진행

기본 방

1. **High-Level Design**

**Client**

**Server**

WSAStartup

socket

connect

Recv

Update

Render

Send

closesocekt

WSACleanup

WSAStartup

socket

Bind

listen

accept

CreateThread

(Recv Thread)

(

CreateThread

(Send Thread)

closesocekt

WSACleanup

Recv Thread

Recv

SetEvent(SEvent)

Return

Send Thread

WaitForMultipleObject

(SEvent)

Update

Send

Return

1. **Low-Level Design**
   1. Client

|  |  |  |
| --- | --- | --- |
| **타입** | **변수 이름** | **설명** |
| #pragma pack(1)  struct  #pragma pack() | recvData | float posX, posY;  float VelX, VelY  int type, idx\_num |
| #pragma pack(1)  struct  #pragma pack() | sendData | float posX, posY;  float VelX, VelY  int type, idx\_num |
| #pragma pack(1)  struct  #pragma pack() | InitData | float mass  float size  Float coef\_Frict |
| SOCKADDR\_IN | serveraddr | 서버 주소 구조체 |
| SOCKET | sock |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **반환값** | **함수 이름** | **인자 값** | **설명** |
| int | WSAStartup |  | 윈속 초기화 |
| Int | socket | AF\_INET, SOCK\_STREAM , 0 |  |
| int | Connet | sock, (SOCKADDR\*)&serveraddr, sizeof(serveraddr) | server와 연결 |
| Int | Recv | sock,(char\*)&recvData, sizeof(recvData) | 데이터 수신 |
|  | Update |  | 입력 키에 의한Player 위치 업데이트 |
|  | Render |  | recvData를 기반으로 화면에 그림 |
| Int | Send | sock, (char\*)&sendData,  sizeof(sendData), 0 | sendData전송 |
| int | Closesocekt | sock | 소켓 닫기 |
| int | WSACleanup |  | 윈속 반환 |

* 1. Server

|  |  |  |
| --- | --- | --- |
| **타입** | **변수 이름** | **설명** |
| #pragma pack(1)  struct  #pragma pack() | recvData | float posX, posY;  float VelX, VelY  int type, idx\_num |
| #pragma pack(1)  struct  #pragma pack() | sendData | float posX, posY;  float VelX, VelY  int type, idx\_num |
| #pragma pack(1)  struct  #pragma pack() | InitData | float mass  float size  Float coef\_Frict |
| **struct** | objects | int type //객체의 종류  int idx\_num //객체의 인덱스  float posX, posY;//위치  float sizeX, sizeY;//크기  float velX, velY;//속도  float mass;//질량  float coefFriction;//마찰계수  float elapsed\_time;//경과시간  int HP //객체의 체력  int State// 객체의 상태 |
| HANDLE | SEvent | Send Thread 동작 제어 이벤트 |

* **Main Thread**

|  |  |  |
| --- | --- | --- |
| **타입** | **변수 이름** | **설명** |
| SOCKADDR\_IN | clientaddr | 클라이언트 주소 구조체 |
| SOCKADDR\_IN | serveraddr | 서버 주소 구조체 |
| SOCKET | listen\_sock |  |
| SOCEKT | client\_sock |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **반환 값** | **함수 이름** | **인자 값** | **설명** |
| int | WSAStartup() |  | 윈속 초기화 |
| int | socket() | AF\_INET, SOCK\_STREAM, 0 | TCP소켓 생성 |
| int | bind() | listen\_sock,  (SOCKADDR\*)&serveraddr, sizeof(serveraddr) | listen\_sock에  지역 IP주소,  지역 포트번호 설정 |
| int | listen() | listen\_sock, SOMAXCONN | listen\_sock의 포트 상태를  LISTENING으로 바꾼다 |
| SOCKET | accept() | listen\_sock,  (SOCKADDR\*)&clientaddr | client\_sock 생성 |
| HANDLE | CreateThread() | NULL, 0, RecvThread,  (LPVOID)&client\_sock, 0,  NULL | Recv Thread 생성  client\_sock전달 |
| HANDLE | CreateThread() | NULL, 0, SendThread,  (LPVOID)&client\_sock, 0,  NULL | Recv Thread 생성  client\_sock전달 |
| int | closesocket() | listen\_sock | 소켓 닫기 |
| int | WSACleanup() |  | 윈속 반환 |

* **SendThread**

|  |  |  |
| --- | --- | --- |
| **타입** | **변수 이름** | **설명** |
| Struct | object | int type //객체의 종류  int idx\_num//객체의 인덱스  float posX, posY;//위치  float sizeX, sizeY;//크기  float velX, velY;//속도  float mass;//질량  float coefFriction;//마찰계수  float elapsed\_time;//경과시간  int HP //객체의 체력  int State// 객체의 상태 |
| #pragma pack(1)  struct  #pragma pack() | sendData | float posX, posY;  float VelX, VelY  int type, idx\_num |
| SOCKET | Client\_sock |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **반환 값** | **함수 이름** | **인자 값** | **설명** |
| DWORD | WaitForMultipleObject | 4, SEvent, TRUE, INFINITE | SEvent가 신호상태가 될 때 까지 대기 |
| void | Update |  | RecvData를 이용한 위치 업데이트 |
| Int | Send | client\_sock, (char\*)&sendData,  sizeof(sendData), 0 | SendData 전송 |

* **RecvThread**

|  |  |  |
| --- | --- | --- |
| **타입** | **변수 이름** | **설명** |
| Struct | object | int type //객체의 종류  int idx\_num//객체의 인덱스  float posX, posY;//위치  float sizeX, sizeY;//크기  float velX, velY;//속도  float mass;//질량  float coefFriction;//마찰계수  float elapsed\_time;//경과시간  int HP //객체의 체력  int State// 객체의 상태 |
| #pragma pack(1)  struct  #pragma pack() | recvData | float posX, posY;  float VelX, VelY  int type, idx\_num |
| SOCKET | Client\_sock |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **반환 값** | **함수 이름** | **인자 값** | **설명** |
| int | recv() | client\_sock,(char\*)&recvData, sizeof(recvData) | RecvData 수신 |
| Bool | SetEvnet | SEvent | SEvent을 신호상태를 만듬 |

1. **팀원 별 역할 분담**

**Client**

**Server**

WSAStartup

socket

connect

Recv

Update

Render

Send

closesocekt

WSACleanup

WSAStartup

socket

Bind

listen

accept

CreateThread

(Recv Thread)

(

CreateThread

(Send Thread)

closesocekt

WSACleanup

Recv Thread

Recv

SetEvent(SEvent)

Return

Send Thread

WaitForMultipleObject

(SEvent)

Update

Send

Return

|  |  |  |
| --- | --- | --- |
|  |  | |
|  | |  |

**김태화: Client & Send Thread**

**이경섭: Server & Recv Thread**

1. **개발 일정**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **월** | **화** | **수** | **목** | **금** | **토** | **일** |
| **10/28** | **10/29** | **10/30** | **10/31** | **11/1** | **11/2** | **11/3** |
| 추진 계획서 준비 | | | |  | 클라이언트 스테이지 구성 기능 추가  **이경섭**  **김태화** | 클라이언트 보강  **이경섭**  **김태화** |
| **11/4** | **11/5** | **11/6** | **11/7** | **11/8** | **11/9** | **11/10** |
| Server Main Thread구현  **이경섭** | Client에서  주고받을 SendData, RecvData 구현  **김태화** | Server Main Thread 구현  **이경섭** |  | Client에서 초기화를 위해 전송할 InitData 구현  **김태화** | Server Main Thread 구현  **이경섭** | 클라이언트 추가 보강  **이경섭**  **김태화** |
| **11/11** | **11/12** | **11/13** | **11/14** | **11/15** | **11/16** | **11/17** |
|  | Send Thread 구현  **김태화** |  |  | Send Thread 구현  **김태화** | Upadate 구현  **김태화** | Send Thread 구현  **김태화** |
| **11/18** | **11/19** | **11/20** | **11/21** | **11/22** | **11/23** | **11/24** |
|  |  |  |  | 조율 및 점검 | 조율 및 점검 |  |
| **11/25** | **11/26** | **11/27** | **11/28** | **11/29** | **11/30** | **12/1** |
|  | **김태화** |  |  | **김태화** | **김태화** | **김태화** |
| **12/2** | **12/3** | **12/4** | **12/5** | **12/6** | **12/7** | **12/8** |
|  |  |  |  |  | 최종 점검 및 조율 | 최종 점검 및 조율 |
| **12/9** | **12/10** | **12/11** | **12/12** |  | | |
|  |  | 최종 점검 및 조율 | **최종 발표** |