네트워크 게임 프로그래밍

Term Project

2019180020 설찬형

2019180004 구윤성

**1. 게임 기획**

구윤성 학우가 윈도우 프로그래밍에서 c++로 만든 프로젝트, 쿠키런 모작을

이용할 예정입니다.

**2. 게임 설명**

쿠키런 모작으로, 쿠키가 달리면서 코인을 먹을 시에 점수를 얻도록 설정하였습니다. 이를

네트워크 통신을 이용해 2인이 게임에 참여할 수 있도록 바꿀 예정입니다. 실행 방법은

스페이스바를 이용해 점프를 하는 방식입니다. 시작할 때, 5개의 목숨으로 시작을 하며,

낙사의 경우에는 목숨이 다 사라집니다. 오브젝트는 4가지가 있으며, 코인을 먹을 시에는

게임이 끝나고 코인이 점수로 환산되며, 물약을 먹을 땐 일정 시간 동안 캐릭터의

크기를 키웁니다. 바나나는 일정 시간 동안 캐릭터의 속도를 느리게 해주며, 적의 경우,

적의 상단을 점프를 이용해 밟게 될 경우, 적에게 피해를 주며 적이 사라지고, 측면에

피격될 경우, 목숨을 하나씩 잃습니다. 이를 네트워크 통신을 이용해 상대의 화면을 보면서

상대의 목숨과 코인을 확인하며 자신의 플레이를 안정적이거나 모험적으로 플레이를

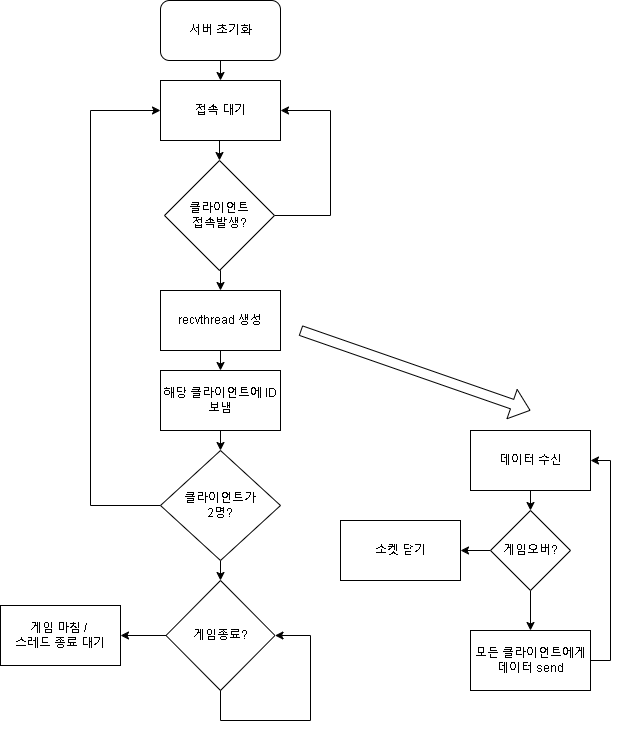
할 수 있도록 하고자 합니다.



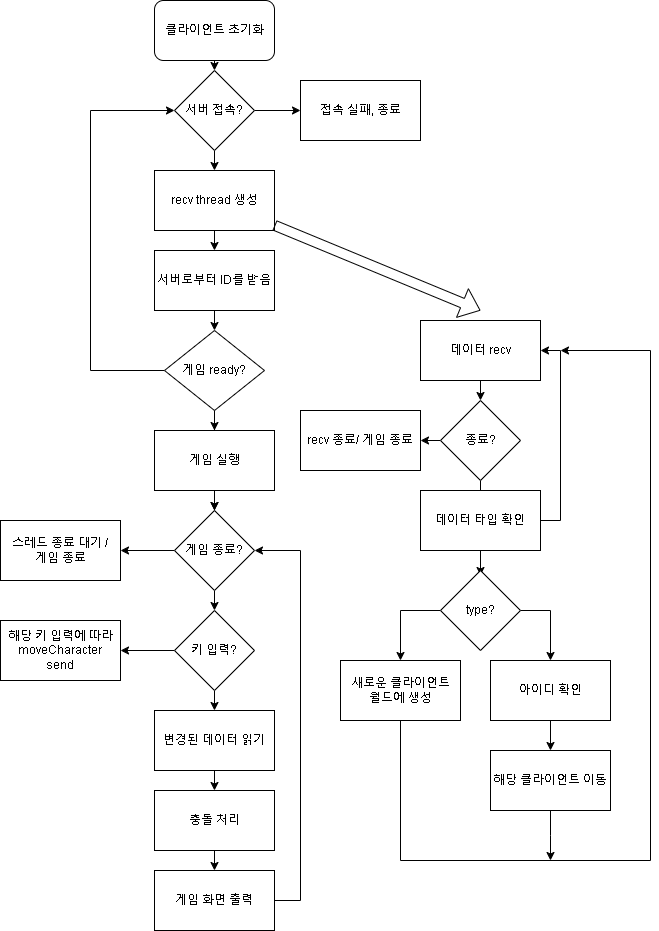
현재 인게임 스크린샷 2인 플레이 시 인게임 스크린샷

**3. High-Level Design**

**Server**

****

**Client**

****

**4. Low-Level Design**

**서버**

1. 패킷

struct LOGIN\_PACKET:

char size 패킷 크기

char type 패킷 타입 (접속 : 0 / 이동 : 1)

Vec2 Pos 플레이어 위치

Vec2 Scale 플레이어 크기  
  
struct MOVE\_PACKET:

char size 패킷 크기

char type 패킷 타입 (접속 : 0 / 이동 : 1)

int id 플레이어 번호

int key 어떤 입력이 발생했는지

1. 전역 변수

#define SERVERPORT

서버포트

#define BUFSIZE

버퍼 사이즈

SOCKET server\_socket

서버 local 소켓

vector <SOCKET> Clients

서버에서 관리하는 클라이언트들의 정보를 담는 공유 데이터

Bool IsReady

2명의 클라이언트 접속을 확인하는 변수

True : 게임 준비

False : 계속 새 클라이언트를 받음

Bool IsRunning

게임 종료 여부를 확인하는 변수

True : 게임 중

False : 게임 종료

1. 스레드 함수

DWORD WINAPI WorkerThread(LPVOID arg)

지역 변수:

- SOCKET client\_socket recv를 받을 클라이언트의 소켓

- sockaddr\_in clientaddr 클라이언트 주소 구조체

- char addr[INET\_ADDSTRLEN] 클라이언트 주소 문자열

- PACKET packet recv에 사용할 버퍼

- int retval recv함수의 반환 값

1. while(true)문을 반복하면서 recv함수를 호출한다
2. retval의 값에 따라 접속을 종료할 지 패킷을 수신할 지 결정한다.
3. retval가 true이면, packet의 타입을 먼저 확인한다
4. 접속한 모든 Clients에게 그대로 패킷을 전달한다
5. retval가 false이면 해당 클라이언트 소켓을 닫는다
6. 기타 함수

bool InitServer()

서버를 초기화하는 함수 (bind / listen 을 여기서 구현한다)

지역변수:

WSADATA was winsock 구조체

sockaddr\_in serveraddr 서버 주소 구조체

int server\_retval bind / listen 반환 값

main()

1. InitServer()함수를 실행, 서버를 초기화 한다
2. While(IsReady)문으로 모든 클라이언트가 접속할 때 까지 기다린다
3. While문 안에서는 SOCKET타입 소켓 client\_socket과 socketaddr\_in 타입 client\_addr를 만든다
4. 새로 생성한 client\_socket에 accept()함수의 결과를 받는다
5. Client\_socket의 값이 INVALID\_SOCKET이 아니면
6. Client\_socket을 Clients에 emplace\_back()하고 새 WorkerThread를 생성한다
7. 새 패킷 LOGIN\_PACKET을 만들고 id를 넣어서 클라이언트에게 보내준다
8. Id는 vector의 크기 값이다. 접속한 순서대로 id를 받는다
9. WorkerThread의 인자로 client\_socket을 받는다
10. INVALID\_SOCKET이면 client\_socket을 닫는다

**클라이언트**

1. 전역 변수

#define SERVERPORT

서버포트

#define BUFSIZE

버퍼 사이즈

SOCKET client\_socket

클라이언트 local 소켓

HANDLE hRecvThread

Recv스레드 핸들  
 int ID

클라이언트 고유 아이디(서버가 보내줌)

Bool IsReady

2명의 클라이언트 접속을 확인하는 변수

True : 게임 준비

False : 계속 새 클라이언트를 받음

Bool IsRunning

게임 종료 여부를 확인하는 변수

True : 게임 중

False : 게임 종료

1. 스레드 함수

DWORD WINAPI RecvThread(LPVOID arg)

지역 변수:

- SOCKET client\_socket recv를 받을 클라이언트의 소켓

- PACKET packet recv에 사용할 버퍼

- int retval recv함수의 반환 값

1. while(true)문을 반복하면서 recv함수를 호출한다
2. retval의 값에 따라 접속을 종료할 지 패킷을 수신할 지 결정한다.
3. retval가 true이면, packet에 담긴 정보를 바탕으로 데이터를 변경한다.
4. 이 때, 변경을 완료할 때까지 변경할 데이터에 lock을 걸어둔다
5. 기타 함수

moveCharacter(int key, SOCKET socket)

1. 새로운 MOVE\_PACKET 패킷을 하나 생성한다.
2. 패킷 사이즈와 타입을 넣는다. (type은 1로 한다)
3. 사용자 입력 (key)을 새 패킷의 key에 넣는다
4. 본인 아이디 ID를 새 패킷의 id에 넣는다
5. 서버로 MOVE\_PACKET을 send()한다.

**5. 팀원별 역할 분담**

구윤성: 서버 메인 스레드 설계, 서버 상태 메시지 수신 스레드 설계

설찬형: 클라이언트 메인 스레드 설계, 클라이언트 상태 메시지 수신 스레드 설계,

서버 패킷 처리 설계

**6. 개발 환경**

언어: C++

개발 도구: Visual Studio 2022

운영 체제: Windows

**7. 개발 일정**

일정 완료, 일정 밀림 및 삭제, 일정 밀림 후 재편성, 일정 새로 생성

설찬형 (클라이언트 메인 스레드 설계, 클라이언트 상태 메시지 수신 스레드 설계,   
서버 패킷 설계)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3 | 4 | 5 | 6 | 7  기획서  재검토 | 8 | 9  서버 패킷 설계  (상태 메시지 송수신 처리) |
| 10  1주차 피드백 및 부족한 진도 보강 | 11 | 12  서버 패킷 처리 관련 로직 구현 | 13 | 14  서버 패킷 처리 관련 로직 구현 | 15  서버 패킷 처리 관련 로직 구현 | 16  기획서 전면 재검토  쿠키런에서  크아로  게임 변경 |
| 17  2주차 피드백  및 부족한  진도 보강  게임 변경 완료 및 기획서 재검토 | 18  크아 클라이언트  수정  (완료) | 19  Move  Character 함수  구현  ->  Change Character  State 구현  (11/27) | 20  크아 클라이언트  수정  (완료) | 21  Move  Character  함수 구현  1p, 2p를  구분하는  함수 구현  ->  Change Character  State 구현  (11/27) | 22  Move  Character 함수  구현  Player Update  함수 구현  ->  Change Character  State 구현  (11/27) | 23 |
| 24  3주차 피드백 및 부족한 진도 보강 | 25  크아 클라이언트  수정  (완료) | 26  server  상태 메시지  구현  ->  Packet 로그  구현 | 27  Change Character  State 구현 | 28  WorkerThread  함수 구현 패킷 타입  처리  (오류로 인해  완료 못함  11/29일 예정) | 29  WorkerThread  새 패킷 생성  구현  (앞의 WorkerThread 함수 구현 일정 밀림에 따른 11/30일)  피드백 후  다시 쿠키런  클라이언트로 | 30  WorkerThread  새 패킷 생성  구현 |
| 12/1  4주차 피드백 및 부족한 진도 보강 | 2 | 3  WorkerThread 스레드 switch(type)  구현 | 4 | 5  Server 게임 종료 관련 구현 설계  -> 랜더링 오류로 인해 무기한 연기 | 6 | 7 |
| 8  5주차 피드백  및 부족한 진도 보강 | 9 | 10  디버깅 및  버그수정 | 11  디버깅 및  버그수정 | 12  디버깅 및  버그수정 | 13 |  |

구윤성 (서버 메인 스레드 설계, 서버 상태 메시지 수신 스레드 설계)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3 | 4 | 5 | 6 | 7  기획서  재검토 | 8 | 9 |
| 10  1주차 피드백 및 부족한 진도 보강 | 11 | 12  InitServer  생성 | 13 | 14  클라이언트  2인용으로  재구성  오류로 인해 일정 밀림 | 15  클라이언트  2인용으로  재구성  오류로 인해 일정 밀림 | 16  기획서 전면 재검토  쿠키런에서  크아로  게임 변경 |
| 17  2주차 피드백  및 부족한  진도 보강  게임 변경 완료 및 기획서 재검토 | 18  서버  Main스레드  생성 | 19 | 20  서버 패킷 수신 구현 확인 / 필요시 보완 | 21 | 22  클라이언트  RecvThread  스레드 함수 생성  -> 일정 변경  상태 메시지 수신 구현과 일정 변경함. | 23  (New)  상태 메시지  수신 구현  RecvThread |
| 24  3주차 피드백 및 부족한 진도 보강 | 25  (New)  상태 메시지  수신 구현  RecvThread | 26  서버 패킷 송신 구현  (1p, 2p)  -> 12/4일로  일정 미룸 | 27  서버 패킷 recvThread 처리 구현  -> 11/29일에 있던 할 일을  27일에 하고자 했으나 오류로 인해 완료 못함. | 28 | 29  서버 패킷 recvThread 처리 구현  피드백 후  다시 쿠키런  클라이언트로 | 30 |
| 12/1  4주차 피드백 및 부족한 진도 보강 | 2  클라이언트  RecvThread  스레드 함수  생성 | 3  상태 메시지  수신 구현  RecvThread | 4  서버 패킷 송신 구현  (1p, 2p) | 5  상태 메시지  수신 구현  RecvThread | 6 | 7  게임 종료 관련 함수  구현  -> 랜더링 오류로 인해  일정 무기한 연기 |
| 8  5주차 피드백  및 부족한 진도 보강 | 9 | 10  디버깅 및  버그수정 | 11  디버깅 및  버그수정 | 12  디버깅 및  버그수정 | 13 |  |