네트워크 게임 프로그래밍

팀 프로젝트 추진 계획서

상징, 원, 로고, 폰트이(가) 표시된 사진

자동 생성된 설명

2020182029이동재

2021184040박병일

2022182018 서가은

목차

1. 애플리케이션 기획

1. 게임 이름
2. 게임 특징
3. 게임 조작키
4. 게임 출처

2. High Level 디자인

1. Title Flow Chart
2. Ingame Flow Chart

3. Low Level 디자인

4. 팀원 별 역할분담

5. 개발환경

6. 개발일정

1. 애플리케이션 기획

1. 게임 이름

< 피코파코 >

텍스트, 스크린샷, 폰트, 그래픽이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 멀티미디어 소프트웨어, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 직사각형, 디자인이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 직사각형, 그래픽이(가) 표시된 사진

자동 생성된 설명

[ 타이틀 화면, 인게임 모습, 스테이지변경 화면, 엔딩 화면 ]

1. 게임 특징

‘피코파크’ 게임을 모작한 2인용 퍼즐 게임이다. 윈도우 API 프로그래밍을 이용하였으며, 협동하여 퍼즐을 풀고 열쇠를 획득하여 방을 탈출하는 것이 목표이다. 스테이지를 모두 클리어 시, 스테이지별 게임 클리어 시간을 출력해주며 게임이 끝난다.

1. 게임 조작키

* WASD : 이동키
* 스페이스 : 총알 발사
* Ctrl : 게임 재시작 투표

1. 게임 출처

2022182018 서가은, 2022182025 윤정민 윈도우 프로그래밍 텀 프로젝트

2. High Level 디자인

1. Title Flow Chart

텍스트, 스크린샷, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

1. Ingame Flow Chart

수정전

스크린샷, 텍스트, 도표, 디자인이(가) 표시된 사진

자동 생성된 설명

<Update Thread 설명>

1. RecvQueue 공유자원 임계영역 권한을 획득한다.
2. RecvQueue 공유자원을 TempQueue에 값을 복사한다.
3. RecvQueue 공유자원 임계영역 권한을 반환한 뒤, TempQueue에 있는 값을 기반으로 Update를 처리한다.
4. SendQueue 공유자원 임계영역 권한을 획득한다.
5. SendQueue에 보낼 데이터를 적재해준다.
6. SendQueue 공유자원 임계영역 권한을 반환한다.

Server의 Update Thread와 Communication Thread는 RecvQueue와 SendQueue를 공유한다.

수정후

텍스트, 스크린샷, 소프트웨어, 디자인이(가) 표시된 사진

자동 생성된 설명

<개발하면서 수정 내용>

std::queue<타입> RecvQueue;

std::queue<타입> SendQueue;

타입에 void\*를 넣으면 클라이언트가 보낸 포인터 주소와 서버가 받아서 따라간 포인터 주소와는 다른 영역이기에 void\*형식을 사용할 수 없다.

패킷들을 전부 부모 패킷에게 상속받는 형태로 만들고 타입을 부모 패킷의 형태로 하더라도 상속받은 데이터가 보내지는 게 아니라 부모 데이터만 보내져서 상속을 이용할 수 없다.

* 플레이어 별 버퍼를 2개 만들고, 입력에 따른 데이터가 버퍼에 차곡차곡 쌓여서 큐처럼 선입선출로 하여 순서제어를 한다.

텍스트, 폰트, 스크린샷, 타이포그래피이(가) 표시된 사진

자동 생성된 설명

데이터를 버퍼에 적재하기 전에 Opcode를 먼저 버퍼에 작성하고, 그 후에 데이터를 버퍼에 작성한다. 순서대로 패킷에 대한 내용을 적재한 이후에 사이즈를 먼저 보낸 후 데이터를 보낸다.



받을 때에도 사이즈를 먼저 받고 따로 저장한 이후 데이터를 플레이어 버퍼에 받는다.



그 후 opcode를 먼저 읽고 읽은 값에 따라서 데이터를 읽는다. 텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

이를 통해 큐와 비슷한 형식으로 동작시킬 수 있으며, 기존의 패킷에서 packetType 멤버변수를 없애도 동작이 가능하여 다음과 같이 수정해주었다.

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 포스트잇 노트, 사각형이(가) 표시된 사진

자동 생성된 설명

3. Low Level 디자인

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

bool SendPacket(SOCKET& sock, const void\* packetData, int packetSize) // 패킷 전송 함수 (클라이언트에서 호출)

bool ReceivePacket(SOCKET& sock, void\* buffer, int bufferSize) // 패킷 수신 함수 (서버에서 호출)

// 각 패킷을 송신하는 함수 정의

bool SendInputPacket(SOCKET& sock, const Input\_Packet& inputPacket) // 입력 패킷

bool SendPlayerIDRequest(SOCKET& sock) // 플레이어 ID 요청

bool SendGTimePacket(SOCKET& sock, const GTime\_Packet& gTimePacket) // 서버 시간

bool SendPlayerIDResponse(SOCKET& sock, const PlayerIDResponsePacket& responsePacket) // 플레이어 ID 전송

bool SendObjectInfoPacket(SOCKET& sock, const ObjectInfo\_Packet& objectInfoPacket) // 오브젝트 정보

bool SendEndGamePacket(SOCKET& sock, const EndGame\_Packet& endGamePacket) // 게임 종료 신호 패킷를 받는 함수

void HandleReceivedPacket(SOCKET& sock, uint8\_t\* buffer) // 수신된 패킷을 처리하는 함수

bool SetSocketTimeout(SOCKET& sock, int timeoutInSeconds); // 타임아웃 설정 함수

bool HandleNetworkError(int errorCode);// 오류 처리

bool RetrySend(SOCKET& sock, const void\* packetData, int packetSize, int retryCount); // 재시도 함수

추가

스레드 함수

DWORD WINAPI CommunicationThreadFunc(LPVOID lpParam) // 서버, 클라이언트

DWORD WINAPI UpdateThreadFunc(LPVOID lpParam) // 서버

클라이언트 함수

bool Initialize(); // Winsock 라이브러리를 초기화

bool ConnectToServer(const std::string& serverIp, int serverPort); // 서버에 연결

bool ReceivePacket(); // 패킷 수신

void HandleReceivedPacket(uint8\_t\* buffer, int bufferSize); // 수신된 패킷 처리

// 각 패킷을 처리하는 함수

bool HandleInputPacket(const Input\_Packet& packet);

bool HandlePlayerIDRequest();

bool HandleServerTimePacket(const GTime\_Packet& packet);

bool HandlePlayerIDResponsePacket(const PlayerIDResponsePacket& packet);

bool HandleObjectInfoPacket(const ObjectInfo\_Packet& packet);

bool HandleEndGamePacket(const EndGame\_Packet& packet);

bool ReceiveGTimePacket(const GTime\_Packet& gTimePacket)

bool SetSocketTimeout(int timeoutInSeconds); // 타임아웃

bool HandleNetworkError(int errorCode); // 네트워크 오류 처리

4. 팀원 별 역할분담

|  |  |
| --- | --- |
| 이동재 | 클라이언트 프레임워크 제작, SendPacket함수, ReceivePacket 함수, ObjectInfoPacket 함수, HandleReceivedPacket 함수 제작 |
| 박병일 | InputPacket함수, PlayerIDRequest 함수, PlayerIDResponse 함수, EndGamePacket함수, RetrySend 함수 제작, ~~Update Thread 구현~~, UpdateThreadFunc |
| 서가은 | SendGTimePacket함수, SetSocketTimeout함수, HandleNetworkError 함수 제작, ~~Communication Thread 구현~~, 리소스 제작, CommunicationThreadFunc |

5. 개발환경

|  |  |
| --- | --- |
| 구분 | 내용 |
| Protocol | TCP/IP |
| IDE | Visual Studio 2022 |
| 사용 API | WINAPI / Windows Socket API |
| 개발 언어 | C / C++ |

6. 개발일정

수정전

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 월 | 화 | 수 | 목 | 금 | 토 | 일 |
| 11/4 | 11/5 | 11/6 | 11/7 | 11/8 | 11/9 | 11/10 |
| 이동재 | 클라이언트 프레임워크 제작 | 개인학습 | 클라이언트 프레임워크 제작 | 클라이언트 프레임워크 제작 | 시험공부 | 검토 및 회의 | 개인 학습 |
| 박병일 | 개인학습 | 개인학습 | UpdateThreadFunc 구현 | UpdateThreadFunc 구현 | UpdateThreadFunc 구현 | 검토 및 회의 | 개인 학습 |
| 서가은 | 다렉텀프제작 | 구조체 선언 | 로딩창 리소스 제작 | 시험공부 | Communication Thread 뼈대 구현 | 검토 및 회의 | 개인 학습 |
|  | 11/11 | 11/12 | 11/13 | 11/14 | 11/15 | 11/16 | 11/17 |
| 이동재 | 시험공부 | 시험공부 | SendPacket 함수 구현 | SendPacket 함수 구현 | 개인학습 | 검토 및 회의 | 개인 학습 |
| 박병일 | SendPlayerIDRequest함수 구현 | 시험공부 | SendPlayerIDResponse 함수 구현 | SendPlayerIDResponse 함수 구현 | 어학자격시험 | 검토 및 회의 | 개인 학습 |
| 서가은 | 3D애니기획발표준비 | 시험공부 | Communication Thread 뼈대 구현 | Communication Thread 뼈대 구현 | 개인학습 | 검토 및 회의 | 개인 학습 |
|  | 11/18 | 11/19 | 11/20 | 11/21 | 11/22 | 11/23 | 11/24 |
| 이동재 | ReceivePacket함수구현 | ReceivePacket함수 구현 | 개인학습 | Merge | SendObjectInfoPacket 함수 구현 | 검토 및 회의 | 개인 학습 |
| 박병일 | SendInputPacket 함수 구현 | SendInputPacket 함수 구현 | 개인학습 | Merge | SendEndGamePacket 함수 구현 | 검토 및 회의 | 개인 학습 |
| 서가은 | SendGTimePacket 함수 구현 | SendGTimePacket 함수 구현 | 개인학습 | Merge | SetSocketTimeout 함수 구현 | 검토 및 회의 | 개인 학습 |
|  | 11/25 | 11/26 | 11/27 | 11/28 | 11/29 | 11/30 | 12/1 |
| 이동재 | SendObjectInfoPacket 함수 구현 | Merge | 개인학습 | HandleReceivedPacket 함수 구현 | HandleReceivedPacket 함수 구현 | Merge | 검토 및 회의 |
| 박병일 | SendEndGamePacket 함수 구현 | Merge | 개인학습 | RetrySend 함수 구현 | RetrySend 함수 구현 | Merge | 검토 및 회의 |
| 서가은 | SetSocketTimeout 함수 구현 | Merge | 개인학습 | HandleNetworkError 함수 구현 | HandleNetworkError 함수 구현 | Merge | 검토 및 회의 |
|  | 12/2 | 12/3 | 12/4 | 12/5 | 12/6 |  | |
| 이동재 | 개인학습 | 최종 점검 | | 보고서 작성 | 제출 & 최종 평가 |
| 박병일 | 개인학습 |
| 서가은 | 개인학습 |

수정후

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 월 | 화 | 수 | 목 | 금 | 토 | 일 |
| 11/4 | 11/5 | 11/6 | 11/7 | 11/8 | 11/9 | 11/10 |
| 이동재 | ~~클라이언트 프레임워크 제작~~ |  | ~~클라이언트 프레임워크 제작~~ | ~~클라이언트 프레임워크 제작~~ |  | ~~검토 및 회의~~ |  |
| 박병일 |  |  | ~~UpdateThread 구현 ->~~  ~~서버 프레임워크~~ | ~~UpdateThread 구현 ->~~  ~~서버 프레임워크~~ | ~~UpdateThread 구현 ->~~  ~~서버 프레임워크~~ | ~~검토 및 회의~~ |  |
| 서가은 |  | ~~구조체 선언~~ | ~~로딩창 리소스 제작~~ |  | ~~CommunicationThreadFunc 구현~~ | ~~검토 및 회의~~ |  |
|  | 11/11 | 11/12 | 11/13 | 11/14 | 11/15 | 11/16 | 11/17 |
| 이동재 |  |  | ~~SendPacket 함수 구현 ->~~  ~~PlayerIDRequest 함수 구현~~ | ~~SendPacket 함수 구현~~ –>  ~~PlayerIDResponse 함수 구현~~ | SendPacket 함수 구현 | 검토 및 회의 |  |
| 박병일 | ~~PlayerIDRequest함수 구현~~ |  | ~~PlayerIDResponse 함수 구현~~ ->  ~~UpdateThread~~ | ~~PlayerIDResponse 함수 구현~~ ->  ~~UpdateThread~~ | UpdateThread | 검토 및 회의 |  |
| 서가은 |  |  | ~~CommunicationThreadFunc 구현~~ | ~~CommunicationThreadFunc 구현~~ |  | 검토 및 회의 |  |
|  | 11/18 | 11/19 | 11/20 | 11/21 | 11/22 | 11/23 | 11/24 |
| 이동재 | ReceivePacket함수구현 | ReceivePacket함수 구현 | Merge |  | ObjectInfoPacket 함수 구현 | 검토 및 회의 |  |
| 박병일 | InputPacket 함수 구현 | InputPacket 함수 구현 | Merge |  | EndGamePacket 함수 구현 | 검토 및 회의 |  |
| 서가은 | GTimePacket 함수 구현 | GTimePacket 함수 구현 | Merge |  | SetSocketTimeout 함수 구현 | 검토 및 회의 |  |
|  | 11/25 | 11/26 | 11/27 | 11/28 | 11/29 | 11/30 | 12/1 |
| 이동재 | ObjectInfoPacket 함수 구현 | Merge | HandleReceivedPacket 함수 구현 | HandleReceivedPacket 함수 구현 | 디버깅 및 수정 |  | 검토 및 회의 |
| 박병일 | EndGamePacket 함수 구현 | Merge | RetrySend 함수 구현 | RetrySend 함수 구현 | 디버깅 및 수정 |  | 검토 및 회의 |
| 서가은 | SetSocketTimeout 함수 구현 | Merge | HandleNetworkError 함수 구현 ->클라 및 서버 로직 새로 구현, 클라에서 모든 패킷 send 구현, 서버에서 모든 패킷 recv 구현 | HandleNetworkError 함수 구현 ->클라 및 서버 로직 새로 구현, 클라에서 모든 패킷 send 구현, 서버에서 모든 패킷 recv 구현 | 디버깅 및 수정 |  | 검토 및 회의 |
|  | 12/2 | 12/3 | 12/4 | 12/5 | 12/6 |  | |
| 이동재 | 디버깅 및 수정 | 최종 점검 | | 보고서 작성 | 제출 & 최종 평가 |
| 박병일 | 디버깅 및 수정 |
| 서가은 | 디버깅 및 수정 |