|  |
| --- |
| 한국산업기술대학교 게임공학과 |
| 포트폴리오 |
| 서버 부분 |

|  |
| --- |
| 김대훈  2019-00-00 |

목차 (제대로 수정)

[1. 멀티플레이어 대전 액션 게임 0](#_Toc18407817)

[기본 설명 (알약전사에 대한 대용이면 제목을 따로 붙여야 한다.) 0](#_Toc18407818)

[세부 설명 0](#_Toc18407819)

[제작하면서 어려웠던 점. 0](#_Toc18407820)

1. 멀티플레이어 대전 액션 게임



**이미지 설명 -** 다수의 클라이언트가 동시 접속하여, 플레이 하는 모습

멀티플레이어 대전 액션 게임에 대한 설명이 없다.

기본 설명 (알약전사에 대한 대용이면 제목을 따로 붙여야 한다.)

**게임 제목**: 알약전사

**프로젝트 소스:** <https://github.com/GogleEarth/Pills-Fighter> (다른 학생들의 포트폴리오도 있으니 따로 github를 파서 올려놓을 것)

**사용 언어:** C++11

**개발 환경**: Visual Studio 2017 64bit

**장르:** 3인칭 로봇 대전 액션

**제작 인원**: 3인 (서버&아트 [본인], 클라이언트, 기획 & 클라이언트)

**제작 기간**: 약 13개월

**목표**: 윈도우 소켓, IOCP를 활용하여 3인칭 대전 액션 게임서버 제작

설명 : 미래를 배경으로한 로봇 전투 액션 게임으로 팀 대항 매치이고…… ..

세부 설명

1. 게임 구조:   
   윈도우 소켓, IOCP를 3인칭 대전 액션 게임서버 제작.  
   1000명의 유저를 받을 수 있으며, 최대10개의 방을 만들 수 있습니다.  
   유저는 로비에서 닉네임을 바꾸거나 방을 만들고 참여할 수 있으며, 방을 만든 플레이어가 게임시작 버튼을 누르면 모든 유저의 로딩이 끝나는 것을 기다린 뒤 게임이 시작됩니다.  
   플레이 가능한 캐릭터는 3종류가 있으며 모든 캐릭터는 근접 무기1종, 원거리 무기2종을 가지고 있습니다.  
   상대편 캐릭터를 처치하고 상대편 스코어를 0으로 만드는 것이 승리 목적이며, 게임 중에는 선택된 맵에 따라서 시간의 경과에 따라 이벤트가 발생합니다.
2. 서버 상세설명:
3. IOCP 사용 이유:   
   다수의 스레드 수는 스레드들간의 컨텍스트 스위칭으로 인하여 운영체제 오버헤드를 발생하는데, 이를 줄이기 위하여 IOCP를 사용하였습니다. (IOCP의 사용이유로는 약하다. IOCP를 사용하지 않고도 스레드 수를 줄일 수 있다. 예. overlapped-io)
4. 스레드 종류와 수
   1. Accept\_thread(1개): 유저의 접속을 받아들이는 스레드로 유저가 접속을 하면 현재 존재하는 방의 정보를 유저에게 알려준 후 소켓을 IOCP에 연결해주고, 수신대기상태로 만듭니다.
   2. Worker\_threads(4개(cpu의 코어 수)): 서버의 실제 작업을 하는 스레드로 유저와 통신을 주고받고 패킷의 해석 및 처리, 각종 이벤트를 처리합니다.
   3. Timer\_thread(1개): 시간에 맞춰 발생할 이벤트를 PostQueuedCompletionStatus()함수를 통해 Worker\_thread에 알려주는 스레드로 오브젝트의 update, 방의 update, 플레이어 리스폰 등의 이벤트를 알려줍니다. 프레임 워크가 가지고 있는 우선순위 큐에 인자 값(오브젝트 index, 방 번호, 이벤트 시간, 발생할 이벤트)을 가지는 EVENT\_ST구조체를 넣어 발생 시간이 되면, IOCP에 이벤트를 추가하고, Worker\_thread에서 해당 이벤트를 처리합니다.
5. 패킷 : 패킷 구조체의 처음 2BYTE는 1BYTE는 패킷의 사이즈, 다음 1BYTE는 패킷의 타입이 들어가며 뒤는 패킷에 따라 필요한 정보가 추가로 붙는 형태입니다. (몇가지 패킷의 구조체 선언을 샘플로 보여줄 것)

1. Over\_Ex구조체 : WSAOVERLAPPED구조체를 확장하여 사용한 구조체로 Worker\_thread에서 GetQueuedCompletionStatus()함수를 이용해 send와 recv만이 아닌 다양한 이벤트 처리를 위해서는 확장된 WSAOVERLAPPED구조체가 필요하였습니다. (over\_ex니 messageBuffer니 하는 이름들은 선배들이 제출했던 포트폴리오에 이미 있던 것들이기 때문에, 이름을 바꿀 것, 제일 좋은 것은 수업시간에 가져온 코드들만 처음부터 새로 짜는 것)
   1. event\_t – 이벤트의 타입
   2. elapsed\_time – 이벤트는 대부분 1프레임(16ms)단위로 실행되는데 그보다 빨리 실행되었을 경우 16ms로 보정해 주기위한 변수
   3. room\_num – 이벤트가 실행될 방 번호
   4. dataBuffer – WSARecv(), WSASend()에 인자로 들어갈 WSABUF구조체
   5. messageBuffer – dataBuffer의 실제 buffer가 들어갈 메모리 공간
2. 동접 테스트 : 추가해야함.

제작하면서 어려웠던 점.

(어려웠던 문제들이 다 개념적인 문제들인데 이런 것 보다 좀더 구체적인 문제들을 추가하고, 해결한 소스코드도 첨부할 것)

1. 해당 서버는 초기에 멀티 쓰레드 서버로 구축되었는데 이후 IOCP 서버로 바꾸기 전까지는 기능을 추가하는데 어려움이 많았습니다. 멀티 쓰레드 서버의 특성상 다수의 쓰레드를 관리하게 되는데 기능을 추가할수록 어디서 버그가 발생하는지 쉽게 발견할 수가 없어서 힘들었습니다. (IOCP도 멀티쓰레드 관련 문제들은 그대로 가지고 간다. IOCP가 멀티쓰레드만 쓴 서버보다 좋은 점은 성능 뿐이다.)
2. thread에 대해 정확히 알지 못해 인자로 클래스의 멤버 함수를 사용할 수 없었던 것을 몰랐던 것이 문제가 됐었습니다. 람다를 사용하는 것을 해답으로 찾긴 하였습니다만 별거 아닌 문제일 수도 있지만 해답을 찾지 못했다면 서버의 코드가 지금과는 많이 달랐을 것입니다. (C++11의 functional을 사용하면 객체의 멤버함수도 별도의 thread로 실행할 수 있다.)