문서 버전 관리

|  |
| --- |
|  |
| 상세 기획서 |
|  |
| 게임 서버 |

**한국산업기술대학교 게임공학과**

2017 November 1

저자: 김선필

|  |  |  |  |
| --- | --- | --- | --- |
| 버전 | 수정 내용 | 수정자 | 수정 날짜 |
| 1.00 | 초안 작성 | 김선필 | 17-11-01 |
| 1.10 | 리슨 서버, 전용 서버 개념 페이지 작성 | 김선필 | 17-11-01 |
| 1.20 | 서버 종류별 장단점 추가 | 김선필 | 17-11-02 |
| 1.30 | TCP/IP 서버 (우리가 만들 서버) -🡪 소켓 프로그래밍. | 김선필 | 17-11-07 |
| 1.40 | IOCP 소개 및 기본 | 김선필 | 17-11-10 |
| 1.50 | IOCP 와 DirectX 둘의 연동…. | 김선필 | 17-11-13 |
|  |  |  |  |

목차

목차

[개요 3](#_Toc498332811)

[1. 문서 컨셉 3](#_Toc498332812)

[게임 서버의 종류 4](#_Toc498332813)

[서버 종류별 장단점 5](#_Toc498332814)

[0. 공통 5](#_Toc498332815)

[1. 리슨 서버 5](#_Toc498332816)

[2. 전용 서버 5](#_Toc498332817)

[TCP/IP서버(Similar to Listen Server) 6](#_Toc498332818)

[IOCP 7](#_Toc498332819)

[IOCP의 개념 7](#_Toc498332820)

[간단한 IOCP 소켓 프로그래밍 구조 디자인 7](#_Toc498332821)

[기타 8](#_Toc498332822)

[1. 이미지 출처 8](#_Toc498332823)

개요

1. 문서 컨셉
   1. 이 문서는 팀원에게 만들 게임의 서버를 이해시키기 위해 서술한다.
   2. 기본적인 서버의 지식과 만들 게임에 어울리는 서버를 선택한다.
2. **가정 사항**
   1. 이 문서는 회의간 사용되는 것으로 가정한다.
   2. 회의 간 설명을 보조하는 용도로 사용한다.

# 게임 서버의 종류

1. 리슨(Listen) 서버 / 전용(Dedicated) 서버
   * + 1. 리슨 서버
          1. 사용자가 호스팅하는 서버
          2. 사용자가 만든 서버에 다른 사용자가 참가 가능
       2. 전용 서버
          1. 한 컴퓨터 또는 여러 대의 컴퓨터가 서버만을 전담하는 방식
          2. 보통의 온라인 게임들이 사용하는 방식
2. 예시
   1. 리슨 서버

콜 오브 듀티 – 리슨 서버

* 1. 전용 서버 – “전용 호스팅 서비스” 라고도 한다.
     1. LOL
     2. 일반적인 온라인 게임들

서버 종류별 장단점

1. 공통
   1. 리슨 서버를 택했을 경우에 장단점이 존재한다.
   2. 전용 서버를 택했을 경우 또한 장단점이 존재한다.
2. 리슨 서버
   1. 게임을 종료할 때 서버도 같이 꺼진다.
   2. 따라서 호스팅을 옮기는 절차가 필요하다.
   3. 랜 파티에서 즐겨쓰는 방식
3. 전용 서버
   1. 한 대 또는 여러 대의 고성능 컴퓨터를 기반으로 돌아간다.
   2. 사용자는 서버에 접속해서 쾌적한 게임을 즐길 수 있다..
   3. 게임을 끄더라도 서버는 계속 돌아간다.

# TCP/IP서버(Similar to Listen Server)

1. 동작방식

-서버는 먼저 실행하여 클라이언트가 접속하기를 기다린다.

- 클라이언트가 서버에게 접속하여 데이터를 보낸다.

- 서버는 클라이언트 접속을 수용하고, 클라이언트가 보낸 데이터를 받아서 처리한다.

- 서버는 처리한 데이터를 클라이언트에게 보낸다.

- 클라이언트는 서버가 보낸 데이터를 받아서 자신의 목적에 맞게 사용한다.

2. 예상 서버 흐름

- 원속 초기화

- socket함수를 이용하여 1isten-socket이라는 소켓함수 생성

- bind 함수를 이용하여 서버의 지역 IP주소와 포트번호를 설정

- listen 함수를 이용하여 클라이언트의 요청을 기다림

- accept함수를 이용하여 클라이언트의 접속을 받아들임

- recv함수를 이용하여 데이터를 수신하고 출력

- send함수를 이용하여 데이터 전송

- closesocket함수를 이용하여 서버 종료

- 원속 종료

# IOCP

## IOCP의 개념

IOCP – (Input/Output Completion Port)

인풋이나 아웃풋을 대기했다가 완성되는 것을 기다려서 완료되면 그것을 받아 처리한다.

소켓 프로그래밍의 응용이라고 보면 되는데, 서버와 같은 유저들이 같이 접속할 가능성이 있는 프로그램을 작성할 때는 IOCP 방식을 사용해서 프로그래밍하는 것이 좋다고 한다.

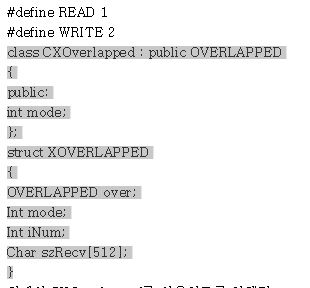
결국 서버의 한 종류이며, 4학년 때 배울 예정인 서버이기 때문에 애초에 이걸로 선택한다.

## 간단한 IOCP 소켓 프로그래밍 구조 디자인

* + - 1. 새로운 IOCP 포트를 생성한다.
      2. Listen 소켓을 하나 만들고 만들어 놓은 IOCP 포트에 추가한다.
      3. 적당한 수의 클라이언트 소켓을 생성하고 accept 시켜 놓는다.
      4. 콜백함수나 쓰레드 등을 이용하여 IOCP 결과값을 기다린다.
      5. 결과값을 이용하여 연산한다.

# IOCP 와 DirectX와의 연동

1. 연동하기 위해 필요한 조건들
   1. IOCP 핸들
      1. Window api 처럼 iocp도 핸들을 사용해서 호출하고 사용한다.
   2. 확장 OVERLAPPED 구조체 또는 클래스
      1. 채팅 서버를 기준으로 사용하는 OVERLAPPED 구조체를 확장해서 사용
      2. 예시)
         1. IOCP로 채팅서버 만들기



* 1. 클라이언트 정보를 저장할 공간
     1. 게임 서버인만큼 거대한 클라이언트의 처리 내용을 담을 공간이 필요하다.
     2. 클라이언트의 정보를 저장하고, 필요한 정보를 꺼낼 수 있어야 한다.

기타

1. 이미지 출처
   1. <http://john-lenon.tistory.com/14> (리슨 서버 게임 이미지 출처)
   2. http://alisa2304.tistory.com/32 (TCP/IP 서버 개념 참조)
   3. <http://cpueblo.com/board/bbs/board.php?bo_table=programming&wr_id=428> (채팅서버)