



소개

AMM42020 게임서버 프로그래밍
정내훈

2020년도 2학기

한국산업기술대학교 게임공학부

내용

- 강좌 소개
- “게임 서버”에 대한 소개

강사 소개

- 경력
 - 1990년부터 온라인게임 개발
 - LPMUD, Archmage
 - 2002년 3월 – 2008년 2월 NCSoft
 - MMORPG 개발 : Lineage forever, Alterlife, Blade & Soul
 - 2015,2016,2017 모바일 게임 서버 개발 Netmarble
 - GSF, FinalShot, Lineage2Revolution
- 전공
 - Parallel Processing
- 관심분야
 - 차세대 게임 서버 구조
- 연락처
 - nhjung@kpu.ac.kr 공학관 E동 314호

개요 – 게임서버 프로그래밍

- 목적

- MMOG(대규모 다중 사용자 온라인 게임) 서버를 디자인하고 구현하기 위해 필요한 지식을 배우고, 실제 제작을 경험해 본다.

개요 – 게임서버 프로그래밍

● 교재

- 배현직, “게임서버 프로그래밍 교과서”, 2019, 길벗
 - 실제 게임서버 엔진을 만든 경험을 수록
 - 저자는 현재 풀어비스 서버 프로그래머
 - 네트워크 게임 프로그래밍의 최신 이슈들을 수록
 - MMOG 보다는 MOG에 가까움
 - 강의는 교재 + a 의 형태
 - 강의 순서는 교재를 따르지 않을 예정.



강의 계획

- 1 주 : 게임 서버 개요
- 2 주 : 네트워크 기본
- 3 주 : 네트워크 응용
- 4 주 : 멀티 스레드
- 5 주 : IOCP (1/2)
- 6 주 : IOCP (2/2)
- 7 주 : 중간고사

강의 계획

- 8 주 : 성능 측정
- 9 주 : DB 사용법
- 10 주 : 스크립트 언어
- 11 주 : 컨텐츠 구현
- 12 주 : 분산 처리, 확장성, 보안
- 13 주 : 사례 연구
- 14 주 : 팀 프로젝트 발표
- 15 주 : 학기말 고사

성적 산출

- 중간 고사 : 20%
- 학기말 고사 : 25%
- 과제 : 45% (15% 숙제, 30% 텀프로젝트)
- 출석 : 10%
 - 결석 1번에 1% 감점, 지각 3번에 1% 감점,
1/4이상 결석 F)

선수 과목

- 모든 전공 필수 과목
- 많은 도움이 되는 과목
 - MM3330 : 데이터베이스
 - MM3220 : 네트워크 게임 프로그래밍
 - 가을 학기 개설 시에는 병행해도 지장이 없도록 할 예정
 - MM1240 : C++
- 질문 : 개발 환경 경험?

강의 편성

- 숙제
 - 강의 내용 실습
 - 각 숙제는 연계되어 있음
 - Delay시 하루당 10%감점
- 텀 프로젝트
 - 1인 프로젝트
 - 클라이언트 담당 X
 - 간단한 MMORPG게임 만들기
 - 숙제와 연계되어 있음

개발 환경

- 운영 체제
 - windows 10
- 언어
 - C++11
 - Visual Studio 2019 (community)
- DB
 - MS SQL 2019 (express)

내용

- 강좌 소개
- “게임 서버”에 대한 소개

게임의 구분

- StandAlone 게임

- 구입 후 혼자 하는 게임
- 게임 머신 만 있으면 실행 가능
 - 게임머신 : PC, 콘솔, 닌텐도 2DS
- 지금은 국내에서 찾아 볼 수 없음
 - 원인 : 불법 복사

- Online 게임

- 게임을 실행할 때 인터넷 연결이 필수인 게임
 - 과금 정보가 게임회사에 저장되어 있음. => 불법 복사 근절
 - 무료 게임 조차 인터넷 연결 필수
 - StandAlone게임의 온라인화 : 스팀
- 혼자 하는 게임도 있고 여럿이 하는 게임도 있고, 둘 다인 게임도 있음

온라인 게임 구분

- 초기 모바일 게임
 - Login 필요 (사용자 과금 정보 서버 저장)
 - 게임은 StandAlone으로 실행, 게임 결과 서버 업데이트
 - 예) 애니팡, 쿠키런, 세븐나이츠
- Peer to Peer (P2P)게임
 - StandAlone게임에 네트워크 대전 지원
 - 로비 서버를 사용한 match making
 - 게임 진행은 사용자 Machine들에서만 이루어짐
 - 예) Diablo1, Starcraft1
- Client/Server 게임
 - 모든 게임 컨텐츠의 실행과 정보가 서버에 저장되어 있음
 - 클라이언트는 단지 그래픽 정보 표현과 사용자 입력 전달만 담당
 - 해킹에 매우 강함.

클라이언트/서버 게임 구분

- MO (Multiplayer Online)

- 한 세션에 여러 명이 모여서 게임을 진행
 - 2명에서 16명, 100명(배그)
- 세션이 끝나면 게임 월드와 플레이어 정보 리셋
- P2P로도 구현 가능
 - Cheating 방지를 위해 C/S로 많이 구현
- 예) StarCraft, Diablo, Overwatch, 배그, LOL

- MMO (Massively Multiplayer Online)

- 게임의 단위가 서버(혹은 Shard, World)
- 같이 플레이하는 플레이어의 단위가 몇 천명
 - 몇 만이 넘는 게임도 있음
- 게임 월드와 플레이어 정보를 DB에 계속 유지 (Persistent World)
- 예) 리니지, 블레이드&소울, 검은 사막, World of Warcraft
- 이번 학기에 다룰 주제

동기화에 따른 구분

- MO (Multiplayer Online)
 - 즉각 적인 반응을 요구하는 게임
 - FPS, RTS
 - 1초에 20번 또는 30번 월드 상태를 모든 클라이언트에 전송
- MMO (Massively Multiplayer Online)
 - 빠른 반응 속도 보다는 전략과 협동 플레이 위주
 - RPG
 - Event발생시에만 데이터 전송
 - 채팅, 타격, 스킬/아이템 사용, 이동(약 1초 간격)

게임 서버란?

- MMO 에서의 게임서버



MMO 장르

● 모바일 : 2020년 8월 31일

Android 앱 Google Play

← → ↺ 🏠 <https://play.google.com/store/apps/collection/cluster?clp=0g4YChYKEHRvcGdyb3NzaW5nX0dBTUUQBxgD:S:ANO1ljLhYwQ&gsr=ChvSDhgKFgoQdG9wZ3Jvc3NpbmdfR0FN...> ☆

Google Play

검색

앱







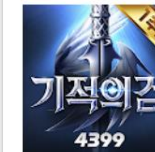



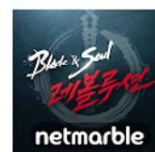


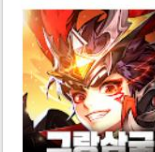
카테고리 ▾ 홈 인기 차트 신작

내 앱
쇼핑하기

계정
결제 수단
Play Points
내 구독
코드 사용
기프트 카드 구매
내 위시리스트
내 Play 활동
자녀 보호 안내

신규

최고 매출 게임

 리니지M NCSOFT ★★★★★	 바람의나라: 연 NEXON Company ★★★★★	 리니지2M NCSOFT ★★★★★	 R2M Webzen Inc. ★★★★★	 V4 NEXON Company ★★★★★	 라그나로크 오리진 GRAVITY Co., Ltd. ★★★★★	 기적의 검 4399 KOREA ★★★★★
 라이즈 오브 킹덤즈 LilithGames ★★★★★	 뮤 아크엔젤 Webzen Inc. ★★★★★	 가디언 테일즈 Kakao Games Corp. ★★★★★	 블레이드&소울 레 Netmarble ★★★★★	 카트라이더 러쉬플러스 NEXON Company ★★★★★	 리니지2 레볼루션 Netmarble ★★★★★	 그랑삼국 YOUZU(SINGAPORE)PT ★★★★★

좋은 게임 서버

- 안정성
 - 서버 다운이 없어야 한다.
 - 서비스에 치명적
 - 1/2 법칙 : 초창기에는 서버 다운 한번에 고객의 반이 그만둔다.
- 보안
 - 해킹, 복사, 오토
- 성능
 - 동접자 수/서버 비용
 - 서버 비용 = 컴퓨터 값 + 네트워크 회선 비 + 임대료 + 운영비...
 - 동접자 수/월드
 - 게임의 재미에 큰 영향
 - 시골 서버와 도시 서버
 - HotSpot
 - 이벤트, 공성전, 길드전, 상대 도시 레이드

게임서버 제작 시 고려점

- 서버 다운

- 원인 : 멀티쓰레드 버그
- 해결책 : **Test**, 유능한 프로그래머 고용

- 보안

- 게임서버를 만드는 이유 (vs. P2P)
- 서버 프로그래머 책임 : 복사, 서버 프로그램 해킹
- 기획 + 서버 프로그래머 책임 : 오토
- 운영자 책임 : 서버 컴퓨터 해킹, **DDOS**

게임서버 제작 시 고려점

- 성능

- 동접자 수와 HotSpot 허용량으로 평가
 - 허용량 : 랙이 걸리기 시작하는 동접 수치
 - HotSpot : 같은 장소에 많은 인원이 모이는 현상
- 개선을 위한 고려사항
 - 부하 분산 : 서버 분할, World 분할, 인스턴스 던전
 - 알고리즘 개선
 - Multi-core 활용
 - 오버헤드 감소
 - 스레드 동기화 최소화, 캐시 적중율 향상, 데이터 복사 감소

게임서버 제작 시 고려점

● 네트워크 비용

▶ 코로케이션 / Dedicated

VAT별도 / 단위(원)						
구분	10M	20M	50M	100M	Giga	비고
방화벽(F/W)	600,000	700,000	800,000	1,000,000	협의	장비 운용
침입탐지(IDS)	400,000	500,000	600,000	800,000		장비운용 + 보안관제
침입방지(IPS)	700,000	800,000	1,000,000	1,200,000		장비운용 + 보안관제
통합보안(UTM)	1,200,000	1,500,000	1,800,000	2,000,000		장비 운용
웹방화벽(WAF)	1,200,000	2,000,000	2,500,000	2,800,000		장비 운용
클린존	-	-	1,500,000	3,000,000		DDoS 관제 및 방어
클린존 플러스	고객사 구성에 따른 협의					고객사 전용 DDoS 대응 및 클린존 연계

2020/3 KT-IDC 월별 네트워크 비용

게임서버 제작 시 고려점

- 네트워크 부하
 - HW한계 (1Gbps, 10Gbps)
 - IDC 한계 (Internet Data Center) => 비용
 - 서버/클라이언트 프로토콜 설계시 주의점
 - 가능한 적은 데이터 전송
 - 가능한 적은 회수의 데이터 전송
 - 한번 보낼 때 마다 헤더 추가. (Packet 구성 overhead)
 - 전송/수신할 때 마다 운영체제 호출 부하. (성능에 영향)
 - 1KB 열 번 보내는 것 보다 10KB 한번 보내는 것이 훨씬 효율적. (10배 가까이 효율적)

게임서버 제작시 고려점

● 확장성

- 성능을 쉽게 올릴 수 있는가?
 - 플레이어 사이의 인터랙션이 가장 큰 문제
 - N^2 으로 부하 증가
 - CPU속도 한계 극복 필요 : 멀티코어 + 멀티 CPU
 - 분산 구조 => 구현 난이도 대폭 증가.
 - 분산 가능한 것은 무조건 분산 구현 (난이도 순서)
 - 네트워크 오버헤드가 제일 큰 문제

● 네트워크 딜레이 은폐

- 거스를 수 없는 자연 법칙 (특수 상대성 이론)
- 기획부터 고려해야 함

게임서버의 필요성

- 왜 만들기 어렵고 유지관리가 어려운 게임 서버를 운영하는가????
 - 보안 때문
 - P2P는 해킹에 무방비
 - 예) 스타크래프트 맵핵
 - 서버 컴퓨터 해킹은 거의 불가능
 - 리니지의 장수 비결
 - 예) Diablo 1 -> Diablo 2 realm server

고성능 게임서버의 필요성

- 성능
 - P2P의 master 서버로는 HotSpot의 부하를 견딜 수 없음.
 - L모게임 서버 :
 - 64bit Deca Core Dual XEON CPU, 메모리 128GB
- 게임성
 - 많은 동접 : 재미
 - 원활한 경제, Community 활성화, 파티 찾기
 - HotSpot : 재미
 - 이벤트, 공성전, 혈맹전
- 경제성
 - 서버 비용, 관리 비용

고성능 게임서버 프로그래밍

- 특징

- 학교에서 배우는 모든 내용이 들어간다.
 - Algorithm, DB, AI, 3D관련 수학, 물리
 - 운영체제, 컴퓨터 구조, 네트워크, 스크립트언어
- 어렵다.
 - 이유
 - Network, DB, Script.... (X)
 - 멀티스레드 최적화 및 안정화..... (O)

- 결론

- 좋은 게임서버 프로그래머의 길은 험하다.
- 좋은 게임서버 프로그래머는 구하기가 어렵다.
- 몸값 상승

숙제 (#1)

- 게임 클라이언트 프로그램 작성
 - 내용
 - 체스 판을 화면에 그린다. (8x8)
 - 체스 말 하나를 화면에 그린다.
 - 커서 키로 말을 상하좌우로 이동한다.
 - 목적
 - 앞으로 작성하게 될 서버 프로그램의 동작을 확인할 수 있는 프로그램
 - 제약
 - Windows에서 Visual Studio 2019로 작성 할 것
 - 그래픽의 우수성을 보는 것이 아님
 - 제출
 - 9월 8일 화요일 오후 1시까지 제출 (1일 당 10% 감점)
 - Zip으로 소스를 묶어서 제출
 - 컴파일 및 실행 가능할 것, 소스만(sdf, obj, log, manifest 같은것 제외!)
 - E-Class의 숙제 시스템을 통해 제출