1. 게임 프로그래머가 배워야 할 것들

클라이언트 / 서버 공통분야

* 프로그래밍 => C / C++ / C# / Windows API
* API -> 시스템, 메모리부분
* C# -> C#으로 서버를 만들기도 한다.
* 알고리즘 => 정렬 / 길 찾기
* 길 찾기 -> A\*알고리즘 짜보기
* 데이터 구조론 => 스택 / 큐 / 링크드리스트 / 트리 / 해쉬 등
* 클라이언트 -> 스택 / 큐 / 링크드리스트까지 구현해보기.
* 서버 -> 트리 / 해쉬까지 구현해보기
* 트리 -> 레드블랙트리
* 기타 => 파싱 / 컴파일러 / 암호화 / 보안
* 파싱 -> 텍스트를 분석하여 구현하는 법
* 컴파일러 -> C언어 같은 것을 만들어 본다. (패키지 RPG에서 쓰는 경우가 있다.)
* 라이브러리 -> C++ STL표준 라이브러리를 쓴다.
* 콘솔 쪽으로 가려면 자신의 라이브러리를 쓰기도 한다.
* 암호화 -> 클라이언트 메모리 핵을 막기 위해 1차로 함
* 암호화는 어차피 다 막을 수는 없음 -> 오토나 핵이 쉽게 만들어지지 않게끔 한다.
* 보안 -> 생활 습관
* 현재 보안을 뚫는 가장 쉬운 방법은 개발자를 뚫는 것이다.
* 이메일 링크는 절대 클릭하지 말 것.
* P2P에서 OS이미지 파일 다운 금지 (어차피 OS에 다 깔려있음)
* 개인용, 개발용 두 개로 운영하는 것이 좋음.
* 공유기를 사용 할 것.

1. 클라이언트 전문기술

* Direct3D / Unreal / Unity3D / OPEN GL
* Direct3D -> 최우선으로 해야 한다. 렌더링 / 쉐이더
* Unreal -> 대형개발에 적합, 확실한 취업, 소스분석이 좋음
* Unity3D -> 빠른 취업
* OPEN GL -> 모바일 / 콘솔

1. 서버 전문기술

* 네트워크 프로그래밍 ( Socket)
* 데이터베이스 MSSQL, MYSQL
* MSSQL -> 많이 씀 / MYSQL –> 해외에서 주력으로 씀 (추천)
* 멀티 스레드 프로그래밍
* 웹 프로그래밍 => PHP, JAVA, 파이썬
* 고수준 디버깅

1. 전통방식 서버 (Statefull Game Server)

* C++을 이용한 TCP Socket 프로그래밍
* Windows Iocp Network Programming
* 고급 스레드 프로그래밍
* 고급 메모리관리 / 디버깅
* 네트워크 인프라나 외부 솔루션 보다는 C++을 통한 직접 게임 서버를 개발 하는 부분에 집중.
* 언어 자체에 난이도가 높음
* MMO, MO, PC게임에 적합함. (ex – 배틀그라운드, 오버워치, 롤, 히오스 등등)

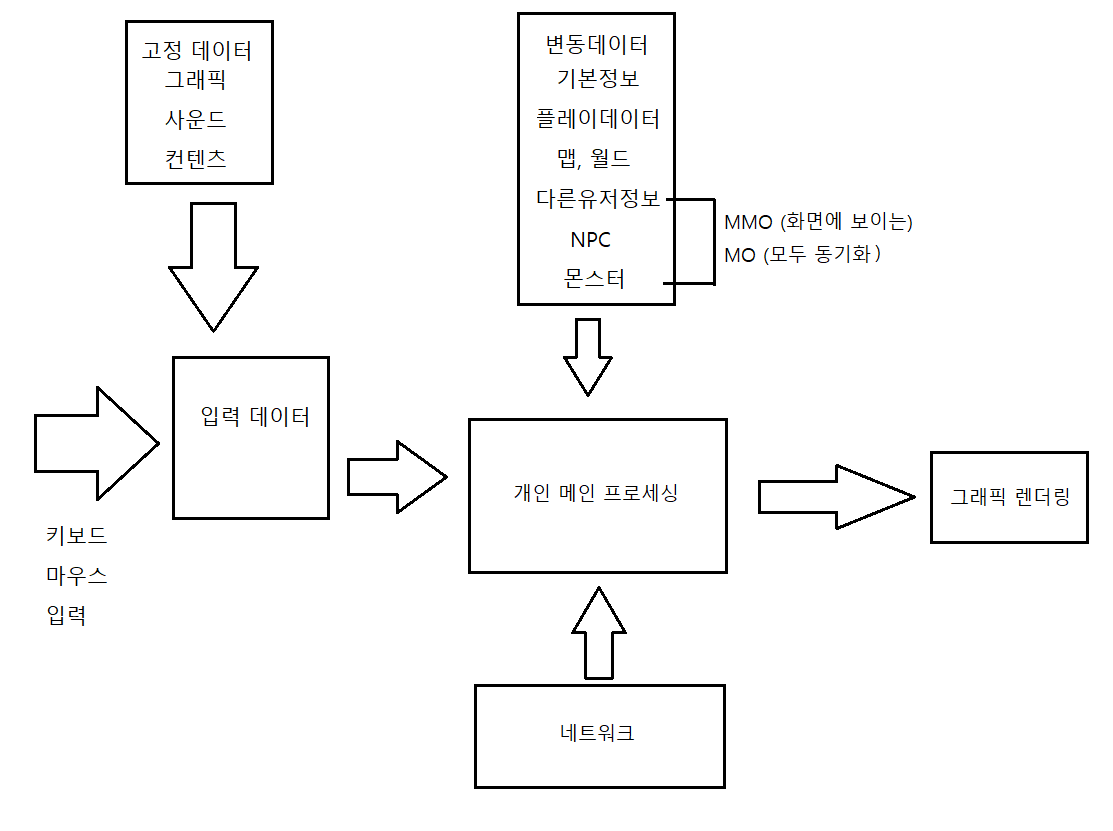
1. 최신 트렌드 서버 (Stateless Game Server)

* 모바일, 싱글, 퍼즐, RPG, 캐주얼게임에 적합
* HTTP 프로토콜 기반 웹서버 방식
* PHP, 파이썬, JAVA 등의 웹언어 사용
* 웹 개발자들이 게임분야로 대거 이동
* 언어 자체 난이도가 굉장히 낮음
* 네트워크 인프라 및, 외부 솔루션들의 의존도가 높음
* AWS(아마존 웹 서비스) / Database
* 배틀그라운드 경우 AWS의 서버를 대여해서 Unreal 데디게이트 서버구축
* 데디게이트 서버 -> MO게임서버 만들기가 쉬움 (Unreal)

1. 게임 구조

* 최근 클라이언트에서 로직을 짤 수 없음. 대부분의 로직은 서버에 있음.
* Ex) 트리오브세이비어 -> 로직을 클라이언트에 넣음 -> 핵이 많음
* 게임 설계 (기본 서버 개발자 + 클라이언트 개발자 + 경험이 많음 기획자)
* 테이블 결정 및 설명
* 모든 데이터를 엑셀로 수치화함 -> CSV로 저장해서 파싱한다.
* 아이템복사 -> 지역별 서버로 나누다 보니 DB에 실시간 동기화가 안돼서 생길 수 있음. / DB동기화 과정에서 자주 일어남.

1. 클라이언트 구조



1. 서버구조

