

게임 클라이언트 신입 면접 예상질문

이서영

C++

OOP

OS

DSA

컴퓨터 그래픽스

멀티스레드

네트워크

기타

인성

C++

C언어와 C++ 언어의 차이점

C#언어와 C++ 언어의 차이점

C++ 언어의 장점은 무엇인가?

- 객체지향 / 상속
- GC / 스마트 포인터 / VM / OP code
- VM X / 멀티패러다임 / 게임 메모리, 리소스 관리

C++

콜 바이 벨류(Call by Value)란?

콜 바이 레퍼런스(Call by Reference)란?

콜 바이 벨류(Call by Value)와 콜 바이 레퍼런스(Call by Reference)의 차이점은?

포인터(Pointer)란?

참조자(Reference)란?

포인터(Pointer)와 참조자(Reference)의 차이점은?

- 함수 호출 / 값 넘기기
- 포인터 넘기기 / 레퍼런스로 넘기기
- 함수 실행 후 결과
- 주소에 대한 변수 담기
- 그 변수와 동일 공간 또다른 이름
- 내부적 동일 작동 해당 메모리 접근 / 포인터 또다른 변수 / 참조 새로운 메모리 X

C++

new란?

malloc이란?

new와 malloc의 차이점은?

정적할당이란?

동적할당이란?

- 힙 메모리 할당 / 클래스 크기
- 힙 메모리 할당 / 프로그래머가 원하는 크기
- 생성, 소멸자 호출
- 스택 메모리 영역 사용 / 함수 호출
- 힙 메모리 영역 사용 / new / 런타임 / 직접 해제 / stl vector, string

C++

가상 함수(Virtual)란?

가상 함수 테이블이란?

- 상속관계 사용할 함수 정하기 / 소멸자 virtual / 실제 메모리 형식
- 객체 _vftable / 클래스마다 가상함수 테이블 / offset

C++

업 캐스팅이란?

다운 캐스팅이란?

업 캐스팅과 다운 캐스팅의 차이점은?

죽음의 다이아몬드란?

- 상속 부모 객체로 형변환
- 상속 자식 객체로 형변환
- 업 안전 / 다운 X 안전, 이미 알 때만 사용
- c1-1 c1-2 동시 상속

C++

static_cast란?

dynamic_cast란?

static_cast와 dynamic_cast의 차이점은?

- 일반적인 캐스팅 / 컴파일타임 캐스팅
- 다운 캐스팅 안전? / 런타임 확인 / 컴파일러 옵션
- dynamic 런타임에 체크 -> 느려짐

C++

얕은 복사란?

깊은 복사란?

스마트 포인터란?

L Value, R Value란?

- 생성자, = 에서 값만 넣기 / 디폴트
- 주소값까지 가서 있는 내용 복사하기
- 스코프 기반 / ref counting 기반 / delete / 메모리 릭 / gameobject - component
weak pointer 상호참조 막기
- 왼쪽 메모리에 쓰일 값 / 오른쪽 임시 값, 저장 안하는 값 / && 생명연장 / move

C++

람다식(Lambda)이란?

static 메서드란?

템플릿이란?

- 이름없는 함수 / 한번만 실행 / 버튼 만들기 / 콜백
- 객체 X / 멤버변수 X, this X, static 변수 O
- 코드 런타임에 생성 / 벡터 타입 자유

OOP

객체지향(OOP)이란?

객체지향(OOP)의 4대 특징이란?

객체지향(OOP)의 5대 원칙이란?

- 객체처럼 생각 / 변수 + 함수 / 상속, 다형성
- 추상화, 캡슐화, 상속, 다형성
- SOLID SRP, OCP, LSP, DIP, ISP

OOP

오버로딩이란?

오버라이딩이란?

오버로딩과 오버라이딩의 차이점은?

- 함수 인자 다른 것 받을 때 처리
- 상속 관계 virtual 함수
- 상황에 따라 맞는 함수 실행 / 컴파일, 런타임 / 메모리에 올라간 타입

OOP

C++에서 클래스(Class)란 무엇인가?

C++에서 구조체(Struct)란 무엇인가?

C++에서 클래스(Class)와 구조체(Struct)의 차이점은?

- 변수 + 함수 / 객체 생성 정의
- 데이터 모아놓기 / 구조체 패딩
- POD / 기본적 막기, 열기

OOP

상속이란?

인터페이스(Interface)란?

추상 클래스(abstract)란?

- 클래스 재사용 / 프로퍼티, 함수 / 오버라이드
- 공통된 기능 보장 / 각자의 기능
- 상위의 클래스 / 객체 생성 X

OS

메모리 구조는 어떻게 되어있는가?

운영체제(OS)란?

- cpu 레지스터, l1 l2 l3 / ram / ssd
- 여러 프로세스 스케줄링 / io

OS

스택(Stack) 영역이란?

스택(Stack) 영역에 용량 제한이 있는가?

힙(Heap) 영역이란?

힙(Heap)은 영역에 용량 제한이 있는가?

스택(Stack) 영역과 힙(Heap) 영역의 차이점은?

- 함수가 실행되며 쌓이고 빠지는 영역 / 콜스택
- 배열 100만개 X
- new로 할당 요청 / 원하는 공간 잡기
- 상대적으로 크다 / alloc 속도 느림
- 메모리 할당 해제 속도 / 실행에 따라 쌓고 빼기, 가능한 영역 찾아서 주기

OS

메모리 단편화란? Fragmentation?

내부 단편화란?

외부 단편화란?

페이징이란?

세그멘테이션이란?

프레그멘테이션 회피 방법

- 메모리 공간은 있는데 사용 X
- 안쓰는 작은 메모리 공간
- 큰 메모리 못 올리는 상황
- 메모리 올릴 때 페이지로 잘라서 가상 테이블로 매핑 / mmu가 cpu가 쓸 때 변환
- code/data/heap/stack 논리적으로 구분
- 페이징으로 메모리 전체를 일정 조각으로 나누면 메모리 조각내서 남은 부분에 넣기

OS

메모리 풀이란?

- 시스템 콜 줄이기 / 미리 사용할 공간 new, 사용 블록 크기 / new delete 오버로딩

DSA

자료구조란?

STL이란?

이터레이터란?

- 데이터 저장 필요 정리 방법들
- 템플릿으로 필요한 도구
- 공통적인 포인터 연산 제공

DSA

리스트(List)의 개념

리스트(List)에서 at를 구현한다면 어떻게 구현해야하는가?

리스트(List)에서 100만번 째 데이터를 찾는데 검색속도를 줄이려면?

- 다음 데이터 위치 반복적으로 들고있기
- 카운팅 하나씩 뒤로가며 찾기
- 리스트 만개씩 쪼개놓기 / 인덱싱 벡터로 바꾸기 / map으로 바꿔서 트리

DSA

벡터(Vector)의 개념

벡터(Vector) 사용 시 중간에 삽입이 가능한가?

벡터(Vector) 재할당(Resize)란 무엇인가?

벡터(Vector) 재할당(Resize) 속도 줄이는 방법은 무엇인가?

벡터(Vector)와 배열(Array)의 차이점

맵(Map)의 Find보다 벡터(Vector)의 Find가 더 빠른 경우는 무엇인가?

- 연속된 메모리 공간 / 동적배열
- 그 자리부터 맨 뒤까지 한 칸씩 밀기
- 공간부족, 두 배 할당, 다 옮기기
- 적당히 크게 받고 줄이기
- 사이즈 결정
- 캐시 히트 / 앞에있을 때 순차탐색

DSA

리스트(List)와 벡터(Vector)의 차이점

리스트(List)와 배열(Array)의 차이점

- 다음주소 포인팅, 바로 옆 저장 / 캐시 히트
- 다음주소 포인팅, 바로 옆 저장 / 계속 늘리기, 할당만 사용가능

DSA

스택(Stack)의 개념

큐(Queue)의 개념

스택(Stack)과 큐(Queue)의 차이점

- LIFO 접시 쌓기
- FIFO 대기열
- 순서에 따라 먼저, 나중에 처리

DSA

맵(Map)의 개념

해쉬 맵(unordered_map)의 개념

해시 테이블(Hash Table)의 개념

맵(Map)과 해쉬 맵(unordered_map)의 차이점

- 키 데이터 쌍 / 트리구조
- 키 해싱 데이터 찾기 / 해시테이블 인덱싱
- 해시 함수로 변환, 미리 만든 테이블에 데이터 넣기, 충돌하면 뒤에 연결
- RBT, 해시테이블 / $\log(n)$ 트리 내려가며 찾기, $O(1)$ 바로 인덱스로 찾기

DSA

트리(Tree)의 개념

이진트리(Binary tree)의 개념

레드 블랙 트리(Red-black Tree)의 개념

- 사이클 X, 계층구조
- 작으면 왼쪽 크면 오른쪽
- 한 쪽 쏠리면 $O(n)$, 빨강 검정 노드로 밸런싱 /

DSA

정렬 알고리즘이란?

정렬 중 가장 빠른 정렬과, 그렇게 생각하는 이유는?

퀵 소트 개념

퀵 소트 시간복잡도

빅오 표기법이란?

- 데이터 오름차순, 내림차순
- `std::sort()` / 크기에 따라 insert quick
- 피벗 작으면 왼쪽 크면 오른쪽, 안된 부분 또 나눠서 하기
- 최악 $O(n^2)$, 정렬되어있는데 피벗 앞에만 잡기 / 일반적 $O(n \log n)$
- 알고리즘 실행 경향성 대략적 표현

컴퓨터 그래픽스

벡터(Vector)의 내적이란?

벡터(Vector)의 외적이란?

- 한 벡터 다른 벡터에 투영한 크기 / diffuse / cos / 앞뒤
- 두 벡터에 수직한 벡터 생성 / 만드는 평행사변형 넓이 / sin / 왼쪽 오른쪽

컴퓨터 그래픽스

로컬 좌표계란?

월드 좌표계란?

- 모델 불러왔을 때 처음 버텍스 위치 좌표
- 오브젝트 월드에 배치, 물체의 위치 좌표

컴퓨터 그래픽스

렌더링 파이프 라인이란?

더블 버퍼링이란?

노멀 맵이란?

ambient / diffuse / specular 이란?

그림자 만드는 방법은?

- cpu가 넣은 데이터 gpu가 처리 / 버텍스 점 위치, world view proj, 테셀레이션 정점 처리, 픽셀 텍스처, 라이팅, 노말매핑 디테일, 후처리
- 스왑체인 백버퍼 present 렌더링 끝, 앞으로 한번에 보여주기
- 노말 텍스처로 라이팅 연산 디테일 / 탄젠트 스페이스, 노말 값 표현 정도 / 하이폴리 로우폴리
- 반사된 기본 빛 / 조명과 면 cos 빛받는 정도 / 눈에 반사된 작은범위 빛
- 조명으로부터 depth 값 텍스처에 저장, 위에 그려주기

컴퓨터 그래픽스

KF와 IK란 무엇인가?

스키닝 애니메이션이란?

- 본 구조 골반이 허리 움직이기, 허리가 골반 움직이기
- 관절 부위 버텍스 여러 본 영향 / 본 구조 정보 애니메이션으로 넘겨서 연산하기 / 각 파츠는 부모의 공간 안에서 움직임

MultiThread

세마포어란?

뮤텝스란?

세마포어와 뮤텝스의 차이점

- 카운팅 기반 한번에 접근 가능한 락
- 한번에 하나만 접근 가능한 락
- 리소스 관리 사용 제어, 하나만 오게 하기

MultiThread

프로세스란?

멀티 프로세스란?

싱글 스레드란?

멀티 스레드란?

- 메모리 올라온 프로그램 / cpu 사용 쪽 읽기
- 각각 여러 프로세스 / os에서 타임 나눠서 쓰기
- 하나의 실행흐름만 갖기 / 콜스택 쪽 쌓이고 빠지기
- 하나 프로세스 여러 함수 스레드 일 나눠주기

MultiThread

멀티 쓰레드의 장/단점은?

프로세스와 쓰레드의 차이점은?

락과 쓰레드의 차이점은?

싱글 쓰레드와 멀티 쓰레드의 차이점은?

멀티 프로세스와 멀티 쓰레드의 차이점은?

- 여러 코어 사용 / 효율적 / 디버깅 어렵 / 실행순서 보장
- 분리된 개별적, 스택만 따로쓰기
- 코드 일부분, 객체, 변수 / 실행하는 큰 흐름
- 코어 사용 숫자 / 일 나눠서 실행 효율
- 여러 프로그램, 하나에서 실행 흐름 여러개 나눠서 하기

MultiThread

동기란?

비동기란?

동기와 비동기의 차이점은?

데드락이란?

크리티컬 섹션이란?

- 일반적으로 쪽 실행
- 필요한 처리 분산해서 나중에 값 받기 / future promise
- cpu가 처리 대기
- 자원 필요한데 다른 곳에서 갖고있어서 그곳에서도 기다리는 중
- 여러 곳에서 실행 못하도록 문 잠그기

Network

TCP란?

UDP란?

TCP, UDP의 차이점은?

- 패킷 보낼 때 연결 된 상태, 순서대로 보내기
- 목적지로 그냥 보내기
- 연결된 상태 유지 / 받은 데이터 정리

기타

동적 라이브러리란?

컴파일 순서

전방 선언이란?

정적 라이브러리란?

- 여러 곳 사용 라이브러리 한번만 메모리 올리기
- 샵 문자 처리해서 붙이기, 컴파일러 obj, 링킹 여러 파일 모아서 exe
- 헤더 붙히다 보면 서로 넣기, 있으니까 믿고 쓰기
- exe 안에 포함, 링크할 때 추가

기타

유니코드란?

아스키코드란?

유니코드와 아스키코드의 차이점은?

- utf-8 16 인코딩 방식, 문자 숫자 매핑
- 가장 처음, 한글 X, 문자를 숫자로 매핑 테이블
- 유니코드 8은 아스키, 안시 상위호환 / 1~3바이트 / 한글 지원

기타

디자인 패턴이란?

오브젝트 풀링(Object Pooling)이란?

- 개발할 때 자주 사용하는 패턴 / 싱글턴, 옵저버, mvc, 팩토리 등
- 필요한 객체 미리 만들어놓고 필요할 때 사용 / 무쌍 몬스터 생성 / 생성삭제 줄이기

기타

리플렉션(Reflection)이란?

트랜잭션이란?

- 런타임 객체 정보 / 메타데이터 씬 정보 저장 / serialization, deserialization
- db 쿼리 날리기 / select, update, insert, delete / 처리할 때 디비 상태

기타

부동 소수점 에러란?

Nan이란?

프로파일러란?

boost 라이브러리란?

- float == 비교 / .이하 $2^{-1,2,3,4}$ 로 계산
- 0/0 숫자가 아닌 값
- cpu가 어디서 부하
- 네트워크, 숫자 정밀도 라이브러리

인성

게임을 만들고 싶어하는 이유는 무엇인가?

게임 개발을 위해 노력하는 것은 무엇인가?

무슨 게임 개발을 해보고 싶은가?

살면서 제일 재미있게 했던 게임은 무엇인가?

좋아하는 게임은 무엇인가?

- 게임 좋다 / 콘텐츠 제작, 몰입 / 욕구 피라미드
- 대학교 소위과목 수강, 게임인재원, 알고리즘
- 오픈월드 RPG, 원신, 잘 팔릴 게임
- 원신, 젤다의 전설보다 연속된 세계
- 오픈월드, 팰월드 최근

인성

본인을 프로그래머라고 생각하는가?

어떤 개발자가 되고싶은가?

존경하는 프로그래머가 있다면?

왜 프로그래머가 되고 싶은가?

어떤 목표를 가지고 어떤 프로그래밍을 하고 싶은가?

개발할 때 중요한 것은 무엇인가?

- 프로그램, 프로 X
- 타 직군 협업 / 게임 함께
- 팀스위니, 언리얼엔진
- 사용자 의미, 재밌는 게임
- 전체에서 어떻게 작동되는지 보기

인성

대화를 주도하는 편인가?

본인의 성격은 어떠한가?

스트레스는 어떻게 관리하는가?

본인의 장,단점은 무엇인가?

단점을 극복하기 위해 어떠한 노력을 했는가?

좌우명이 따로 있는가?

- 주도 X,
- infp
- 일기쓰기, 노래방
- 차분함 / 재미없다
- 생각 많이, 내 컨텐츠
- 하면 된다

인성

업무의 성취감이란 무엇이라고 생각하는가?

워라벨과 돈 중 어떠한 것이 더 중요한가?

입사 후 3년 내 목표는 무엇인가?

연봉은 얼마나 받고 싶은가?

- 설정한 목표를 이루는 것
- 신입 돈 + 성장 효율
- 1 1인분 2 전체 파악 3 문제 해결
- 회사 인턴 내규

인성

팀 프로젝트 중 팀원이 일을 못하면 어떻게 할 것인가?

팀 프로젝트에서 가장 중요한 것은 무엇인가?

동료나 상사와 갈등이 있을 때 어떻게 해결할 것인가?

다른 지원자와 무슨 차별점이 존재하며, 왜 당신을 뽑아야하는가?

- 원인파악, 그만큼 더 하기
- 다른 사람 함께 일하기
- 원인파악 문제 해결
- 기록, 게임인재원 1년 / 과거 기록, 뭘 해야할지 판단

인성

따로 자기계발을 하고 있는가?

근래에 읽었던 책은 무엇이며, 가장 인상깊은 책은 무엇이고, 어떠한 장르를 좋아하는가?

최근 일어난 사회 문제에 대해 아는 것이 무엇이 있는가?

그 중 IT 관련 소식에 대해 아는 것이 있는가?

최종적으로 무엇을 개발하고 싶은가?

- 영어자막 영화 드라마
- 프랑켄슈타인 너진똑 / 빅터 프랭클 죽음의수용소에서 / 분석, 중년게이머 김실장, 슈카월드
- 메이플 유저 큐브 확률, 260 6차 라라
- 감동을 주는 콘텐츠 제작