\* garbage collection

\* memory leak

\* Iterator Invalidation

for(iterator it = map.begin(); it != map.end(); ++it) {

map.erase(it->first);

// whoops, now map has been restructured and iterator

// still thinks itself is healthy

}

\* Thread-local storage

\* c++ placement new (with example)

\* Meta-programming(particularlly "template meta-programming")

\* GPGPU

\* 아래의 클래스를 싱글톤 클래스로 만드시오 (지연 생성(Lazy Initialization), 카피 불가(non-copyable) 개념 포함)

class Engine

{

public:

Engine() {}

~Engine() {}

void CreateWindow(int width, int height) {}

};

\* "Big O notation"에 대해서 설명하시오. 바이너리 검색 알고리즘과 이론적으로 가장 빠른 정렬 알고리즘의 계산 복잡도를 Big O notation으로 나타내시오

\* C++의 const\_cast, static\_cast, dynamic\_cast, reinterpret\_cast에 대해 설명하시오

\* "undefined behavior"에 대해 예를 들어 설명하시오

\* Inline 함수에 대해 설명하시오. 이것이 성능에 미치는 영향은? 어떤 경웨 Inline 함수를 사용할 수 없는가?

\* STL의 모든 Container를 적고 그 중 하나를 골라서 내부 구조에 대해 설명하시오

\* std::vector<T>::reserve(size\_t) 함수가 하는 일은 무엇인가?

\* std::vector를 상용하는 것이 std::set을 사용하는 것보다 유리한 경우가 있는데 그 이유를 설명하시오

\* std::lower\_bound와 std::upper\_bound의 차이점에 대해 설명하시오

\* std::unique() 함수가 하는 일은 무엇인가?

\* C-style의 배열을 STLalgorithm함수에 사용할 수 있는가? 있다면 그 이유는 무엇인가?

\* 하나의 STL smart pointer가 C++11에서 제거되었다. 어떤 smart pointer이고 그 이유는 무엇인가?

\* 아래 코드에서 Foo의 생ㅇ성자가 출력하는 내용은 무엇인가?

void f(unsigned long long) { printf("ull"); }

void f(float) { printf("float"); }

template <typename Type>

void f(Type)

{

printf("template");

}

struct Foo

{

void f(double) { printf("scoped double"); }

Foo()

{

f(5ull + 1.0f);

::f(5ull + 1.0f);

}

};

\* 아래 Template함수에서 "std::vector<Type>::iterator" 앞의 "typename"을 삭제하면 올바른 C++ 프로그램이 아닌데 그 이유는 무엇인가?

template <typename Type>

void TestEachElement(std::vector<Type>& vector)

{

typename std::vector<Type>::iterator e = vector.end();

for (typename std::vector<Type>::iterator i = vector.begin(); i != e; ++i)

{

Test(\*i);

}

}

\* C++11 표준에서는 템플릿화된 typedef문을 작성할 수 있게 되었다. 이 기능이 왜 필요한지 예를 들어 설명하시오

\* C++11 표준에 추가된 개념인 "move semantics"에 대해 설명하시오. 왜 이것이 언어기능에 포함될 정도로 중요한가?

\* 아래의 주제드렝 대해서 설명하시오

A. Shadow map

B. Deferred Rendering

C. Gamma Correction

D. Screen-space post processing

\* 렌더링 엔진을 최적화하는 일반적인 절차에 대해 설명하시오. 어떤 툴이 사용될 수 있는가? 어떤 의사결정 과정을 거쳐야 하는가?

\* 아래는 DX11의 렌더링 파이프라인을 순서 없이 나열한 것이다. 순서에 맞게 배열하시오

Domain Shader STage, Tessellator Stage, Input-Assembler Stage,

Pixel-Shader Stage, Geometry Shader Stage, Vertex Shader Stage,

Output-Merger Stage, Hull Shader Stage, Rasterizer Stage

\* 엔진에서 "asset compilation pipeline"이란 무엇인가? 이상적인 asset compilation pipeline의 속성은 무엇인가? 좋지 않은 asset compilation pipeline이ㅣ 가지는 문제점에 대해 설명하시오

\* 일반 모델의 topology로 triangle lists를 사용해야 하는가 아니면 triangle strips를 사용해야 하는가? 각각의 장단점에 대해 설명하시오

\* "post transform vertex cache"란? 왜 이것이 중요한가?

\* 가능하면 셰이더에서 벡터를 정규화하는 것은 피해야 한다. 어떤 방식으로 회피할 수 있는지 설명하시오. 일반적이거나 부분적인 예를 들어 설명하시오

\* HLSL의 ddx()와 ddy()는 무슨 함수인가? 무슨 용도로 사용되는가?

\* Cs\_5\_0 Compute Shader를 작성하고 있다. 지금 작성 중인 셰이더는 아주 병렬적으로 동작하는데 링크드리스트로 되어 있는 여러 개의 bin에 출력 적어 넣어야 한다. 올바른 결과를 얻기 위해 mutex와 비슷한 기능을 만들어서 bin에 결과가 동시에 쓰이는 것을 막고자 한다. Compute shader에서 mutex와 비슷한 기능을 어떻게 만들 수 있는지 설명하시오 (이 문제는 매우 어려운 문제로 창의적인 생각과 약간의 노력이 필요합니다)

\* 아래의 원형들 간의 충돌 여부를 어떻게 알 수 있는지 그 방법을 설명하시오. 알고리즘 전체를 기술할 필요는 없고 이해 정도를 알 수 있을 정도로 기술하시오

A. Line - Line (2D space)

B. Line - Plane

C. Sphere - Sphere

D. Plane - Sphere

E. Line - Sphere (closet point)

\* Delta-time 당 이동 속도가 1m인 물체가 이동 할 때 아래와 같이 장애물을 통과해 버리는 문제를 해결할 수 있는 방법에 대해 기술하시오

\* 아래에 주어진 코드에서 본 행렬을 계산하는 함수인 "ComputerForwardKinematic()"을 작성하시오

struct vector

{

float x, y, z;

}

struct quaternion

{

float x, y, z, w;

}

// pwill appllied first

exern quaternion operator \*(const quaternion &q, const quaternion &p);

exern vector operator \*(const quaternion &q, const vector &t);

exern vector operator +(const vector &t1, const vector &t2);

struct SBonePose

{

int parentIdx; // parent index in bone list, -1 for root bone

vector t;

quaternion q;

}

const int NUM\_BONES = 100;

SPonePose skeletonRelativeBonePose[NUM\_BONES]; // every bone relative pose, ordered by bone index

// comute position/rotation of a specific bone in model space

void ComputerForwardKinematic(int boneIdx, vector &bonePosition, quaternion& boneRotation)

{

// Code here

}

\* Iverse kinematic의 개념과 목적(사용 예)에 대해 설명하시오

\* 2개의 벡터를 보간하는 방법 중 "linear interpolation"과 "spherical interpolation"의 차이점에 대해 설명하시오

\* 3D 회전에 쿼터니온을 사용할 때

A. 허수 부분(w)을 inverse하면 어떤 효과가 나는가?

B. 스칼라 부분(x, y, z)와 허수 부분을 동시에 inverse하면 어떤 효과가 나는가?

\* 좋은 게임 AI란 무엇인가?

\* 스크립트로 된 AI로직의 장점과 단점에 대해 설명하시오

\* 특히 MMO용 AI에 필요한 요구사항에는 어떠한 것이 있는가?

\* 길찾기 시스템에 A에서 B로 이동하는 경로를 계산하는 함수 외에 어떠한 함수들이 필요한가?

\* AI를 위해서는 NPC의 어떤 속성들이 중요하게 쓰이는가?

--------------------------------------------------------------------------------------------------------------------------------

\* 일반적인 축 에 대해서 각 만큼 벡터 v를 회전하는 공식을 유도하시오

\* Field of view가 이고 뷰 영역의 종횡비가 인 투영 변환을 점 P (x, y, z, 1)에 적용했을 때 나오는 투영된 점 P’ (x’, y’, z’, w’)을 구하시오. 정규화된 장치 좌표는 (-1, -1)에서 (1, 1). z투영은 무시하시오