

C++

面向对象三大特征:封装,继承,多态

1. 简述一下C++中三大特征的体现:

...

2. 讲讲C++中多态如何实现? 讲个游戏开发中使用多态的例子

静态多态:函数重载、模板实现

动态多态: 函数重写

3. 虚函数与虚表指针: C++如何知道,我调用一个基类对象身上的方法时,应该执行哪个子类对象身上的函数?

每个类会维护一张虚函数表,这个虚函数表会在编译期间被构建出来,存放在常量存储区。

在创建虚函数表时,如果派生类重写了基类的方法,那么派生类虚函数表将保留重写后的虚函数地址;

如果有一个虚函数派生类没有重写,则虚函数表中保留的是基类虚函数的地址。

编译期间,编译器将函数调用转化为访问虚表指针,虚表指针指向的内容就是虚函数的地址。

4. C++模板和函数重写都是实现多态的方式,两者有什么明显的区别,优缺点?

模板是静态多态,其多态由编译器在编译阶段对模板展开实现; 函数重写使用虚函数表和虚表指针,在运行期推断实现;

两者有性能区别

5. 讲讲静态链接和动态链接的区别, 优缺点

静态链接: 将用到的所有目标文件链接在一起, 生成一个可执行文件。运行时, 所有库都会加载到内存中。优点: 性能较好。 缺点: 浪费内存空间, 更新库困难。

动态链接: 程序运行时将依赖的目标文件加载到内存中。优点: 节省空间, 更新遍历。 缺点: 性能较差

6. 简单介绍一下C++模板, 通常情况下, 可以将模板声明现在头文件中,模板定义写在C++中吗?

不行，这样会导致模板函数调用的位置在进行模板特化时，找不到对应的模板函数实现。因为模板函数在编译成动态链接库的时候，由于没有地方调用过模板函数，所以模板函数不会存储特化版本的符号。此时，调用模板函数的位置也就找不到模板函数的实现了。

C++语法

1. 讲讲const int* a和 int* const a的区别

写在后面表示指针本身不可指向其他内存

2. const放在函数声明开头和const放在函数声明结尾的区别

放开头：表示返回值是常量

放结尾：const成员函数，表示只有const对象可以访问。const成员函数只能访问const成员函数

3. 如何初始化static类型变量？

全局静态变量、类成员静态变量在main函数之前初始化；局部静态变量在运行到这段位置时初始化

C++内存分配

1. 讲讲堆和栈的区别？我在函数中定义的临时变量存放在哪里？

- 管理方式不同:栈由编译器自动管理,堆的分配与释放工作由程序员控制。
- 碎片问题:对于堆来说,频繁的new/delete操作会产生大量磁盘碎片,而栈是先进后出的数据结构,不会有碎片产生
- 空间大小:栈默认1~2m,而32位系统下堆内存可达到4G左右
- 访问效率:有专门的寄存器存放堆的地址,压栈出栈有专门的指令执行,效率很高。堆的分配由库函数提供,效率较低还会产生碎片。

2. C++ 如何在堆上申请一块内存?(malloc、 new) 讲一个malloc和new的区别

- 返回类型的安全性
new返回的是对象类型的指针,无需进行类型转换。故new是符合类型安全性的运算符。
malloc返回void *,需要通过强制转换成我们需要的类型
- new在分配内存之后会构造对象,调用对象的构造函数;使用delete之后也会调用对象的析构函数。
malloc仅仅是开辟了一块内存。

C++智能指针

1. 讲一讲C++11有哪些智能指针?(shared_ptr, unique_ptr, weak_ptr)
2. 简述一下shared_ptr是怎么实现的

普通指针+引用计数; 指向同一对象的所有智能指针共用一个引用计数;

3. 什么时候用shared_ptr, 什么时候用weak_ptr

可以用weak_ptr解决循环引用问题; 另外对一些资源弱依赖时,可以用weak_ptr,防止内存没办法及时回收

4. 讲一讲垃圾回收算法,UE4用的什么方式?

引用计数、标记清除

UE的实现方式:

启动垃圾回收,加锁 (保持所有对象的引用关系不变)

设置所有对象为"不可达"标记 (根对象、特殊对象 除外)

遍历根对象列表,根对象引用到的对象去除"不可达"标记

收集所有仍然标记为"不可达"的对象,全部删除

5. UE4怎么维护对象之间的引用关系?如何保证一个UObject在需要的时候,不被引擎GC?

使用UPROPERTY()宏告诉UE4对象引用关系,UObject需要处于引用链上,或者通过AddToRoot变为根对象

6. GC是一个比较耗时的操作,有哪些优化方式?

1. 打开簇,将Character,Weapon等生命周期一致的 Actor 对象勾选 Cluster
2. 最好的优化还是减少UObject对象数量 (包括:少用蓝图宏,Level内的Actor数量控制)
3. 优化GC调用时机,原则上能不调用就不调用,可在关键点调用
4. 采用对象池,不要频繁清理和生成大的对象

C++11

1. 用过哪些C++11的特性?
2. 简单介绍一下右值引用。左值和右值的区别是什么。

左值: 可以取地址的,有名字,非临时的就是左值。

右值: 不能取地址,没有名字,临时的就是右值。

3. 移动构造函数和拷贝构造函数的区别是什么?编译器会生成默认的移动构造函数吗?

拷贝构造函数触发一次对象拷贝,移动构造函数直接接管临时对象的那一块内存,性能更好。

当没有自定义的移动构造函数、移动赋值运算符、拷贝构造函数,拷贝赋值运算符或者析构函数时,编译器会自动生成一个非explicit的inline public隐式移动构造函数T::T(T&&),即使用户自定义了上述函数,也可以用default来显式地告诉编译器要生成隐式移动构造函数。

如果显示声明了构造函数,那需要显示声明移动构造函数,可以为 `A(const A&& Item) = default;`

4. `std::move`函数的作用是什么? 什么情况应该使用`std::move`?

操作系统

```
for (int i = 0; i < 4000; i++)  
for (int j = 0; j < 4000; j++)  
    nums[i][j] = 1;
```

比较i在前和j在前, 哪种方式遍历速度更快, 为什么?

i在前快, 因为这种方式遍历的内存连续, Cache命中率高于后者

1. 什么是虚拟内存, 为什么要引入虚拟内存的概念?

虚拟内存操作系统引入的, 一块连续的虚拟地址空间。程序通过虚拟寻址技术访问到物理内存。
好处:

1. 为进程提供了一致的地址空间, 降低程序员对程序内存管理的复杂性
2. 让程序可以使用比物理内存还要大得多的内存
3. 保护每个进程的地址空间不被其他进程破坏

2. 简述CPU通过虚拟内存寻址到物理内存的方法

先查缓存, 找不到内容则查页表, 找不到内容触发缺页中断, 置换新页表进来, 然后通过页表找到物理地址

3. 讲一个页面置换算法

最优页面置换算法(OPT): 每次都置换最晚被再次使用的页, 理想状况, 现实中不存在

先进先出算法 (FIFO) : 驻留最久的页面被淘汰

最近最久未使用算法 (LRU) : 最久未被使用的页面被淘汰

时钟页面置换算法(Clock):

数据结构

1. 用过STL吗,讲几个用过的STL数据结构

2. 如何批量删除一个vector中符合条件的元素?

for循环需要倒着遍历删除

使用迭代器遍历删除的话,需要`it = vec.erase(it)`来删除,要注意`erase`后`it`迭代器会失效

3. 如果我在使用迭代器遍历vector的同时,使用`push_back`往数组中增添元素,这个操作会有危险吗? 讲讲vector内存扩容, 扩容因子如何选取?

有可能,触发内存扩容导致迭代器失效

4. 用过map吗,讲讲std:map内部实现

map底层是红黑树,一种平衡二叉树,特点是树的最长路径不超过最短路径的两倍。

增加查找删除都是 $\log n$ 的复杂度

5. map和unordered_map的区别,如何判断该使用哪一种?

红黑树保序, 哈希表快

6. 如何解决哈希表冲突问题

再散列、链地址

7. 了解最短路径算法吗,为什么游戏开发常常选择A算法?(A算法的优势)。

使用了先验知识,往往能更快找到最短路径; 缺点是若估价函数选取不对,可能消耗更大。

8. MC洪水泛滥效果, 如何实现?

lua

1. 讲讲什么是元表? _index和_newindex的区别是什么? 如何忽略_index和_newindex的影响

本身是一个table, 用来描述一个table的行为, 通过 `setmetatable(table, metatable)` 来指定一个元表

_index, 当访问table中的值, 取不到时触发

_newindex, 对table中一个不存在的key赋值时触发

通过`rawset`和`rawget`可以忽略_index和_newindex的影响

图形学

1. 简单介绍一下图形学管线,三角形光栅化是怎么做的?

顶点着色器：转换顶点数据空间

几何着色器：将顶点形成图元，并进行细分

图元装配：图元装配将顶点装配成指定图元的形状

光栅化：图元映射到像素

片段着色器：计算像素颜色

测试和混合：后处理

三角形光栅化：判断点是否位于三角形中，可以通过点和三角形三个顶点依次叉乘，判断符号是否相同

2. phong着色法

Phong光照模型将光照分为三部分：环境光(Ambient)、漫反射光(Diffuse)和镜面(Specular)光照。环境光即来自其它物体，而非直接光源的光照。在Phong模型中是一个常数。漫反射即粗糙物体表面均匀的反射光线到各个方向所产生的光照效果。镜面光照也称高光反射，由光滑物体表面平行地向一个方向反射出来的光照效果。Phong光照模型中，视线向量和反射向量的角度不允许大于90度。如果大于90度的话，点乘的结果就会是负数，镜面的贡献成分就会变成0。

3. 讲一讲法线贴图,为什么法线贴图看起来往往是蓝色的? 为什么法线贴图要存在切线空间?

法线存储在切线空间，切线空间Z轴为模型法线方向，所以(0, 0, 1)就是法线方向。但是向量取值范围(-1, 1)纹理只能存(0,1)，所以做了映射， $\text{pixel} = (\text{normal} + 1) / 2$ ，法线(0,0,1)映射为纹理(0.5,0.5,1)

世界坐标系：当面改变朝向改变时，贴图中的法线就失效了

模型坐标系：需要根据面的朝向，生成不同的法线贴图。想象一个立方体的砖头，他们的6个面的法线情况是一样的，也需要根据面的不同朝向生成对应的法线贴图

UE

- Actor和UObject的关系是什么,Actor相比UObject多了哪些特性? 所有Actor都能放入场景吗?

Actor有生命周期,UE做了整套网络同步,支持添加ActorComponent;

只有包含SceneComponent的Actor能放入场景

- UE可以通过哪些方式实现服务端和客户端的通讯? 如何确定何时使用RPC, 何时使用属性同步?

RPC和属性同步

- 了解哪些UFUNCTION函数标记? BlueprintImplementableEvent 和 BlueprintNativeEvent 有什么区别?

前者定义在C++，实现在蓝图；后者定义在C++，如果蓝图有实现则调用蓝图实现，否则可以调用C++实现

- UHT是什么,UBT是什么?
- 一个背包系统，背包里非常多的图标，并且套在一个滚动框中，如何优化？

游戏相关

1. 例如以撒中的子弹系统，如何设计？
 - 如何实现道具对子弹产生的效果？
 - 如何实现子弹运动？如何检测子弹与敌人的碰撞？
 - 子弹撞到敌人，如何通知敌人受到伤害？
 - 子弹数量太多，发现子弹销毁和创建的消耗很大，如何优化？
2. 如何设计一个成就系统？
3. fps游戏中，玩家A客户端发现一个敌人，进行射击并爆头了，服务器是如何处理这一个过程的？
 1. 命中判定，客户端发送射击信息，服务器做简单校验；
 2. 服务器做延迟补偿，判定命中信息，客户端要做预测，比如播放子弹轨迹和特效，不然表现不好

手撕代码

【NC27 集合的所有子集(一)】 <https://www.nowcoder.com/share/jump/25774691730286959429>