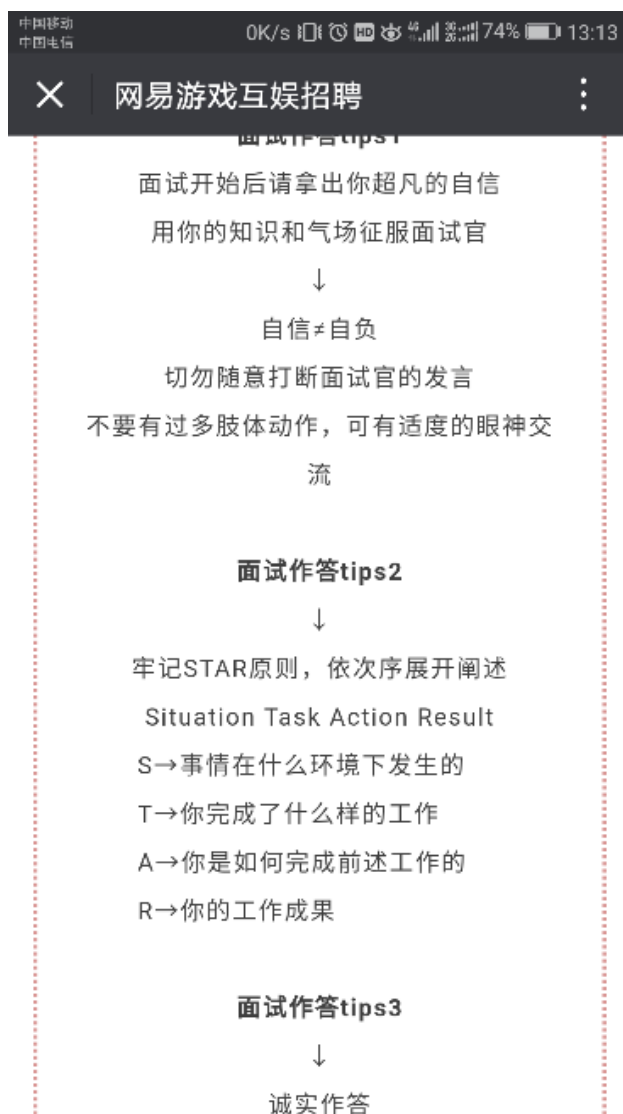


(复杂) 周围同学_游戏公司客户端开发面试题汇总_2018年



一、巨人：主要问项目

1. 笔试题翻出来再考考
2. tcp连接有多少种状态？怎样查看？<http://blog.csdn.net/billcyj/article/details/79465733>
3. 写一个strcpy函数<http://blog.csdn.net/BillCYJ/article/details/79529592>
4. stl与boost熟悉吗？<http://blog.csdn.net/billcyj/article/details/78815053>
5. 用过stl中哪些组件？答：STL有三个主要概念：容器、迭代器、算法，容器又分为序列容器（我用过vector、deque）、关联容器（set、map、multiset、multimap）、容器适配器（stack、queue）

6. map与set底层是？元素有序吗？<http://blog.csdn.net/billcyj/article/details/78996415>
7. 怎样避免一个对象给另外一个对象赋值？答：最常见的方法是把对应类中的拷贝构造函数和拷贝赋值函数定义为private类型。拷贝构造函数(e.g.A a; A b(a);)在对还没存在的对象赋值时会调用；赋值函数(e.g.A a; A b;b=a;)在对已经存在的对象赋值会调用。
8. 熟悉哪些设计模式？<http://blog.csdn.net/billcyj/article/details/79394582>
9. 线程之间的通信方式有哪些？答：锁机制、信号、信号量机制（信号量只能置一次初值，再P操作和V操作）
10. 使用全局变量需要加锁吗？不加锁会带来哪些后果？答：（和数据库的产生死锁题很像），如果多线程全部是读的话，就没必要加锁；一旦有一个线程可能去写，就要加读写锁了。以免读脏数据、丢失修改等情况发生。
11. 用过哪些锁？答：线程安全的单例模式中的互斥锁lock、数据库中读写数据时的读写锁。
12. 最后主要是项目了，问的比较细，就不详细说了。
13. 说说vector内部实现<http://blog.csdn.net/billcyj/article/details/78801834>
14. 问一下你项目中的技术
15. 简单介绍一下vector，list，map然后又问什么是红黑树，以及红黑树的自平衡问题，并且让你说一下它是如何保持平衡的<http://blog.csdn.net/billcyj/article/details/78996415>
16. 手写快速排序。答：在Partition函数中取第一个元素作为分区元素，先从右到左遍历再从左到右遍历，重复上一步骤直到left>=right，返回分区元素。再在快排函数中双重递归。

二、完美世界：

面试的时候更多也是会从语法方面问问题

客户端开发：

1. 一个空类，编译器会自动为这个类实现哪些成员函数？答：对于一个没有实例化的空类，编译器是不会给它生成任何函数的；当实例化一个空类后，编译器会根据需要生成相应的函数。如果我们在代码中需要生成一个A的实例，比如A a;编译器就会根据上面的实例，给类A生成构造函数和析构函数，由于三五法则，还会生成拷贝构造函数和赋值函数。
2. 智能指针，那几个指针的用途？答：智能指针类型有：shared_ptr、unique_ptr和weak_ptr，可令动态内存管理更为安全。对于一块没有任何用户使用它的内存，智能指针会自动释放它。shared_ptr允许多个指针指向同一个对象；每个shared_ptr都有一个的计数器（引用计数），能增能减，如果计数为0，shared_ptr的析构函数就会销毁对象。unique_ptr独占所指向的对象，不支持普通的拷贝或赋值，但可拷贝或赋值一个即将销毁的unique_ptr，如作为函数返回值。weak_ptr弱引用，指向shared_ptr所管理的对象。引入的一种智能指针来协助shared_ptr工作，它不共享指针，不能操作资源，只是一个观察者，但它可以使用一个成员函数lock()从被观测的shared_ptr获得一个可用的shared_ptr对象，从而操作资源。
3. 析构函数能否抛出异常？原因？（构造函数呢）答：析构函数不应该抛出异常！因为这样会导致程序过早结束或出现不明确的行为。当析构函数中的异常非抛不可，那么就必须要吧这种可能发生的异常完全封装在析构函数内部，也就是把try-catch写在析构函数内部。构造函数中尽量不要抛出异常，

能避免的就避免，因为可能导致析构函数不能被调用，所以可能会造成内存泄露或系统资源未被释放。

4. 构造、析构函数中能调用虚函数？<http://blog.csdn.net/billcyj/article/details/78998504>
5. 了解socket相关的知识？<http://blog.csdn.net/billcyj/article/details/79465733>
6. 你了解B/S和C/S吗？答：C/S全名Client/Server，称为客户/服务器模式，客户端需要安装专用的客户端软件；B/S全名Browser/Server，称为浏览器/服务器模式，客户机上只要安装一个浏览器，不用安装任何专门的软件，客户端零维护。非网页游戏基本上都是用C/S，网页游戏用B/S。B/S系统对服务器要求比较高，因为几乎所有运算都放在服务器端，而C/S却可以把很多运算放在客户端，从而减轻服务端压力；但网页游戏更新起来方便，升级服务器就行，不用管客户端，因为根本没有客户端。
7. 几个人组建一个局域网打游戏，几台主机是如何通信的？服务器在吗？答：局域网之间的通信不一定经过服务器，一般只会经过两台电脑和电脑之间连接的交换机或者路由器设备

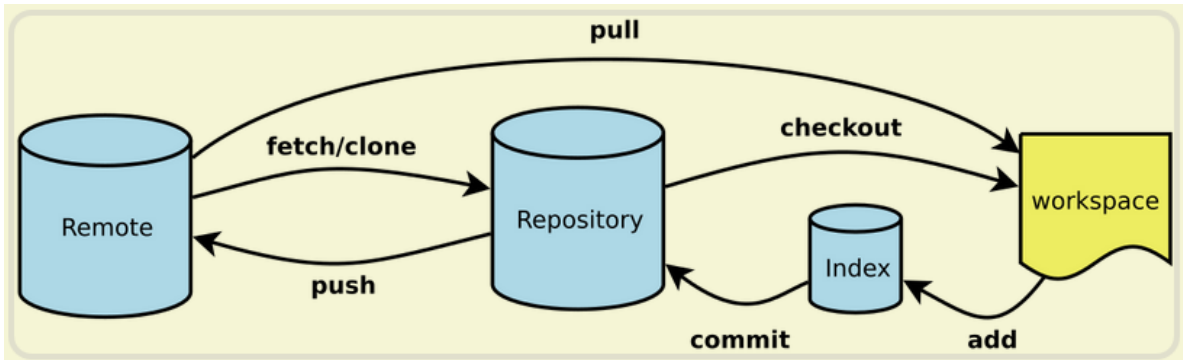
三、腾讯游戏：

天美J5一面：

1. 数组中前k大的数：法一先快排再取数， $O(n\log)$ ；法二用Partition， $O(n)$ ；法三用红黑树实现的set， $O(n\log k)$
2. 操作系统的PCB <http://blog.csdn.net/billcyj/article/details/79029074>
3. 渲染流程、用shader做了啥，面试者说：做了一些案例，比如高光效果，面试官就问blinnphong模型的细节：<http://blog.csdn.net/billcyj/article/details/79539536>
4. 别慌，面试的时候，去尽量想起来；项目的话，要说自己学到了什么；你往哪说他就往哪问，深不可测。

天美一面：

1. 红黑树<http://blog.csdn.net/billcyj/article/details/78996415>
2. 翻转二叉树<http://blog.csdn.net/billcyj/article/details/79212605>
3. TCP粘包，UDP丢包<http://blog.csdn.net/billcyj/article/details/79465733>
4. 多线程和协程的区别<http://blog.csdn.net/billcyj/article/details/79029074>
5. 工厂模式的应用<http://blog.csdn.net/billcyj/article/details/79394582>
6. Git的使用：<http://blog.csdn.net/billcyj/article/details/79447716>



fetch/clone不自动合并，pull会自动合并，但若有冲突，都需手动处理。

未知：牛客网

1. cpp如何实现多态<http://blog.csdn.net/billcyj/article/details/79523110>
2. cpp多继承会出现什么问题。答：二义性<http://mp.csdn.net/mdeditor/78998504>
3. 数组和队列区别。答：数组或链表 都可以实现 栈和队列
4. 栈溢出能做什么坏事。答：<http://blog.csdn.net/BillCYJ/article/details/78783741>
5. C#内存管理<http://blog.csdn.net/BillCYJ/article/details/79606982>
6. 检查整数的二进制表示中1的个数<http://blog.csdn.net/BillCYJ/article/details/79609871>
7. 一个线程包含什么。答：线程的实体包括程序、数据和TCB。
8. 快排是最快的吗？答：不是最快的，可以用空间换时间
9. 怎么光栅化一个三角面<http://blog.csdn.net/billcyj/article/details/79539536>
10. 虚拟内存和实存<http://blog.csdn.net/billcyj/article/details/79039828>
11. 了解其他引擎吗。答：COCOS适合做2D游戏，源码还开源；虚幻4适合做大型游戏，比如吃鸡或Steam上的大型游戏，但要收5%的利润；Unity适合基本全部游戏类型，但专业版需要花几万钱买。
12. 怎么学习。答：书、CSDN、知乎。
13. 项目闲聊

魔方：

1. 问项目
2. 数据结构在游戏中的应用<https://blog.csdn.net/BillCYJ/article/details/79790951>
3. 字节对齐<https://blog.csdn.net/billcyj/article/details/79383036>
4. 3D游戏场景管理<https://blog.csdn.net/billcyj/article/details/79791056>

天美J5二面：40分钟 没问项目

1. 排序数组和红黑树的查询对比。答：如果数组未排序，就按快排来说，时间复杂度也有 $O(n\log n)$ ；如果已排序，就用二分查找，时间复杂度为 $O(\log n)$ ，查找速度相当快，而红黑树平衡了查找性能和插

入性能，查找、插入、删除的时间复杂度都为 $O(\log n)$ ，但红黑树的查找速度因此而稍微逊色于二分，但红黑树在插入和删除这两方面的性能是非常优秀的。

2. AVL树和红黑树的对比。答：AVL树是严格平衡二叉搜索树（深度差最大为1），红黑树是弱平衡二叉搜索树（每条路的黑色节点数相等），导致AVL树在插入和删除时的旋转次数会大于RBT，所以，如果查找多插入删除少就用AVL，如果插入删除多就用RBT，另外，如果插入的数据分布均匀也可用AVL，分布不均就用RBT。由于红黑树中最长路径最多为最短路径的两倍，所以它的查找是由最低保证的，最差也是 $O(\log n)$ 。
3. A B线程通信方式：锁机制（互斥锁等）、信号量机制（PV操作）、信号（通知）
4. Mono实现跨平台机制：Mono为Unity提供了一个虚拟机运行环境，Unity的跨平台，就是通过Mono将**C#代码**编译成了通用**中间语言CIL**（Common Intermediate Language），然后Mono运行时利用 动态编译JIT 或 静态编译AOT 将CLI编译成**目标平台的原生代码**实现的。
5. Mono垃圾回收机制：一般，不需要我们去管GC，然而GC并不是实时性的，所以我们的资源使用完后，GC不会立刻执行清理来释放内存，因为GC不知道我们是否还会使用，所以它就等待，先处理其他的東西，一段时间后，发现这些东西不再用了，才执行清理，释放内存。补充：
<https://blog.csdn.net/billcyj/article/details/79606982>
6. 如何减少GC：字符串拼接，用StringBuilder中的Append方法。
7. IL是什么：又称CIL，即中间语言
8. 什么是 IL2CPP：把IL再弄回静态的CPP
9. Unity的实体-组件模式 以及优点缺点：对于一个实体来说，如果是对其能力进行扩展的，那就可将这样的特性挂在对象身上。优点：可视化和容易上手；缺点：组件提供的功能非常有限，但是扩展很方便。其他的缺点只有被坑过才印象深刻，比如滑动器。
10. Unity用到的设计模式：1. 单例模式：一个类只能有一个实例，单例非常重要经常用，比如游戏页面管理，现在我要关闭某个页面，页面上的很多子窗口也需要一起关闭。单例模式一般都是xxxController、xxxManager等。2. 中介者模式3. 观察者模式4. 工厂模式5. MVC模式
<https://blog.csdn.net/BillCYJ/article/details/79816418>
11. 骨骼动画原理,蒙皮原理：先摆好骨骼，骨骼之间有牵连关系，再蒙皮（也就是将皮肉绑到骨骼上，主要的工作就是分配权重），最后再做动画。
12. 哈希/红黑树对比：<https://blog.csdn.net/billcyj/article/details/78996415> 里的第6点，哈希表的应用：unorderedmap，红黑树的应用：set、map等

权衡因素有四个：查找速度、构造速度、数据量、内存使用、数据有序性

1. 反射用法：很少情况下是非用反射不可的，大多数情况下反射是为了提高程序的灵活性，避免将程序写死到代码里。因此一般框架中使用较多。因为框架要适用更多的情况。对一个未知的.dll，不使用反射根本不知道内部有些什么。
2. C++多态：**一个接口多种实现**。分为编译时的多态（即重载运算符和函数重载，**函数重载**的用处：墨盘的按钮设置动态监听时）和运行时的多态（即函数覆盖，涉及虚函数和基类指针，**函数覆盖**的用

处：可根据对象的动态类型，来决定调用哪个子类中的同名方法)

3. 函数隐藏：即父类和子类中的函数的同名覆盖，父类中该函数未加virtual 或 即使加了virtual但参数列表不同，这两种情况都会发生同名覆盖

4. 虚函数表：**编译器会为每个包含虚函数的类创建一个虚函数表**

5. 虚拟内存之页式：第4点<https://blog.csdn.net/billcyj/article/details/79039828>

天美J5 二面：

哈希表和红黑树：略

蒙皮：略

项目里有啥自己研究的东西：性能优化，UnityShader，渲染流程

性能优化：<https://blog.csdn.net/billcyj/article/details/79441193>

未知：面了57分钟

1. 先问了我项目，然后问项目具体实现，然后出了几道题：

2. 算法题：如何判断两个矩阵相交

假定矩形是用一对点表达的(minx, miny) (maxx, maxy)，那么两个矩形 为：

rect1{(minx1, miny1)(maxx1, maxy1)}

rect2{(minx2, miny2)(maxx2, maxy2)}相交的结果一定是个矩形，构成这个相交矩

形：minx = max(minx1, minx2)

miny = max(miny1, miny2)

maxx = min(maxx1, maxx2)

maxy = min(maxy1, maxy2)

1. 渲染流程：<https://blog.csdn.net/billcyj/article/details/79539536>

2. 场景管理：<https://blog.csdn.net/billcyj/article/details/79791056>

3. 语言题：略

4. 堆，栈：<https://blog.csdn.net/billcyj/article/details/78783741>

5. 哈希表，解决哈希表冲突的方法<https://blog.csdn.net/billcyj/article/details/79820902>

6. 设计模式：<https://blog.csdn.net/billcyj/article/details/79816418>

7. 虚函数：略

8. 面向对象的三个特性：封装（比如：属性）、继承、多态（静态和动态）

9. C++怎么实现多态：略

10. 判断链表是否有环：快慢指针

未知：

1. 最复杂的那个项目
2. TCP和UDP的差别：问到计算机网络的题目，都说计算机网络我们才开始学，我就简单地说一下我的理解：<https://blog.csdn.net/billcyj/article/details/79465733>
3. 快速排序，问了时间复杂度<https://blog.csdn.net/billcyj/article/details/78461001>

这个公式怎么来的：快排是将数组一分为二到底，所以要 $O(\log n)$ 次此操作，每次操作都会调用Partition函数遍历一遍数组，即 $O(n)$ 还问了最坏情况下是什么样的：是待排序元素已经排好序。第一趟经过 $n-1$ 次比较后第一个元素保持位置不变，并得到一个 $n-1$ 个元素的子序列；第二趟

1. 渲染流程：略
2. 值类型和引用类型的区别：略
3. 虚函数：在基类用**virtual**声明成员函数为虚函数，**虚函数是多态性的基础**，其调用的方式是**动态联编**（程序运行时才决定调用基类的还是子类的该函数）。**虚函数的作用**：可在派生类中重新定义此函数，为它**赋予新的功能**。
4. 变量声明与堆、栈内存的联系：<https://blog.csdn.net/billcyj/article/details/78783741>
5. MonoBehaviour的生命周期，OnEnable和awake谁先执行：
<https://blog.csdn.net/billcyj/article/details/79441370>

未知：

1. 项目：略
2. 渲染流程：略
3. 性能优化：略
4. UDP TCP：略
5. MVP矩阵：略

未知：

1. C#和C++的理解：略
2. 树的遍历<https://blog.csdn.net/billcyj/article/details/79212605>

未知：第一天 一面52min 考察个人的全方面基础 有个人的demo是比较好，因为一面和HR都玩了我做的游戏。很多话题也是通过这个游戏来进行开展的。

涉及到项目、C++、操作系统、设计模式、图形学，方方面面都涉及到了的。

1. 为什么vector超出了它的capacity要在别的地方申请一块2倍大小的内存，而不是在后面直接申请呢？vector的底层是数组，需要连续的地址，所以需要重新申请
2. map和hashmap的底层，对比红黑树和hash表<https://blog.csdn.net/billcyj/article/details/78996415> 第六点

3. 虚函数机制, 虚函数表相关的问题, 存储位置、表大小等等
<https://blog.csdn.net/BillCYJ/article/details/78998504> 第八点
4. 虚函数机制带来的开销有哪些? 通过指针或引用调用虚函数时, 跟普通函数调用相比, 会多一个根据虚函数指针找到虚拟函数表的操作
5. 操作系统在运行一个程序背后会有哪些流程? 操作系统先将其读入内存, 分配好存储地址, 然后执行程序, 如果是用的页式, 再利用页面调度策略、CPU轮转进行处理。
6. 进程状态, 调度转换: 略
7. 在程序中栈的作用: 略
8. 项目相关问题: 略
9. 一个游戏在你看来可以分为哪些模块? 略
10. 在这些模块中, 设计模式又可以做哪些的应用呢? <https://blog.csdn.net/billcyj/article/details/79816418>
11. 渲染管线中有哪些比较重要的变换? MVP+屏幕映射 (Unity自动完成屏幕映射), 投影矩阵是用于降维的, 从三维降到二维, 但**真正的投影发生在后面的屏幕映射中**。
12. 投影变换如何实现的? 通过投影矩阵将三维降为二维, 再在屏幕映射中, 通过齐次除法获得二维坐标。

同一个人, 第三天 二面 27min 考察个人的综合素质

进去就是做一道数学题(还要分类讨论的那种, 考察是否考虑了所有case)

1. 一来就是一道数学题, 两个圆相向而行, 分别位于(P1,P2)速度为(V1,V2), 半径为(R1,R2)求相遇时间? 考察分类讨论的能力, 略
2. 你玩哪些游戏? : 略
3. 在一个游戏中如果你感觉到不流畅可能是哪些原因造成的? 性能优化的四个方面
4. 在没有任何测量工具下, 在游戏中如何判断是由于网络原因还是性能原因(肯定面试官不会让看Ping (PING值是网络延迟值)和FPS (每秒的帧数))? 从外观: 顶点、三角面的数量, 从代码里看: shader里是不是用了循环或者sin或者cos等
5. 你了解哪些游戏引擎? 你最熟悉哪个? 略
6. 你说说你最了解的引擎的优点和缺点? 略
7. 你最近在学哪些技术? 应该回答学习unity相关更深入的知识
8. 为什么你会学c++11这些东西而不是学习unity相关更深入的知识呢? 略
9. 你觉得你的技术薄弱点在哪些地方? 不会Linux、游戏引擎底层知识不是很完善, 比如: 图形学, 我们大四才学, 我还打算看看《游戏引擎架构》这本书。

第四天, HR面 20多min

未知：PC游戏客户端开发 1个多小时

自我介绍：略

说一个你觉得最难的项目：略

你项目中的难点：略

C++的内存布局：略

内存对齐？实现原理？为什么对齐？<https://blog.csdn.net/BillCYJ/article/details/79936359>

怎样实现一个最小类或者结构体？不让内存对齐

看过哪些书？：略

对游戏感兴趣吗？：略

接下来的规划。：略

有了解游戏那边的实现原理和开发流程吗？：略

TCP和UDP在游戏通信中的选择和原因：略

为什么要采用UDP。UDP丢包了怎么办？：略

说说你知道的排序算法以及特点：略

知道图像压缩算法吗？知道哪些？deflate 和 lzma

最短寻路算法：略

写一个函数，计算n个斐波那契的和：略

有什么好的内存管理模式吗？页式、段式、段页式

智能指针，智能指针有什么缺点？讲了普通指针混用导致的引用计数问题、还有一些用法上的局限性（比如不能将智能指针对象作为STL的元素）

讲讲你除了观察者模式，还知道哪些设计模式：略

天美 PC游戏客户端开发

9月21日，初试(一面)，整个过程60分钟左右：

1.自我介绍，由于楼主主要搞游戏开发这方面的东西，一上来就介绍了我自己的游戏开发编程经历，面试官也是搞游戏开发的。

2.C++基础。面试官主要问了这些问题：

C++中的cast(4个cast,const_cast,dynamic,reinterpret,static)：用于数据类型的转换

虚析构函数;构造函数调用虚函数;虚继承；虚函数是怎么实现的等

<https://blog.csdn.net/BillCYJ/article/details/78998504>

3.算法。

链表逆置;

手写归并排序

一个很简单的智力题(一个桶装满10斤油，另外有一个能装3斤油的空桶和一个能装7斤油的空桶。试用这三个桶把10斤油平分两份)：略

4.光照在哪里计算(很多地方，比如世界空间，相机空间或者tangent space);

什么是法线贴图？蓝紫色的图，装的是法线信息，可以用来做一些凹凸纹理。

5.图形学方面的东西。略

6.面试官问了接下来的我的方向和发展问题:游戏客户端开发，联网大型游戏

9月23日上午，复试(二面)，整个过程25-30分钟左右。

1.自我介绍，二面面试官看起来很严厉，问题不难，但是感觉有一点压力。

2.C++基础。

智能指针(shared_ptr,unique_ptr,weak_ptr和auto_ptr)：略

C++内存模型：略

栈和堆的区别，栈和堆是怎么实现的：略

extern "C"的作用：为了能够正确实现C++代码调用其他C语言代码。加上extern "C"后，会指示编译器这部分代码按C语言的进行编译，而不是C++的

虚函数的覆写：略

函数重载：两个函数在同一个类中、两个函数的名字相同、两个函数的形参列表一定不同。（属于编译时的多态）

3.手写快排：略

4.详细介绍3D流水线。vertexshader(mvp矩阵)->光栅化阶段->pixelshader->(alpha测试，模板测试和深度测试和颜色混合等)

5.HDR：高动态范围图像，可以提供更多的动态范围和图像细节。

9月23日下午，HR(面试)，整个过程20分钟。

HR面试问了家庭问题，工作地点，兴趣爱好，成就等等。自己是什么情况，如实回答就好。

感受:一面和二面的面试官感觉很年轻，还是有一定的压力;由于楼主偏向游戏开发，面试官问了很多游戏开发的问题，很多基础比如计算机网络，操作系统，数据库都没问。

杂七杂八：

1.Tcp为什么慢，udp如何在应用层改进，像tcp一样可靠，或者tcp改进像udp一样快。（TCP流量控制、拥塞控制，UDP没有相关限制，不会缩小对方发送缓冲区，因此速度快。加快TCP的传输可以扩大缓冲区大小，或者减小流量控制的力度，让数据每次都刚好能完整发送一个包单位）

2.const <https://blog.csdn.net/BillCYJ/article/details/78879847>

3.100盏灯全关着，100个人，1个人开关自己整数倍的全部灯，当最后一个人完成操作，几盏灯亮着？考察因数，1，4，9，16，25。。。100这些数的因数都是奇数个（除非这个数是完全平方数,比如9,其中一对3X3只算一个因数,所以才能出现奇数个因数）

4.用一个基类的指针处理派生类，应该将基类的析构函数设置成虚函数还是将派生类的析构函数设置为虚函数呢？为什么？基类，为了能通过基类指针调用派生类的析构函数。

5.怎么保证TCP的可靠性？通过顺序编号seq和确认（ACK）来实现的，能保证数据正确到达目的地。TCP是面向字节流的，发送和接收对于此协议来说，没有什么头和尾，全部顺序投递；而UDP是面向包的，每次接收与发送都是一个数据块。UDP主要是面向不可靠连接的，不能保证数据正确到达目的地。

6.长连接：lient方与Server方先建立通讯连接，连接建立后不断开，然后再进行报文发送和接收，常用于点对点通讯。

6.短连接：Client方与Server每进行一次报文收发交易时才进行通讯连接，交易完毕后立即断开连接。此种方式常用于一点对多点通讯，比如多个Client连接一个Server.

7. TCP/IP如何保证长连接安全？TCP保持长连接，会用到KeepAlive心跳机制，发送端会在规定时间内，向接收端发送一个探测报文段，看对方是否还在连接状态，一般应该是两个小时吧没记错的话，至于怎么识别各自的连接，肯定是用套接字，通过套接字，你可以知道IP，端口等信息

8. c++ 引用和指针的区别？指针是一个变量，只不过这个变量存储的是一个地址，指向内存的一个存储单元；而引用跟原来的变量实质上是同一个东西，只不过是原变量的一个别名而已。

9. 八个体积相同的球，其中一个最重，用天平最少称几次？两次：第一次一边放三个，如果平衡，那么就把剩下的两个一边一个加上，根据下垂就可知道；如果不平衡，那么就把不平衡那边中的3个球选2个一边一个放到初始化过的天平上，就可以知道答案了。

四、网易互娱

网易引擎UI研究 28分钟

1.渲染流程：略

2.堆栈：<https://blog.csdn.net/billcyj/article/details/78783741>

3.链表和数组的区别：<https://blog.csdn.net/billcyj/article/details/79217877>

4.C#的list底层怎么实现的：<https://blog.csdn.net/billcyj/article/details/79790951>

5.还有UI组件位置的表示：RectTransform

6.相对坐标与绝对（又叫世界）坐标怎么转化、怎么实现加减：儿子的世界坐标=父亲的世界坐标+儿子的相对坐标。

7.4*4矩阵在游戏中可以做什么：需要4*4的不是为了表示“三维坐标”，而是为了表示“三维坐标的变换”，通过一个4*4矩阵整合了平移矩阵、旋转矩阵这两种操作，也就是可以同时完成平移、旋转。如果用3*3矩阵计算平移，很简单，就是加减一个差量；如果用3*3矩阵计算旋转时就需要和差化积，sin和cos等运算很慢，无法只用一个3*3矩阵，就完成平移和旋转，并且矩阵乘法在 GPU 很快。那么三维矢量怎么和4*4矩阵相乘呢？(x,y,z,w)这样由4个分量组成的矢量便被称为齐次坐标，齐次坐标(x,y,z,w)等价于三维坐标(x/w,y/w,z/w)，因此只要w分量的值是1，就可以表示三维矢量了。叶劲峰和陈嘉栋所答

9.面向对象特征及其怎么实现及其应用

继承：父类子类

多态：<https://blog.csdn.net/billcyj/article/details/79523110>

封装：举例：属性和私有字段

网易雷火盘古 游戏研发工程师 视频面 30分钟

1.自我介绍：略

2.进程调度的方式：1.先来先服务2.短作业优先3.时间片轮转4.优先级5.响应比6.多级反馈队列

3.线程同步的方法：互斥量、信号、信号量

4.死锁是怎样的：略

5.TCP、UDP区别：TCP是面向连接的、可靠的、面向字节流的（UDP面向报文）、交换数据之前需建立连接（UDP不是连接型的，所以资源消耗小、处理速度快）

6.为什么要字节对齐：为了效率<https://blog.csdn.net/BillCYJ/article/details/79936359>

7.虚函数会增加类的大小吗？会，每一个具有虚函数的类都有一个虚函数表VTABLE，里面按在类中声明的虚函数的顺序存放着虚函数的地址，这个虚函数表VTABLE是这个类的所有对象所共有的，也就是说无论用户声明了多少个类对象，只有一个VTABLE，还有一个VPTR虚函数指针，这个指针指向VTABLE的首地址，每个类的对象都有属于自己的指针。

8.图形学了解吗：略

9.stable_sort和sort区别：是否交换元素的前后顺序

10.异常安全讲一下：try、catch、throw捕获错误呈现给用户。我们采用try、throw、catch模块就能找到程序的错误，当try中达到异常的判定条件，程序将其throw，然后程序就从try模块跳转到catch，catch就是用于输出错误信息的。

11.手写strcpy <https://blog.csdn.net/billcyj/article/details/79529592>

手写strncpy：<https://blog.csdn.net/billcyj/article/details/79529592>

12.两个矩形怎么判断重叠：略

13.你有什么问题问我吗：略

网易雷火盘古 游戏研发工程师 30分钟

1. 手写单例模式：<https://blog.csdn.net/billcyj/article/details/79394582>

2. 三角形跟正方形碰撞检测：检查顶点在不在对方范围内

3. 如果有成千上万个三角形怎么办，然后问我怎么存储这些三角形：四叉树

4. 打乱数组的算法题：洗牌算法，主要是要保证效率，首先从所有元素中随机选取一个与第一个元素进行交换，然后在第二个之后选择一个元素与第二个交换，直到最后一个元素。这样能确保每个元素在每次交换中，在每个位置的概率都是1/n。也可以从后往前，代码相对简单些。

```
void random(int a[], int n) {    int index, tmp, i;    srand(time(NULL));    for(i=n-1; i>0; i--)    {        index=rand()%i; swap(a[i],a[index]);    } }
```

1. 如何实现智能指针：不用手动delete的指针，加一个计数器 构造时加一 析构时减一 判断是否为0，为零就自动delete

五、米哈游

米哈游 一面50分钟

1.如何用shader实现一张图的局部放大？使用抓屏通道, GrabPass。使用GrabPass {}后，可以用_GrabTexture访问截屏的纹理，后续的Pass通道使用这个抓屏再处理。

2.数字从1到n+2，删除其中两个数字，再将这个数组打乱，求被删除的两个数字。

法一：如果使用辅助内存，则开辟一个长度为n+2的数组，里面装0或1，初始值全为0，只要出现了数字，就将数字对应的0设为1，最后再遍历一次辅助内存即可。

法二：不使用辅助内存，就在原数组上做修改：某数字出现过就将 索引为该数字对应的数字变为负数，最后遍历一遍，值为正数的 对应的索引就是所缺的两个数字。注意：遇到最后两个数字，跳过即可，后期再处理。

3.哈希表的各种速度以及为什么：查找都是 $O(1)$ 的时间复杂度，因为哈希表存储的是键值对 $\langle \text{key}, p \rangle$ ，其查找的时间复杂度与元素数量多少无关，哈希表在查找元素时是**通过哈希函数H，根据哈希码值key来定位元素的位置p**，从而直接访问元素，因此，哈希表查找的时间复杂度为 $O(1)$

<https://blog.csdn.net/BillCYJ/article/details/79820902>

4.哈希表 建表时用求余法的话，余数应该怎么取，比如，输入是一个长度为m字符串？取小于m的最大的质数。

5.游戏性能优化：略

6.渲染流程：略

7.Unity能实现旋转的三种方式？1.欧拉角（陌生的你来到了成都，站在盐市口茫然四顾，想知道春熙路怎么走？告诉你春熙路的相对坐标，很具有操作性，这就是欧拉角）、2.四元数、3.3*3矩阵和4*4矩阵

8.如何求A数组里有的 而B数组里没有的 数字？不使用辅助内存，两个for循环遍历直接输出即可。

9.有红、蓝、白三种颜色的小球若干个，如何按照如下方式输出：红色的全部在左边、白色的全部在中间、蓝色的全部在右边：用前后指针，元素互换即可，遍历一遍即可。

10.项目：略

11.静态、动态批处理的具体流程：<https://blog.csdn.net/BillCYJ/article/details/79935983>

完美世界

服务端开发：

1.C++从源代码到可执行文件经过了哪些过程？

2.动态链接库和静态链接库的区别？

3.函数传参有几种方式？传引用和指针有什么区别呢？

4.删除单链表的某个节点？拓展：不允许判断头节点。

5.Linux熟悉吗？

6.C#你熟悉吗？

7.你感受最深的项目是？在项目中你觉得最难的部分是？

8.ISO网络模型有几层？每层是什么？分别作用？

9.TCP三次握手内容？四次挥手内容？TIME_WAIT状态的作用？

10.谈谈你对OO的理解？

11.设计模式了解过吗？举例几个？

12.手写一个C++单例Singleton模式？

13.知道什么是控制反转吗？简单讲一下？

14.如何只用减法实现乘法？

15.只遍历一遍找到数组中次大值。

网易雷火：重视unity 有游戏demo能加分

一面 45min

1.C++虚函数机制(虚函数表那一套)

2.C++析构函数可以申明为虚函数吗？构造函数可以申明为虚函数吗？

3.C++static的作用

4.项目，主要讨论之前我做的一个小游戏用的战斗框架FSM+GOAP。

5.谈谈你对MVC的理解

6.GOAP和FSM或者BT的区别？好处？坏处？

7.STL的map的实现

8.如果让你自己做一个俄罗斯方块小游戏，你会如何设计

9.如果让你实现word里面的撤销功能，你会如何设计

10.你所知道的排序算法、其时间复杂度和稳定性？

11.可以实习时间..

12.平时玩的游戏..

13.你希望做PC端游还是移动端手游？

二面 40min

1.进程有哪些调度算法，算法流程，各自的优缺点？

2.死锁

3.为什么要进行字节对齐？

4.手写代码strncpy，从源字符串拷贝n个字符到目标字符串(char* s1,char* s2,size_t n)

5.以上题目,如果考虑s1,s2区间有冲突怎么办？如果不开辟新内存怎么解决这个问题？

6.虚函数相关问题，以及带来的开销？

7.FSM的相关问题，结合项目相关的一些问题。

8.std::map的实现

9.std::sort的实现

10.如何判断两个轴平行的矩形是否相交？

11.在游戏中，假如在一个很大的游戏地图上，有障碍，如何实现近似的最短路径，也就是实现游戏中的自动寻路的效果。可以预处理，但是在每次寻路要求寻找结果的算法尽可能的快！

12.使用过脚本语言吗？

13.了解过渲染管线相关知识吗？

三面 30min

1.聊项目

2.unity中animator相关的问题，比如mask,layer,base machine,IK等等。

3.聊draw call以及相关优化问题

4.聊渲染相关问题，比如游戏中的阴影是如何实现的

5.写过 unity 的 shader 吗？

6.用过哪些unity 的组件、插件

7.使用的UGUI还是NGUI？

8.如果我们也给你offer，你会在哪些因素上考虑如何在腾讯offer和我们公司offer上取舍？

9.客户端和移动端有倾向吗？

网易互娱：

一面：

STL的allocator实现

对象的存储格式

堆栈区别

delete this的使用

虚函数调用

三次握手，四次挥手

八叉树和平衡树

单链表判环

01背包和多重背包

预防ddos攻击

怎么管理世界中玩家可见其他玩家的问题。。。。

二面：

港项目

半小时代码题，一道LeetCode代码手写

问了下为什么要做项目，什么时候开始做的，多少行了，有没有在github上。。。

介绍C++17的新特性（项目中用到了）

比较msvc和libcxx中unordered_xxx的实现

std::string的优化问题（讲了下拼接的几种优化方法和几种不同的STL对SSO的实现问题）

CWO实现的string为啥被废止（扯了下folly中string的三种情况）

有没有实习

平常喜欢玩什么游戏

有没有打算考研

讲解手写的代码

问了下参加了哪些面试和手里的offer情况。。。。

两个函数，不互相关联，一个写在一个循环中调用N次，一个分别写在两个循环中调用N次，问什么情况下第二种比第一种快（cache和分支预测？）

马走日字，走到特定点最少用了D步，问给定任意K是否存在一个路径走了K步也可以走到（小于肯定不行，K-D是偶数随便，要讨论的是K-D是奇数情况）

没做出。。。。。（我开始讲了BFS，A*，DP都被否了ORZ）

光子光速面试内容：

智能指针有哪些，介绍一下？

blinphone光照模型？

快排为什么为 $n \lg n$ ？ $\lg n$ 来自双重递归，也就是二分； n 来自 n 个数

虚函数表指针存在哪里？运行的时候才决定调用哪个函数，存在对象里，通过这个指针指向虚函数表

STL里面的sort的原理？

有哪几种光照反射？漫反射、高光反射、环境反射原理？

弱网络的时候，我那个宠物秘境的弹窗如何处理的？要么弹就会全部弹完，因为数据是在弹之前获取到的；要么就没获取到数据，就一个都不弹。

z-buffer是做啥的？深度缓冲是啥？也就是同一个点有前后两个物体，他是怎么用z-buffer来先渲染后面那个物体，再渲染前面这个物体的？

一个预制体是一次性加载进来？

墨盘的实现？索引器和字典的对比？

消消乐选关界面一次性加载完？还是？

Profiler看哪些参数？用Profiler怎么知道是哪一个函数造成的高内存？

深度测试是怎么决定渲染的先后顺序的？

overflow的解决方法？

RGB用half？那顶点用half？去查查，并且要考虑具体硬件和具体精度要求

unity加载和释放一个预制体的流程？

协程怎么实现伪多线程？和迭代器有什么关系？

AssetBundle是啥？咋用？

注意：

- 1.不要去抢话，等别人问完
- 2.回答任何问题都要考虑前提条件