我遇到的U3D客户端面试题By阿禾

1. 请描述一个洗牌算法。

答：Fisher-Yates算法。 先在数组中随机抽选一个元素，与最后的元素交换位置。 接着再从第一个至倒数第二个元素当中抽选一个元素，再与倒数第二个元素交换位置。 以此类推。

1. Seed如果相同，会导致结果可预测。 请写一个随机的函数，解决此问题。

答：不知道。 我想到的都是伪随机。

1. 你描述的Quick Sort是用递归函数。 能否想出不用递归函数的方式？

答：预先算出总共有多少层递归。 写两层循环实现原本的流程。

1. 手写单例模式。

答：就是写static instance。 但面试官还有追问如何避免其他地方透过new关键字实例化。 我没有回答出来，事后发现构造函数也可以设private。

1. 手写Queue的Enqueue与Dequeue函数。 面试官有提示用双链表。
2. List内部怎么实现？

答：我回答List是一个数组的封装。

1. List增加元素时，遇到元素超出数组范围怎么解决？

答：我回答内部会将数组的长度扩充为两倍。

1. 假如我有n个元素要往List内加，n很大，可能有1万个。 请问复杂度是多少？

答：我回答就是直接循环遍历一遍，为O（n），但面试官说是错的。

1. List与LinkedList两者在遍历时，哪一个比较快？

答：List比较快。 因为LinkedList每次检索元素时，都要遍历一次链表。 List因为是数组，是直接索取。

1. 手写河内塔（Hanoi Tower）的递归函数。 假设a、b、c分别代表三根柱子。 N是塔层数。 用Hanoi（a， b， c， n）来表示将塔从a搬至c的过程。 请写出递归函数。

答：其实中间还有一个步骤是要将第n层从a搬至c，这个要用另外的函数表示。

Hanoi(a, b, c, n)

Hanoi(a, c, b, n - 1)

Print（“将a的第n层移动至c”）

Hanoi(b, a, c, n - 1)

1. 请举出工作中遇到较复杂或困难的问题，并且如何解决。

答：我说我用Memory Profiler查图片内存占用，看哪些图片重复，或者需要调整压缩格式。 其实没有什么技术深度......

1. Field与Property的差别？

答：Field是一个类里面声明的变数，Property是getter与setter。

1. Stack是什么进，什么出？ Queue是什么进，什么出？

答：Stack后进先出，Queue后进后出。

1. RawImage与Image的差别？

答：RawImage是用Texture，Image是用Sprite。

1. Lua的\_\_index与\_\_newindex的差别？

答：\_\_index是在索引table不存在的key时触发，\_\_newindex是在向table不存在的 key赋值时触发。

1. Lua的闭包是什么？

答：闭包英文是closure，指的是让函数能够访问其作用域以外的变数的功能。

1. SkinnedMesh的内部原理是什么？

答：将每个顶点受到哪些骨骼牵动的信息，以及权重的信息都存在顶点信息上。 当执行骨骼动画时，这些顶点也就跟着动。

1. C#的List与ArrayList的差异？

答：List有泛型，ArrayList没有泛型。 ArrayList加入元素的函数是用object作为参数，所以会有装箱、拆箱的问题。 List则无。

1. 请用lua的元表实现继承。

答：用\_\_index来实现。 子类继承父类，父类有子类所没有的变数。 此关系可以用\_\_index来实现。

local person = {

name = “no name”

}

local worker = {}

setmetatable(worker, {\_\_index = person})

print(worker.name)

person有一个字段是name。 worker可以通过setmetatable以及\_\_index，让worker也能获取到person的name 。 类似继承的概念。

1. XLua实现跨语言调用的原理是什么？

答：用C#的P/Invoke功能，调用xlua.dll的函数，里面是与lua虚拟机交互的函数。

1. Lua的数据结构有哪些？

答：number、string、table、userdata。

1. AssetBundle的压缩格式有哪些？

答：Uncompressed：无压缩。 加载快，包体大。

LZ4：可以局部加载。 内存占用较少。 但无法局部释放。 必须全部一起释放。 包体中等。

LZMA：需要加载整个AssetBundle，内存峰值高。 包体最小。

1. Unity的meta文件的GUID、localID、fileID是什么？

答：GUID：用来识别Asset的hash，每个Asset各自有独特的GUID。

localID：每个Asset内可能有多个组件，用localID来识别，是这个Asset之下的局部ID。

fileID：就是localID。

1. Local space怎么转成world space？

答：用齐次矩阵。

1. Normal Map为什么蓝蓝的？

答：因为在储存normal vector3的信息时，会将z设为1，然后以RGBA的形式存在贴图上。 因为B = 1，所以偏蓝色。

1. 常见的数据结构有哪些？

答：数组、链表、跳表、栈、队列、树、堆、哈希表、图

1. 如何判断一个点在一个多边形内？

答：假设该点为P。 如果多边形是凸多边形。 沿着逆时针方向，计算出每个边的矢量，并且算出每个向量起点至P的矢量，将两者做外积。 若每个边的外积阶为出纸面，则可判断P在凸多边形内部。 若不为凸多边形，则将多边形分割成多个凸多边形，再逐一判断。

1. 请描述render pipeline。

答：应用阶段：cpu汇整数据传给gpu。

几何阶段：vertex shader、坐标变换、裁剪

光栅化阶段：片元组装、三角形遍历、fragment shader、对片元进行各种测试（Scissor Test、Alpha Test、Stencil Test 、Depth Test）

1. 请问齐次矩阵是做什么用的？

答：将一个点的集合集体进行平移、剪力形变（Shearing）、缩放、旋转。

1. 请问Unity的shader内，有一个MVP是什么？

答：投影用的。

1. 请问Canvas的底层原理是？

答：待查。

1. 请问DrawCall是什么？ 有什么优化方式？

答：cpu下达给gpu绘图的一道指令。 Dynamic batching，合并网格点，共享材质球。 Culling，将看不见的面删掉。

1. 请描述计算机内存的架构？

答：待查。

1. 请问GC回收的底层原理是？

答：Garbage Collector在自己的线程上运行，在一些预定的时机点执行回收，也可以手动呼叫GC. Collect()。 首先会建立起整个应用程序的引用关系的graph，再检查哪些是无用的对象。 接着将无用的对象回收，内存会腾出一些空间，还会进行内存空间的压缩。

1. 请问Unity的物理模拟的底层原理是？

答：待查。

1. 请问GPU成像前需要接收哪些信息？

答：vertex、vertex color、normal vector、uv。

1. 请问C#的Dictionary的底层原理？

答：用的是hash table。 将key带入一个hash function，算出value的位置。

1. 请问C#的Dictionary遇到hash冲突时，怎么处理？

答：用的是单链表，相同hash的key放在同一个bucket，并且用单链表串接。 取得该key的value时，会遍历单链表，直到找到key相同的元素并返回。