Unity面试题00

Date:2020/07/27

1. 一个二叉树，定义有N为有叶子结点的结点，L为叶子结点，那么给定一串前序遍历过的字符串假设（NNLLL），求这棵树高度。例如：输入为“NNLLL”，输出为3
2. 反转链表、计算二叉树高度
3. 递归反转链表
4. 全排列的下一个数组 follow up时间复杂度变化
5. 以 前缀数 这种数据结构为基础 对树上的节点进行拓扑排序
6. Builder Design Pattern in Modern C++

<https://dzone.com/articles/builder-design-pattern-in-modern-c>

1. GameLoop

<https://teamavocado.co/core-loop/>

1. Unity调用安卓Android的Toast

<https://www.cnblogs.com/guxin/p/unity-call-android-toast.html>

1. Ytb算法 <https://www.youtube.com/watch?v=FBJKwjTwNTo>
2. 水果忍者AI的源代码 <https://github.com/LupascuAndrei/fruit_ninja_ai>
3. 如何对抽象类进行单元测试？ ​

<https://enterprisecraftsmanship.com/posts/how-to-unit-test-an-abstract-class/>​​​

1. Unity教程之-Unity3d中针对Android Apk的签名验证（C#实现）

<https://www.cnblogs.com/porter/p/6961629.html>

1. Built-in shader 转 URP shader API变化速查手册

<https://teofilobd.github.io/unity/shaders/urp/graphics/2020/05/18/From-Built-in-to-URP.html>

1. UUG Online直播回放：DOTS从原理到应用-雨松MOMO

<https://www.bilibili.com/video/BV1sD4y1Q7an>

1. 自学计算机科学

<https://github.com/keithnull/TeachYourselfCS-CN/blob/master/TeachYourselfCS-CN.md>

1. 为什么这么设计系列文章

<https://draven.co/whys-the-design/>

1. Unity中Button组件对Enter及Space键的响应

<https://indienova.com/u/yangblood/blogread/26037>

1. 更快的整数解析C++

<https://kholdstare.github.io/technical/2020/05/26/faster-integer-parsing.html>

1. Renderdoc1.8 支持 vulkan shader 调试

<https://github.com/baldurk/renderdoc/issues/1908>

1. 算法/竞赛 <https://github.com/Xunzhuo/OI_Sharing>
2. 使用Unity实现Voxelizer网格变换

<https://zhuanlan.zhihu.com/p/141408477>

1. 游戏编程模式（原版）

<http://gameprogrammingpatterns.com/>

1. Nsight <https://developer.nvidia.com/nsight-visual-studio-edition>
2. Unity 保证设备唯一标识符

<https://blog.csdn.net/Ling_SevoL_Y/article/details/88839157>

1. A Universe of Sorts

[https://bollu.github.io/#best-practices-for-array-indexing](https://bollu.github.io/" \l "best-practices-for-array-indexing)

1. Unity Sprite Dicing

<https://github.com/Elringus/SpriteDicing>

1. 入门：

Code: The Hidden Language of Computer Hardware and Software

操作系统结构基础

Operating System Concepts

Modern Operating Systems

操作系统稍难的

Computer Systems: A Programmer's Perspective

看完了操作系统后推荐看的

Computer Architecture：A Quantitative Approach

1. C ++ RPC库–基于Protobuf和独立于网络的核心库

<https://github.com/IronsDu/gayrpc>

1. Unity 开源LOD

<https://github.com/Unity-Technologies/UnityMeshSimplifier>

1. SIGGRAPH 2019四大精彩演讲回顾

<https://mp.weixin.qq.com/s/-4b-JvVH4WNpXhQGbeMF1g>

1. 这个高逼格场景居然不用网格模型!?Unity无模型研究-第一期

<https://www.bilibili.com/video/av82236765/?share_medium=web&share_source=weibo&bbid=3CEBCA0A-B0CE-418F-8095-87C5C79F26A2155804infoc&ts=1578235867961>

1. 每个贴图指定不同mipmap减低纹理带宽

<https://www.xuanyusong.com/archives/4699>

1. 游戏编程入门指南

<https://indienova.com/indie-game-development/game-programming-for-beginners/>

1. UE4 实时虚拟角色材质篇之 Skin Material

<https://indienova.com/indie-game-development/unreal-engine-4-skin-material-1/>

1. C# 简述Action与function

<https://www.cnblogs.com/Xue-Dragon/p/10578190.html>

1. Unity后台下载

<https://github.com/Unity-Technologies/BackgroundDownload>

1. 遗传算法和神经网络 AI驾驶

<https://github.com/martenq1337/COPS_AI>

1. 如何控制C#Socket的连接超时时间

<https://blog.csdn.net/zb872676223/article/details/17379005>

[https://www.it1352.com/423683.html#0-tsina-1-68841-397232819ff9a47a7b7e80a40613cfe1](https://www.it1352.com/423683.html" \l "0-tsina-1-68841-397232819ff9a47a7b7e80a40613cfe1)

1. Unity3d 使用 RSA和DES 加密网络数据包

<https://blog.csdn.net/huutu/article/details/51533066>

1. 算法题 <http://disscode.com/>
2. 《塞尔达传说：旷野之息》风格的水着色器

<https://mp.weixin.qq.com/s/7bvBBldIH3J7YrhRRsevPA>

1. 中国科学技术大学课程资源

<https://github.com/USTC-Resource/USTC-Course>

1. Coding技能树搭建

<https://weibo.com/ttarticle/p/show?id=2309404378055274546207>

1. C#反射使用时注意BindingFlags的用法

<https://blog.csdn.net/weixin_38109688/article/details/80147535>

1. Unity中利用反射遍历类或者结构体中的每一个字段属性 、类型 、值

<https://gameinstitute.qq.com/community/detail/124359>

1. 空间软体 <https://darkblueheaven.com/spatialsoftware/>
2. 解析SQL <https://tomassetti.me/parsing-sql/>

Date:2020/07/10

# Unity相关

1. Q：GC

A： <https://blog.csdn.net/wangjiangrong/article/details/89011349>

<https://www.cnblogs.com/wangqiang3311/p/10280000.html>

1. Q：如何做UI层裁剪

A： 用shader做区域裁剪；用stencilbuffer；多加一个相机render texture

1. Q：UI层drawcall，不用Z缓冲，既能保证渲染顺序，又能一个drawcall实现

A：

1. Q：Alpha混合公式

A：

1. Q：脚本生命周期？两个脚本awake什么时候调用

A： <https://blog.csdn.net/iFasWind/article/details/81094938?utm_source=blogxgwz2>

1. Q：Unity资源优化

A：

1. Q：循环ScrollView怎么实现

A：

1. Q：分帧加载、缓冲列表

A：

1. Q：NGUI怎么优化

A：

1. Q：Panel上的static勾了有什么效果？什么情况勾选？

A：<https://blog.csdn.net/cp1001/article/details/25334793>

1. Q： 特效层级穿插怎么控制？

A：RenderQueue；short order

1. Q：sharedmaterial和material区别？什么情况下使用

A：<https://blog.csdn.net/Fivelin/article/details/79862854>

1. Q：ECS项目架构图，如何搭建？如果怪物有血条，怎么修改

A：

1. Q：NGUI怎么减少drawcall

A：

1. Q：假设特效粒子都优化了，脚本性能都优化了,ui的dc还是很高怎么办

A：

1. Q：Ecs思想？怎么理解数据和状态？Unity里有类似ecs里的机制吗？用Ecs实现和Unity机制实现有什么区别。

A：

1. Q：假设有资源加载在其他线程上，这个线程如何更新资源加载到Unity的UI？需要加锁吗

A：

# C#相关

1. Q：C#修饰符有什么，区别？internal怎么用

A：:

1. Q：Linkedlist和List区别

A：

1. Q：Set和Dictionary怎么去重的？如果有两个gameobject，希望他们会被set去重怎么做？重写哈希函数。

A： <https://blog.csdn.net/a_dev/article/details/79424513>

<https://www.zhihu.com/question/50903503?sort=created>

<https://www.cnblogs.com/alibai/p/3520179.html>

<https://www.cnblogs.com/junjieok/p/5705586.html>

<https://blog.csdn.net/BillCYJ/article/details/90519496>

<https://docs.microsoft.com/zh-cn/previous-versions/ms173147(v=vs.80)>

<http://www.manongjc.com/detail/15-hnccavqukdflpgw.html>

<https://www.cnblogs.com/mq0036/p/9166893.html>

1. Q：委托是什么？

A：

1. Q：Struct和Class有什么区别？值传递和引用传递区别？

A：<https://www.cnblogs.com/jjg0519/p/10341010.html>

<https://blog.csdn.net/lsl277879661/article/details/56481650>

1. Q： 重载和重写的区别

A：<https://www.cnblogs.com/hhx626/p/6010464.html>

1. Q： CLR加载机制

A：

1. Q：反射是什么？原理？自己在C++中如何实现反射？

A：

1. Q：C#和C++区别？C#会内存泄漏吗？托管资源会泄露吗？

A：

# C++相关

1. Q：C++多态、虚表

A:

# Lua相关

1. Q：Lua多态的实现

A：

1. Q：Lua实现面向对象

A:

1. Q：Lua调用C#函数原理

A：

# 渲染相关

1. Q： 渲染各种矩阵变换，卡通渲染，模型轮廓怎么画（法线外扩）

A：

1. Q：水的模拟（菲涅尔效应,CubeMap,grab pass）

A：

# 网络相关

1. Q：优化网络层，UDP分片机制

A：

1. Q：网页浏览过程

A：

1. Q：帧同步、状态同步区别、介绍、优化点（浮点、定点）

A：

1. Q：帧同步怎么防作弊？需要仲裁服怎么做？怎样用定点数代替浮点数

A：

1. Q：断线重连如何不用从头开始跑

A：

1. Q：状态同步怎么优化手感？预测出现失败怎么办？

A：

# 算法相关

1. Q：找到链表倒数第K个值

A：

1. Q：1到n个数（不重复），选取n-1个数，怎么知道没选哪个

A：

1. Q：判断单链表有环

A：

1. Q：PVS算法 （核心思想、定义什么数据结构存储viewpoint、描述pvs的运作流程）

A：<https://blog.csdn.net/u012501320/article/details/25081933>

1. Q：在项目中所有函数头和尾hook，在头尾记录时间戳，调用所有函数一遍，输出日志。根据日志，找出所有函数中运行时间最长的前三（运行新跟那个最差的前三）。用两个栈，一个站类似于括号匹配，另一个栈记录函数调用关系。

A：

# 设计模式相关

1. Q：实现一个观察者和消息队列更新UI的框架

Date:2020/07/09

#Unity相关

1. Q:场景管理（10000人在场景中，插入一个新人）

A:

1. Q：C#字典 contains key时间复杂度

A：

1. Q：NGUI Drawcall

A：

1. Q：Unity Shadermap 实现

A：

#算法相关

1. Q：哈希表实现

A：

Date:2020/07/07

#Unity相关

1. Q：UGUI适配

A：

1. Q：Unity生命周期

A：

#网络相关

1. Q：Proto协议关键字

A：<https://blog.csdn.net/code_style/article/details/82751720>

<https://zhuanlan.zhihu.com/p/69932332>

1. Q：proto和json区别

A：

1. Q：protobuffer怎么用，序列化和反序列化

A：

#算法相关

1. Q：翻转Int数

A：

1. Q：A\*算法

A：

Date:2020/07/06

#Unity相关

1. Q：UGUI显示3D粒子

A：2017：<https://blog.csdn.net/weixin_45725888/article/details/104340114>

2019：<https://blog.csdn.net/ak47007tiger/article/details/93538164>

1. Q：AB包加载卸载 若从streaming加载

A：Unity2017 <https://www.cnblogs.com/Roz-001/p/11237198.html>

从StreamingAsssets文件夹加载 <https://www.cnblogs.com/zouqiang/p/9053426.html>

Unity2019 <https://blog.csdn.net/a1191835397/article/details/96989037>

1. Q：动静态合批

A：Unity合批 <https://gameinstitute.qq.com/community/detail/114323>

Batch <https://gameinstitute.qq.com/community/detail/113040>

# Unity 优化之25种在渲染中不能合批的原因

# <https://github.com/Unity-Technologies/BatchBreakingCause>

# <https://www.jianshu.com/p/43efe4d35ccd>

1. Q：渲染层级由谁决定

A：<https://zhuanlan.zhihu.com/p/93107721>

<https://blog.csdn.net/qq_34444468/article/details/79148799>

<https://blog.csdn.net/yuanyuanluoluo/article/details/78373781>

1. Q：优化方向

A:

1. Q:UGUI NGUI重绘原理

A:UGUI Rebuild <https://blog.csdn.net/qq_39574690/article/details/104446897>

<https://www.jianshu.com/p/eb1087030665>

<https://zhuanlan.zhihu.com/p/46208753>

<http://www.luyixian.cn/news_show_377085.aspx>

NGUI Rebuild

1. Q：Unity 判断敌人在玩家方位（点乘，叉乘）

A：<https://blog.csdn.net/yupu56/article/details/53609028>

<https://www.jianshu.com/p/df8ac577e5da>

<https://www.cnblogs.com/lifangti/p/unity3D.html>

# 渲染相关

1. Q：裁剪

A：

1. Q：模版测试

A：

# Lua相关

1. Q：lua一般问如何做面向对象 如何做继承

A：

1. Q：lua如何与宿主语言交互

A：

1. Q：lua GC

A：

# C++相关

1. Q：静态链接动态链接原理

A：

1. Q：内存

A：

1. Q：哈希

A：

1. Q：STL

A：

# 网络

1. Q：帧同步

A：