一．什么是渲染管道？

是指在显示器上为了显示出图像而经过的一系列必要操作。

渲染管道中的很多步骤，都要将几何物体从一个坐标系中变换到另一个坐标系中去。

主要步骤有：

本地坐标->视图坐标->背面裁剪->光照->裁剪->投影->视图变换->光栅化。

二．如何优化内存？

有很多种方式，例如

1.压缩自带类库；

2.将暂时不用的以后还需要使用的物体隐藏起来而不是直接Destroy掉；

3.释放AssetBundle占用的资源；

4.降低模型的片面数，降低模型的骨骼数量，降低贴图的大小；

5.使用光照贴图，使用多层次细节(LOD)，使用着色器(Shader)，使用预设(Prefab)。

三、动态加载资源的方式？(有时候也问区别，具体请百度)

1.Resources.Load();

2.AssetBundle

四：什么是协同程序？

在主线程运行的同时开启另一段逻辑处理，来协助当前程序的执行，协程很像多线程，但是不是多线程，Unity的协程实在每帧结束之后去检测yield的条件是否满足。

五：Unity3d中的碰撞器和触发器的区别？

碰撞器是触发器的载体，而触发器只是碰撞器身上的一个属性。当Is Trigger=false时，碰撞器根据物理引擎引发碰撞，产生碰撞的效果，可以调用OnCollisionEnter/Stay/Exit函数；当Is Trigger=true时，碰撞器被物理引擎所忽略，没有碰撞效果，可以调用OnTriggerEnter/Stay/Exit函数。如果既要检测到物体的接触又不想让碰撞检测影响物体移动或要检测一个物件是否经过空间中的某个区域这时就可以用到触发器

六：物体发生碰撞的必要条件？

两个物体都必须带有碰撞器（Collider），其中一个物体还必须带有Rigidbody刚体，而且必须是运动的物体带有Rigidbody脚本才能检测到碰撞。

七：请简述ArrayList和List的主要区别？

####ArrayList存在不安全类型（ArrayList会把所有插入其中的数据都当做Object来处理） 装箱拆箱的操作（费时） List是接口，ArrayList是一个实现了该接口的类，可以被实例化

八：如何安全的在不同工程间安全地迁移asset数据？三种方法

1.将Assets目录和Library目录一起迁移

2.导出包，export Package

3.用unity自带的assets Server功能

九：OnEnable、Awake、Start运行时的发生顺序？哪些可能在同一个对象周期中反复的发生

Awake –>OnEnable->Start，OnEnable在同一周期中可以反复地发生。

十：MeshRender中material和sharedmaterial的区别？

修改sharedMaterial将改变所有物体使用这个材质的外观，并且也改变储存在工程里的材质设置。不推荐修改由sharedMaterial返回的材质。如果你想修改渲染器的材质，使用material替代。

十一：Unity提供了几种光源，分别是什么

四种。

平行光：Directional Light

点光源：Point Light

聚光灯：Spot Light

区域光源：Area Light

十二：简述一下对象池，你觉得在FPS里哪些东西适合使用对象池

对象池就存放需要被反复调用资源的一个空间，当一个对象回大量生成的时候如果每次都销毁创建会很费时间，通过对象池把暂时不用的对象放到一个池中（也就是一个集合），当下次要重新生成这个对象的时候先去池中查找一下是否有可用的对象，如果有的话就直接拿出来使用，不需要再创建，如果池中没有可用的对象，才需要重新创建，利用空间换时间来达到游戏的高速运行效果，在FPS游戏中要常被大量复制的对象包括子弹，敌人，粒子等

十三：CharacterController和Rigidbody的区别

Rigidbody具有完全真实物理的特性，Unity中物理系统最基本的一个组件，包含了常用的物理特性，而CharacterController可以说是受限的的Rigidbody，具有一定的物理效果但不是完全真实的，是Unity为了使开发者能方便的开发第一人称视角的游戏而封装的一个组件

十四：简述prefab的用处

在游戏运行时实例化，prefab相当于一个模板，对你已经有的素材、脚本、参数做一个默认的配置，以便于以后的修改，同时prefab打包的内容简化了导出的操作，便于团队的交流。

十五：请简述sealed关键字用在类声明时与函数声明时的作用

sealed修饰的类为密封类，类声明时可防止其他类继承此类，在方法中声明则可防止派生类重写此方法。

十六：请简述private，public，protected，internal的区别

public：对任何类和成员都公开，无限制访问

private：仅对该类公开

protected：对该类和其派生类公开

internal：只能在包含该类的程序集中访问该类

十七：使用Unity3d实现2d游戏，有几种方式？

1. 使用本身的GUI，在Unity4.6以后出现的UGUI

2.把摄像机的Projection(投影)值调为Orthographic(正交投影)，不考虑z轴；

3.使用2d插件，如：2DToolKit，和NGUI

十八：在物体发生碰撞的整个过程中，有几个阶段，分别列出对应的函数

三个阶段，1.OnCollisionEnter 2.OnCollisionStay 3.OnCollisionExit

十九：Unity3d的物理引擎中，有几种施加力的方式，分别描述出来

rigidbody.AddForce/AddForceAtPosition，都在rigidbody系列函数中。大家可以自己去查看一下rigidbody的API

二十：什么叫做链条关节？

Hinge Joint，可以模拟两个物体间用一根链条连接在一起的情况，能保持两个物体在一个固定距离内部相互移动而不产生作用力，但是达到固定距离后就会产生拉力。

二十一：物体自身旋转使用的函数？

Transform.Rotate()

二十二：Unity3d提供了一个用于保存和读取数据的类(PlayerPrefs)，请列出保存和读取整形数据的函数

PlayerPrefs.SetInt() PlayerPrefs.GetInt()

二十三：Unity3d脚本从唤醒到销毁有着一套比较完整的生命周期，请列出系统自带的几个重要的方法。

Awake——>OnEnable–>Start——>Update——>FixedUpdate——>LateUpdate——>OnGUI——>OnDisable——>OnDestroy

二十四：物理更新一般放在哪个系统函数里？

FixedUpdate，每固定帧绘制时执行一次，和Update不同的是FixedUpdate是渲染帧执行，如果你的渲染效率低下的时候FixedUpdate调用次数就会跟着下降。FixedUpdate比较适用于物理引擎的计算，因为是跟每帧渲染有关。Update就比较适合做控制。

二十五：在场景中放置多个Camera并同时处于活动状态会发生什么？

游戏界面可以看到很多摄像机的混合。

二十六：如何销毁一个UnityEngine.Object及其子类？

使用Destroy()方法;

二十七：请描述为什么Unity3d中会发生在组件上出现数据丢失的情况

一般是组件上绑定的物体对象被删除了

二十八：LOD是什么，优缺点是什么？

LOD(Level of detail)多层次细节，是最常用的游戏优化技术。它按照模型的位置和重要程度决定物体渲染的资源分配，降低非重要物体的面数和细节度，从而获得高效率的渲染运算。缺点是增加了内存。

二十九：MipMap是什么，作用？

MipMapping：在三维计算机图形的贴图渲染中有常用的技术，为加快渲染进度和减少图像锯齿，贴图被处理成由一系列被预先计算和优化过的图片组成的文件，这样的贴图被称为MipMap。

三十：请描述Interface与抽象类之间的不同

抽象类表示该类中可能已经有一些方法的具体定义，但接口就是公公只能定义各个方法的界面 ，不能具体的实现代码在成员方法中。类是子类用来继承的，当父类已经有实际功能的方法时该方法在子类中可以不必实现，直接引用父类的方法，子类也可以重写该父类的方法。实现接口的时候必须要实现接口中所有的方法，不能遗漏任何一个。

三十一：.Net与Mono的关系？

mono是.net的一个开源跨平台工具，就类似java虚拟机，java本身不是跨平台语言，但运行在虚拟机上就能够实现了跨平台。.net只能在windows下运行，mono可以实现跨平台跑，可以运行于linux，Unix，Mac OS等。

三十二：简述Unity3D支持的作为脚本的语言的名称

Unity的脚本语言基于Mono的.Net平台上运行，可以使用.NET库，这也为XML、数据库、正则表达式等问题提供了很好的解决方案。Unity里的脚本都会经过编译，他们的运行速度也很快。这三种语言实际上的功能和运行速度是一样的，区别主要体现在语言特性上。JavaScript、 C#、Boo

三十三：U3D中用于记录节点空间几何信息的组件名称，及其父类名称

Transform 父类是 Component

三十四：向量的点乘、叉乘以及归一化的意义？

1.点乘描述了两个向量的相似程度，结果越大两向量越相似，还可表示投影

2.叉乘得到的向量垂直于原来的两个向量

3.标准化向量：用在只关系方向，不关心大小的时候

三十五：为何大家都在移动设备上寻求U3D原生GUI的替代方案

不美观，OnGUI很耗费时间，效率不高，使用不方便

三十六：请简述如何在不同分辨率下保持UI的一致性

NGUI很好的解决了这一点，屏幕分辨率的自适应性，原理就是计算出屏幕的宽高比跟原来的预设的屏幕分辨率求出一个对比值，然后修改摄像机的size。UGUI通过锚点和中心点和分辨率也解决这个问题

三十七：什么是LightMap？

LightMap:就是指在三维软件里实现打好光，然后渲染把场景各表面的光照输出到贴图上，最后又通过引擎贴到场景上，这样就使物体有了光照的感觉。

三十八：Unity和cocos2d的区别

1. Unity3D支持C#、javascript等，cocos2d-x 支持c++、Html5、Lua等。

2. cocos2d 开源 并且免费

3. Unity3D支持iOS、Android、Flash、Windows、Mac、Wii等平台的游戏开发，cocos2d-x支持iOS、Android、WP等。

三十九：C#和C++的区别？

简单的说：C# 与C++ 比较的话，最重要的特性就是C# 是一种完全面向对象的语言，而C++ 不是，另外C# 是基于IL 中间语言和.NET Framework CLR 的，在可移植性，可维护性和强壮性都比C++ 有很大的改进。C# 的设计目标是用来开发快速稳定可扩展的应用程序，当然也可以通过Interop 和Pinvoke 完成一些底层操作。更详细的区别大家可以参考这里

四十：结构体和类有何区别？

结构体是一种值类型，而类是引用类型。（值类型、引用类型是根据数据存储的角度来分的）就是值类型用于存储数据的值，引用类型用于存储对实际数据的引用。那么结构体就是当成值来使用的，类则通过引用来对实际数据操作

四十一：ref参数和out参数是什么？有什么区别？

ref和out参数的效果一样，都是通过关键字找到定义在主函数里面的变量的内存地址，并通过方法体内的语法改变它的大小。不同点就是输出参数必须对参数进行初始化。ref必须初始化，out 参数必须在函数里赋值。ref参数是引用，out参数为输出参数。

四十二：C#的委托是什么？有何用处？

委托类似于一种安全的指针引用，在使用它时是当做类来看待而不是一个方法，相当于对一组方法的列表的引用。用处：使用委托使程序员可以将方法引用封装在委托对象内。然后可以将该委托对象传递给可调用所引用方法的代码，而不必在编译时知道将调用哪个方法。与C或C++中的函数指针不同，委托是面向对象，而且是类型安全的。

四十三：C#中的排序方式有哪些？

选择排序，冒泡排序，快速排序，插入排序，希尔排序，归并排序

四十四：射线检测碰撞物的原理是？

射线是3D世界中一个点向一个方向发射的一条无终点的线，在发射轨迹中与其他物体发生碰撞时，它将停止发射 。

四十五：Unity中，照相机的Clipping Planes的作用是什么？调整Near、Fare两个值时，应该注意什么？

剪裁平面 。从相机到开始渲染和停止渲染之间的距离。

四十六：如何让已经存在的GameObject在LoadLevel后不被卸载掉？

void Awake()

{

DontDestroyOnLoad(transform.gameObject);

}

四十七：请简述GC（垃圾回收）产生的原因，并描述如何避免？

GC回收堆上的内存

避免：1.减少new产生对象的次数

2.使用公用的对象（静态成员）

3.将String换为StringBuilder

四十八：反射的实现原理？

审查元数据并收集关于它的类型信息的能力。实现原理：在运行时根据程序集及其中的类型得到元数据。下面是实现步骤：

1. 导入using System.Reflection;

2. Assembly.Load(“程序集”)加载程序集,返回类型是一个Assembly

3. 得到程序集中所有类的名称

foreach (Type type in assembly.GetTypes())

{

string t = type.Name;

}

4. Type type = assembly.GetType(“程序集.类名”);获取当前类的类型

5. Activator.CreateInstance(type); 创建此类型实例

6. MethodInfo mInfo = type.GetMethod(“方法名”);获取当前方法

7. m.Info.Invoke(null,方法参数);

四十九：简述四元数的作用，四元数对欧拉角的优点？

四元数用于表示旋转

相对欧拉角的优点：

1.能进行增量旋转

2.避免万向锁

3.给定方位的表达方式有两种，互为负（欧拉角有无数种表达方式）

五十：移动相机动作在哪个函数里，为什么在这个函数里？

LateUpdate，是在所有的update结束后才调用，比较适合用于命令脚本的执行。官网上例子是摄像机的跟随，都是所有的update操作完才进行摄像机的跟进，不然就有可能出现摄像机已经推进了，但是视角里还未有角色的空帧出现。

五十一：GPU的工作原理

简而言之，GPU的图形（处理）流水线完成如下的工作：（并不一定是按照如下顺序） 顶点处理：这阶段GPU读取描述3D图形外观的顶点数据并根据顶点数据确定3D图形的形状及位置关系，建立起3D图形的骨架。在支持DX8和DX9规格的GPU中，这些工作由硬件实现的Vertex Shader（定点着色器）完成。 光栅化计算：显示器实际显示的图像是由像素组成的，我们需要将上面生成的图形上的点和线通过一定的算法转换到相应的像素点。把一个矢量图形转换为一系列像素点的过程就称为光栅化。例如，一条数学表示的斜线段，最终被转化成阶梯状的连续像素点。 纹理帖图：顶点单元生成的多边形只构成了3D物体的轮廓，而纹理映射（texture mapping）工作完成对多变形表面的帖图，通俗的说，就是将多边形的表面贴上相应的图片，从而生成“真实”的图形。TMU（Texture mapping unit）即是用来完成此项工作。 像素处理：这阶段（在对每个像素进行光栅化处理期间）GPU完成对像素的计算和处理，从而确定每个像素的最终属性。在支持DX8和DX9规格的GPU中，这些工作由硬件实现的Pixel Shader（像素着色器）完成。 最终输出：由ROP（光栅化引擎）最终完成像素的输出，1帧渲染完毕后，被送到显存帧缓冲区。

总结：GPU的工作通俗的来说就是完成3D图形的生成，将图形映射到相应的像素点上，对每个像素进行计算确定最终颜色并完成输出。

五十二：什么是渲染管道？

是指在显示器上为了显示出图像而经过的一系列必要操作。 渲染管道中的很多步骤，都要将几何物体从一个坐标系中变换到另一个坐标系中去。主要步骤有：

本地坐标->视图坐标->背面裁剪->光照->裁剪->投影->视图变换->光栅化

五十三：如何优化内存？

有很多种方式，例如

1.压缩自带类库；

2.将暂时不用的以后还需要使用的物体隐藏起来而不是直接Destroy掉；

3.释放AssetBundle占用的资源；

4.降低模型的片面数，降低模型的骨骼数量，降低贴图的大小；

5.使用光照贴图，使用多层次细节(LOD)，使用着色器(Shader)，使用预设(Prefab)。

6.代码中少产生临时变量

五十四：动态加载资源的方式？他们之间的区别

1.Resources.Load();

2.AssetBundle

区别参考

五十五：请描述游戏动画有哪几种，以及其原理？

主要有关节动画、骨骼动画、单一网格模型动画(关键帧动画)。

关节动画：把角色分成若干独立部分，一个部分对应一个网格模型，部分的动画连接成一个整体的动画，角色比较灵活，Quake2中使用这种动画；

骨骼动画，广泛应用的动画方式，集成了以上两个方式的优点，骨骼按角色特点组成一定的层次结构，有关节相连，可做相对运动，皮肤作为单一网格蒙在骨骼之外，决定角色的外观；

单一网格模型动画由一个完整的网格模型构成，在动画序列的关键帧里记录各个顶点的原位置及其改变量，然后插值运算实现动画效果，角色动画较真实。

五十六：alpha blend工作原理

Alpha Blend 实现透明效果，不过只能针对某块区域进行alpha操作，透明度可设。

五十七：写出光照计算中的diffuse的计算公式

diffuse = Kd x colorLight x max(N\*L,0)；Kd 漫反射系数、colorLight 光的颜色、N 单位法线向量、L 由点指向光源的单位向量、其中N与L点乘，如果结果小于等于0，则漫反射为0。

五十八：两种阴影判断的方法、工作原理。

本影和半影：参考本影和半影

本影：景物表面上那些没有被光源直接照射的区域（全黑的轮廓分明的区域）。

半影：景物表面上那些被某些特定光源直接照射但并非被所有特定光源直接照射的区域（半明半暗区域）

工作原理：从光源处向物体的所有可见面投射光线，将这些面投影到场景中得到投影面，再将这些投影面与场景中的其他平面求交得出阴影多边形，保存这些阴影多边形信息，然后再按视点位置对场景进行相应处理得到所要求的视图（利用空间换时间，每次只需依据视点位置进行一次阴影计算即可，省去了一次消隐过程）

五十九：Vertex Shader是什么，怎么计算？

顶点着色器是一段执行在GPU上的程序，用来取代fixed pipeline中的transformation和lighting，Vertex Shader主要操作顶点。

Vertex Shader对输入顶点完成了从local space到homogeneous space（齐次空间）的变换过程，homogeneous space即projection space的下一个space。在这其间共有world transformation, view transformation和projection transformation及lighting几个过程。

六十：下列代码在运行中会产生几个临时对象？

string a = new string("abc");

a = (a.ToUpper() + "123").Substring(0, 2);

在C#中第一行是会报错的（Java中倒是可行）。

应该这样初始化：

string b = new string(new char[]{'a','b','c'});

答案为：5个临时对象

六十一：下列代码在运行中会发生什么问题？如何避免？

List<int> ls = new List<int>(new int[] { 1, 2, 3, 4, 5 });

foreach (int item in ls)

{

Console.WriteLine(item \* item);

ls.Remove(item);

}

产生运行时错误，在 ls.Remove(item)这行，因为foreach是只读的。不能一边遍历一边修改。

六十二：Unity3D是否支持写成多线程程序？如果支持的话需要注意什么？

仅能从主线程中访问Unity3D的组件，对象和Unity3D系统调用

支持：如果同时你要处理很多事情或者与Unity的对象互动小可以用thread,否则使用coroutine。

注意：C#中有lock这个关键字,以确保只有一个线程可以在特定时间内访问特定的对象

六十三：Unity3D的协程和C#线程之间的区别是什么？

多线程程序同时运行多个线程 ，而在任一指定时刻只有一个协程在运行，并且这个正在运行的协同程序只在必要时才被挂起。除主线程之外的线程无法访问Unity3D的对象、组件、方法。

Unity3d没有多线程的概念，不过unity也给我们提供了StartCoroutine（协同程序）和LoadLevelAsync（异步加载关卡）后台加载场景的方法。 StartCoroutine为什么叫协同程序呢，所谓协同，就是当你在StartCoroutine的函数体里处理一段代码时，利用yield语句等待执行结果，这期间不影响主程序的继续执行，可以协同工作。

六十四：矩阵相乘的意义及注意点

用于表示线性变换：旋转、缩放、投影、平移、仿射

注意矩阵的蠕变：误差的积累

六十五：为什么dynamic font在unicode环境下优于static font

Unicode是国际组织制定的可以容纳世界上所有文字和符号的字符编码方案。

使用动态字体时，Unity将不会预先生成一个与所有字体的字符纹理。当需要支持亚洲语言或者较大的字体的时候，若使用正常纹理，则字体的纹理将非常大。

六十六：当一个细小的高速物体撞向另一个较大的物体时，会出现什么情况？如何避免？

穿透（碰撞检测失败）

六十七：请简述OnBecameVisible及OnBecameInvisible的发生时机，以及这一对回调函数的意义？

当物体是否可见切换之时。可以用于只需要在物体可见时才进行的计算。

六十八：什么叫动态合批？跟静态合批有什么区别？

如果动态物体共用着相同的材质，那么Unity会自动对这些物体进行批处理。动态批处理操作是自动完成的，并不需要你进行额外的操作。

区别：动态批处理一切都是自动的，不需要做任何操作，而且物体是可以移动的，但是限制很多。静态批处理：自由度很高，限制很少，缺点可能会占用更多的内存，而且经过静态批处理后的所有物体都不可以再移动了。

参考

六十九：简述StringBuilder和String的区别？

String是字符串常量。

StringBuffer是字符串变量 ，线程安全。

StringBuilder是字符串变量，线程不安全。

String类型是个不可变的对象，当每次对String进行改变时都需要生成一个新的String对象，然后将指针指向一个新的对象，如果在一个循环里面，不断的改变一个对象，就要不断的生成新的对象，所以效率很低，建议在不断更改String对象的地方不要使用String类型。

StringBuilder对象在做字符串连接操作时是在原来的字符串上进行修改，改善了性能。这一点我们平时使用中也许都知道，连接操作频繁的时候，使用StringBuilder对象。

七十：Unity3D Shader分哪几种，有什么区别？

表面着色器的抽象层次比较高，它可以轻松地以简洁方式实现复杂着色。表面着色器可同时在前向渲染及延迟渲染模式下正常工作。

顶点片段着色器可以非常灵活地实现需要的效果，但是需要编写更多的代码，并且很难与Unity的渲染管线完美集成。

固定功能管线着色器可以作为前两种着色器的备用选择，当硬件无法运行那些酷炫Shader的时，还可以通过固定功能管线着色器来绘制出一些基本的内容。

七十一：已知strcpy函数的原型是：char \* strcpy(char \* strDest,const char \* strSrc); 1.不调用库函数，实现strcpy函数。2.解释为什么要返回char \*

char \* strcpy(char \* strDest,const char \* strSrc)

{

if ((strDest==NULL)||(strSrc==NULL))

throw "Invalid argument(s)";

char \* strDestCopy=strDest;

while ((\*strDest++=\*strSrc++)!='\0');

return strDestCopy;

}

七十二：C#中四种访问修饰符是哪些？各有什么区别？

1.属性修饰符 2.存取修饰符 3.类修饰符 4.成员修饰符。

属性修饰符：

Serializable：按值将对象封送到远程服务器。

STATread：是单线程套间的意思，是一种线程模型。

MATAThread：是多线程套间的意思，也是一种线程模型。

存取修饰符：

public：存取不受限制。

private：只有包含该成员的类可以存取。

internal：只有当前工程可以存取。

protected：只有包含该成员的类以及派生类可以存取。

类修饰符：

abstract：抽象类。指示一个类只能作为其它类的基类。

sealed：密封类。指示一个类不能被继承。理所当然，密封类不能同时又是抽象类，因为抽象总是希望被继承的。

成员修饰符：

abstract：指示该方法或属性没有实现。

sealed：密封方法。可以防止在派生类中对该方法的override（重载）。不是类的每个成员方法都可以作为密封方法密封方法，必须对基类的虚方法进行重载，提供具体的实现方法。所以，在方法的声明中，sealed修饰符总是和override修饰符同时使用。

delegate：委托。用来定义一个函数指针。C#中的事件驱动是基于delegate + event的。

const：指定该成员的值只读不允许修改。

event：声明一个事件。

extern：指示方法在外部实现。

override：重写。对由基类继承成员的新实现。

readonly：指示一个域只能在声明时以及相同类的内部被赋值。

static：指示一个成员属于类型本身，而不是属于特定的对象。即在定义后可不经实例化，就可使用。

virtual：指示一个方法或存取器的实现可以在继承类中被覆盖。

new：在派生类中隐藏指定的基类成员，从而实现重写的功能。 若要隐藏继承类的成员，请使用相同名称在派生类中声明该成员，并用 new 修饰符修饰它。

七十三：Heap与Stack有何区别？

1.heap是堆，stack是栈。

2.stack的空间由操作系统自动分配和释放，heap的空间是手动申请和释放的，heap常用new关键字来分配。

3.stack空间有限，heap的空间是很大的自由区。

七十四：值类型和引用类型有何区别？

1.值类型的数据存储在内存的栈中；引用类型的数据存储在内存的堆中，而内存单元中只存放堆中对象的地址。

2.值类型存取速度快，引用类型存取速度慢。

3.值类型表示实际数据，引用类型表示指向存储在内存堆中的数据的指针或引用

4.值类型继承自System.ValueType，引用类型继承自System.Object

5.栈的内存分配是自动释放；而堆在.NET中会有GC来释放

6.值类型的变量直接存放实际的数据，而引用类型的变量存放的则是数据的地址，即对象的引用。

七十五：请写出求斐波那契数列任意一位的值得算法

递归实现：

int Fib1(int index)

{

if(index

迭代实现：

int Fib5(int index)

{

if(index

参看更多实现方法

七十六：协同程序的执行代码是什么？有何用处，有何缺点？

function Start() {

// 协同程序WaitAndPrint在Start函数内执行,可以视同于它与Start函数同步执行.

StartCoroutine(WaitAndPrint(2.0));

print ("Before WaitAndPrint Finishes " + Time.time );

}

function WaitAndPrint (waitTime : float) {

// 暂停执行waitTime秒

yield WaitForSeconds (waitTime);

print ("WaitAndPrint "+ Time.time );

}

作用：一个协同程序在执行过程中,可以在任意位置使用yield语句。yield的返回值控制何时恢复协同程序向下执行。协同程序在对象自有帧执行过程中堪称优秀。协同程序在性能上没有更多的开销。

缺点：协同程序并非真线程，可能会发生堵塞。

七十七：什么是里氏代换元则？

里氏替换原则(Liskov Substitution Principle LSP)面向对象设计的基本原则之一。通俗点：就是子类对象可以赋值给基类对象，基类对象不能赋值给子类对象

参考

七十八：Mock和Stub有何区别？

Mock与Stub的区别：Mock:关注行为验证。细粒度的测试，即代码的逻辑，多数情况下用于单元测试。Stub：关注状态验证。粗粒度的测试，在某个依赖系统不存在或者还没实现或者难以测试的情况下使用，例如访问文件系统，数据库连接，远程协议等。

七十九：概述序列化：

序列化简单理解成把对象转换为容易传输的格式的过程。比如，可以序列化一个对象，然后使用HTTP通过Internet在客户端和服务器端之间传输该对象

八十：堆和栈的区别？

栈通常保存着我们代码执行的步骤，如在代码段1中 AddFive()方法，int pValue变量，int result变量等等。而堆上存放的则多是对象，数据等。我们可以把栈想象成一个接着一个叠放在一起的盒子。当我们使用的时候，每次从最顶部取走一个盒子。栈也是如此，当一个方法（或类型）被调用完成的时候，就从栈顶取走，接着下一个。堆则不然，像是一个仓库，储存着我们使用的各种对象等信息，跟栈不同的是他们被调用完毕不会立即被清理掉。

八十一：概述c#中代理和事件？

代理就是用来定义指向方法的引用。

C＃事件本质就是对消息的封装，用作对象之间的通信；发送方叫事件发送器，接收方叫事件接收器

八十二：客户端与服务器交互方式有几种？

socket通常也称作”套接字”,实现服务器和客户端之间的物理连接，并进行数据传输，主要有UDP和TCP两个协议。Socket处于网络协议的传输层。

http协议传输的主要有http协议 和基于http协议的Soap协议（web service）,常见的方式是 http 的post 和get 请求，web 服务。

八十三：Unity和Android与iOS如何交互？

Unity可以到处Android和iOS的工程，然后通过安卓或者iOS的类去给Unity发消息，调用Unity中的方法

八十四：如何在Unity3D中查看场景的面试，顶点数和Draw Call数？如何降低Draw Call数？

在Game视图右上角点击Stats。降低Draw Call 的技术是Draw Call Batching

这个在5.0以后在window-》Profiler下面，快捷键是cmd + 7（ctl + 7

八十五：请问alpha test在何时使用？能达到什么效果？

Alpha Test ,中文就是透明度测试。简而言之就是V&F shader中最后fragment函数输出的该点颜色值（即上一讲frag的输出half4）的alpha值与固定值进行比较。AlphaTest语句通常于Pass{}中的起始位置。Alpha Test产生的效果也很极端，要么完全透明，即看不到，要么完全不透明。

八十六：UNITY3d在移动设备上的一些优化资源的方法

1.使用assetbundle，实现资源分离和共享，将内存控制到200m之内，同时也可以实现资源的在线更新

2.顶点数对渲染无论是cpu还是gpu都是压力最大的贡献者，降低顶点数到8万以下，fps稳定到了30帧左右

3.只使用一盏动态光，不是用阴影，不使用光照探头

粒子系统是cpu上的大头

4.剪裁粒子系统

5.合并同时出现的粒子系统

6.自己实现轻量级的粒子系统

animator也是一个效率奇差的地方

7.把不需要跟骨骼动画和动作过渡的地方全部使用animation，控制骨骼数量在30根以下

8.animator出视野不更新

9.删除无意义的animator

10.animator的初始化很耗时（粒子上能不能尽量不用animator）

11.除主角外都不要跟骨骼运动apply root motion

12.绝对禁止掉那些不带刚体带包围盒的物体（static collider ）运动

NUGI的代码效率很差，基本上runtime的时候对cpu的贡献和render不相上下

13每帧递归的计算finalalpha改为只有初始化和变动时计算

14去掉法线计算

15不要每帧计算viewsize 和windowsize

16filldrawcall时构建顶点缓存使用array.copy

17.代码剪裁：使用strip level ，使用.net2.0 subset

18.尽量减少smooth group

19.给美术定一个严格的经过科学验证的美术标准，并在U3D里面配以相应的检查工具

八十七：四元数有什么作用？

对旋转角度进行计算时用到四元数

八十八：将Camera组件的ClearFlags选项选成Depth only是什么意思？有何用处？

如果把摄像机的ClearFlags勾选为Deapth Only,那么摄像机就会只渲染看得见的对象，把背景会完全透明，这种情况一般用在两个摄像机以上的场景中

八十九：在编辑场景时将GameObject设置为Static有何作用？

设置游戏对象为Static时，这些部分被静态物体挡住而不可见时，将会剔除（或禁用）网格对象。因此，在你的场景中的所有不会动的物体都应该标记为Static。

九十：有A和B两组物体，有什么办法能够保证A组物体永远比B组物体先渲染？

把A组物体的渲染对列大于B物体的渲染队列，通过shader里面的渲染队列来渲染

九十一：将图片的TextureType选项分别选为““Texture”和“Sprite”有什么区别

Sprite作为UI精灵使用，Texture作用模型贴图使用。Sprite需要2的整次幂，打包图片省资源

九十二：问一个Terrain，分别贴3张，4张，5张地表贴图，渲染速度有什么区别？为什么？

没有区别，因为不管几张贴图只渲染一次。

九十三：什么是DrawCall？DrawCall高了又什么影响？如何降低DrawCall？

Unity中，每次引擎准备数据并通知GPU的过程称为一次Draw Call。DrawCall越高对显卡的消耗就越大。降低DrawCall的方法：

1. Dynamic Batching

2. Static Batching

3. 高级特性Shader降级为统一的低级特性的Shader。

九十四：实时点光源的优缺点是什么？

可以有cookies – 带有 alpha通道的立方图(Cubemap )纹理。点光源是最耗费资源的。

九十五：Unity的Shader中，Blend SrcAlpha OneMinusSrcAlpha这句话是什么意思？

作用就是Alpha混合。公式：最终颜色 = 源颜色 x 源透明值 + 目标颜色 x（1 - 源透明值）

九十六：简述水面倒影的渲染原理

原理就是对水面的贴图纹理进行扰动，以产生波光玲玲的效果。用shader可以通过GPU在像素级别作扰动，效果细腻，需要的顶点少，速度快

九十七：简述NGUI中Grid和Table的作用？

对Grid和Table下的子物体进行排序和定位

九十八：请简述NGUI中Panel和Anchor的作用

1. 只要提供一个half-pixel偏移量，它可以让一个控件的位置在Windows系统上精确的显示出来（只有这个Anchor的子控件会受到影响）

2. 如果挂载到一个对象上，那么他可以将这个对象依附到屏幕的角落或者边缘

3.UIPanel用来收集和管理它下面所有widget的组件。通过widget的geometry创建实际的draw call。没有panel所有东西都不能够被渲染出来,你可以把UIPanel当做Renderer

九十九：能用foreach遍历访问的对象需要实现\_接口或声明\_\_\_\_方法的类型

IEnumerable；GetEnumerator

第二部分

1.以下哪一个选项不属于Unity引擎所支持的视频格式文件（D）

A.后缀为mov的文件 B.后缀为mpg的文件

C.后缀为avi的文件 D.后缀为swf的文件

2.Unity引擎使用的是左手坐标系还是右手坐标系（A）

A.左手坐标系 B.右手坐标系

C.可以通过ProjectSetting切换右手坐标系 D.可以通过Reference切换左手坐标系

3.什么是导航网格（NavMesh）（B）

A.一种用于描述相机轨迹的网格 B.一种用于实现自动寻址的网格

C.一种被优化过的物体网格 D.一种用于物理碰撞的网格

4.下列选项中有关Animator的说法错误的是？（D）

A.Animator是Unity引擎中内置的组件

B.任何一个具有动画状态机功能的GameObject都需要一个Anim组件

C.它主要用于角色行为的设置，包括StateMachine、混合树BlendTrees以及同通过脚本控制的事件

D.Animator同Animation组件的用法是相同的

5.Application.loadLevel命令为（A）

A.加载关卡 B.异步加载关卡 C.加载动作 D.加载动画

6.下列选项中，关于Transform组件的Scale参数描述正确的是（A）

A.Transform组件的Scale参数不会影响ParticleSystem产生粒子的大小

B.Transform组件的Scale参数不会影响GUITexture的大小

C.添加Collider组件后的GameoObject，其 Collider 组件的尺寸不受Transform组件的Scale参数影响

D.添加Rigidbody组件后的物体，大小将不再受Transform组件中 Scale 参数的影响

7.在 Unity 中的场景中创建 Camera 时，默认情况下除了带有Transform、Camera、GUILayer、Flare Layer 组件之外，还带有以下哪种组件（C）

A.Mouse Look B.FPS Input Controller C.Audio Listener D.Character Motor

8.如果将一个声音剪辑文件从Project 视图拖动到 Inspector 视图或者 Scene 视图中的游戏对象上，该游戏对象会自动添加以下哪种组件（C）

A.Audio Listener B.Audio Clip C.Audio Source D.Audio Reverb Zone

9.下列叙述中有关 Prefab 说法错误的是哪一项（B）

A.Prefab 是一种资源类型 B.Prefab 是一种可以反复使用的游戏对象

C.Prefab 可以多次在场景进行实例 D.当一个 Prefab 添加到场景中时，也就是创建了它的一个实例

10.关于 MonoBehaviour.LateUpdate 函数描述错误的是：（B）

A.当 MonoBehaviour 类被启用后，每帧调用一次

B.常被用于处理 Rigidbody 的更新

C.在所有 Update 函数执行后才能被调用

D.常被用于实现跟随相机效果，且目标物体的位置已经在 Update 函数中被更新

11.下列哪个函数不属于碰撞事件（C）

A.OnCollisionEnter B.OnCollisionExit

C.OnCollisionUpdate D.OnCollisionStay

12.以下关于 MonoBehaviour.OnGUI()的描述错误的是（D）

A.如果 MonoBehaviour 没有被启用，则OnGUI函数不会被调用

B.用于绘制和处理 GUI events

C.每帧可能会被绘制多次，每次对应于一个 GUI event

D.每帧被调用一次

13.以下哪组摄像机中 Normalized View Port Rect 的数值设置可以使摄像机显示的画面位于1280\*720分辨率的屏幕画面右上角（D）

A.X=640,Y=360,W=640,H=360 B.X=640,Y=0,W=640,H=360

C.X=0,Y=0,W=0.5,H=0.5 D.X=0.5,Y=0.5,W=0.5,H=0.5

14.在Unity引擎中，Collider所指的是什么（D）

A.collider 是Unity引擎中所支持的一种资源，可用作存储网格信息

B.Collider 是Unity引擎中内置的一种组件，可用对网格进行渲染

C.Collider 是Unity引擎中所支持的一种资源，可用作游戏对象的坐标转换

D.Collider 是Unity引擎中内置的一种组件，可用作游戏对象之间的碰撞检测

15.以下关于WWW.LoadFromCacheOrDownload描述正确的是（C）

A.可被用于将 Text Assets 自动缓存到本地磁盘

B.可被用于将 Resource 自动缓存到本地磁盘

C.可被用于将 Asset Bundles 自动缓存到本地磁盘

D.可被用于将任意格式的Unity资源文件自动缓存到本地磁盘

16.如何实现加载外部视频并播放？

外部视频文件：目前测试仅支持ogg格式（设置网络获取视频时，必须将MIME设置.ogg为application/octet-stream）

场景设置：MainCamera上添加AudioSource脚本；播放物体上（如Plane）添加MovieTest脚本

MovieTest脚本：

using UnityEngine;

using System.Collections;

public class MovieTest : MonoBehaviour

{

//视频纹理

protected MovieTexture movTexture;

AudioClip audio;

AudioSource AudioSource1;

void Start()

{

StartCoroutine(DownLoadMovie());

}

void OnGUI()

{

if (GUILayout.Button("播放/继续"))

{

//播放/继续播放视频

if (!movTexture.isPlaying)

{

movTexture.Play();

AudioSource1.Play();

}

}

if (GUILayout.Button("暂停播放"))

{

//暂停播放

movTexture.Pause();

AudioSource1.Pause();

}

if (GUILayout.Button("停止播放"))

{

//停止播放

movTexture.Stop();

AudioSource1.Stop();

}

}

IEnumerator DownLoadMovie()

{

WWW www = new WWW ("http://127.0.0.1/Wildlife.ogg");//"file://" + Application.dataPath + "/Resources/Wildlife.ogg");

yield return www;

movTexture = www.movie;

//获取主相机的声源

AudioSource1 = Camera.main.GetComponent(typeof(AudioSource)) as AudioSource;

//获取视频的声音设置到声源上

AudioSource1.clip = movTexture.audioClip;

audio = AudioSource1.clip;

//设置当前对象的主纹理为电影纹理

renderer.material.mainTexture = movTexture;

//设置电影纹理播放模式为循环

movTexture.loop = true;

}

}

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

17.游戏对象B是游戏对象A的子物体，游戏对象A经过了旋转，请写出游戏B围绕自身的Y轴进行旋转的脚本语句，以及游戏对象B围绕世界坐标的Y轴旋转的脚本语句。

绕世界坐标旋转：transform.Rotate (transform.up\*speed\*Time.deltatime);

绕自身Y轴旋转：transform.Rotate (Vector.up\*speed\*Time.deltatime);

18.Unity中用过哪些插件？具体功能

FXMaker，制作粒子特效；NGUI，制作2D界面；EasyTouch，摇杆；shaderForge，制作shader；Itween，制作动画；

19.当删除Unity工程Assets目录下地meta文件时会导致什么？为什么？

会导致在场景中游戏对象看不到，或者报错，材质找不到资源。多人协作的时候会导致资源的重复产生。因为每个资源文件都对应一个.meta文件，这个.meta文件中的guid就是唯一标识这个资源的。材质就是通过这个guid来记录自己使用了那些资源，而且同一个资源的guid会因为不同的电脑而不同，所以当你上传了丢失了.meta文件的资源的时候，到了别人的机器上就会重新产生guid，那个这个资源就相当于垃圾了。

20.频繁创建GameObject会降低程序性能为什么？怎么解决？

频繁创建游戏对象，会增加游戏的Drawcall数，降低帧率，GPU会一直在渲染绘制。可以通过对象池来管理对象：当需要创建一个游戏对象时，先去对象池中查找一下对象池中是否存在没有被正在使用的对象，如果有的话直接使用这个对象，并把它标记为正在使用，没有话就创建一个，并把它添加到池中，然后标记为使用中。一个游戏对象使用完毕的时候，不要销毁掉，把它放在池中，标记为未使用。

——————————-

21.关于Vector3 的API，以下说法正确的是（BC）

A.Vector3.normalize 可以获取一个三维向量的法线向量

B.Vector3.magnitude 可以获取一个三维向量的长度

C.Vector3.forward 与 Vector3(0,0,1)是一样的意思

D.Vector3.Dot(向量A,向量B)是用来计算向量A与向量B的叉乘

22.以下哪个函数在游戏进入新场景后会被马上调用（B）

A.MonoBehaviour.OnSceneWastLoaded()

B.MonoBehaviour.OnSceneEnter()

C.MonoBehaviour.OnLevelEnter()

D.MonoBehaviour.OnLevelWastLoaded()

23.采用Input.mousePosition 来获取鼠标在屏幕上的位置，以下表达正确的是（C）

A.左上角为原点（0，0），右下角为（Screen.Width, Screen.Height）

B.左下角为原点（0，0），右下角为（Screen.Height, Screen.Width）

C.左下角为原点（0，0），右上角为（Screen.Width, Screen.Height）

D.左上角为原点（0，0），右下角为（Screen.Height, Screen.Width）

24.如何通过脚本来删除其自身对应的Gameobject（A）

A.Destroy(gameObject) B.this.Destroy()

C.Destroy(this) D.其他三项都可以

25.某个GameObject有一个名为MyScript的脚本，该脚本中有一个名为DoSomething 的函数，则如何在该Gameobject的另外一个脚本中调用该函数？（A）

A.GetComponent().DoSomething()

B.GetComponent

C.GetComponent().Call(“DoSomething”)

D.GetComponent

26.Animator.CrossFade 命令作用是：（B）

A.动画放大 B.动画转换 C.Update() D.OnMouseButton()

27.OnEnable,Awake,Start 运行时的发生顺序？（A）

A.Awake->OnEnable->Start B.Awake->Start->OnEable

C.OnEnable-Awake->Start D.Start->OnEnable->Awake

28.以下选项中，正确的是（D）

A.Mathf.Round方法作用是限制 B.Mathf.Clamp方法作用是插值

C.Mathf.Lerp方法作用是四舍五入 D.Mathf.Abs方法作用是取得绝对值

29.以下选项中，将游戏对象绕Z轴逆时针旋转90度（C）

A.transform.rotation = Quaternion.Euler(0,0,90)

B.transform.rotation = Quaternion.Angle(0,0,90)

C.transform.Rotate(new Vector3(0,0,90))

D.transform.Rotate(new Vector3(90,0,0))

30.public static function InitializeServer(connections:int,listenPort:int,useNat:bool):NetworkConnectionError;解释一下函数，参数以及返回值的意思。

初始化服务器。connections是允许的入站连接或玩家的数量，listenPort是要监听的端口，useNat设置NAT穿透功能。如果你想要这个服务器能够接受连接使用NAT穿透，使用facilitator，设置这个为true。如果有错误会有返回错误。

31.请写出以下函数的含义和运算结果

delegate b Func<a, b>(a a1);

static void Main(string[] args)

{

Func<int, bool> mFunc = x => x == 5;

Console.WriteLine(mFunc(6));

}

false，就是定义一个delegate，返回值类型为 b，有一个参数，类型为a。

32.编写一个函数，输入一个32位整数，计算这个整数有多少个bit为1.

uint BitCount (uint n)

{

uint c = 0; // 计数器

while (n > 0) {

if ((n & 1) == 1) // 当前位是1

++c; // 计数器加1

n >>= 1; // 移位

}

return c;

}

33.某游戏中的装备系统有16种附加属性，每种附加属性使用一个32位的ID表示(比如10001表示加人物hp的附加属性，10002表示加人物mp的附加属性），一件装备做多有4个附加属性，请写一个程序输出所有附加属性的组合。

#

34.请实现如下函数，在Unity中有一副骨骼树，请使用递归方式与非递归方式实现先序遍历，在Unity的Console输出所有骨骼名。

可能用到的函数

public Transform GetChild（int index）;

public int Transform.childCount

public void OutputTree(Transform root)

{

}

#

35.简要解释下数据库中ACID的含义。

ACID是指在可靠数据库管理系统（DBMS）中，事务所具有的四个特性：原子性（Atomicity）、一致性（Consistency）、隔离性（Isolation）、持久性（Durability）。

原子性是指事务是一个不可分割的工作单位，事务中的操作要么都发生，要么都不发生。

一致性是指在事务开始之前和事务结束以后，数据库的完整性约束没有被破坏。这是说数据库事务不能破坏关系数据的完整性以及业务逻辑上的一致性。例如：对银行转帐事务，不管事务成功还是失败，应该保证事务结束后ACCOUNT表中aaa和bbb的存款总额为2000元。

隔离性多个事务并发访问时，事务之间是隔离的，一个事务不应该影响其它事务运行效果。这指的是在并发环境中，当不同的事务同时操纵相同的数据时，每个事务都有各自的完整数据空间。由并发事务所做的修改必须与任何其他并发事务所做的修改隔离。事务查看数据更新时，数据所处的状态要么是另一事务修改它之前的状态，要么是另一事务修改它之后的状态，事务不会查看到中间状态的数据。

持久性，意味着在事务完成以后，该事务所对数据库所作的更改便持久的保存在数据库之中，并不会被回滚。即使出现了任何事故比如断电等，事务一旦提交，则持久化保存在数据库中。

36.32位整数256和255按位异或后的结果是（511）

37.unix/linux 系统将所有的I/O设备模型化为文件,c语言中的（stdin）,(stdout)和(stderr)分别表示标准输入，标准输出，标准错误。

38.k层二叉树最多有 2^k - 1 个结点。

39.中缀算式(8 + x\*y)-2x/3的后缀算式是\_。

40.对以下二叉树进行前序遍历的结果是（ABDGCEHF）

41.写出一个C类ip地址（192.168.1.105）,该ip地址一个合法的网格掩码是（255.255.255.224）.

42.c/c++程序运行时有堆内存与栈内存之分，请写一个语句在堆中分配一个整数：(int a = new int(4)),在栈内存中分配一个整数：（int a = 5）。

43.简述从C/C++源代码生成可执行文件所经历的几个过程和每个过程中所做的事情。

预编译-》编译-》链接-》执行

44.简述TCP建立连接的过程，最好能画出时序图。

三次握手；

45.有一个c语言函数

unsigned int getN(unsigned int v){

v--;

v |= v >> 1;

v |= v >> 2;

v |= v >> 4;

v |= v >> 8;

v |= v >> 16;

v++;

return v;

}

请问这个函数的功能是什么？

返回的结果是v 或者 2 ^ 跟下v次方 + 1（比如125返回128，128返回的就是128）

46.列出c语言中static关键字的用法。

static 关键字至少有下列几个作用：

（1）函数体内static 变量的作用范围为该函数体，不同于auto 变量，该变量的内存只被分配一次，因此其值在下次调用时仍维持上次的值；

（2）在模块内的static 全局变量可以被模块内所用函数访问，但不能被模块外其它函数访问；

（3）在模块内的static 函数只可被这一模块内的其它函数调用，这个函数的使用范围被限制在声明它的模块内；

（4）在类中的static 成员变量属于整个类所拥有，对类的所有对象只有一份拷贝；

（5）在类中的static 成员函数属于整个类所拥有，这个函数不接收this 指针，因而只能访问类的static 成员变量。

47.解释一下Unity3D中的协程（coroutine）是什么？并举例说明

在主线程运行的同时开启另一段逻辑处理，来协助当前程序的执行，协程很像多线程，但是不是多线程，Unity的协程实在每帧结束之后去检测yield的条件是否满足。

48.物体自旋转使用的函数叫什么？物体绕某点旋转使用函数叫什么？

物体自旋转函数transform.Rotate，物体绕某点旋转函数：transform.RotateAround

49.使用prefab预制物体有什么好处？

1.Prefab一个重要的优势就是编辑prefab资源后，场景中所有使用Prefab克隆的游戏对象将全部使用新编辑的资源，无需一个一个的给场景中的对象赋值。

2.当游戏中需要频繁创建一个物体时，使用Prefab能够节省内存。

3.当你在一个场景中增加一个Prefabs，你就实例化了一个Prefabs。

50.设有如下关系表R1和R2

R1(NO, NAME, SEX, AGE, CLASS)

R2(NO, SUBJECT, SCORE)

主关键字是NO,其中NO为学号，NAME为姓名，SEX为性别，AGE为年龄，CLASS为班号，SUBJECT为科目，SCORE为成绩。写出实现下列功能的SQL语句。查找学号为20的学生的姓名，科目，成绩。

SELECT NAME,SUBJECT,SCORE FORM R1 INNER JOIN R2 ON R1.NO=R2.NO WHERE R1.NO = 20

51.描述多人联网中位置的同步和聊天通讯的实现方法，并编程写出如何实现多人协同中交互操作。（交互操作例如，血值为100的两个角色人物可以相互射击双方，射击一次减去血值10，当血值为0的时候证明已经死亡）

#

52.当实例化一个prefab对象，prefab对象上面挂载了一个继承MonoBehaviour的脚本

1.请问这个脚本分别会按顺序调用哪些函数，并列出哪些函数是反复进去的

#

2.请问当这个实例化对象先调用了SetActive(false)，然后又调用了SetActive(true),那么这个脚本依次会调用哪些函数方法？

#

3.如果在脚本的Awake（）函数中，调用了this.gameObject.AddComponent()(PS:GameController为另一个继承Mono的脚本类)，请问此时脚本函数的调用顺序是否发生变化？

53.if(go.CompareTag(“Enemy”))和if（go.tag == “Enemy”）两种判断方法哪种是合适的，为什么？

第一种合适，因为第二种会占用更多内存。

54.DestroyImmediate和Destroy的区别是？

DestroyImmeditate 销毁对象的时候，会立即释放资源。Destroy只是从该场景销毁，但是还在内存当中。

55.详细说明Unity中资源加载的方法，以及他们的区别？

1.通过Resources模块，调用它的load函数：可以直接load并返回某个类型的Object，前提是要把这个资源放在Resource命名的文件夹下，Unity不关有没有场景引用，都会将其全部打入到安装包中。

2.通过bundle的形式：即将资源打成 asset bundle 放在服务器或本地磁盘，然后使用WWW模块get 下来，然后从这个bundle中load某个object。

56.从代码角度上如何进行内存优化？

#

57.分别写出Invoke和协程的几种用法？

#

58.以下函数的功能是计算斐波那契数列的第n项，请填空

int func(int n)

{

if(n<=2） return 1;

return n + func(n-1);

}

59.C语言中宏定义中若有多行，可以使用字符（\）.

60.C语言中32位整数能表达的最小的数是-2^31~2^31-1.

61.使用（点乘）向量运算可以很方便地计算出三维空间中两个单位向量之间夹角的cos值。

62.类unix系统中某文件的权限为：drw-r–r–，用八进制数值形式表示该权限为(411),首位d代表目录（文件夹）

63.若有如下程序段，其中s，a，b，c均已定义为整型变量，且a，c均已赋值（c大于0）

s = a;

for(b = 1; b <= c; b++)

s = s + 1;

则与上述程序段功能等价的赋值语句是（B）

A.s = a + b B.s = a + c C.s = s + c D.s = b + c

64.简述static和const关键字的作用

static 关键字至少有下列几个作用：

（1）函数体内static 变量的作用范围为该函数体，不同于auto 变量，该变量的内存只被分配一次，因此其值在下次调用时仍维持上次的值；

（2）在模块内的static 全局变量可以被模块内所用函数访问，但不能被模块外其它函数访问；

（3）在模块内的static 函数只可被这一模块内的其它函数调用，这个函数的使用范围被限制在声明它的模块内；

（4）在类中的static 成员变量属于整个类所拥有，对类的所有对象只有一份拷贝；

（5）在类中的static 成员函数属于整个类所拥有，这个函数不接收this 指针，因而只能访问类的static 成员变量。

const 关键字至少有下列几个作用：

（1）欲阻止一个变量被改变，可以使用const 关键字。在定义该const 变量时，通常需要对它进行初始化，因为以后就没有机会再去改变它了；

（2）对指针来说，可以指定指针本身为const，也可以指定指针所指的数据为const，或二者同时指定为const；

（3）在一个函数声明中，const 可以修饰形参，表明它是一个输入参数，在函数内部不能改变其值；

（4）对于类的成员函数，若指定其为const 类型，则表明其是一个常函数，不能修改类的成员变量

；

（5）对于类的成员函数，有时候必须指定其返回值为const 类型，以使得其返回值不为“左值”。

65.用你熟悉的语言及你认为最简洁的方法书写计算 s = 1!+2!+3!+…+num!的代码。num为输入，s为输出。（！代表阶乘 3！= 1 \* 2 \* 3

Console.ReadLine(num)

int s = 0;

for(int i = 1; i <= num; i++)

{

s += JieCheng(num);

}

public int JieCheng(int num)

{

if(num < 0)

{

Console.WriteLine("error");

return;

}

if(num <=1)

{

return 1;

}

else {

return num \* JieCheng(num - 1)

}

}

66.用你熟悉的语言从一个字符串中去掉相连的重复字符，例如原字符串“adffjkljaalkjhl”变为“adfjkljalkjhl”

int GetResult(char[] input, char[] output)

{

int i, j, k = 0;

int flag;

int length;

if(input == NULL || output == NULL)

{

return -1;

}

length=strlen(input);//求数组的长度

for(i = 0; i<length; i++)

{

flag = 1;

for(j = 0; j < i; j++)

{

if(output[j] == input [i])

flag = 0;

}

if(flag)

output[k++] = input[i];

}

printf("最终的字符串为：");

output[k] = '\0';

for(int m = 0; m < output.Length; m++)

{

print (output [m]);

}

return 0;

}

第三部分

1.哪种实时光源是Unity中没有的？（D）

A.点光源 B.方向光 C.聚光灯 D.日光灯

2.如何在Unity中创建地形系统？(D)

A.Terrain->Create Terrain B.Component->Create Terrain

C.Asset->Create Terrain D.Windows->Create Terrain

3.以下哪种操作步骤可以在场景中添加“Wind Zone”?(B)

A.Terrain->Wind Zone B.GameObject->Create other->Wind Zone

C.Component->Physics->Wind Zone D.Assets->Create->Wind Zone

4.在Unity编辑器中创建一个Directional Light，以下步骤正确的是？(B)

A.Edit->Rendering Setting->Directional Light

B.GameObject->Create Other->Directional Light

C.Component->Rendering->Directional Light

D.Assets->Directional Light

5.下列哪一项不属于Camera中的“Clear Flags”？（D）

A.Skybox B.Solid Color C.Depth Only D.Background

6.以下哪种脚本语言是Unity编辑器所不支持的？（D）

A.Javascript B.C# C.Boo D.Perl

7.对于Prefab，以下说法错误的是？（D）

A.Prefab资源可以在项目中多次重复使用

B.由Prefab实例出的GameObject，其在Hierarchy试图中表现为蓝色

C.Prefab上的组件信息一经改变，其实例出的GameObject也会自动改变

D.实例出的GameObject上的组件信息一经改变，其对应出的Prefab也会自动改变

8.下面哪种做法可以打开Unity的Asset Store？（A）

A.Windows->Asset Store

B.Edit->Asset Store

C.File->Asset Store

D.Assets->Asset Store

9.在哪个面板中可以修改物体的空间属性，如位置，朝向，大小等（B）

A.Project B.Inspector C.Hierarchy D.Toolbar

10.如何为一个Asset 资源设定一个Label，从而能够方便准确的搜索到？（D）

A.在Project窗口中选中一个Asset，右键->Create->Label

B.在Project窗口中选中一个Asset，右键->Add Label

C.在Project窗口中选中一个Asset，在Inspector窗口中点击添加Label的图标

D.在Project窗口中选中一个Asset，在Inspector窗口中点击按钮“Add Label”

11.Mecanim系统中，Body Mask的作用是？（D）

A指定身体的某一部分是否参与骨骼动画

B.指定身体的某一部分是否参与物理模拟

C.指定身体的某一部分是否可以输出骨骼信息

D.指定身体的某一部分是否参与渲染

12.以下哪种操作步骤可以打开Unity编辑器的Lightmapping视图？（C）

A.File->Lightmapping B.Assets->Lightmapping

C.Windows->Lightmapping D.Component->Lightmapping

13.下列关于光照贴图，说法错误的是？（C）

A.使用光照贴图比使用实时光源渲染要快

B.可以降低游戏内存消耗

C.可以增加场景真实感

D.多个物体可以使用同一张光照贴图

14.如何为物体添加光照贴图所使用的UV?(B)

A.不用添加，任何时候都会自动生成

B.更改物体导入设置，勾选“Generate Lighting UVs”

C.更改物体导入设置，勾选“Swap UVs”

D.更改物体导入设置，在UVs选项中选择“Use Lightmaps”

15.在哪个模块下可以修改Render Path？（A）

A.Camera B.Light C.Render Settings D.Project Setting->Quality

16.以下哪项技术下不是目前Unity所支持的Occlusion Culling技术？（D）

A.PVS only B.PVS and dynamic objets

C.Automatic Portal Generation D.Dynamic Only

17.关于Vector3的API，以下说法正确的是？（C）

A.Vector3.normalize可以获取一个三维向量的法线向量

B.Vector3.magnitude可以获取一个三维向量的长度

C.Vector3.forward与Vector3（0，0，1）是一样的

D.Vector3.Dot(向量A，向量B)是用来计算向量A与向量B的叉乘

18.下列那些选项不是网格层属性的固有选项？（B）

A.Default B.Walkable C.Not Walkable D.Jump

19.写出你对游戏的理解及游戏在生活中的作用，对Unity3D软件理解最深入的地方。

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

作者：崔小妹儿Jessica

来源：CSDN

原文：https://blog.csdn.net/qq\_26270779/article/details/53609069

版权声明：本文为博主原创文章，转载请附上博文链接！