

1. 简述蒙皮动画 (SkinMesh) 的工作原理。
 2. 下列代码在运行中会发生什么问题？如何避免？
 3. 请简述值类型与引用类型的区别。
 4. C#里面把数据类型分为两大类，分别为值类型和引用类型。
 5. 列出你所知道的渲染效率优化技巧。
 6. unity3d对象从唤醒到销毁有一段生命周期，请列出系统自己调用的几个重要方法。
 7. 列出你所知道的C#性能优化技巧。
 8. 图形学上三角形对比其他形状有何优点？
 9. 如何在不同的场景传递数据？
 10. 列出你所知道的渲染效率优化技巧。
 11. Lua语言中pairs和ipairs有何区别？
 12. Lua语言，假设有一个table t1，如何判断t1内有没有元素？
 13. 有限状态机与行为树的区别。
 14. const和readonly有什么区别？
 15. c#中class和struct的区别与适用。
 16. Hash的意义。
 17. Update和FixedUpdate有什么区别？
 18. 列出C#容器，并说明。[整理中]
 19. 反射的实现原理及作用？[整理中]
- 备选池：

1. 简述蒙皮动画 (SkinMesh) 的工作原理。

Mesh附着在骨骼顶点上，通过顶点权重，进行运动变形。

2. 下列代码在运行中会发生什么问题？如何避免？

```
1 List ls = new List(new int[] {1, 2, 3, 4, 5});
2 foreach(int item in ls)
3 {
4     Console.WriteLine(item * item);
5     ls.Remove(item);
6 }
```

问题：

1. 运行时错误，因为foreach是只读的。不能一边遍历一边修改。
2. 语法错误，List类是泛型类，缺少指定类型。

修正：

```
1 List<int> ls = new List<int>(new int[] {1, 2, 3, 4, 5});
2 for (int i = ls.Count - 1; i >= 0; i--)
3 {
4     System.Console.WriteLine(ls[i] * ls[i]);
5     ls.Remove(ls[i]); //或 ls.RemoveAt(i);
6 }
```

3. 请简述值类型与引用类型的区别。

1. 值类型存储在内存栈中，引用类型数据存储在内存堆中，实际存放的是内存堆中对象的地址。
2. 值类型存取快，引用类型存取慢。
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. 栈的内存是自动释放的，堆内存是.NET中会由GC来自动释放。
5. 值类型继承自System.ValueType（它的父类是System.Object），引用类型继承自System.Object。

4.C#里面把数据类型分为两大类，分别为值类型和引用类型。

1. 值类型包括基本数据类型(int, double等)，结构(struct)和枚举(enum)。
2. 引用类型包括接口(interface)，数组，Object类型，类(class)，委托(delegate)，字符串(string)，null类型等。

5.列出你所知道的渲染效率优化技巧。

1. 删除无效骨骼
2. 减少模型面数
3. 避免动态阴影
4. 减少在片段着色器进行运算

6.unity3d对象从唤醒到销毁有一段生命周期，请列出系统自己调用的几个重要方法。

Awake -> OnEnable -> Start -> FixedUpdate -> Update -> LateUpdate -> OnDisable -> OnDestroy

7.列出你所知道的C#性能优化技巧。

1. 减少new操作
2. StringBuilder替代String
3. 不要在循环中使用try-catch异常处理

8.图形学上三角形对比其他形状有何优点？

几乎所有形状都能近似分解成多个三角形

9.如何在不同的场景传递数据？

PlayerPrefs

10.列出你所知道的渲染效率优化技巧。

1. 删除无效骨骼

2. 减少模型面数
3. 避免动态阴影
4. 减少在片段着色器进行运算

11.Lua语言中pairs和ipairs有何区别？

pairs遍历table中所有元素，ipairs遍历table元素，直到遇到nil。

12.Lua语言，假设有一个table t1，如何判断t1内有没有元素？

#t1 > 0 则有元素，否则没有。

13.有限状态机与行为树的区别。

有限状态机(FSM)的优势之一是简单。但是FSM需要用转换连接状态，因此，状态失去了模块性。行为树的主要优势之一就是其更好的封装性和模块性，让游戏逻辑更直观。

14.const和readonly有什么区别？

const关键字用来声明编译时的常量，readonly用来声明运行时的常量。

15.c#中class和struct的区别与适用。

<区别>：

1. class 是引用类型，structs是值类型，既然class是引用类型，class可以设为null。但是我们不能将struct设为null,因为它是值类型。
2. 当你实例一个class，它将创建在堆上。而你实例一个struct，它将创建在栈上
3. 你使用的是一个对class实例的引用。而你使用的不是对一个struct的引用。（而是直接使用它们）
4. 当我们将class作为参数传给一个方法，我们传递的是一个引用。struct传递的是值而非引用。
5. structs 不可以有初始化器，class可以有初始化器。
6. Classes 可以有明显的无参数构造器，但是Struct不可以
7. 类使用前必须new关键字实例化，Struct不需要
8. class支持继承和多态，Struct不支持. 注意：但是Struct 可以和类一样实现接口
9. 既然Struct不支持继承，其成员不能以protected 或Protected Internal 修饰
10. Class的构造器不需要初始化全部字段，Struct的构造器必须初始化所有字段
11. Class可以定义析构器但是Struct不可以
12. Class比较适合大的和复杂的数据，Struct适用于作为经常使用的一些数据组合成的新类型。

<适用>：

1. Struct有性能优势，Class有面向对象的扩展优势。
2. 用于底层数据存储的类型设计为Struct类型，用于定义应用程序行为的类型设计为Class。
3. 如果对类型将来的应用情况不能确定，应该使用Class。

16.Hash的意义。

<概念>：

hash（散列、杂凑）函数，是将任意长度的数据映射到有限长度的域上。

<原理>：

将m分成固定长度（如128位），依次进行hash运算，然后用不同的方法迭代即可（如前一块的hash值与后一块的hash值进行异或）。如果不够128位怎么办？用0补全或者用1补全随意，算法中约定好就可以了。

<特征>：

1. hash值有两大特征：重复、不可逆。
2. 解释下为什么不可逆，就好比（-2）的平方等于4,2的平方等于4.你知道结果是4，但是不能逆推回原始值是多少。
3. 如果两个散列值相同，两个输入值很可能是相同的，但也可能不同，这种情况称为“散列碰撞（collision）”。

<用途>：

常用于加密算法，例如md5 sha1都是hash算法

17.Update和FixedUpdate有什么区别？

Update是渲染帧，跟设备有关。FixedUpdate是固定帧，与设备无关。与Rigidbody有关就用FixedUpdate。

18.列出C#容器，并说明。[整理中]

<https://blog.csdn.net/silangquan/article/details/51106968>

1. Array
2. ArrayList
3. List
4. LinkedList
5. HashSet
6. HashTable
7. Dictionary

19.反射的实现原理及作用？[整理中]

<http://blog.163.com/xuanmingzhiyou@yeah/blog/static/1424776762011612115124188/>

反射的定义：

审查元数据并收集关于它的类型信息的能力。元数据（编译以后的最基本数据单元）就是一大堆的表，当编译程序集或者模块时，编译器会创建一个类定义表，一个字段定义表，和一个方法定义表等。

反射的作用：

1. 可以使用反射动态地创建类型的实例，将类型绑定到现有对象，或从现有对象中获取类型
2. 应用程序需要在运行时从某个特定的程序集中载入一个特定的类型，以便实现某个任务时可以用到反射。
3. 反射主要应用与类库，这些类库需要知道一个类型的定义，以便提供更多的功能。

备选池：

1. u3d中碰撞器和触发器的区别？
----- <http://www.cnblogs.com/infly123/p/3920393.html>
2. 物体发生碰撞的必要条件？
----- <https://blog.csdn.net/matako/article/details/8588214>
3. 动态加载资源的方式和区别？
----- <https://blog.csdn.net/leonwei/article/details/18406103>
4. 列出几个Profile时你关注的的数据及其意义。
----- <http://gad.qq.com/article/detail/39805>
 - 1)BehaviourUpdate
 - 2)Animator.Update
 - 3)MeshSkinning.Udpate
5. 当你Instantiate一个Prefab时，以下这几种资源分别是复制还是引用？GameObject，transform，texture，mesh，material。
----- <http://gad.qq.com/article/detail/22992>

复制+引用: mesh，material；
引用: texture；
复制: GameObject transform。
6. 使用C#实现一个冒泡算法示例。
-----[看评论] <http://www.cnblogs.com/tobin/archive/2008/04/28/1174687.html>
7. 不使用自带物理组件，实现一个Ball类，使其按照抛物线运动。
----- https://blog.csdn.net/sinat_20559947/article/details/53389157
8. C# String类型和StringBuilder类型有何差异？
----- <https://www.kancloud.cn/wizardforcel/learning-hard-csharp/111552>
9. 请写出线性插值算法。
----- [参考1] https://blog.csdn.net/gggg_ggg/article/details/41213141
----- [参考2] <https://blog.csdn.net/xbinworld/article/details/65660665>
10. 请简述streamingAssetsPath，dataPath，persistentDataPath的作用，以及各个平台的差异（Window，Android，IOS）
----- <https://blog.csdn.net/qiaoquan3/article/details/53942710>
 - 1)streamingAssetsPath 数据流缓存目录
 - 2)dataPath 数据文件目录
 - 3)persistentDataPath 持久化数据目录
11. 向量的点乘、叉乘以及归一化的意义？
----- [参考1] <http://www.qingkt.com/a/Unitymianshiti/graphic/4810.html>
----- [参考2] <https://www.jianshu.com/p/e726581375b2>
12. 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。而LoadLevelAsync则允许你在后台加载新资源和场景，所以再利用协同，你就可以前台用loading条或动画提示玩家游戏未卡死，同时后台协同处理加载的事宜 Asynchronous同步。