1.git 中的ssh key 有什么作用？

加密传输

众所周知ssh是加密传输。

加密传输的算法有好多，git可使用rsa，rsa要解决的一个核心问题是，如何使用一对特定的数字，使其中一个数字可以用来加密，而另外一个数字可以用来解密。这两个数字就是你在使用git和github的时候所遇到的public key也就是公钥以及private key私钥。

其中，公钥就是那个用来加密的数字，这也就是为什么你在本机生成了公钥之后，要上传到github的原因。从github发回来的，用那公钥加密过的数据，可以用你本地的私钥来还原。

如果你的key丢失了，不管是公钥还是私钥，丢失一个都不能用了，解决方法也很简单，重新再生成一次，然后在github.com里再设置一次就行

2.Git与SVN的区别：

1 核心点：Git是分布式的，而SVN不是分布式的

2 Git有Local Repository 被你的代码库，不用服务端也可以Commit

3.什么是C#的序列化

1.C#中[Serialized]和[NonSerialized]是用来干嘛的？

0 [Serialized] 用来标定一个类可以被序列化；

1 反之[NonSerialized]用来标定一个类不能被序列化；

2 可以被序列化是什么意思？

就是指一个类可以的对象可以被转换成容易传输的格式的过程，这里的容易传输的格式具体指的是：二进制流，流文件（一组连续的一维数据包含开头和结尾）

4.C# 委托与事件的详细解析

2.C#中的委托（委托类型的使用方法（方式，写法））

1)定义委托类型 public delegate int Call(int numl,int num2);

2)声明委托对象 Call objCall

3)创建委托对象，将方法与委托关联起来

objCall=new Call(这里面是一个方法)

4)向委托增加关联的方法或者减少关联的方法（+=/-=）

2.1)C#内置的委托类型有哪些？

Func<> 和 Action<>

区别：

Func<T1，T2,TResult>是有返回值的委托

Action<T1,T2,T3>是没有返回值的委托.

3.C#中的事件

需要在委托的基础之上去定义和使用，可以将其理解为：事件是委托的封装，将委托封装成属性之后就是事件,其用法和委托如出一辙.

5.为什么要用委托与事件？

0 在一些等待的过程当中，某个对象需要一直监听某个消息的时候使用（例如：大王A设下一个鸿门宴，命令B,C两队人马埋伏在屏风的后面。若大王左手举杯，则B队人马杀出来；若大王右手举杯，则C队人马杀出来；若大王摔杯，则BC两队人马同时杀将出来。乍一看这个可以使用if语句来判断，之后等待条件成立即可；问题是如此一来B，C需要时刻盯着大王的手去看，if判断也需要不停的去执行，这显然是不合理的；如何解决这个问题就是为什么要使用委托的答案）

1 委托 ，它代表的必定是某一个方法，可以理解为将一个方法给委托出去，而被委托出去的方法必定有与之关联的其他方法，一般在其他的类里面.

5.Unity中的DontDestroyOnLoad()所带的缺陷是什么？

6.GitExtensions创建分支的实质是什么？

7.GitExtensions stage的实质是什么？

8.Unity Camera里面各种选项所对应的意思和实质是什么？

9.C# where 泛型类型约束--GoogleC#相关

10.虚方法和抽象类的区别是什么？

7)C# 虚方法的使用

①可以使用基类调用派生类的方法--使用基类对象.方法名调用的是派生类的实现

②满足条件：基类使用virtual标注，派生类使用override标注覆盖，访问修饰符相同.

③多重派生情况下：使用基类.方法名：调用的是最高级别派生类的方法实现.

7.1)为什么要使用虚方法?

答：虚方法是在继承的前提下，多态的一种表现形式。不同的对象调用同一个方法，就会有不同的表现行为.

11.C#静态家族---- 静态变量，属性，方法的作用以及为什么要使用静态变量

12.Unity中异步加载场景和同步加载场景的区别？几种场景加载方式的区别是什么？已知的常用的场景加载方式是什么？场景加载进度的获取方式是什么？场景加载状态的获取方式是什么？

13.C# 中的this

14.单例模式的实质？已有单例写法为什么会成为单例？单例的优缺点？

15.UGUI 给按钮绑定点击事件的4种方法

16.Get Set 方法深入理解--博客

17.Windows软连接是个啥？

18.Stack.Pop() 与Stack.Peek()的区别

19.C# 中的Var和Dynamic

20.找时间测一下Unity中Time类的详细意义.

21.param关键字

22.C# out 关键字的用法：

通常一个方法只能返回一个值，但是如果在某些时候，我们想要返回多个值，例如某个方法将一个浮点数分割成一个整数和一个小数返回去。这个时候我们就要用到out关键字。

23.C#中的readonly:

readonly 关键字是可以在字段上使用的修饰符。当字段声明包括 readonly 修饰符时，该声明引入的字段赋值只能作为声明的一部分出现，或者出现在同一类的构造函数中.

-->readonly 与const之间的区别

24.C#里面的反射机制是什么样子的？

25. int? firstX=points?.FirstOrDefault()?.X

对于运算符 ?. --如果运算符前面的对象为NULL，则不进行后面的获取成员的运算，直接返回NULL

26.语法糖的好处和坏处是什么？

27.C# Invoke和Unity Invoke的区别是什么？

28..NetCoreSDK是个啥？？

29.对象池的再理解