**1. 解释一下Rigidbody，组建中is kinematic，参数在什么情况下使用？**

答：RigidBody组件的Is Kinematic属性打上勾（设为true，使其不受物理引擎驱动，Wall是为了防止其移动，Person是为了防止其受到力不断旋转—看的）

**2. 编写一个脚本对刚体的几种常用方法**

答：

//创建刚体引用： private Rigidbody cube1Rigidbody; // Use this for initialization void Start () { //获取刚体对象： cube1Rigidbody = GetComponent<Rigidbody>(); }注意上面的代码，引用在成员变量中创建，而对象则在start（）中获取。这么写的具体原因需要对C#有一定的了解。

如果要在游戏中对刚体进行进一步的操作，则需要在update（）中写入代码进行控制，例如：

void Update () { cube1Rigidbody.AddForce(new Vector3(1,0,0)); }cube1Rigidbody

为之前创建的成员变量，如果定义在其他方法内的话将无法访问到。AddForce（）是刚体对象Rigidbody的一个方法，表示施加力，Vector3是向量类，这里表示创建了一个3维向量（1,0,0）。整句代码的作用是对刚体施加x方向1牛顿的作用力。

如果要将操作绑定至键盘上，我们可以这样写：

//检测键盘水平输入，默认为AD及左右键 float h = Input.GetAxis("Horizontal"); cube1Rigidbody.AddForce(new Vector3(h,0,0));Input是一个输入对象，GetAxis表示获取键盘输入，Horizontal是水平的意思，即获取键盘水平上的输入，一般默认为AD或左右键。这个方法最终会返回一个float类型的数字，我们将这个数字代入到vector中，变可以通过键盘输入来控制刚体的受力了。同理，加入垂直方向上的控制只要在多加一句相似的代码即可：

float v = Input.GetAxis("Vertical");

即获取垂直方向上的输入，默认WS和上下键，吧v替换到vector中的第二项即可。当然，这样不好控制物体的受力变化，我们可以对vector进行乘操作来控制它，当然这样也是不方便的，我们还能在整个类的成员变量中加入一个公有的变量来控制，使用public定义的成员变量可以和unity编辑界面绑定。

**3.了解Unity3D游戏引擎自带的规则碰撞器，并导入一个模型为其添加合适的碰撞器**

Sphere Collider（球形碰撞器）： 可以调整半径大小。

Capsule Collider（胶囊碰撞器）： 可以调整半径、高度和方向，即使是圆柱体也可以用这个做近似的碰撞效果。

Wheel Collider（轮子碰撞器）： 可以调整虚拟质量、半径高度、悬浮距离、弹力、阻力等等参数，主要用于附加在类似于轮胎这种对象上，模拟驱动力的产生，同时提供碰撞范围，

Mesh Collider（网格碰撞器）： 这是个强力的碰撞器，用它能创造出不规则的碰撞范围，但需要的性能消耗也比较大。所以虽然像前面的树的模型本应该使用网格碰撞器，但是一般场景中的树木数量都比较多，所带来的性能开销也相当可观，所以官方也建议用胶囊作为树木的碰撞器。只有在的确需要不规则碰撞的时候再用此神器。

Terrain Collider（地形碰撞器）： 为什么我们的第一人称视角控制器不会从“地面”上往下坠？选中游戏中的地形对象，就能发现它的身影啦。 注：在官方文档中有解释地形的Create Tree Collider（创建树木碰撞器）选项，说是勾选这个，就可以自动为用地形画刷刷出来的树木创建碰撞器，但是笔者尝试了一下貌似无效，不知道哪里设置错了。