1. 解释一下Rigidbody组件中Is Kinematic参数在什么.情况下使用。

答：rigidbody刚体是使GameObject游戏对象具有物理行为的主要组件。附着了刚体的GameObject, 将立刻受到重力的影响。

在某些情况下，可能希望GameObject拥有刚体，而不受物理引擎的控制。例如，您可能希望直接从脚本代码控制字符，但仍允许其由触发器检测。这种从脚本产生的非物理运动称为运动学上的运动。

Rigidbody刚体组件具有称为Is Kinematic("运动学(上)的")的属性，将GameObject从物理引擎的控制中移除，并允许其从脚本运动地移动，可以从脚本中更改Is Kinematic的值，以允许对象打开和关闭物理，但这伴随着性能开销，应该谨慎使用。

刚体（Rigidbody）是Unity物理引擎中的重要组件，一个游戏对象添加了刚体组件以后就开始接受物理力学的影响了，比如重力、空气摩擦力、碰撞产生的推力。少数情况下，你可能想在脚本中控制一个添加了刚体组件的对象的运动状态，但却不是通过施加力的方式，有一种方法可以实现，就是在刚体组件中勾选is kinematic。启用了kinematic后，刚体组件就应该被称为运动学刚体了，刚体因此而失去了对物理力学的影响，比如说不再受重力的影响，不在碰撞力下移动。好处是这个对象能够触发碰撞函数，坏处是产生了一些额外的性能开销。

1. 编写一个脚本对刚体的几种常用方法进行测试(脚本源码极其注释)。

答：（1）基本代码。

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class cube1 : MonoBehaviour {

// Use this for initialization

void Start () {

}

// Update is called once per frame

void Update () {

}

（2）获取刚体。

//创建刚体引用：

private Rigidbody cube1Rigidbody;

// Use this for initialization

void Start () {

//获取刚体对象：

cube1Rigidbody = GetComponent<Rigidbody>();

（3）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。

void Update () {

cube1Rigidbody.AddForce(new Vector3(1,0,0));

}

（4）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));

（5）其他测试。

using UnityEngine;

using System.Collections;

public class cubeScript : MonoBehaviour {

private Rigidbody rb;

// Use this for initialization

void Start () {

rb = GetComponent <Rigidbody>();

}

// Update is called once per frame

void Update () {

if (Input.GetKeyDown (KeyCode.Alpha1)) {

// 给当前游戏对象添加一个力

// rb.AddForce (new Vector3(0.0f, 10.0f, 0.0f));

// 给当前游戏对象添加 力矩/扭矩，扭矩可以使物体旋转

// rb.AddTorque (new Vector3(0.0f, 10.0f, 0.0f));

// 给当前游戏对象在指定的位置上添加一个力

// rb.AddForceAtPosition (new Vector3(0.0f, 10.0f, 0.0f), new Vector3(0.5f, 0.5f, 0.5f));

// 给当前游戏对象在某个点添加一个爆炸力,不能持续添加，只能炸一下

rb.AddExplosionForce (1500.0f, Vector3.zero, 5.0f);

}

}

}

1. 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(回答有哪些碰撞器)。

答：Box Collider（立方体碰撞器）：点击它，在场景视图中的立方体就会出现一个绿色的轮廓，这就是碰撞器的真身了。

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。 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。