

常用获取物体对象的方法和属性

GameObject

方法	说明
GameObject.Find(string name)	根据名称查找游戏对象。如果找到游戏对象，则返回该游戏对象，否则返回null
GameObject.FindWithTag(string tag)	根据标签查找游戏对象。如果找到游戏对象，则返回该游戏对象，否则返回null
GameObject.FindGameObjectWithComponent(Type component Type)	根据组件类型查找游戏对象。如果找到游戏对象，则返回该游戏对象，否则返回null
GameObject.FindGameObjectsWithComponent(Type componentType)	根据组件类型查找所有游戏对象。返回一个包含所有找到的游戏对象的列表
GameObject.FindChild(string name)	根据名称查找子游戏对象。如果找到子游戏对象，则返回该子游戏对象，否则返回null
GameObject.GetComponent(Type componentType)	获取游戏对象的组件。如果游戏对象具有该类型的组件，则返回该组件，否则返回null
GameObject.GetComponentInChildren(Type componentType)	获取游戏对象及其子游戏对象中的组件。如果游戏对象或其子游戏对象具有该类型的组件，则返回该组件，否则返回null
GameObject.GetComponentInParent(Type componentType)	获取游戏对象的父游戏对象中的组件。如果游戏对象的父游戏对象具有该类型的组件，则返回该组件，否则返回null

Transform

方法	说明
Transform.Find(string name)	根据名称查找游戏对象。如果找到游戏对象，则返回该游戏对象的Transform组件，否则返回null
Transform.FindWithTag(string tag)	根据标签查找游戏对象。如果找到游戏对象，则返回该游戏对象的Transform组件，否则返回null
Transform.FindGameObjectWithComponent(Type comp	根据组件类型查找游戏对象。如果找到游戏对象，则返回该游戏对象的Transform组件，否则返回null

方法	说明
onentType)	
Transform.FindGameObjects WithComponent(Type comp onentType)	根据组件类型查找所有游戏对象。返回一个包含所有找到的游戏对象的列表，并返回每个游戏对象的Transform组件
Transform.FindChild(string n ame)	根据名称查找子游戏对象。如果找到子游戏对象，则返回该子游戏对象的Transform组件，否则返回null
Transform.GetComponent(Ty pe componentType)	获取游戏对象的组件。如果游戏对象具有该类型的组件，则返回该组件的Transform组件，否则返回null
Transform.GetComponentIn Children(Type componentTy pe)	获取游戏对象及其子游戏对象中的组件。如果游戏对象或其子游戏对象具有该类型的组件，则返回该组件的Transform组件，否则返回null
Transform.GetComponentIn Parent(Type componentTyp e)	获取游戏对象的父游戏对象中的组件。如果游戏对象的父游戏对象具有该类型的组件，则返回该组件的Transform组件，否则返回null
Transform.GetChild	根据子物体的序号查找子物体，参数子物体的序号,从0开始 返回值:transform
Transform.GetSiblingIndex	获得该物体在兄弟节点之间的序号,参数:无 返回值:序号。 获得本物体在兄弟物体之间的序号。
Transform.Ischildof	判断是否是另一个物体的子物体，参数:其它物体的transform 返回值:true或false
Transform.parent	获取该物体的父物体
Transform.root	获取最上一级的父物体。
transform.childCount	该物体的子物体总数，不包括子物体的子物体。与transform.GetChild组合使用可以用来遍历所有子物体。

C#特性: [SerializeField]与[HideInInspector]

- SerializeField意思是可序列化字段，HideInInspector意思是在属性窗口中隐藏
- 在字段前面加上[SerializeField]，则变量会出现在属性面板中
- 在字段前面加上[HideInInspector]，则变量不会出现在属性面板中
- 适用于想在Unity属性面板中通过拖曳方式赋值，又不希望为public

输入系统 Input System

- Unity默认输入定义在 编辑 ➡ 项目设置 ➡ 输入管理器
- Input.GetAxis(按键名称) 轴输入，一般用于手柄
- Input.GetButton, Input.GetButtonDown, Input.GetButtonUp 一般用于按钮和键盘的输入

Transform移动

- 每帧改变物体的位置，就能让物体连续移动起来

//常见写法有

```
transform.position += new Vector3(0,0,1) * Time.deltaTime; //游戏空间的坐标方位  
transform.Transform(new Vector3(0,0,1)) * Time.deltaTime); //游戏物体局部的坐标方位
```

- 其中，Time.deltaTime代表上一帧到当前这一帧的时间间隔，乘以deltaTime的目的是当帧率不稳定时，让物体每秒的移动速度为恒定值

创建和销毁物体

Instantiate方法

- 使用预制体

添加组件

- gameObject.AddComponent

销毁组件或物体

- Destroy(物体组件);
- Destroy(组件对象);
- Destroy(对象, 延迟时间);

刚体运动方法

- transform移动是一种最直接的移动方式，强行改变了物体位置
- 如果角色是刚体（具有Rigidbody组件），可以通过修改“力”和“速度”间接让物体运动起来，这种运动更自然，更容易达到自然的效果

//常见写法有

```
rigid.AddForce(new Vector3(0,0,1)); //施加力，通过加速度修改速度  
rigid.velocity=new Vector.3(g,,1); //直接修改速度
```

