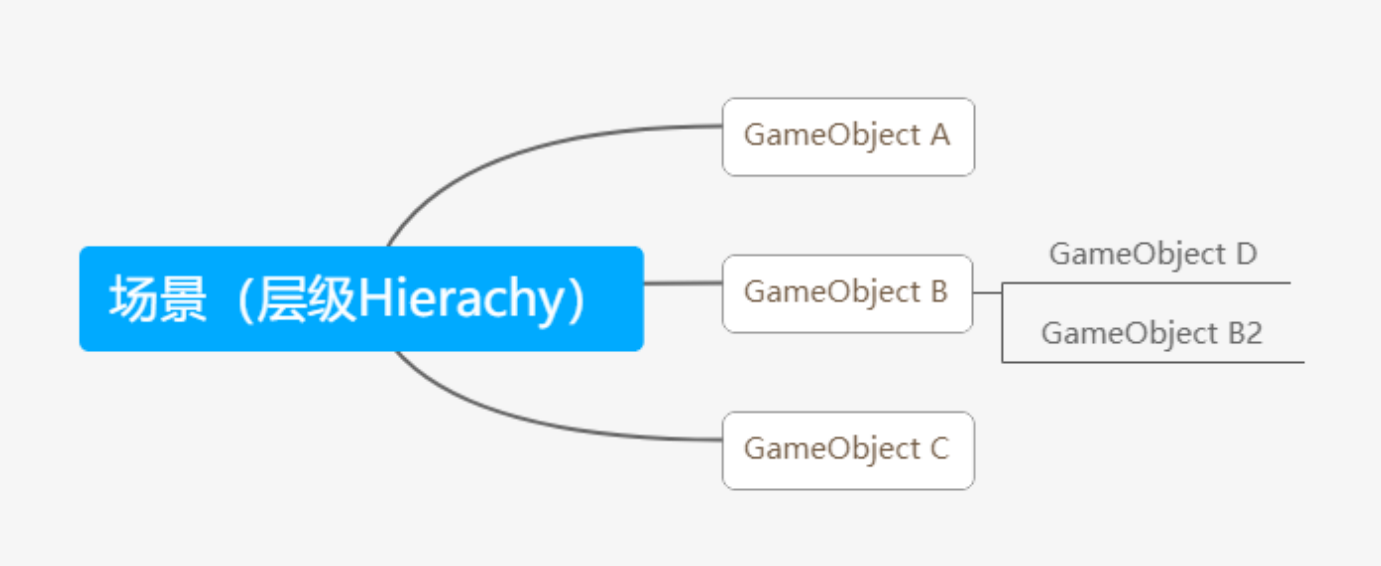
Unity基本架构

场景Scene

- 一个场景可以包含多个游戏物体

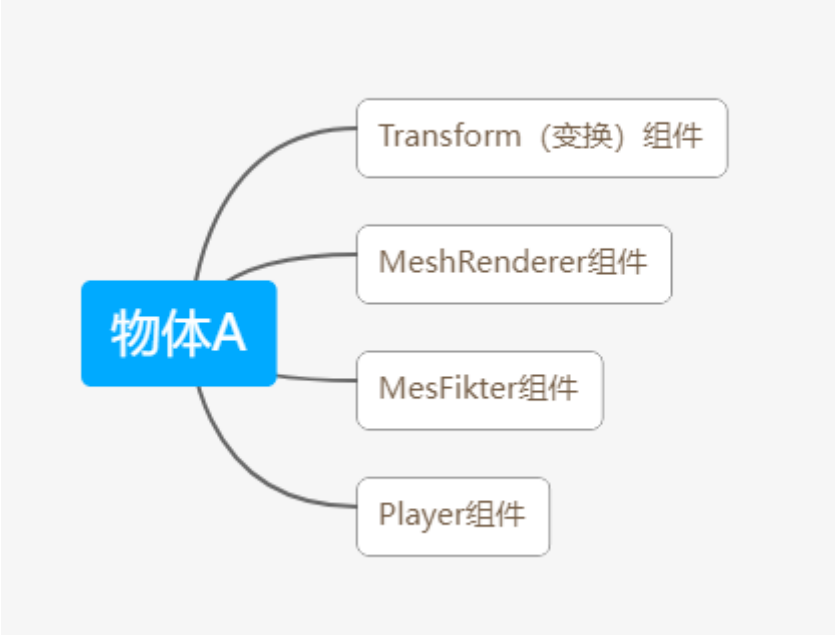


游戏物体(具有父子关系)

- 这些物体具有父子关系
- 子物体在父物体的坐标系下，父物体发生改变子物体也要发生改变

组件

- 组件是实现具体功能的最小单位
- 组件时挂载在物体上的
- 每个物体有多个组件



组件

组件的功能

- Transform变换组件，负责实现物体的位置、旋转、缩放、父子关系
- 一个3D模型，有MeshFilter提供三维网格，有MeshRenderer网格渲染器，才能被渲染、被看到
- 一个物体具有物理阻挡的效果，是因为它有碰撞体Co11ider组件
- Rigidbodyl刚体组件，可以让物体被物理系统控制，受到力的作用

组件的数量

- 如果挂载多个，都会生效
- 某些组件限定了只能挂载一个，比如刚体组件

脚本组件——一种特殊的组件

- 为了让脚本能更好的和Unity引擎融合，Unty提供了“脚本组件”这一概念
- 符合一定条件的class就可以作为组件 继承MonoBehaviour,它是脚本组件的基类 文件名与c1ass名称相同
- 脚本组件是一项很好的创新，它让你自己实现的功能，与引擎提供的功能有同等 地位
- 也可以有不是组件的脚本

获取组件对象

- 获取组件的方法

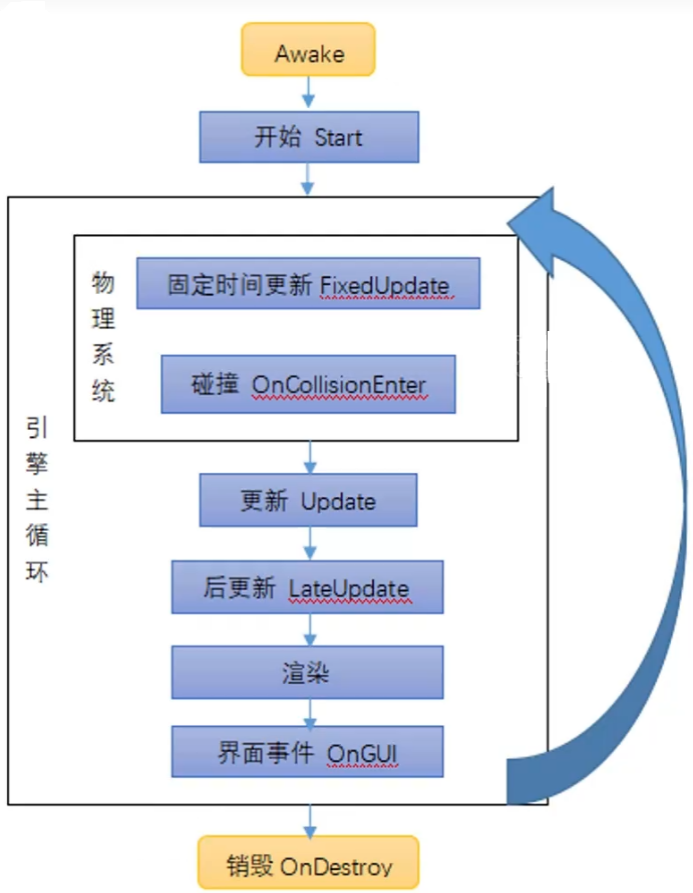
```
//物体.GetCompoent<组件类名> 或 组件.GetComponent<组件类名>

rigide = GetComponent<Rigidbody>();//获取挂载对象的刚体组件(使用组件方法)
rigide = this.gameObject.GetComponent<Rigidbody>();//与上面相等，获取挂载对象的刚体组件(使用对象方法)
```

- 通过transform可以随时获得到变换组件，不需要GetComponent();
- 总是可以用某个组件对象代指一个物体

MonoBehaviour的生命周期

- 脚本作为一个组件，有自己的"生命周期"
- 脚本的生命周期与物体、引擎息息相关
- 脚本跟随着引擎的处理，在遇到各种事件时，引擎都会通知脚本（调用函数）进行处理
- 开发者的任务就是找到合适的事件，编写合适的代码



获取物体对象

- 获取物体对象方法

```
//通过名称获取物体GameObject.Find(名字) 无法获得被禁用的物体

GameObject banana = GameObject.Find("Banana");//查找香蕉物体
```

```
//通过标签查找物体：Tag 无法获得被禁用的物体

GameObject banana = GameObject.FindGameObjectWithTag("Player");//查找标签为Player的物体
//FindGameObjectWithTag()和FindGameObjectsWithTag()的区别
//FindGameObjectWithTag()方法返回的是一个单个的游戏对象
//FindGameObjectsWithTag()方法返回的是一个游戏对象数组
```

```
//通过transform.Find(子物体名称) 可以获得被禁用的物体

GameObject canvas = GameObject.Find("Canvas");//查找父物体
Transform trans = canvas.transform.Find("BagPanel");//查找子物体
```