什么是Fratcal：Fratcal是分形，分形简单来说是大的物体由相同形态的更小的物体组成，比如一个大的四棱锥由等比例缩小的四棱锥组成，同时小的四棱锥又由更小的四棱锥组成……

Start是在执行Awake和OnEnable之后的下一帧执行。即用Instantiate创建物体之后，在本帧执行物体的Awake和OnEnable，在下一帧开始执行Start和Update

使用Instantiate复制物体的时候，同时也会复制该物体下的所有子物体

Unity中默认球体Sphere包含许多顶点，远多于方块Cube。所以使用Shpere分形在深度小于等于6时很流畅，当高于这个深度时就会严重影响帧率。

Mesh：是指模型的网格，建模就是建网格。细看Mesh，可以知道Mesh的主要属性内容包括顶点坐标，法线，纹理坐标，三角形绘制序列等其他有用属性和功能。因此建网格，就是画三角形；画三角形就是定位三个点。

Mesh Filter：内包含一个Mesh组件，可以根据MeshFilter获得模型网格的组件，也可以为MeshFilter设置Mesh内容。

Mesh Render：是用于把网格渲染出来的组件。MeshFilter的作用就是把Mesh扔给MeshRender将模型或者说是几何体绘制显示出来。

Unity中的顺规是zxy

用循环代替递归能大幅度优化性能。

TSR是什么意思？

它代表平移-旋转-缩放（translation-rotation-scale）。在此上下文中的平移意味着定位或偏移。

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

Jobs System的思想是利用CPU的多核和特殊的SIMD指令（代表单指令多数据）来尽可能有效地利用CPU的并行处理能力。这是通过将工作定义为单独的片来实现的。这些Job的编写方式与常规C＃代码类似，但是随后通过Unity的Burst编译器进行编译，该编译器通过执行常规C＃所没有的一些结构性约束而实现了积极的优化和并行化。

Job无法与对象一起使用，仅允许使用简单值和结构类型。它仍然可以使用数组，但是我们必须将它们转换为通用NativeArray类型。

不必每次迭代都显式调用Execute方法。我们可以安排Job，以便它自己执行循环。这是通过调用带有两个参数的Schedule来完成的。第一个是我们想要的迭代次数，它等于我们正在处理的parts数组的长度。第二个是JobHandle结构值，用于强制作业之间的顺序依赖性。最初，我们将使用默认值，该默认值不强制执行任何约束。Schedule不会立即运行该作业，而只是安排它以供以后处理。它返回一个JobHandle值，该值可用于跟踪作业的进度。我们可以通过在句柄上调用Complete来延迟代码的进一步执行，直到作业完成为止。

将所有作业捆绑在一起以仅等待最后一个作业的完成，这样做的好处是可以延迟等待完成。一个常见的示例是在LateUpdate方法中安排Update中的所有作业，执行其他操作并延迟调用Complete，这是在所有常规Update方法完成后调用的。也可以将完成延迟到下一帧甚至更晚。

Vector3和float3类型之间以及四元数和四元数类型之间存在隐式转换

我们可以将单个作业的工作分散到多个CPU内核上。这是通过在Job而不是Scedule上调用ScheduleParallel来完成的。

[Unity项目性能优化:多线程Jobs计算并绘制大量物体\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV14s4y1472X/?vd_source=ba280f9b8e4815b5dd13c1f1f6db3155)

文本

描述已自动生成