# 小立方体堆叠成大立方体

## 1.背景

为了让堆叠效果更加逼真自然，达到较好的动态效果，这里借助物理引擎进行运动仿真。记录运动过程中每个小立方体的方位信息，生成动画文件。播放时，加载对应动画文件，生成动画信息，并应用到各个小立方体，形成堆叠效果。

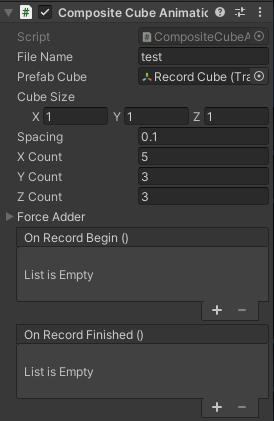
## 2.录制

### 2.1环境搭建

根据项目使用情景搭建对应场景元素。配置元素的物理属性，包括碰撞体、物理材质以及刚体等等。

### 2.2录制组件

应用物理、录制动画过程并生成动画文件。录制组件面板如下：

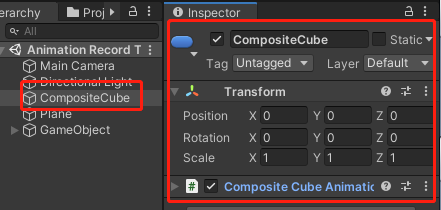


参数：

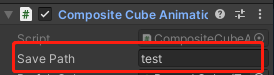
|  |  |
| --- | --- |
| **File Name** | 存储文件名。根路径为StreamingAsset |
| **Prefab Cube** | 立方体预制 |
| **Cube Size** | 立方体大小，以米为单位 |
| **Spacing** | 立方体间距 |
| **X Count** | X轴立方体数量 |
| **Y Count** | Y 轴立方体数量 |
| **Z Count** | Z 轴立方体数量 |
| **Force Adder** | 施加在立方体上的力组件集合 |
| **OnRecordBegin** | 开始录制事件通知 |
| **OnRecordFinished** | 录制结束事件通知 |

### 2.3录制过程

* 新建一个GameObject并调整对应方位
* 挂载动画组件



* 路径设置



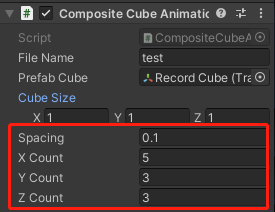
由于录制是生产过程，用在编辑态下，故根目录设在：Asset/StreamingAssets。

* 小立方体预制设置

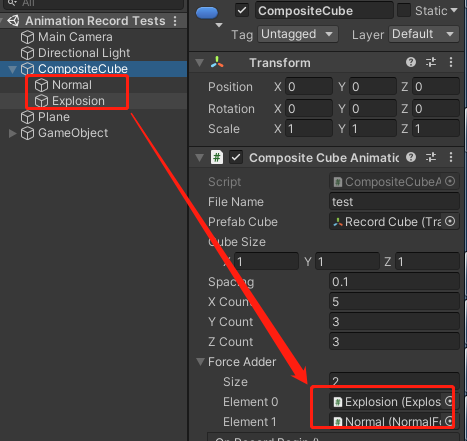


定义小立方体预制以及尺寸。小立方体在运行时生成。

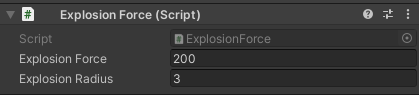
* 小立方体个数以及间距设置



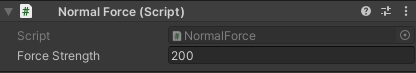
* 关联力组件



运动过程中需进行力的施加。为了进行力的可视化调整，快速迭代验证效果，这里将常用的添加力方式进行组件化（对应Rigidbody组件中的AddForce以及AddExplosionForce）。若提供的组件不能满足要求，可以自定义组件。以下是对应组件面板：



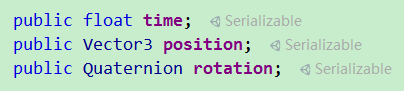
对应Rigidbody.AddExplosionForce



对应Rigidbody.AddForce

### 2.4存储

要完整记录一个立方体的方位信息，必须包含位置、定向以及缩放。总共包含10个字（10 channel），为了节约内存使用以及目前的动画需求，可以忽略缩放。故目前存储的方位信息只包含位置和定向。方位信息附加时间戳形成动画关键帧信息。



考虑到数据的体量，采用二进制格式进行存储，提升处理速度及文件大小。

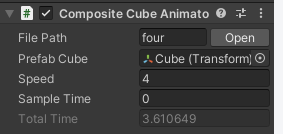
## 3.播放

### 3.1环境搭建

根据项目使用情景搭建对应场景元素。特别的：不需要物理引擎模块的配置

### 3.2 播放组件

读取文件，生成动画片段并进行播放。组件如下所示：



参数：

|  |  |
| --- | --- |
| **File Path** | 文件路径（支持相对路径。若是相对路径，根目录为StreamingAsset） |
| **Prefab Cube** | 立方体预制 |
| **Speed** | 播放速率 |
| **Total Time** | 总时长（只读） |
| **Sample Time** | 采样时间 |

为方便Timeline的编辑，改组件可以在编辑态下运行。

### 3.3 播放过程

* 预制设置



* 动画文件路径



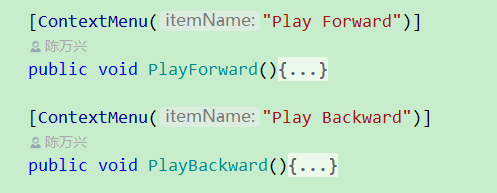
* 动画速率调整



* 定位到某一时间点



* 调用API进行播放



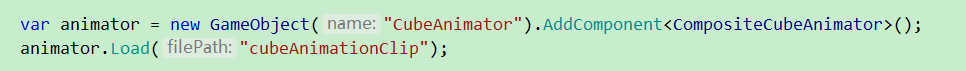
### 3.4 API使用

|  |  |
| --- | --- |
| **void Load(string filePath)** | 加载动画文件 |
| **void PlayForward()** | 大立方体打散成小立方体 |
| **void PlayBackward()** | 小立方体堆叠成大立方体 |

* 加载动画文件

void Load(string filePath);

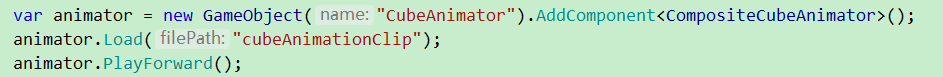
代码示例：



* 大立方体打散成小立方体

void PlayForward()

代码示例：



* 小立方体堆叠成大立方体

void PlayBackward()

代码示例：

