# 1.Mesh，材质纹理以及MeshRenderer的作用

|  |
| --- |
|  |

# 2.MeshRenderer的主要参数详解，开启阴影投影

|  |
| --- |
|  |

# 3.MeshRenderer如何切换材质

|  |
| --- |
|  |

# 4.演练

## 1.新建一个项目起名：classTen-MeshRenderer，然后在assets文件夹里面新建3个文件夹：AssetsPackage,Scenes和Scripts，然后把当前场景保存到Scenes文件夹里面起名main

|  |
| --- |
|  |

## 2.在场景在新建一个立方体

|  |
| --- |
|  |

## 3.点击Cube节点，在右边的属性寄检查器里面有一个MeshRenderer属性，默认是勾选上的，如果我们把它取消勾选，你会发现，此时立方体不见了

|  |
| --- |
|  |
|  |

## 4.勾选上MeshRenderer，立方体又重新出现了

|  |
| --- |
|  |

## 5.点击右边的lightmapSettings,发现它又一些选项如是投射阴影还是接收阴影，阴影是否开源烘培，还有灯光映射的大小，默认是64

|  |
| --- |
|  |

### 下面有一个显示投射方式的选项默认是关闭的，还有一个接收阴影选项默认是开启的

## 6.下面我们来给MeshRenderer设置一个自定义材质，我们在AssetsPackage文件夹里面新建一个Mats文件夹

|  |
| --- |
|  |

## 7.然后Mats在里面新建一个材质，起名叫做red

|  |
| --- |
|  |
|  |

## 8.我们把材质的形状维持默认形

|  |
| --- |
|  |

## 9. 勾选USE\_ALBEDOMAP选项，然后map选择back，然后点击√按钮

|  |
| --- |
|  |
|  |

## 10.然后我们取消勾选这个选项，点击Albedo,把颜色改为红色

|  |
| --- |
|  |

### 点击应用按钮，就创建了一个红色材质

## 11.点击立方体节点，把它red材质拖拽到他的属性面板中MeshRenderer的Materials一栏中，就会把他的材质设置为红色材质

|  |
| --- |
|  |

### 当然，也可以直接拖拽到场景中的立方体上面

## 12.默认cocos creator的使用阴影功能是关闭的，如果我们需要这个功能可以把它打开，点击场景节点，然后勾选shadow一栏下面的复选框

|  |
| --- |
|  |

## 13.为了更好的观察阴影我们在场景在创建一个平面物体

|  |
| --- |
|  |

## 14.把光源设置成垂直于平面的方向

|  |
| --- |
|  |

## 15.我们做了这些设置后，还是看不见阴影，因为还有设置要做，我们需要给平面节点启用接收阴影功能，我们来看一下平面节点

|  |
| --- |
|  |

## 16，我们发现他的接收阴影设置是开启的。然后我们需要看看立方体的投射阴影功能是否开启，我们点击一下立方体节点

|  |
| --- |
|  |

## 17，发现他的ShowCastingMode是关闭的，我们把它打开并且把立方体往上拉一段距离，这样子就可见阴影了

|  |
| --- |
|  |

## 18仔细观察这个阴影，你会发现它比较怪异，

|  |
| --- |
|  |

## 19，我们需要回到场景节点，在shadows里面的type一栏里面，默认是Planar，我们选择ShadowMap，效果就好多了

|  |
| --- |
|  |

# 我们来学习一下通过代码来修改物体的材质

## 1.场景一个新的绿色材质

|  |
| --- |
|  |

## 2.获取材质资源又2种方法，第一种是用代码加载资源文件，第二种是编辑器绑定材质资源，我们这里使用第二种方法，在Scripts文件夹里面新建一个ChangeMat脚本，然后提交到Cube节点上面

|  |
| --- |
|  |

## 3.双击这个脚本用vscode打开，然后把多余的内容删除

|  |
| --- |
|  |

## 4.个组件类添加一个数据成员，然后给他添加@property装饰器。此时你就可以在编辑器里面看到这个数据成员

|  |
| --- |
|  |

## 5.回到编辑器，已经能够看到这个数据成员了

|  |
| --- |
|  |

## 6.然后我们可以把绿色材质拖拽到这个数据成员上面

|  |
| --- |
|  |

## 7.此时材质已经绑定好，我们回到代码，在这里先找到MeshRenderer组件实例，然后使用他的setMaterial方法来设置材质

|  |
| --- |
|  |

### 注意：材质和Prefab有点不一样，材质不需要实例化代码，运行程序效果如下

|  |
| --- |
|  |

### 材质替换成功