



## 5.1 初识 unity



### 5.1.1 Unity 简介

Unity 是由 Unity Technologies 公司开发的一个多平台的综合型游戏开发工具,是一个全面整合的专业游戏引擎。2004 年,Unity 诞生于丹麦的阿姆斯特丹,2005 年发布了 Unity 1.0 版本。起初应用于 MAC 平台,主要针对 Web 项目和 VR 的开发。2008 年推出 Windows 版本,并开始支持 iOS,才逐步从众多的游戏引擎中脱颖而出,并顺应移动游戏的潮流而变得炙手可热。2010 年,Unity 开始支持 Android,继续扩散影响力。2012 年 Unity 上海分公司成立,正式进军中国市场。

Unity 不仅限于游戏行业,在虚拟现实、增强现实、工程模拟、3D 设计、建筑设计展示等方面也有着广泛的应用。国内使用 Unity 进行虚拟仿真教学平台、房地产三维展示等项目开发的公司非常多,例如绿地地产、保利地产、中海地产、招商地产等大型房地产公司的三维数字楼盘展示系统,很多都是使用 Unity 进行开发。

Unity 提供强大的关卡编辑器,支持大部分主流的 3D 软件格式,使用 C# 或 JavaScript 等高级语言实现脚本功能,使开发者无须了解底层复杂的技术,快速地开发出具有高性能、高品质的交互式产品。

随着 iOS、Android 等移动设备的大量普及和虚拟现实在国内的兴起,Unity 因其强大的功能、良好的可移植性,在移动设备和虚拟现实领域得到了广泛的应用和传播。

安装 Unity 可以使用 Unity Hub 管理程序。

第一步: 下载 Unity Hub。下载地址为 <https://store.unity.com/cn/download>, 下载时需要满足网页中显示的几个条件,并注册账户。

第二步: 安装 Unity Hub 完成之后打开,选择需要安装的 Unity 版本,单击下载即可。

### 5.1.2 Unity 项目框架

Unity 项目文件包含多个场景,项目运行时,可以在这些场景间切换。每个场景中可以创建多个游戏对象,场景由游戏对象组成。游戏对象的特性和功能被细分成不同的组件,选择不同的组件就可以组合出不同的游戏对象。脚本也是一种组件,游戏对象需要挂载脚本,添加相应的脚本即可。Unity 项目的框架结构如图 5-1 所示。

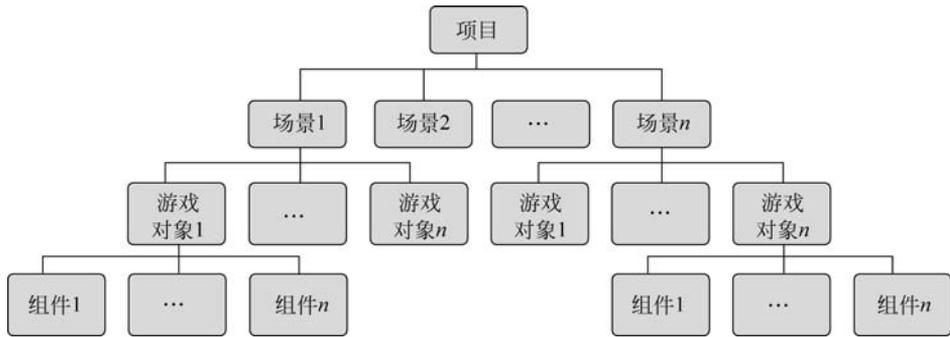


图 5-1 Unity 项目的框架结构

## 5.2 窗口界面

### 5.2.1 场景窗口

开发 Unity 产品,首先需要创建 Unity 项目。Unity 项目创建好后,可以打开 Unity Editor(Unity 编辑器)进行编辑,Unity 编辑器界面包含有 Scene(场景窗口)、Hierarchy(层级面板)、Project(工程面板)、Inspector(检视面板)等,如图 5-2 所示。

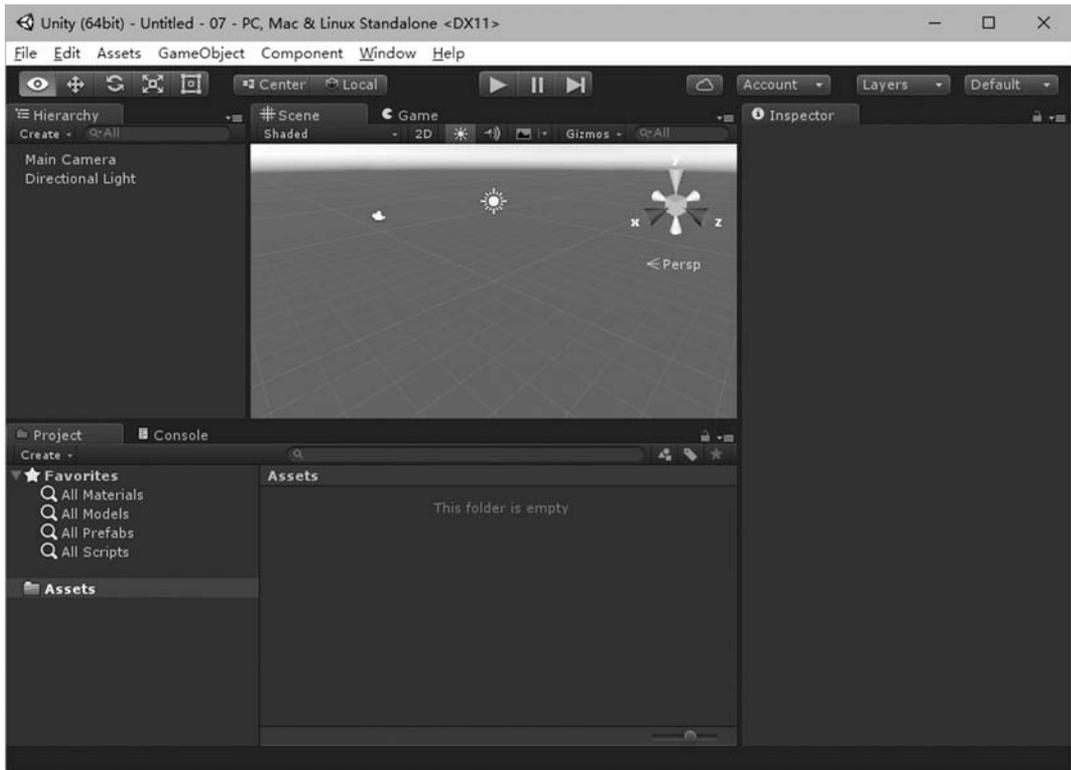


图 5-2 界面组成

一个项目可以包含多个场景,项目运行时,可以在这些场景间切换,每个场景可以创建和编辑多个对象。

### 1. 对象基本变换

#### 1) 操作工具栏

场景操作工具栏包括 5 个按钮,如图 5-3 所示,各按钮功能如下。



图 5-3 工具栏

手形工具,选择后按住鼠标左键用于移动场景,按住鼠标右键用于旋转场景(以自身为轴心),按住 Alt 键+鼠标左键旋转场景(以选中物体的坐标轴为轴心),按住 Alt+鼠标右键缩放场景。

移动工具,用于移动场景中的物体。

旋转工具,用于旋转场景中的物体。

缩放工具,用于缩放场景中的物体。

矩形工具,用于改变物体长款比例。

#### 案例 1: 对象变换操作

实现对象的三种基本变换操作,包括移动、旋转和缩放,分别改变对象的位置、角度和大小。三种基本变换的操作控制框如图 5-4~图 5-6 所示。

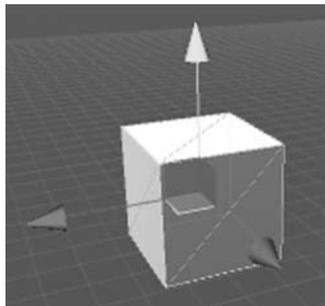


图 5-4 移动

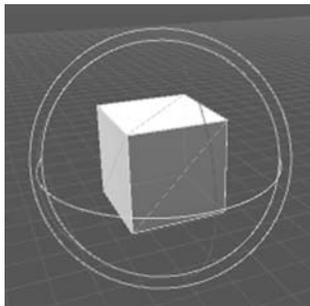


图 5-5 旋转

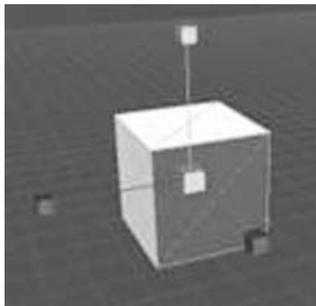


图 5-6 缩放

### 2. 场景漫游

为方便对象的编辑,可以平移、环视、缩放场景视图,使场景中的对象最大化显示,还可以漫游场景。

#### (1) 平移。

① 平移按钮按下后,按住鼠标左键拖动;

② 按住鼠标滚轮(或中键)拖动。

#### (2) 环视。

① 按住鼠标右键拖动;

② Alt+鼠标左键拖动。

#### (3) 缩放。

① 滚动鼠标滚轮;

② Alt+鼠标右键拖动。

#### (4) 聚焦。

即对象最大化,在 Hierarchy 面板中:

① 选中对象,按下 F 键;

② 双击选中对象。

(5) 场景漫游。

按住鼠标右键后,分别按下 W、S、A、D 键,可以实现向前、后、左、右四个方向的漫游。

### 3. 视图控制

场景视图分为 2D 投影视图和 3D 立体视图两大类。

(1) 2D 投影视图: Front 前视图、Back 后视图、Left 左视图、Right 右视图、Top 顶视图、Bottom 底视图。

(2) 3D 立体视图: Perspective 透视图(Persp)、Orthographic 正交视图(Iso)。

视图切换方法: ①可以通过单击场景视图右上角的坐标轴架; ②在坐标轴架的右键菜单中选择相应的视图。

★提示: Game 窗口也叫游戏视图,是摄像机渲染的视图,也就是向开发者展示内容。

## 5.2.2 层级面板

层级(Hierarchy)面板按名称列出了场景中的所有对象,当在场景中创建或删除对象时,Hierarchy 面板将同步更新。对象间存在父子层级关系时,Hierarchy 面板可以清晰地查看对象父子关系。

### 1. 创建 3D 物体

在 Hierarchy 面板右键→3D Object→选择要创建的 3D 物体。

### 2. 删除物体

在 Hierarchy 面板单击要删除的物体→右键→Delete。

### 3. 重命名

在 Hierarchy 面板单击要重命名的物体→再次点击。

### 4. 显示/隐藏物体

在 Hierarchy 面板单击目标物体→勾选或取消图示复选框。

## 5.2.3 项目面板

项目(Project)面板列出了开发者创建或导入的所有资源,包括场景、脚本、模型、材质、贴图、音频、预置对象等,通常这些资源被分门别类地放置在不同的文件夹中,而所有资源又被放置在 Assets 文件夹中。项目资源面板的资源采用与资源管理器中组织方式一样,左侧是树形导航窗格,右侧是浏览窗格。

★提示: 当场景中有相同物体时,将它制作成预制件,这样当物体需要修改时可以批量修改,修改预制件则所有由此预制件生成的物体都会随之变化。生成预制件方法: Hierarchy 面板中选中目标物体→拖到 project 面板中就生成了预制件,生成后物体的颜色会变成蓝色。

## 5.2.4 检视面板

检视面板(Inspector)实现组件的添加、移除和组件属性的查看、编辑。

每个对象都有一个 Transform 组件,当创建一个游戏对象时,会自动为该对象创建 Transform 组件。Transform 组件主要通过 Position、Rotation 和 Scale 属性来控制游戏对象的移动位置、旋转角度和缩放比例,如图 5-7 所示。



图 5-7 Transform 组件

## 5.3 物理引擎

现实生活中的物体遵循自然界的物理现象和物理定律,计算机软件中对物理自然现象的模拟通过物理引擎来实现。物理引擎通过为刚性物体赋予真实的物理属性的方式计算运动、旋转和碰撞反应。

物理引擎的作用,就是使虚拟世界中的物体运动符合真实世界的物理定律,以使项目更加富有真实感。

### 5.3.1 刚体

刚体,在物理学中的定义是形状不会发生改变的理想化模型,即在受力之后其大小、形状和顶点相对位置都保持不变的物体,例如铅球落到地上时其形状是基本不变的。刚体是相对于软体和流体而言的。在虚拟世界中刚体常作为物理模拟的基本对象。

刚体使物体能在物理控制下运动,刚体可通过接受力与扭矩,使物体像现实世界一样运动。

#### 1. 刚体的添加

刚体组件的添加方法有以下三种。

方法一:【Component】→【Physics】→【Rigid body】。

方法二:【Inspector】面板下端,单击【Add Component】按钮,选择【Physics】→【Rigid body】。

方法三:脚本添加。

```
GameObject obj = GameObject.Find("box");
//实例化"box"类型的对象 obj
obj.gameObject.AddComponent<Rigidbody>();
//为 obj 对象添加刚体组件
```

#### 2. 属性设置

Mass:物体的质量。

Drag:阻力,物体移动时受到的阻力,0 表示无阻力,一般钢铁是 0.001,羽毛是 10。

Angular Drag:角阻力,物体旋转时受到的阻力。

Use Gravity:是否使用重力,激活后受重力影响。

Is Kinematic:激活后,不再受物理引擎控制。

### 5.3.2 碰撞器

#### 1. 添加碰撞器

选择物体→[Component]→[Physics]→选择碰撞器。

常见碰撞器:

Box Collider:立方体碰撞器,形状是立方体的碰撞器。

Sphere Collider:球形碰撞器,形状是球形的碰撞器。

Capsule Collider:胶囊碰撞器,形状是胶囊形的碰撞器。

Mesh Collider:网格碰撞器,根据网格形状生成的碰撞器。

## 2. 碰撞检测

虚拟场景中,当主角与其他 GameObject 发生碰撞时,需要进行一些操作或完成一些功能,这时,就需要检测到碰撞现象,即碰撞检测。碰撞检测实现方法如下。

(1) OnCollisionEnter(Collision collisionInfo)

当 collider/rigidbody 进入另一个 rigidbody/collider 时 OnCollisionEnter 被调用。

(2) OnCollisionExit(Collision collisionInfo)

当 collider/rigidbody 离开另一个 rigidbody/collider 时 OnCollisionExit 被调用。

(3) OnCollisionStay(Collision collisionInfo)

当 collider/rigidbody 逗留在另一个 rigidbody/collider 时 OnCollisionStay 被调用。

案例 2: 高空坠物

要点: 物体因重力作用,从高空落下,碰到物体发生了不同的碰撞效果。

(1) 在 Hierarchy 面板中,单击 Create→3D Object→Plane,添加组件 Plane,作为场景的地面,在 Inspector 中,设置 scale 为 2,2,2。

(2) 同样的方法,添加组件 Cube,作为坠物。在 Inspector 中,修改 Position 中高度 Y 的值为 5,Rotation 中三个变量的值设置为任意值,使物体旋转一定的角度。

(3) 在 Project 面板中,单击 Create→Material,新建材质,命名为 CubeMaterial。选择 CubeMaterial 材质,在 Inspector 中,设置 Albedo 为红色,如图 5-8 所示。最终效果如图 5-9 所示。



图 5-8 材质的设置

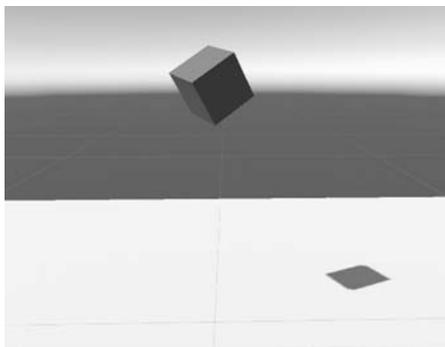


图 5-9 最终效果图

(4) 选择 Cube,在 Inspector 中,单击“Add Component”,添加刚体 Rigidbody 组件,勾选“Use Gravity”属性。设置如图 5-10 所示。

(5) 运行程序,Cube 从高空落下,并落在 Plane 上,如图 5-11 所示。

此时,问题来了,为什么 Cube 会落到 Plane 上,而不是穿透继续下落呢? 主要是因为 Unity 自带的 3D 物体,都添加了碰撞器。

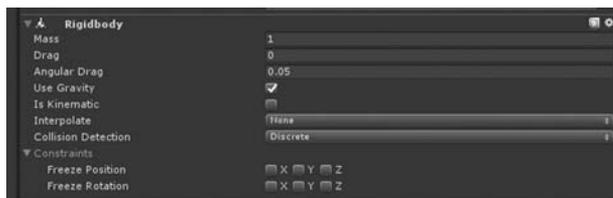


图 5-10 Rigidbody 设置界面

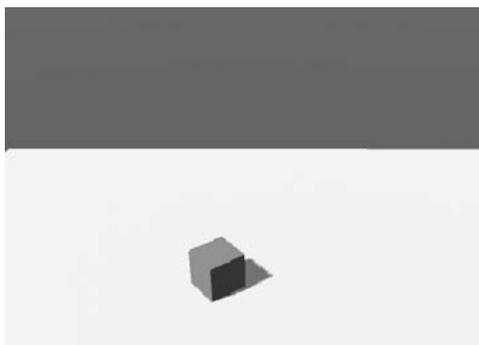


图 5-11 运行后的效果

如果要让两个物体发生碰撞,至少有一个物体要带有刚体组件。即碰撞条件是:碰撞两物体都添加碰撞组件,运动的物体添加刚体组件,如图 5-12 所示。运动物体 A 在场景(静止 B)中运动,如果 A 碰到 B 则发生碰撞,产生碰撞效果。

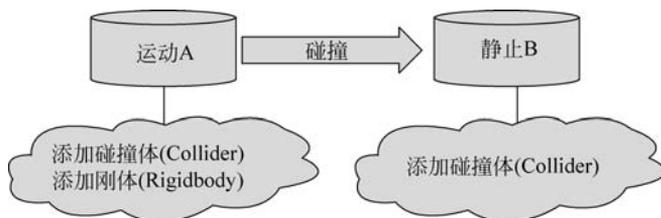


图 5-12 两物体碰撞产生碰撞效果的条件

修改上面的案例,验证碰撞效果。

(1) 在场景中,单击 Hierarchy 面板中的 Create→3D Object→Cube,添加中间层物体 Cube,命名为 Cube2,修改 Rotation(3,0.1,3)。

(2) 在 Project 中,单击 Create→Material,创建材质,命名为 Cube2Material,修改 Inspector 中的 Albedo 为紫色,并将材质球拖入到 Cube2 上。

(3) 选择 Cube2,单击 Inspector 中的 Box Collider 碰撞体下的“Edit Collider”,如图 5-13 所示。编辑 Cube2 的碰撞器,使之一半有碰撞器,一半没有碰撞器,如图 5-14 所示。

(4) 运行程序,最终效果是物体掉落到有碰撞体区域发生了碰撞效果,在无碰撞体区域发生了穿墙而过的效果,如图 5-15 所示。



图 5-13 编辑碰撞体界面



图 5-14 编辑碰撞体后的最终效果

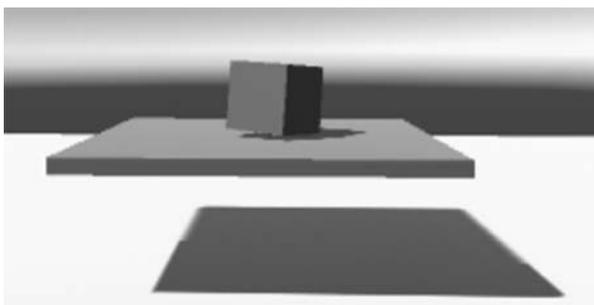


图 5-15 添加碰撞体的运行效果

观察图 5-13 可知,未选中选项“Is Trigger”。“Is Trigger”用来设定碰撞体是否转换成触发器。如果选中,则碰撞体将成为一个触发器,当碰撞体与触发器相碰时,会触发事件,但是不像碰撞那样产生物理效应,碰撞体会直接穿过触发器。

继续修改上一个案例,将物体 Cube2 的 Box Collider 中的 Is Trigger 勾选,如图 5-16 所示。再运行程序,最终效果如图 5-17 所示。



图 5-16 触发器设置

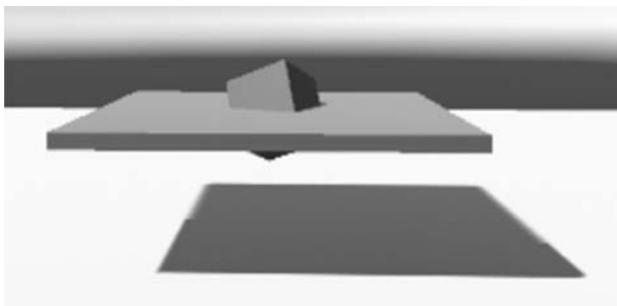


图 5-17 更改为触发器后的运行效果

## 5.4 地形

Unity 提供了一个功能强大、制作灵活的地形系统 Terrain,可以实现快速创建各种地形,如添加草地、山石等材质,添加树木、花草等对象,从而创建出逼真自然的地形环境。

### 5.4.1 导入资源包

制作地形,需要导入资源包。资源包是 Unity 开发的可供用户使用的各种资源,也可以是第三方开发的各种资源(免费或收费),包括 3D 模型、贴图和材质、环境、粒子系统、摄像机、着色器、音频、动作、脚本等。导入资源包方法如下。

方法一: 菜单导入,依次单击菜单栏的 Assets→Import package。

方法二: 在 Project 面板中空白处单击鼠标右键选择 Import package。

方法三: 资源包直接拖入 project 面板中。

### 5.4.2 创建地形

单击菜单【GameObject】→【3D Object】→【Terrain】,在场景中自动添加一个 Terrain 对象,如图 5-18 所示。

该对象包括三个组件: Transform 组件、Terrain 组件和 Terrain Collider 组件。

Terrain 对象不能通过 Transform 组件中的“Scale”属性修改大小,需要通过“Terrain 组件”中“设置选项卡”中的“Terrain Width 和 Terrain Height”属性进行设置。Terrain 组件对地形进行编辑和修改。

### 5.4.3 编辑地形

“Terrain 组件”中的 7 个按钮就是绘制地形工具,如图 5-19 所示。



图 5-18 创建地形



图 5-19 编辑地形按钮

图 5-19 的解释:

-  提升/降低地形高度;
-  绘制目标高度;
-  平滑高度;
-  绘制纹理贴图;
-  绘制树木;
-  绘制花草;
-  地形设置。

案例 3: 室外地形

要点: 绘制室外起伏地形,并添加植物。

(1) 单击 Hierarchy 面板的 Create 菜单,在弹出下拉菜单选择 3D object/Terrain,新建地形,如图 5-20 所示。

(2) 单击 Inspector 中 Terrain 标签的地形设置按钮,如图 5-21 所示。

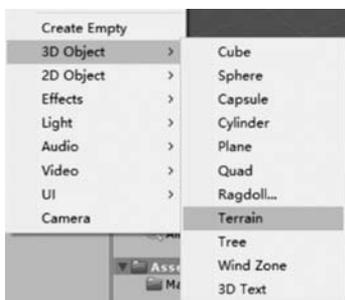


图 5-20 新建地形



图 5-21 设置地形

(3) 单击 Terrain 标签的地形高度设置按钮,如图 5-22 所示,绘制效果如图 5-23 所示。



图 5-22 地形高度按钮

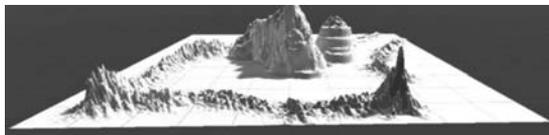


图 5-23 地形效果

(4) 导入 Unity 的标准资源包 standard package,单击 Terrain 标签中的纹理设置按钮,如图 5-24 所示。然后打开对话框,选择贴图,然后在场景中绘制,绘制效果如图 5-25 所示。

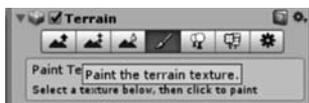


图 5-24 纹理按钮

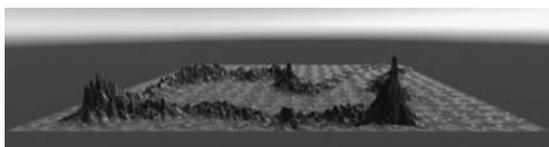


图 5-25 纹理效果

(5) 单击 Terrain 标签中的添加树木按钮,如图 5-26 所示。打开对话框,选择树木贴图,然后在场景中绘制,可以多次选择多次绘制,绘制效果如图 5-27 所示。



图 5-26 添加树木按钮



图 5-27 绘制效果

## 5.5 材质和贴图

### 5.5.1 材质

材质是指定给对象的曲面或面,以在渲染时按某种方式出现的数据信息。主要用于描述对象如何反射和传播光线,为对象表面加入色彩、光泽、纹理和不透明度等,它包含基本材质属性和贴图。

Unity 中材质是一种资源,不是一种可以单独显示的对象,通常赋给场景中的对象,对象表面的色彩、纹理等特性由添加给该对象的材质决定。

#### 1. 创建材质

方法一: 菜单【Assets】→【Create】→【Material】。

方法二: Assets 面板,右键菜单【Create】→【Material】。

#### 2. 为对象指定材质

方法一: 直接将材质拖动到场景的对象上。

方法二: 将材质拖到 Hierarchy 面板的对象名称上。

### 5.5.2 贴图

贴图是指定给材质的图像。可以将贴图指定给构成材质的大多数属性,可以影响对象的颜色、纹理、不透明度以及表面质感等。Unity 中通过 Material 类的“MainTexture”属性来表现对象表面的纹理贴图。

#### 1. 贴图指定给材质

有两种方法可以将一个贴图纹理应用到一个属性。

方法一: 将贴图纹理从资源面板中拖动到方形纹理上面。

方法二: 单击 Select(选择) 按钮,然后从出现的对话框中选择纹理。

#### 2. 贴图类型

导入 Unity 中的图片,默认为 Texture 类型,可以直接指定给材质的某个属性,在 Inspector 面板中可以将其设置为其他类型,如 Normal Map(法线贴图)、Sprite(精灵贴图)、Cursor(鼠标贴图)等。

Shader 着色器: 专门用来渲染 3D 图形的技术,可以使纹理以某种方式展现。实际就是一段嵌入到渲染管线中的程序,可以控制 GPU 运算图像效果的算法。

Texture 纹理: 附加到物体表面的贴图。

Albedo: 反照率参数,控制表面的基色,一般我们都是给 Albedo 参数分配纹理贴图。

Metallic: 金属参数,决定了表面的“金属化”。当金属化参数调整到更大时,材质更金属化,它将更多地反映环境。

Smoothness: 平滑度参数,平滑度越低则漫反射越多,而调高平滑度则镜面反射变多。

Normal Map: 法线贴图,是一种凹凸贴图。它们是一种特殊的纹理,在不增加模型面数的情况下,允许将表面细节加到能捕获光(接收光照)的模型中,看起来就像由实际的模型面

来表示一样。

Height Map: 高度贴图,与法线贴图类似,这种技术更复杂,性能也更高。Height map 通常与 normal map 一起使用,通常它们用于给纹理贴图负责渲染突起的表面提供额外的定义。

#### 案例 4: 对象旋转

要点: 建立立方体,添加材质,按下 R 键,立方体绕 Y 轴旋转。

制作步骤如下。

(1) 单击 Hierarchy 面板的 Create 菜单,在弹出下拉菜单选择 3D Object/Cube, 建立立方体,如图 5-28 所示。

(2) 将 Project 面板的 Assets 中的材质图拖到立方体上,如图 5-29 所示。效果如图 5-30 所示。

(3) 在 Project 面板的 Assets 中,右击新建脚本,如图 5-31 所示。

(4) 添加旋转语句,如图 5-32 所示。

(5) 运行程序,按下 R 键,立方体沿 Y 轴旋转。

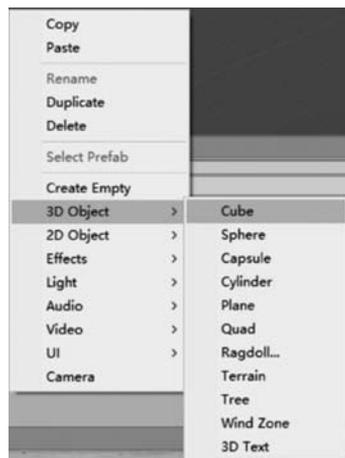


图 5-28 建立立方体

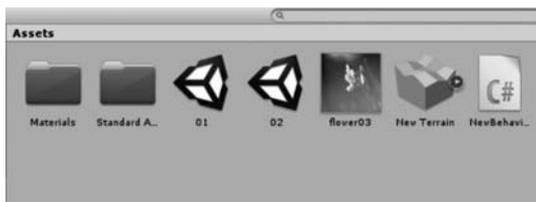


图 5-29 添加 Assets 的图



图 5-30 添加材质效果

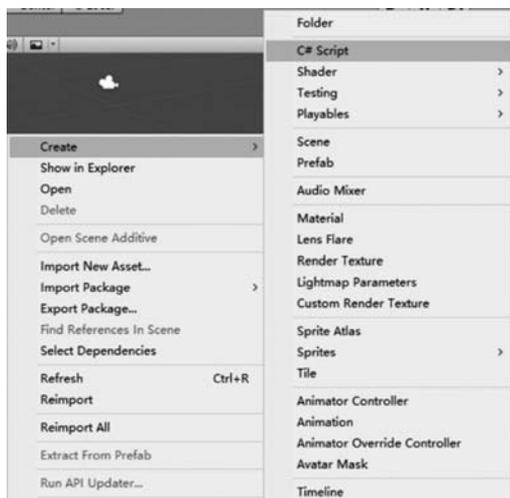


图 5-31 新建脚本

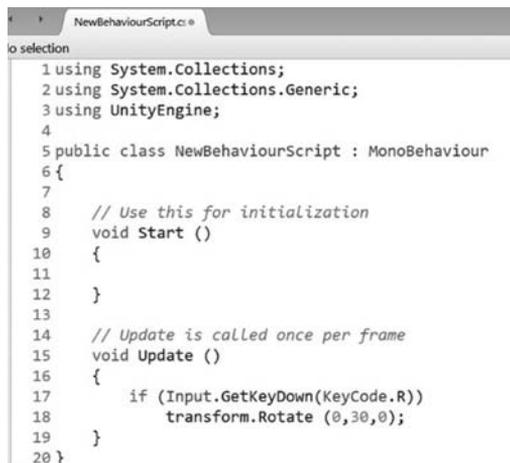


图 5-32 添加语句

## 5.6 光照系统

光照是模拟真实灯光的对象,如建筑内部各种灯具、舞台和电影工作时使用的灯光设备以及太阳光本身。灯光是一种特殊对象,它不被渲染显示,但可以影响周围物体表面的光泽、色彩和亮度,通常与材质、环境共同作用,增强了场景的清晰度、真实感、层次性。不同种类的灯光对象有不同的投射方法,模拟真实世界中不同种类的光源。

当新建一个场景时,场景中默认创建一个方向光——Directional light。

### 5.6.1 光照类型

现实世界中光源的类型包括直接光、间接光、环境光、反射光等。在 Unity 中提供了 4 种直接光,分别是平行光、点光源、聚光灯和区域光。

#### 1. 光照类型

**平行光:** 由光源发射出的相互平行的光。使用平行光,可以把整个场景都照亮,可以认为平行光是整个场景的主光源,一般用于模拟太阳光或月光等户外光线,如图 5-33 所示。

**点光源:** 点光源的光线由光源中心向周围  $360^\circ$  发射,照射区域范围为一个球体。通常用来模拟灯泡等光源,如图 5-34 所示。

**聚光灯:** 聚光灯的光线投射区范围是一个圆锥体,向一个方向发射。聚光灯可以用来模拟舞台聚光灯或手电筒等光源的灯光,如图 5-35 所示。

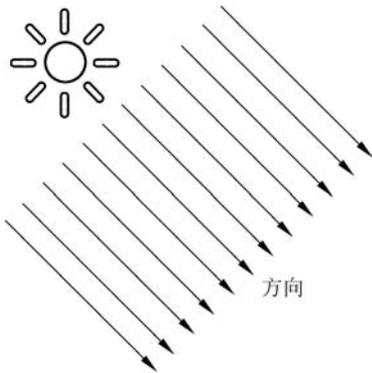


图 5-33 平行光效果图

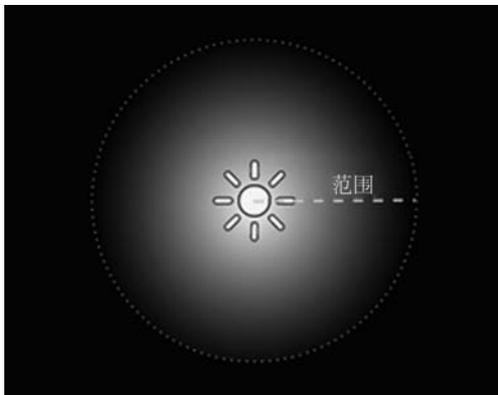


图 5-34 点光源效果图

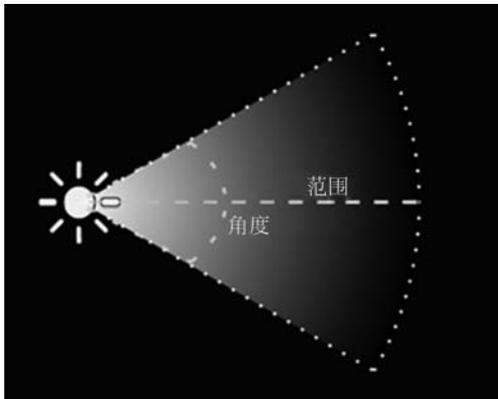


图 5-35 聚光灯效果图

**区域光:** 区域光由一个面向一个方向发射光线,只照射区域内物体,并且只在烘焙时有效,如图 5-36 所示。

#### 2. 灯光属性

灯光常用属性有 Type(灯光类型)、Range(灯光照射范围)、Color(灯光颜色)、Intensity

(灯光亮度)等,如图 5-37 所示。

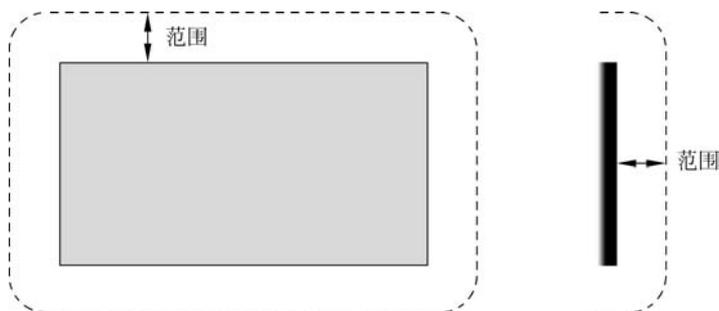


图 5-36 区域光效果图

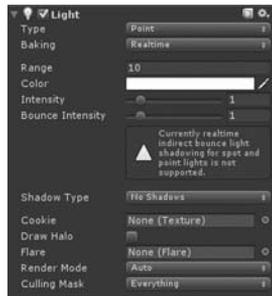


图 5-37 灯光属性

## 5.6.2 实时光照

所谓实时光照是指实时更新光线信息,在运行状态时任意修改光源所有的变化可以立即更新。

操作步骤:

- (1) 选择光源→【Mode】→【Realtime】。
- (2) 在工具栏找到【Window】→【Light】→【Settings】。
- (3) 在灯光设置中将环境光模式调成【Realtime】→勾选自动更新【Auto Generate】。

## 5.6.3 灯光烘焙

灯光烘焙就是使用烘焙技术将光线效果、阴影信息等预渲染成贴图信息作用在物体上,烘焙灯光只对静态物体有效。

操作步骤:

- (1) 将物体设置成静态物体,点选物体→勾选静态【Static】。
- (2) 将灯光设置成烘焙模式,点选光源→【Mode】→【Baked】。
- (3) 在光照设置中将环境光模式改成【Baked】→取消勾选自动更新【Auto Generate】→单击【Generate Lighting】开始烘焙。

## 5.7 动画

### 5.7.1 动画剪辑

导入到 Unity 中的 3D 动画称为动画剪辑(Animation Clip),动画剪辑包含一段相对完整的动画,一个角色可以带多个动画剪辑。当把带有动画的 3D 模型导入到 Unity 中时,会自动创建动画剪辑。

Animation Clip 动画剪辑,用于存储角色或者简单动画的动画数据,它是动作的简单“单元”,例如“走路”“跑步”或者“跳跃”等,对动画动作的修改和编辑通过 Animation 视图完成。

通过 Animation 视图也可以创建新的动画剪辑文件,扩展名为 .anim。动画剪辑数据和模型对象是分离的,同一个动画剪辑可以应用不同的模型对象。

## 5.7.2 动画状态机

### 1. Animator 组件

要实现角色对象的动画控制,需要为角色对象添加 Animator 组件,并且需要将创建好的动画控制器赋给 Animator 组件的“Animator Controller”属性,如图 5-38 所示。

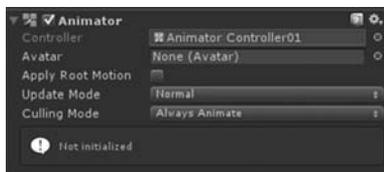


图 5-38 Animator Controller

### 2. 动画控制器和 Animator 视图

动画控制器的创建方法:在 Assets 面板中单击右键,选择【Create】→【Animator Controller】。

动画控制器在 Animator 视图进行编辑,如图 5-39 所示。

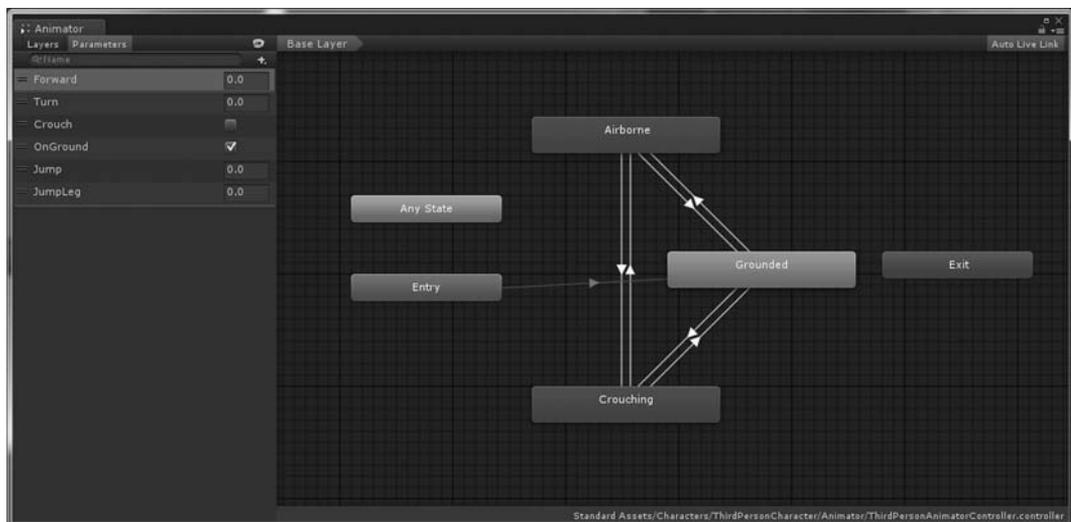


图 5-39 Animator

通过 Animator 视图打开动画控制器,可以看到一个空的动画控制器,包含一个动画入口 Entry、一个动画出口 Exit 和一个任意动画状态 Any State。

可以往动画控制器中添加动画剪辑,动画剪辑添加到 Animator 视图中,就称为动画状态,一个动画剪辑就是一个动画状态,初始动画状态显示为橙色。

### 3. 动画状态过渡

一个角色可以有多种动画状态(动作),当满足一定条件时,可以从一种动画状态过渡到另一种动画状态。

创建动画过渡的方法,在动画状态 A 上单击右键,选择【Make Transition】,如图 5-40 所示,然后拖动鼠标到另一个动画状态 B 上,就创建了从动画状态 A 到动画状态 B 的动画过渡,显示为方向箭头,如图 5-39 所示。



图 5-40 Make Transition

## 5.8 音频系统

### 5.8.1 音频概述

音频是虚拟现实和游戏设计开发流程中不可缺少的一环,通常在创作的最后阶段添加。音频可以起到烘托环境气氛、突出故事情节、辨别对象位置等作用。

导入到 Unity 中的音频文件称为音频剪辑(Audio Clip)。音频资源有压缩和不压缩两种方式,不进行压缩的音频将采用音频源文件,而采用压缩的音频文件会先对音频进行压缩,此操作会减小音频文件的容量,但是在播放时需要额外的 CPU 资源进行解码,所以需要制作快速反应的音效时,最好使用不压缩的方式,而背景音乐可以使用压缩的音频文件。任何格式的音频文件被导入 Unity 后,在内部自动转化成.ogg 格式。

### 5.8.2 音频组件

音频剪辑需要配合两个组件来实现音频的监听和播放。

#### 1. 音频监听组件

音频监听组件(Audio Listener)是用于接收声音的组件,配合音频源为虚拟现实和游戏创建听觉体验。该组件的功能类似于麦克风,当音频监听组件挂载到游戏对象上,任何音频源,只要足够接近音频监听组件挂载的游戏对象,都会被获取并输出到计算机等设备的扬声器中输出播放。如果音频源是 3D 音效,监听器将模拟在 3D 世界声音的位置、速度和方向。

音频监听组件默认添加在主摄像机上。该组件没有任何属性,只是标注了该游戏对象具有接收音频的作用,同时用于定位当前的接收位置。

添加方法:【Component】→【Audio】→【Audio Listener】。

#### 2. 音频源组件

音频源组件(Audio Source)用于播放音频剪辑文件,通常挂载在游戏对象上。该组件负责控制音频的播放,通过组件的属性设置音频剪辑的添加和播放方式,如图 5-41 所示。如果音频文件是 3D 音效,音频源也是一个定位工具,可以根据音频监听对象的位置控制音频的衰减。

添加方法:【Component】→【Audio】→【Audio Source】。

AudioClip: 音频片段,将需要播放的音频文件放入其中,支持.aif,.wav,.mp3,.ogg 格式。

Play On Awake: 在唤醒时开始播放,勾选后,在游戏运行以后,就会开始播放。

Loop: 循环,勾选后,声音进入“单曲循环”状态。

Mute: 静音,勾选后,静音,但音频仍处于播放状态。

Volume: 音量,0: 无声音; 1: 音量最大。

Spatial Blend: 空间混合,设置声音是 2D 声音,还是 3D 声音。2D 声音没有空间的变化,3D 声音有空间的变化,离音源越近听得越明显。

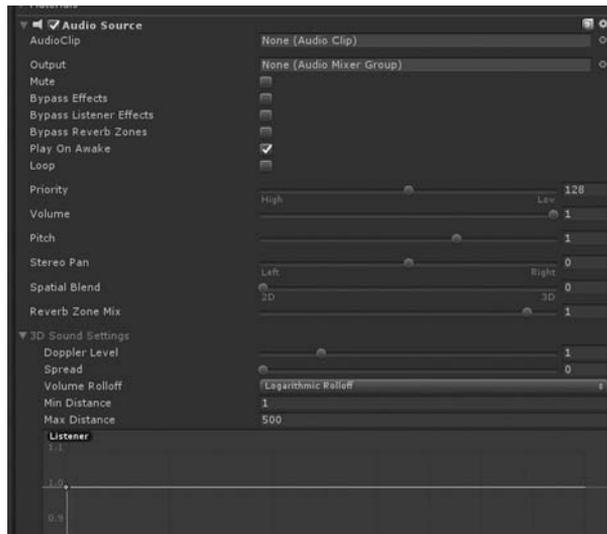


图 5-41 Audio Source