第 3 章 我的第一个 Unity APP（GameObject）

为了熟悉 Unity，本章使用一步一步的方法让你了解一个简单的 Unity 游戏场景是怎么设计出来的。本章要完成的游戏对象效果如图 3-0 所示。

图 3-0　本章的目标完成图

3.1　游戏主菜单——创建项目

现在一步一步地介绍如何用 Unity 创建一个游戏主菜单。在本节中将会学会以下技巧：

（1）3D 对象的排列。

（2）纹理图片的添加。

（3）纹理属性的设置。

（4）纹理尺寸的设置。

STEP1：打开 Unity 工具

先把 Unity 打开，不管用 Windows 还是 Mac 过程都一样。若为 Windows，则选择 Start|All Programs|Unity|Unity，如图 3-1 所示。

如果使用苹果计算机 Mac，则把 finder 文件目录打开，运行 Applications|Unity|Unity，如图 3-2 所示。

图 3-1　打开 Unity 工具

图 3-2　在 Mac 中打开 Unity 工具

STEP2：创建一个新的项目

进入 Unity 5 之后，单击 New Project 按钮，如图 3-3 所示。

图 3-3　单击 New Project 按钮

或者是在 Unity 5 菜单中，选择 File | New Project 命令，如图 3-4 所示，用来创建一个新的作品。

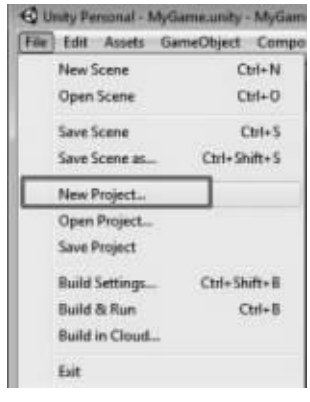

图 3-4　选择 File|New Project 命令

STEP3：设置项目

在图 3-5 所示的对话框中设置：

(1) 项目名称，在此笔者是用 MyGame。

(2) 选择要存储路径。

(3) 这个项目是 2D 还是 3D 游戏，在此设置为 3D 游戏。

(4) 单击 Create Project 按钮用来创建项目。

完成后，就会进入 Unity 的工作窗口。

STEP4：在 Project 中创建文件夹

如图 3-6 所示，加上一个文件夹，用来准备等一下游戏主画面的图片，可依照以下步骤设置：

(1) 在 Project|Assets 的空白处右击。

(2) 在弹出的快捷菜单中选择 Create|Folder 命令，就可以创建一个文件夹。

(3) 将文件夹命名为 image。

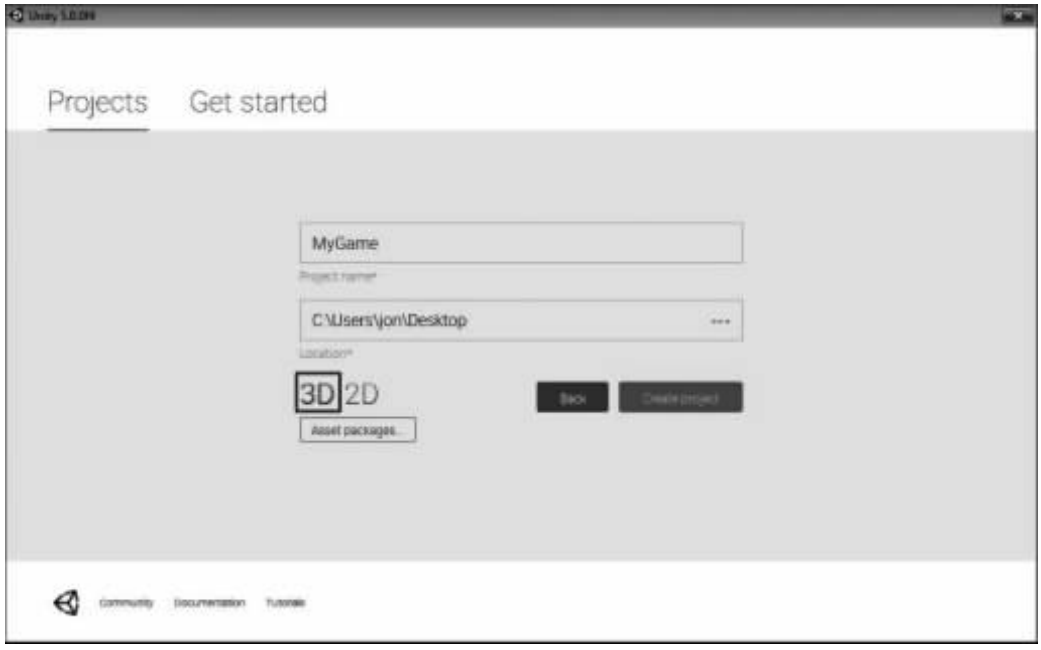

图 3-5　设置项目

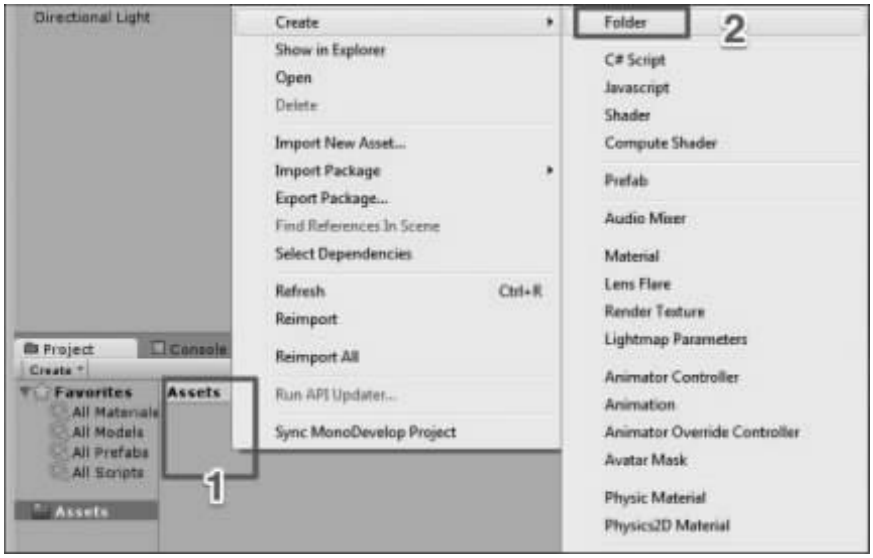

图 3-6　创建文件夹 image

3.2　游戏主菜单——显示后台图片

STEP5：添加图片到项目中

打开文件目录，把事先准备好的图片全部选中。如果没有设计的游戏主菜单的图片，可以使用光盘 SampleCode\Ch3\pic 里面的图片。

使用鼠标拖拉的方法，把图片拖到刚刚创建的文件夹内，如图 3-7 所示。

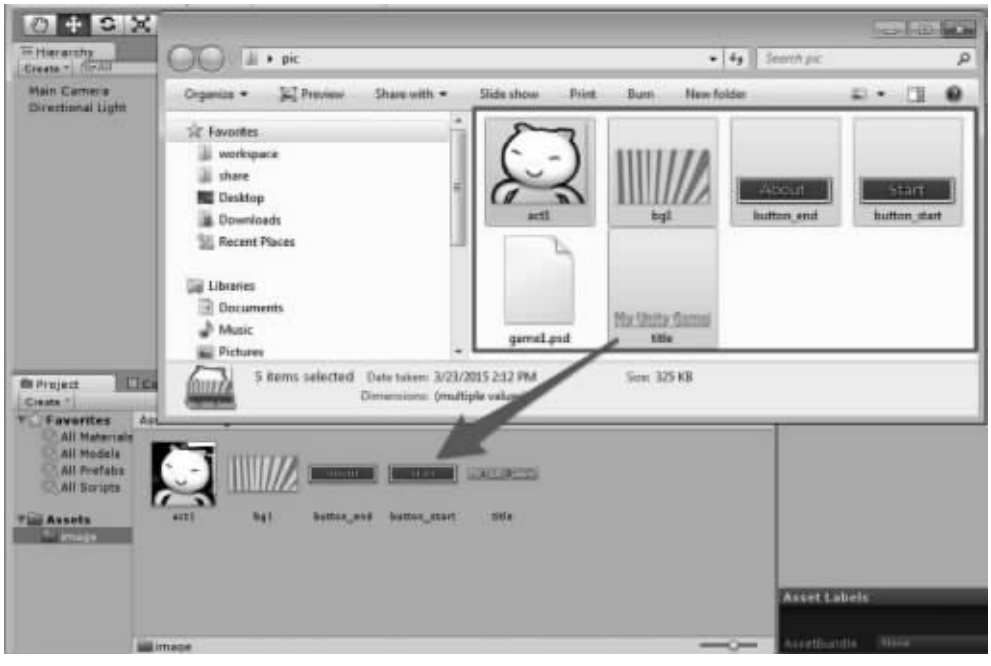

图 3-7　添加图片到项目中

Unity 5.1 版当前图片的纹理支持 PNG、JPG/JPEG、GIF、BMP、PSD。

STEP6：加上平面到游戏中

在菜单中选择 GameObject|3D Object|Plane 命令加上一个平面，如图 3-8 所示，用来摆放游戏主菜单的后台图片。

STEP7：设置平面

如图 3-9 所示，便会出现一个平面在 Scene(场景)窗口中，在 Hierarchy(层次)窗口中选择这个平面。

然后，在 Inspector 窗口中把名称和数值调整一下：

(1) Position X,Y,Z：是调整该对象的 X,Y,Z 的位置。

(2) Rotation X,Y,Z：是调整该对象的 X,Y,Z 的旋转角度。

(3) Scale X,Y,Z：是调整该对象的 X,Y,Z 的大小。

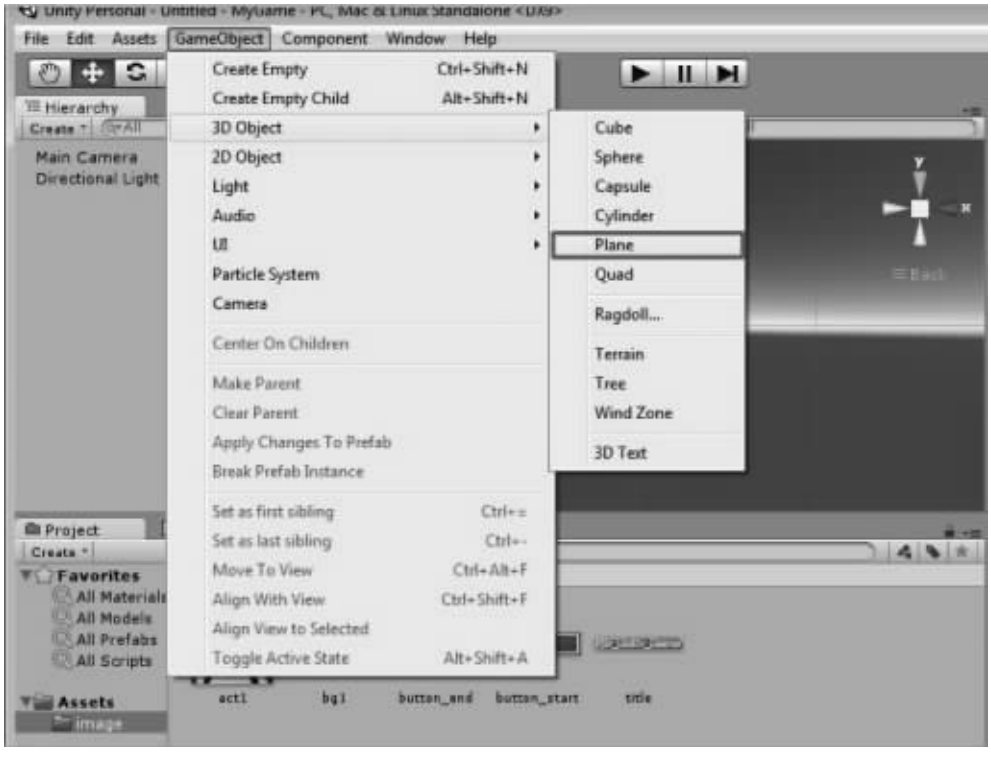

图 3-8　选择 GameObject|3D Object|Plane 命令加上一个平面

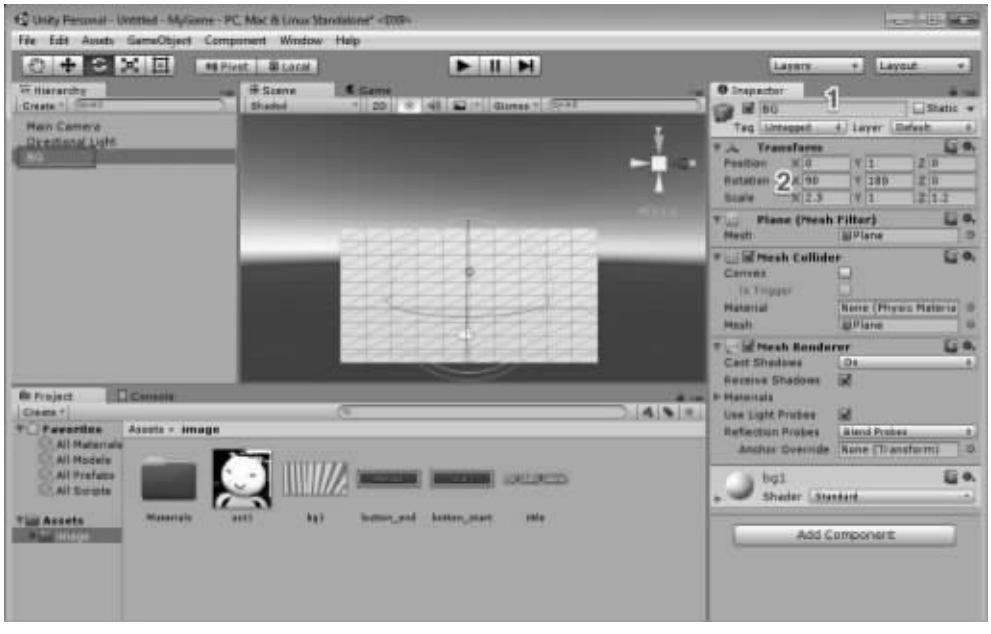

图 3-9　平面在 Scene(场景)窗口中

可把名称和数值调整成图 3-10 所示。

此时就会发现后台图片摆放的位置、尺寸有所改变。

STEP8：添加纹理图片

如图 3-11 所示，添加纹理图片到平面上，可通过以下步骤完成：

（1）拖拉图片 bg1 到 BG 平面的 Inspector 窗口下方。

（2）操作完成后，原本白色的平面就会把图片当成纹理显示。

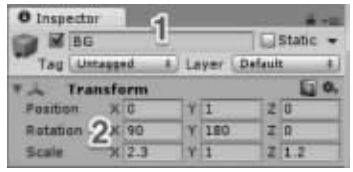

图 3-10　调整后的平面的数值和名称

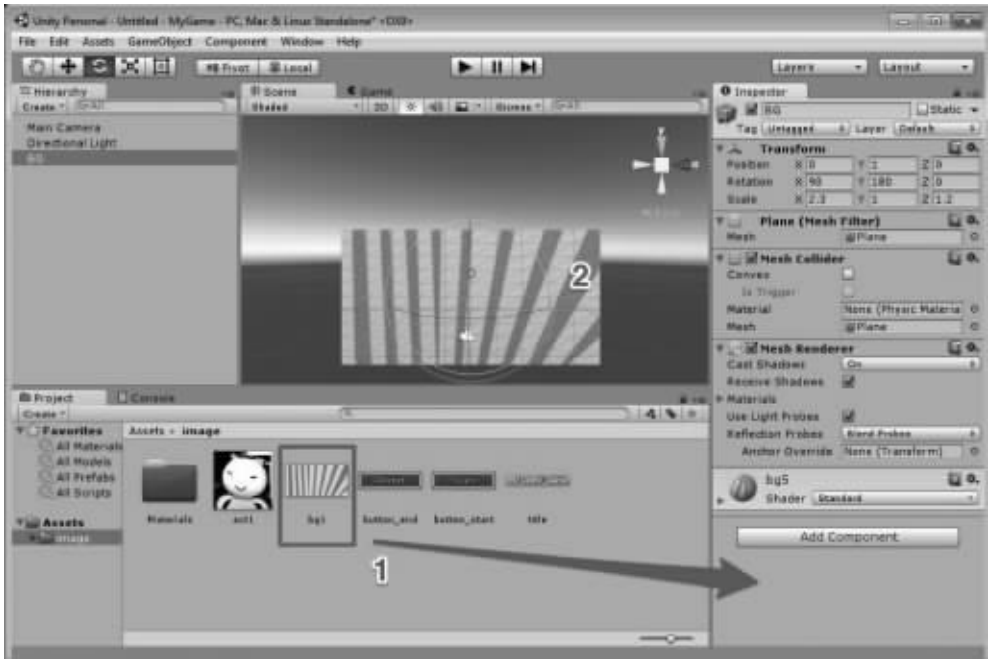

图 3-11　添加纹理图片

3.3　游戏主菜单——显示游戏名称

STEP9：加上一个平面到游戏中并对其进行设置

再添加一个平面在游戏场景中，可选择 GameObject|3D Object|Plane 命令，如图 3-12 所示。

把平面的位置调整到合适的位置，用它来摆放游戏名称。比较特别的是 Z 的部分调整为－0.01，以避免和 BG 后台放在同一个 0 的位置，所以与摄影机靠得近一些，如图 3-13 所示。

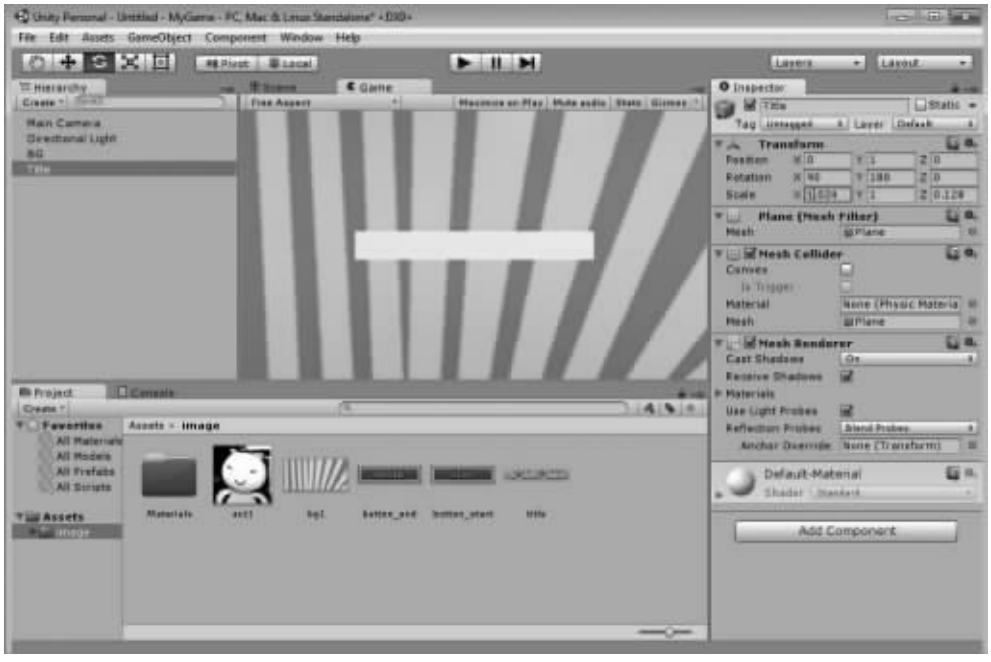

图 3-12　选择 GameObject|3D Object|Plane 命令

图 3-13　调整平面后的数值

STEP10：添加纹理图片

如图 3-14 所示，添加纹理图片到平面上，可通过以下步骤完成：

（1）拖拉图片 title 到 Title 平面的 Inspector 窗口下方。

（2）操作完成后，原本白色的平面就会把图片当成纹理显示。

这时游戏窗口的画面如图 3-15 所示，用户可以通过选择 Game 或“运行游戏”来看一下现在游戏的效果。

STEP11：纹理去背

这个纹理看起来有些怪怪的，原因是使用了 PNG 的图片，游戏名称后面部分的图片是镂空透明的，可依照以下步骤调整：

（1）选中 Title 平面的 Inspector 窗口中的图片旁边的三角形，做展开的动作。

（2）修改 Rendering Mode（显示模式）为 Cutout（修剪）。

这样就可以达到纹理去背的效果，如图 3-16 所示。

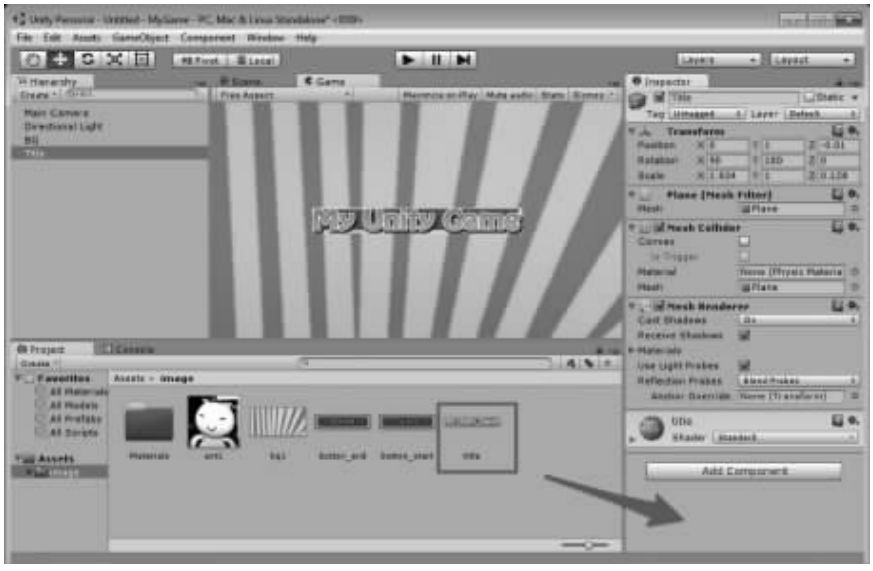

图 3-14　添加纹理图片

图 3-15　游戏现在的情况

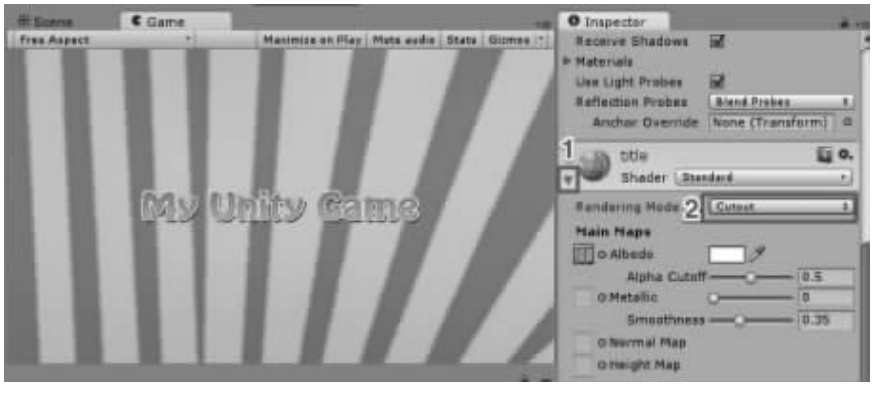

图 3-16　纹理去背

3.4 游戏主菜单——添加按钮

接下来添加两个按钮，以便将来处理按钮的选择。请延续上一节继续以下的步骤来设计游戏的主菜单。

STEP12：添加 Start 按钮

添加第一个按钮——Start 按钮。同样地，如 STEP9 添加一个平面对象，并调整为图 3-17 所示。

（1）调整名称为 Button_Start 平面，并设置 Position（位置）为 0，0.46，－0.01；Rotation（旋转）为 90，－180，0；Scale（尺寸）为 0.512，1，0.126。

（2）添加纹理图片到该平面上，并拖拉图片 button_start 到 Button_Start 平面的 Inspector 窗口下方，此时会显示纹理图。

（3）选中 Button_Start 平面的 Inspector 窗口中的图片旁边的三角形，做展开的动作，修改 Rendering Mode（显示模式）为 Cutout（修剪）。

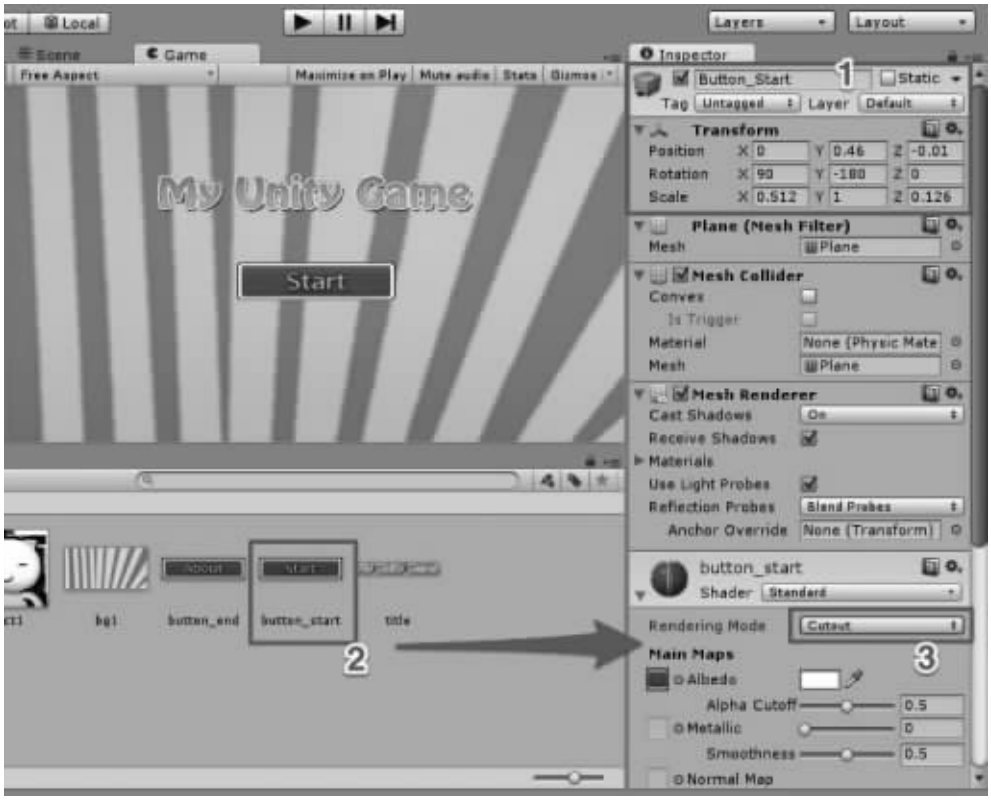

图 3-17　添加 Start 按钮

STEP13：添加 About 按钮

同上一步骤，再添加另外一个按钮——About 按钮。同样地，如 STEP9 添加一个平面对象，并调整为图 3-18 所示。

(1) 调整名称为 Button_End 平面，并设置 Position(位置)为 0，－1.14，－0.01；Rotation(旋转)为 90，－180，0；Scale(尺寸)为 0.512，1，0.126。

(2) 添加纹理图片到该平面上，并拖拉图片 button_end 到 Button_End 平面的 Inspector 窗口下方，此时会显示纹理图。

(3) 选中 Button_End 平面的 Inspector 窗口中的图片旁边的三角形，做展开的动作，修改 Rendering Mode(显示模式)为 Cutout(修剪)。

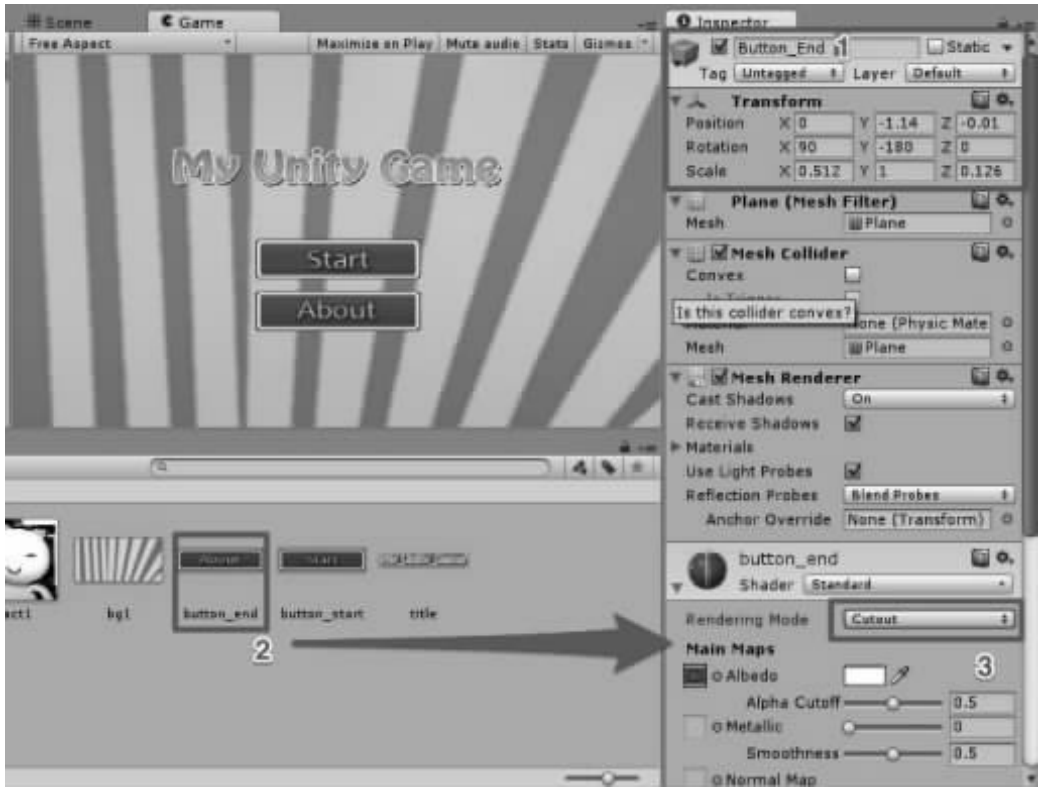

图 3-18　添加 About 按钮

3.5　游戏主菜单——添加主角

接下来添加一个主角的图片，用来介绍如何显示主角。请延续上一节继续以下的步骤来设计游戏的主菜单。

STEP14：添加主角

同步骤 9，再添加一个平面用以摆放主角，并调整为图 3-19 所示。

（1）调整名称为 act1 平面，并设置 Position（位置）为 4.72，－2.26，－0.01；Rotation（旋转）为 90，－180，0；Scale（尺寸）为 0.512，1，0.512。

（2）添加纹理图片到该平面上，并拖拉图片 act1 到 act1 平面的 Inspector 窗口下方，此时会显示纹理图。

（3）选中 act1 平面的 Inspector 窗口中的图片旁边的三角形，做展开的动作，修改 Rendering Mode（显示模式）为 Cutout（修剪）。

图 3-19　添加主角

STEP15：设置纹理压缩和尺寸

在这里，对画面要求较高的读者会发现一个问题，就是游戏中的画面的画质，好像图片都是模糊的，并不像当初在绘图软件上看得那样清晰。下面依照图 3-20 所示步骤调整一下每一张图片的压缩比例和纹理图片的尺寸：

（1）选中图片。

（2）在 Inspector 窗口中就会出现图片的属性，调整 Max Size（图片最大的比例）为清一色突变的，实际尺寸稍微设置一个比较大的。若是为小朋友设计的，则不选择 256。对于 Format，系统默认值 Compressed 是有压缩的，将其调整为合适的影像格式。如果可能，可选择 Truecolor 32 bits 的全彩图片。

（3）单击 Apply 按钮。

图 3-20　设置纹理压缩和尺寸

3.6　游戏主菜单——存储游戏场景

STEP16：存储游戏场景

至此这个游戏主菜单的画面处理已基本完成，这里介绍如何修改和存储游戏场景。选择 File|Save Scene 命令，如图 3-21 所示，并且在弹出的对话框中输入文件名称，这里使用 MyGame 为文件名，完成之后单击 Save 按钮，就可以完成存储的动作，如图 3-22 所示。

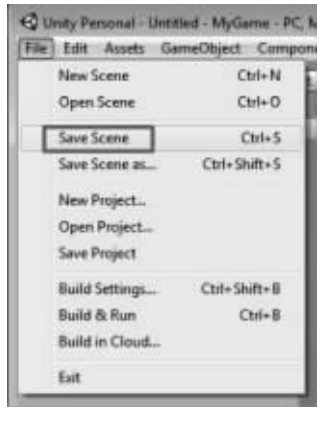

图 3-21　存储游戏场景

STEP17：运行游戏

若想运行游戏，单击画面中的 Play 按钮，就会开始新游戏，如图 3-23 所示。

如果要让游戏结束，再单击一次 Play 按钮即可。本样例可以通过在光盘中选择 Sample\ch03\MyGame\Assets\MyGame.unity 运行程序。运行结果如图 3-24 所示。

教学视频：

为了避免大家在刚上手的时候碰到问题，在这里特意把

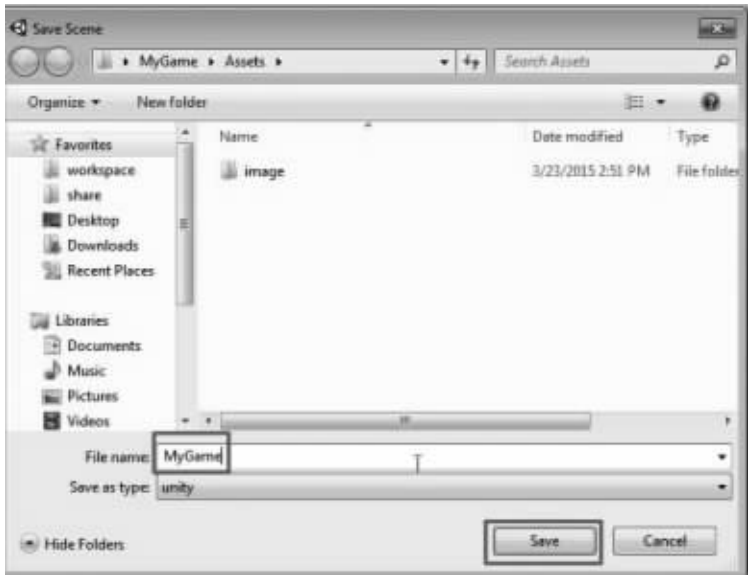

图 3-22　设置存储游戏场景的文件名称

图 3-23　单击 Play 按钮开始新游戏

图 3-24　运行结果

本节中的内容用影片教学,通过录制的整个过程,方便了解本节的内容。读者可以观看光盘中的教学视频 unity_3-1 。

3.7　GameObject(游戏对象)

在我的第一个 Unity APP 中,通过 GameObject 对象来组成游戏的主菜单。在当时的步骤中利用 GameObject 菜单(如图 3-25 所示)选取一个平面用来增添菜单项。接下来介绍

几个常用的游戏对象，这是 Unity 的基本概念。所有游戏中的对象都是通过 GameObject 菜单来完成的，所以它非常重要，而每一个游戏对象都可以在下拉菜单 GameObject 中看到。本节将介绍几个常使用的对象。

图 3-25　GameObject 菜单

3.7.1　Camera(摄影机)

玩家看到的画面实际上是通过 Main Camera 所看到的画面来呈现的，只要将摄影机调整到合适位置，就会让游戏呈现不同的表现效果，如图 3-26 所示。

图 3-26　设置 Camera(摄影机)的 Inspector 属性

3.7.2 Directional Light(太阳光)

在这个游戏中,通过 Hierarchy(层次)窗口可以看到 Directional Light(太阳光)游戏对象,这是设置游戏中使用有方向性的阳光,让对象显示出来。该对象通常都是用以设置环境中的太阳光,如图 3-27 所示。

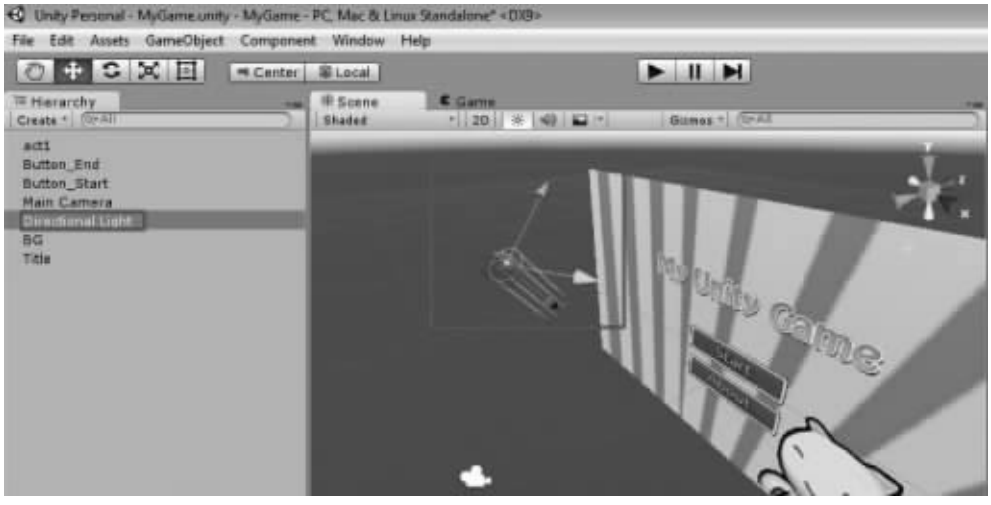

图 3-27 Directional Light(太阳光)对象所呈现画面

常用属性如图 3-28 所示。

(1) Type:灯光的种类。

(2) Color:灯光的颜色。

(3) Intensity:灯光的强弱调整。

(4) Cookie:灯光的遮照纹理。如果想达到太阳光从窗户外面透射过来的效果,可在这里放上窗户造型的纹理,就会产生出家的效果。

(5) Cookie Size:灯光的遮照纹理的尺寸。

(6) Shadow Type:影子的效果。

(7) Draw Halo:晕影的效果。

(8) Flare:亮光。简单地说就是产生类似将摄影机对着太阳的时呈现的晕影效果。

(9) Render Mode:可以选择是否主动采光。

(10) Culling Mask:可以选择产生效果的游戏对象。

图 3-28 设置 Directional Light(太阳光)对象的 Inspector 属性

3.7.3　Point Light(光点)

与 Directional Light(太阳光)游戏对象非常类似的是 Point Light(光点),大多都是用在火把、火炬、爆炸的灯光效果。Point Light(光点)的效果就是一个小范围的灯光,很适合制作局部的小灯光、火把等效果,如图 3-29 所示。

图 3-29　Point Light(光点)对象所呈现画面

常用属性如图 3-30 所示。

(1) Type:灯光的种类。

(2) Range:灯光的范围。

(3) Color:灯光的颜色。

(4) Intensity:灯光的强弱调整。

(5) Cookie:灯光的遮罩材质。如果想达到太阳光从窗户里面照射下来的效果,可在这里放上窗户造型的材质,就会产生出家的效果。

(6) Cookie Size:灯光的遮罩材质的大小。

(7) Shadow Type:影子的效果。

(8) Draw Halo:光晕的效果。

(9) Flare:亮光。简单地说就是会产生类似将摄影机对着太阳时呈现的光晕效果。

(10) Render Mode:可以选择是否会主动采光。

(11) Culling Mask:可以选取产生效果的游戏对象。

(12) Lightmapping:光照贴图。

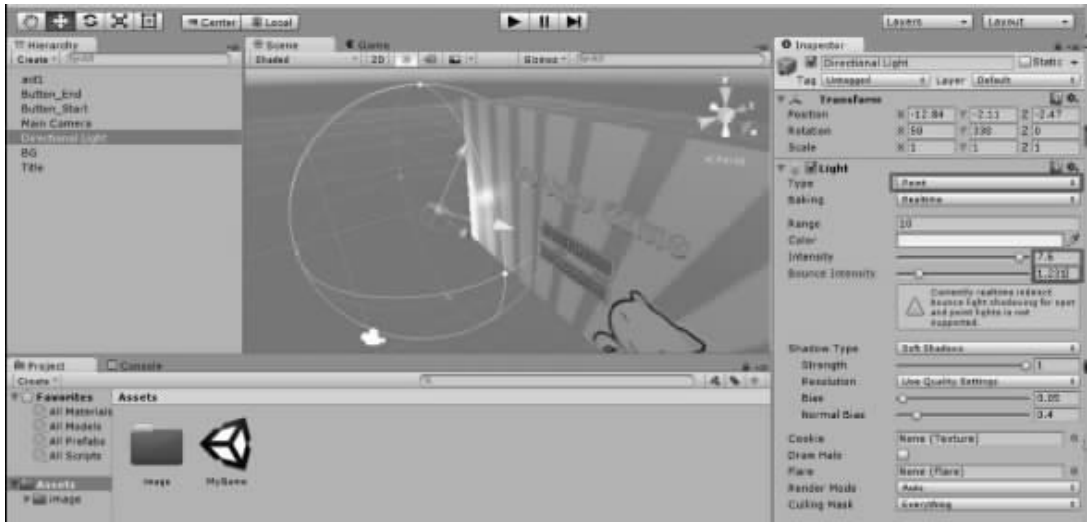

图 3-30　设置 Point Light(光点)对象的 Inspector 属性

3.7.4 Spot Light(聚光灯)

Spot Light(聚光灯)就是产生一个聚光灯的效果,如图 3-31 所示,通常都是用于书桌上的灯光或舞台上的投射灯光。

Spot Light(聚光灯)就是产生一个小范围的灯光,很适合制作局部的聚光灯或者角色中的影子效果。

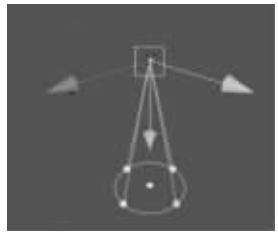

图 3-31 Spot Light(聚光灯)对象所呈现画面

常用属性如图 3-32 所示。

(1) Type:灯光的种类。

(2) Range:灯光的范围。

(3) Spot Angle:聚光灯的角度。

(4) Color:灯光的颜色。

(5) Intensity:灯光的强弱调整。

(6) Cookie:灯光的遮罩材质。如果想达到太阳光从窗户里面照射下来的效果,可在这里放上窗户造型的材质,就会产生出家的效果。

(7) Cookie Size:灯光的遮罩材质的大小。

(8) Shadow Type:影子的效果。

(9) Draw Halo:光晕的效果。

(10) Flare:亮光。简单地说就是会产生类似将摄影机对着太阳时呈现的光晕效果。

(11) Render Mode:可以选择是否会主动采光。

(12) Culling Mask:可以选取产生效果的游戏对象。

(13) Lightmapping:光照贴图。

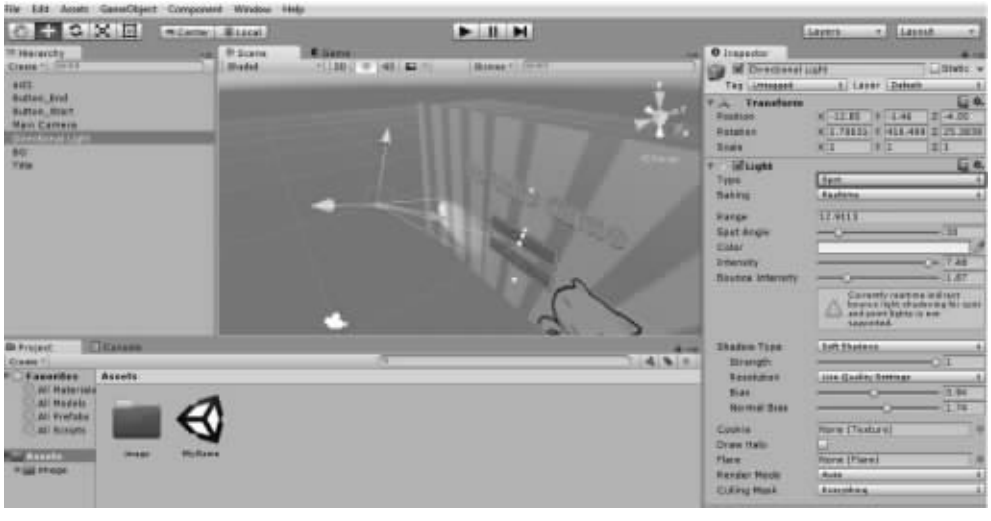

图 3-32 设置 Spot Light(聚光灯)对象的 Inspector 属性

3.7.5　Area Light(区域型灯光)

Area Light(区域型灯光)是一个本地的放射性的灯光,大多是用在房间内的日光灯或是房间里的主要灯光,如图 3-33 所示。

常用属性如图 3-34 所示。

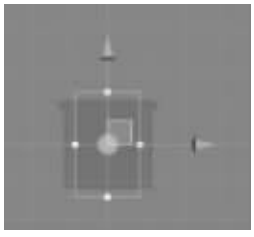

图 3-33　Area Light(区域型灯光)对象所呈现画面

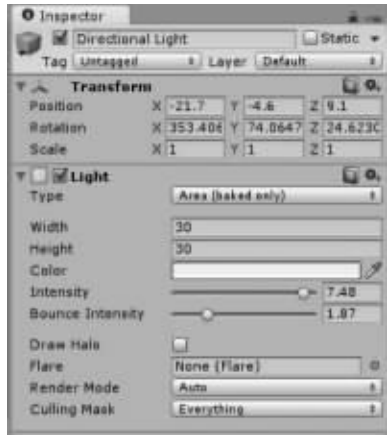

图 3-34　设置 Area Light(区域型灯光)对象的 Inspector 属性

(1) Type:灯光的种类。

(2) Intensity:灯光的强弱调整。

(3) Width:灯光的宽。

(4) Height:灯光的高。

(5) Color:灯光的颜色。

3.7.6　Cube(方块)

在我的第一个 Unity APP 中,通过 Plane(平面)对象来组成游戏的主菜单,与此类似的还有 Cube(方块)对象,如图 3-35 所示,它是一个正方形的对象。

常用属性如图 3-36 所示。

1. Transform(转换)

(1) Position X,Y,Z:对象位置。

(2) Rotation X,Y,Z:对象旋转角度。

(3) Scale X,Y,Z:对象缩放。

2. Mesh Filter(网状模型)

选择外观的样子。

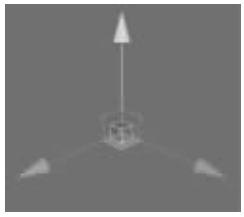

图 3-35　Cube(方块)对象所呈现画面

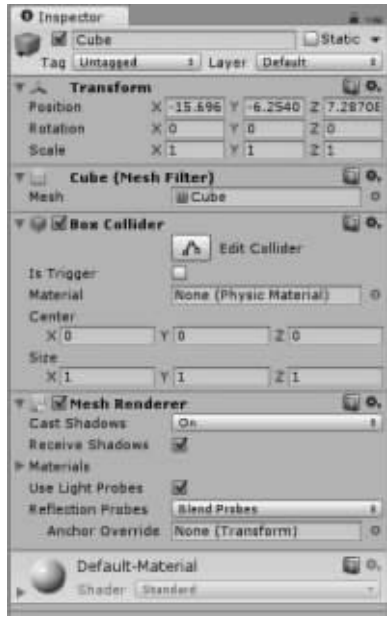

图 3-36　设置 Cube(方块)对象的 Inspector 属性

3. Box Collider(正方形的碰撞)

(1) Is Trigger：碰撞到此对象是否会产生 Trigger(触发)事件。有 OnTriggerEnter、OnTriggerStay、OnTriggerExit 三个选项。

(2) Material：材质。

(3) Center X,Y,Z：碰撞的外框的中心点的位置。

(4) Size X,Y,Z：碰撞的外框的大小。调整为 1.5 后的样子如图 3-37 所示。

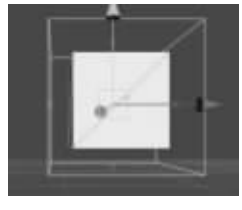

图 3-37　Cube(方块)对象的碰撞外框大小调整为 1.5 后的样子

4. Mesh Renderer(网状渲染)

(1) Cast Shadows：投射阴影。

(2) Receive Shadows：接收可以显示其他对象的阴影。

(3) Materials：材质。

(4) Size：网状渲染材质的数量。

(5) Element：网状渲染材质图。

(6) Use Light Probes：使用光探头。

5. Default-Diffuse(材质)

(1) Shader：材质图的贴图运算方法。

(2) Main Color：材质主要的颜色。

(3) Tiling：材质显示的重复数量。

(4) Offset：材质图位移。

3.7.7　Sphere(球体)

Sphere 是一个球体对象，如图 3-38 所示。

常用属性如图 3-39 所示。

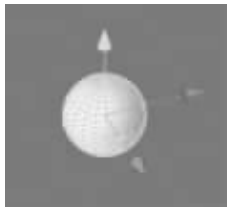

图 3-38　Sphere(球体)对象所呈现画面

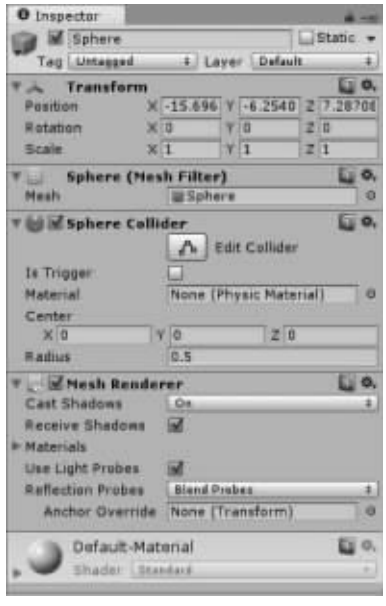

图 3-39　设置 Sphere(球体)对象的 Inspector 属性

1. Transform(转换)

(1) Position X,Y,Z：对象位置。

(2) Rotation X,Y,Z：对象旋转角度。

(3) Scale X,Y,Z：对象缩放。

2. Mesh Filter(网状模型)

选择外观的样子。

3. Sphere Collider(球体的碰撞)

(1) Is Trigger：碰撞到此对象是否会产生 Trigger(触发)事件。

(2) Material：材质。

(3) Center X,Y,Z：碰撞的轮廓的中心点的位置。

(4) Radius：碰撞的轮廓的半径。碰撞轮廓调整为 1.5 后的样子如图 3-40 所示。

4. Mesh Renderer(网状渲染)

(1) Cast Shadows：投射阴影。

(2) Receive Shadows：接收可以显示其他对象的阴影。

(3) Materials：材质。

(4) Size：网状渲染材质的数量。

(5) Element：网状渲染材质图。

(6) Use Light Probes：使用光探头。

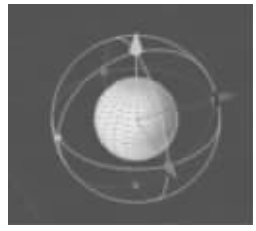

图 3-40 Sphere(球体)对象的碰撞轮廓尺寸调整为 1.5 后的样子

5. Default-Diffuse 纹理

(1) Shader：材质图的贴图运算方法。

(2) Main Color：材质主要的颜色。

(3) Tiling：材质显示的重复数量。

(4) Offset：材质图位移。

3.7.8 Capsule(球柱体)

Capsule 是一个球柱体对象，如图 3-41 所示。

常用属性如图 3-42 所示。

图 3-41 Capsule(球柱体)对象所呈现画面

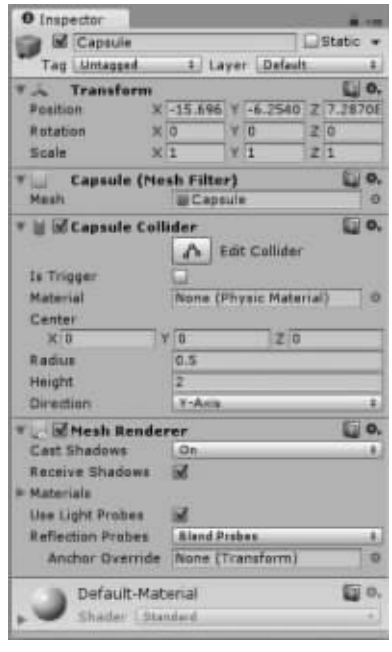

图 3-42 设置 Capsule(球柱体)对象的 Inspector 属性

1. Transform(转换)

(1) Position X,Y,Z：对象位置。

(2) Rotation X,Y,Z：对象旋转角度。

(3) Scale X,Y,Z：对象缩放。

2. Mesh Filter(网状模型)

选择外观的样子。

3. Capsule Collider(球柱体的碰撞)

(1) Is Trigger：碰撞到此对象是否会产生 Trigger(触发)事件。

(2) Material：材质。

(3) Center X,Y,Z：碰撞的轮廓的中心点的位置。

(4) Radius：碰撞的轮廓的半径。调整为 1.5 后的样子如图 3-43 所示。

(5) Height：碰撞的轮廓的高度。

(6) Direction：碰撞的轮廓的方向。

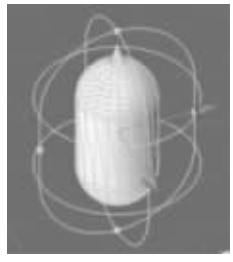

图 3-43　Capsule(球柱体)对象的碰撞外框大小调整为 1.5 后的样子

4. Mesh Renderer(网状渲染)

(1) Cast Shadows：投射阴影。

(2) Receive Shadows：接收可以显示其他对象的阴影。

(3) Materials：材质。

(4) Size：网状渲染材质的数量。

(5) Element：网状渲染材质图。

(6) Use Light Probes：使用光探头。

5. Default-Diffuse 材质

(1) Shader：材质图的贴图运算方法。

(2) Main Color：材质主要的颜色。

(3) Tiling：材质显示的重复数量。

(4) Offset：材质图位移。

3.7.9　Cylinder(圆柱体)

Cylinder 是一个圆柱体对象,可以用在圆柱形的物体上,如图 3-44 所示。

常用属性如图 3-45 所示。

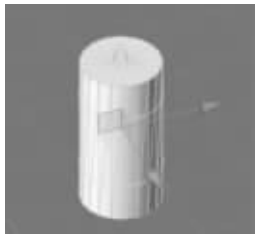

图 3- 44　Cylinder(圆柱体)对象所呈现画面

图 3-45　设置 Cylinder(圆柱体)对象的 Inspector 属性

1. Transform(转换)

(1) Position X,Y,Z：对象位置。

(2) Rotation X,Y,Z：对象旋转角度。

(3) Scale X,Y,Z：对象缩放。

2. Mesh Filter(网状模型)

选择外观的样子。

3. Cylinder Collider(圆柱体的碰撞)

(1) Is Trigger：碰撞到此对象是否会产生 Trigger(触发)事件。

(2) Material：材质。

(3) Center X,Y,Z：碰撞的轮廓的中心点的位置。

(4) Radius：碰撞的轮廓的半径。调整为 1.5 后的样子如图 3-46 所示。

(5) Height：碰撞的轮廓的高度。

(6) Direction：碰撞的轮廓的方向。

图 3-46　Cylinder(圆柱体)对象的碰撞轮廓 Radius(半径)调整为 1 和 Height 调整后的样子

4. Mesh Renderer(网状渲染)

(1) Cast Shadows：投射阴影。

(2) Receive Shadows：接收可以显示其他对象的阴影。

(3) Materials：材质。

(4) Size：网状渲染材质的数量。

(5) Element：网状渲染材质图。

(6) Use Light Probes：使用光探头。

5. Default-Diffuse 材质

(1) Shader：材质图的贴图运算方法。

(2) Main Color：材质主要的颜色。

(3) Tiling：材质显示的重复数量。

(4) Offset：材质图位移。

3.7.10　Plane(平面)

Plane 是一个平面对象，大多用在地板、墙壁等平面物体上。在我的第一个 Unity APP 中，通过 Plane(平面)对象组成了游戏的主菜单，如图 3-47 所示。

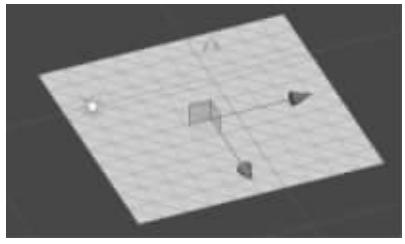

图 3-47　Plane(平面)对象所呈现画面

常用属性如图 3-48 所示。

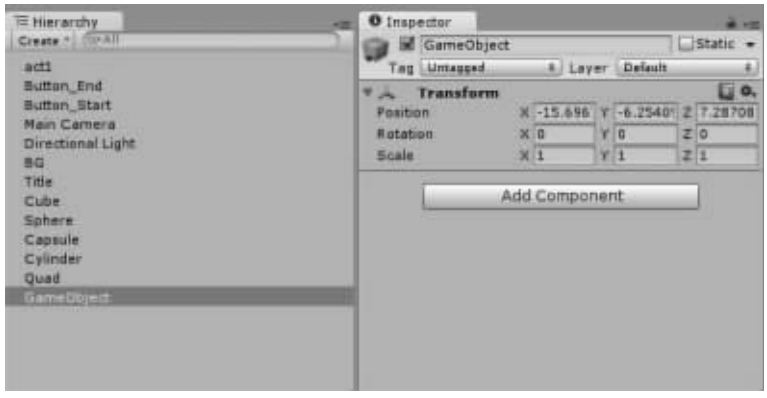

图 3-48　设置 Plane(平面)对象的 Inspector 属性

Transform(转换)：

(1) Position X,Y,Z：对象位置。

(2) Rotation X,Y,Z：对象旋转角度。

(3) Scale X,Y,Z：对象缩放。

Mesh Filter 网状模型

Mesh 碰撞

(1) Is Trigger：碰撞到此对象是否会产生 Trigger 触发事件。

(2) OnTriggerEnter、OnTriggerStay、OnTriggerExit。

(3) Material：材质。

(4) Convex：凸出来。

(5) Smooth Sphere Collision：光滑的球体碰撞。

(6) Mesh：网状。

Mesh Renderer 网状渲染。

(1) Cast Shadows：投射阴影。

(2) Receive Shadows：接收和可以显示其他对象的阴影。

(3) Materials。

(4) Size：网状渲染纹理的数量。

(5) Element：网状渲染纹理图。

(6) Use Light Probes：使用光探头。

Default-Diffuse 材质：

(1) Shader：材质图的贴图运算方法。

(2) Main Color：材质主要的颜色。

(3) Tiling：材质显示的重复数量。

(4) Offset：材质图位移。

3.7.11 Quad(四角形)

Quad(四角形)类似于 Plane(平面)对象，但其边缘是只有一个点长，且表面坐标在空间中的 XY 平面上。此外，Quad(四角形)只有两个三角形，而 Plane(平面)里面有近 200 个三角形，如图 3-49 所示。

使用 Quad(四角形)时在一个场景中的对象必须简单，它可以作为一个图像或影片显示。

Quad(四角形)对象的效果就是一个平面，它的常用属性如图 3-50 所示。

1. Transform(转换)

(1) Position X,Y,Z：对象位置。

(2) Ratation X,Y,Z：对象旋转角度。

(3) Scale X,Y,Z：对象缩放。

图 3-49　Quad(四角形)对象所呈现画面

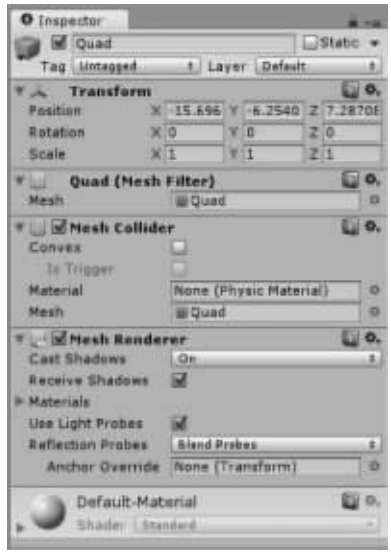

图 3-50　设置 Quad(四角形)对象的 Inspector 属性

2. Mesh Filter(网状模型)

设置外观的样子。

3. Mesh Collider(网状碰撞)

(1) Is Trigger：碰撞到此对象是否会产生 Trigger(触发)事件。

(2) Material：材质。

(3) Convex：凸出来，如图 3-51 所示。

(4) Smooth Sphere Collision：光滑的球体碰撞。

(5) Mesh：网状。

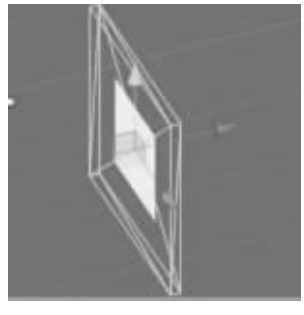

图 3-51　Quad(四角形)对象调整 Convex(凸出来)后的样子

4. Mesh Renderer(网状渲染)

(1) Cast Shadows：投射阴影。

(2) Receive Shadows：接收可以显示其他对象的阴影。

(3) Materials：材质。

(4) Use Light Probes：使用光探头。

5. Default-Diffuse 材质

(1) Shader：材质图的贴图运算方法。

(2) Main Color：材质主要的颜色。

(3) Tiling：材质显示的重复数量。

(4) Offset：材质图位移。

3.7.12 Create Empty(空白的游戏对象)

Create Empty 是一个空白的游戏对象，可以定义该空白对象的位置、角度与尺寸，其具有群组的观念，如图 3-52 所示。

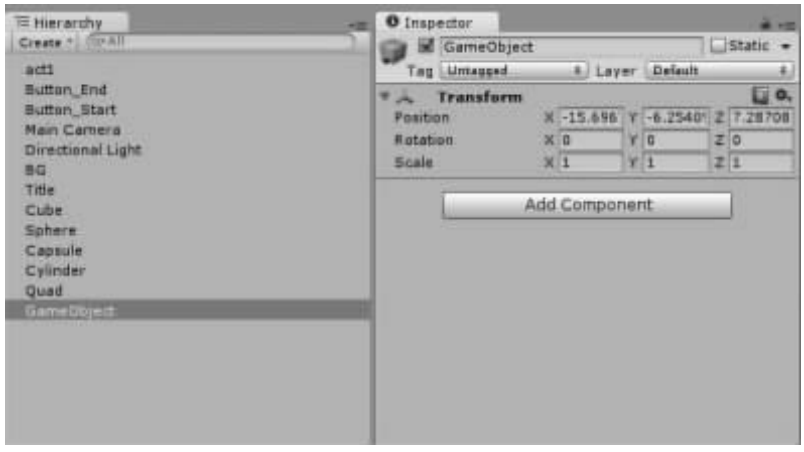

图 3-52 Create Empty(空白的 GameObject 游戏对象)

Transform(转换)：

(1) Position X,Y,Z：对象的位置。

(2) Rotation X,Y,Z：对象的旋转角度。

(3) Scale X,Y,Z：对象缩放。

另外，Create Empty 空白的游戏对象中拥有子目录的概念。在 Hierarchy 窗口中，用鼠标拖拉的方法把对象拉到这上面，就成为一个群组。

如果要撤销，也是用鼠标拖拉的方法，拉到其他地方可以解除群组的概念，如图 3-53 所示。

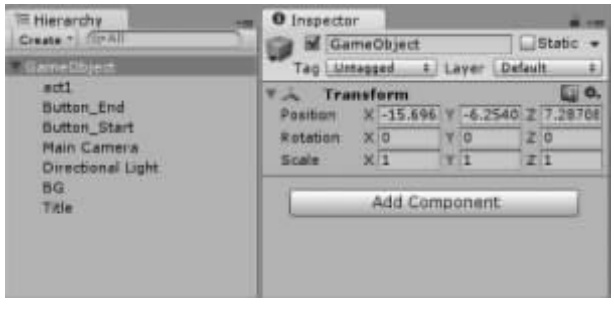

图 3-53 GameObject 游戏对象子目录的概念且可以成为一个群组

本章习题

1. 问答题

(1) 添加一个平面时，选择 GameObject|3D Object 中的什么命令？

A. Plane　　B. Cube　　C. Sphere　　D. Tree

(2) Unity 3D 支持哪些图片纹理格式？

A. JPG　　B. BMP　　C. GIF　　D. 以上皆是

(3) 在步骤 STEP15 中所提到的 Max Size 和 Format 的用途是什么？

2. 实作题

(1) 为本样例添加其他的按钮和图片，以增加游戏主画面的效果。

(2) 依照本章的教学，设计出您的游戏主画面。

(3) 依照本章的教学，通过 GameObject 对象来组成游戏的主菜单，并尝试添加不同的灯光、正方形和球体的游戏对象。