

沉浸式环境教程

目标：

- 渲染项目

加载玛雅人

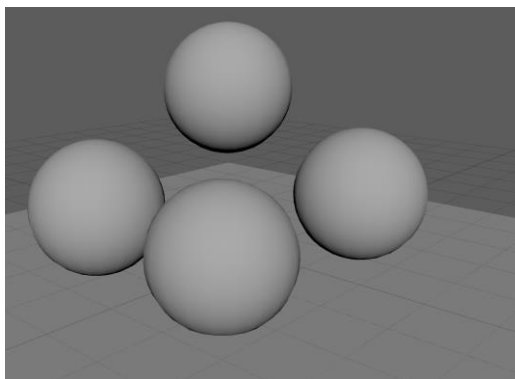


生成对象: 渲染

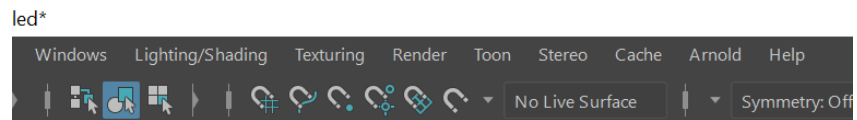
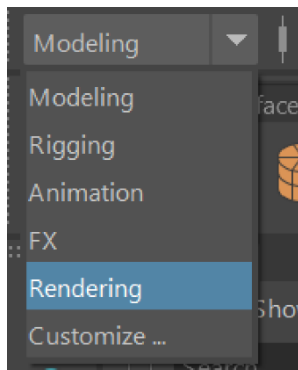
目的：创建元素并呈现为外部文件格式。

渲染是将场景中的对象提取到外部文件 (如 jpg/2) 的方法。Png.

创建一个简单的场景这样我们就可以测试渲染了



从这里, 转到下拉菜单部分并选择渲染, 这将修改菜单系统



ui 中还有一组快速呈现的图标。



让我们来看看这些图标



这是打开的呈现视图。渲染视图是您可以看到的位置所有的并能够确定所使用的渲染引擎。



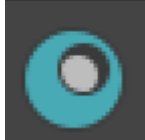
这是呈现当前视图。视图是您在场景中看到的内容。所以,如果你在透视视图中,这就是你会看到的,侧面视图,你会看到侧面视图等等。在渲染引擎之间交换时,必须重新渲染场景。



这将使用 **arnold** 呈现当前场景。阿诺德是照片逼真的渲染。它使用最大的处理能力,因为它通过反射光线的三个不同方面来计算物体的光线。**arnold** 元素将通过此渲染看到。

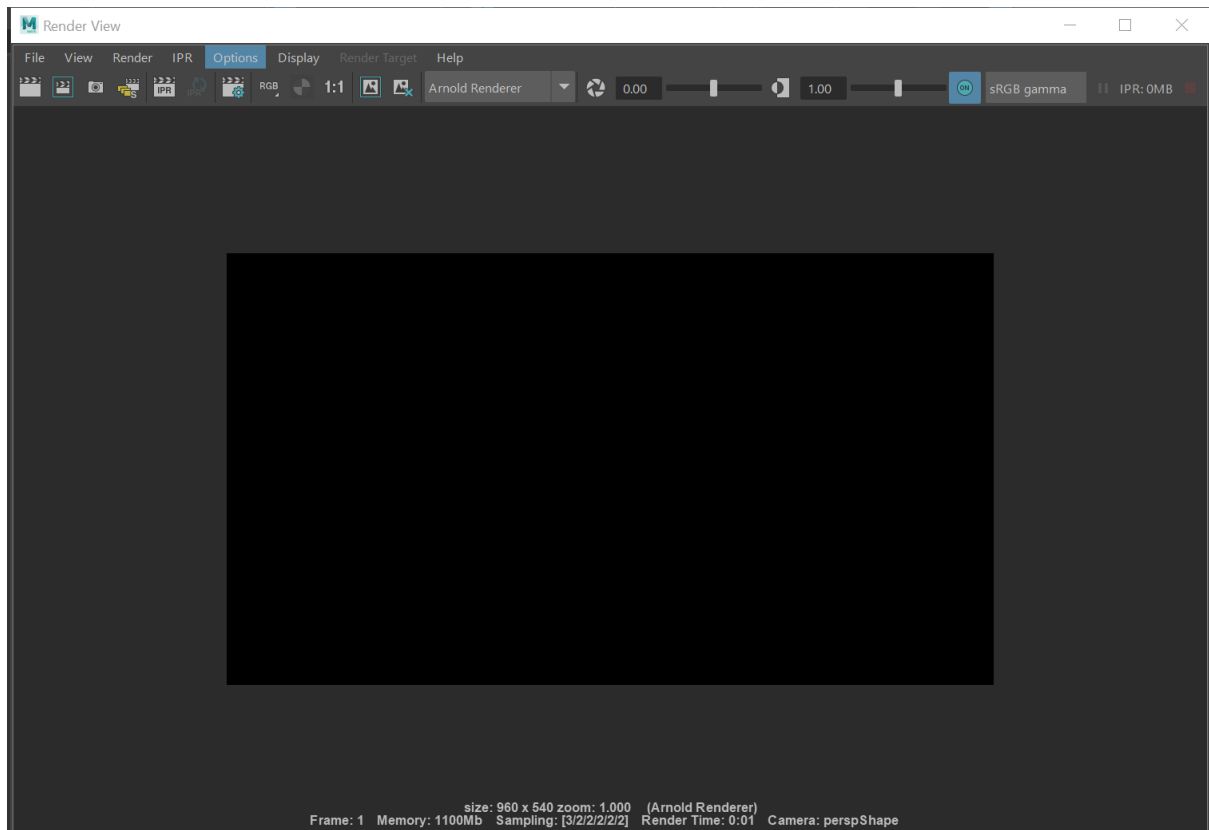


这是渲染设置,即放置文件的位置、要创建的文件类型。



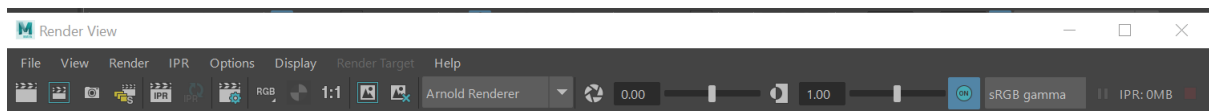
这是在超阴影系统。此区域列出了场景,和允许您快速预览设置项目上的材料。

首先,单击 "渲染当前视图" 图标。这打开呈现视图。

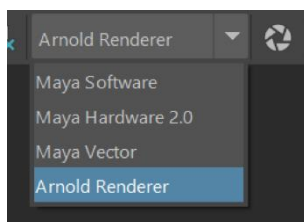


默认渲染为 **arnold**。正如你所看到的, 场景是纯黑色的, 这是因为我们没有应用任何 **arnold** 纹理。

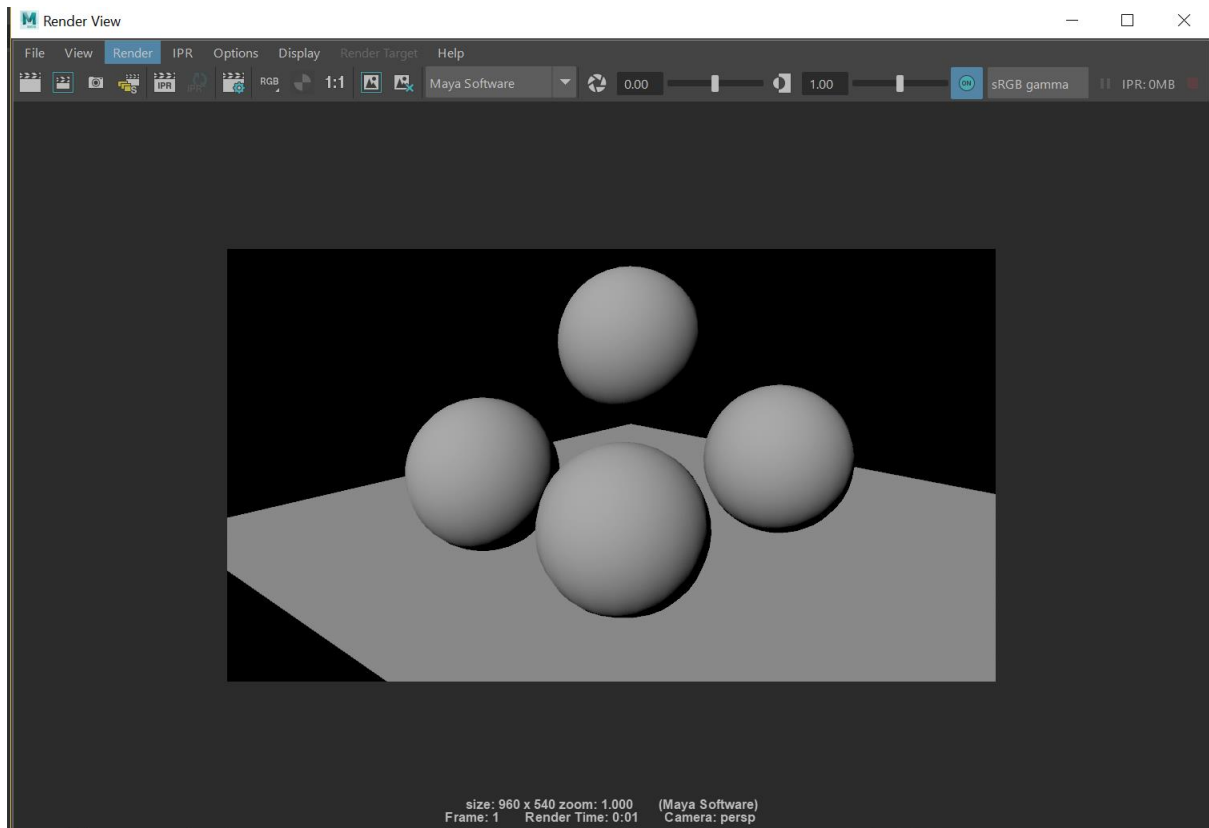
呈现窗口的菜单系统允许您更改大量元素



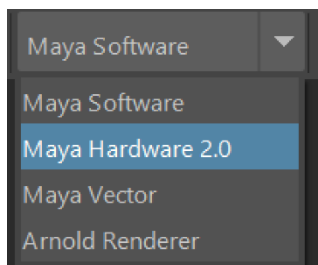
从下拉下, 我们将通过渲染引擎。



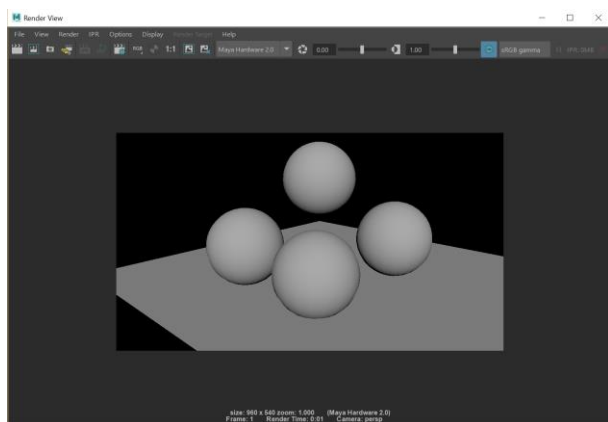
将渲染引擎更改为 **maya** 软件, 然后单击渲染当前场景图标。



正如你所看到的, 因为这是一种实现渲染的简单方法, 光线从表面反弹, 我们的基本纹理解析, 我们可以看到场景。下一次交换到 maya 硬件。

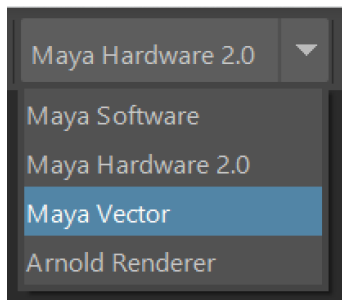


和以前一样, 我们已经更改了引擎, 现在单击以重新渲染场景。

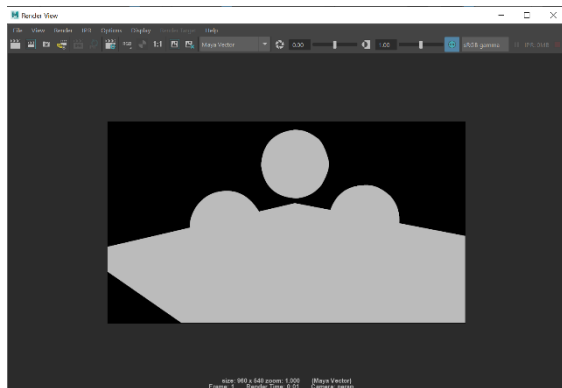


正如您所看到的, 图像渲染得非常快。这是一个伟大的方式来推动渲染到图形卡的激烈渲染场景。

最后,我们来看看玛雅向量。



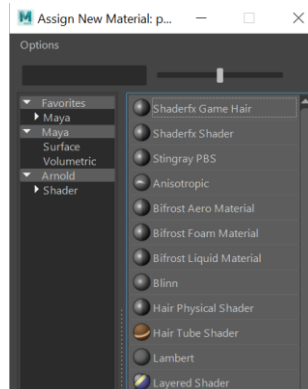
记住渲染场景。



请注意, 细节已离开渲染的对象, 我们只剩下所有内容的外部形状。

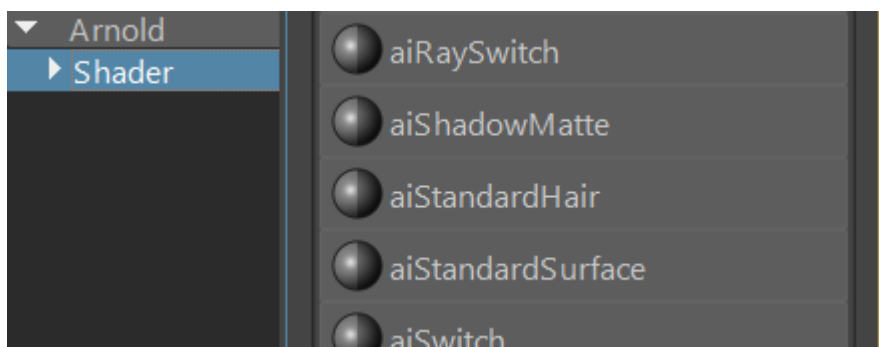
现在, 让我们添加一些材料到我们的场景。让我们为此使用菜单系统, 它是亮化的-> 分配新材料。您可以始终右键单击对象并选择分配新材料。在最前面的球体上这样做。

这将提供以下菜单系统



从侧面菜单部分选择 **arnold** 和着色立表面。

器, 然后从右侧菜单中选择独

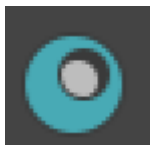


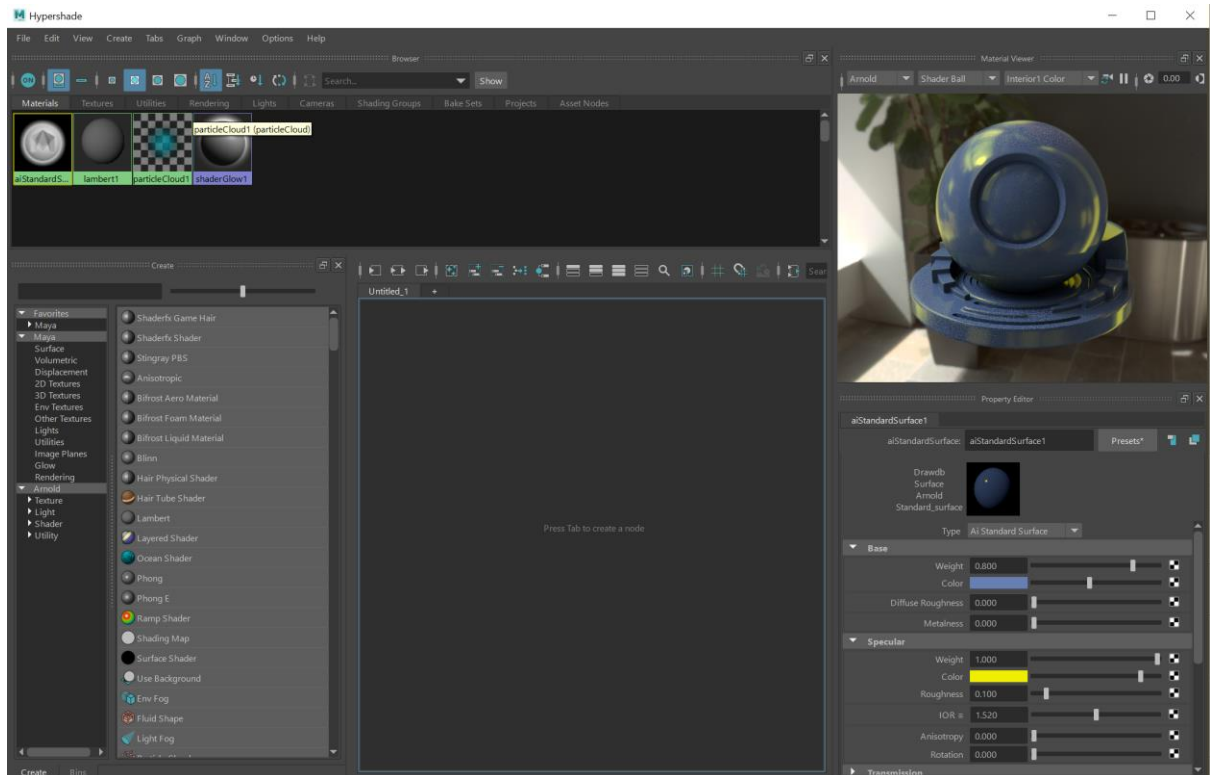
从属性编辑器有4个元素到着色器, 他们是基地, 镜面反射, 传输和子表面。



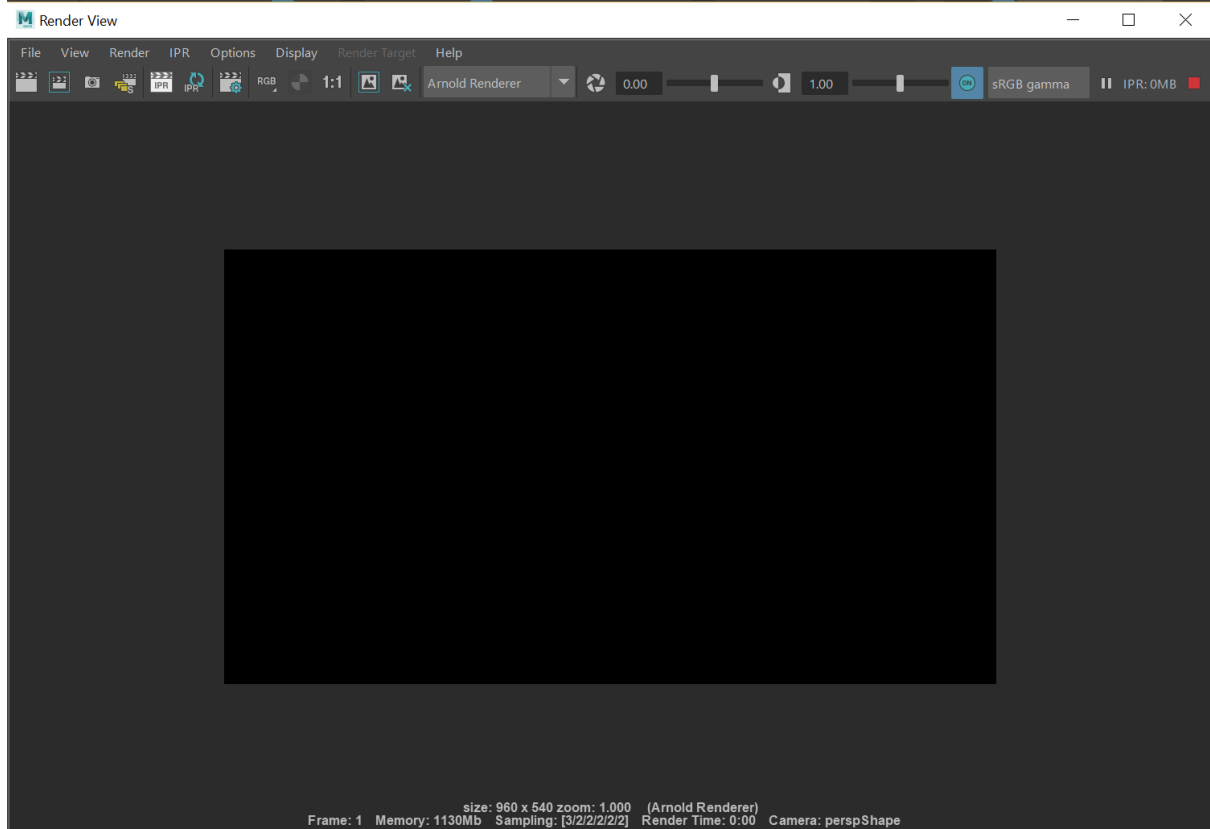
从这里开始, 玩的基础和镜面颜色, 以获得这些元素将如何呈现的感觉。

而不是渲染, 进入超阴影看看材料的选择会是什么样子。



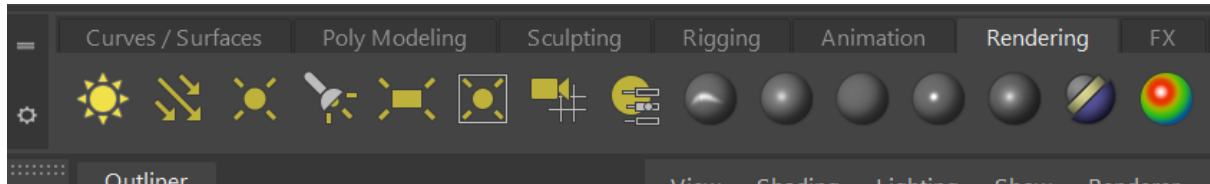


在这里, 您可以操作材质的元素并查看其外观类型。现在, 返回透视视图, 并使用 **ipr** 图标渲染当前场景。

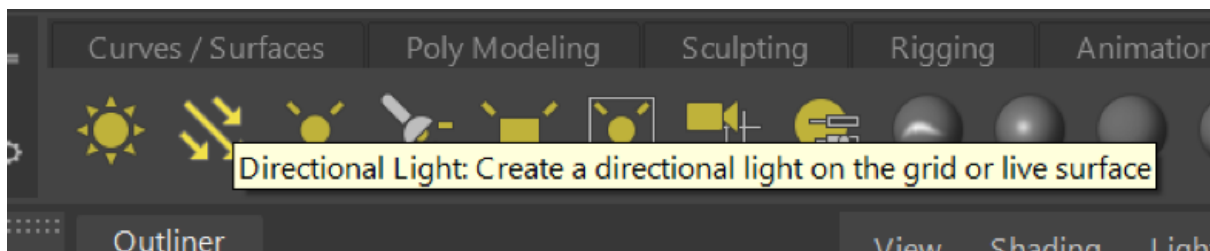


场景呈现为空白, 这是因为我们开始使用 **arnold** 纹理, 我们需要考虑正在发生的灯光。**arnold** 似乎不喜欢默认的光系统, 所以我们将把我们自己的方向光。

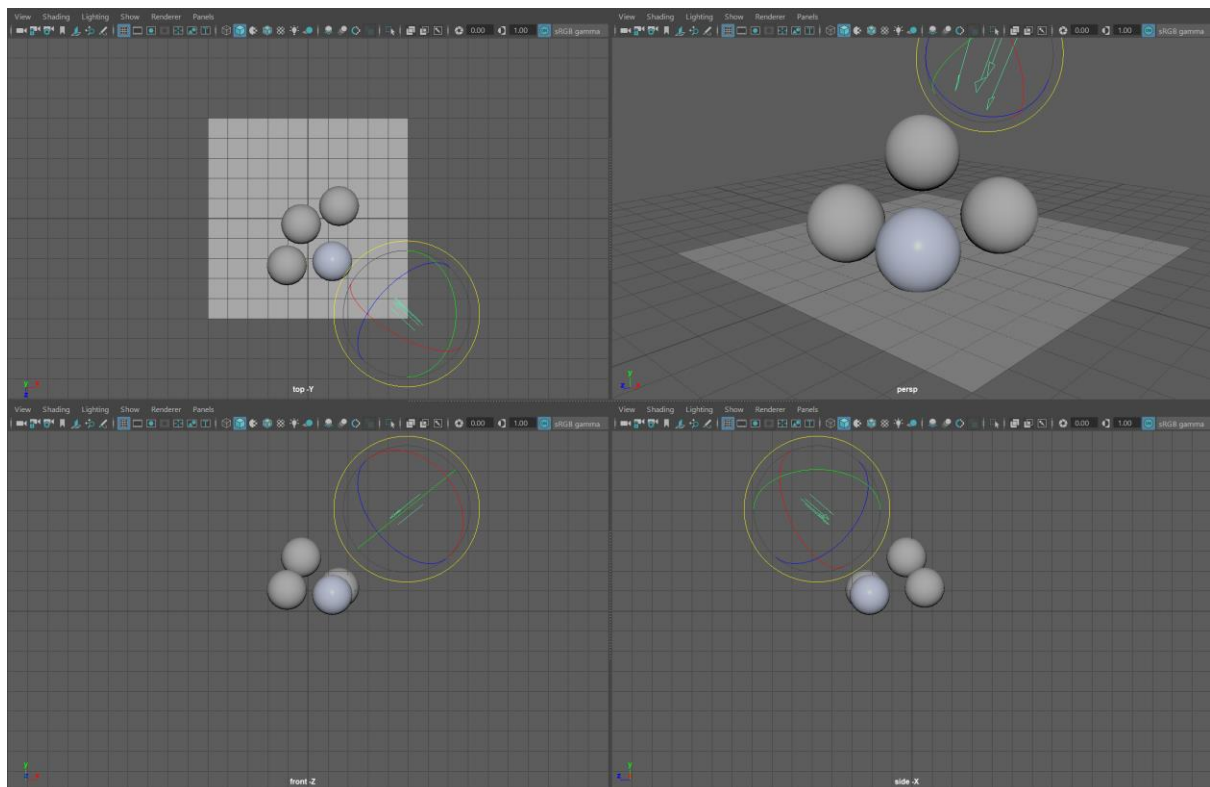
单击 "渲染" 选项卡



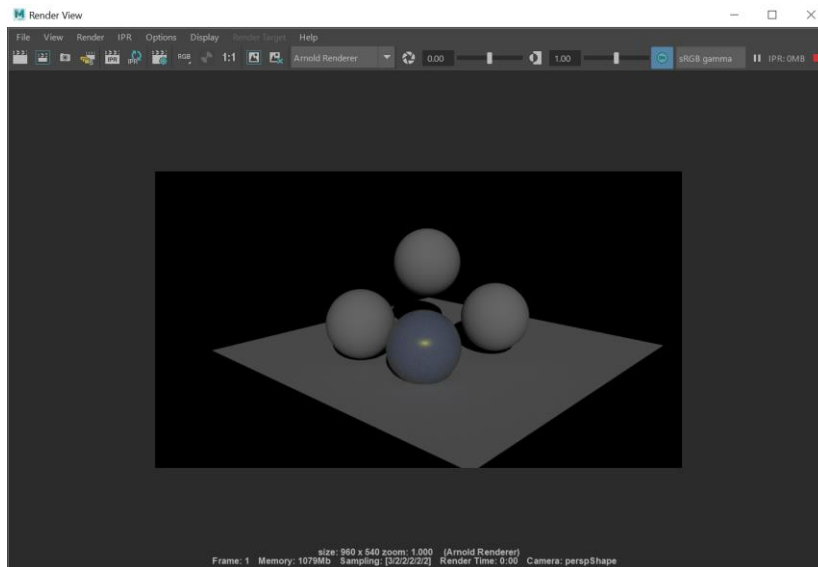
然后点击方向灯。



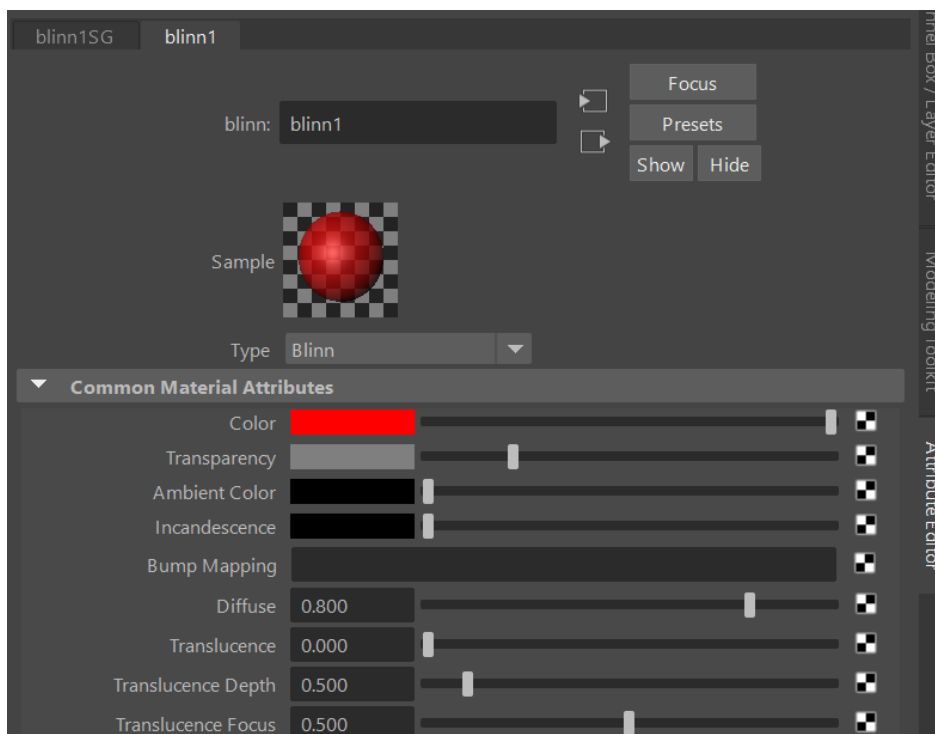
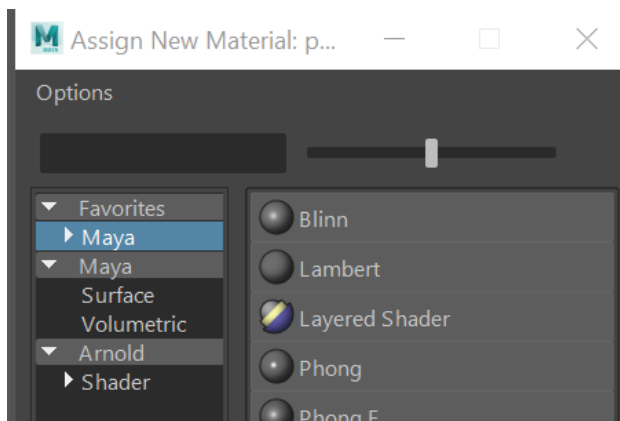
把这个光当作一个标准的物体来对待, 移动它, 使它面向球体。



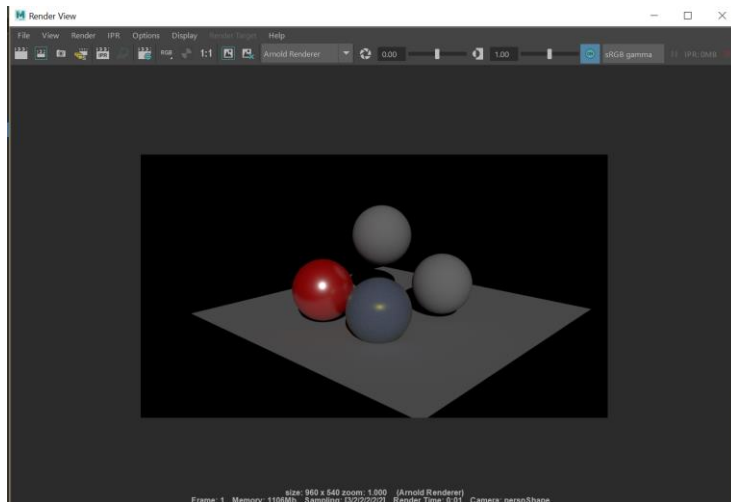
定位后, 再次使用 **ipr** 图标并查看呈现的输出。



选择另一个球并应用布林材料。



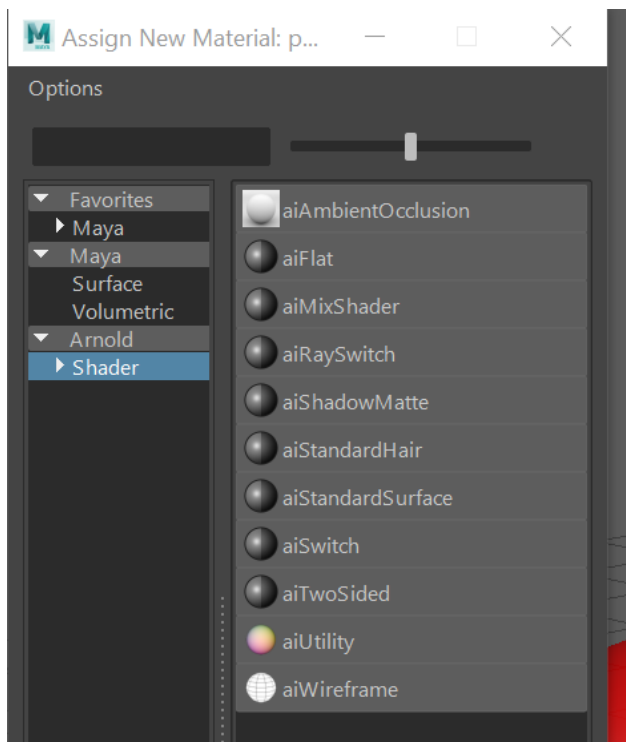
再次渲染场景



请注意, 知识产权呈现处理布林对象非常好, 如果您交换到其他渲染引擎和测试, 您将看到元素的呈现效果不是很好。

接下来, 我们将使中间球成为一个镜像球。

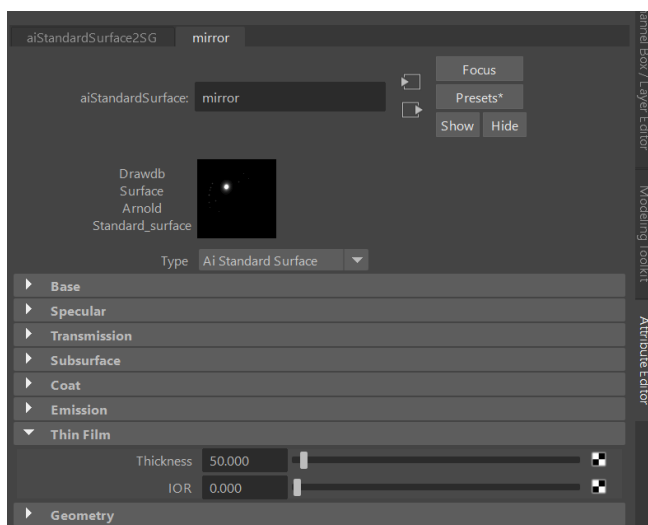
右键单击浮动球并分配新材料。选择独立表面



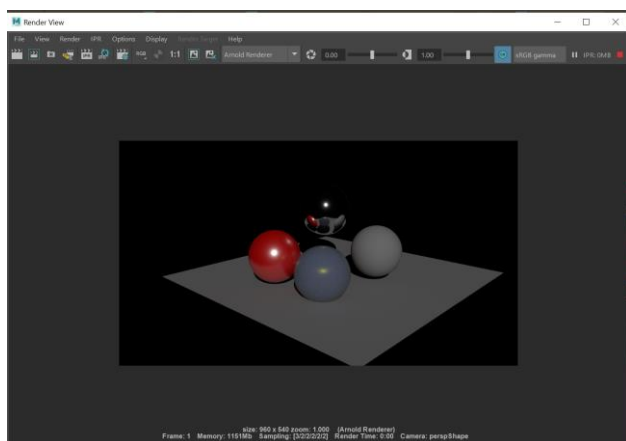
命名材质镜像, 然后向下滚动查看属性, 找到 "薄膜" 部分, 将厚度更改为 50, 并将其 ior 降至0。

ior 是折射率, 一个网站, 让我们你会看到更多关于这一点是:

<https://support.solidangle.com/display/A5AFMUG/Thin+Film>

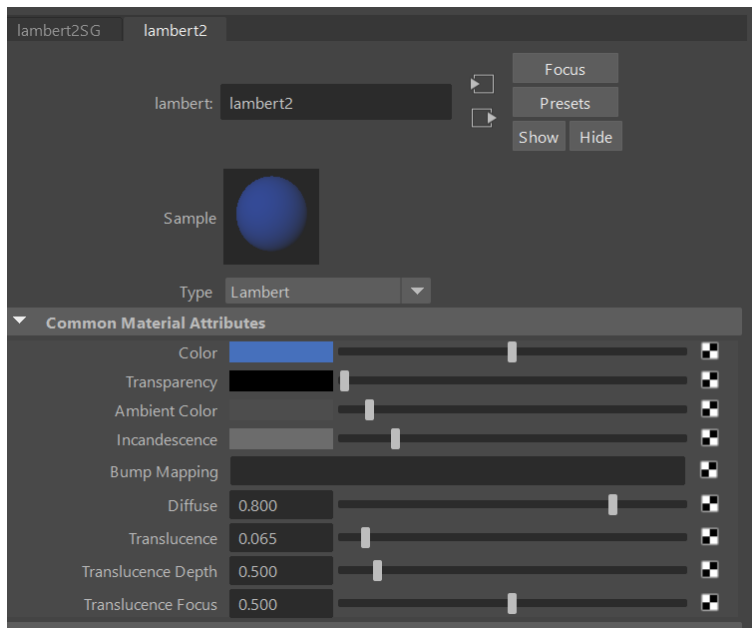


渲染图像后

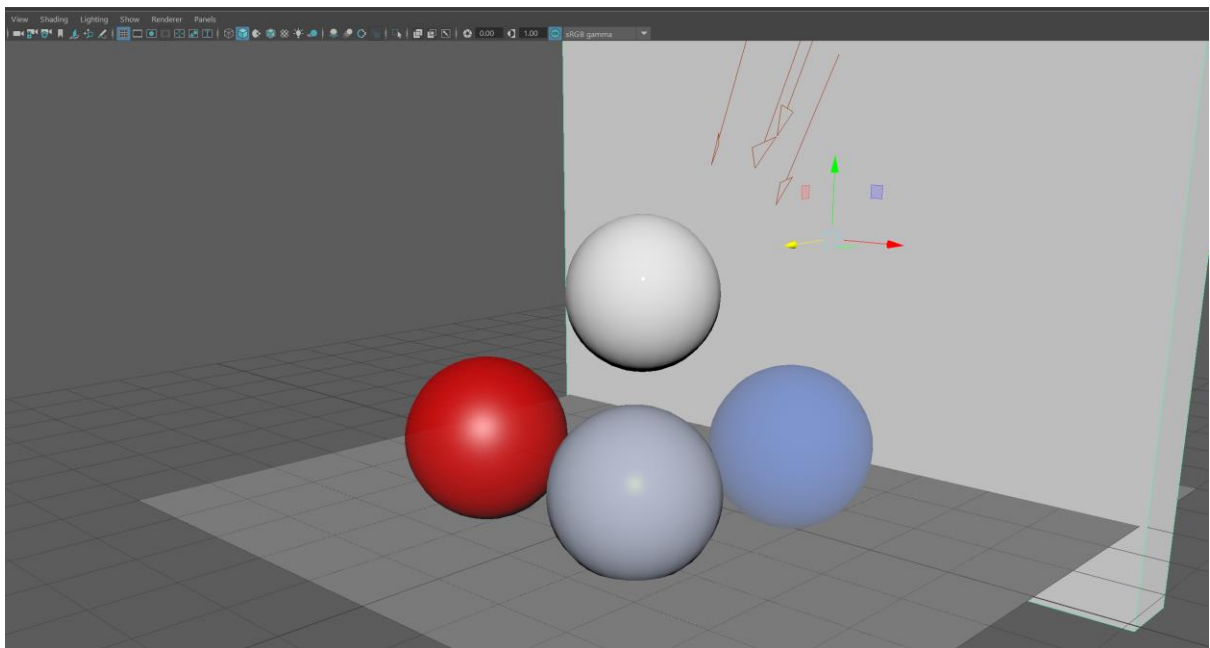


你应该有一个反映其他球体的球体。

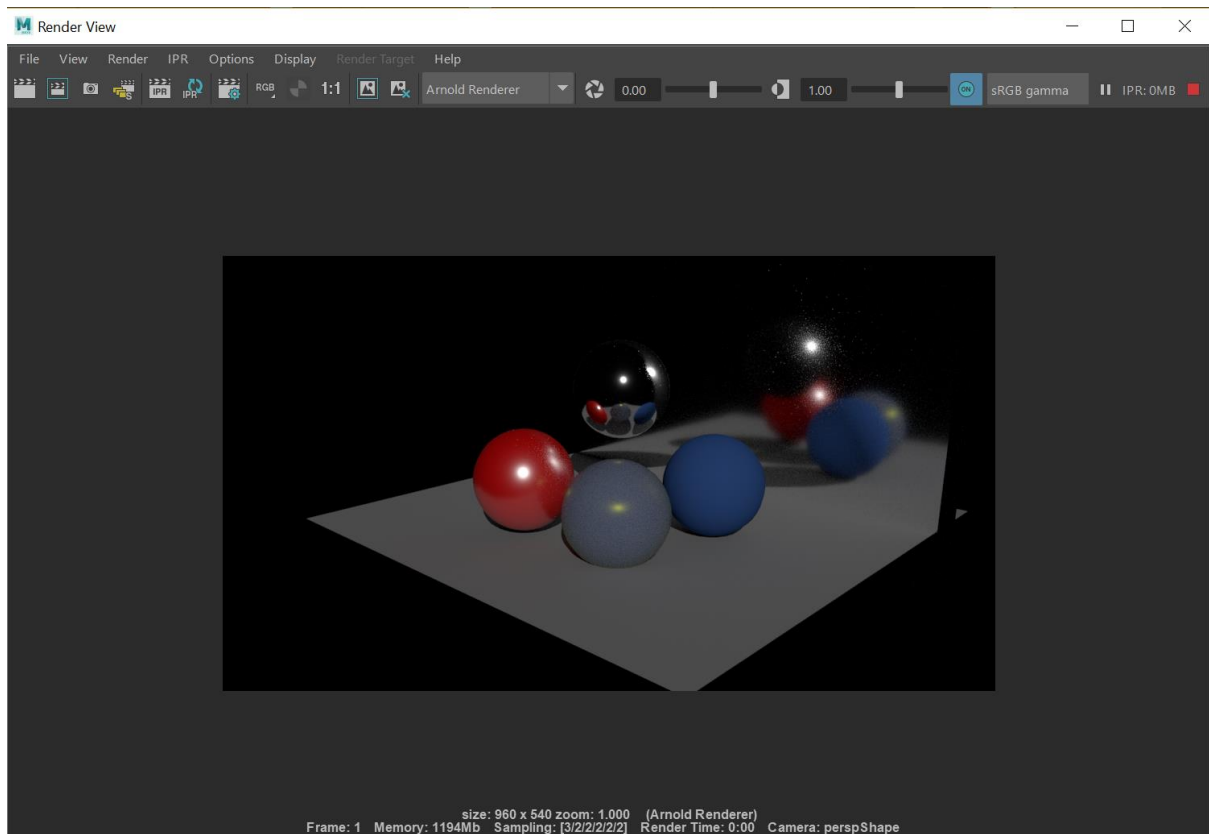
在最后的球体上, 放置任何你想要的材料。在这种情况下, 我刚刚应用了一个蓝色的兰伯特材料



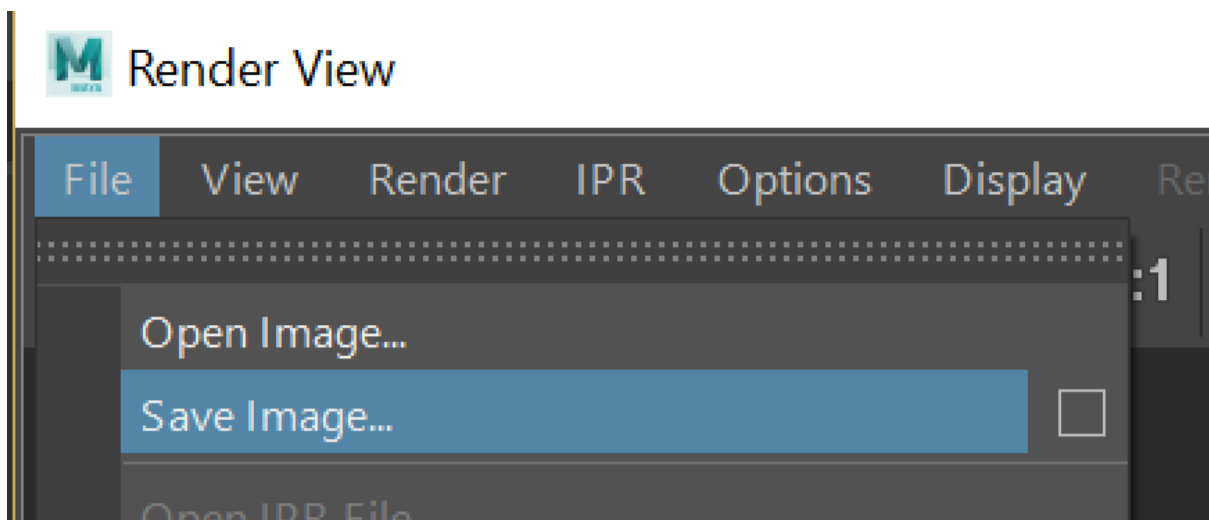
接下来, 在平面上的新球体后面添加一个立方体墙, 并应用镜像材料。你最终应该得到以下。



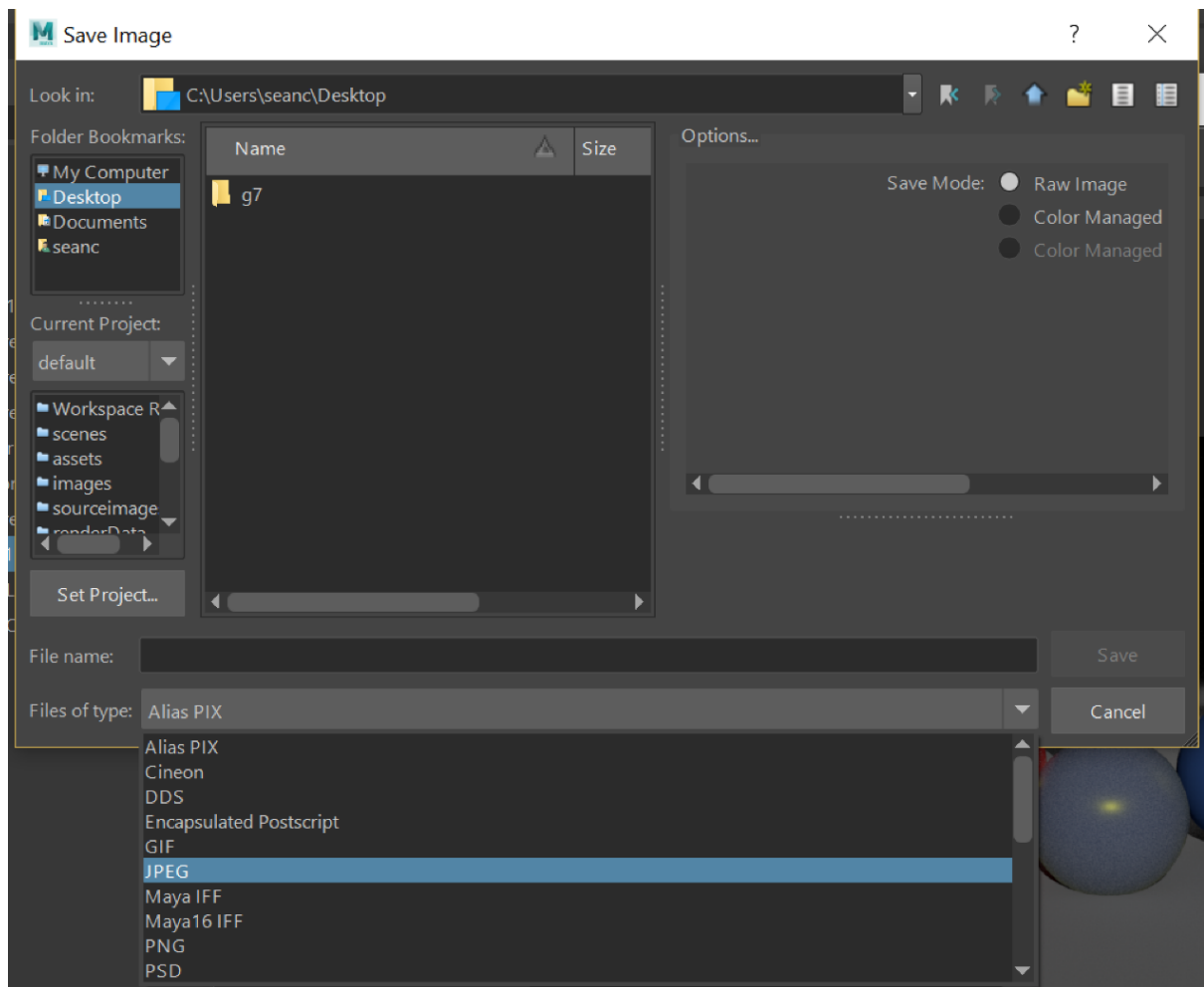
这使得像这样:



若保存图像, 请在渲染视图中转到 "菜单" 文件 > "保存图像"。



选择所需的位置和文件类型。

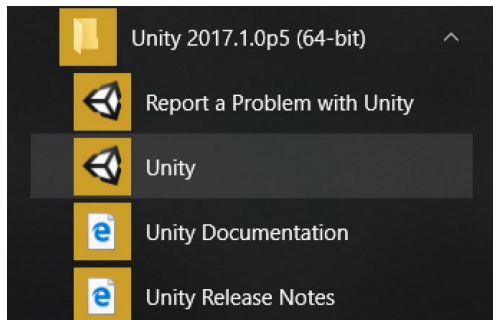


统一

目标：

- 场景和碰撞触发器

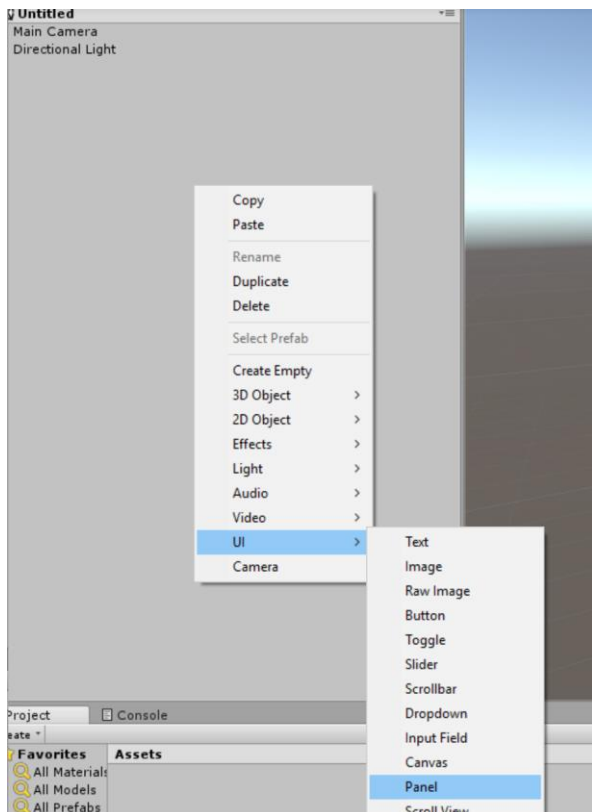
加载统一



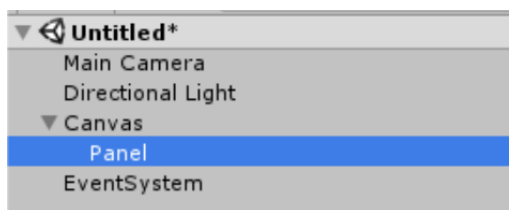
生成对象:现场s打瓦和菜单创建

目的：创建多个场景并在它们之间进行交换, 即菜单系统

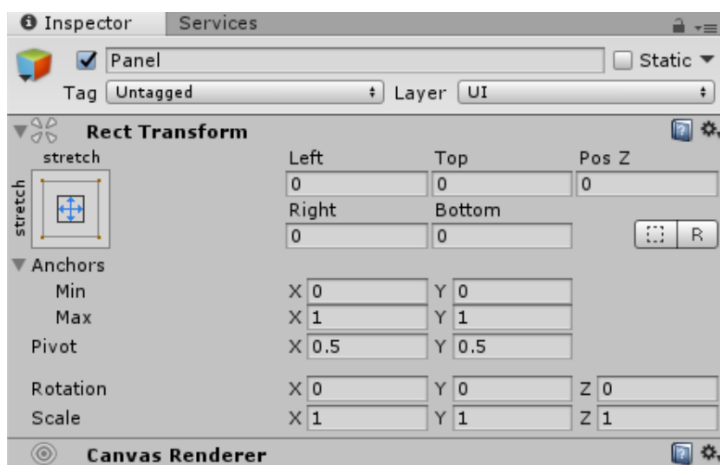
从右键单击开始层次结构并创建一个面板, 这是通过 **ui** 元素完成的。



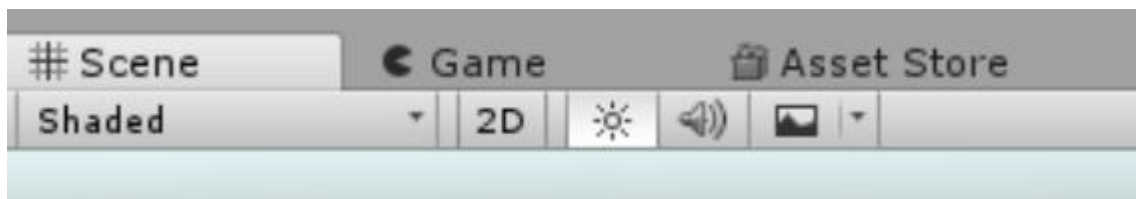
创建面板时, 您将看到它添加了一个事件系统和画布。"画布" 旨在保存用户界面的所有元素。



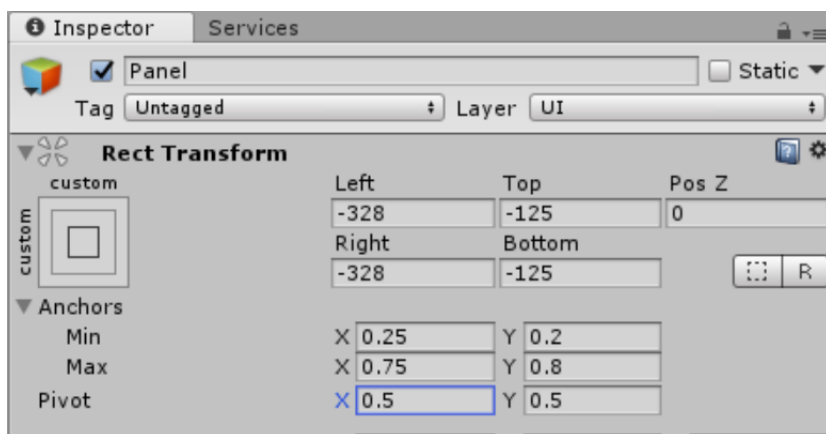
在检查专员方面, 专家小组有很多元素, 如锚点。锚阿雷将元素固定在面板内的一种方法是设置屏幕, 然后更改 "左"、"顶部"、"右侧" 和 "底部" 值。



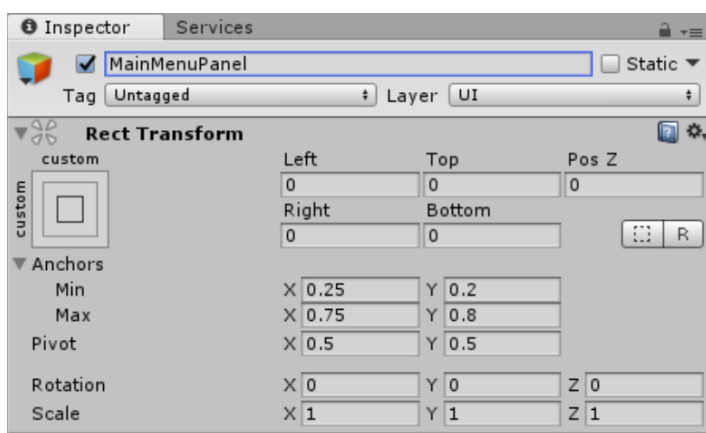
从3d 更改为 2d, 以便能够更轻松地查看场景。点击2d。



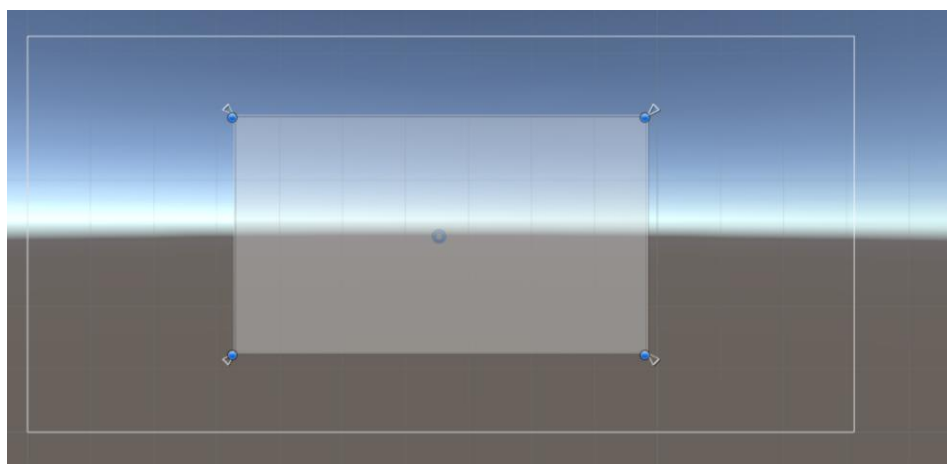
接下来的变化, 锚点的最小值分别为 x 0.25 & 0.75 和 y 的值为0.2 和0.8。完成此操作后, 您应该看到以下内容。



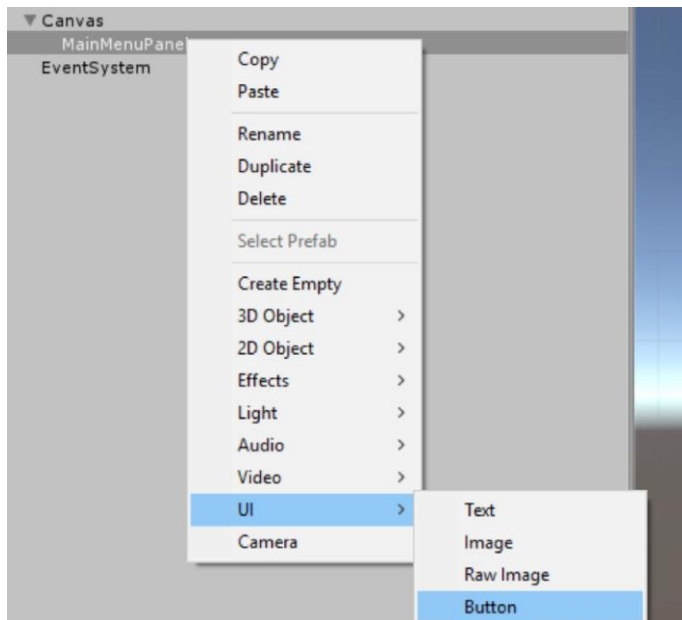
从这里归零出左、上、右和下的值。此外, 将面板的名称更改为主菜单面板.然后滚动以检查场景。



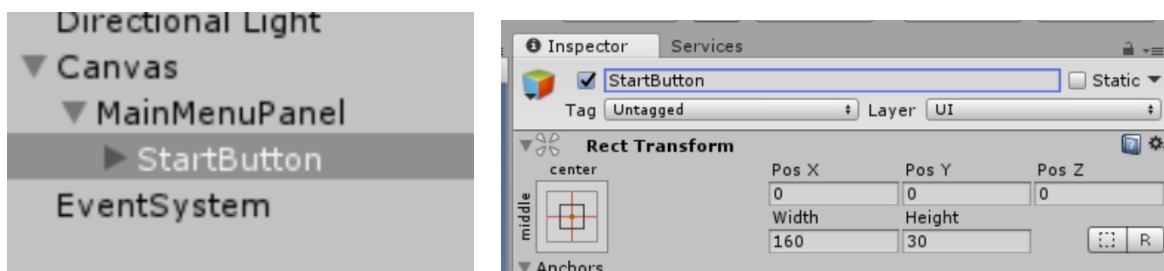
这些值将面板居中进入屏幕, 可以在下面查看。白色框是观景区域。



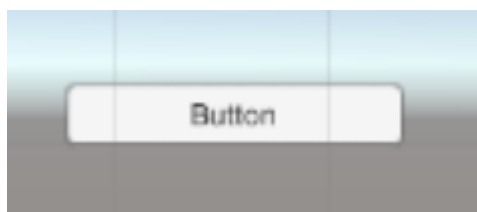
在我们的菜单系统,我们将有三个按钮, 开始游戏, 帮助和退出。首先, 右键单击层次结构面板, 并创建一个按钮作为子主菜单面板。



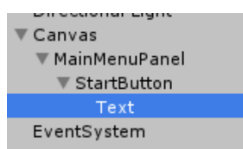
创建后, 为第一个按钮指定一个名称, 如启动按钮, 在检查器面板中。



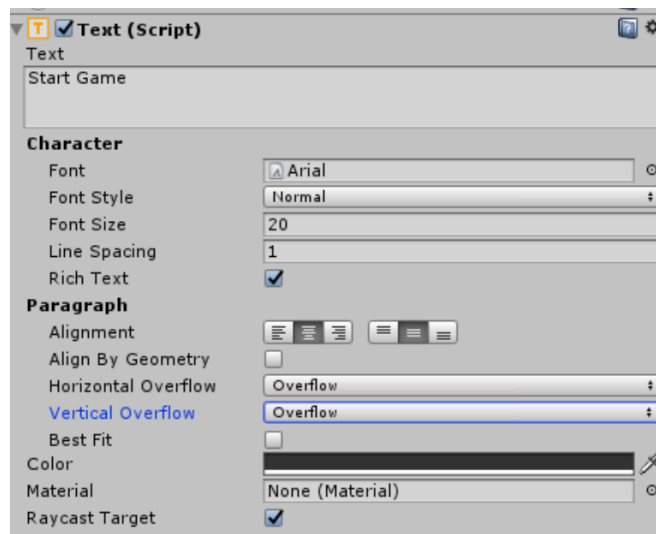
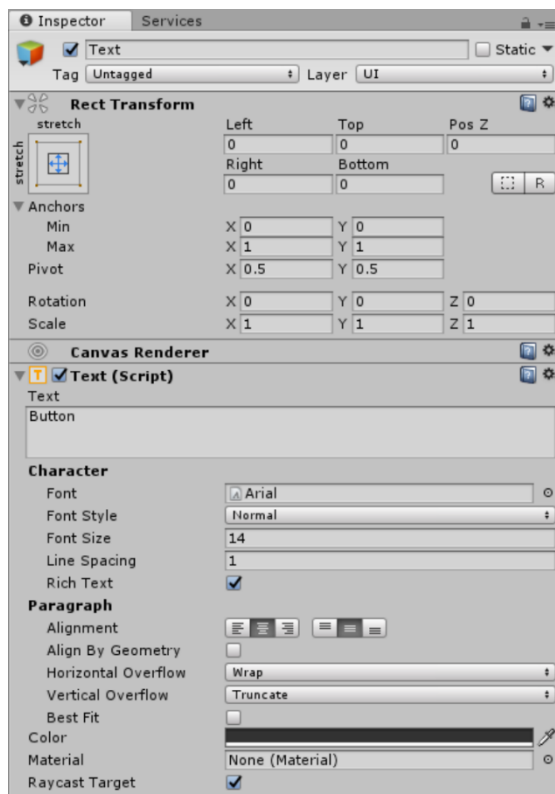
请注意, 在 "场景" 视图中, 该按钮上有按钮文本。



若要更改此操作, 请通过层次结构面板转到按钮的文本元素。



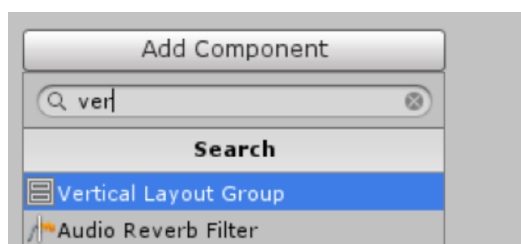
在检查器面板中,您可以更改文本、字体样式、字体大小以及文本的作用方式。在这种情况下,将文本从按钮更改为 "开始游戏",将字体增加到20并将垂直和水平溢出都改变为溢出。



当您查看场景视图时,该按钮以布局为中心,文本将填充场景。

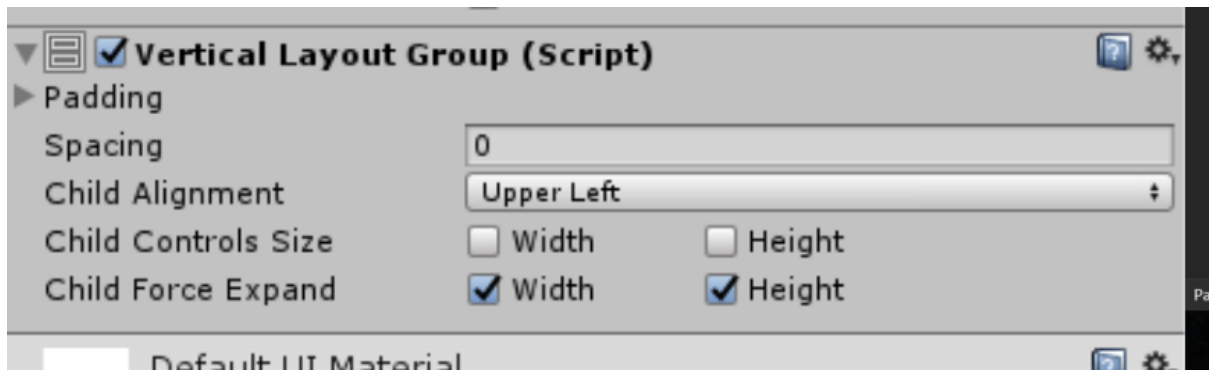
接下来,我们要创建其他两个按钮,我们可以创建一个新按钮并执行相同的步骤,但随后我们将处理按钮的对齐。最好的做法是将垂直布局组添加到主菜单面板.点击查看主菜单面板,然后将新组件添加到其检查器中。

若要查找,请使用搜索功能。

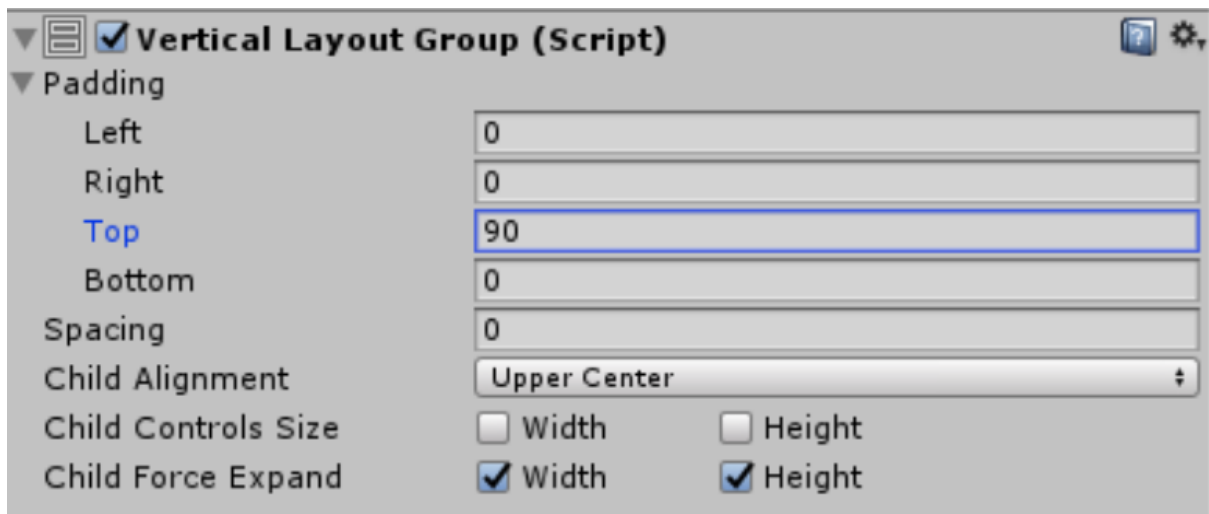


这将自动将按钮放在面板的左上角,这不是我们想要做的,因此,我们可以对垂直布局组对其进行定位。

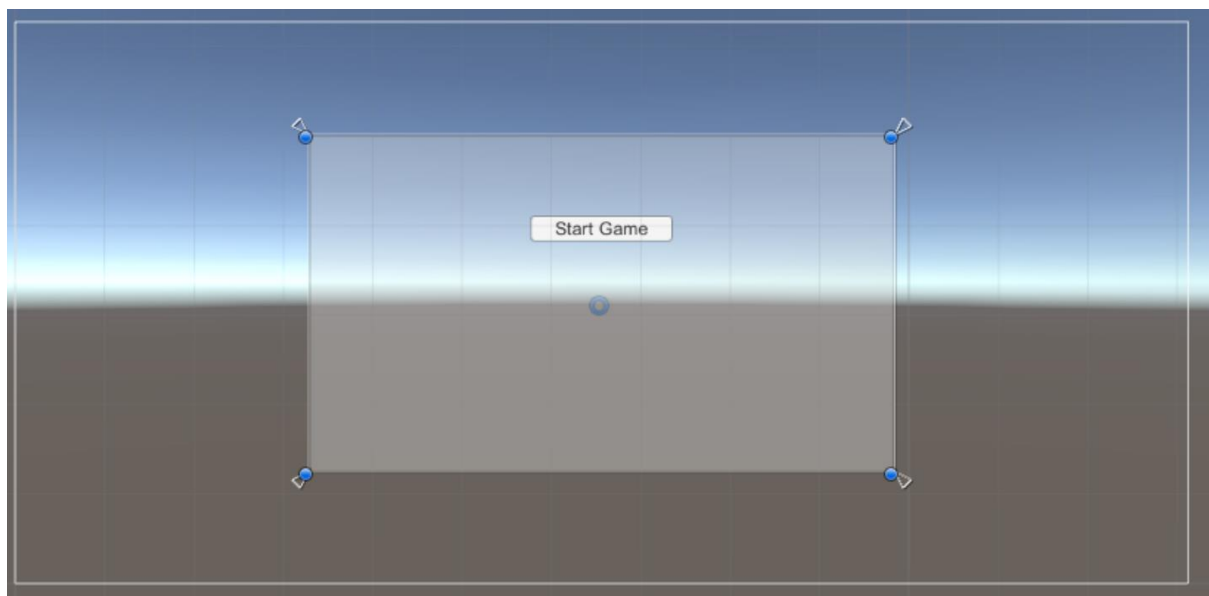




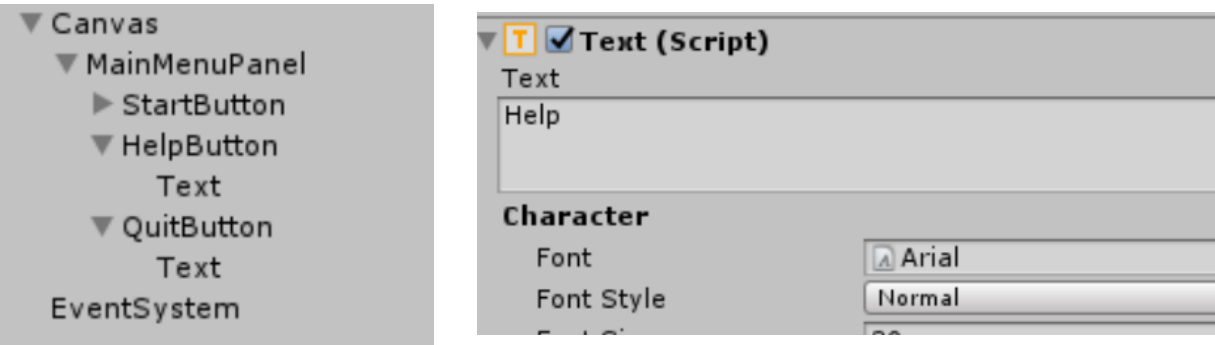
这些更改来自于下拉菜单菜单, 将其更改为上部中心并添加一个90的顶部填充。



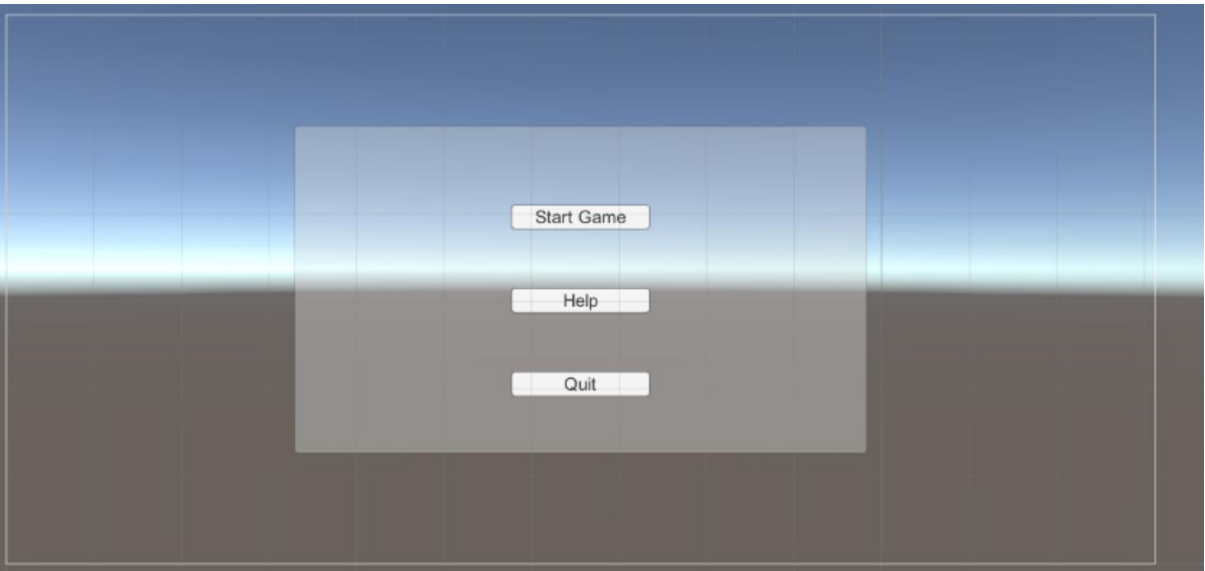
这将产生以下



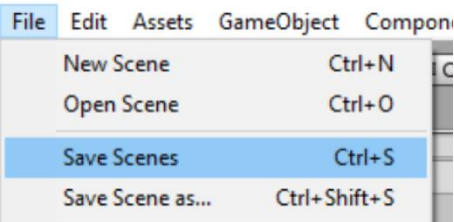
下我们可以复制和粘贴启动按钮对象, 并将其重命名为 "帮助并退出"。还可以更改每个按钮的文本以匹配它的功能。



这将产生以下结果

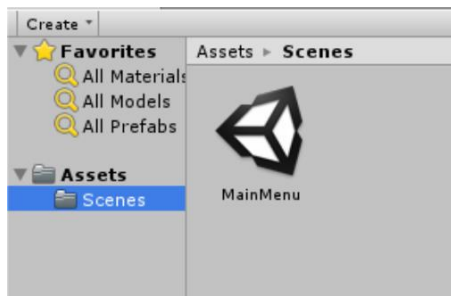


接下来, 我们必须保存场景, 以便我们可以访问它。转到菜单, 点击 "文件" > "保存场景"

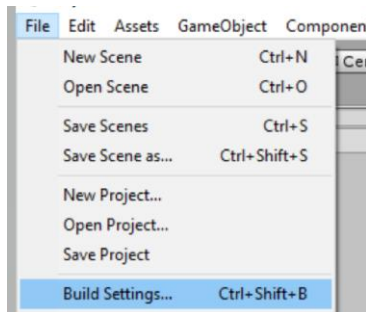


在 "资产" 文件夹中创建一个名为 "场景" 的文件夹, 并将此场景另存为主菜单.

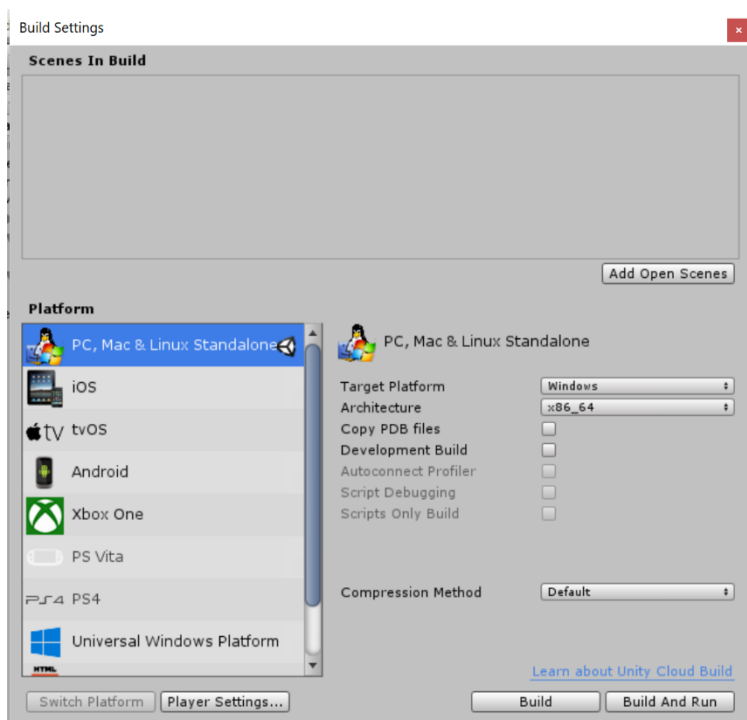
你最终应该得到以下。



现在转到 "菜单", 然后转到 "生成设置"



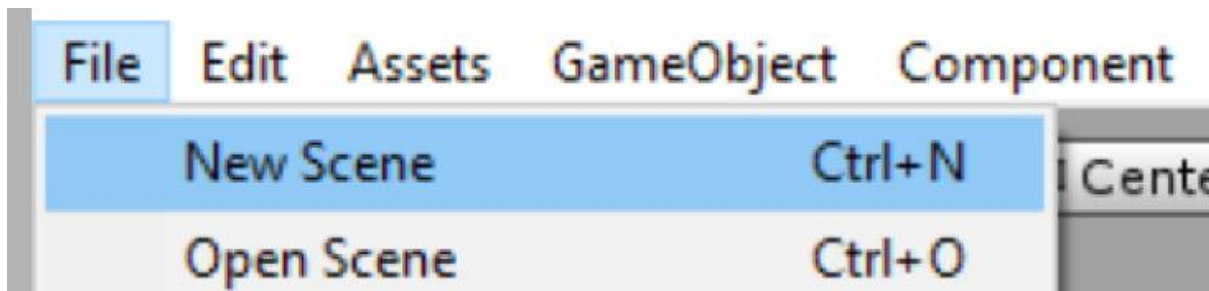
这将创建下面的弹出窗口, 让我们链接我们的场景。



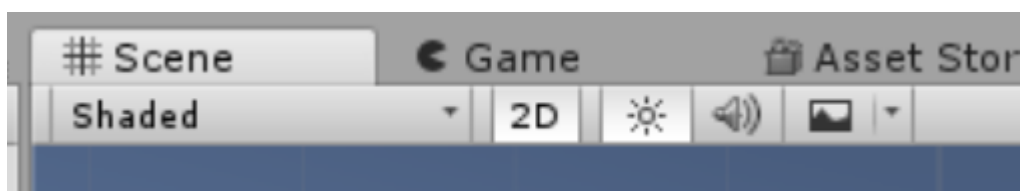
点击此处, 您应该会看到以下内容:



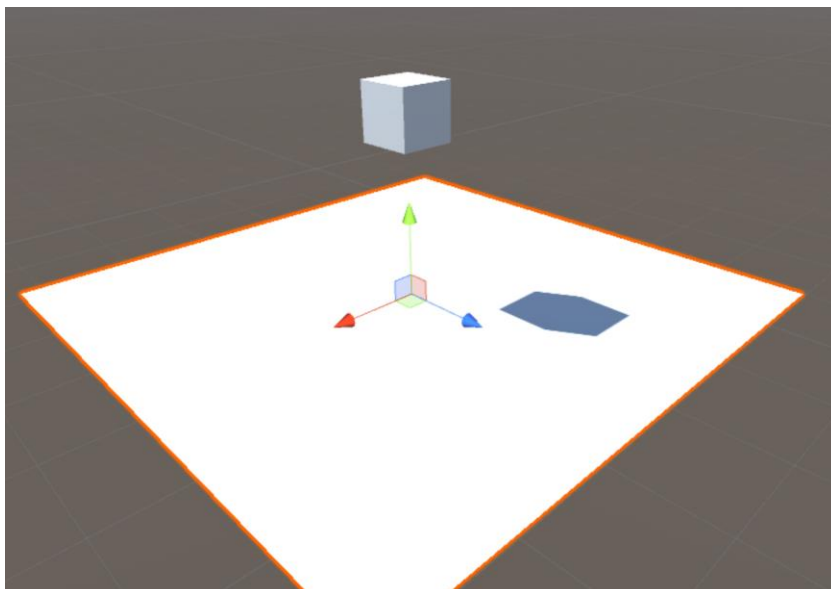
现在, 这已经列出, 请注意, 这是有的数量0在它旁边。下一步转到文件-> 新场景



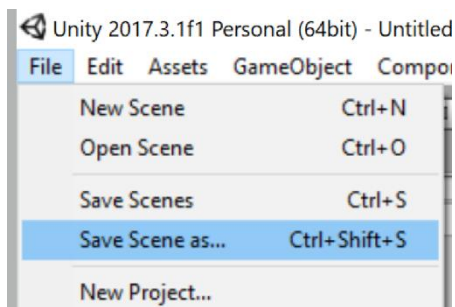
在此场景中, 通过单击2d 图标切换回3d 视图



向其中添加平面和多维数据集。将立方体提升到平面上方, 然后应用刚体到多维数据集, 以便当游戏运行时, 多维数据集将下降到飞机。

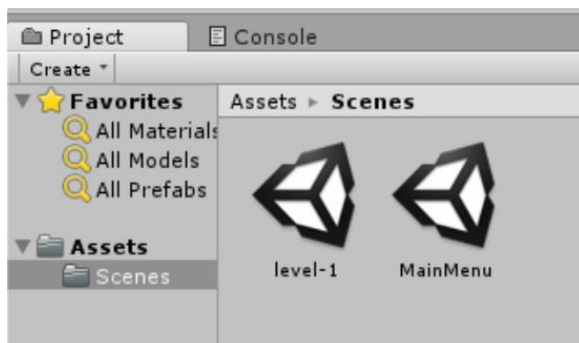


从这里转到菜单并执行文件 > 将场景另存为。这样, 您可以将场景保存到场景文件夹中。

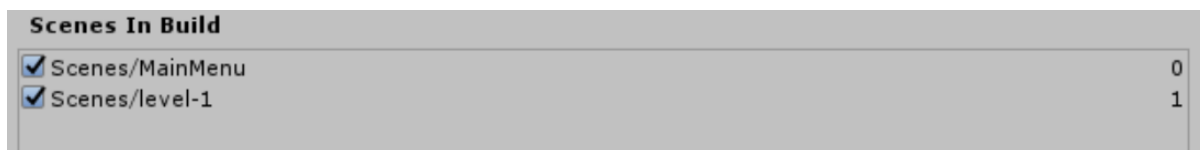


将此场景命名为1。

您应该看到以下内容。



再次, 进入生成设置并添加打开的场景。



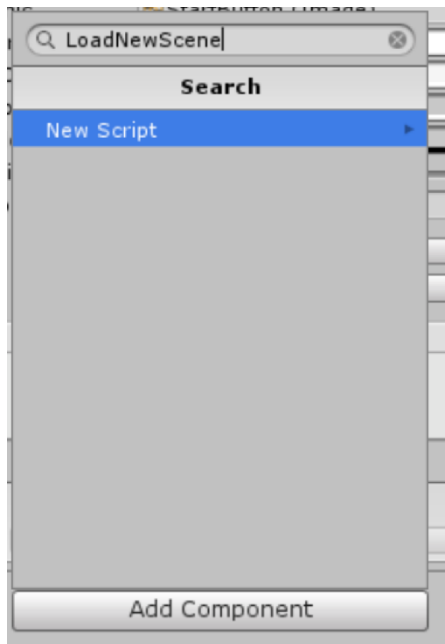
请注意, 级别1的索引号为1。这将对我们将应用于开始游戏按钮的代码有用。

现在开门通过双击它在资产文件夹中的主菜单场景。

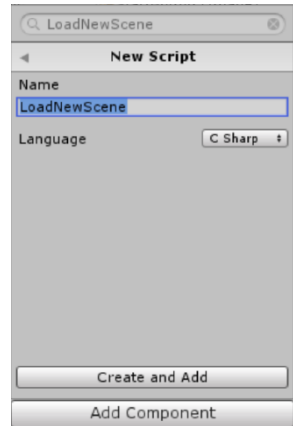
切换回2d 视图, 单击 "画布" 并将其居中晒你在做什么。



在此处, 点击启动按钮并添加一个新的抄本t它。



通过在搜索栏中输入未知名称, 添加组件将自动找出您需要该名称的新脚本, 单击新脚本, 然后创建和添加。



在资产文件夹中创建一个名为 "脚本" 的新文件夹, 然后将新脚本文件拖入其中。

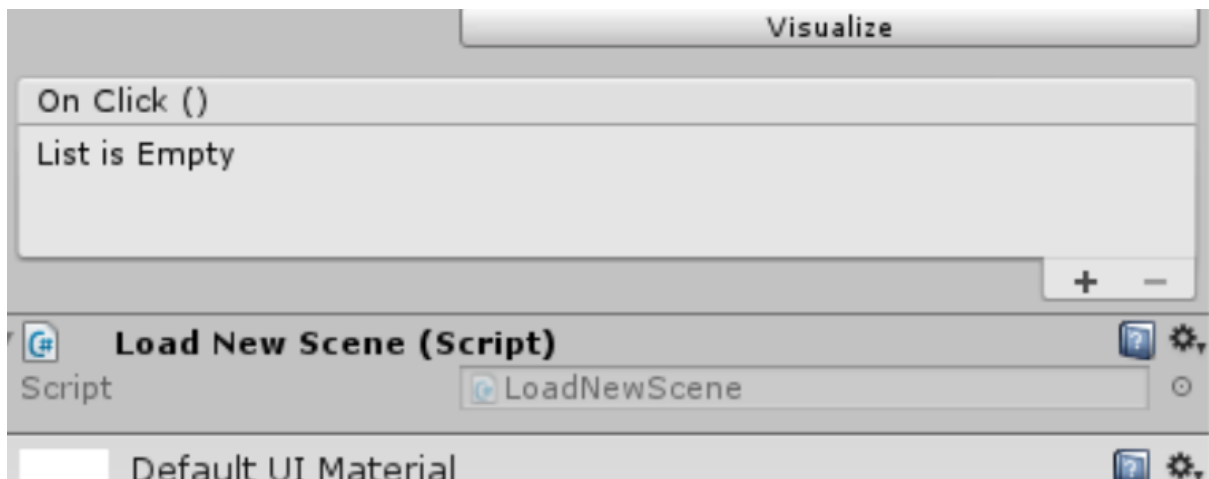


从这里在编辑器中加载脚本。并输入以下代码。

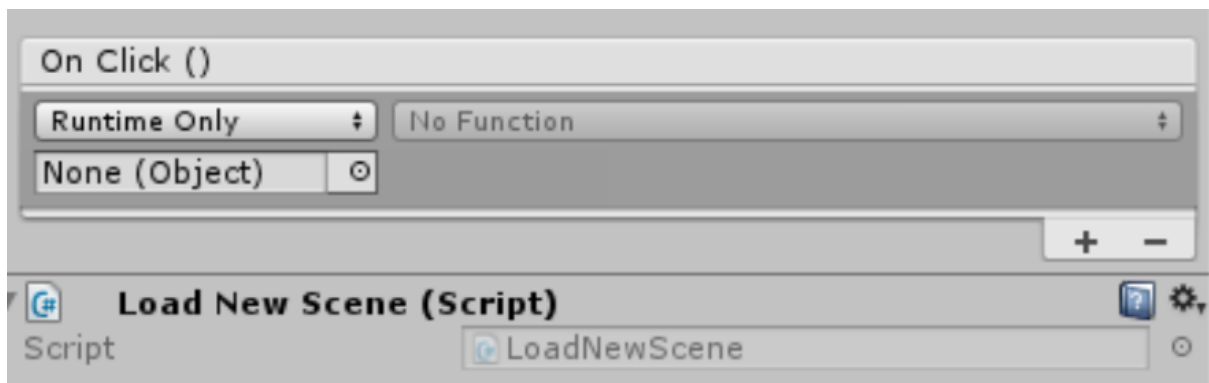
```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  public class LoadNewScene : MonoBehaviour {
7
8      public void LoadByIndex(int sceneIndex)
9      {
10         SceneManager.LoadScene(sceneIndex);
11     }
12 }
13
```

请注意要添加的 "使用" 行。一旦完成, 保存并回到团结。

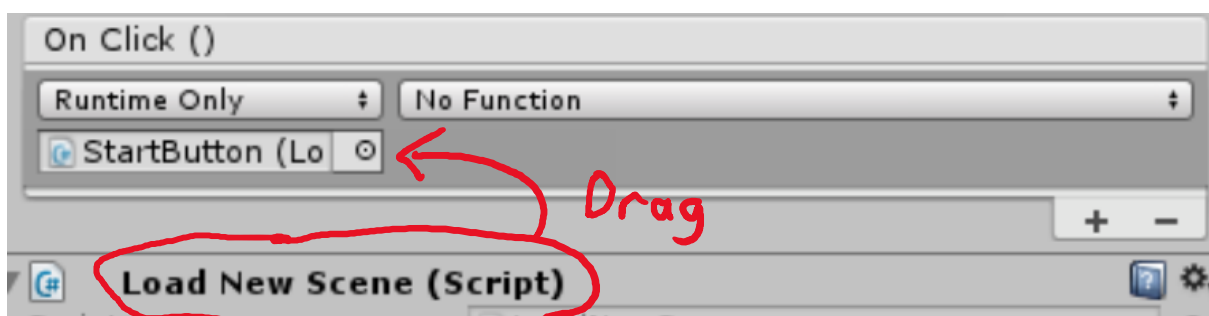
威特赫单击 "开始" 按钮, 您应该会看到以下



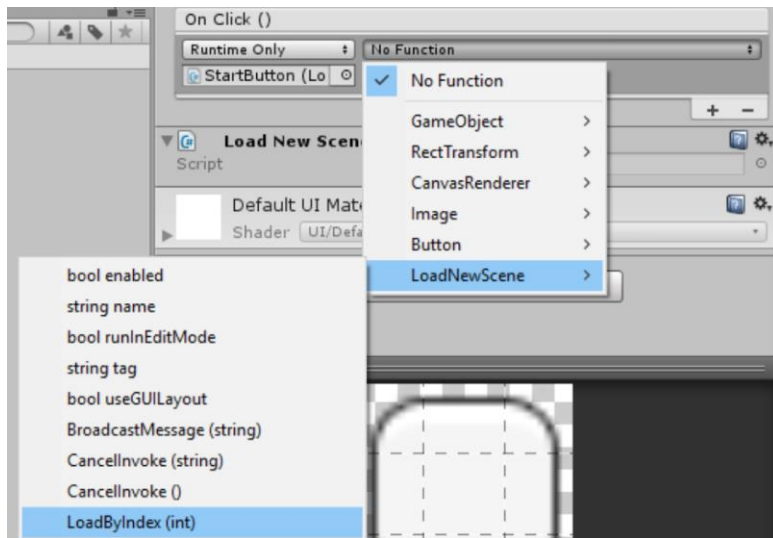
由于这是一个按钮, 我们需要添加一个点击事件, 这个事件将是我们刚刚编写的脚本, 从点击 + 符号开始点击点击部分。您应该看到以下内容。



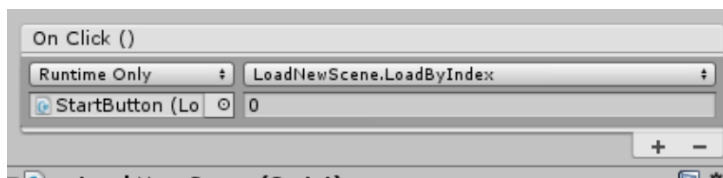
接下来, 我们将加载的新场景拖到 "无对象" 设置上。它应该改变如下:



从这里我们需要能够提供我们也想移动的场景, 这是通过下拉菜单从脚本中选择函数来完成的。所以, 点击无函数->负载新闻场景->负载 byindex

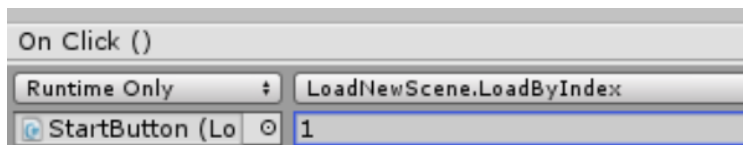


这将单击时的区域更改为:

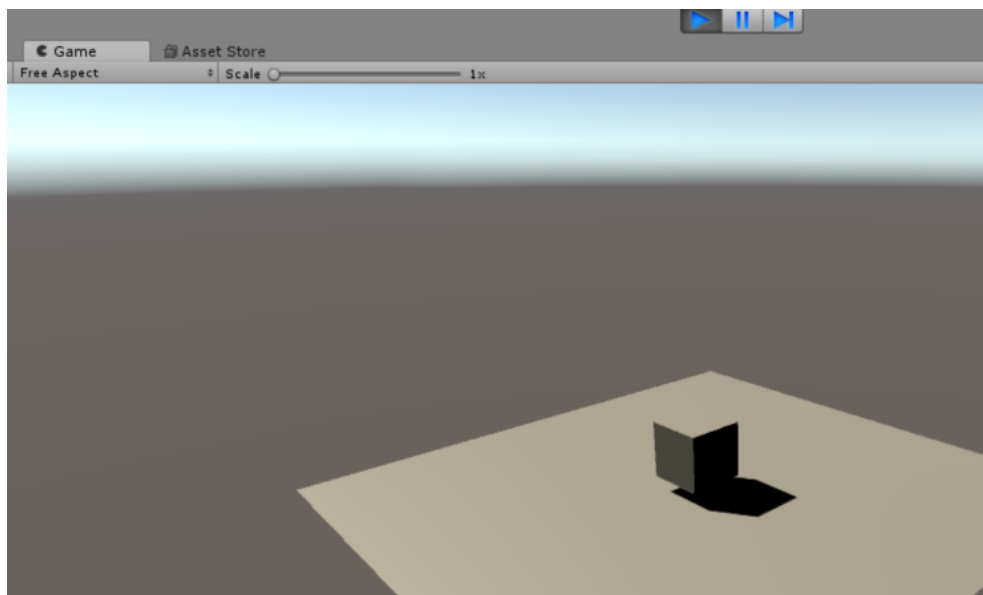


正如你可以看到的参数, 它被设置为 0, 如果你从这里回忆, 我们想点击并转到一级场景, 它的索引号是 1, 它位于生成设置中。

因此, 将0更改为 1, 然后运行它并测试单击。

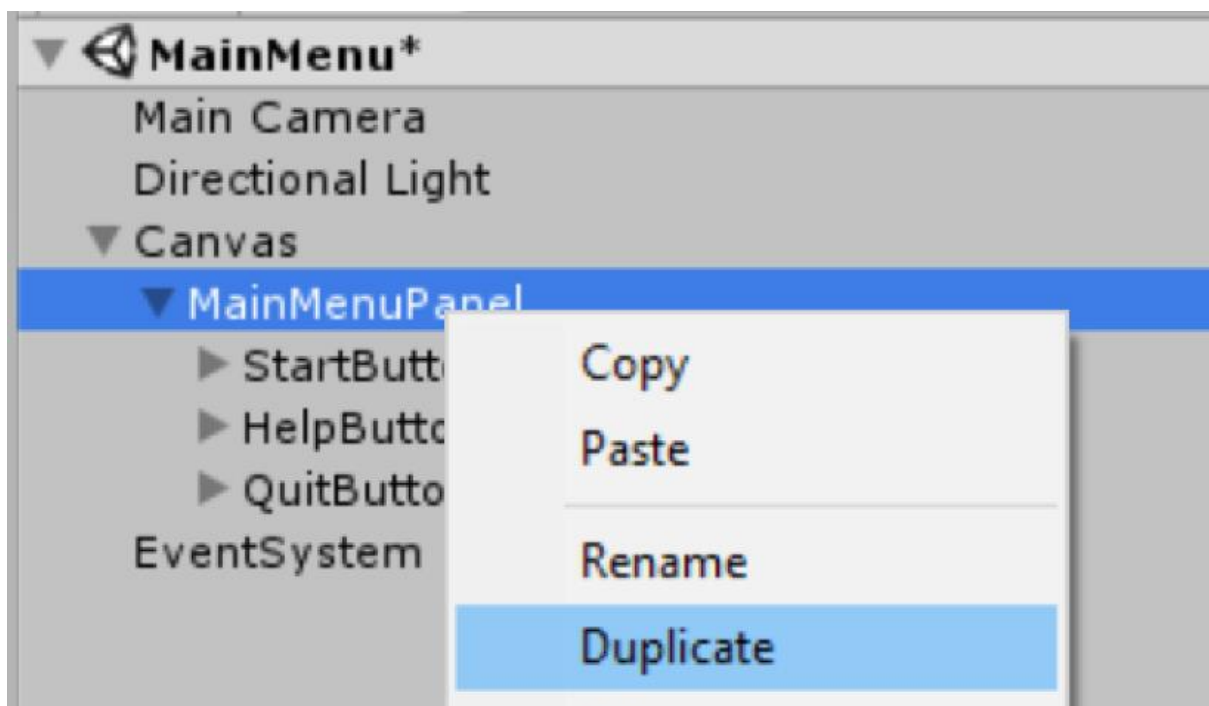


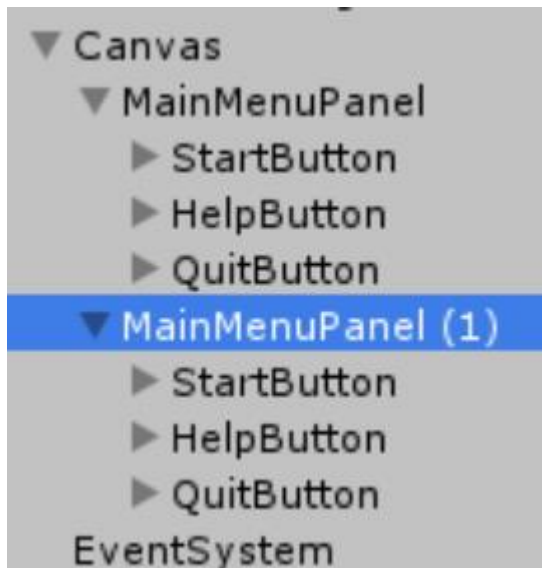
这应该在主菜单场景和1级场景。



现在, 这是工作, 我们将看看帮助菜单项。

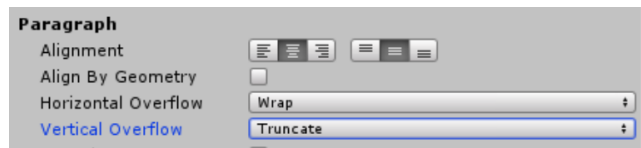
在层次结构面板中, 单击曼梅努布并复制它。



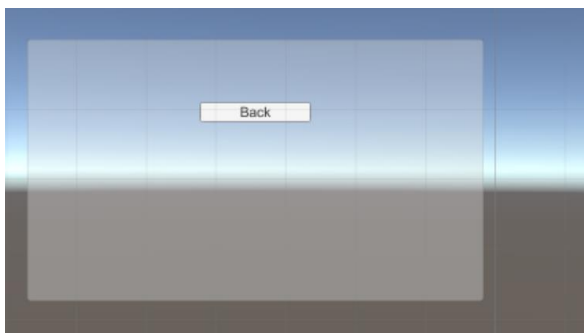


现在, 随着第二个面板删除启动按钮和帮助按钮.将 "面板" 重命名为 "帮助" 面板。重命名时按钮自背键.

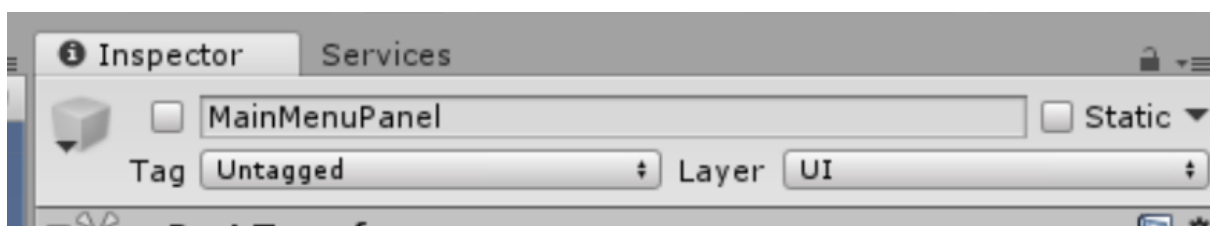
如果从按钮引发 "后退" 一词, 请进入检查器上的文本设置并更改溢出设置。



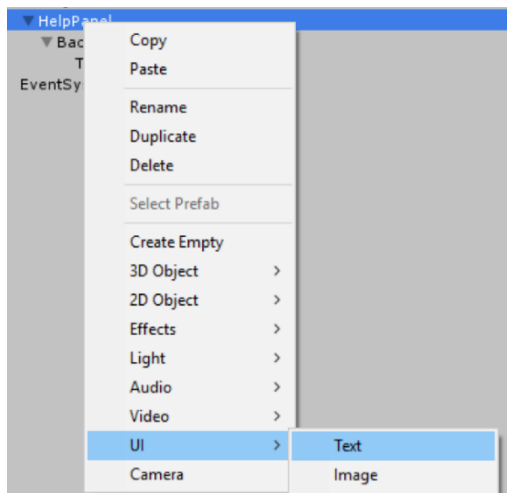
您应该看到以下内容。



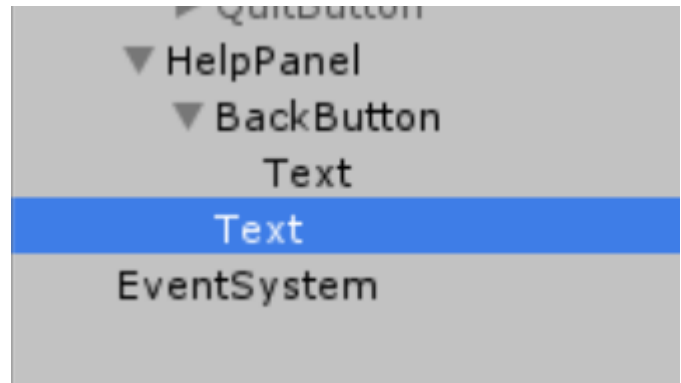
若要隐藏主菜单面板, 关闭旁边的活动它名字。



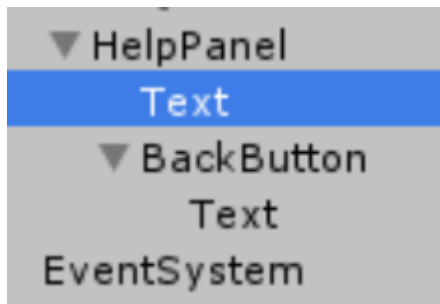
现在, 回到后面的按钮, 接下来我们要添加一些文本。在层次结构面板中, 单击 "帮助" 按钮, 然后右键单击, 转到 ui > 文本



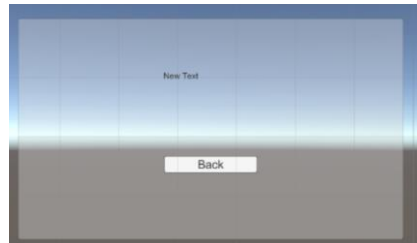
您应该看到以下内容



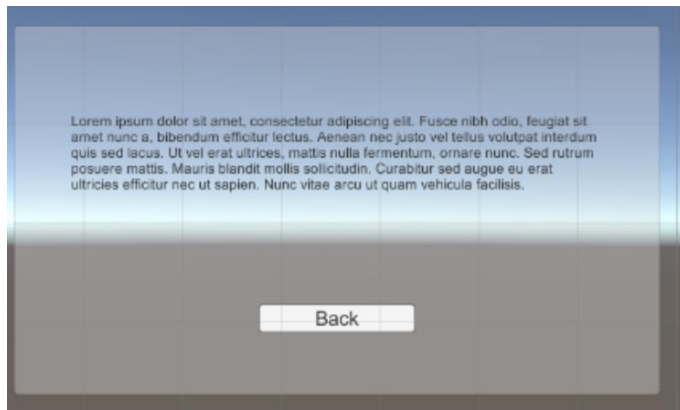
将文本拖到按钮上方。



这将纠正对齐方式, 并将场景中的文本居中。

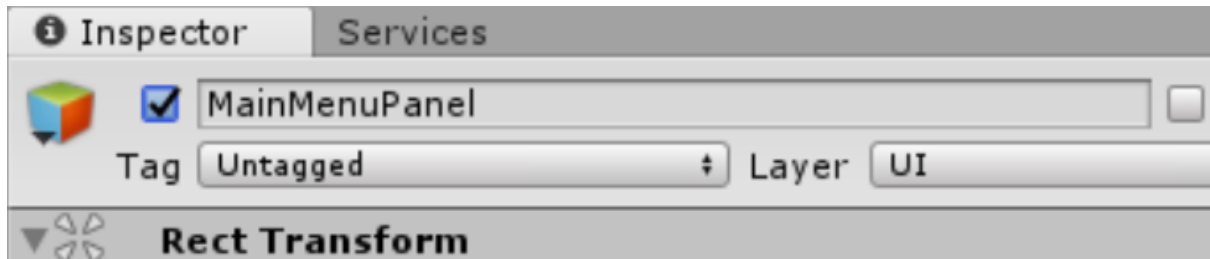
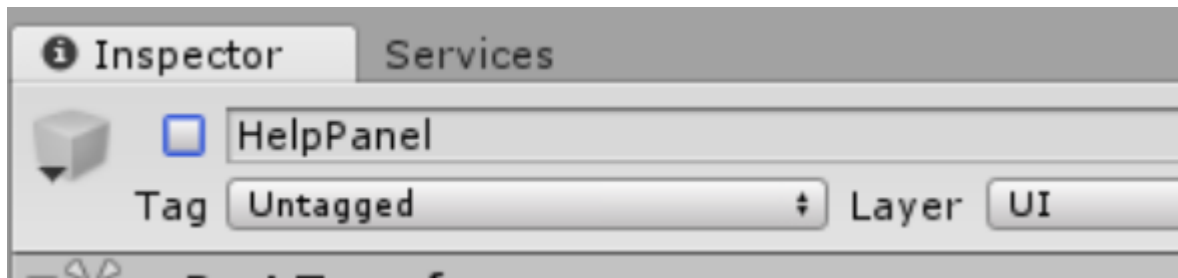


进入文本设置并添加一些 **lorem ipsum** 文本。你最终应该会得到这样的结果:



现在内容已经处理好了, 我们会让按钮工作。这将通过交换面板的活动状态在统一范围内完成。

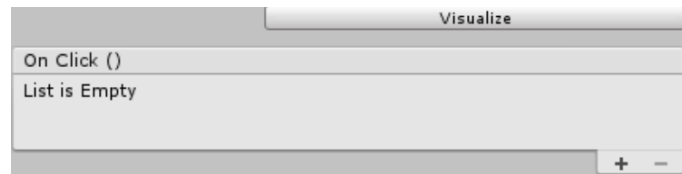
在帮助面板上, 使其处于非活动状态, 并使主菜单面板积极。



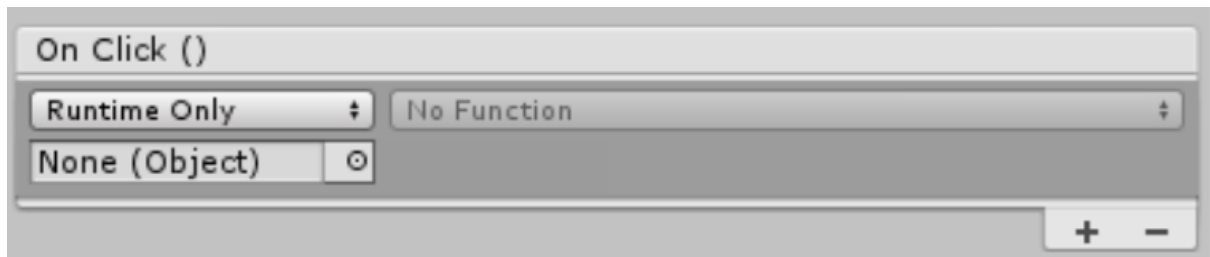
一旦这样做了一个主菜单面板, 点击 "帮助" 按钮



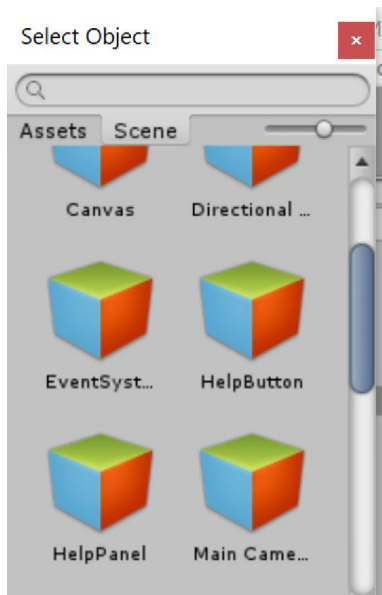
在检查器面板中, 我们有 "点击" 区域。



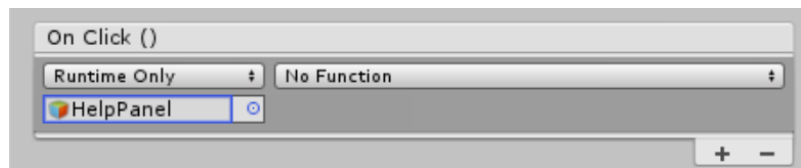
单击 + 符号。



单击对象旁边的目标图标, 然后选择帮助面板



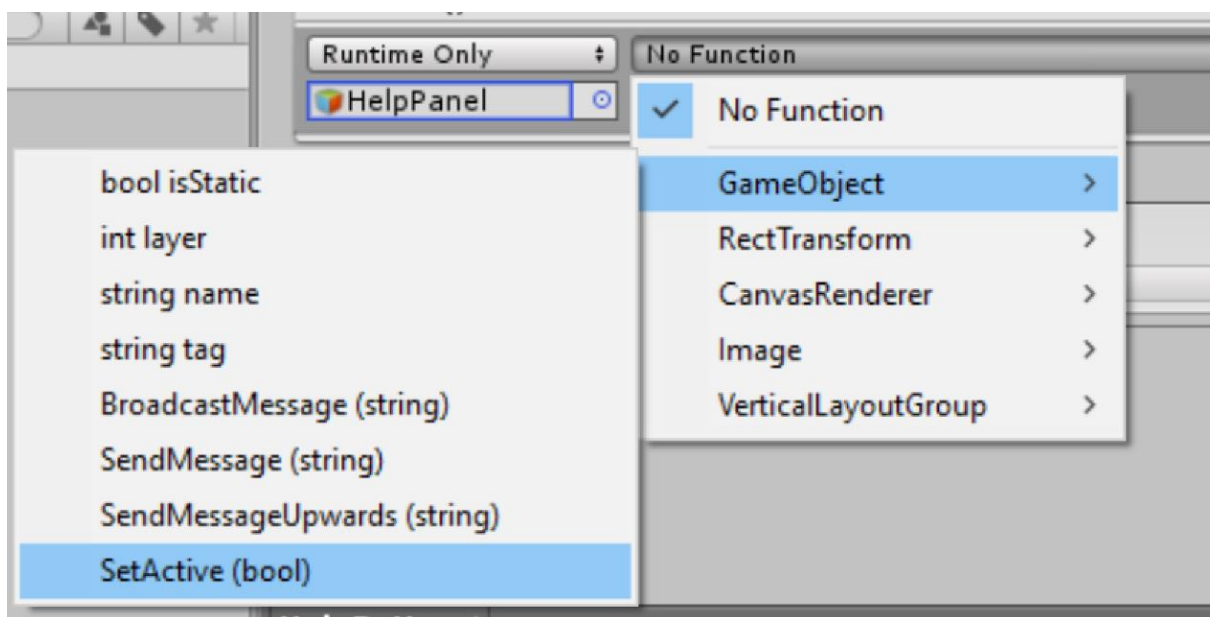
这将在检查器面板中提供以下视图。



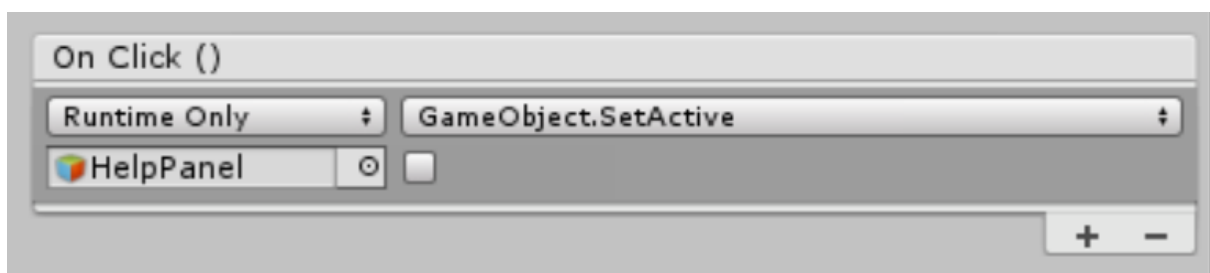
从这里, 我们将使用无函数下拉列表, 并更改setactive选项。

我们要找的是当我们点击帮助按钮时, 我们希望帮助面板以成为活动状态, 即显示给最终用户和mainmenupanel以变为非活动状态。

为此, 添加setactive(bool) 设置。

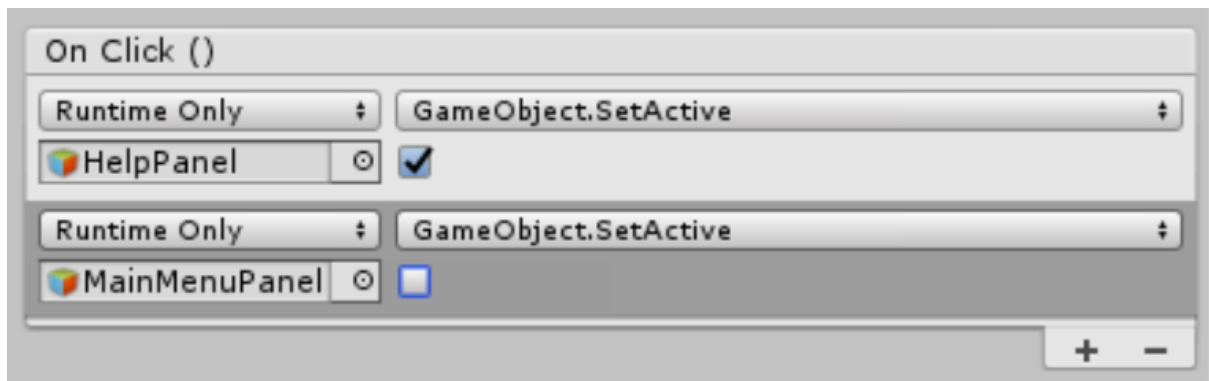


这提供了以下内容:



如果复选框未被勾选, 则表示为 false, 即不setactive.在这种情况下, 我们要添加一个刻度, 就像当我们单击帮助按钮时, 我们希望看到帮助面板出现。

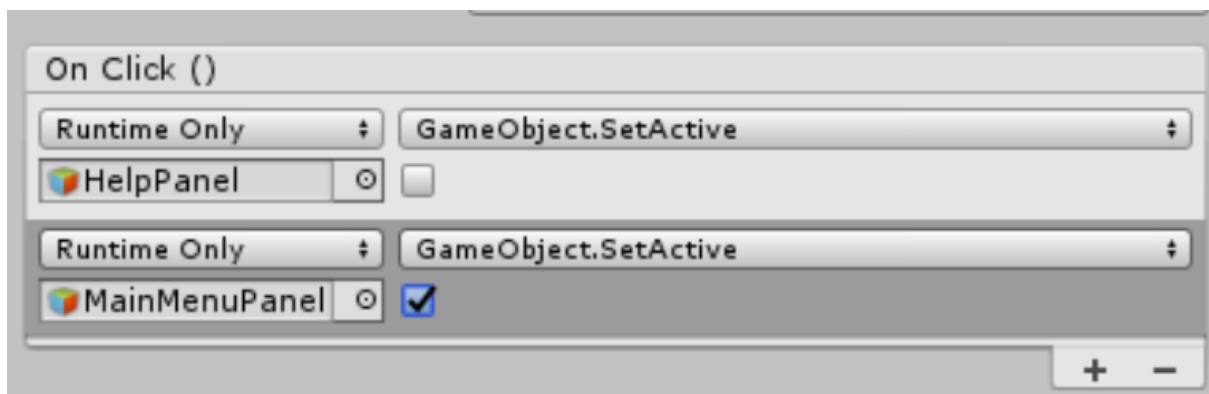
完成此操作后, 添加主菜单面板到单击时的部分, 并将其设置为非活动状态。您应该最终以点击式部分结束, 如下所示:



运行该程序并对其进行测试。

这将为您提供帮助系统的工作菜单。现在, 在后面的按钮上重复相同的步骤。帮助面板.

您的 "后退" 按钮最终应具有以下功能:

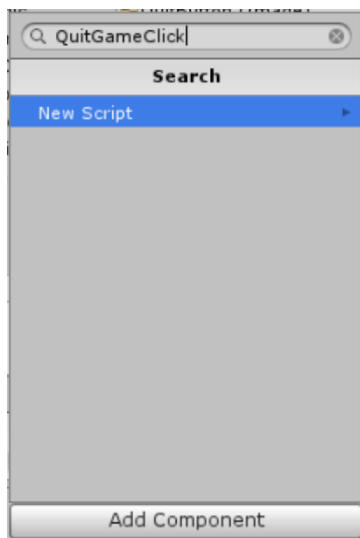


这是相反的帮助按钮.运行并测试它。

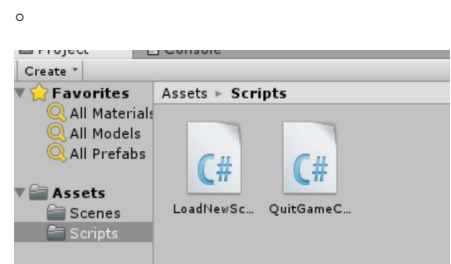
最后我们需要将退出功能添加到菜单系统。

单击层次结构中的 "退出" 按钮。

然后在 "添加组件" 部分中, 对退出游戏点击;创建新脚本并转到代码。



完成此操作后, 将新脚本拖到脚本文件夹中, 然后打开要编码的脚本

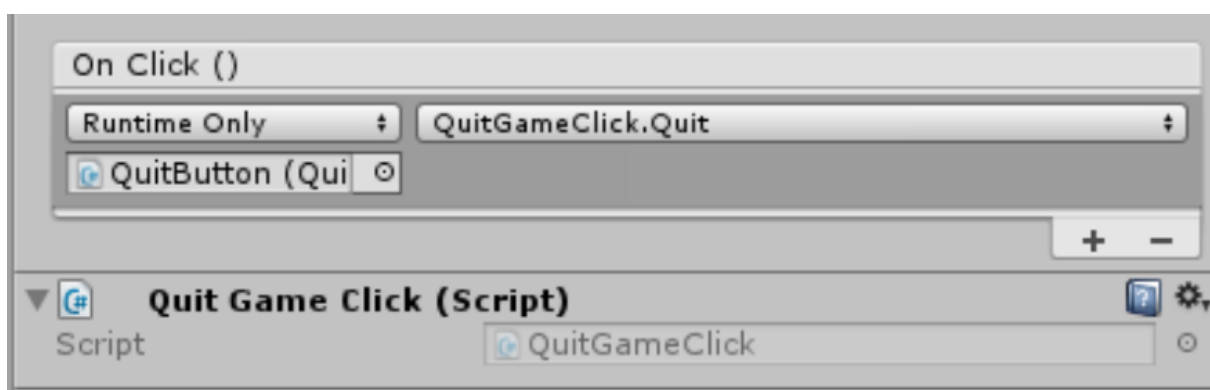


```
public class QuitGameClick : MonoBehaviour {
    public void Quit()
    {
        #if UNITY_EDITOR
            UnityEditor.EditorApplication.isPlaying = false;
        #else
            Application.Quit ();
        #endif
    }
}
```

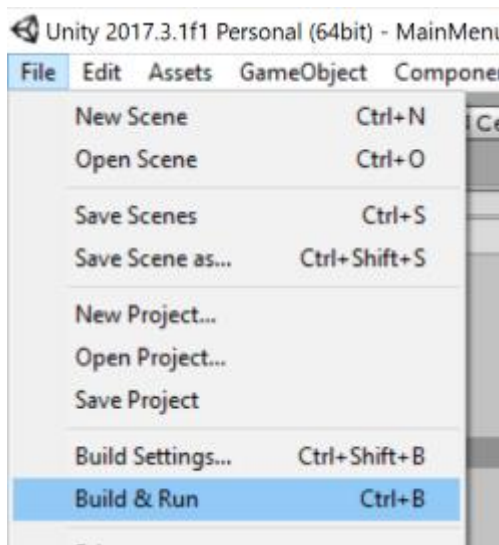
请注意, 这具有退出游戏的效果, 一旦它在统一编辑器外部使用, 并退出统一编辑器。

保存此脚本后, 我们需要将其分配给按钮 onclick 事件, 就像我们对场景管理器所做的那样。

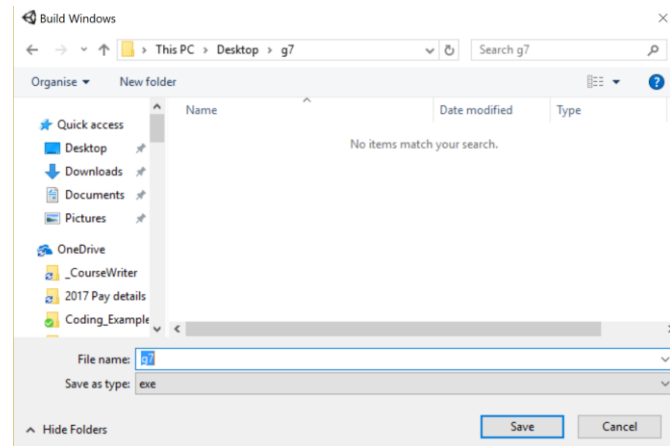
退出按钮的检查器应具有以下内容:



要作为实际的游戏转到菜单并执行文件 > "生成和运行"



给 exe 一个名称, 并保存到您可以找到的位置, 在这种情况下, 我将使用桌面。



一旦它已经建成, 应该有一个配置窗口弹出打开, 选择默认值, 然后点击播放!来测试它。

