



虚幻引擎

解锁最终区域的钥匙： 在虚幻引擎中使用条件语句

学生指南

活动 4

解锁最终区域的钥匙：在虚幻引擎中使用条件语句

概述

虚幻引擎是一个沉浸式的 3D 游戏引擎，人们用它打造了许多风靡世界的游戏。虽然有些游戏需要专业团队才能制造出最终产品，但你即便没有经验也可以开始尝试使用虚幻引擎。你不只是关注计算机编程的基本概念，而是直接创建一个游戏。这一系列的活动旨在引导你完成一个 3D 游戏的制作过程，同时突出重要的计算概念。我们希望打造游戏的承诺和热情能帮助你更好地了解计算机编程基本概念的背景，同时也能更有学习动力。

整个项目由五（5）项“编程一小时”活动组成，项目结合了制作 3D 游戏所需的所有概念和说明，你可以运行这个游戏，还可以与朋友分享。这些活动和包含的项目文件既可以从头到尾按顺序完成，也可以选择任何一个活动单独完成。每个活动都有激动人心的新挑战和新发现，展示了使用虚幻引擎开发游戏的力量。

关于本活动

我们已经制作了跳跃难关、移动平台、可拾取物和强化效果，现在该介绍最终挑战了。完成游戏的秘密就隐藏在高空要塞里一扇紧锁的大门之后。

你需要使用蓝图可视化脚本系统，创建钥匙并将它隐藏在世界场景中。有了之前创建的蓝图、浮岛、可拾取的金币、跳跃强化效果和一把钥匙，你就能测试自己的游戏设计能力了。当玩家找到钥匙、解锁大门，并抵达最终的目标点后，他们就能看到自己的得分。玩家可以为最高分展开竞争！

在本次活动结束后，你就能从头到尾完整地体验自己的游戏了。

开始入门

如果你尚未下载**虚幻引擎**和**“编程一小时”项目**，请查看[《入门指南》](#)。如果你已经完成了下载，那么打开项目就能开始学习了！

在本课程中，你将会学习使用**条件语句**。条件为寻找解锁大门的钥匙。当你打开大门进入后，就能完成游戏。祝你好运！

编程概念

你将深度探索虚幻引擎的蓝图可视化脚本系统。在本活动中，你将创建一个用于解锁最终关卡的钥匙。为了实现这个目标，你将使用布尔变量和条件语句。条件语句是一种编程概念，仅在满足特定条件时执行代码逻辑。

在构建蓝图资产时，你将在解决复杂问题的过程中，使用代码将其分解成一系列简单的步骤。你可以将这些步骤按逻辑顺序连接，以实现自己的目标。编程会让你学会宝贵的技能，使你能够将繁杂的问题分解成一系列简单的步骤。掌握这个过程后，你的问题解决能力会有所提升，还能够创建出你能想象的任何东西！

准备活动

如果你已经完成了活动 1、2 和 3，就可以选择跳过这一部分，使用已经做好的内容进行构建。

如果你直接从活动 3 开始，或者希望从头开始，请参照这些指示以开启新的项目。

因为我们会从活动 3 开始，你需要加载完成版的活动 1 和 2，以便从头开始游戏。各关卡的完成版示例包含在示例项目中，参照下列步骤加载关卡 1 和 2 的内容。

已完成的关卡位于“关卡”面板中，你可以找到**窗口->关卡**并显示它。点击它，并将窗口拖动到**世界大纲视图**旁边，方便我们之后使用。

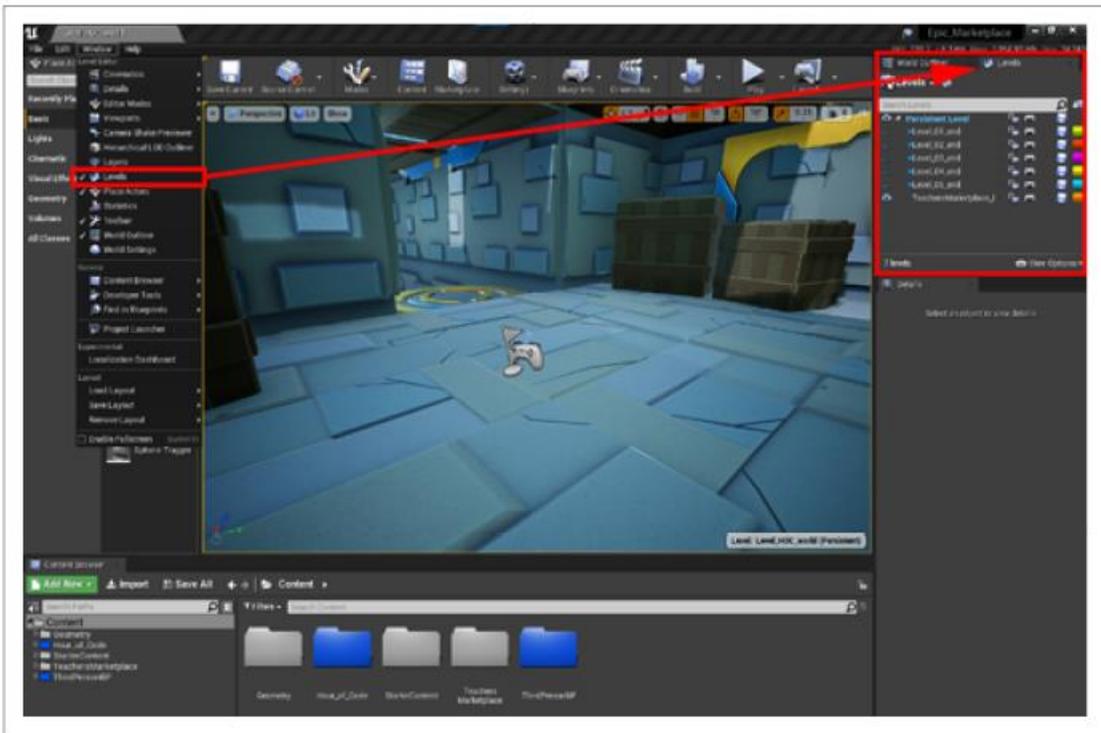


图 001a - 将“关卡”面板移动到“世界大纲视图”面板旁边。

要加载完成的关卡，找到**关卡**面板，**右键**点击 **Level_01_End**，并选择**修改流送方法->固定加载**。这样你在运行游戏时就会加载示例关卡。对 **Level_02_End** 和 **Level_03_End** 重复这一步骤。你无需加载其他关卡，所以你可以选择暂时勾选**蓝图**选项。

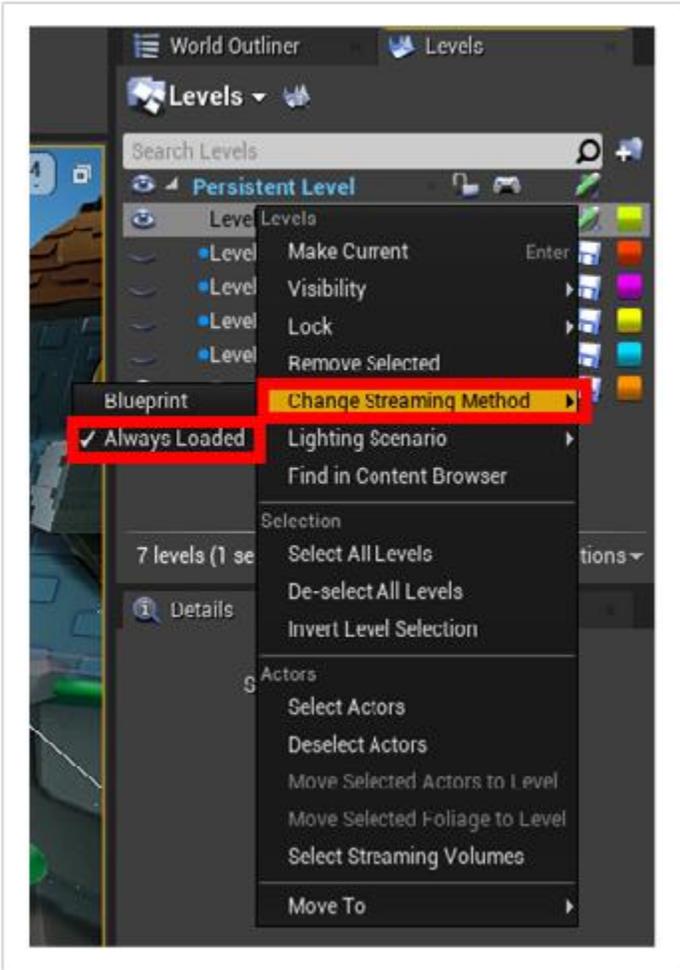


图 001b - 如果你没有创建自己的关卡，则加载关卡 1、2 和 3。

注意：确保**持久关卡**显示为**粗体字**。如果你双击其他关卡，比如双击 **Level_01_end**，该关卡就会激活。这意味着在**视口**中添加 Actor 时，它会被加入到 **Level_01_end** 关卡中。其他关卡依然是可见的，但无法在**视口**中编辑。在这些活动中，我们只会向**持久关卡**中添加资产。**粗体字**标出的关卡名称就是激活的关卡。

错误排查

如果你向**持久关卡**以外的关卡添加了 Actor，只需要选中这个 Actor，然后在**持久关卡**上**点击右键**，选择**将选中的 Actor 移动到关卡中**。

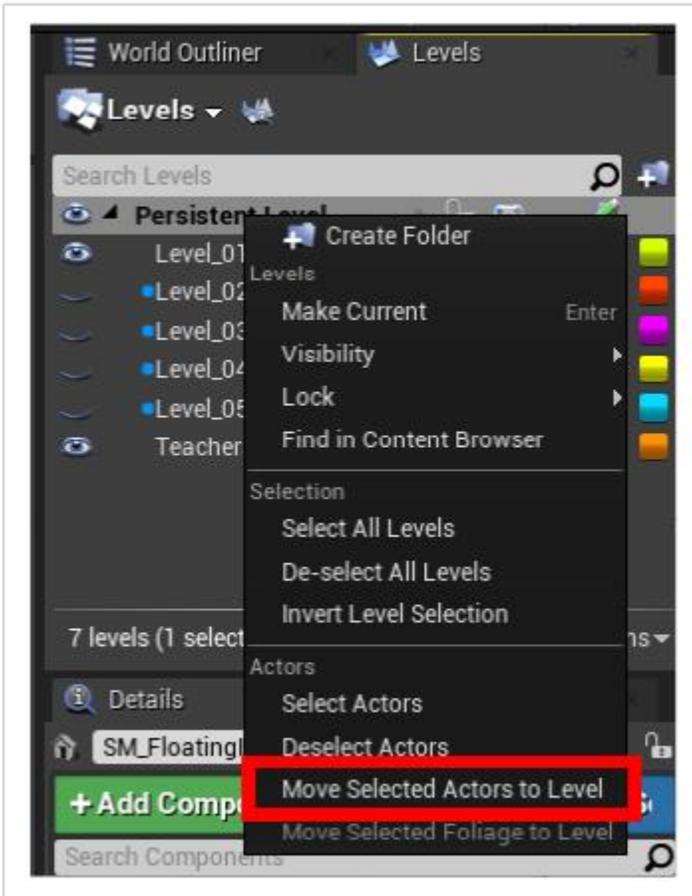


图 001c - 如果不慎将 Actor 放置在了 Level_##关卡里，将 Actor 移动到持久关卡中。

省时窍门

我们这个项目添加了一些实用的工具，先花点时间浏览一下这些窍门吧。

摄像机书签

摄像机书签可以帮助你**在编辑器中迅速改变位置**。要了解它们的原理，先在**视口**中的任意位置点击一下，然后按下键盘顶部的数字键 **1** 或者 **2**。你就会发现，摄像机跳转到了特定的位置。

- 1 = 活动 1 的开始位置
- 2 = 活动 2 的开始位置

这能让你更快地在关卡中移动，而无需手动从一个位置转换到另一个位置。在本次课程中，按钮 1-7 都分配了重要的书签，你也可以按下 **Ctrl+ 键盘顶部任意数字** 来分配自己的摄像机书签。试着使用数字键 8-0 吧。

到你了：使用 **Ctrl+8**，在关卡的某处设置另一个摄像机书签。要检查它能否正常运行，你可以按下 1-7 之间的任意数字，使用其他摄像机书签。现在按下键盘上的 8 键，它应该能带你跳转到自己创建的书签位置。它能够正常运行吗？

从当前位置运行

你知道你可以从当前摄像机所在的位置开始游戏吗？要设置这个功能，只需打开**运行**按钮旁边的下拉菜单，选择**当前摄像机位置**选项即可。这可以为测试关卡时节省很多时间，只是要注意，如果你在虚空的上方开始游戏，你的玩家会掉进虚空中并重生。

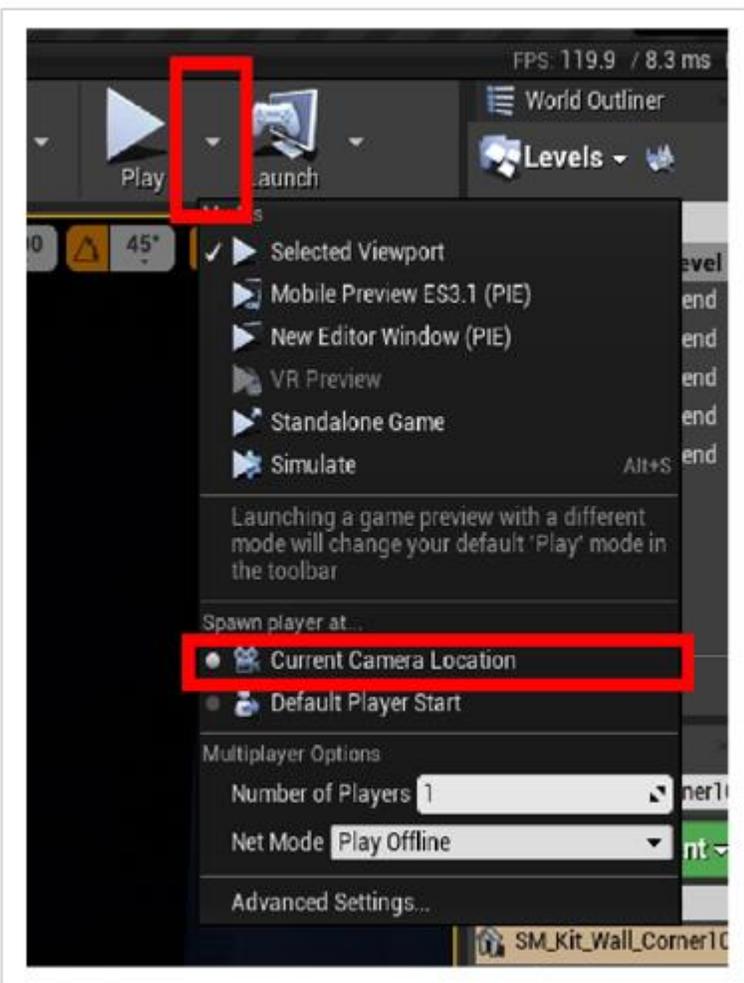


图 001d - 将“运行”按钮设置为从“当前摄像机位置”开始。

本系列课程旨在向学生介绍使用虚幻引擎（业内标准游戏开发工具之一）中的计算机科学概念。

我们开始吧！

如果你运行游戏，并且来到了大型建筑的门前，你会发现需要钥匙才能打开它。

我们来测试一下，直接在门口开始游戏。在**视口**中点击一下，按下数字 **5** 键，移动到摄像机的 5 号摄像机书签。确保“运行”按钮的选项设置为**当前摄像机位置**，然后点击“运行”。

走到门口，系统会显示文本“需要钥匙”，并且会阻止你打开大门。

我们先停止游戏，这样才能在编辑器里施展魔法！

条件语句

条件语句会检查条件是否为**真**或者为**假**，它们通常也被称作**如果/就语句**，或者**如果/就/否则语句**。简而言之，它们的原理是这样的。

如果已经捡起钥匙 = 真

如果还未捡起钥匙 = 假

在蓝图中，它们也被称为“**分支**”，分支节点看起来是这样的。

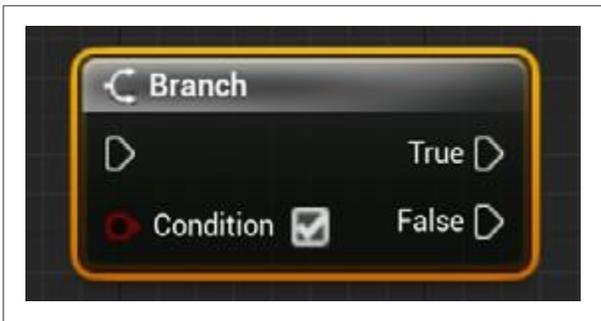
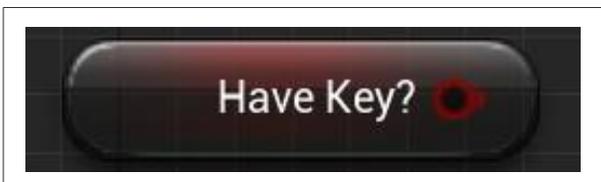


图 001

什么是布尔？

在编程中，条件可以被存储为**布尔**，缩写为 **bool**。数据可以被存储为两种状态，要么是**开**（真），要么是**关**（假）。我们之后会在第 14 步中使用存储的数据。**布尔**是红色的，所以在蓝图中可以一眼认出。



到底什么是蓝图？

我们在关卡中所做的主要就是构建“钥匙”**蓝图**。你可以将**蓝图**视作一个行李箱，其中包含了各种各样的物体和代码，而代码可以与游戏的多个部分交流。

例如，你的行李箱中也许放着手机，手机可以和 GPS 卫星通信。如果航空公司不慎运丢了你的行李箱，你就能使用 GPS 的代码与手机取得通信，帮助你找回行李箱。最终你发现，行李箱先你一步运达了目的地，就在你背后的行李带上。

如果你需要再了解一下**蓝图**，可以直接回顾之前的“编程一小时”活动。如果你学习起来感觉有困难，可以在学习过程中与同学们多多交流，排忧解难。制作游戏需要团队合作，不应该独自完成。如果你的同学请求你的帮助，请耐心地协助他们。这也能帮助你巩固理解，还可以确保他们在未来能够投桃报李。

接下来的这一部分共有 19 个步骤，请先思考我们尝试实现的目标，并紧密跟随每一步的知识，以理解如何构建我们的解决方案。

要了解如何连接到**布尔**，在**内容浏览器**中找到**内容->Hour_of_Code->Blueprints** 文件夹双击打开 **BP_Door_Locked**。

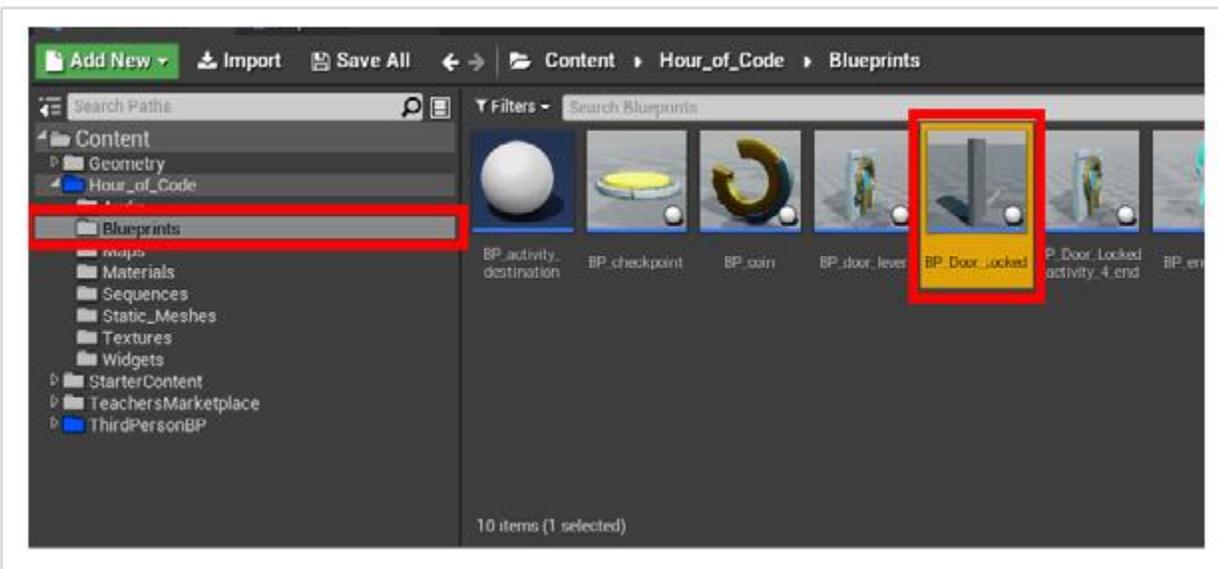


图 026 - “BP_Door_Locked”蓝图

点击**事件图表**选项卡。你可以看到一个标为**活动 4**的黄色评论框区域。目前，**Target**还没有连接到分支。在蓝图中，你可以点击并拖动它，在构建代码时连接不同的节点。

快试试吧！

在黄色区域中，只需要从“Have Key？”布尔引脚中拖出一条线，连接到**分支的 Condition**引脚里。

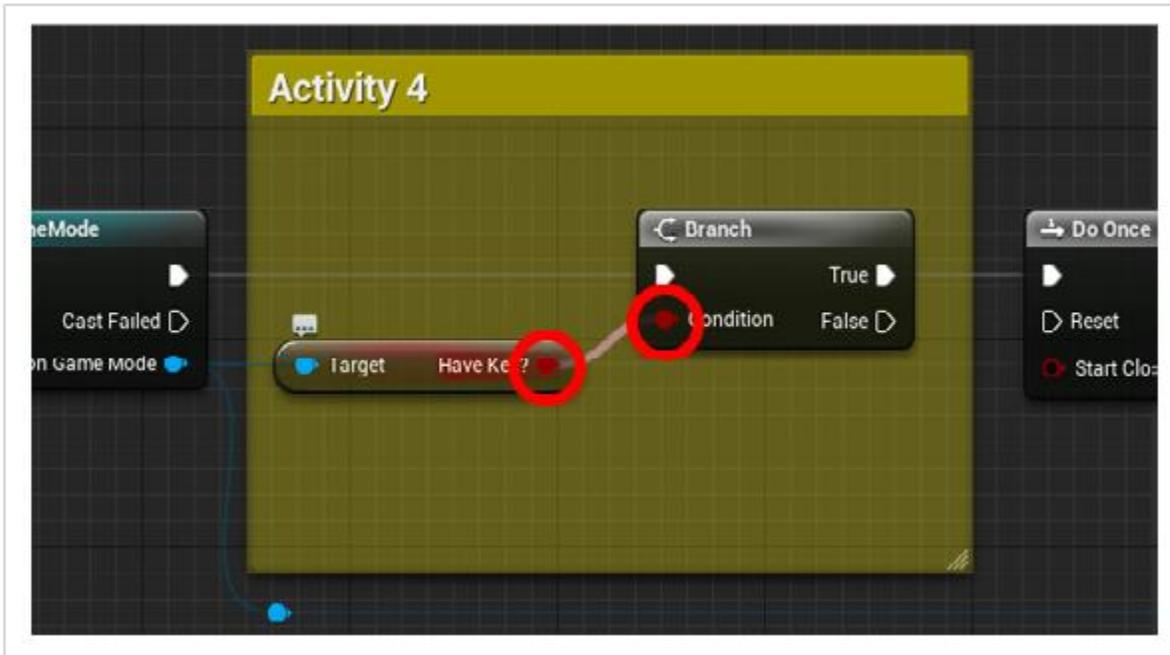


图 025 - 将“Have Key?”布尔连接到分支 Condition。

现在蓝图就会检查条件“玩家是否拥有钥匙？”这个条件的结果可能为真或者假。听起来是不是非常简单？

我们现在从头开始，创建一把能够打开大门的钥匙。

1.要从头构建蓝图 Actor，进入**内容->Hour_of_Code->Blueprints** 文件夹，在空白处**点击右键**，选择**蓝图类**。

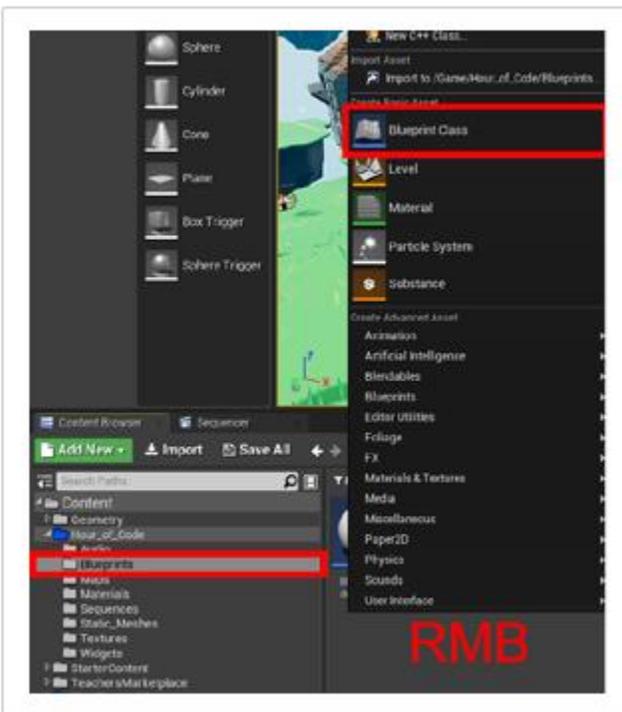


图 002 - 在“蓝图”区域中点击右键，新建一个“蓝图类”。

2. 选择列表顶部的“Actor”类。



图 003 - 选择“Actor”作为蓝图父类。

3. 将这个新蓝图命名为 **BP_key**。

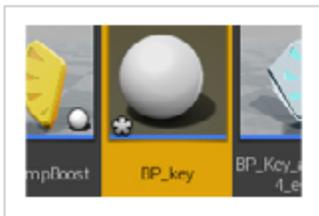


图 004 - 全新的钥匙蓝图。

4. 双击打开 **BP_key** 蓝图。如果它没有自动加载，就点击**视口**选项卡。接下来，我们会组装玩家能够看见和互动的部件。

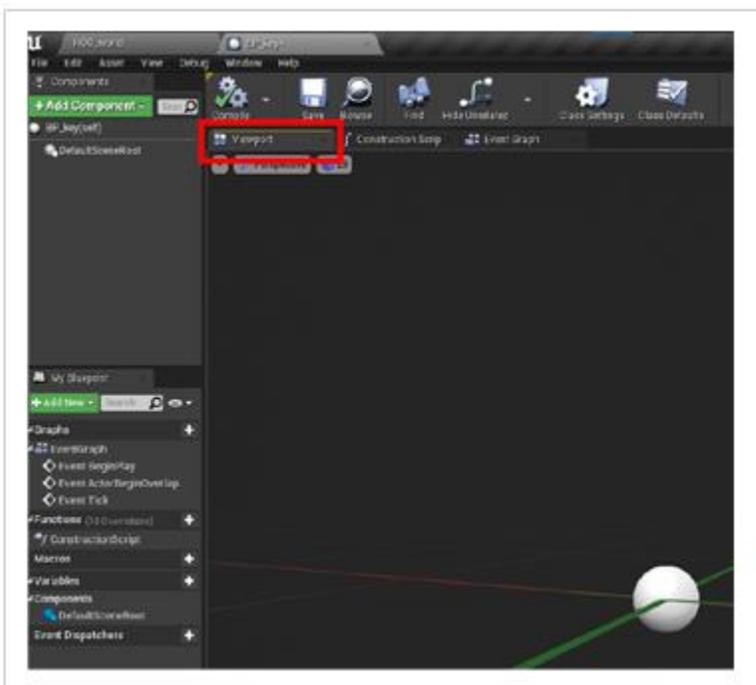


图 005 - 为钥匙蓝图选择视口。

5. 我们需要在视口中添加一些**组件**，这样玩家才知道他们能够捡起什么，**角色**也才能与之互动。首先我们要添加**球体碰撞组件**。球体碰撞组件会定义钥匙的**碰撞区域**。当玩家进入碰撞区域后，它就会激活碰撞侦测。在界面左上角点击绿色的**添加组件**按钮，搜索“**sphere collision**”，然后在下拉菜单中点击 **Sphere Collision** 选项。一个线框的球体就会出现在视口中。

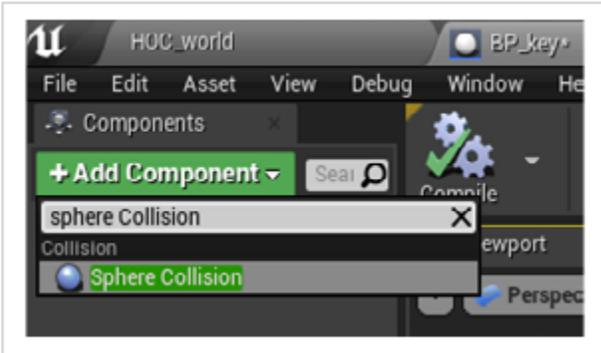


图 006 - 选择球体碰撞。

6. 然后点击组件选项卡中的 **DefaultSceneRoot**，这能确保我们后续添加的组件不会成为 **Sphere Collision** 的子项。



图 006b - 在添加组件时点击 DefaultSceneRoot。

再次点击**添加组件**，搜索**静态网格体组件**。注意，要选择列表底部的**静态网格体组件**。



图 007 - 选择列表底部的“静态网格体组件”。

7.现在，在“组件”面板中选中“静态网格图组件”。在屏幕右侧的“细节”面板中找到“静态网格体”部分。点击显示为“无”的下拉菜单，搜索“key”，然后选择“SM_key”。如果你愿意的话，也可以移动钥匙，将它放置在球体的正中。

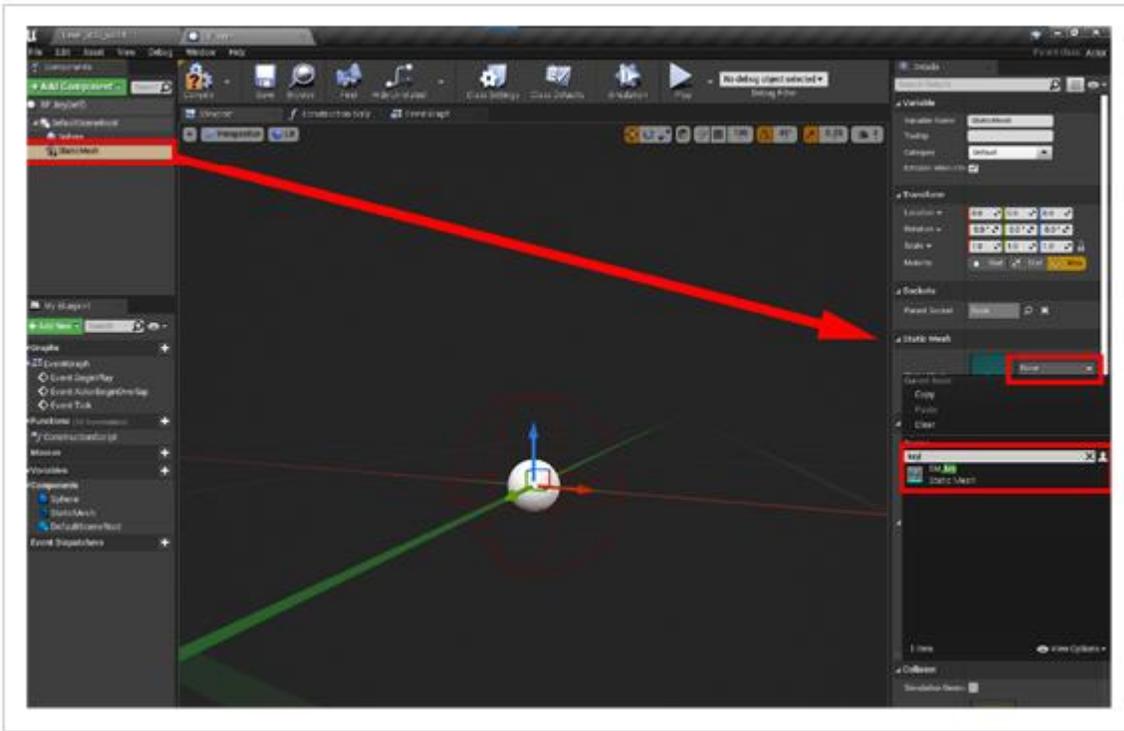


图 008 - 选择钥匙作为静态网格体。

8.现在，我们需要稍作整理。你应该遵守最佳的做法，这样在未来才能有所裨益。对资产命名能够帮助我们追踪它们。在内容面板中选中一项，按下 **F2** 来重命名。将 **Sphere** 重命名为 **Sphere_coll**，因为这是碰撞球体。然后将 **Static Mesh** 重命名为 **SM_key**，因为它就是一把钥匙。静态网格体 (SM) 是一个不会改变形状的三维物体。

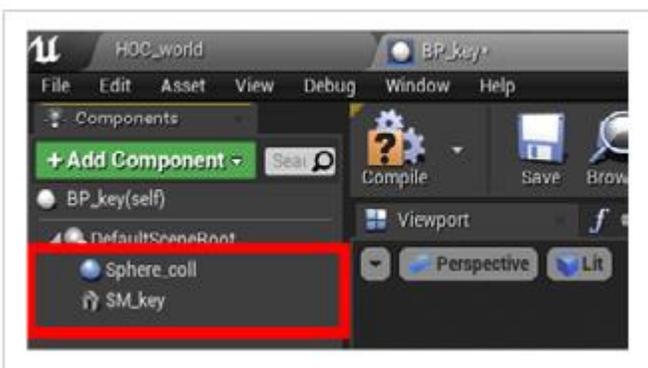


图 009 - 创建碰撞球体和钥匙。

9. 点击界面顶部的“保存”按钮，保存我们的作品。



图 010

10. 点击**事件图表**选项卡，我们会在这里编写代码，让蓝图能够运行。你会在图表中看到 3 个节点。全部选中它们，然后按下键盘上的 **Delete** 键。我们不需要它们。

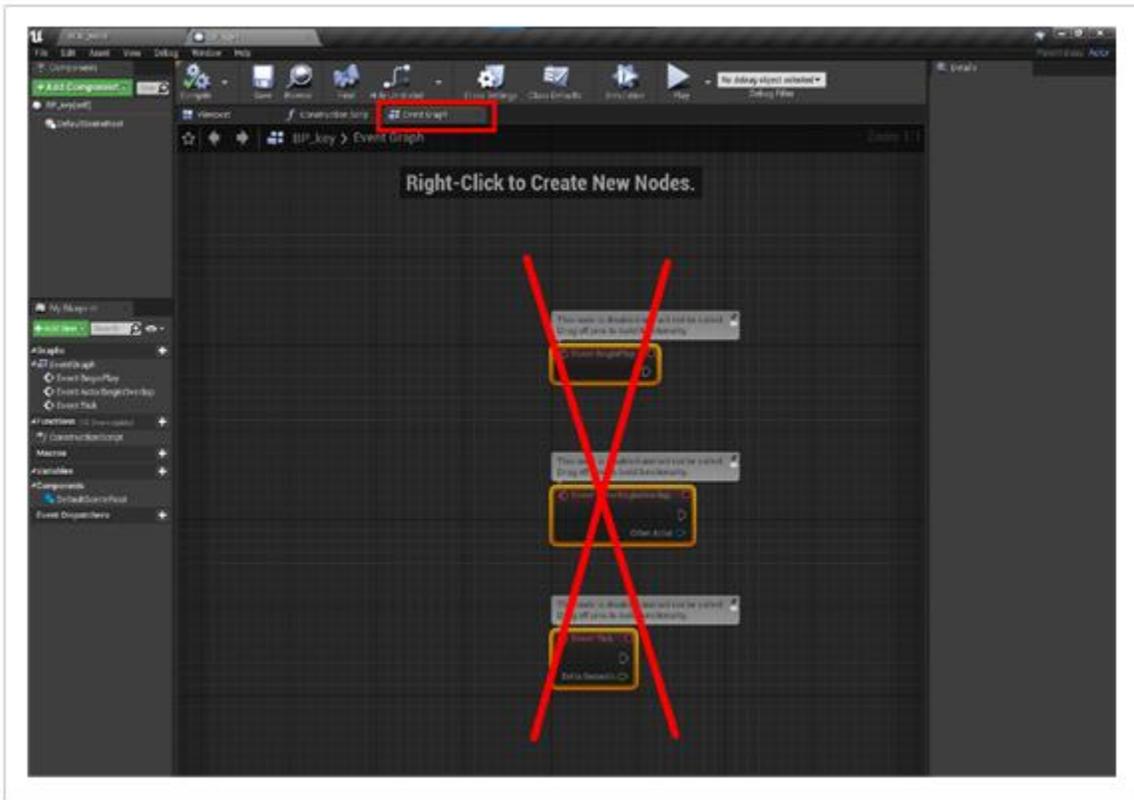


图 008 - 选择钥匙作为静态网格体。

11. 在组件选项卡中选中 **Sphere_coll**，滚动到**细节**面板的底部。在**事件**部分中点击第二个绿色按钮**组件开始重叠时**。这就会创建一个当**角色**触碰到 **Sphere_coll** 时触发的**事件**节点。（换言之，当角色捡起钥匙时就会执行。）

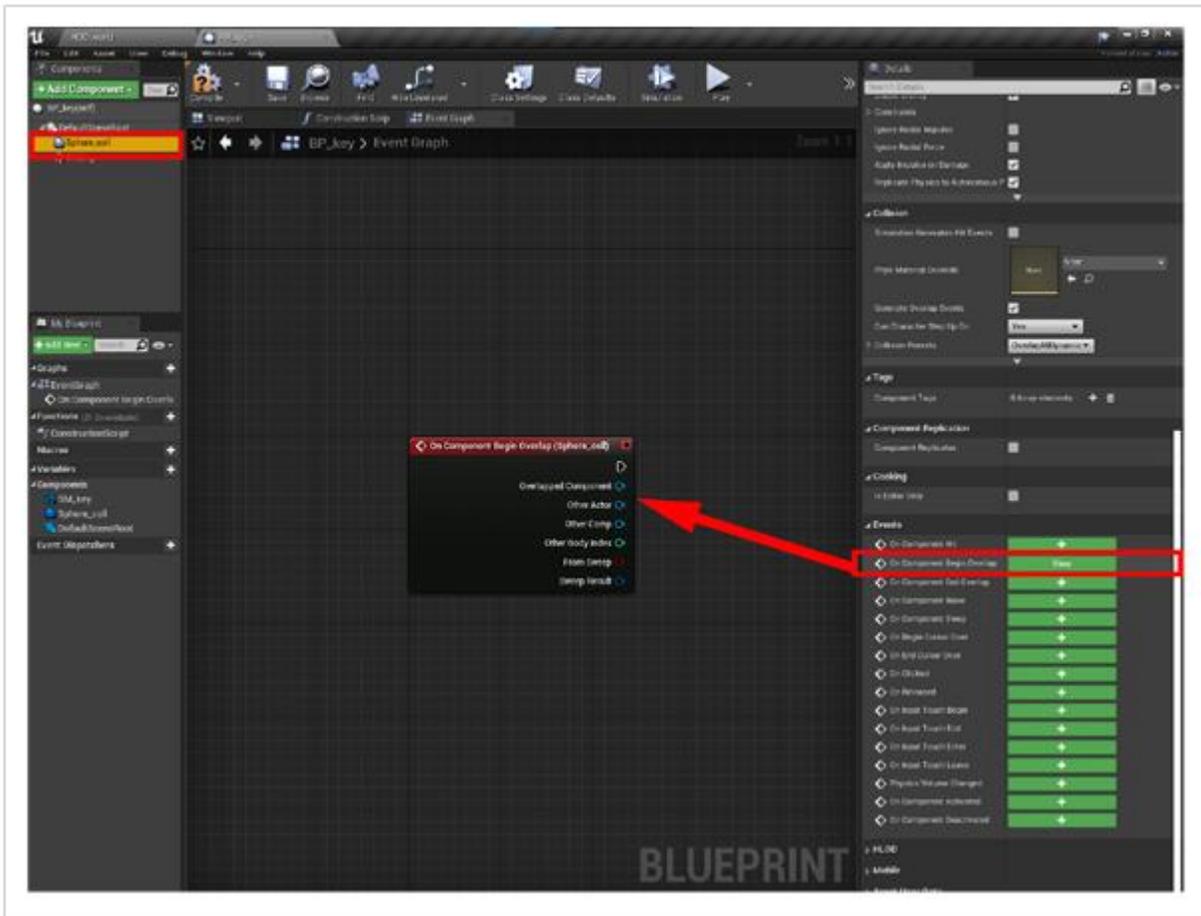


图 012 - 添加用于侦测玩家触碰钥匙的事件。

12. 我们需要让这个蓝图与游戏交流，所以我们需要将它**转换**为游戏。首先我们点击执行引脚，在事件图表的空白背景处拖出一根线，添加**执行行动**。在搜索栏中输入“**类型转换为 thirdpersongamemode**”。选择“类型转换为 **ThirdPersonGameMode**”，这就会创建出一个新的节点。

小贴士：使用鼠标滚轮可以放大或缩小，点击并按住鼠标右键就能拖动视野。如果玩家拥有钥匙，并且“has_key”值为真，玩家就能解锁大门。



图 013 - 与游戏交流的第一个行动。

13.从 **Object** 引脚处拖出一条线，搜索“获取游戏模式”。选择**获取游戏模式**，这会创建出另一个新节点。

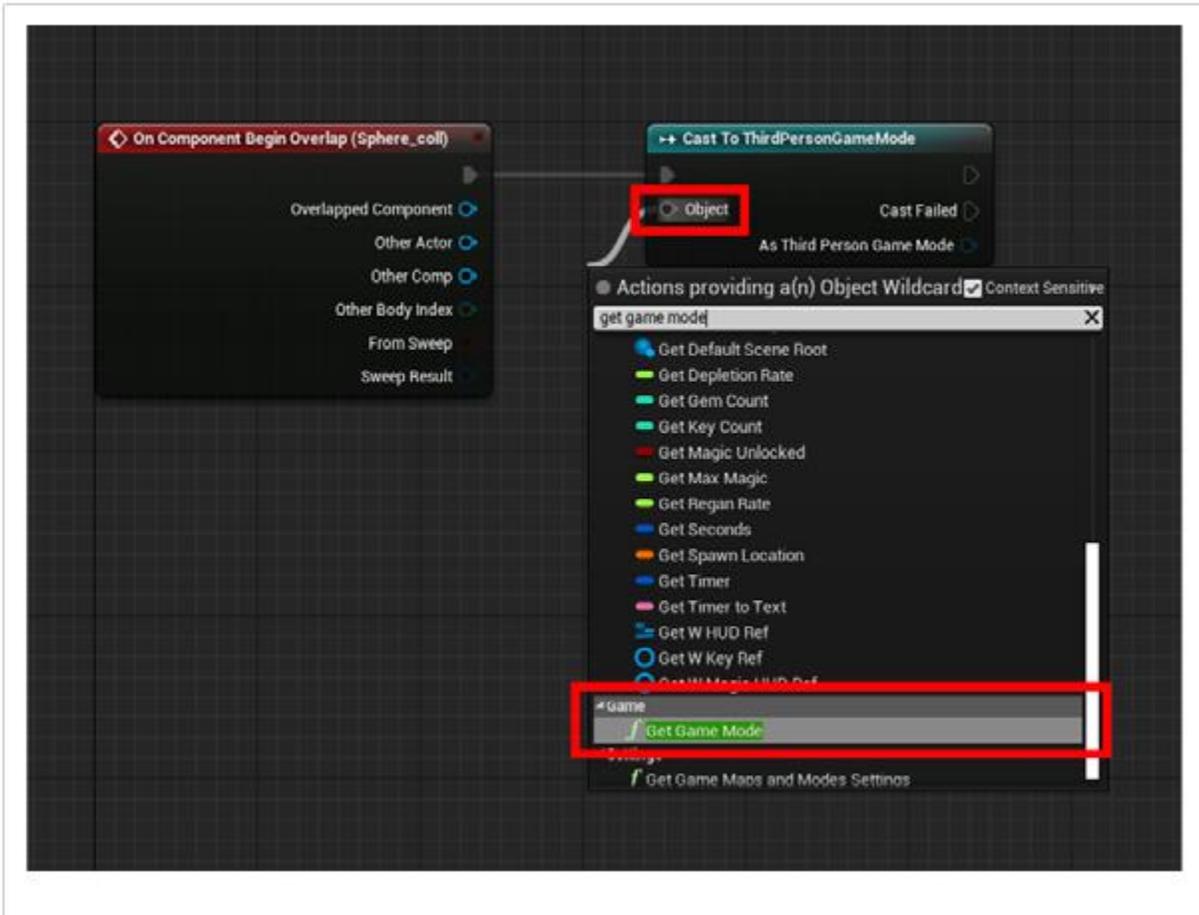


图 014 - 将“Object”属性定义为**获取游戏模式**。

14.现在蓝图已经能够和游戏交流了，而游戏需要知道自己应该干什么。我们会让游戏在角色的物品栏中添加一把钥匙。要实现这个目标，从**类型转换为 Third Person Game Mode** 底部的 **As Third Person Game Mode** 蓝色引脚处拖出一根线，搜索“**设置 have key**”，选择“**设置 Have Key?**”

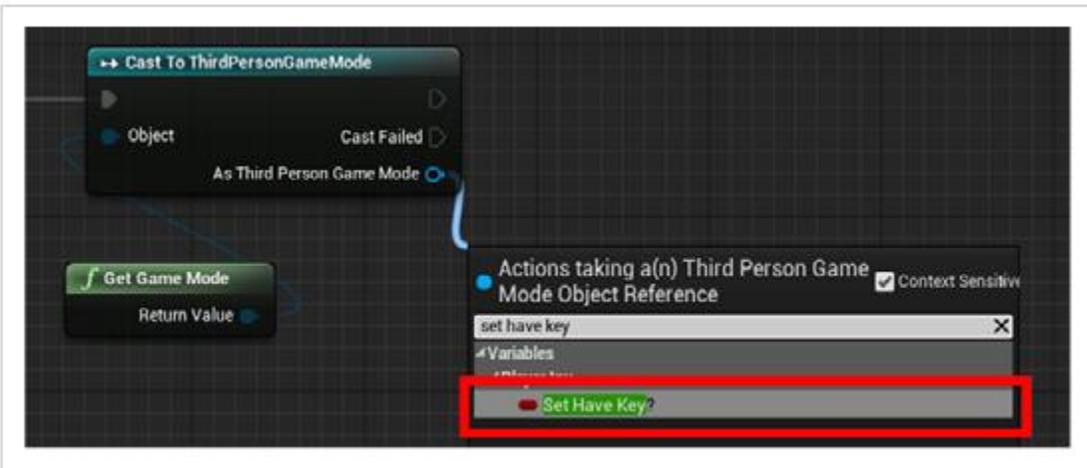


图 015 - 设置“Have Key”变量。

15.将**类型转换为 ThirdPersonGameMode**的执行引脚与“设置”节点连接（见下图），然后将这个**布尔设置为真**。

这里的问题是“玩家是否拥有钥匙？”
我们会通过这个**布尔**节点，将回答设置为是（真）。

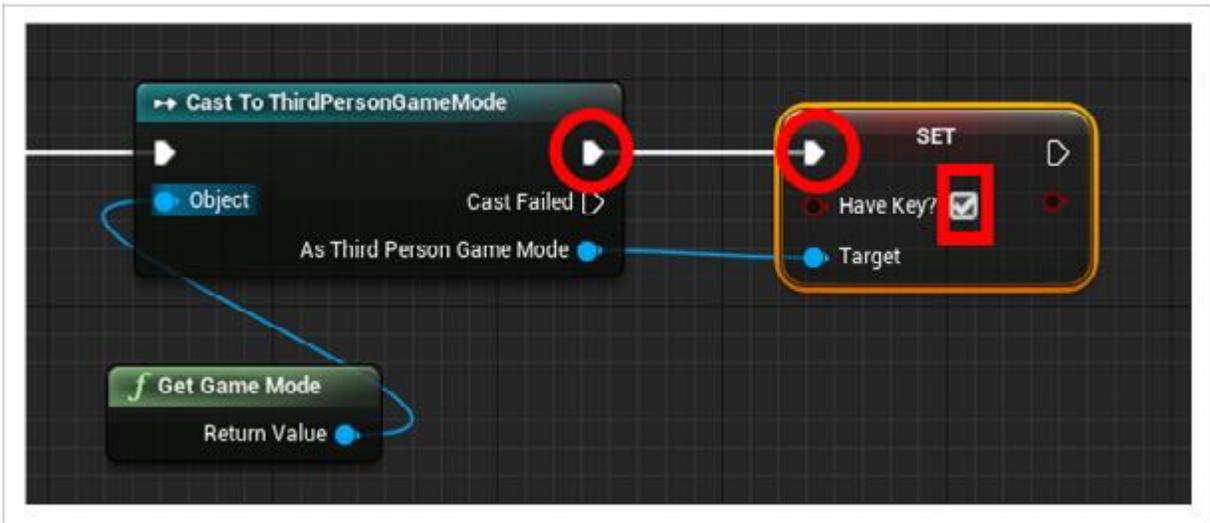


图 016 - 将“Have Key”变量设置为“真”。

16.我们现在要添加一个音效，让玩家知道自己捡起了钥匙。从“设置”的执行引脚处拖出，搜索“播放音效”，选择**播放音效 2D**。

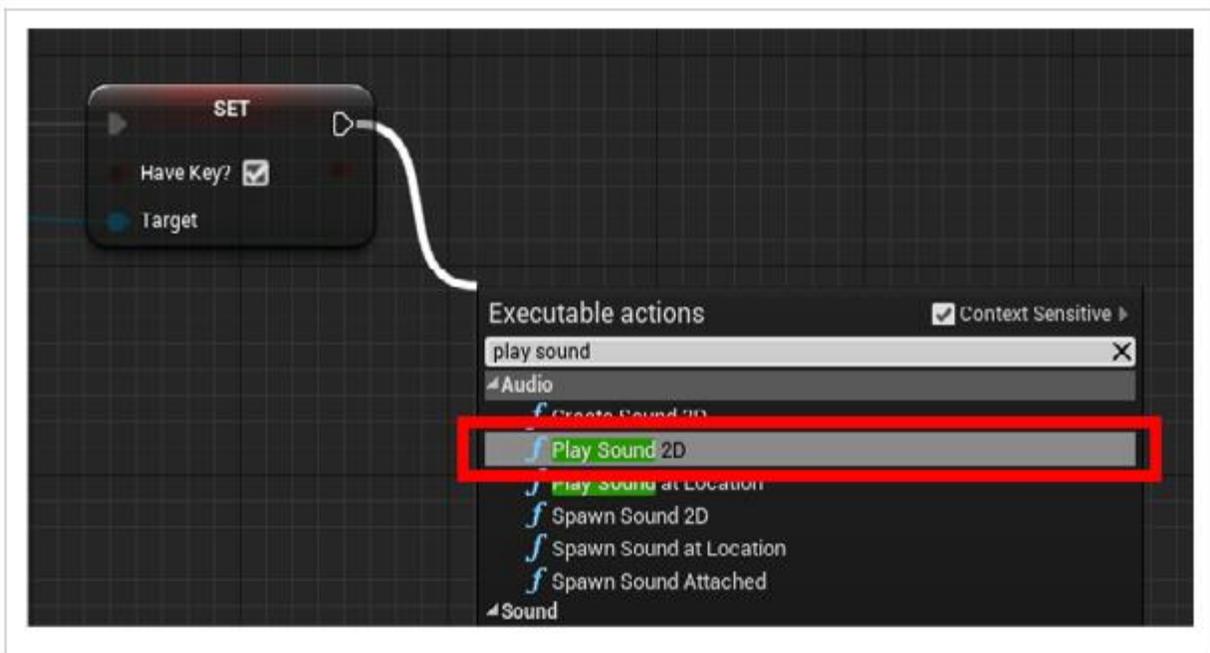


图 017 - 接下来，播放音效来表示钥匙已被捡起。

17.我们需要选择播放的音效。点击**选择资产**下拉菜单，搜索“key”，选择 **SFX_key**。

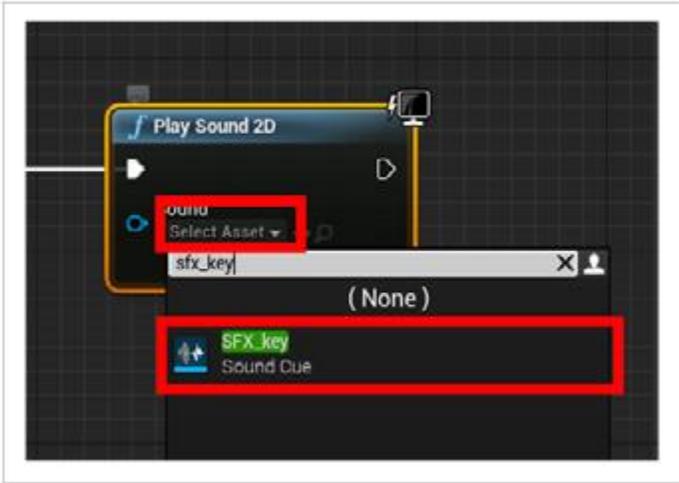


图 018 - 选择捡起钥匙时播放的音效。

18. 我们需要添加最后一个节点，让蓝图销毁自己。毕竟，在角色触碰钥匙并捡起它之后，关卡就不需要用到这个蓝图了。从“播放音效 2D”的执行引脚处拖出一根线，输入**销毁 Actor**，选择**销毁 Actor**。

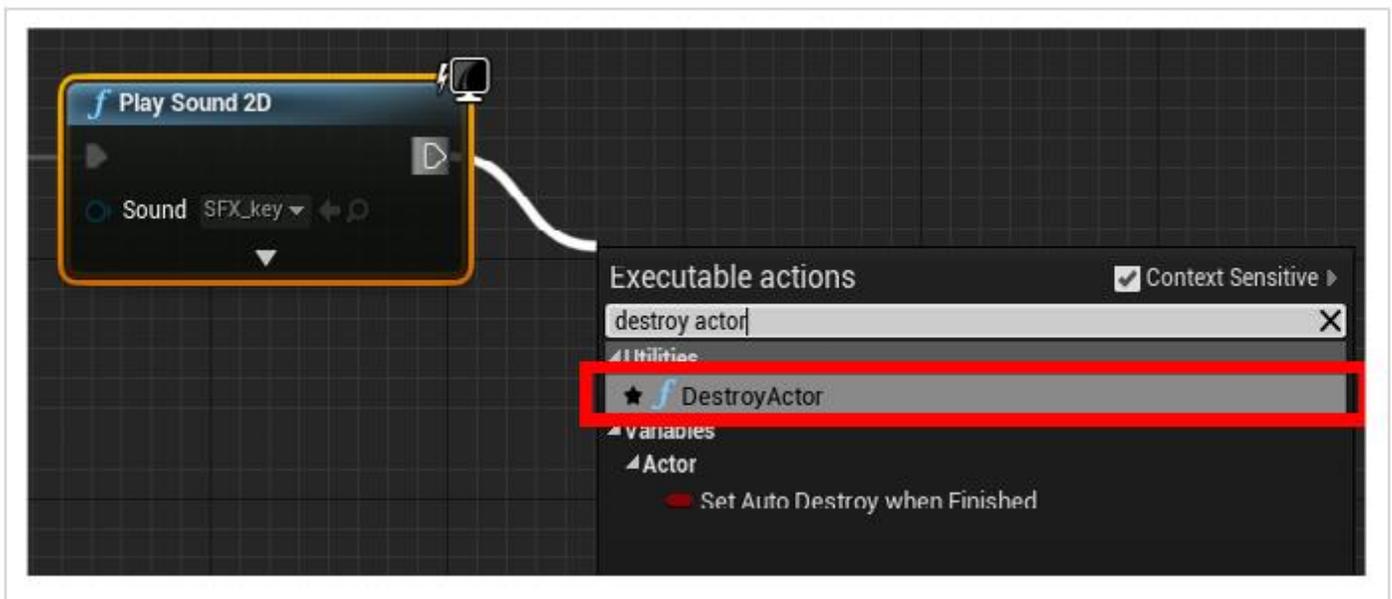


图 019 - 最后，在捡起钥匙后使用“销毁 Actor”移除钥匙。

19. 为了完成这个蓝图，我们需要**编译**代码，并且**保存**我们完成的工作。点击**编译**按钮时，问号就会变成绿色的勾。

查看下方的图片，与你的作品对比一下。

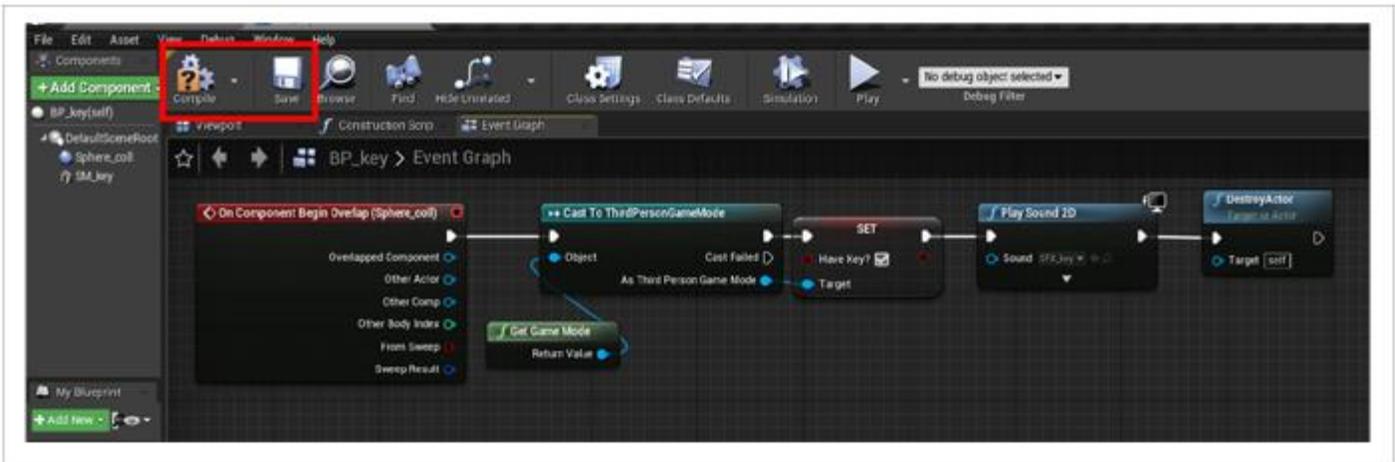


图 020 - 保存并编译蓝图。

现在，该运行蓝图并进行测试了。

确保你已经编译并保存了蓝图，并且点击右上角的 X 来关闭蓝图编辑器。（确保你关闭的是蓝图编辑器的窗口，不要不小心关闭了整个项目。）

现在，你应该返回世界编辑器了。在视口内点击，按下数字键 5，跳转到 5 号摄像机书签。在内容浏览器中，打开“内容”->“Hour_of_Code”->“Blueprints”。将一个 **BP_key** 拖动到关卡中，放在最终岛屿的大门附近。按下 W 激活移动工具，向上拖动蓝色箭头，让钥匙悬浮在地面上方。

按下“运行”按钮，测试一下你的钥匙。

要对钥匙进行全面的测试，不要在开始游戏时立即走到钥匙上。



图 021 - 在游戏中添加钥匙以进行测试。

首先，走到门口。你会看到显示的讯息，玩家“需要钥匙”。这是因为你还没有捡起钥匙。



图 022 - 在没有钥匙的情况下尝试开门时，应该会显示讯息“需要钥匙”。

然后，捡起钥匙。你应该会听到金属般的钥匙音效，然后钥匙就会消失。

注意：一定要确保扬声器能够正常运作。

接下来，走到门口。讯息“需要钥匙”应该**不会**显示在屏幕上。

最后，按下键盘上的 **E** 键，门就会打开，屏幕左下角也会出现一个钥匙的图标。不要担心这个图标，它只是暗示了一下我们会在活动 5 中学习的内容。



图 023 - 当玩家拥有钥匙时，门就会打开。屏幕上将会显示一个图标，这暗示了我们会在下一个活动中学习的内容。

现在门打开了，我们还剩下最后一个任务。我们要添加终点目标。你可以在 **Blueprints** 文件夹中找到 **BP_endGoal**。将它拖动到关卡中，放置在建筑后方的斜坡顶部。

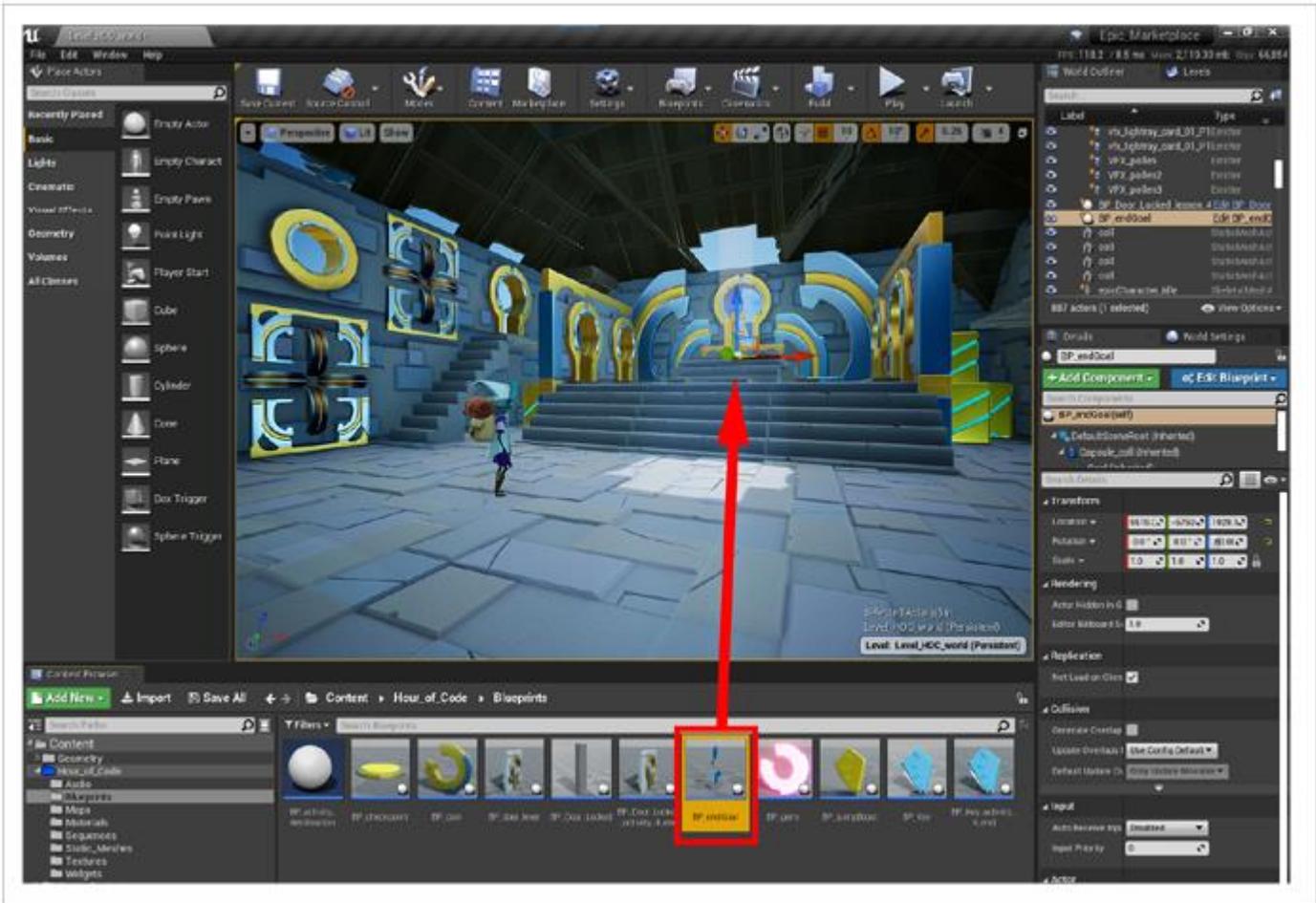


图 027 - 在锁定的房间中放置终点目标。

放置好目标之后，你就可以从头到尾完整地运行游戏了。你会注意到，当你接触到终点目标后，玩家就会看到多种数据，包括拾取的金币和宝石数量，以及完成关卡所耗费的时间。

要从头到尾完整地体验游戏，点击**运行**按钮旁的下拉菜单箭头，选择**默认玩家出生点**选项。现在点击**运行**按钮，就会让你从头开始。

记得要**保存所有**内容。

游戏设计挑战

你的游戏已经可以正常运行了，现在尝试多添加一些金币，再将钥匙移动到更好的位置。你能让游戏变得更有趣味性和挑战性吗？让其他人测试一下你的新游戏，并为你提出反馈意见。

