

# 房间

本页介绍了高级房间系统的工作原理以及如何利用它将关卡的各个部分链接在一起。

- [房间系统](#)
- [客房](#)
- [传送器](#)
- [传送器序列](#)
- [推子](#)
- [2D 蒙版](#)

## 房间系统

长期以来，自上而下引擎提供了[将不同场景连接在一起](#)的能力，或者使用[传送器](#)将关卡的各个部分连接在一起。在 v1.8 中，自顶向下引擎现在配备了强大且多功能的**房间系统**。这让您可以将关卡切成"房间"（它们是视觉上实际的房间还是只是区域由您决定），并使用[传送器\\*\\*](#)将这些房间连接在一起类（这些传送器是否在视觉上表示为门、传送门、虚无或其他任何东西也取决于您）。它利用 Unity 的 **Cinemachine** 摄像机系统的强大功能来处理房间之间的摄像机转换，以及一些 **MoreMountains** 类来处理淡入淡出和精灵蒙版。本页解释了如何做到这一点，并涵盖了房间系统\*\*的其他功能和选项。

该**KoalaRooms**演示现场是该系统的一个很好的例子，展示了很多房间，和不同的方式来设置传送器。您还可以查看 **MinimalRooms3D** 演示场景，了解如何设置 3D 房间的示例。

## 客房

首先，您需要在关卡中创建**房间**。无论您的关卡是由瓷砖、2D 精灵还是 3D 模型组成，都无关紧要。要创建一个房间，只需创建一个新的空对象，添加**BoxCollider2D**到它，如果你在 2D 的时候，或者**BoxCollider**如果你在 3D 的时候，和**室**组成。调整碰撞器的大小，使其与您房间的大小相匹配。

嵌套在房间下，作为孩子，您需要添加：

- 一个**限制器**：一个空的游戏对象，带有一个 BoxCollider。
- 一个**Cinemachine**虚拟摄像头。在此摄像机上，您可能需要一个 CinemachineCameraController 和一个 CinemachineConfiner。在那个 CinemachineConfiner 的检查器上，您需要将 Confine Mode 设置为 Confine3D，然后将您刚刚创建的限制器拖到包围盒插槽中。在 2D 中，您可能还想检查限制屏幕边缘。
- 一个或多个**传送器**（我们会讲到）



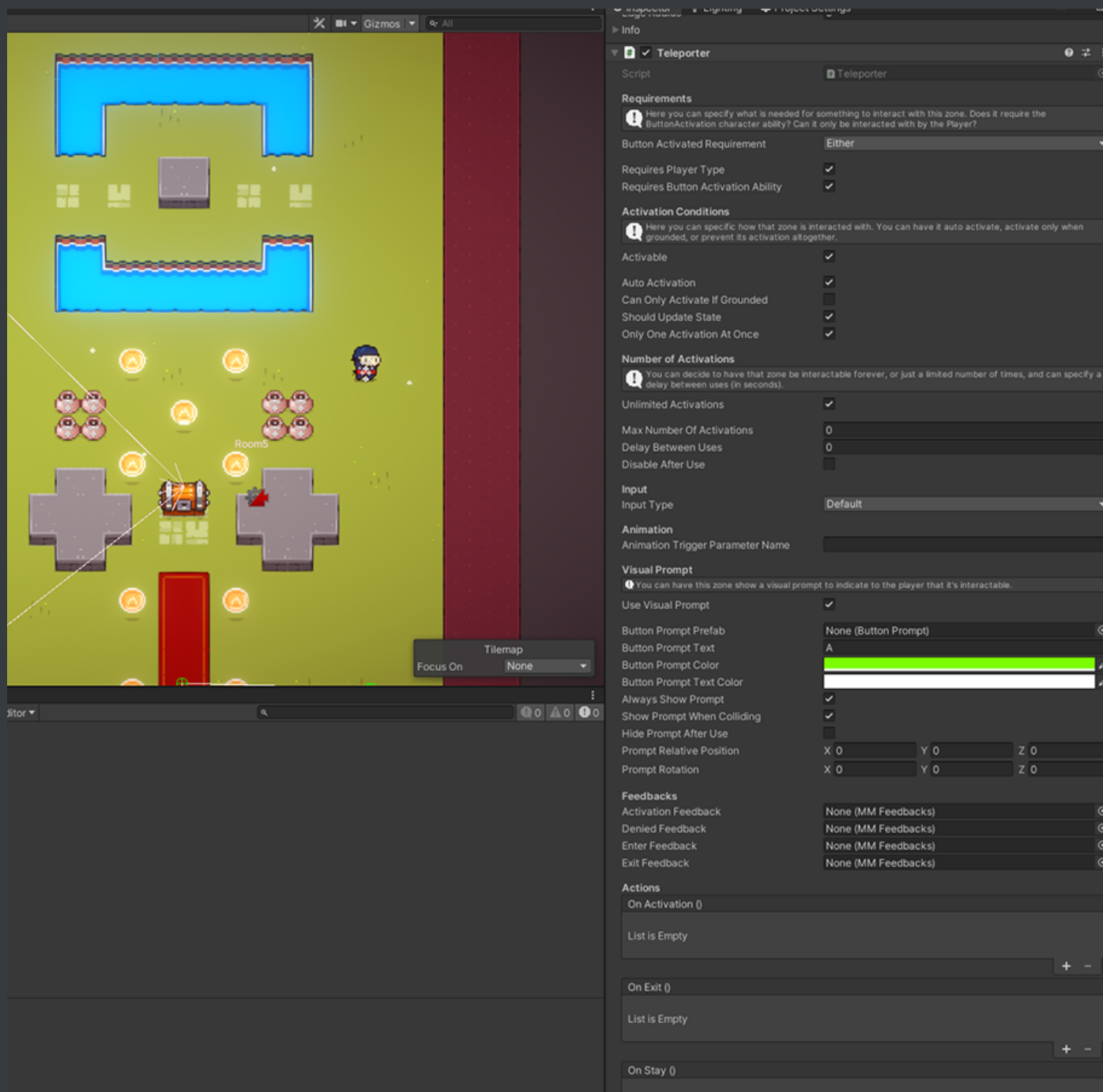
## KoalaRooms 演示场景中的 Room4

Room 类负责在开始时动态**调整限制器的大小**以匹配相机大小（仅在 2D 中），跟踪其状态（已访问或未访问）并在玩家进入房间、退出房间、进入房间时触发事件第一次。

一旦你创建了一个房间，添加更多房间的最好方法是**复制**它，重新定位它，并调整它的碰撞器的大小以创建更多。确保为它们提供**唯一的名称**。虽然不是强制性的，但这将使以后更容易找到它们。

## 传送器

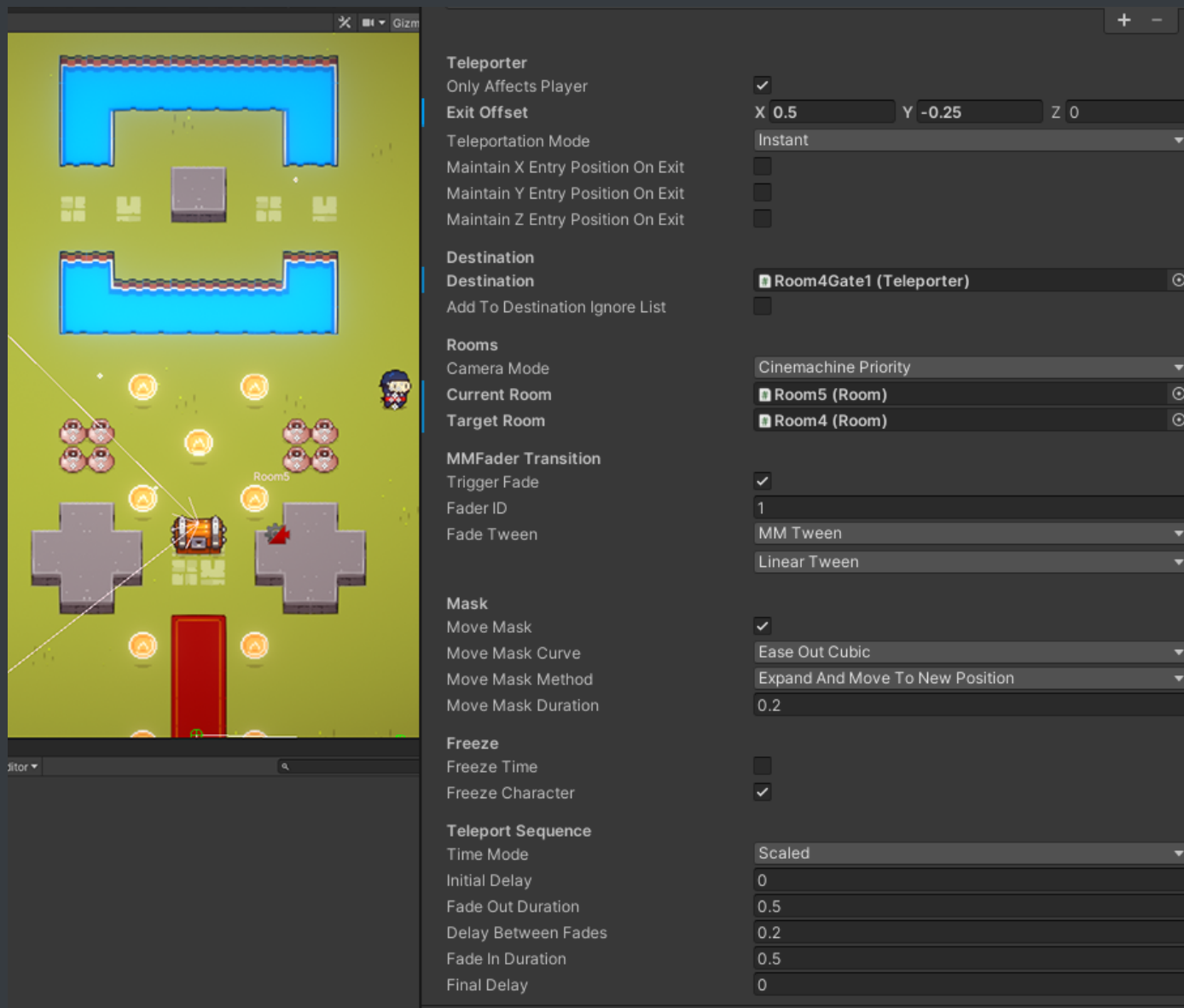
现在您有多个房间，是时候添加从一个房间到另一个房间的方法了。这是使用**Teleporter**类及其许多选项完成的。



## 传送器检查器的按钮激活部分

首先，在您的房间内添加一个空游戏对象，向其添加**BoxCollider2D**（2D）或**BoxCollider**（3D）（IsTrigger 设置为 true），然后向其添加**Teleporter**脚本。传送器有很多选择。您可以定义激活要求、激活条件、激活数量、提示、输入、反馈等等。对于房间之间的门，也许您只想将**AutoActivation**设置为 true，这样任何进入碰撞器的角色都会被移动到新房间。

在**传送师**的具体领域是在检查的底部（顶部是共同所有ButtonActivated区）。



## 传送器的检查员的其余部分

唯一需要设置的是**Destination**，您必须在其中拖放另一个传送器，以及**Target Room**，您需要在其中拖放 Destination Room。

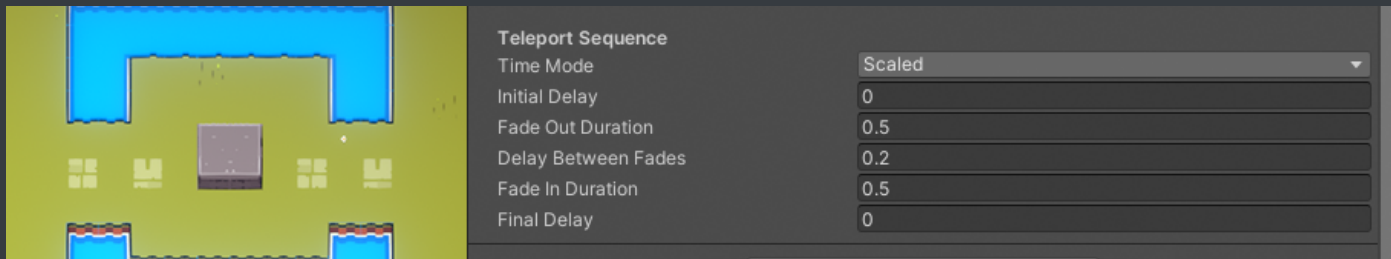
在**Teleporter**部分，您可以决定该传送器是否仅影响玩家角色、退出偏移量（当有物体退出时添加到传送器的变换位置）、传送是应该是即时的还是进入和退出位置之间的补间，以及是 x 还是 y 位置应该保持。

该**冻结**部分，您可以选择是否要在传送点冻结时间和/或字符输入时。

在**房间**部分，如果您的 Teleporter 嵌套在房间下，则不必设置**CurrentRoom**，它将被自动检测到。您有不同的相机模式选项，推荐使用**Cinemachine 优先级**。

# 传送器序列

当进入 Teleporter 并激活它时（通过按下输入，或简单地输入它，或通过脚本，取决于您的设置），您会触发一系列称为**Teleporter Sequence**的事件。它被设计为尽可能可定制，并且易于扩展。它由几个步骤组成，您可以从 Teleporter 的检查器底部自定义其持续时间（以秒为单位）。



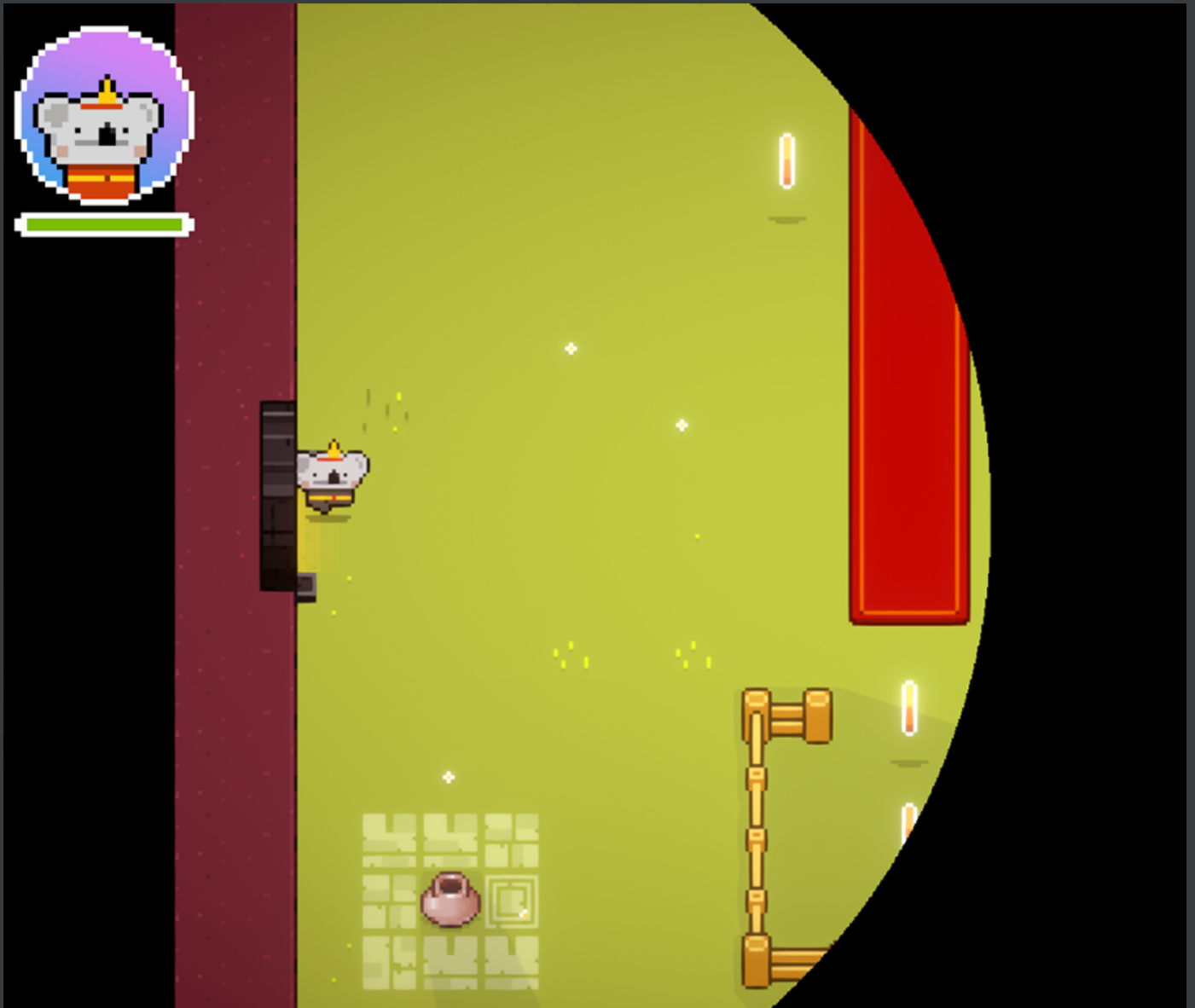
## 传送器序列的检查员

默认情况下，这些步骤的工作方式如下（请注意，并非每个步骤的所有事件都可能发生，具体取决于您的设置）：

- **SequenceStart**：激活区域（减少使用计数器，触发反馈等），防止相机跟随，冻结时间，冻结角色
- 等待 *InitialDelay*
- **AfterInitialDelay**：通过 MMFader 淡出场景
- 等待 *FadeOutDuration*
- **AfterFadeOut**：立即或通过开始平滑的补间位置传送对象，开始相机过渡，让当前房间知道玩家已经离开，开始遮罩过渡
- 等待 *DelayBetweenFades*
- **AfterDelayBetweenFades**：使相机再次跟随，通过MMFader淡入场景
- 等待 *FadeInDuration*
- **AfterFadeIn**：暂时没有
- 等待最终延迟
- **SequenceEnd**：解冻时间和角色

每个步骤都是一个单独的方法，如果您想添加更多功能，您可以轻松覆盖。

# 推子



在 KoalaRooms 演示场景中运行的 MMFaderRound

Teleporter 的检查器可让您检查在过渡到目标时是否要触发**淡入淡出**。这反过来会触发一个 MMFFadeEvent，它会被场景中的 MMFader 对象捕获，如果你有的话。大多数演示场景都可以，通常在它们的 UICamera 对象中。默认情况下，引擎带有两个推子，经典的**MMFader**（它将对 UI 元素的不透明度进行补间 - 通常是在整个屏幕上拉伸的黑色不透明精灵，但不一定），以及**MMFaderRound**，它将跨 UI 应用遮罩过渡元素。

## 2D 蒙版

通常在 2D 中，当处理关卡中的多个房间时，您希望**混淆**玩家在特定时间不在的房间。当然，您如何决定这样做与您的渲染选择密切相关。虽然尝试为此提供 3D 解决方案有点没用，因为它可能会干扰您的特定着色器，但引擎为该问题提供了 2D 解决方案：MMSpriteMask。您可以从头开始创建一个，或者简单地复制**KoalaRooms**演示中的一个，它已准备好使用。

在它的检查器上，您将能够在开始时自动设置场景中的所有渲染器（这很有用，否则它实际上会掩盖场景视图中的事物，使关卡编辑更加棘手）。如果您希望某个对象在运行时被 Mask 忽略，只需在其上放置一个**NoMask**标签即可。