

Character 类

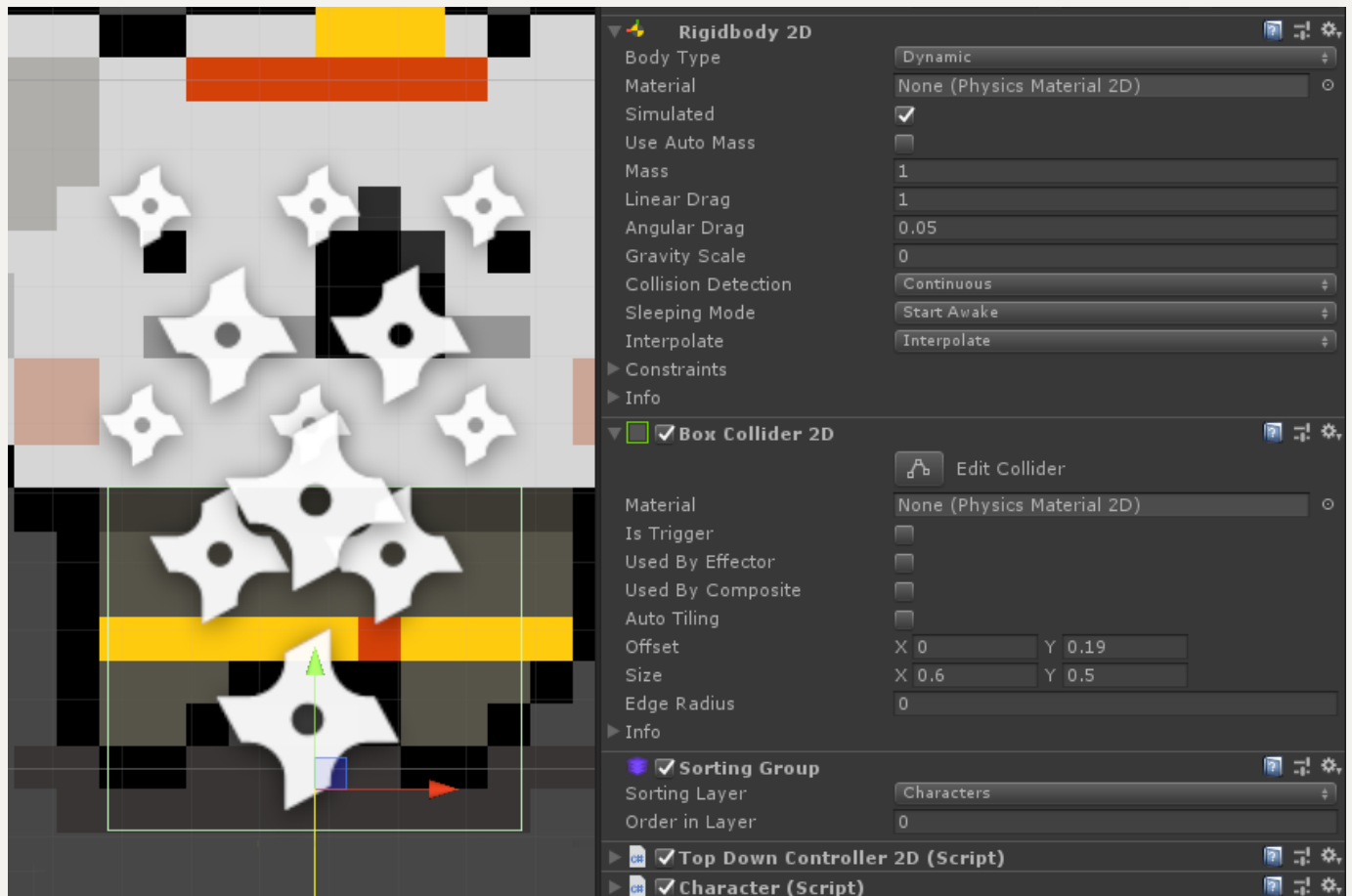
本页介绍了 Character 类、它使用的状态机以及如何使用它来创建您的角色。

- [介绍](#)
- [2D 特定组件](#)
- [3D 特定组件](#)
- [自顶向下控制器](#)
- [特点](#)
- [健康](#)
- [角色能力](#)
- [等级制度](#)

介绍

引擎中的每个角色都需要许多组件才能工作。其中一些是特定于**2D** 的，其中一些是特定于**3D** 的。它们中的大多数是常见的。此页面将为您提供有关在 TopDown Engine 中顺利创建角色所需了解的主要角色的信息。

2D 特定组件

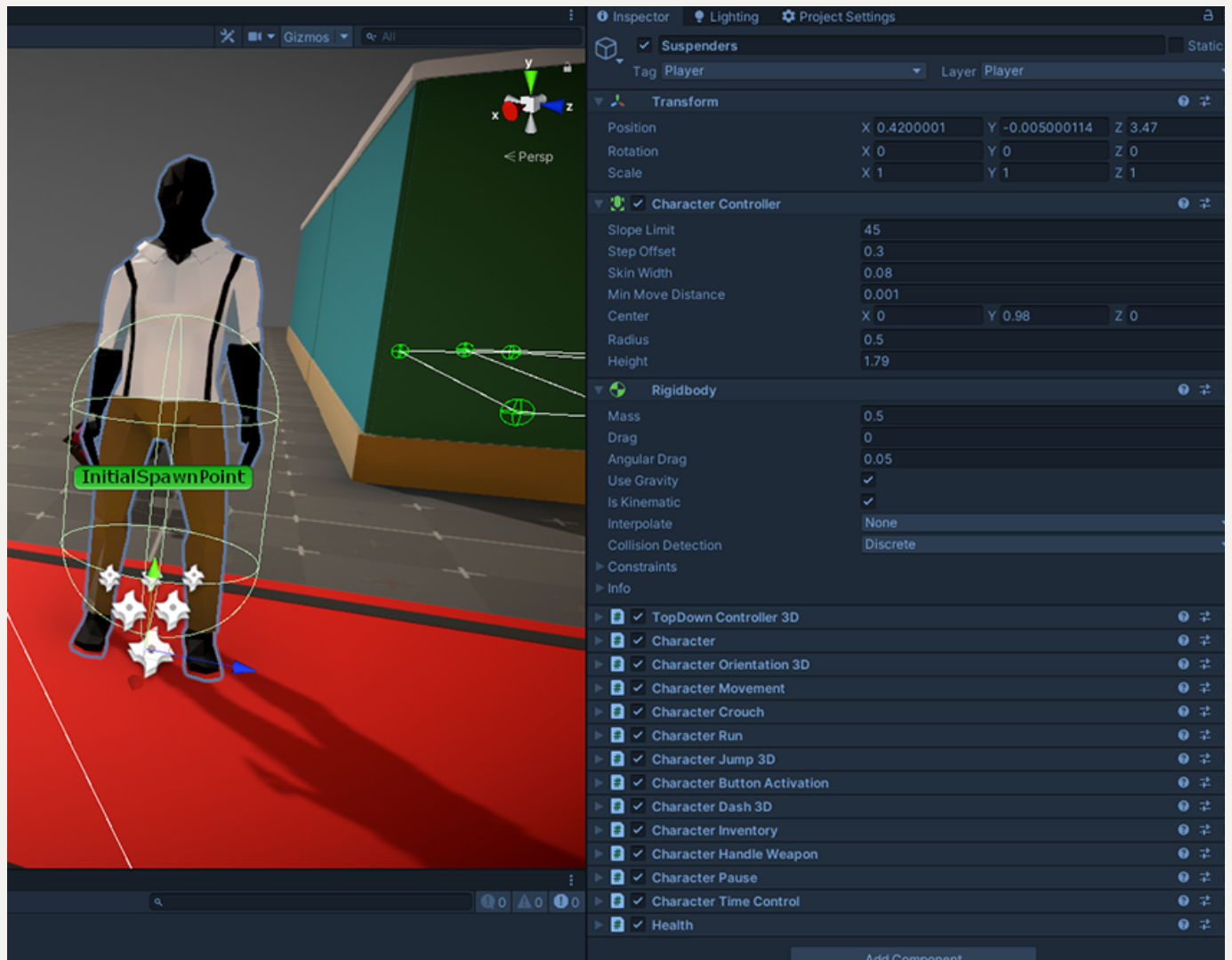


2D 角色的刚体和碰撞器设置

如果你想要一个**2D** 角色，你需要一个**RigidBody2D**组件（动态、连续碰撞检测、开始唤醒、插值），以及一个**BoxCollider2D**（非触发）。设置碰撞器没有任何规则或义务，但我喜欢让我的碰撞器包含角色底部的一半，如上图所示。这将允许角色的头部位于其上方的墙壁前，并且通常会给角色在“地面”上占据的位置提供良好的错觉。

此外，您可能希望使用**SortingGroup**组件，设置为字符排序层。这将自动处理排序，以便您的角色出现在 Y 轴下方对象的“后面”，以及其上方对象的“前面”。

3D 特定组件



3D 角色的刚体和碰撞器设置

如果你想要一个**3D** 角色，你需要一个刚体（运动学，使用重力，离散碰撞检测）和一个角色控制器。

自顶向下控制器

那么你的角色将需要一个 TopDownController。选择 TopDownController2D 或 TopDownController3D 版本，而不是基础版本。两者的设置略有不同，但大多数情况下，您需要确保正确设置图层蒙版（控制器应将图层视为地面、墙壁等）。

特点

这是连接所有其他类的中心类。它本身并没有什么作用，但确实充当了一个中心点。这就是你定义玩家是人工智能还是玩家控制的地方，它的模型和动画师所在的位置，诸如此类。它也是在运行时控制所有角色能力的类。

这是一个非常核心的类。从它的检查器中，您将能够定义一些东西。如果是玩家角色，**PlayerID** 字段非常重要。它必须与 InputManager 中的完全匹配（这就是 InputManager 知道要控制哪些字符的方式）。

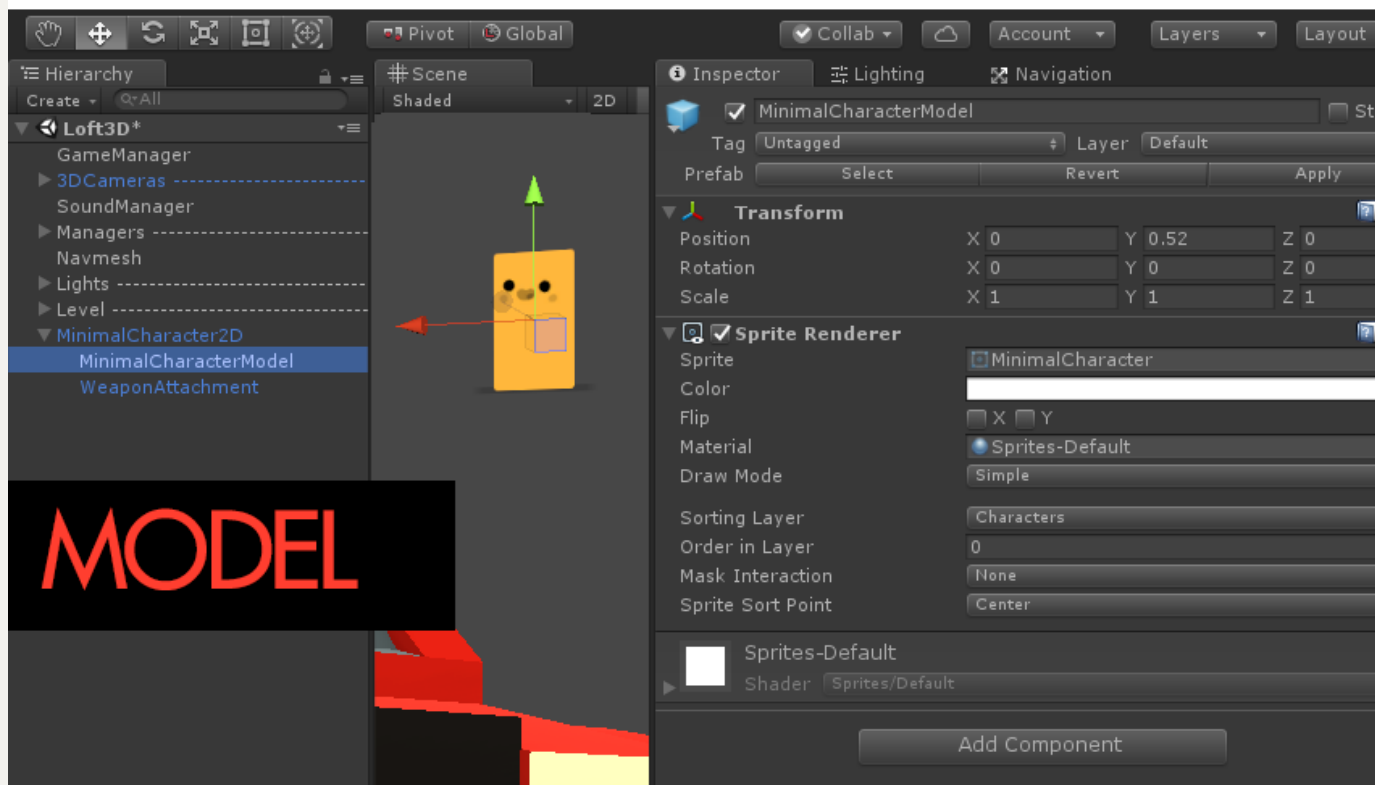
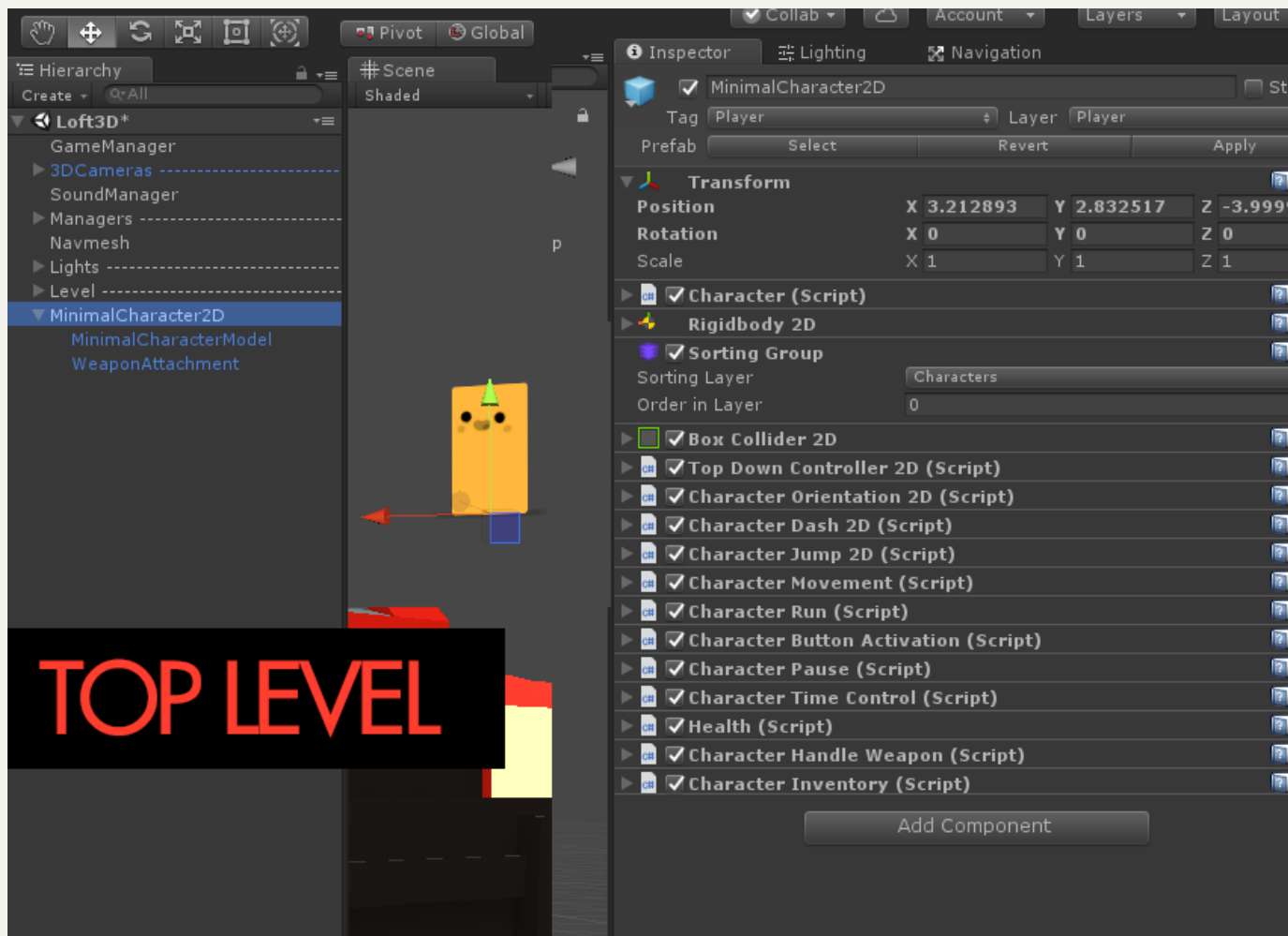
健康

Health 组件处理伤害、健康获得/损失，以及最终的死亡（和重生）。如果你的角色可以受到伤害，这是强制性的。它的各种方法将允许您从初始健康值中删除/添加健康点，它将处理损坏和相关效果（视觉和碰撞等）。

角色能力

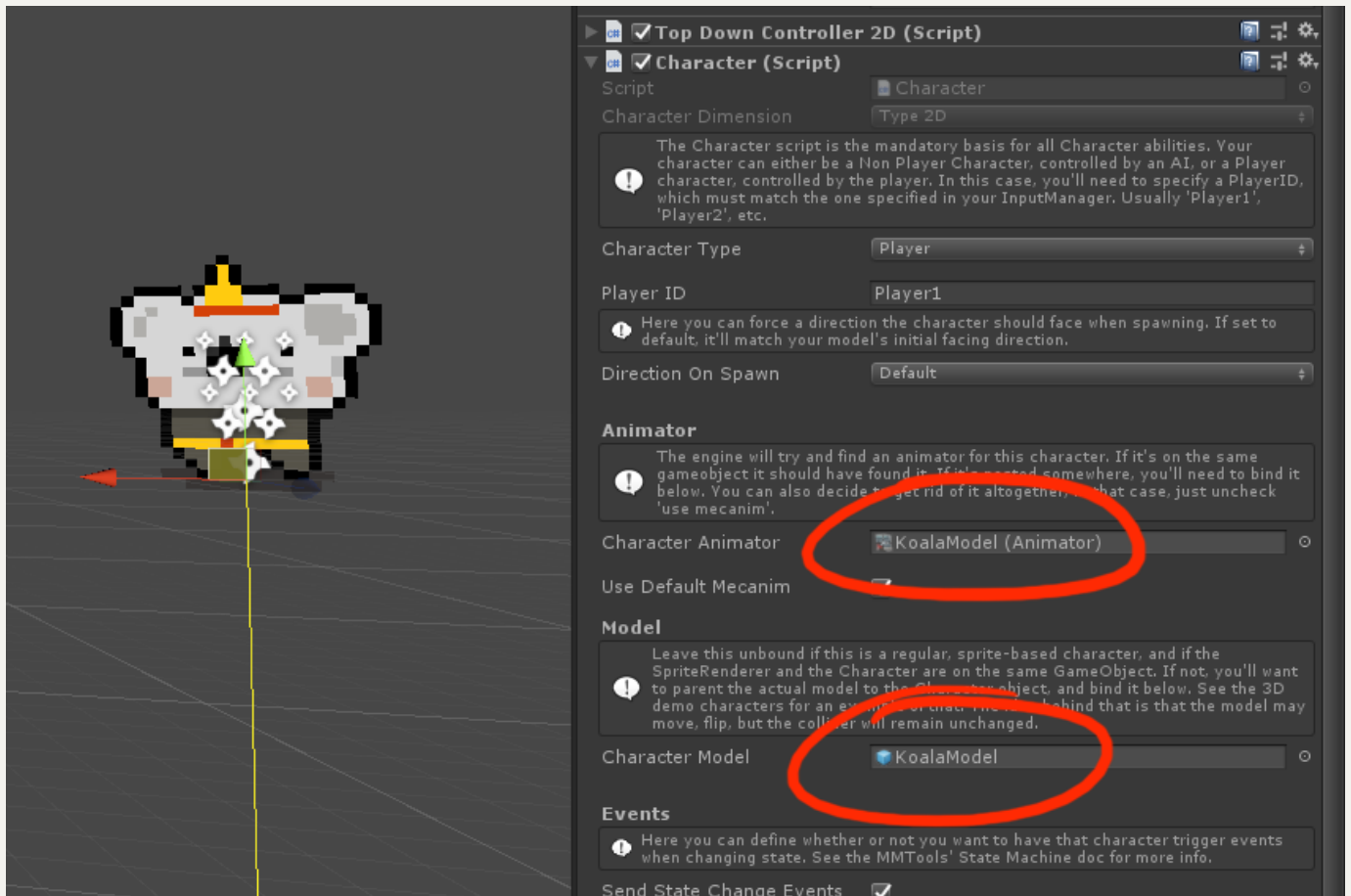
角色的最后（但并非最不重要的）部分将是角色能力。该资产包含超过 15 种能力，从简单的如 CharacterMovement 到更复杂的如武器处理。它们都是可选的，你可以选择你想要的任何一个。您当然也可以轻松创建自己的能力来构建自己的游戏。[您可以查看此页面](#)以了解有关角色能力的更多详细信息。

等级制度



推荐的 Characters 层次结构

无论您如何创建角色，真正重要的一件事是了解如何构建角色。你要分开的逻辑从（在刚体，TopDownController，角色能力等），视觉（模型/精灵/脊柱设置等）。您可以拥有更多层（也许您的动画师位于其自己的层上，等等），但推荐的设置是上图中的设置：具有所有逻辑的顶层，以及嵌套在其中的模型。



让 Character 组件知道 Animator 和 Model 的位置

确保从检查器将模型链接到角色类，动画师也是如此。其他一些类（定向能力、健康）也可能要求您通过他们的检查员告诉他们模型或动画师的位置。