武器

此页面描述了 TopDown Engine 武器背后的机制，以及如何创建和使用您自己的武器。

介绍

自上而下引擎包括一个**通用武器系统**，允许您的角色装备一种武器（基于射弹、近战或您能想到的任何武器）、切换到另一种武器并使用它。该引擎还包括一些您可以查看的武器示例，您可以在创建自己的武器时将其用作基础。

CharacterHandleWeapon 能力

为了能够装备和使用武器，角色必须有一个**CharacterHandleWeapon**组件。从它的检查器中，您可以选择为您的角色选择一种武器，定义它是否可以拾取新**武器**，并指定一个**武器附件**。这是对您的角色（或嵌套在其预制件中）的转换，以便武器附加到其上。如果不指定，武器将附加到预制件的根级别。

什么**CharacterHandleWeapon**然后部分将做的是检查有关武器使用按钮的变化，如果它被按下/释放，转移，目前装备信息武器。您还可以从其检查器中定义输入**缓冲**参数，这将影响输入请求的持续时间（它们执行的时间越长，系统"记住"您希望使用另一种武器的时间越长）。

武器类

自上而下引擎中的所有武器都源自**Weapon**类。你当然可以从那个转移，但这就是所有例子的工作方式。Weapon 类旨在进行扩展，并将定义所有或大多数武器共有的许多内容。以它为基础，您将能够创造一切，从霰弹枪到火箭发射器，还有武士刀或擒抱枪。除了为动画、声音、状态管理提供坚实的基础之外，它还允许您在子类中定义使用武器时会发生什么，并从那里构建。您可以查看**ProjectileWeapon**和**MeleeWeapon**类，以了解如何从相同的脚本创建截然不同的武器的示例。

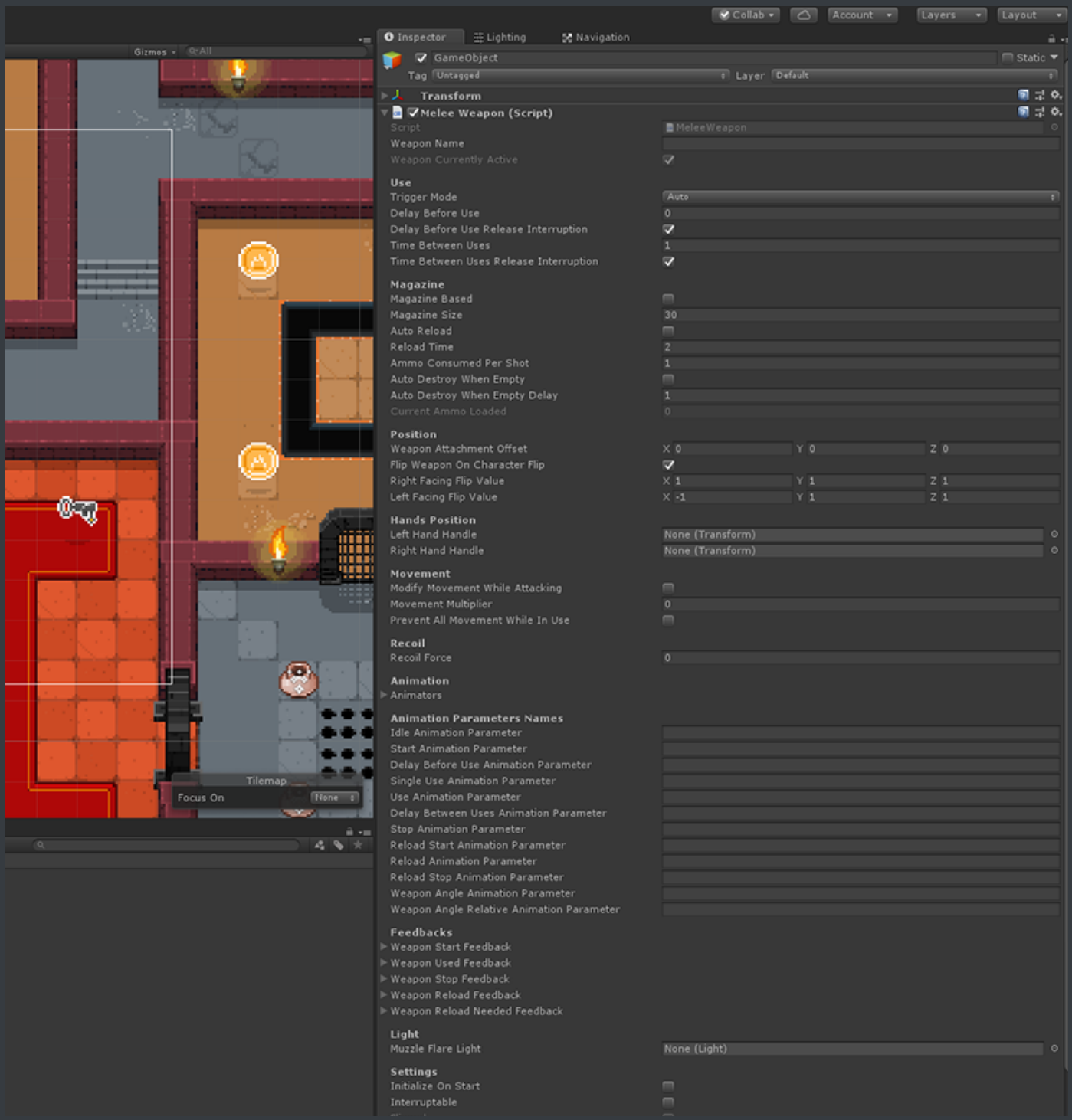
制造武器

要创建您自己的武器，您可以按原样重用 ProjectileWeapon 或 MeleeWeapon 脚本，扩展它们，或创建您自己的 Weapon 子类。在任何情况下，您都需要**创建一个武器预制件**。为此，您需要一个游戏对象。基本上，您可以选择拥有可见的武器或不可见的武器。

拥有精灵、模型或空游戏对象后，只需将武器（ProjectileWeapon、MeleeWeapon 或您自己的）脚本添加到其中即可。从其检查器中，您将能够设置动画、声音、开火时触发的效果，并指定角色翻转时武器应如何翻转。

下面是一步一步创建近战武器以供参考：

- 创建一个空的游戏对象
- 添加一个近战武器组件



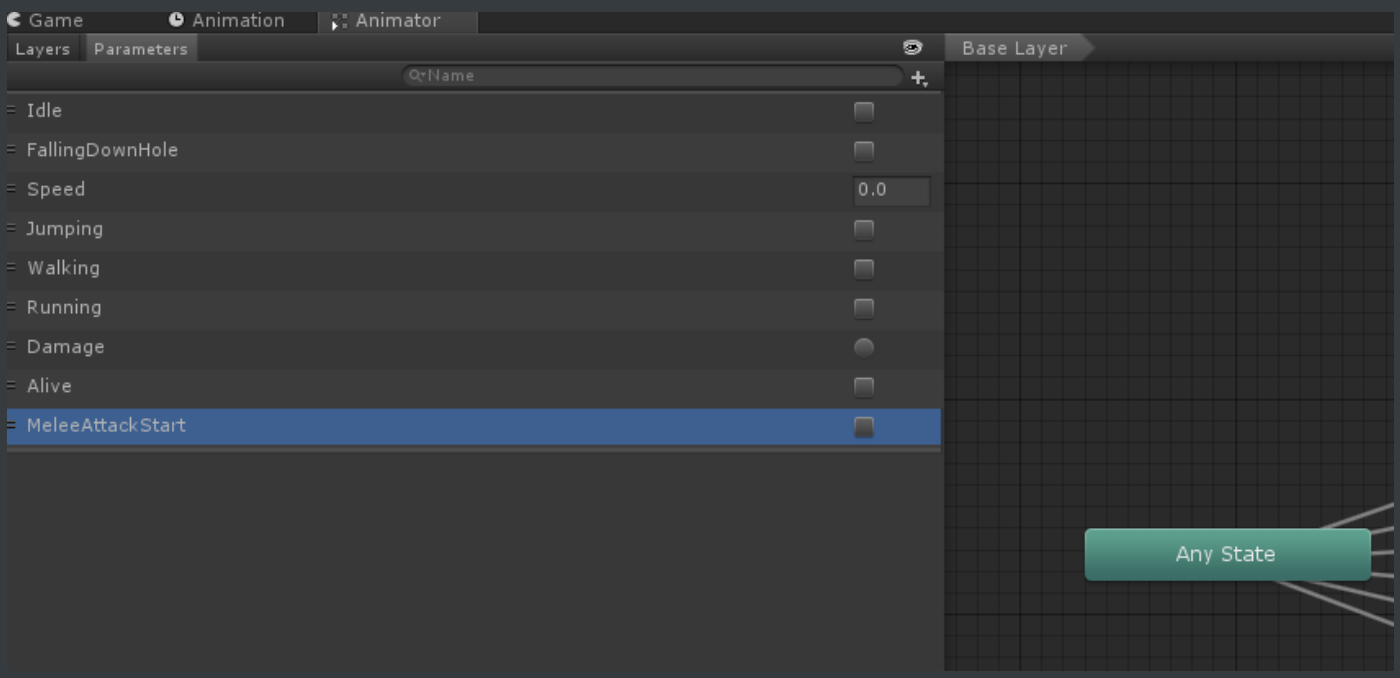
带有 MeleeWeapon 组件的空游戏对象

- 在近战武器检查器中，设置以下参数：触发模式为半自动，伤害区域形状为矩形，区域大小为 {1,1}，区域偏移为 {2,0}（这样伤害区域将在我们的角色），主动持续时间为 0.2（这是一次短暂的攻击，基本上是一拳）。然后将伤害造成的目标层掩码设置为敌人（以及您想要对其造成伤害的任何其他事物），并将伤害造成的伤害设置为 10。
- 将此游戏对象重命名为 MeleeAttackWeapon（或您喜欢的任何名称）
- 将此游戏对象拖入您的层次结构中以制作预制件
- 选择您的角色，并将新创建的预制件拖到其 InitialWeapon 插槽中（或者您可以通过脚本

使用 ChangeWeapon 方法来装备它）

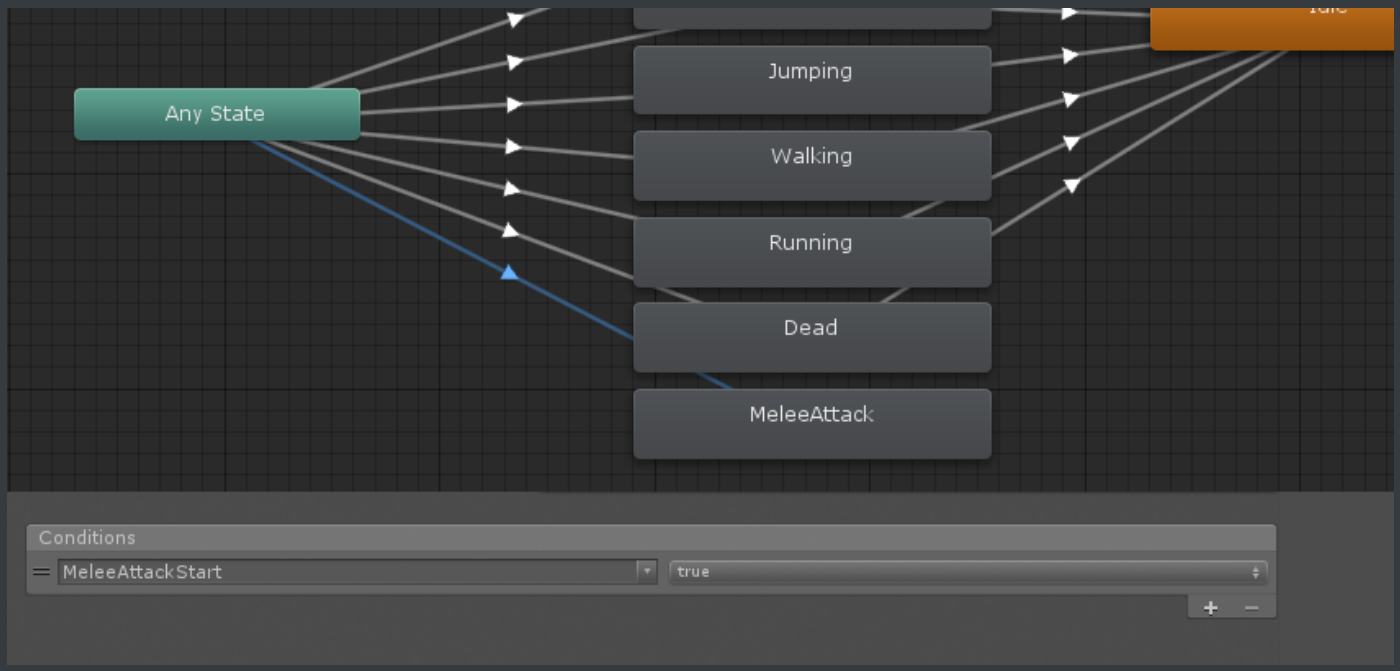
将武器绑定到我们的角色

- 就是这样，你现在有了一个有效的武器。按播放，然后按 E（默认情况下），您将四处挥拳并破坏敌人/物体。但是到目前为止，您在攻击时可能什么也没看到。让我们将动画绑定到该攻击。
- 选择你的角色的动画师，如果你的攻击动画不是这样的话，把它拖进去。
- 创建一个新的动画参数（参数面板右上角的小 + 按钮），在我们的示例中，我们将其命名为"MeleeAttackStart"



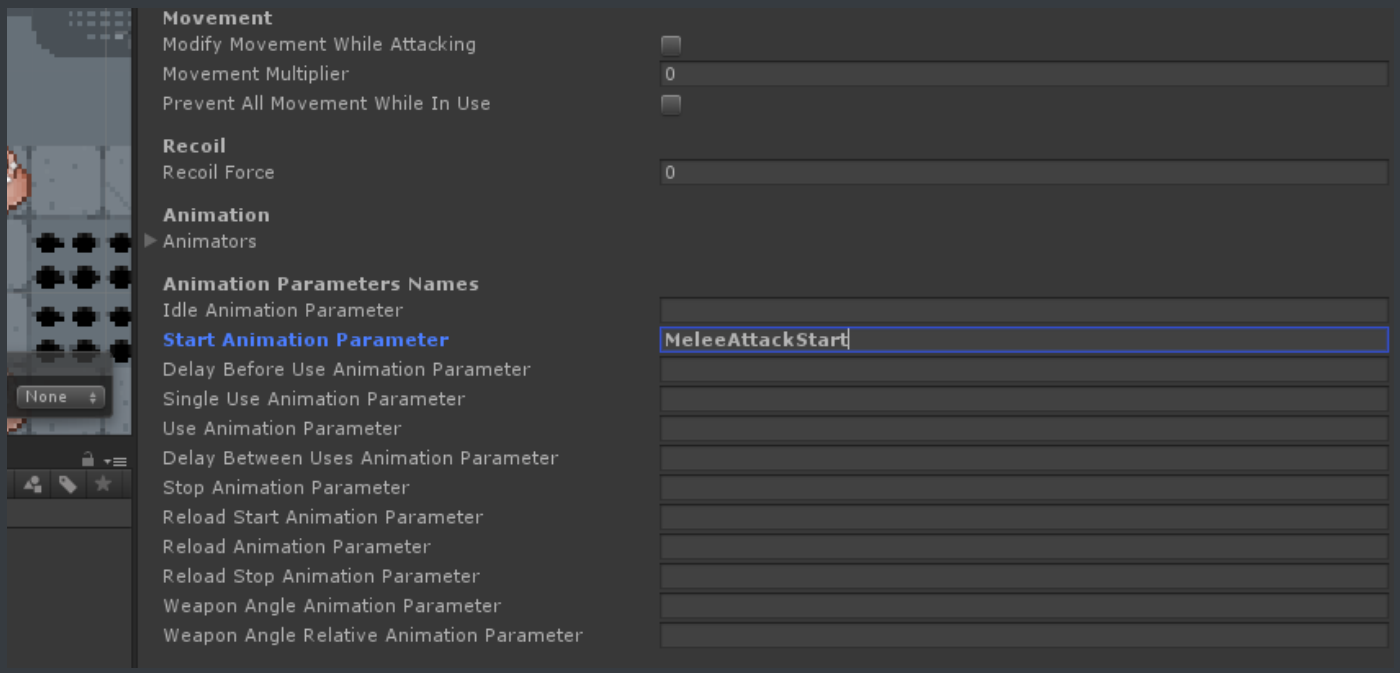
添加新的动画参数

- 创建到该动画的过渡。



为我们的攻击动画添加一个新的过渡

- 返回武器的检查器，并用动画参数的名称填充 StartAnimation 字段。在我们的示例"MeleeAttackStart"中。



为我们的攻击动画添加一个新的过渡

- 再次按下播放，用你的武器攻击，你现在看到你的角色在攻击。您可以添加其他动画，用于何时进行攻击、结束等。

射弹武器

射弹武器需要附加一个**对象池组件**。您需要添加**SimpleObjectPooler**或**MultipleObjectPooler**到你的武器，否则它不会射击任何东西。对象池是一种在射弹发射时回收射弹的方法，这样您就不会在进行时实例化/销毁对象，这对您的游戏性能来说要好得多。简单的对象池器将允许您只发射一种类型的射弹（这对大多数武器来说都很好），但是如果您想变得有趣并让它射击不同类型的东西，请选择 MultipleObjectPooler。在对象池器的检查器中，在 GameObjectToPool 字段中拖动要拍摄的对象，定义池大小（游戏将存储和重复使用的项目数量，具体数量取决于您的游戏、关卡大小等。基本上这个想法是你希望你的池永远不会被完全使用，但也不要太大）。

此外，您可以从检查器中定义每次射击的弹丸分布和数量。扩展由最大角度（在所有 3 个轴上）定义，可以是随机的，也可以不是。如果不是随机的，您可以将其与每次射击的多个射弹结合以创建图案。

弹丸

射弹武器通常发射射弹（我知道）。恰当命名的 Projectile 类将处理最常见的用途，从子弹到火箭。它将允许您创建以特定速度和加速度沿一个方向（受武器方向影响或不受其影响）移动的射弹。

对于大多数射弹，您需要添加一个 DamageOnTouch 组件，这样它们就会伤害到他们的目标，如果您希望它们在击中之前被摧毁，则可能会添加一个 Health 组件。

定时

创建武器时，时机至关重要，因为您希望将武器的组件与其动画完美同步。所有武器都带有两个变量，您可以直接从检查器中调整：**DelayBeforeUse**（输入注册和实际武器使用之间的时间 - 射击、斧头吊索等）和**TimeBetweenUses**（需要经过的时间才能拥有另一个武器使用（无论是在自动状态下，还是通过重复按下等）。在此期间武器不会"射击"）。

此外，近战武器还有两个你可以使用的变量：**InitialDelay**（武器使用和伤害区域激活之间的延迟）和**ActiveDuration**（伤害区域保持激活的时间）。这最后两个参数非常特定于损坏区域，当然您需要将这些值与 DelayBeforeUse / TimeBetweenUses 同步以防止冲突。

您还可以设置武器以防止在攻击时移动，确保您的角色不会在攻击过程中四处乱跑。

组合武器

该引擎内置了对**组合武器的**支持，而且它们非常容易创建。您可以查看 KoalaSword 预制件，了解它们如何工作的示例。

重复使用时，带有 ComboWeapon 组件的武器将在 Inspector 中从上到下依次**循环**添加到同一对象的所有武器。如果您已选中**Droppable Combo**选项，则可以"丢弃"组合，这意味着如果您未能在 Drop Combo Delay 内再次使用您的武器，您使用的下一个武器将排在第一位。

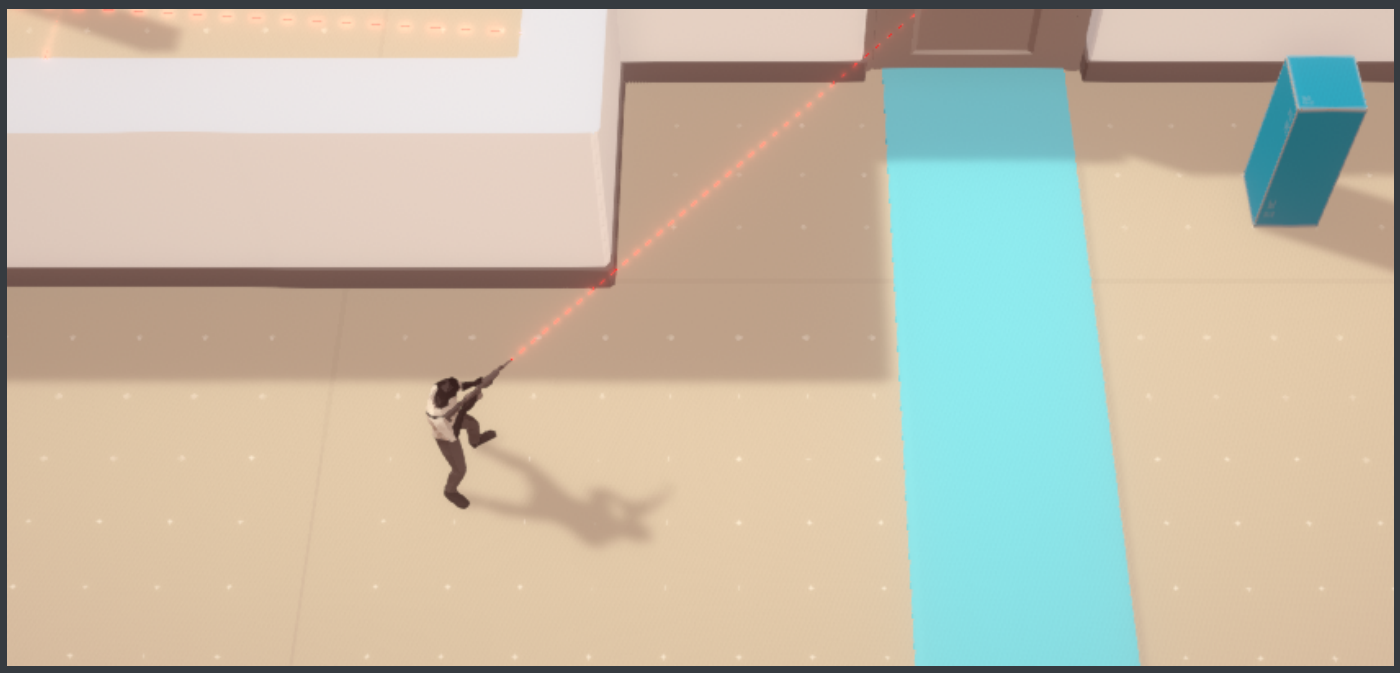
要创建一个，只需将多个武器组件（近战武器、射弹武器或您自己的武器扩展）添加到一个对象，然后像设置任何其他武器一样设置它们中的每一个。然后添加一个**ComboWeapon**组件。在其中，您只需要定义是否可以删除组合，以及延迟时间（以秒为单位）。可丢弃的组合意味着如果您在两次攻击之间等待太长时间（超过指定的延迟），则下一次攻击将再次成为第一次攻击，否则它将移动到下一个。

次要武器

如果需要，您的角色可以一次装备不止一件武器。为此，您可以向其添加**CharacterHandleSecondaryWeapon**功能。它的工作方式与 CharacterHandleWeapon 完全一样（它扩展了它），但查看不同的输入。如果需要，您可以创建更多类似的能力（CharacterHandleThirdWeapon 等），只需覆盖 HandleInput 方法，遵循 CharacterHandleSecondaryWeapon 示例。

它也适用于库存。在您的 InventoryWeapon 上，您可以选择定义一个**HandleWeaponID**。这是一个 int，对应于您的目标能力类中指定的 ID。CharacterHandleWeapon 为 1，CharacterHandleSecondaryWeapon 为 2，如果您要创建更多这些，则还需要覆盖并增加该 int。HandleWeaponID 为 2 的武器将自动转到您的 CharacterHandleSecondaryWeapon。

武器激光瞄准器



运行中的激光瞄准器组件

您可以将**WeaponLaserSight**添加到您的武器中。它当然更适合作为射弹武器，但**没有限制**。它会做的是像激光瞄准器一样在武器前面投射激光。您可以设置碰撞遮罩，以便激光在遇到墙壁或平台时停止。您还可以从检查器中自定义激光的外观。

武器瞄准

将**WeaponAim**组件（WeaponAim2D 或 WeaponAim3D）添加到您的武器将允许您控制其方向，允许您的角色向上、向下射击等。您将能够从其检查器中定义其控制模式：

- **关**：没有武器瞄准
- **PrimaryMovement**：武器将瞄准您移动移动杆/键的任何方向。
- **SecondaryMovement**：武器将瞄准您将次要运动输入指向的方向。例如，默认情况下，这是 xbox pad 上的正确棒。
- **PrimaryThenSecondaryMovement**：如果有，将使用主要运动，否则使用次要运动
- **SecondaryThenPrimaryMovement**：如果有，将使用次要运动，否则将使用主要运动
- **鼠标**：武器将指向鼠标指针的方向
- **脚本**：武器不会通过常规输入瞄准，而是由另一个脚本控制（对人工智能、自动瞄准等有用）

这可以为每个武器定义，但也可以在**InputManager**上针对所有武器在全局级别强制**执行**。只需设置它的 WeaponForcedMode，它就会自动应用于玩家角色使用的所有武器。

您还可以定义**旋转模式**：

- **自由**：360°全旋转
- **严格4个方向**：上、下、左、右
- **严格的 8 个方向**：上、下、左、右和对角线（左上、右下等）

最后，您可以定义一个最大和最小角度来限制移动，例如只允许向上射击，或防止角色向后射击。

WeaponAim 组件还允许您在屏幕上显示武器的**十字线**。您需要指定一个显示模式（UI 或场景），一个游戏对象预制件（资产中包含两个演示，称为 SceneReticle 和 UIReticle，但您当然可以使用自己的）。您可以设置与标线应该在的角色的距离，或者它是否应该跟随鼠标，如果它应该在您瞄准武器时旋转，死区（与标线不会影响到的角色的距离）目标），以及是否应该隐藏鼠标指针。还有一些选项可以让场景 UI 标线与斜坡一起移动。如果您有一个激活的十字线，您还可以从检查器中定义 CameraTarget 应该如何移动。

自动瞄准

如果您希望您的角色自动瞄准敌人，您可以向您的武器添加**WeaponAutoAim2D**或**WeaponAutoAim3D**组件。这将需要武器上的 WeaponAim2D/3D 组件，设置为脚本模式（见上文）。在它的检查器中，您将能够定义一个图层蒙版，其中包含要在其上寻找目标的图层，以及将其视为障碍物的图层。您将能够设置扫描的半径和频率、旋转设置、如何影响相机目标、设置 AimMarker 预制件以突出显示目标，并定义在找到、更改或不再有目标时播放的反馈找到目标。您将在上校演示场景中找到所有这些设置的示例。

弹药

多亏了 InventoryEngine，引擎本身就支持**弹药**和创建**基于弹药的武器**的可能性。您可以创建两种基于弹药的武器：基于库存的武器和常规武器。

基于库存的武器将要求您的角色附加一个库存（通过 CharacterInventory 能力），而常规基于弹药的武器只会消耗无限弹药，但两者都允许您定义弹匣大小、是否重新加载等。

所有这些设置都可以在武器的检查器上进行设置。在那里你可以定义弹匣大小，武器是否在弹

匣空时自动重新加载，或者玩家是否需要按下重新加载键，重新加载需要多长时间（以秒为单位），以及多少每次射击消耗的弹药。

选中"基于杂志"复选框将告诉武器它应该从角色的库存中取出弹药。为此，您还需要武器上的另一个组件 WeaponAmmo。在那里您将能够指定系统应在库存中查找的弹药的名称（例如"子弹"），库存的名称是什么（通常是 MainInventory 但您可以选择任何您喜欢的名称，只要它已设置以这种方式在您的库存中），武器的弹药指示器应该显示的最大弹药，以及武器在装备时是否应该从库存中装载弹药。

武器和库存

如果您的角色正在使用库存，玩家现在可以使用输入快捷方式**切换武器**（循环使用库存中的所有可用武器）。默认情况下，这绑定到 xbox pad 上的 LB，或键盘上的 t。

此外，如果您的角色有武器库存，并且在开始关卡时该库存包含武器，则该武器将自动装备。

武器 IK

从武器的检查器中，您可以定义左右手附件以与 Unity 的内置动画**IK 系统一起使用**。这会将您角色的手连接到您武器的特定点，当您卸下武器时将它们分开，并让您的角色在您移动武器时自然地移动和瞄准。如果您想要一个随时可用的示例，Loft 演示中的演示字符都是这样设置的。

除了正确创建你的头像（Unity 有关于这方面的很棒的文档），你唯一需要简化这个过程的是 WeaponIK 类，理想情况下设置为与你的头像的动画师相同的级别。此类允许 3D 角色抓住其当前武器的手柄，并查看其瞄准的任何地方。这涉及到一些设置。你需要在你的角色上有一个 CharacterHandleWeapon 组件，它需要一个激活 IKPass 的动画师（这在动画师的图层选项卡中设置）并且动画师的头像必须设置为人形。

对于您想暂时禁用 IK 的情况（例如，在冲刺期间），您可以在动画师上添加**WeaponIKDisabler**，并且您将能够指定布尔动画参数名称列表，当为真时, 应该导致 IK 被关闭。

敌人武器

您会在整个引擎的演示中找到许多敌人使用的武器示例。创建一个**与**为玩家角色创建武器**完全相同的过程**。但是，您通常不能简单地复制玩家武器并将其放在敌人身上。根据武器的不同，您可能希望更改目标层、瞄准方法等，但总体逻辑保持不变。

弹药展示

AmmoDisplay是一个组件，您可以添加到 GUI 矩形以使其显示进度条和文本，以显示您的杂志和库存中剩余多少弹药。只需将它添加到一个空的 GUI 对象，然后在它的检查器中，您需要将它绑定到一个**ForegroundBar**和一个 Text 对象，并指定要显示该信息的玩家（通常是 Player1）。最后，选择您的**GUIManager**，并将您的 AmmoDisplay 拖到其**AmmoDisplays**数组中。

武器模型

虽然在许多情况下，您希望将武器的逻辑和视觉效果放在同一个对象上，但有时您可能希望将两者分开。该引擎为基于库存的武器提供了两种不同的方法，您希望在装备不同武器时能够激活不同的模型：

- 该**WeaponModel**类是你可以把你的对象特征下的类。从它的检查员，您将能够设置应该激活此模型的武器的名称。此名称必须与确切的武器项目**ID**匹配。然后你可以让模型瞄准实际武器的瞄准方向，让它动画，覆盖武器使用，重新绑定 IK，它甚至带有 MMFeedbacks 钩子，除了已经在武器上的钩子。在这里，您将找到设置 WeaponModel 的简单步骤，您可以在**上校**演示场景中找到它的实际示例。这就是上校步枪的工作原理，请在上校预制件上查找 FBIAssaultRifleWeaponModel 节点以供参考。
- 如果您不需要那么**多功能**，您也可以简单地在您的角色上放置一个**WeaponModelEnabler**（在顶级，以及能力、健康等）。它要简单得多，只需要您在角色上定义一个节点列表，以便在装备某种武器时启用这些节点。要指定哪个武器对应哪个模型，您需要放置一个**WeaponAnimationID**，它必须与您在武器上设置的 WeaponAnimationID 相匹配（在其"动画参数"面板下）。您将在文档的 Recipes 部分找到设置 WeaponModelEnabler 的步骤，并在 MinimalSword3D 演示场景中找到在 LoftTie 预制件上运行的示例。

这两个系统将允许您基于当前武器启用（子）模型，并且应该涵盖大多数用例。当然，您会想要选择**其中一个**，而不是两者都选择。