Unity Ragdoll

# Unity自动创建的Ragdoll

官方文档：<https://docs.unity3d.com/cn/current/Manual/wizard-RagdollWizard.html>

Unity可以简单快速的创建Ragdoll，只需要选择一些必要的骨骼，系统会为其自动添加所需组件，该方法只适用于做死亡时使用，一旦开启就会倒下。

操作方法：

从菜单栏中选择 GameObject > 3D Object > Ragdoll

图形用户界面, 应用程序

描述已自动生成

可打开对话框用于创建Ragdoll的对话框，将骨骼拖入对应的位置即可。

表格

中度可信度描述已自动生成

经过尝试，对话框中命名会和实际模型有所差别，以下是经过试验后比较合理的一套对应方法：

Pelvis：骨盆

Left Hips：左大腿

Left Knee：左小腿

Left Foot：左脚

Right Hips：右大腿

Right Knee：右小腿

Right Foot：右脚

Left Arm：左大臂

Left Elbow：左小臂

Right Arm：右大臂

Right Elbow：右小臂

Middle Spine：脊柱

Head：头部

Total Mass：总体重

设置好以后单击Create就可以自动创建一个布娃娃了，创建布娃娃时要注意忽略自身碰撞体之间的碰撞。

# 手动创建Ragdoll

思路：通过Configurable joint对关节进行约束，实现布娃娃的效果。

手动创建需要为各个重要的骨骼手动添加Rigidbody、Collider和Configurable joint，

[Rigidbody](https://docs.unity.cn/cn/2020.3/Manual/class-Rigidbody.html)：刚体，用于物理模拟

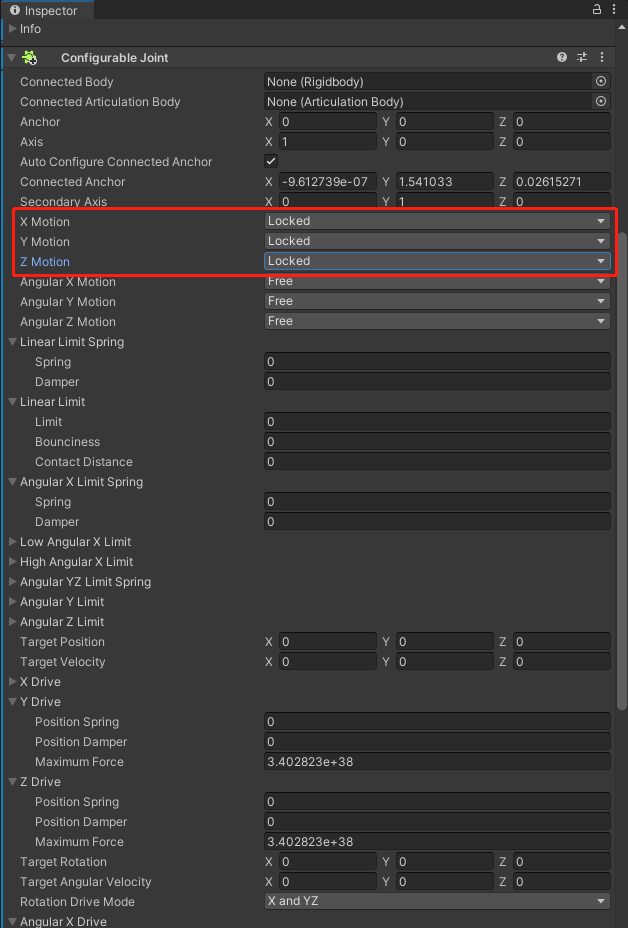
[Collider](https://docs.unity3d.com/cn/current/ScriptReference/Collider.html)：碰撞体，用于产生碰撞

[Configurable joint](https://docs.unity3d.com/cn/current/Manual/class-ConfigurableJoint.html)：用于约束关节

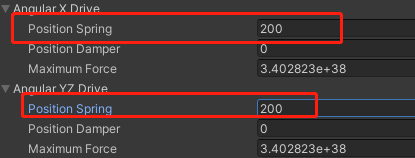
实现：

为骨盆、左右腿、脊柱、左右手和头部添加这些组件，为Configurable joint组件中的Connected Body赋值，值为上一级挂载了刚体的对象，比如大腿的上一级就是骨盆，小腿的上一级时大腿，最后全部会直接间接与骨盆连接。而骨盆不需要添加值，为骨盆添加组件的原因是对其进行一些不会倒地的约束。

除骨盆以外，将其它的所有关节的移动锁住，即将X Motion，Y Motion，Z Motion的值设为Locked。



再设置对旋转的弹簧扭矩，如图，除骨盆外设置为200（该值需要根据不同需求自行设置），值越大，向回弹的弹簧力度越大。



骨盆的该值要设的大一些，否则会倒地。

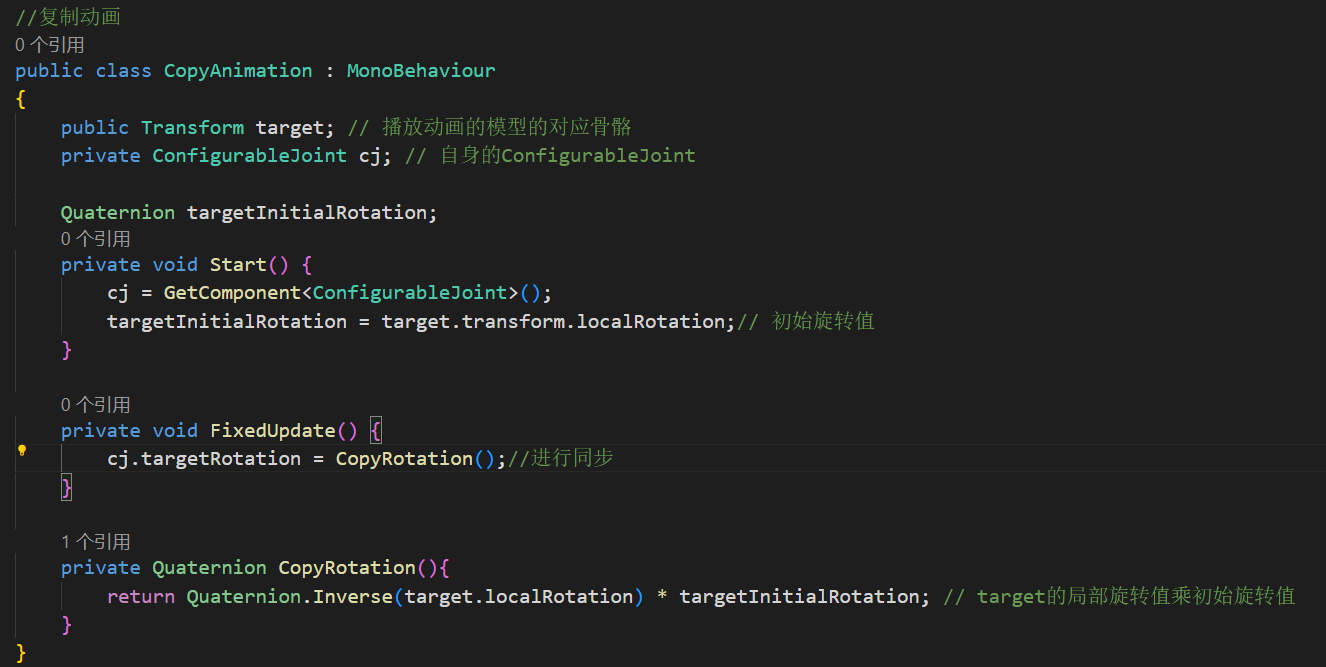
同样要注意忽略自身碰撞体之间的碰撞。

# 与动画的混合

思路：物理与动画是冲突的，可通过同步的方式去进行混合，即场中有两个模型，一个负责物理，一个负责动画，在动画模型播放时，将各个骨骼的旋转值同步给物理的模型，以达到让物理去模拟动画的效果，但此方法会受碰撞体与其它碰撞之间的影响比如产生摩擦，所以同步效果并不好。

实现：

该脚本负责同步骨骼



脚本中只有一个重要方法：

CopyRotation方法通过Quaternion.Inverse()方法来获取目标对象的逆旋转值，然后将其乘以初始旋转值targetInitialRotation，得到的结果就是当前对象与目标对象之间的旋转差。

将该脚本挂在物理模型所有需要同步的骨骼上，target选择动画模型的相应骨骼，这样运行时物理模型就会向动画模型去同步。

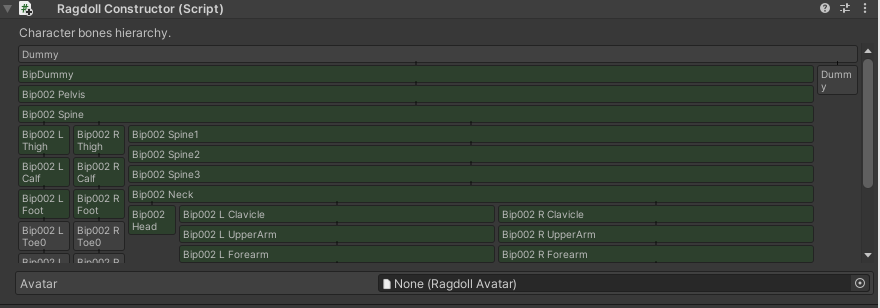
# 布娃娃插件：Ragdoll Mecanim Mixer

参考链接：<https://zhuanlan.zhihu.com/p/411635857>

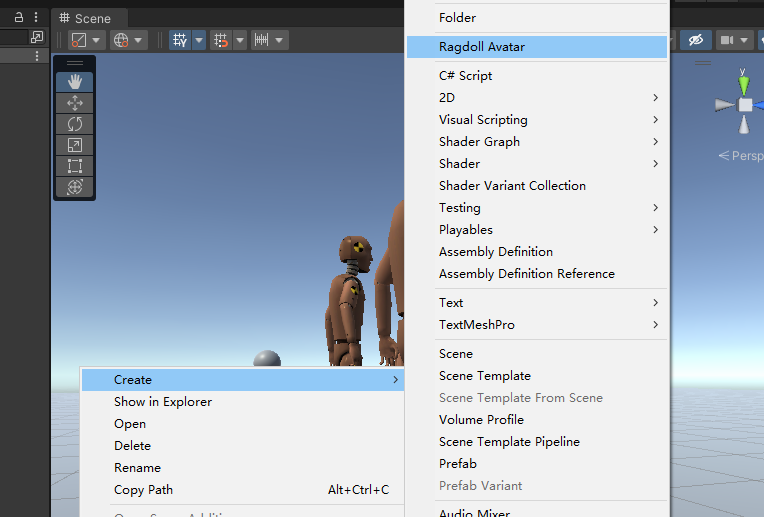
该插件可以自动创建布娃娃效果并实现自身碰撞忽略，而且碰撞体与骨骼分离，物理模拟驱动碰撞体，动画驱动骨骼，再根据需求进行同步，达到物理与动画混合的效果。混合效果比单纯同步好，也可以在动画驱动和物理驱动之间切换，可以满足绝大部分需求。

使用方法：

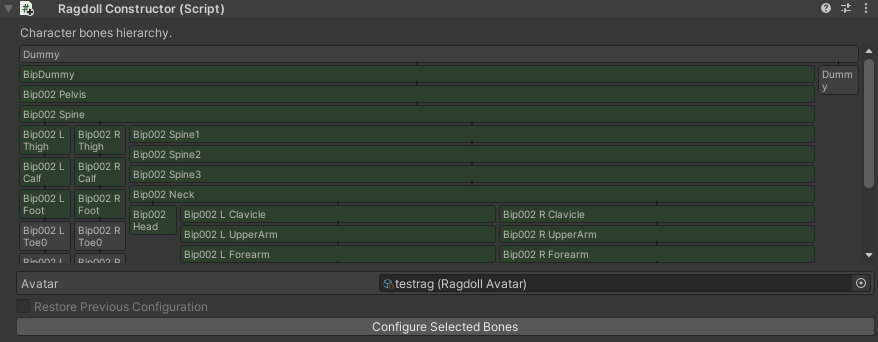
为人物模型挂载上Ragdoll Constructor，选择需要添加刚体碰撞体的骨骼



在Project视图中右键选择Create会有一个Ragdoll Avatar选项，新建后将其拖到Ragdoll Constructor组件的Avatar的位置上

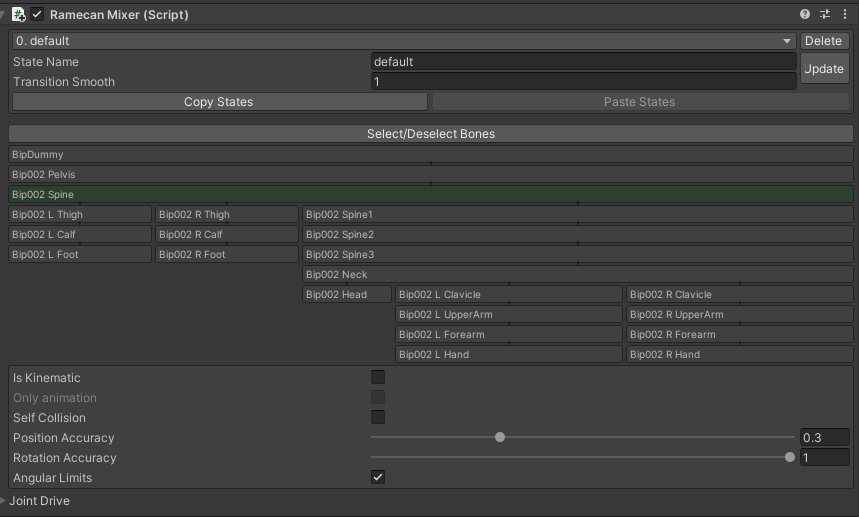


设置好后如下图所示:



单击按钮后会有一些简单的设置如总质量、物理材质等。单击创建即可创建好一个与动画混合的布娃娃，需要注意的是,若使用该插件，必须要有一个动画才可以使用，否则会缩成一团。

创建好后会自动挂载一个新的组件叫做Ramecan Mixer这个组件主要负责控制多种状态切换及动画融合功能，可以配置每个骨骼的详细参数，当不需要Ragdoll物理效果时可以选择 Is Kinematic 和 Only Animation，完全交由动画驱动，当需要Ragdoll物理效果时取消勾选 Is Kinematic，可以配置每根骨骼的详细参数和角度限制。

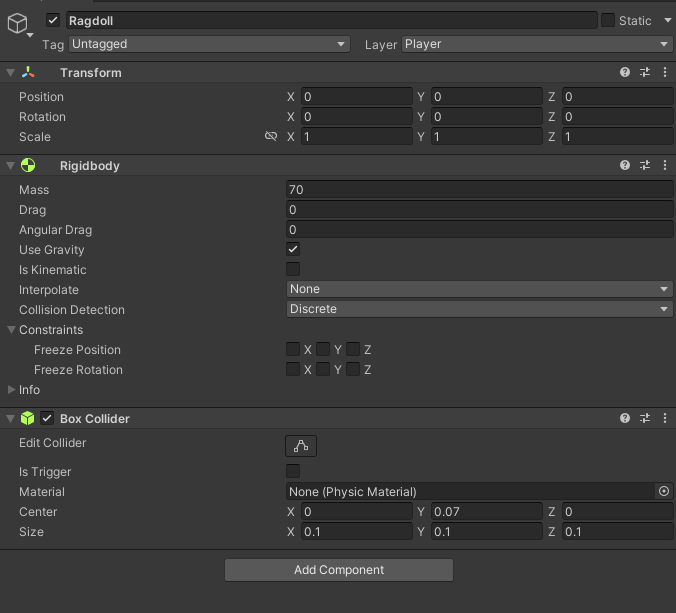


# 被撞后移动

思路：在所有碰撞体的父级创建一个刚体和小的碰撞体，在碰撞发生时，在碰撞中逐帧获取的冲量总和施加给这个刚体，由这个刚体去控制移动（实际就是一个刚体在移动），同时使用这个总冲量去解算出速度，由速度驱动动画混合树混合出一套动画，再将动画与物理模拟进行混合，得到最终的动画，为了移动起来更逼真，禁用掉脚部的物理模拟，以防止脚部与地面摩擦产生各种不好的效果。

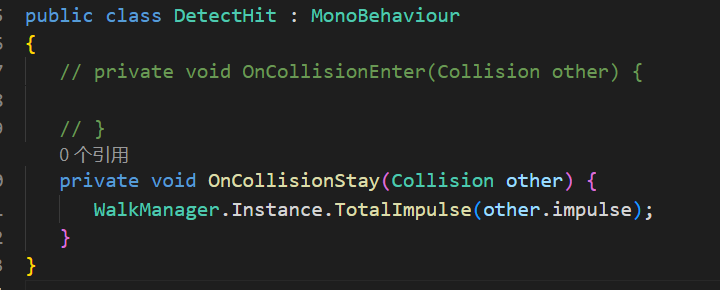
实现：

在碰撞体的父级创建好刚体和碰撞体，质量设置与全身质量相加一致。





要拿到被撞时的总冲量，要为全身碰撞体创建一个通过OnCollisionStay()方法拿碰撞冲量的方法，再将该冲量相加后传递给父物体上的刚体

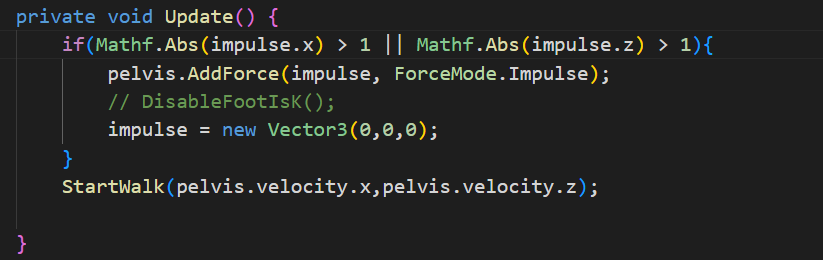


WalkManager脚本中有个TotalImpulse方法用于接收传递来的冲量并相加

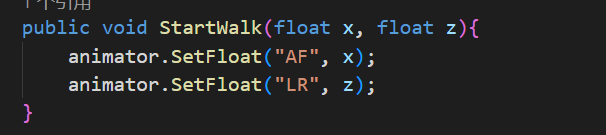
文本

描述已自动生成

由于碰撞开始的那一刻，物体就已经开始移动了，而不是等到碰撞结束，所以我在Update中分批次给冲量，即这一帧拿到多少冲量就给多少冲量，然后将使用后的冲量清零，下一帧使用下一帧拿到的冲量。



if(Mathf.Abs(impulse.x) > 1 || Mathf.Abs(impulse.z) > 1)的目的是出现一些微小的变化也会产生一些冲量，虽然这些冲量施加后不会有明显变化，但还是给它设置了一个范围，StartWalk(pelvis.velocity.x,pelvis.velocity.z);是定义了一个名为StartWalk的方法，用于更新动画混合树，pelvis.velocity.x和pelvis.velocity.z分别是刚体移动时的速度向量的x和z值



混合树的情况如下：

电脑萤幕画面

描述已自动生成

有五个动画，分别是向前后左右走和站着不动，其中

AF的值：为1向前走，为-1向后走

LR的值：为1向右走，为-1向左走

而刚体移动时的速度为一个Vector3类型的速度向量，x值对应AF，y值对应LR。

被撞后可能父刚体移动了，但是碰撞体和动画没有移动，是因为动画不会受影响，而碰撞体被动画拉了回来，所以需要一个同步的脚本，让移动的刚体的位置影响到最外层的对象，也就是它的父级，原理与子物体影响父物体旋转一样，都是通过添加中间层的方式实现

