

【Unity探究】物理碰撞实验

转载

huang9012 于 2014-04-06 16:26:13 发布 7918 收藏 4
分类专栏: [Unity3D](#)



[Unity3D 专栏收录该内容](#)

212 篇文章 7 订阅
订阅专栏

这几天为了准备面试，所以决定对平时学习中的盲点扫盲一下，首先想到的就是物理碰撞。以前没有好好研究过，一直模糊不清，到底什么条件下才可以产生物理碰撞呢？只要其中一个有Rigidbody就可以了吗？所以进行了下面的实验。

以下内容参考了[Unity圣典的组件手册](#)。

-----分割线-----

首先查找资料，发现Unity组件文档中将碰撞体分为了3个类型，如下所示。

Static Collider 静态碰撞器

These are GameObjects that do **not** have a Rigidbody attached, but **do** have a Collider attached. These objects should remain still, or move very little. These work great for your environment geometry. They will not move if a Rigidbody collides with them.

指的是没有附加刚体而附加了碰撞器的游戏对象。这类对象会保持静止或者很轻微的移动。对于环境模型十分好用，当和刚体碰撞时不会移动。

Rigidbody Collider 刚体碰撞器

These GameObjects contain both a Rigidbody and a Collider. They are completely affected by the physics engine through scripted forces and collisions. They might collide with a GameObject that only contains a Collider. These will likely be your primary type of Collider in games that use physics.

指的是同时附加了刚体和碰撞器的游戏对象。通过脚本的力量和碰撞完全受物理引擎的影响。可以和只包含碰撞器的游戏对象碰撞。将会成为你游戏中使用物理效果的基本类型碰撞器。

Kinematic Rigidbody Collider 运动学刚体碰撞器

This GameObject contains a Collider and a Rigidbody which is marked IsKinematic. To move this GameObject, you modify its Transform Component, rather than applying forces. They're similar to Static Colliders but will work better when you want to move the Collider around frequently. There are some other specialized scenarios for using this GameObject.

指的是同时包含碰撞器和刚体，并且激活IsKinematic的一类游戏对象，要移动这类游戏对象，要修改它的Transform组件（指的是position和rotation这类属性），而不是用力。它们很像静态碰撞器，不过如果你想要不停地到处移动碰撞器，它们会更好用。这类游戏对象还有许多其他的独特使用情景。

运动学刚体碰撞器是个非常懒惰和霸道的家伙，它碰到其他碰撞器完全没有反应（其实还是有点反应，会触发一些碰撞函数的，后面有解释）。它不受力、重力或扭矩的影响。可以通过设置Transform的position和rotation来准确的操作它们或者让它们动起来，但是，它们可以和其他的非运动学刚体互相作用。对于放在运动学刚体碰撞器上面的刚体，会受到运动学刚体施加的摩擦力。

还有一类叫做角色控制器。

Character Controllers 角色控制器

You use **Character Controllers** if you want to make a humanoid character. This could be the main character in a third person platformer, FPS shooter or any enemy characters.

如果想制作一个类似人的角色那就使用角色控制器。这可以是第三人称平台游戏、第一人称射击游戏的主要角色或任何敌对角色。

These Controllers don't follow the rules of physics since it will not feel right (in Doom you run 90 miles per hour, come to halt in one frame and turn on a dime). Instead, a Character Controller performs collision detection to make sure your characters can slide along walls, walk up and down stairs, etc.

这类控制器不遵循物理规则因此它感觉上不对劲（在Doom中，你跑到了90英里每小时，然后马上停下而且可以极快的转身）。不过，角色控制器执行碰撞检测以保证你的角色可以沿着墙滑动，上下台阶等等。

Character Controllers are not affected by forces but they can push Rigidbodies by applying forces to them from a script. Usually, all humanoid characters are implemented using Character Controllers.

角色控制器不受力影响（但是重力效果还是有的）但是可以被由代码施加的力推动。通常，所有类似人的角色都用角色控制器来执行。

Character Controllers are inherently unphysical, thus if you want to apply real physics - Swing on ropes, get pushed by big rocks - to your character you have to use a Rigidbody, this will let you use joints and forces on your character. Character Controllers are always aligned along the Y axis, so you also need to use a Rigidbody if your character needs to be able to change orientation in space (for example under a changing gravity). However, be aware that tuning a Rigidbody to feel right for a character is hard due to the unphysical way in which game characters are expected to behave. Another difference is that Character Controllers can slide smoothly over steps of a specified height, while Rigidbodies will not.

角色控制器本身不具物理特性，因此如果想应用真正的物理作用——在绳上摇摆，被大石头推动——到你的角色，必须用刚体，这会允许你把铰链或力用到你的角色上。角色控制器永远沿Y轴对齐，因此，如果你的角色需要在空间中改变方向那也必须用到刚体（比如在变化的引力控制下）。但是，要意识到调整一个刚体在角色上表现自然是很难的，因为游戏角色自身的非物理特性总想表现出来。另一个困难是，角色控制器可以从特定高度的台阶上平滑地滑下，而刚体不会。

The Controller does not react to forces on its own and it does not automatically push Rigidbodies away.

控制器不会对加在它自身上的力做出反应，也不会自动推开其他刚体。（这里我没有明白，因为我实验过在一个刚体内部实例化一个控制器，刚体是会被弹开的。如果有人懂请告诉我，不胜感激！）

If you want to push Rigidbodies or objects with the Character Controller, you can apply forces to any object that it collides with via the `OnControllerColliderHit()` function through scripting.

如果想让角色控制器推开其他刚体或者对象，你可以在对象附加的脚本中添加`OnControllerColliderHit()`函数，这样对它们施加力就能够产生碰撞。

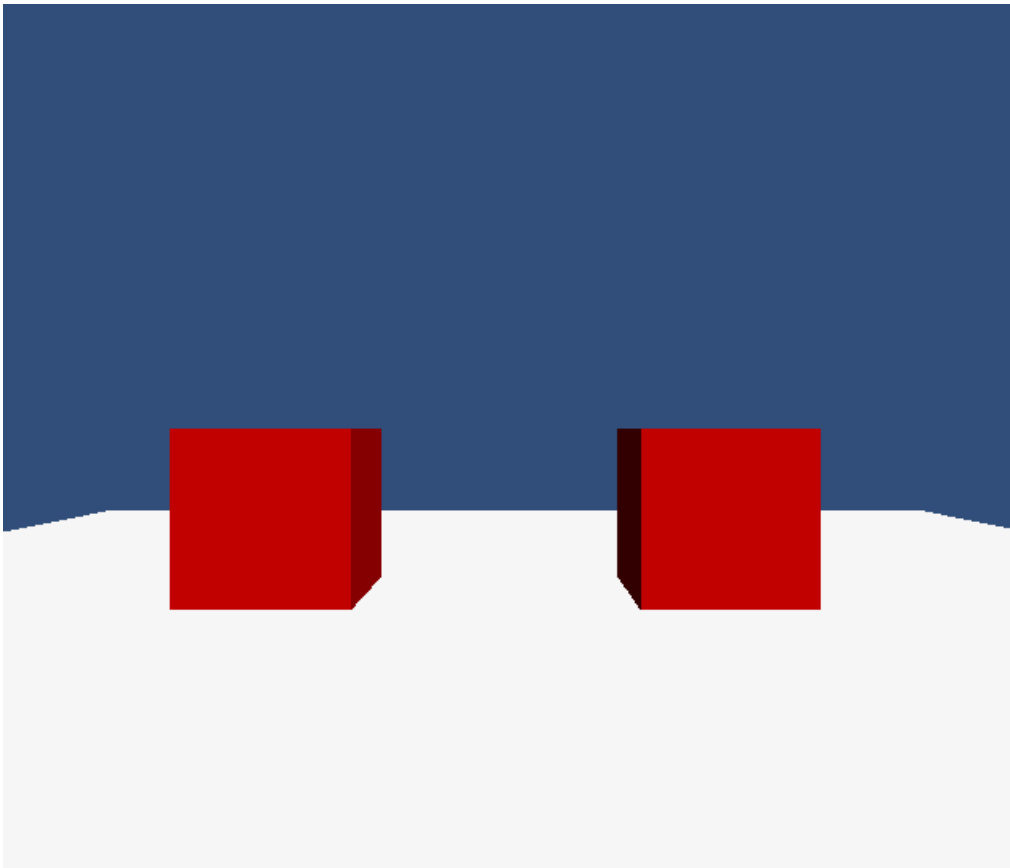
也就是说，角色控制器可以使用物理效果影响其他对象，前提是你自己写了脚本。

On the other hand, if you want your player character to be affected by physics then you might be better off using a **Rigidbody** instead of the Character Controller.

而角色控制器不能通过物理效果被其他对象影响。因此，如果你想让你的游戏角色被物理效果影响，那就最好使用刚体而不是角色控制器。

角色控制器和运动学刚体有点像，因为它们都不受力的作用（运动学刚体更像是一个刚体，只是不会受到其他任何外力的影响，就像是古代的国王，虽然和普通平民一样都是人，但是只能他欺负别人，别人无法欺负他；而角色控制器本身就不具有物理特性，不可以推开别人（除非写了自己的脚本），也不会被别人影响，只是它可以执行碰撞检测）。但是也有几点不同的地方。首先，运动学刚体是无法通过代码施加力的作用来推动，但是角色控制器是可以的。第二，运动学刚体不会受到重力的作用，而角色控制器是有重力效果的。其次，角色控制器只能沿Y轴旋转，也就是人永远都是直立状态，不会因为受到力的作用就趴下啦，而刚体是可以的。第三，角色控制器可以从特定高度台阶上平滑地滑下，而刚体不会。

说了这么多，晕了吧，恩，我也晕了。所以我们来做下实验！我们通过一个最简单的场景，即一个平行光、一个平面、两个正方体来试验一下。通过给两个正方体添加不同的碰撞体类型来观察是否会发生碰撞。



下面的实验中，一个正方体保持静止，另一个正方体我们通过代码来控制运动，即上下左右方向键可以控制它的上下左右运动。

移动的静态碰撞器VS静止的静态碰撞器

没有任何碰撞效果。这在意料之中，因为没有一个是添加了Rigidbody属性，肯定不会碰撞啦！

移动的静态碰撞器VS静止的刚体碰撞器

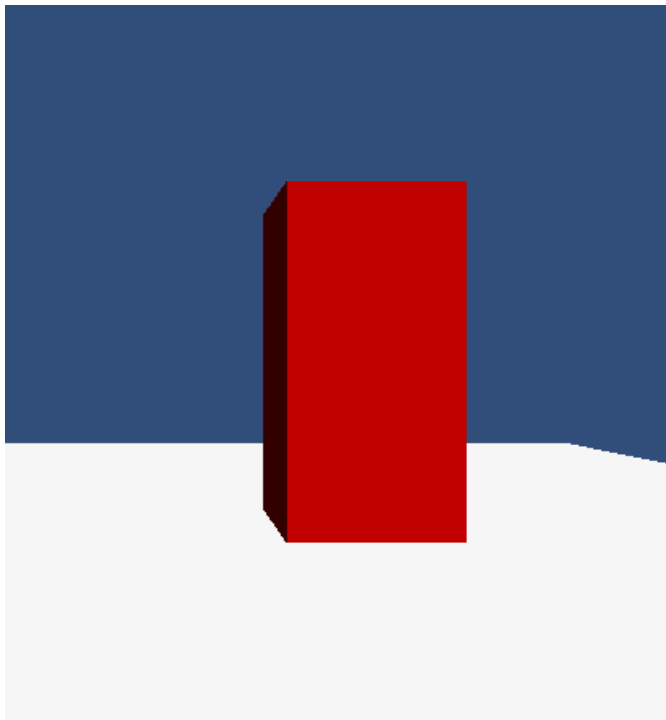
没有任何碰撞效果。

这个结果有点意外，明明一个已经添加了Rigidbody，为什么还是无法发生碰撞呢？这是因为添加了Rigidbody的正方体是静止的，而静止的刚体会进入休眠。好吧，又遇到了新的名词，刚体休眠。例如，当一个正方体掉到地板上静止后就会进入刚体休眠。刚体休眠完全自动发生。只要刚体的速度低于sleepAngularVelocity和sleepVelocity，该刚体就会开始休眠。其空闲一些帧后，就会被设置成休眠状态。处于休眠状态中的物体，不会再对其进行碰撞检测和模拟。这会节约大量的CPU开销。这就解释了为什么我们这个实验里没有碰撞发生，因为它睡觉去了，而那个移动的静态碰撞器又无法唤醒它，所以它压根就不知道有人碰到了自己。这里又有一个问题了，那么什么情况下可以唤醒已经休眠了的刚体呢？有4中情况：

- 被施加了外力，也就是说在代码里使用了AddForce；

- 刚体的属性发生了变化；
- 和它通过关节连接的刚体发生了移动，因为连带关系，所以它也被唤醒了；
- 被其他刚体碰撞器碰撞。但是，**只有处于运动状态中的刚体（包括普通的刚体碰撞器和运动学刚体碰撞器）能唤醒休眠中的刚体，静态碰撞器不能唤醒休眠中的刚体**。即如果你将一个静态碰撞器(没有和任何刚体绑定过)绑定到处于休眠中的刚体，或者将静态碰撞器抽离处于休眠中的刚体时，这个休眠中的刚体是不会被唤醒的。但是如果你将一个运动中的刚体从它所属的且处于休眠状态中的顶层刚体移出时，处于休眠中的刚体将会被唤醒，且会在图形更新里被重新正确的计算。

最后这句话时什么意思呢？我们再做两个实验好了。还是两个正方体，我们将一个正方体放在另一个的上方。



在两个实验里，上方的正方体都是普通的刚体碰撞器，而下方的正方体有所不同，我们可以通过代码移动下方的正方体使得它从上方正方体的下面移开。

第一个实验里，下方的正方体是一个静态碰撞器。当我们移开后，上方的正方体直直地下落，直到碰到地板，期间没有任何物理效果（除了重力效果）。

第二个实验里，下方的正方体是一个运动学刚体碰撞器。当我们移开下方正方体的过程中，我们会发现上方的正方体会受到摩擦力的左右而也发生了一定移动，当两个正方体完全分离后，上方的正方体晃动着掉到了地板上（因为之前受到了摩擦力）。

好了，有点明白了吗？静态碰撞器是不会唤醒休眠刚体的（第一个实验），而运动学刚体由于会对刚体产生力的作用而可以唤醒休眠中的刚体。

再提一个问题，如果上面的正方体是运动学刚体碰撞器呢？那么无论下面是什么碰撞器它都没有反应，因为它是国王啦，不受到任何外力的影响，包括重力！

移动的刚体碰撞器VS静止的静态碰撞器

有碰撞效果，具体为：对于移动的刚体碰撞器在遇到静态碰撞器后会产生物理效果，如反弹、无法前进等；对于静止的静态碰撞器，会产生轻微的碰撞效果，如轻微的晃动，但是不会移动。

移动的刚体碰撞器VS静止的刚体碰撞器

有碰撞效果，对于两个物体都出现正常的物理效果，如反弹等。

移动的静态碰撞器VS静止的运动学刚体碰撞器

没有任何碰撞效果。运动学刚体碰撞器是不受到任何外力作用的。

移动的运动学刚体碰撞器VS静止的静态碰撞器

没有任何碰撞效果。运动学刚体碰撞器是不受到任何外力作用的。

移动的运动学刚体碰撞器VS静止的刚体碰撞器

有碰撞效果，但仅刚体碰撞器有碰撞效果，如反弹等，运动学刚体碰撞器不受力的作用。

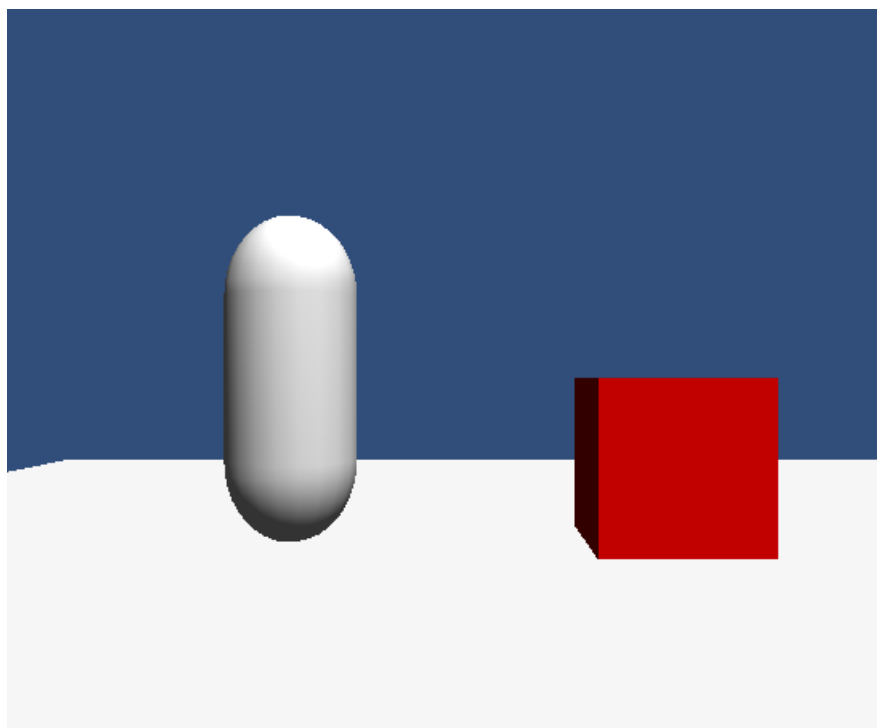
移动的刚体碰撞器VS静止的运动学刚体碰撞器

有碰撞效果，但仅刚体碰撞器有碰撞效果，如反弹等，运动学刚体碰撞器就像一堵墙一样一动不动。

移动的运动学刚体碰撞器VS静止的运动学刚体碰撞器

两个太上皇相见了会怎么样呢？没有任何反应.....就像两个静态碰撞器一样。

下面，将一个正方体换成角色控制器，继续实验。



移动的静态碰撞器VS静止的角色控制器

没有任何碰撞效果。

移动的刚体碰撞器VS静止的角色控制器

有碰撞效果，刚体碰撞器遇到角色控制器后会发生反弹，无法继续前进，而角色控制器没有任何效果。

移动的运动学刚体碰撞器VS静止的角色控制器

没有任何碰撞效果，会互相穿过。

移动的角色控制器VS静止的静态碰撞器

没有任何碰撞效果，会互相穿过。

移动的角色控制器VS静止的刚体碰撞器

有碰撞效果，刚体碰撞器被弹开。

移动的角色控制器VS静止的运动学刚体碰撞器

没有任何碰撞效果，会互相穿过。

碰撞信息和触发信息

对于碰撞时是否会发出碰撞函数（或触发信息），可以见下表。碰撞信息是指OnCollisionEnter() OnCollisionStay()和 OnCollisionExit()这三个函数，而触发信息指的是OnTriggerEnter() OnTriggerStay和OnTriggerExit()三个函数。

Collision detection occurs and messages are sent upon collision 碰撞后有碰撞检测并有碰撞信息发出						
	Static Collider 静态碰撞器	Rigidbody Collider 刚体碰撞器	Kinematic Rigidbody Collider 运动学刚体碰撞器	Static Trigger Collider 静态触发碰撞器	Rigidbody Trigger Collider 刚体触发碰撞器	Kinematic Rigidbody Trigger Collider 运动学刚体触发碰撞器
Static Collider 静态碰撞器		Y				
Rigidbody Collider 刚体碰撞器	Y	Y	Y			
Kinematic Rigidbody Collider 运动学刚体碰撞器		Y				
Static Trigger Collider 静态触发碰撞器						
Rigidbody Trigger Collider 刚体触发碰撞器						
Kinematic Rigidbody Trigger Collider 运动学刚体触发碰撞器						

Trigger messages are sent upon collision 碰撞后有触发信息						
---	--	--	--	--	--	--

	Static Collider 静态碰撞器	Rigidbody Collider 刚体碰撞器	Kinematic Rigidbody Collider 运动学刚体碰撞器	Static Trigger Collider 静态触发碰撞器	Rigidbody Trigger Collider 刚体触发碰撞器	Kinematic Rigidbody Trigger Collider 运动学刚体触发碰撞器
Static Collider 静态碰撞器					Y	Y
Rigidbody Collider 刚体碰撞器				Y	Y	Y
Kinematic Rigidbody Collider 运动学刚体碰撞器				Y	Y	Y
Static Trigger Collider 静态 触发碰撞器		Y	Y		Y	Y
Rigidbody Trigger Collider 刚体触发碰撞器	Y	Y	Y	Y	Y	Y
Kinematic Rigidbody Trigger Collider 运动学刚体触发碰撞器	Y	Y	Y	Y	Y	Y

根据脚本手册里写的，若要能够接收触发信息，两个碰撞器中必有一方含有刚体而至少一方是触发器。上表也不一定完全正确，在我的经验里，那些意外状况大概是因为刚体休眠造成系统没有检测。例如，上表中的第四行，当移动的静态触发碰撞器依次和静止的刚体碰撞器、运动学刚体碰撞器、刚体触发碰撞器、运动学刚体触发碰撞器碰撞时，实际上只有刚体碰撞器的情况下才发送了触发信息，原因我也不是很清楚，我想可能是因为刚体休眠吧，因为静态触发碰撞器并不能唤醒进入休眠的刚体，因此就会进行碰撞检测。