Box2D详解2 碰撞筛选

DioysosLai 2014-5-20

相信玩过游戏的人都知道，己方英雄释放的技能只能对敌方英雄有伤害，而不能伤害到己方英雄（要是能伤害到自己人，那不是坑爹啊！）在一些小游戏中，如果涉及到这些设置，一方面我们可以自己进行碰撞过滤，另一方面，如果我们使用到了Box2D，那么，我们就可以使用Box2D自带的碰撞过滤（Collision filtering），通过三个标志位实现碰撞过滤：

类别标志位(categoryBits)、遮罩标志位(maskBits)、分组索引(groupIndex)。

**类别标志位和遮罩标志位解释：**

**所谓类别标志位，就是指明对象类型，即告诉别人“我是谁”。所谓遮罩标志位，就是致命碰撞对象，即告诉别人“我想和谁发生碰撞”。**

例如《猫抓老鼠》类型游戏，猫的类型就是“猫”，遮罩标志位就是“猫|老鼠”。老鼠类型是“老鼠”，其遮罩标志位就是“猫|老鼠”。这样，猫和老鼠可以互相发生碰撞。但，有一点要指明就是，二者的碰撞必须是“你情我愿”。如果猫的遮罩标志位是“猫|老鼠”，即猫想和老鼠碰撞，但如果老鼠的遮罩标志位是“老鼠”（注意：这里没有“猫”），即老鼠不想和猫发生碰撞，那么老鼠和猫最终结果是不能发生碰撞的。

具体来说，我们可以用一段代码来表示这种“你情我愿”的关系：

bool collide = (filterCat.maskBits & filterMouse.categoryBits) != 0 &&

(filterCat.categoryBits & filterMouse.maskBits) != 0;

在默认情况下,类别标志位值为0x0001，遮罩标志位默认值为0xFFFF，即默认，我们创建的任何对象，都是“I am the god， I can collision with everything！”。如果，所以对象我们都没改变其类别标志位和遮罩标志位，那么，所有对象默认都是可以互相碰撞的。

**如何设置类别标志位和遮罩标志位：**

每一个标志位是一个16位的整数，因此我们可以设置16种不同的碰撞类型。具体方法，可以先设置我们要设置物理世界对象类型：

enum CABOX2DMASK

{

BOX2D\_MASK\_WORD = 0X0001, ///< 世界

BOX2D\_MASK\_BALL = 0x0002, ///< 球

BOX2D\_MASK\_BASKET = 0x0004, ///< 球框

… = 0x0008, ///< …

…

…

};

然后，我们在初始化我们世界对象时，可以更根据要求来设置对象的类别标志位和遮罩标志位。

例如如下代码：

// Create ball body and shape 创建球的body

b2Body \*body;

b2BodyDef ballBodyDef;

ballBodyDef.type = b2\_dynamicBody; ///< 指定body的类型为dynamic body。默认值是static body，那意味着那个body不能被移动也不会参与仿真。

ballBodyDef.position.Set(BALLPOSITION[m\_baBall[i].size].x/PTM\_RATIO, BALLPOSITION[m\_baBall[i].size].y/PTM\_RATIO);

ballBodyDef.userData = m\_baBall[i].ball; ///< 设置body的user data属性为篮球精灵。可以设置任何东西，但是，你设置成精灵会很方便，特别是当两个body碰撞的时候，你可以通过这个参数把精灵对象取出来，然后做一些逻辑处理。

body = m\_world->CreateBody(&ballBodyDef);

b2CircleShape circle;

circle.m\_radius = m\_baBall[i].ball->getWidth()/2.0f/PTM\_RATIO;

b2FixtureDef ballShapeDef;

ballShapeDef.shape = &circle;

**ballShapeDef.filter.categoryBits = BOX2D\_MASK\_BALL;**

**ballShapeDef.filter.maskBits = BOX2D\_MASK\_WORD | BOX2D\_MASK\_BALL ;**

ballShapeDef.density = 1.0f; ///< 单位体积的质量（密度）。因此，一个对象的密度越大，那么它就有更多的质量，当然就会越难以移动.

ballShapeDef.friction = 0.2f; ///< 就是摩擦力。它的范围是0-1.0， 0意味着没有摩擦，1代表最大摩擦，几乎移不动的摩擦。

ballShapeDef.restitution = 0.5f; ///< 回复力。它的范围也是0到1.0. 0意味着对象碰撞之后不会反弹，1意味着是完全弹性碰撞，会以同样的速度反弹。

body->CreateFixture(&ballShapeDef);

这里我们设置我们的对象为“我是一个**BOX2D\_MASK\_BALL**”, 我可以和“**BOX2D\_MASK\_WORD | BOX2D\_MASK\_BALL ”**发生碰撞。

**如何动态改变类别标志位和遮罩标志位：**

在很多情况下，我们的世界对象达到某个条件时，我们要改变其类别标志位和遮罩标志位。那么，我们要首先获取这个世界对象的内容，然后在该别其值。具体方法，可以参考如下：

for (b2Body \*b = m\_world->GetBodyList(); b; )

{

///遍历world对象里面的所有body，然后看body的user data属性是否为空，如果不为空，就可以强制转换成精灵对象。接下来，就可以根据body的位置来更新精灵的位置了。

if (b->GetUserData() != NULL)

{

b2Body\* b2node = b;

b = b2node->GetNext();

boxSprite\* ballData = (boxSprite\*)b2node->GetUserData();

if (ballData->getDead()) ///< 改变对象类别、遮罩标志位

{

**b2Fixture\* fixture = b2node->GetFixtureList();**

**b2Filter filter = fixture->GetFilterData();**

**//change whatever you need to, eg.**

**filter.categoryBits = BOX2D\_MASK\_BASKET;**

**filter.maskBits = BOX2D\_MASK\_WORD | BOX2D\_MASK\_BASKET ;**

**fixture->SetFilterData(filter);**

ballData->setDead(false);

}

ballData->setPosition(ccp(b2node->GetPosition().x\*PTM\_RATIO,

b2node->GetPosition().y\*PTM\_RATIO));

ballData->setRotation(-1\*CC\_RADIANS\_TO\_DEGREES(b2node->GetAngle()));

}

else

{

b = b->GetNext();

}

}

具体，类别标志位和遮罩标志位的功能，就差不多是这样了。

分组索引：

分组索引优先于类别和遮罩标志位，具体作用如下：

-定制器如果有分组索引为零，那么使用分类/遮罩规则   
-如果两个分组索引的值不为零并且不相同，那么使用分类/遮罩规则   
-如果两个分组索引的值相同，并且为正数，则产生碰撞   
-如果两个分组索引的值相同，并且为负数，则不产生碰撞

我们设置对象时，对象的分组索引默认为0。

好的，Box2D的碰撞过滤，就差不多是这样的。希望对大家有点用处。下面是我的qq：906391500，欢迎大家一起交流，一起成长。