

```
#include "TestScene.h"
```

```
USING_NS_CC;
```

```
Scene* Test::createScene()
```

```
{
```

```
    auto scene = Scene::createWithPhysics();
```

```
    scene->getPhysicsWorld()->setGravity(Vect(0,0));
```

```
    auto layer = Test::create();
```

```
    scene->getPhysicsWorld()->setDebugDrawMask(PhysicsWorld::DEBUGDRAW_ALL);
```

```
    scene->addChild(layer);
```

```
    #if COCOS2D_DEBUG
```

```
    scene->getPhysicsWorld()->setDebugDrawMask(PhysicsWorld::DEBUGDRAW_ALL);
```

```
    #endif
```

```
    return scene;
```

```
}
```

```
bool Test::init()
```

```
{
```

```
    if ( !Layer::init() )
```

```
    {
```

```
        return false;
```

```
    }
```

```
    _director = Director::getInstance();
```

```
    _visibleSize = _director->getVisibleSize();
```

```
auto origin = _director->getVisibleOrigin();
```

```
auto enemy = Sprite::create("enemy.png");
```

```
enemy->setPosition(_visibleSize.width/2, _visibleSize.height - ((enemy->getContentSize().height/2) * 1.2));
```

```
setPhysicsBody(enemy);
```

```
this->addChild(enemy, 1);
```

```
auto testCollide = MoveTo::create(2, Vec2(enemy->getPositionX(), _visibleSize.height/2));
```

```
enemy->runAction(testCollide);
```

```
auto cube = Sprite::create("cube.png");
```

```
cube->setPosition(_visibleSize.width/2, _visibleSize.height/2);
```

```
setPhysicsBody(cube);
```

```
this->addChild(cube, 0);
```

```
Vector<SpriteFrame*> frames;
```

```
Size playerSize = cube->getContentSize();
```

```
frames.pushBack(SpriteFrame::create("cube.png", Rect(0, 0, playerSize.width, playerSize.height)));
```

```
frames.pushBack(SpriteFrame::create("cube2.png", Rect(0, 0, playerSize.width, playerSize.height)));
```

```
frames.pushBack(SpriteFrame::create("cube3.png", Rect(0, 0, playerSize.width, playerSize.height)));
```

```
frames.pushBack(SpriteFrame::create("cube4.png", Rect(0, 0, playerSize.width, playerSize.height)));
```

```
auto cubeAnimation = Animation::createWithSpriteFrames(frames, 0.5f);
```

```
auto animate = Animate::create(cubeAnimation);
```

```
cube->runAction(RepeatForever::create(animate));

initPhysics();

return true;
}

void Test::setPhysicsBody(cocos2d::Sprite* sprite)
{
    auto body = PhysicsBody::createCircle(sprite->getContentSize().width/2);
    body->setContactTestBitmask(true);
    body->setDynamic(true);
    sprite->setPhysicsBody(body);
}

bool Test::onCollision(PhysicsContact& contact)
{
    enemy->setVisible(false);
    return false;
}

void Test::initPhysics()
{
    auto contactListener = EventListenerPhysicsContact::create();
    contactListener->onContactBegin = CC_CALLBACK_1(Test::onCollision, this);
    getEventDispatcher()->addEventListenerWithSceneGraphPriority(contactListener, this);
}
```