

# 现代操作系统应用开发实验报告

学号： 15331248

班级： 上午班

姓名： 潘承远

实验名称： Cocos2d(HW14)

## 一.参考资料

- [http://blog.csdn.net/tonny\\_guan/article/details/39584055](http://blog.csdn.net/tonny_guan/article/details/39584055) 了解碰撞检测
- <http://www.cocoachina.com/bbs/read.php?tid-236333-page-1.html> 了解 onContactBegin 多次调用问题
- 《物理引擎与粒子系统》这是老师给的 PPT,了解了物理引擎、粒子系统
- <https://my.oschina.net/soarwilldo/blog/487942> 了解刚体的几个有用的函数如何使用
- 《hw14》这是 TA 给的 PPT，了解了本次作业的要求以及需要注意的几个点

## 二.实验步骤

- 仔细阅读课件与相关文档
- 仔细阅读已经给的代码，明确思路
- 完成基本要求“至少使用一种粒子系统”

下面是关键代码：

```
//添加粒子系统
auto paopao = ParticleSystemQuad::create("particle_texture2.plist");
paopao->setPositionType(ParticleSystemQuad::PositionType::RELATIVE);
paopao->setPosition(visibleSize.width / 2, visibleSize.height / 2);
paopao->setScale(0.6f);
paopao->setScaleX(1.0f);
this->addChild(paopao, 1);
```

- 完成基本要求“为玩家和箱子设置刚体属性”

下面是关键代码：

```
// 设置角色刚体属性
auto playerbody = PhysicsBody::createBox(player1->getContentSize(), PhysicsMaterial(10.0f, 0.0f, 1.0f));
playerbody->setCategoryBitmask(0x0000FFFF);
playerbody->setCollisionBitmask(0x0000FFFF);
playerbody->setContactTestBitmask(0x0000FFFF);
player1->setPhysicsBody(playerbody);
```

```
// 为箱子设置刚体属性
auto boxbody = PhysicsBody::createBox(box->getContentSize(), PhysicsMaterial(10.0f, 0.0f, 1.0f));
boxbody->setCategoryBitmask(0xFFFF0000);
boxbody->setCollisionBitmask(0xFFFF0000);
boxbody->setContactTestBitmask(0xFFFF0000);
boxbody->setTag(++counter);
box->setPhysicsBody(boxbody);
```

- 完成基本要求 “箱子下落过程过程能穿透玩家，不会发生碰撞。”

我将箱子的掩码都设置为“0xFFFF0000”

将角色的掩码设置为“0x0000FFFF”

可知掩码按位与为 0，将会忽略掉碰撞，所以箱子下落过程过程能穿透玩家，不会发生碰撞

- 完成基本要求 “箱子碰到船或者碰到其他箱子之后，能和玩家发生碰撞”

下面是关键代码：

```
bool FriendShip::onConcactBegin(PhysicsContact & contact) {
    auto BodyA = contact.getShapeA()->getBody();
    auto BodyB = contact.getShapeB()->getBody();
    if (BodyA->getCategoryBitmask() != 0x0000FFFF && BodyB->getCategoryBitmask() != 0x0000FFFF) {
        BodyA->setCategoryBitmask(0x0000FFFF);
        BodyA->setCollisionBitmask(0x0000FFFF);
        BodyA->setContactTestBitmask(0x0000FFFF);
        BodyB->setCategoryBitmask(0x0000FFFF);
        BodyB->setCollisionBitmask(0x0000FFFF);
        BodyB->setContactTestBitmask(0x0000FFFF);
    }
    return true;
}
```

当发生碰撞时，将 A、B 的掩码设置为相同，这样就能够完成要求

- 完成基本要求 “控制玩家左右移动”

下面是关键代码：

```
auto position = player1->getPosition();
if (IsPlayer1Left && leftmove) {
    auto moveto = MoveTo::create(0.15f, Vec2(position.x - 1, position.y));
    player1->runAction(moveto);
    player1->setFlippedX(true);
    if (holdbox1 != NULL) {
        auto position1 = holdbox1->getPosition();
        auto movet = MoveTo::create(0.15f, Vec2(position1.x - 1, position1.y));
        holdbox1->runAction(movet);
    }
}

if (IsPlayer1Right && rightmove) {
    auto moveto = MoveTo::create(0.15f, Vec2(position.x + 1, position.y));
    player1->runAction(moveto);
    player1->setFlippedX(false);
    if (holdbox1 != NULL) {
        auto position1 = holdbox1->getPosition();
        auto movet = MoveTo::create(0.15f, Vec2(position1.x + 1, position1.y));
        holdbox1->runAction(movet);
    }
}
```

- 完成基本要求 “使用关节举起和扔下箱子”

下面是关键代码：

```

    if (IsPlayer1Hold && holdbox1 == NULL) {
        auto position = player1->getPosition();
        for (list<PhysicsBody*>::iterator it = boxes.begin(); it != boxes.end(); it++) {
            if (player1->getBoundingBox().intersectsRect((*it)->getNode()->getBoundingBox())) {
                joint1 = PhysicsJointDistance::construct(player1->getPhysicsBody(), (*it),\
                    player1->getAnchorPoint(), (*it)->getNode()->getAnchorPoint());
                m_world->addJoint(joint1);
                holdbox1 = (*it)->getNode();
                auto rotate = holdbox1->getRotation();
                auto rotateby = RotateBy::create(0.01f, 360 - rotate);
                holdbox1->runAction(rotateby);
                holdbox1->setPosition(position.x, position.y + 35);
                player1->setSpriteFrame(IdleWithBox1);
                break;
            }
        }
    } // end if IsPlayer1Hold
    else if (!IsPlayer1Hold && holdbox1 != NULL) {
        auto pos = holdbox1->getPhysicsBody();
        if (LastPlayer1Press == 'D')
            holdbox1->getPhysicsBody()->applyImpulse(Vec2(1500000, 1500000), Vec2(0, 0));
        else if (LastPlayer1Press == 'A')
            holdbox1->getPhysicsBody()->applyImpulse(Vec2(-1500000, 1500000), Vec2(0, 0));
        holdbox1 = NULL;
        player1->setSpriteFrame(frame1);
        m_world->removeAllJoints();
    }
}

```

- 完成加分项 “实现人物跳跃”

下面是关键代码：

```

case cocos2d::EventKeyboard::KeyCode::KEY_UP_ARROW:
    // 跳
    player1->getPhysicsBody()->applyImpulse(Vec2(0, 5000000), Vec2(0, 0));
    // Todo
    break;

```

- 完成加分项 “举着箱子的情况下跑动和跳跃”

由于左右移动的时候 holdbox1 会随着 player1 左右移动，并且 player1 跳跃的时候是向上施加一个力，所以该加分项在前面的基础上得以完成

- 完成加分项 “轮船倾斜以及翻船”

下面是关键代码：

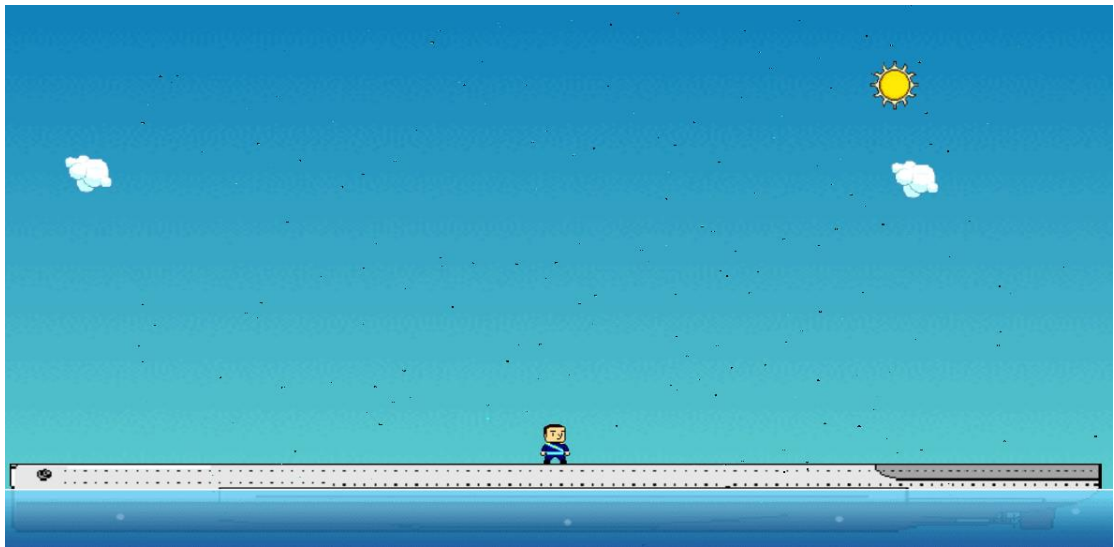
```

void FriendShip::updateShip(float dt) {
    list<PhysicsBody*>::iterator it = boxes.begin();
    int rotate = 0;
    for (; it != boxes.end(); it++) {
        list<PhysicsBody*>::iterator temp = it;
        temp++;
        if (temp == boxes.end() && LastTag != (*it)->getTag()) {
            rotate = ((*it)->getNode()->getPosition().x - visibleSize.width / 2) / 100;
            LastTag = (*it)->getTag();
            break;
        }
    }
    ship->runAction(RotateBy::create(0.1f, rotate));
}

```

### 三.实验结果截图

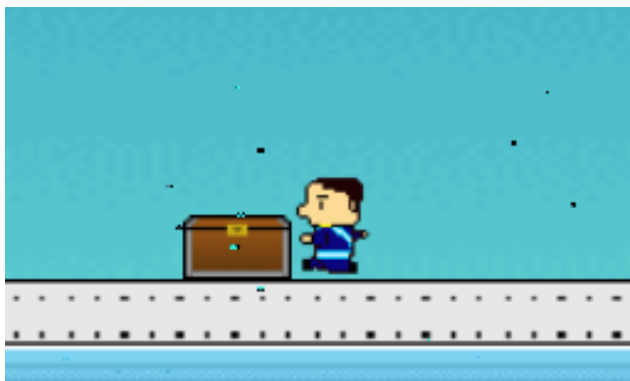
- 基本要求 “至少使用一种粒子系统”



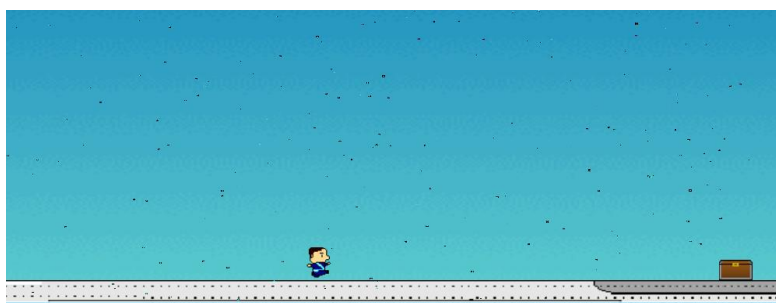
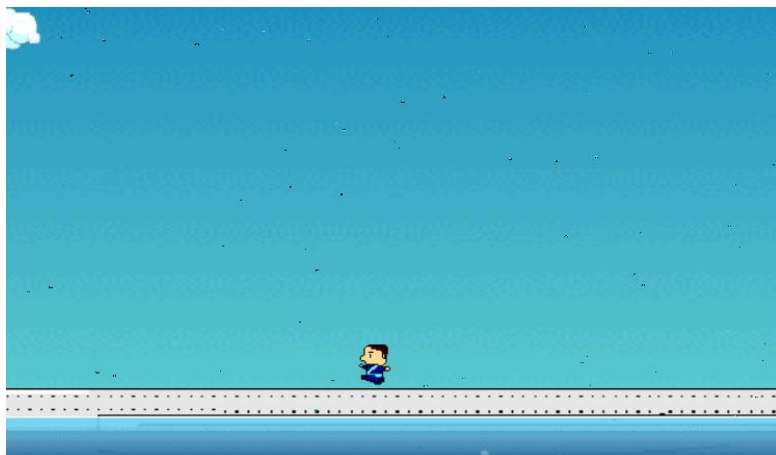
- 基本要求 “为玩家和箱子设置刚体属性”、基本要求 “箱子下落过程过程能穿透玩家，不会发生碰撞。”  
箱子直接穿过刚体小人



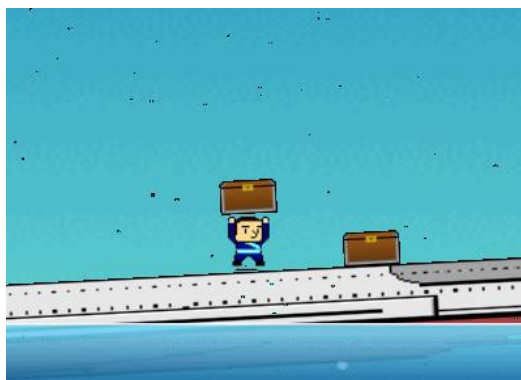
- 基本要求 “箱子碰到船或者碰到其他箱子之后，能和玩家发生碰撞”  
这时候刚体小人不能够再移动，箱子可以和刚体小人发生碰撞



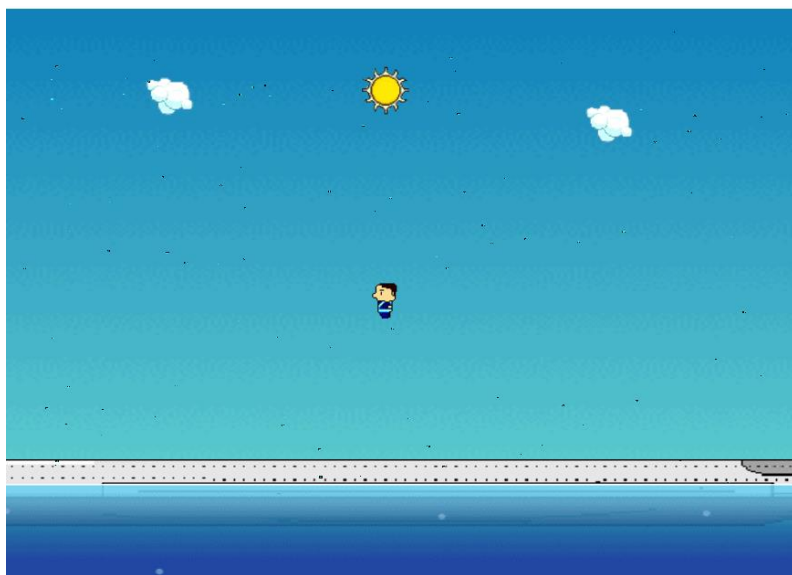
- 基本要求 “控制玩家左右移动”



- 基本要求 “使用关节举起和扔下箱子”



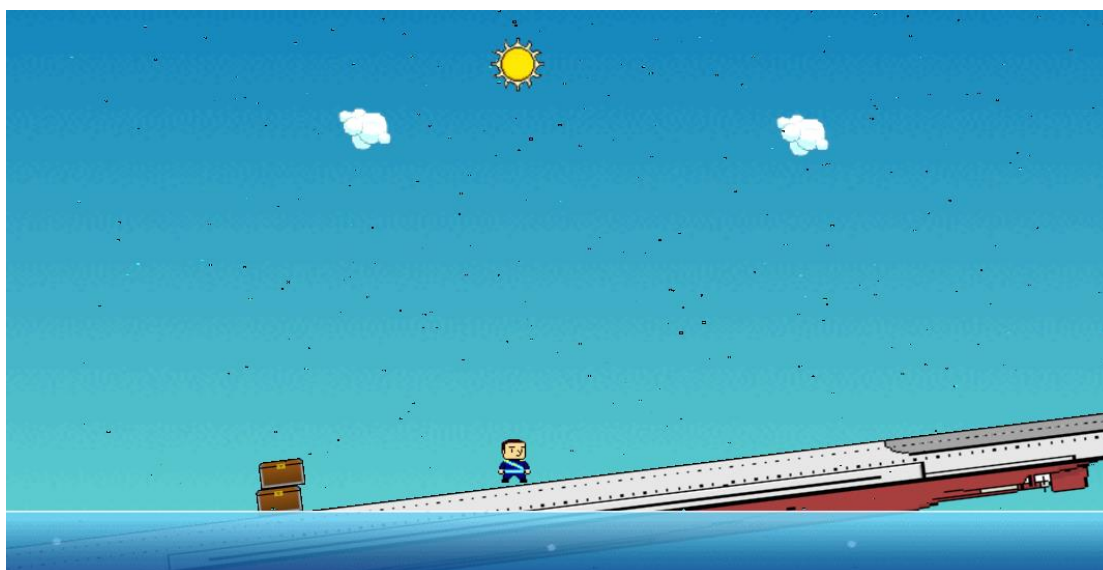
- 加分项 “实现人物跳跃”



- 加分项 “举着箱子的情况下跑动和跳跃”



- 加分项 “轮船倾斜以及翻船”



#### 四.实验过程遇到的问题

- 就是之前玩家碰到箱子的时候，玩家如果此时继续向前走，那么将会推动箱子，然而 TA 给的 demo 是当玩家碰到箱子的时候玩家不能继续推动箱子向前走，只能够跳跃或举起箱子，于是我将代码修改为如下，成功解决。

```
bool leftmove = true;
bool rightmove = true;
for (list<PhysicsBody*>::iterator it = boxes.begin(); it != boxes.end(); it++) {
    if (player1->getBoundingBox().intersectsRect((*it)->getNode()->getBoundingBox())) {
        if (holdbox1 != NULL && holdbox1->getPhysicsBody()->getTag() != (*it)->getTag()) {
            if ((*it)->getPosition().x < player1->getPhysicsBody()->getPosition().x) leftmove = false;
            else rightmove = false;
        }
        if (holdbox1 == NULL) {
            if ((*it)->getPosition().x < player1->getPhysicsBody()->getPosition().x) leftmove = false;
            else rightmove = false;
        }
    }
}
```

Leftmove 用于判断是否还能够向左移动，rightmove 用于判断是否还能够向右移动



- 玩家举起箱子后向左或者向右走动，如果此时碰到箱子，刚体小人会“摔倒”，这是因为游戏模仿了现实当中的受力情况，TA 给的 demo 是刚体小人任何时候都不会“摔倒”，于是我在 update 函数中添加了如下的代码，成功解决：

```
player1->getPhysicsBody()->addMoment(99999999999999999999);
```

- 之前一直不知道如何才能够“放下”箱子，也就是  $45^\circ$  角方向施加一个力。通过查资料，学会了 applyImpulse 函数的调用。

- 之前停止动画的时候只使用了 stopAllActionByTag() 这一个函数，然后发现之后再也不会有这样的动画出现。通过查资料知道这是因为调用该函数之后会销毁掉所有的动画，所以必须在后面添加 setSpriteFrame();

```
// 停止动画和运动
player1->stopAllActionsByTag(11);
player1->setSpriteFrame(frame1);
// Todo
break;
```

- 船的倾斜问题，查了很多资料还是没有找到想要的方法。于是自己只能使用旋转动画，通过判断船左右的箱子的数量来决定船倾斜的角度。

## 五.思考与总结

本次实验让自己了解了物理世界的一些基本概念，也学会用代码来实现物理世界。此外粒子系统相对于物理世界学起来说要简单很多。虽然本次实验没能够非常好地完成船的旋转，但是基本要求都能够较好完成，这也让我体会到了物理引擎的魅力。