

# 现代操作系统应用开发实验报告

学号：15331418

班级：晚上班

姓名：郑柏川 实验名称：物理引擎与粒子系统

## 一. 参考资料

Ppt

[http://blog.csdn.net/tonny\\_guan/article/details/39267411](http://blog.csdn.net/tonny_guan/article/details/39267411)

<http://blog.csdn.net/w18767104183/article/details/39249557>

<http://blog.csdn.net/yuzhiboyi/article/details/38498567>

## 二. 实验步骤

### 1. 至少使用一种粒子系统：

上网搜了一下，自带的有这几种，然后用了烟花粒子效果（最好看的一个）

内置的11粒子系统说明如下：

ParticleExplosion。爆炸粒子效果，属于半径模式。

ParticleFire。火焰粒子效果，属于重力径模式。

ParticleFireworks。烟花粒子效果，属于重力模式。

ParticleFlower。花粒子效果，属于重力模式。

ParticleGalaxy。星系粒子效果，属于半径模式。

ParticleMeteor。流星粒子效果，属于重力模式。

ParticleSpiral。漩涡粒子效果，属于半径模式。

ParticleSnow。雪粒子效果，属于重力模式。

ParticleSmoke。烟粒子效果，属于重力模式。

ParticleSun。太阳粒子效果，属于重力模式。

ParticleRain。雨粒子效果，属于重力模式。

```
auto PF = ParticleFireworks::create();
PF->setPosition(player1->getContentSize().width / 2, player1->getContentSize().height / 2);
PF->setScale(0.5f);
player1->addChild(PF, -1);
```

## 2. 为玩家和箱子设置刚体属性

参考 ppt 的

```
auto playerbody = PhysicsBody::createBox(player1->getContentSize(), PhysicsMaterial(1.0f, 0.0f, 1.0f));
playerbody->setCategoryBitmask(0x0000FFFF);
playerbody->setCollisionBitmask(0xFFFFFFFF);
playerbody->setContactTestBitmask(0xFFFFFFFF);
playerbody->setDynamic(true);
playerbody->setRotationEnable(false);
player1->setPhysicsBody(playerbody);
```

```
auto boxbody = PhysicsBody::createBox(box->getContentSize(), PhysicsMaterial(100.0f, 0.0f, 1.0f));
boxbody->setCategoryBitmask(0xFFFFFFFF);
boxbody->setCollisionBitmask(0xFFFF0000);
boxbody->setContactTestBitmask(0xFFFFFFFF);
boxbody->setDynamic(true);
boxbody->setRotationEnable(false);
box->setPhysicsBody(boxbody);
```

改变掩码

```
auto Sprite1 = (Sprite*)contact.getShapeA()->getBody()->getNode();
auto Sprite2 = (Sprite*)contact.getShapeB()->getBody()->getNode();
if (Sprite1->getTag() == 2)
    Sprite1->getPhysicsBody()->setCollisionBitmask(0xFFFFFFFF);
if (Sprite2->getTag() == 2)
    Sprite1->getPhysicsBody()->setCollisionBitmask(0xFFFFFFFF);
return true;
```

### 3. 玩家移动

```
if (IsLeft || IsRight) {
    if (IsLeft) {
        if (Press == 'D') {
            player1->setFlipX(true);
            Press = 'A';
        }
        player1->getPhysicsBody()->setVelocity(Vec2(-100, player1->getPhysicsBody()->getVelocity().y));
    }
    else {
        if (Press == 'A') {
            player1->setFlipX(false);
            Press = 'D';
        }
        player1->getPhysicsBody()->setVelocity(Vec2(100, player1->getPhysicsBody()->getVelocity().y));
    }
}

if (player1->getSpriteFrame() == frame1) {
    auto animation = RepeatForever::create(Animate::create(AnimationCache::getInstance()->getAnimation("player1RunW")));
    animation->setTag(11);
    player1->runAction(animation);
}
else if (player1->getSpriteFrame() == IdleWithBox1) {
    auto animation = RepeatForever::create(Animate::create(AnimationCache::getInstance()->getAnimation("player1RunW")));
    animation->setTag(11);
    player1->runAction(animation);
}
```

### 4. 举起箱子，扔箱子

按下回车，如果正在举箱子，则扔箱子，反之举起

```
joint1 = PhysicsJointDistance::construct(
    player1->getPhysicsBody(), box->getPhysicsBody(),
    player1->getAnchorPoint(), box->getAnchorPoint());

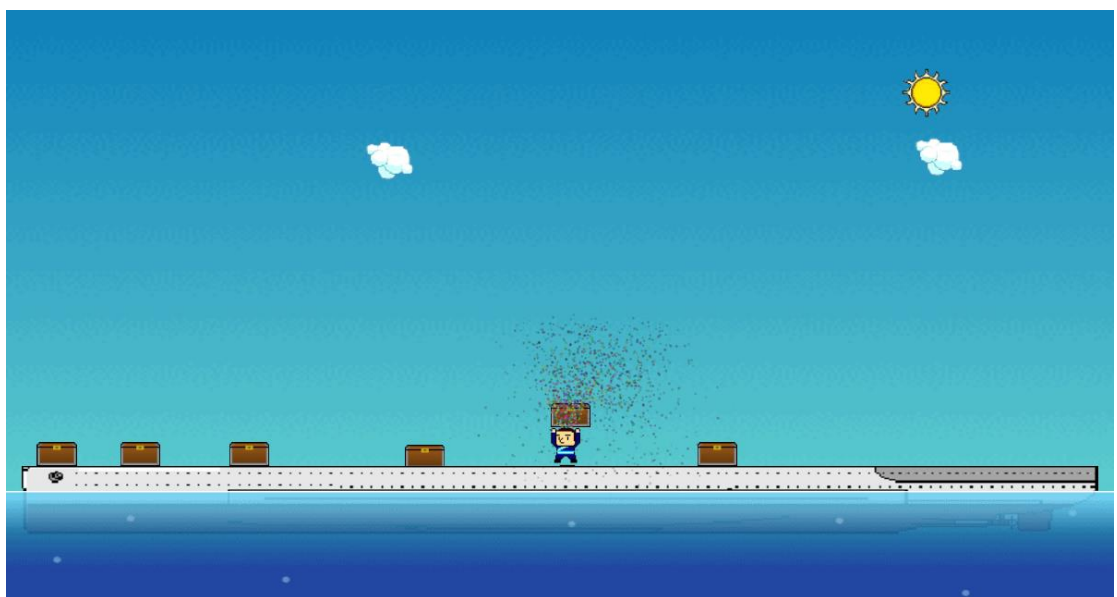
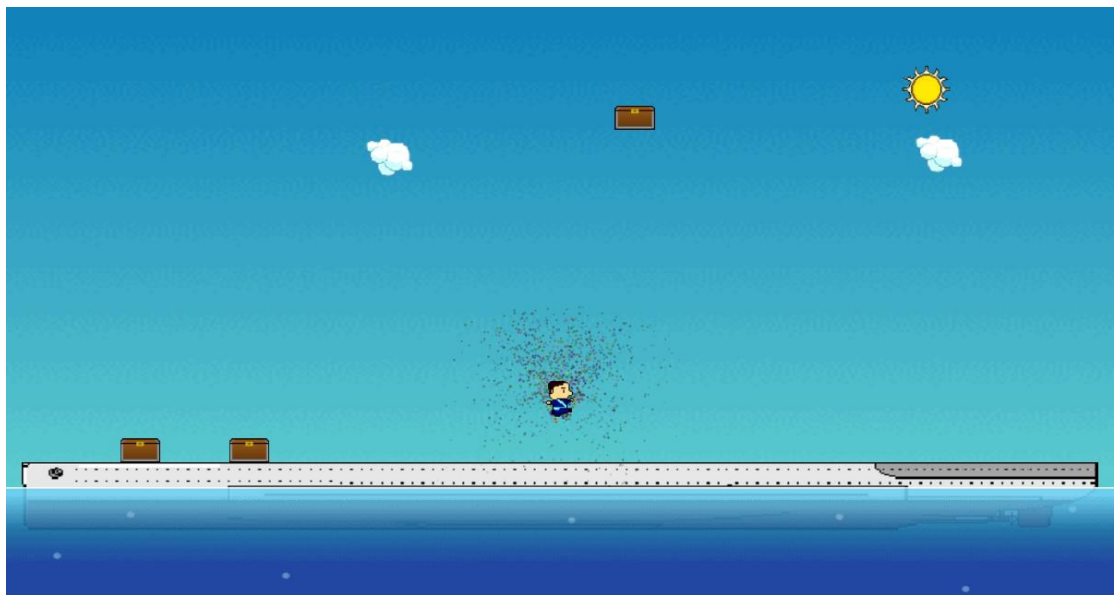
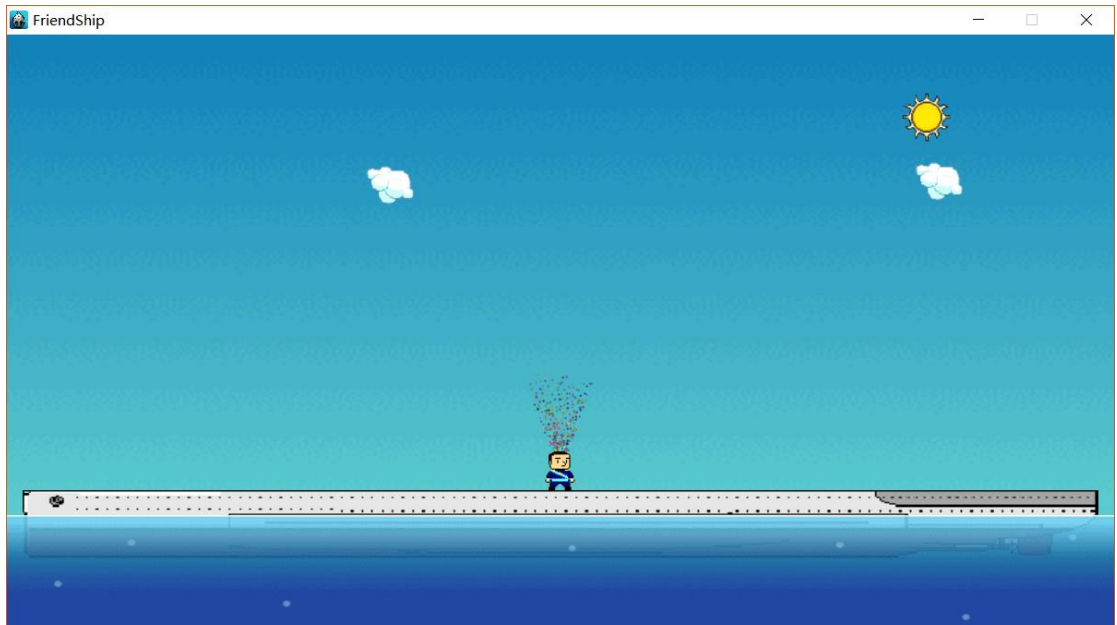
holdBox = box;

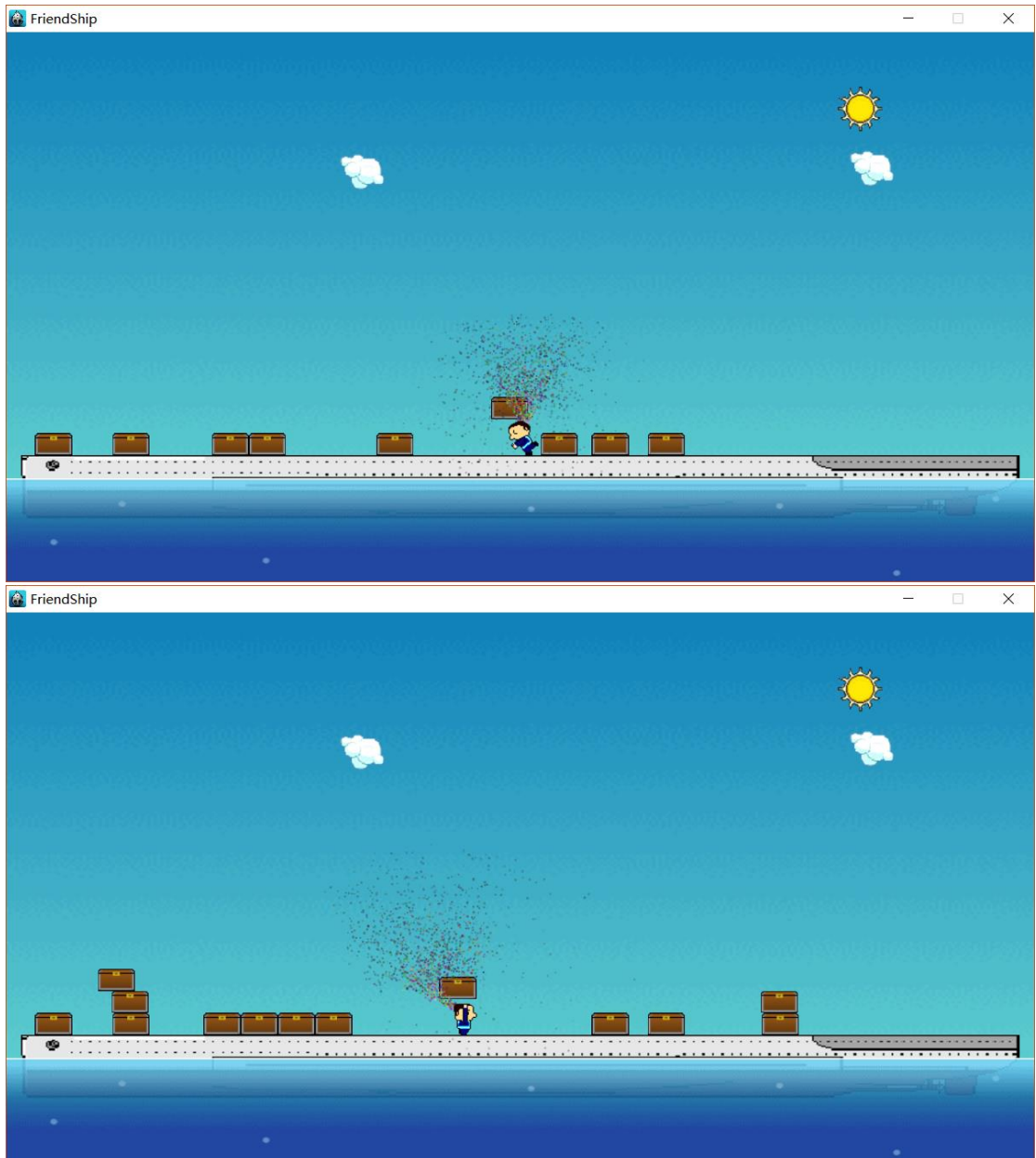
m_world->addJoint(joint1);
player1->setDisplayFrame(IdleWithBox1);
IsPlayer1Hold = true;

holdBox->removeComponent(holdBox->getPhysicsBody());
holdBox->setPhysicsBody(boxbody);

if (LastPlayer1Press == 'A') {
    holdBox->getPhysicsBody()->setVelocity(Vec2(-400 + holdBox->getPhysicsBody()->getVelocity().x, holdBox->getVelocity().y));
}
else {
    holdBox->getPhysicsBody()->setVelocity(Vec2(400 + holdBox->getPhysicsBody()->getVelocity().x, holdBox->getVelocity().y));
}
m_world->removeJoint(joint1);
auto animation = Animate::create(AnimationCache::getInstance()->getAnimation("player1PutDownAnimation"));
animation->setTag(11);
```

## 三．实验结果截图





#### 四．思考与总结

这是个鬼畜的游戏，刚开始无论做什么动作都会各种旋转跳跃，但是一个一个判定就不会了例如密度，旋转，虽然还是有 bug。

已经迟交好多天了就不水了。