

《现代操作系统应用开发》HW14 实验报告

姓名：羊伊 学号：15331349

一、参考资料：

homework14.pdf

课件

二、实验步骤：

1. 至少使用一种粒子系统。(如人物出场或箱子碰撞)

做出一个云遮太阳并下雨的效果。粒子系统即建一个 ParticleRain 并添加到场景中。

```
// add sun and cloud
auto sunSprite = Sprite::create("sun.png");
sunSprite->setPosition(rand() % (int)(visibleSize.width - 200) + 100, 550);
this->addChild(sunSprite);
auto cloudSprite1 = Sprite::create("cloud.png");
cloudSprite1->setPosition(rand() % (int)(visibleSize.width - 200) + 100, rand() % 100 + 450);
this->addChild(cloudSprite1);
auto cloudSprite2 = Sprite::create("cloud.png");
cloudSprite2->setPosition(sunSprite->getPositionX(), sunSprite->getPositionY()-10);
this->addChild(cloudSprite2);
ParticleRain* Rain = ParticleRain::create();
Rain->setPosition(rand() % (int)(visibleSize.width - 200) + 100, 550);
this->addChild(Rain);
```

2. 为玩家和箱子设置刚体属性, 重点是掩码和材料的设置。
3. 箱子下落过程过程能穿透玩家, 不会发生碰撞。

玩家：

因为人不会触发监听器, 所以 setContactTestBitmask 全为零, 另外两个要保证能与落地后的箱子发生碰撞。

```
// 设置角色刚体属性
// Todo
auto physicsBody = PhysicsBody::createBox(Size(32.0f, 36.0f), PhysicsMaterial(0.1f, 0.0f, 0.1f));
physicsBody->setRotationEnable(false);
physicsBody->setCategoryBitmask(0xffffffff);
physicsBody->setCollisionBitmask(0x00000001);
physicsBody->setContactTestBitmask(0x00000000);
player1->setPhysicsBody(physicsBody);
```

箱子：

在初始状态下箱子的 CategoryBitmask 与玩家的

CollisionBitmask 相与，为 0，所以不发生碰撞。箱子的 ContactTestBitmask 不为 0，所以与箱子或者船体碰撞了以后会触发监听器。

```
// 1002
auto physicsBody = PhysicsBody::createBox(Size(37.0f, 22.0f), PhysicsMaterial(0.1f, 0.1f, 600.0f))
;
physicsBody->setVelocity(Vec2(0, 0));
physicsBody->setCategoryBitmask(0xFFFFFFFF);
physicsBody->setCollisionBitmask(0xFFFFFFFF);
physicsBody->setContactTestBitmask(0xFFFFFFFF);
physicsBody->setRotationEnable(false);
boxes.push_back(physicsBody);
box->setPhysicsBody(physicsBody);
```

4. 箱子碰到船或者碰到其他箱子之后，能和玩家发生碰撞。

因为掉落的箱子可能为 SHAPEA 也可能为 SHAPEB，所以在这里为了减少一次判断就直接都设置一下他们的 CategoryBitmask 了，设置之后就 and 玩家的 CollisionBitmask 不为 0 了。

```
// 箱子碰到船或者碰到其他箱子之后改变掩码，可以与玩家发生碰撞
bool Friendship::onConcactBegin(PhysicsContact & contact) {
    auto box1 = contact.getShapeB()->getBody();
    auto box2 = contact.getShapeA()->getBody();
    auto sp1 = (Sprite*)box1->getNode();
    auto sp2 = (Sprite*)box2->getNode();
    sp1->getPhysicsBody()->setCategoryBitmask(0xffffffff);
    sp2->getPhysicsBody()->setCategoryBitmask(0xffffffff);
    return true;
}
```

5. 控制玩家左右移动。

不知道为什么要把这两个 case 放在一起.....

总之就是给人一个初速度让他前进。

```

switch (code) {
case cocos2d::EventKeyboard::KeyCode::KEY_LEFT_ARROW:
case cocos2d::EventKeyboard::KeyCode::KEY_RIGHT_ARROW:
    // 左右移动
    if (code == EventKeyboard::KeyCode::KEY_LEFT_ARROW) {
        IsPlayer1Left = true;
        player1->setFlippedX(true);
        player1->getPhysicsBody()->setVelocity(Vec2(-300, 0));
    } else {
        IsPlayer1Right = true;
        player1->setFlippedX(false);
        player1->getPhysicsBody()->setVelocity(Vec2(300, 0));
    }
    break;
}

```

6. 使用关节举起和扔下箱子。(固定距离关节或其他)

扔掉箱子：根据人物当前运动的方向，决定箱子扔出去的方向，同样的，还是给箱子初速度让他飞出去。

```

case cocos2d::EventKeyboard::KeyCode::KEY_ENTER:
    if (IsPlayer1Hold) {
        IsPlayer1Hold = false;
        joint1->removeFromWorld();
        auto a = (Sprite*)(joint1->getBodyA()->getNode());
        auto b = (Sprite*)(joint1->getBodyB()->getNode());
        auto box = (Sprite*)(boxes.front()->getNode());
        if (a->getSpriteFrame() == box->getSpriteFrame()) {
            if (player1->getPhysicsBody()->getVelocity().x < 0) {
                a->getPhysicsBody()->setVelocity(Vec2(-300, 300));
            } else {
                a->getPhysicsBody()->setVelocity(Vec2(300, 300));
            }
        } else {
            if (player1->getPhysicsBody()->getVelocity().x < 0) {
                b->getPhysicsBody()->setVelocity(Vec2(-300, 300));
            } else {
                b->getPhysicsBody()->setVelocity(Vec2(300, 300));
            }
        }
        if (!IsPlayer1Left && !IsPlayer1Right) player1->setSpriteFrame(frame1);
    } else {

```

遍历 box 的 list，找到第一个碰到人的箱子。以箱子和人的刚体为源，添加一个 joint，然后将箱子放到人头上，并且将 collision 设为 false，这样他们之间就不会有秘制鬼畜的碰撞了。

```

} else {
    for (list<PhysicsBody*>::iterator i = boxes.begin(); i != boxes.end(); i++) {
        if(player1->getBoundingBox().intersectsRect((*i)->getNode()->getBoundingBox())) {
            IsPlayer1Hold = true;
            joint1 = cocos2d::PhysicsJointDistance::construct(player1->getPhysicsBody(), *i,
                player1->getAnchorPoint(), (*i)->getNode()->getAnchorPoint());
            joint1->createConstraints();
            joint1->setCollisionEnable(false);
            player1->setSpriteFrame(IdleWithBox1);
            (*i)->getNode()->setPosition((player1->getPosition().x, player1->getPositionY()
                + player1->getContentSize().height/2);
            m_world->addJoint(joint1);
            break;
        }
    }
}

```

7. 实现人物跳跃

按↑的时候跳跃，用判断速度的方式来限制他只能往上跳一下。

```

case cocos2d::EventKeyboard::KeyCode::KEY_UP_ARROW:
    IsPlayer1Jump = true;
    if(player1->getPhysicsBody()->getVelocity().y == 0)
        player1->getPhysicsBody()->setVelocity(Vec2(0, 300));
    break;

```

8. 举着箱子的情况下跑动和跳跃

因为这里是判断人物的 frame 的，所以只要在拿箱子的时候设置人物的 frame 为 idlewithbox1 就可以实现了。跳跃没有什么限制。

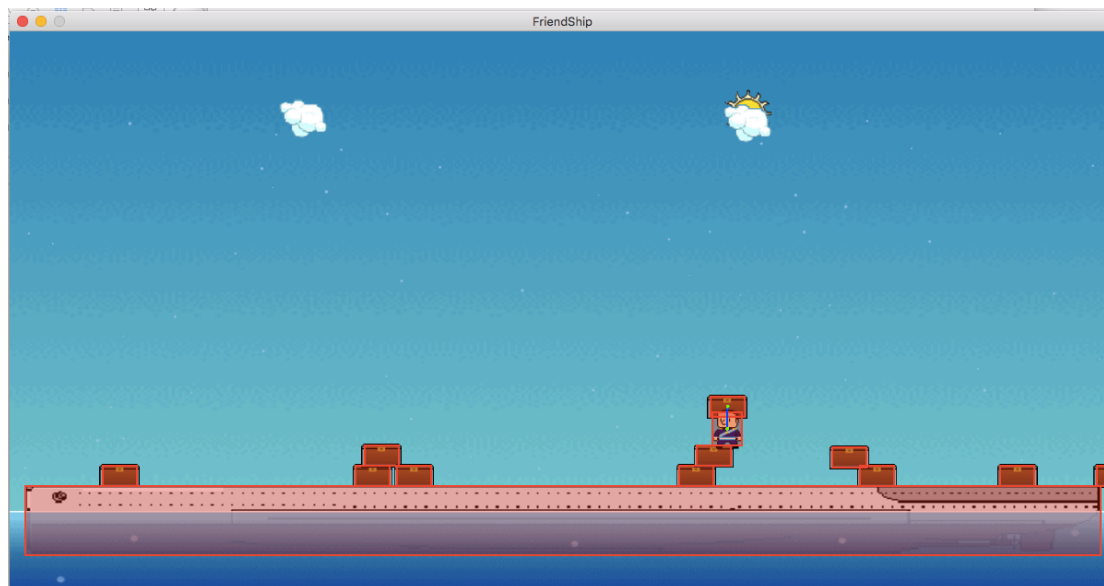
```

if (IsPlayer1Left || IsPlayer1Right) {
    if (player1->getSpriteFrame() == frame1) {
        auto animation = RepeatForever::create(Animate::create(AnimationCache::getInstance()->
            getAnimation("player1RunWithoutBoxAnimation")));
        animation->setTag(11);
        player1->runAction(animation);
    }
    else if (player1->getSpriteFrame() == IdleWithBox1) {
        auto animation = RepeatForever::create(Animate::create(AnimationCache::getInstance()->
            getAnimation("player1RunWithBoxAnimation")));
        animation->setTag(11);
        player1->runAction(animation);
    }
}

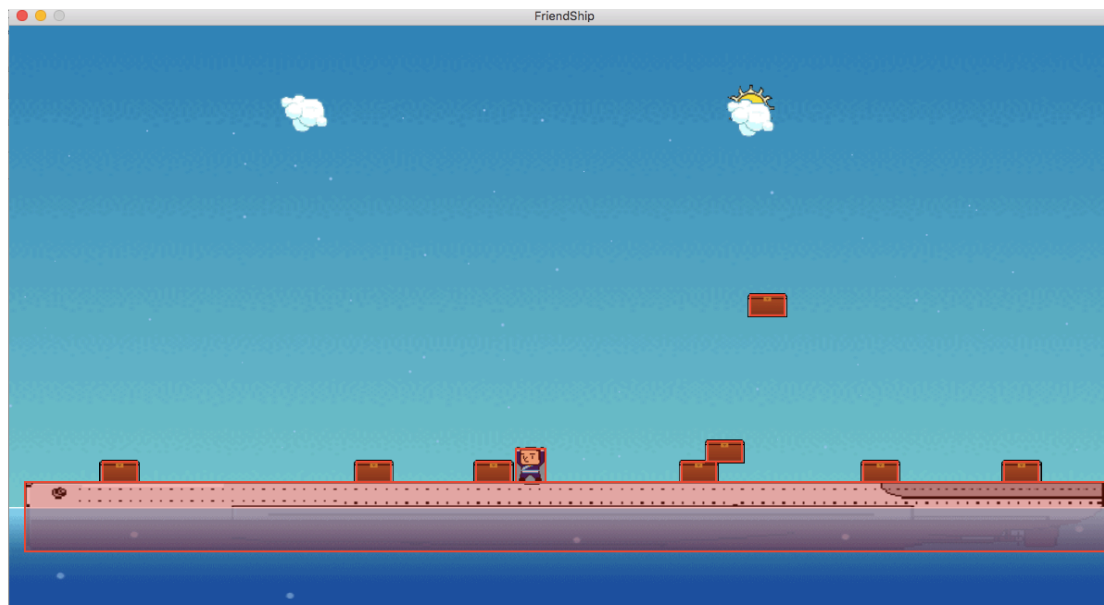
```

三、效果截图

举着箱子跳跃：



把箱子扔出去：





四、收获与感受

感觉 TA 上学期期末做一个比这个更丰富的游戏好 6 啊 = = 创意很好 挺有趣的，用户体验也很好。这个也给我的期末作业提供了一点思路。

这次作业只是简单地完成几个小的功能。碰撞那部分不难，主要就是设置好三个掩码能达到效果就行了。joint 那里如果没有设置 collision 就会很鬼畜，没设置摩擦力箱子就不会老老实实的呆在人的头上。

感觉游戏开发是个很注意细节的工作，不过就是注意了细节用户体验才会好的。

总之这次收获还是挺多的，也让我坚定了以后不要做游戏开发的信念 = =