

# 现代操作系统应用开发实验报告

学号： 14331388

班级： 教务三班

姓名： 郑泽佳

实验名称： 实验 11

## 一. 参考资料

<http://www.cocos2d-x.org/reference/native-cpp/V3.0alpha0/db/d94/classcoco>

[s2d 1 1 user default.html](#)

[课件](#)

[HomeWork 指导](#)

## 二. 实验步骤

### 1、随机产生怪物

实现人物翻转

```
break;
case 'A':
    if (lastCid != 'A')
        player->setFlipX(true);
    lastCid = 'A';
    x = player->getPosition().x - 40;
case 'D':
    if (lastCid != 'D')
        player->setFlipX(false);
    lastCid = 'D';
```

通过 Demo 中所提供的 monster.cpp 已经定义好的 createMonster,再通过自定义调度器在 HelloWorld 中创建 Monster 精灵并随机设置其位置在 0-VisibleSize 范围内。

调度器调用的函数如下：

```
void HelloWorld::createMonster(float dt) {
    auto fac = Factory::getInstance();
    auto m = fac->createMonster();
    float x = random(origin.x, visibleSize.width);
    float y = random(origin.y, visibleSize.height);
    m->setPosition(x, y);
    addChild(m, 3);
    fac->moveMonster(player->getPosition(), 1.0);
}
```

设置怪物移动的方向

```
void Factory::moveMonster(Vec2 playerPos, float time) {
    Vector<Sprite*>::iterator it = monster.begin();
    for (; it != monster.end(); it++) {
        Vec2 monsterPos = (*it)->getPosition();
        Vec2 direction = playerPos - monsterPos;
        direction.normalize();
        (*it)->runAction(MoveBy::create(time, direction * 30));
    }
}
```

## 2、怪物碰到角色后，角色掉血

通过调度器模拟监听器作用，监听怪物是否与人物发生碰撞

调度器调用的函数为：

```
void HelloWorld::hitByMonster(float dt) {
    auto fac = Factory::getInstance();
    Sprite* collision = fac->collider(player->getBoundingBox());
    if (collision != NULL) {
        player->stopAllActions();
        fac->removeMonster(collision);
        actionEvent(this, 'X');
    }
}
```

通过 collider()函数判断是否有怪物与人物发生碰撞，若有则人物掉血，并把怪物删除

Collider()函数如下：

```

Sprite* Factory::collider(Rect rect) {
    Vector<Sprite*>::iterator it = monster.begin();
    for (; it != monster.end(); it++) {
        if ((*it)->getBoundingBox().intersectsRect(rect)) {
            break;
        }
    }
    if (it == monster.end())
        return nullptr;
    return *it;
}

```

removeMonster()函数如下:

```

void Factory::removeMonster(Sprite* sp) {
    Animation* animation = Animation::create();
    for (int i = 0; i < monsterDead.size(); i++)
        animation->addSpriteFrame(monsterDead.at(i));
    animation->setDelayPerUnit(0.5f / 4.0f);
    auto action = Animate::create(animation);
    auto sequence = Sequence::create(action,
        CallFunc::create(CC_CALLBACK_0(Sprite::removeFromParent, sp)), NULL);
    sp->runAction(sequence);
    monster.eraseObject(sp);
}

```

### 3、角色可以攻击怪物，打倒怪物后回血

将 Label'X'从界面中删除而又不删除其作用，不把 Label'X'添加到层中也就不显示出来

```

auto item5 = MenuItemLabel::create(menuLabel5, CC_CALLBACK_1>HelloWorld::actionEvent, this, 'X');
// ...
auto menu = Menu::create(item1, item2, item3, item4, item6, NULL);
// ...

```

角色的攻击范围：

```

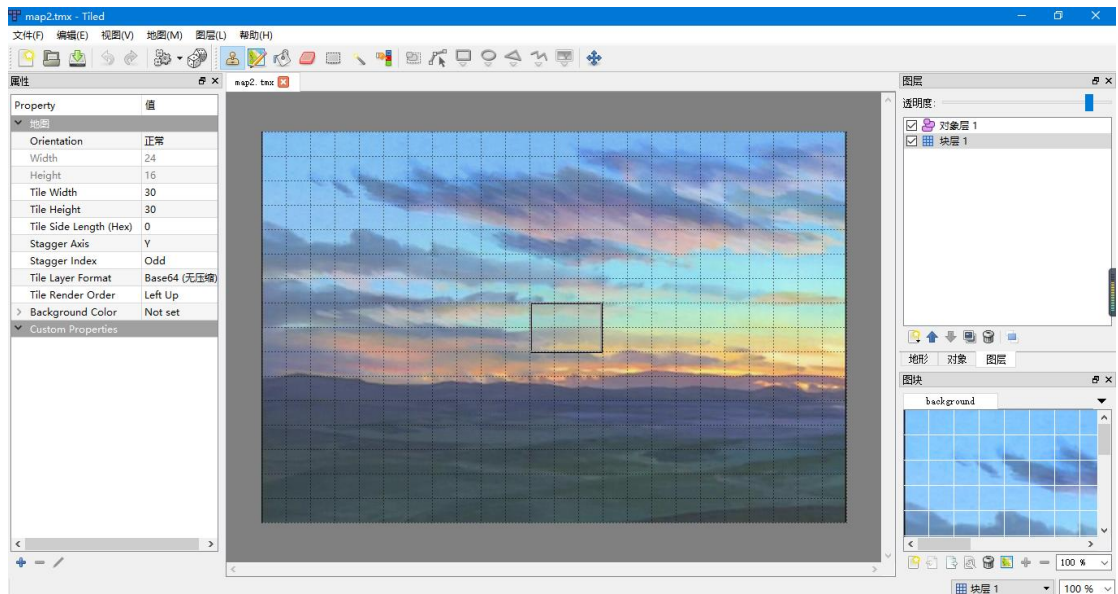
auto playerRect = player->getBoundingBox();
auto attackRect = Rect(playerRect.getMinX() - 40, playerRect.getMinY(),
    playerRect.getMaxX() - playerRect.getMinX() + 80,
    playerRect.getMaxY() - playerRect.getMinY());
Sprite* attack = fac->collider(attackRect);

```

只要攻击范围与怪物的边界有交集则角色回血，怪物死亡

### 4、使用 tilemap 创建地图

下载好软件 Tiled,根据图层的大小设置像素大小为 720\*480。设置内容如下：



把生成的.tmx 文件和图片资源放入项目中的 Resource 中，并在代码中导入该资源

```
//Load a map
TMXTiledMap* tmx = TMXTiledMap::create("map2.tmx");
tmx->setPosition(visibleSize.width / 2, visibleSize.height / 2);
tmx->setAnchorPoint(Vec2(0.5, 0.5));
tmx->setScale(Director::getInstance()->getContentScaleFactor());
this->addChild(tmx, 0);
```

##### 5、使用本地数据存储，记录打倒的怪物数量，并将倒计时改为显示打倒数

使用 UserDefault.xml 保存本地数据，游戏每次开始运行读取保存好的数据。

```
//database store data
#define database UserDefault::getInstance()
if (!database->getBoolForKey("isExit")) {
    database->setBoolForKey("isExit", true);
}
killNum = database->getIntegerForKey("Number");
```

将倒计时改为显示的打倒数

```
cocos2d::Label* num;
int killNum;
```

```

TTFConfig ttfConfig;
ttfConfig.fontFilePath = "fonts/arial.ttf";
ttfConfig.fontSize = 36;
char str[5];
sprintf(str, "%d", killNum);
num = Label::createWithTTF(ttfConfig, str);
num->setPosition(Vec2(origin.x + visibleSize.width / 2,
    origin.y + visibleSize.height - num->getContentSize().height));
addChild(num, 1);

```

攻击到怪物就增加打倒怪物的数量并保存到 UserDefalut.xml 中，

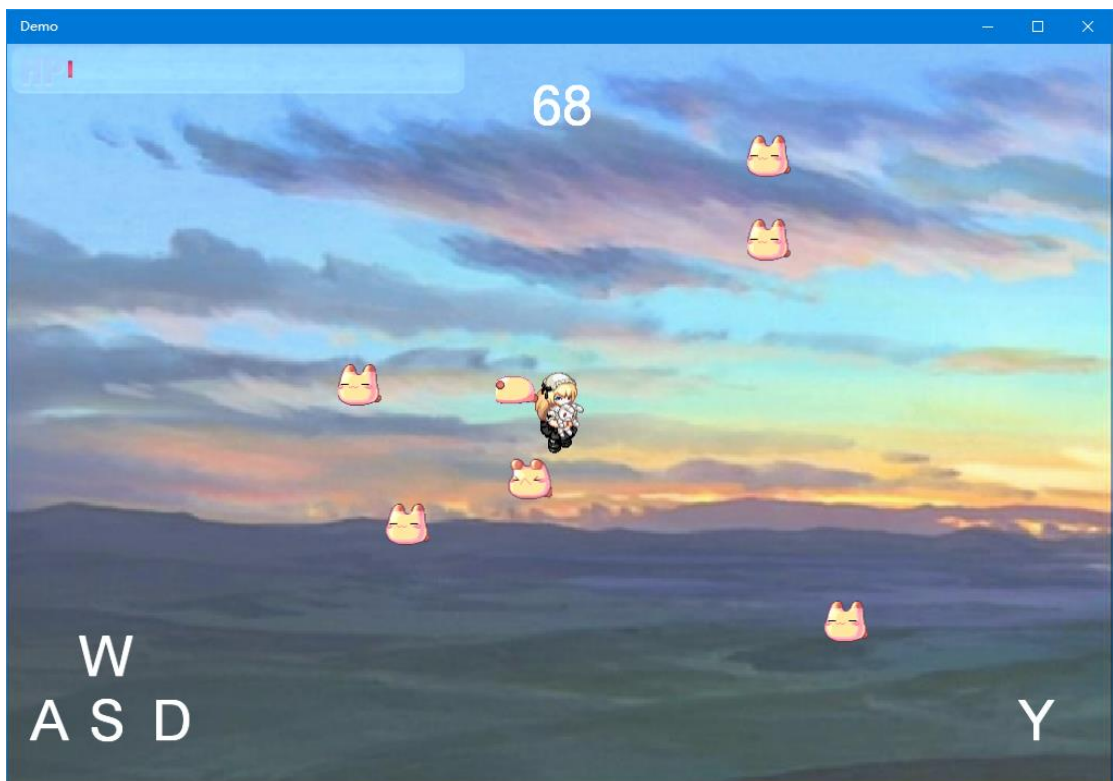
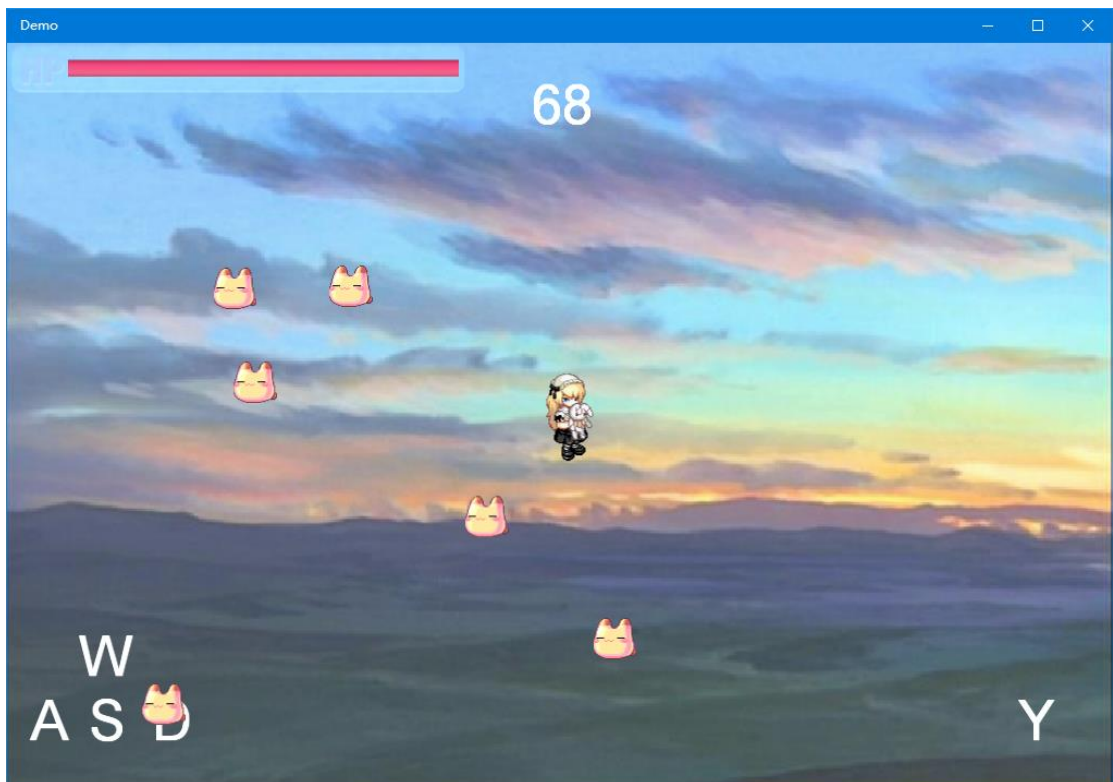
```

if (attack != NULL) {
    player->stopAllActions();
    fac->removeMonster(attack);
    killNum++;
    update();
    progressFromTo = ProgressFromTo::create(1.5f, pT->getPercentage(), pT->getPercentage() + 2);
    pT->runAction(progressFromTo);
}
break;

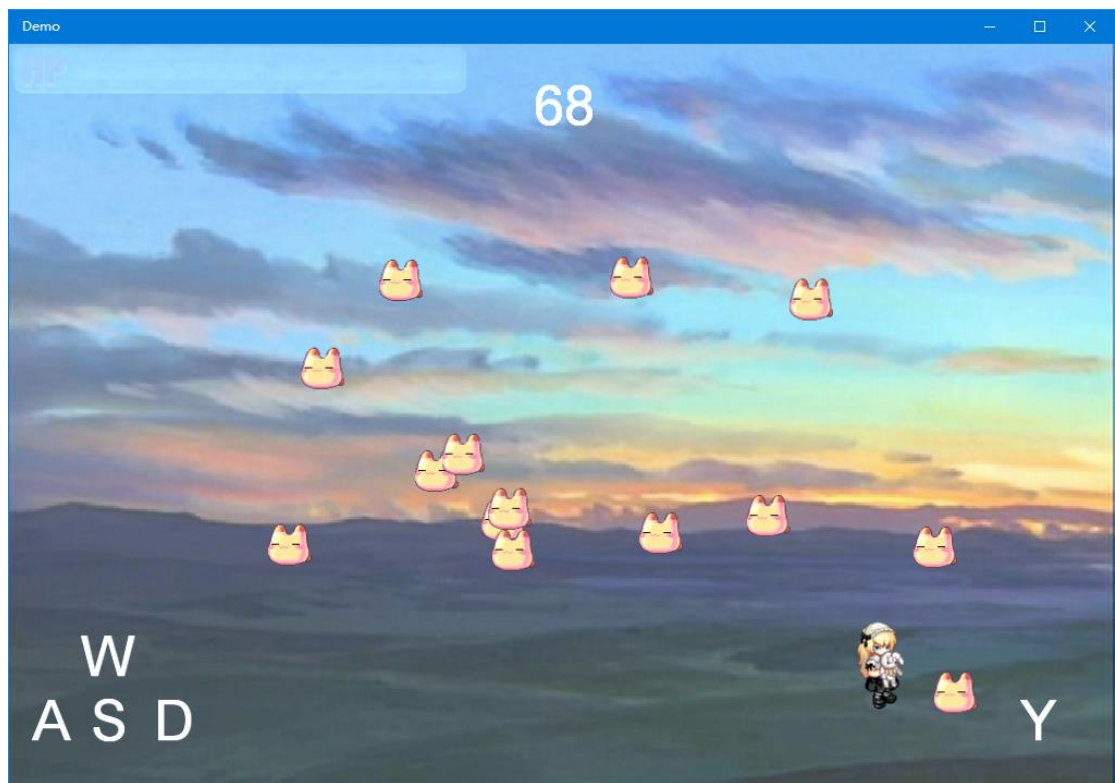
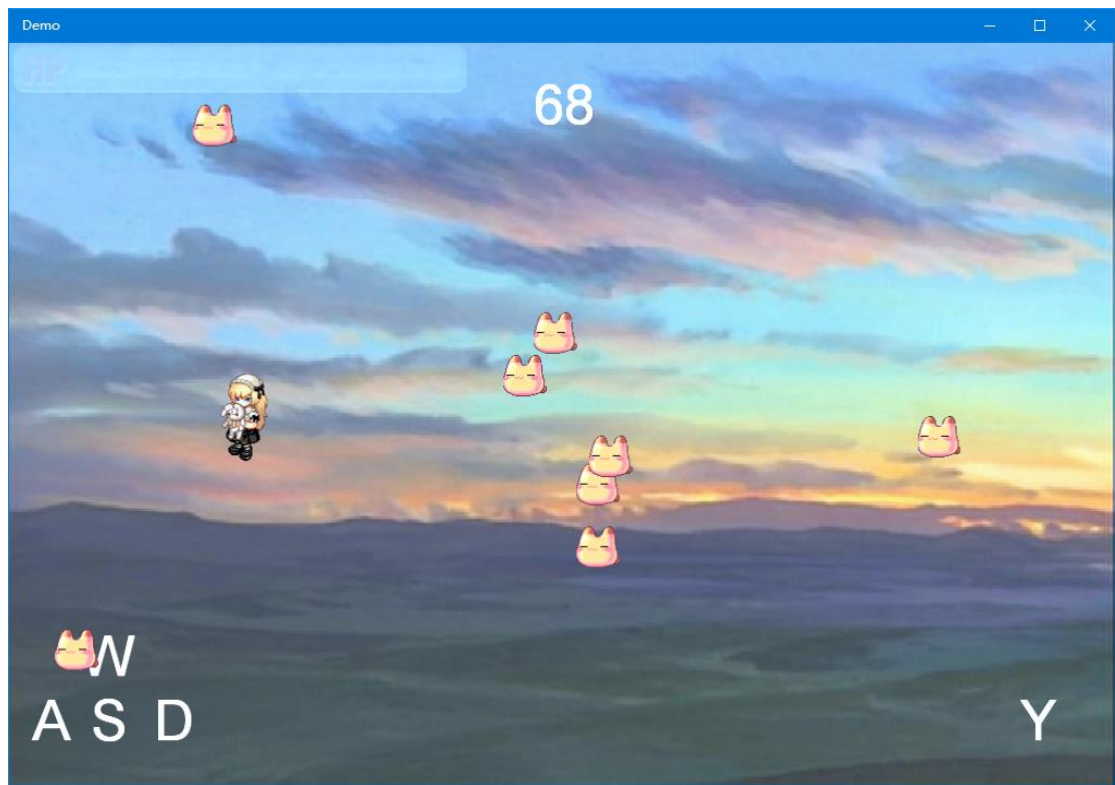
void HelloWorld::update() {
    char str[5];
    sprintf(str, "%d", killNum);
    num->setString(str);
    database->setIntegerForKey("Number", killNum);
    database->flush();
}

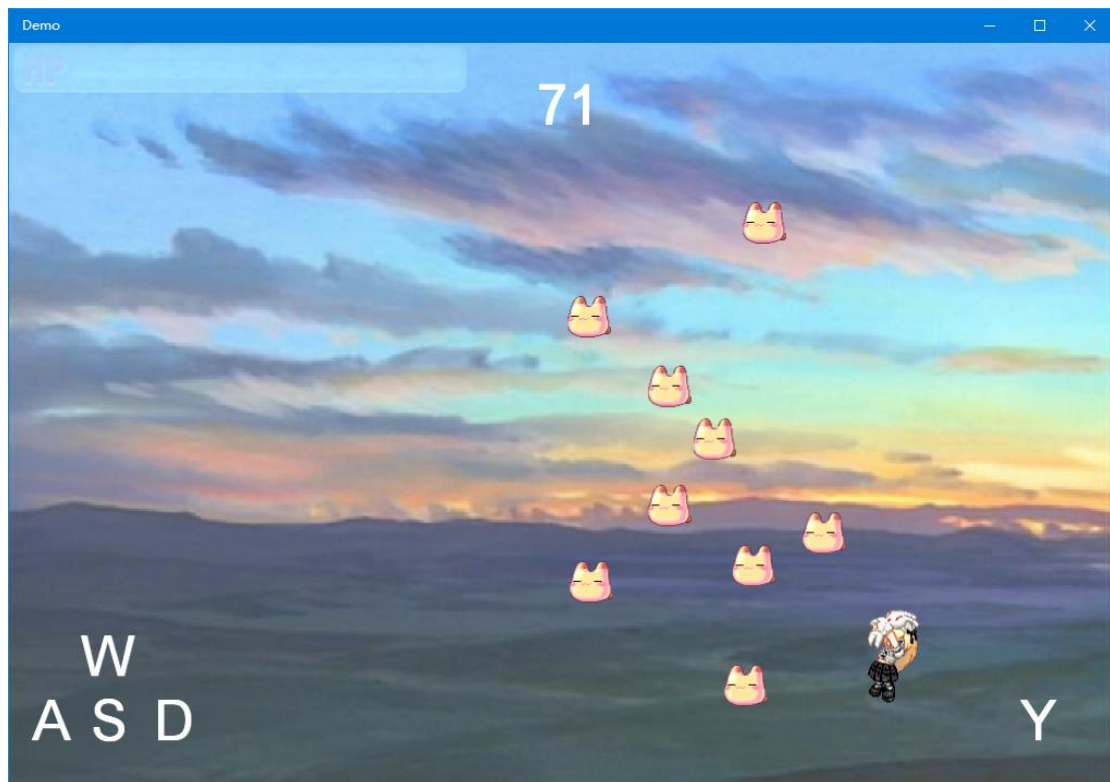
```

### 三．实验结果截图









#### 四 . 实验过程遇到的问题

问题一：一开始对于人物的动作时的抽搐现象没有好的解决方法，查询方法知道可以通过 ActionManager 来管理人物的动作

解决方案：通过判断人物正在执行的动作为 0 个，才执行新的动作。

```
if (player->getNumberOfRunningActions() == 0)
    player->runAction(Spawn::create(action, moveTo, nullptr));
```

问题二：导入 map2.tmx 时程序报错

解决方法：将 map2.tmx 属性中 Tile Layer Format 改为 Base64(无压缩)即可

#### 五 . 思考与总结

最近的作业给的资料的非常详细，基本上不需要额外地去网上查找相关资料就可以做出来，但感觉学的不是很扎实，至今对 cocos 程序运行的整个流程也还不是很清楚，例如程序的启动从哪里开始，程序从哪里结束进程的。所以还是需要进一步学习吧。