

# 现代操作系统应用开发实验报告

学号： 14307004

班级： 15 软工 1 班

姓名： 蔡冠文

实验名称： Homework12

## 一. 参考资料

请在这里列出对本实验有帮助你所参考的资料或者网站。

[http://api.cocos.com/d4/d8a/classcocos2d\\_1\\_1\\_rect.html#a7579247391a0cf6ffe630b0ec0501937](http://api.cocos.com/d4/d8a/classcocos2d_1_1_rect.html#a7579247391a0cf6ffe630b0ec0501937)

[http://api.cocos.com/dc/d82/classcocos2d\\_1\\_1\\_scheduler.html#aef25756829a908b7f63483dff0e29889](http://api.cocos.com/dc/d82/classcocos2d_1_1_scheduler.html#aef25756829a908b7f63483dff0e29889)

## 二. 实验步骤

### 1. 加载地图

```
//创建地图
TMXTiledMap* tmx = TMXTiledMap::create("map.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);
```

### 2. 使用 setFlip 左右翻转

```
id HelloWorld::leftCallback(cocos2d::Ref* pSender) {
    if (running)
    {
        return;
    }
    auto currentPos = player->getPosition();
    player->setFlipX(true);
```

3. 按照 PPT 的做法，填充 createMonster(),removeMonster()

```
Sprite* Factory::createMonster() {
    Sprite* mons = Sprite::create("Monster.png", CC_RECT_PIXELS_TO_POINTS(Rect(364, 0, 42, 42)));
    monster.pushBack(mons);
    return mons;
}

void Factory::removeMonster(Sprite* sp) {
    Animation* animation = Animation::createWithSpriteFrames(monsterDead, 0.1f);
    Animate* animate = Animate::create(animation);
    Sequence* seq = Sequence::create(animate, CallFunc::create(
        CC_CALLBACK_0(Sprite::removeFromParent, sp)), NULL);
    sp->runAction(seq);
    monster.eraseObject(sp);
}
```

4. 在 HelloWorldScene 里面先调用一下这几个函数，monster 相关的都在上次作业的 update 里面。

```
//获取工厂，生成怪物，放置在场景中
auto fac = Factory::getInstance();
if (fac->getNum() < 10)
{
    auto m = fac->createMonster();
    float mx = random(origin.x, visibleSize.width);
    float my = random(origin.y, visibleSize.height);
    m->setPosition(mx, my);
    addChild(m, 3);
}

hitByMonsters(dt);
if (running) {
    return;
}
else {
    fac->moveMonster(player->getPosition(), 1);
}
}
```

5. 调用区域重叠的 API 判断矩形区域是否重叠，填充 collider ( ) 函数

```
Sprite* Factory::collider(Rect rect) {
    //判断矩形区域是否有重合
    for (auto &i : monster) {
        if (i->getBoundingBox().intersectsRect(rect))
        {
            return i;
        }
    }
    return NULL;
}
```

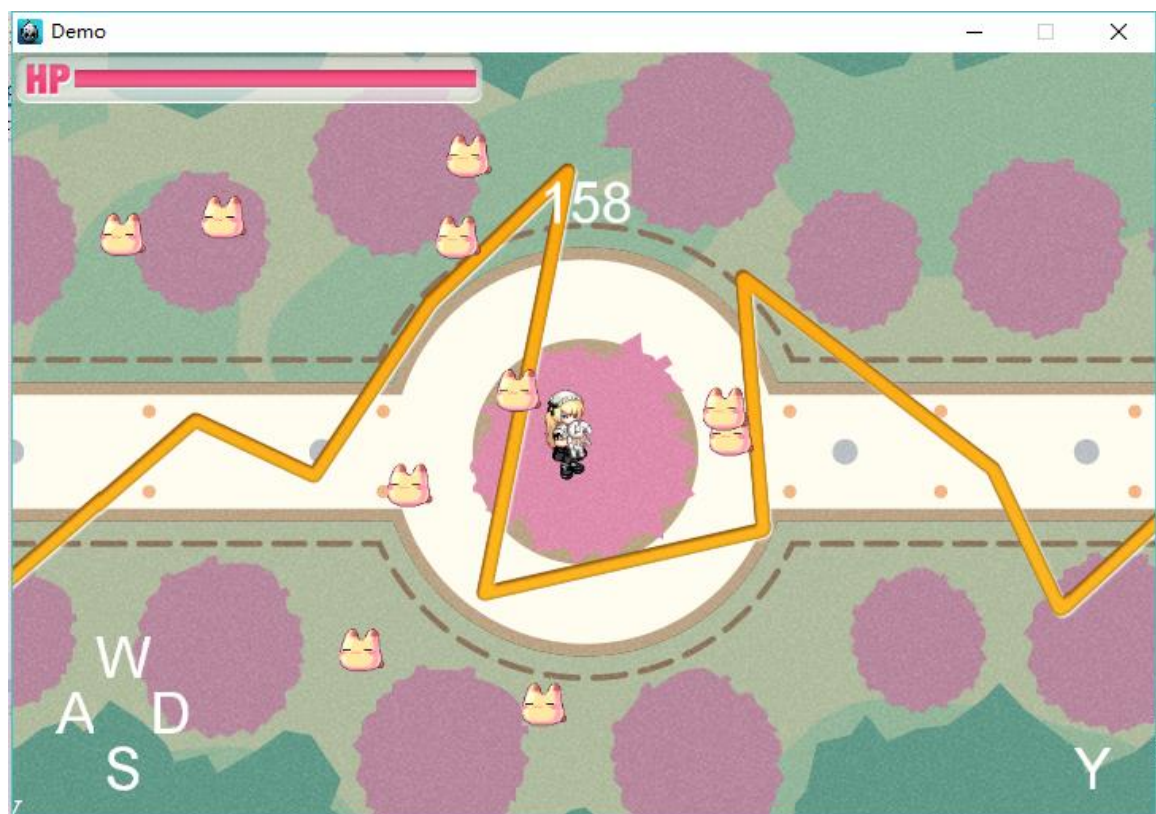
6. 把原来 Y 的动作做一些修改 满足 collider 的触发才能回血。X 的动作在 HitByMonster 里面调用。

```
void HelloWorld::hitByMonsters(float dt)
{
    auto fac = Factory::getInstance();
    Sprite* collision = fac->collider(player->getBoundingBox());
    if (collision!=NULL)
    {
        fac->removeMonster(collision);
        xCallback(this);
    }
}
```

### 三 . 实验结果截图

请在这里把实验所得的运行结果截图。

生成 monster



碰撞后掉血



按 Y 打死怪物并回血（翻转成左边）



#### 四．实验过程遇到的问题

请在这里写下你在实验过程中遇到的问题以及解决方案。

##### 1. collider 怎么写

解决方案：请教同学，搜索到 Rect 区域重叠的 API

## 2. 出现空指针错误

Microsoft Visual Studio



引发了异常: 读取访问权限冲突。

this->player 是 nullptr。

如有适用于此异常的处理程序, 该程序便可安全地继续运行。

解决方案：在 init 调用了帧相关的函数 hitByMonster，把它放回去 update 里面调用即可。

## 3. 怪兽移动的时候一格一格的跳着动，而不是平滑地先走一段再停

解决方案：

```
else {  
    fac->moveMonster(player->getPosition(), 1);  
}
```

在 update 里面调用这个 moveMonster 函数，float dt 以为是和 update 里面的 dt 是一样的，一开始的参数是 dt，后来经同学指点才发现，这里的 time 是决定距离的，换成了 1，每次移动的距离减小，看得像是平滑的移动。

## 五 . 思考与总结

请在这里写下你本次试验的心得体会以及所思所想。

这次实验在上次的基础上更进一步，进一步熟悉了动画和事件的处理。这次 monster.cpp 独立开来，不再是臃肿地往 HelloWorldScene 里面添加函数，代码分层架构更清晰。通过这次实验也更加熟悉了如何调用其它代码文件的方式，巩固了一下 C++ 的知识。

在上次的实验报告中我就想到，update 只是做了一个倒计时的工作，以后会不会更进一步呢？马上在这次就用得上了，这一次的 update 基本负责了所有 monster 函数的调用，包括 create, remove, move 等等。

另外还学到的一个小技巧就是地图的载入和制作。当然这次只是用了现有 tmx 文件导入