《现代操作系统应用开发》HW12 实验报告

姓名：羊伊 学号：15331349

一、参考资料：

homework12.pdf

二、实验步骤：

1.使用 tilemap 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.随机产生怪物

采取调度器的形式，每两秒随机生成一个怪物。

```
schedule(schedule_selector(HelloWorld::update), 2.0f, kRepeatForever, 0);
```

```
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 HelloWorld::update(float dt) {
    auto fac = Factory::getInstance();
    for (int i = 0; i < 1; i++) {
        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(), dt);
}
```

3.角色可以攻击怪物

这个部分和上周的大致相同，依旧是 Y 的回调函数。不同的地方是增加了对是否打到怪物的判断。如果打到了，就增加 hp。

```
void HelloWorld::AttackMenuCallback(Ref* pSender) {
    if (!canAct()) return;
    Animate* _attack = Animate::create(AnimationCache::getInstance()->getAnimation("attack"));
    player->runAction(_attack);
    Rect playerRect = player->getBoundingBox();
    Rect attackRect = Rect(playerRect.getMinX()-40, playerRect.getMinY(),playerRect.getMaxX()-
        playerRect.getMinX()+80, playerRect.getMaxY()-playerRect.getMinY());
    auto fac = Factory::getInstance();
    auto collision = fac->collider(attackRect);
    if (collision != NULL) {
        fac->removeMonster(collision);
        if(percent < 100) {
            pT->runAction(CCProgressTo::create(2, percent+20));
            percent += 20;
        }
        killed++;
        String* temp = String::createWithFormat("%d", killed);
        time->setString(temp->_string);
        database->setIntegerForKey("value", killed);
    }
}
```

判断是否有碰撞的方法是看怪兽的位置是否在打击范围内。

```
Sprite* Factory::collider(Rect rect) {
    for (auto i = monster.begin(); i != monster.end(); i++) {
        if (rect.containsPoint((*i)->getPosition())) {
            return *i;
        }
    }
    return NULL;
}
```

4..怪物碰到角色后，角色掉血

为了保证及时的判断是否碰到怪物，采用调度器的形式，每 0.2 秒判断一次。如果碰到了，那么掉血。如果血掉光了，就停止游戏。

```
schedule(schedule_selector(HelloWorld::update), 2.0f, kRepeatForever, 0);
schedule(schedule_selector(HelloWorld::HitByMonster), 0.2f, kRepeatForever, 0);
return true;
```

```
void HelloWorld::HitByMonster(float dt) {
    if (!canAct()) return;
    auto fac = Factory::getInstance();
    auto collision = fac->collider(player->getBoundingBox());
    if (collision != NULL) {
        fac->removeMonster(collision);
        if(percent >= 20) {
            pT->runAction(CCProgressTo::create(2, percent-20));
            percent -= 20;
        }
        Animate* _dead = Animate::create(AnimationCache::getInstance()->getAnimation("dead"));
        player->runAction(_dead);
        if (percent == 0) {
            Director::getInstance()->pause();
        }
    }
}
```

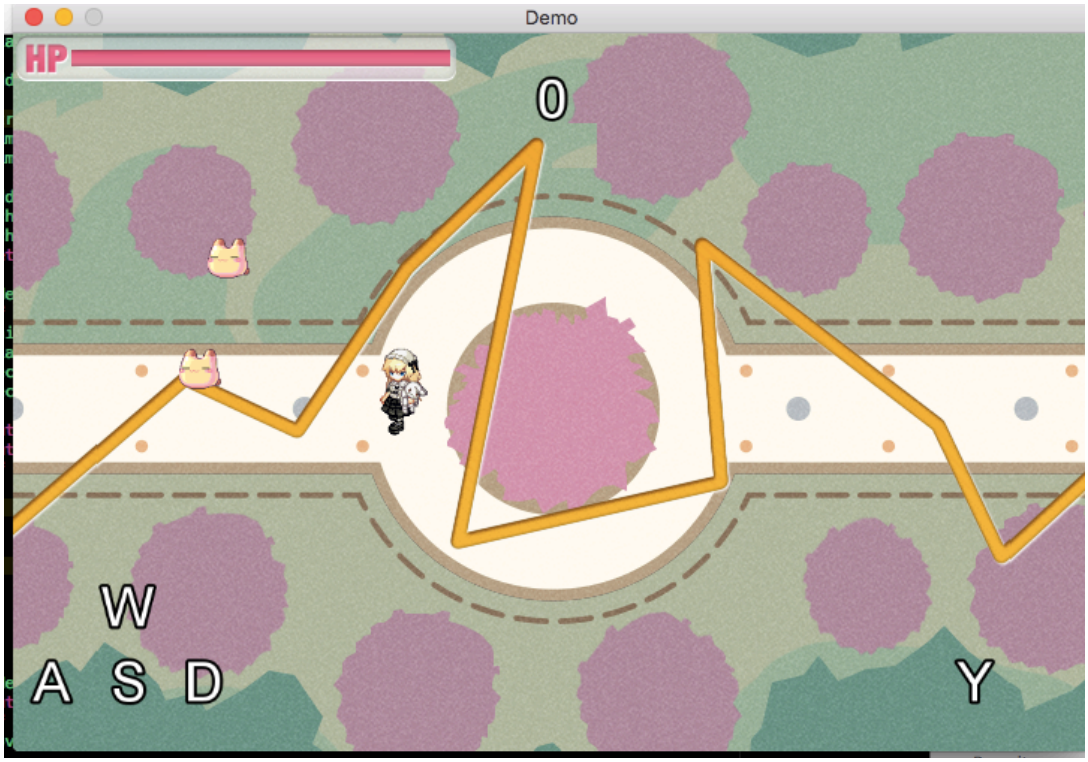
5.加分项：使用本地数据存储，记录打到的怪物数量，并将倒计时改为显示打倒数量。

```
        killed++;
        String* temp = String::createWithFormat("%d", killed);
        time->setString(temp->_string);
        database->setIntegerForKey("value", killed);
```

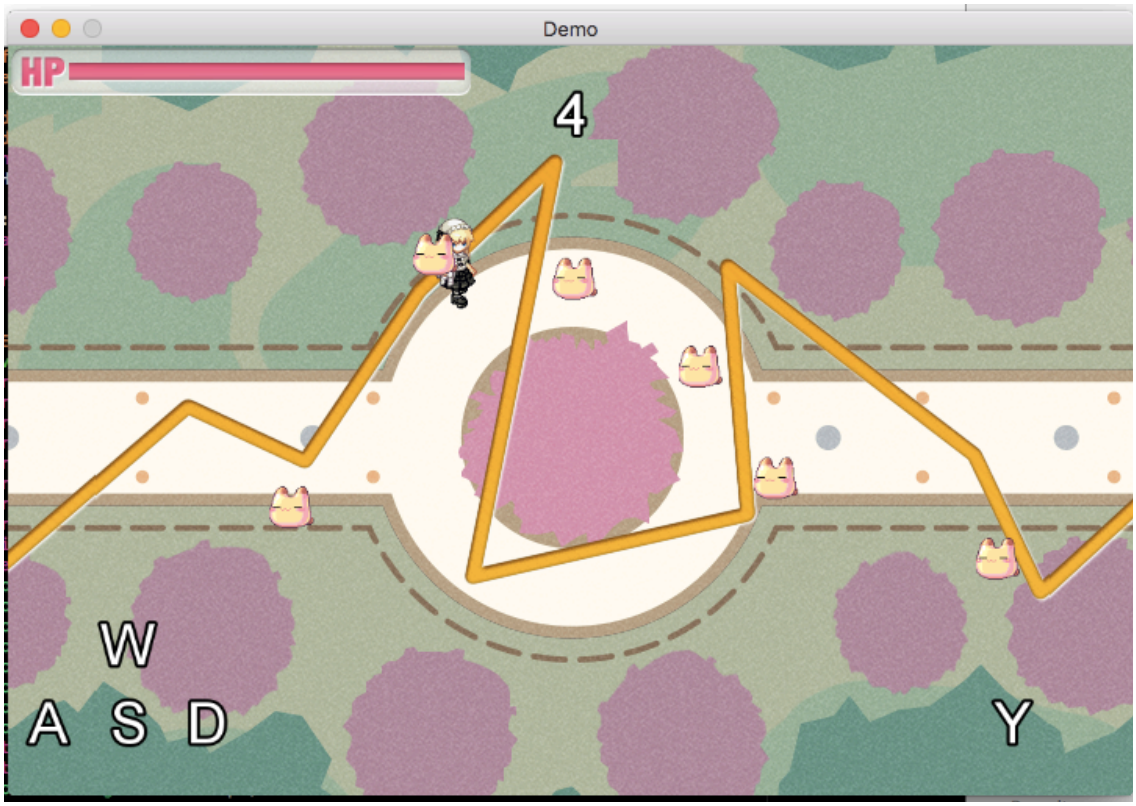
```
/Users/LeLe/Documents/
```

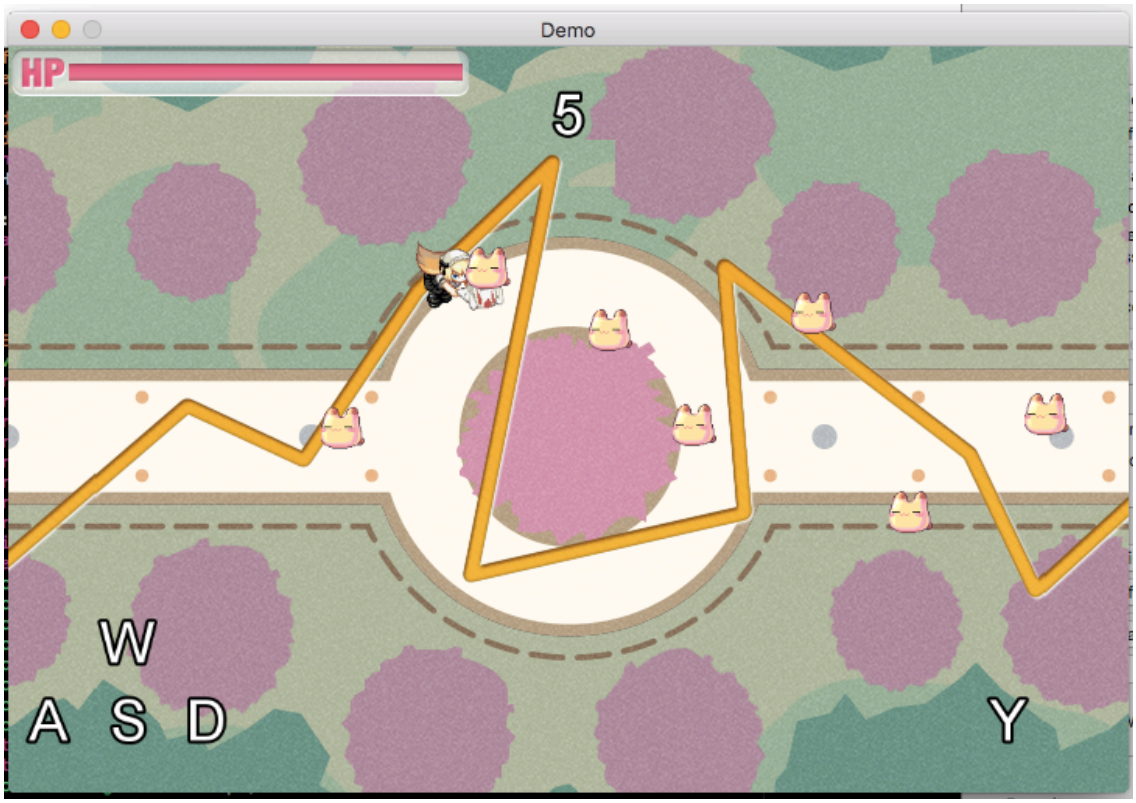
三、效果截图

人物翻转：



打击后增加分数





四、收获与感受

本次实验内容不是很难，感觉难都难在测试时的游戏操作上了，我这种游戏渣控制不好总是截不到什么图…