

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： Homework11

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/dc/d79/classcocos2d_1_1_menu_item_font.html

<http://blog.csdn.net/chaoyuan899/article/details/17022191>

二 . 实验步骤

请在这里简要写下你的实验过程。

1. 创建按钮，调整布局，将 main 加入界面。

设置按钮，添加 CC_CALLBACK1

```
// 设置按钮
```

```
auto up = MenuItemFont::create("W", CC_CALLBACK_1(HelloWorld::upCallback, this));
```

setItem 函数设置字体字号位置。

```
void HelloWorld::setItem(MenuItemFont * item, Vec2 pos)
{
    item->setFontNameObj("arial");
    item->setFontSizeObj(36);
    item->setPosition(pos);
}
```

```
setItem(up, Vec2(70, 100));
setItem(down, Vec2(70, 30));
setItem(left, Vec2(40, 65));
setItem(right, Vec2(100, 65));
```

按照 PPT 的方法创建 menu:

```
// 创建menu
auto menu = Menu::create(up, down, left, right, x, y, NULL);
menu->setPosition(Vec2::ZERO);
this->addChild(menu, 3);
```

2. 为 WSAD 添加四个方向移动的函数，调用 moveTo。每次点击，对应的 xy 坐标会发生变化。

```
void HelloWorld::upCallback(cocos2d::Ref* pSender) {
    if (running)
    {
        return;
    }
    auto currentPos = player->getPosition();
    auto moveTo = MoveTo::create(0.5, Vec2(currentPos.x, currentPos.y + 40));
```

3. 给四个回调函数分别添加 run 的动画。

```
auto animation = Animation::createWithSpriteFrames(run, 0.1f);
auto animate = Animate::create(animation);

auto spawn = Spawn::create(moveTo, animate, NULL);
```

4. 仿照 attack 写 dead 的动画，给按钮 X 和按钮 Y 添加动画。

```
void HelloWorld::xCallback(cocos2d::Ref* pSender) {
    auto animation = Animation::createWithSpriteFrames(dead, 0.1f);
}

void HelloWorld::yCallback(cocos2d::Ref* pSender) {
    auto animation = Animation::createWithSpriteFrames(attack, 0.1f);
```

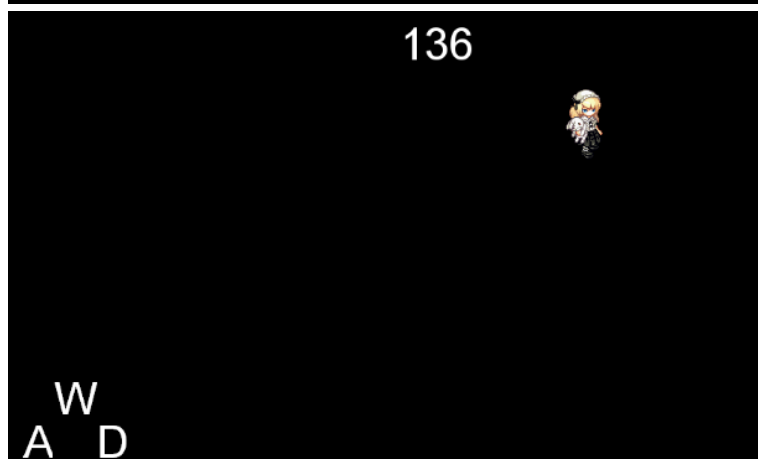
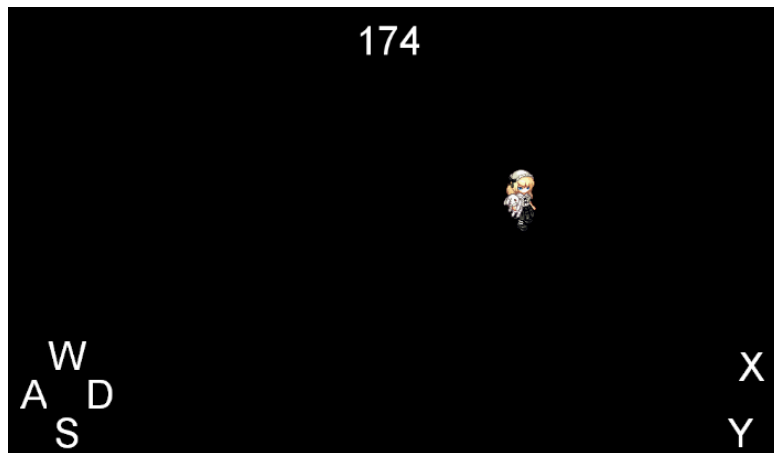
5. 调用 progressTo，使血条变化

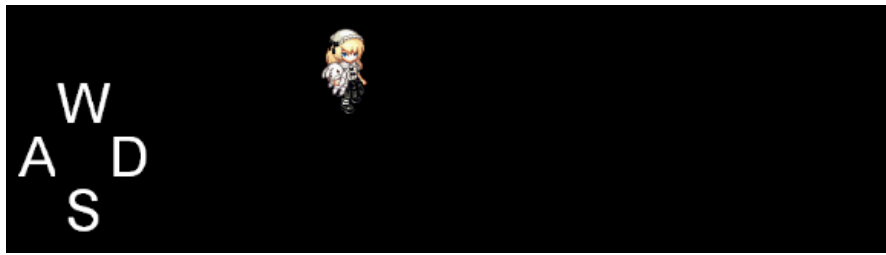
```
// 血条变化，调用progressTo
auto change = ProgressTo::create(1, pT->getPercentage() + 20);
pT->runAction(change);
```

6. 重载 update 函数，调用 scheduleUpdate();

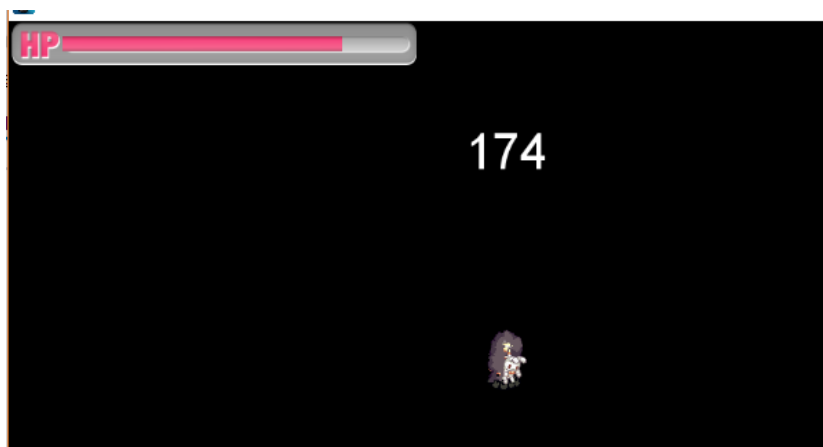
三．实验结果截图

上下左右移动：

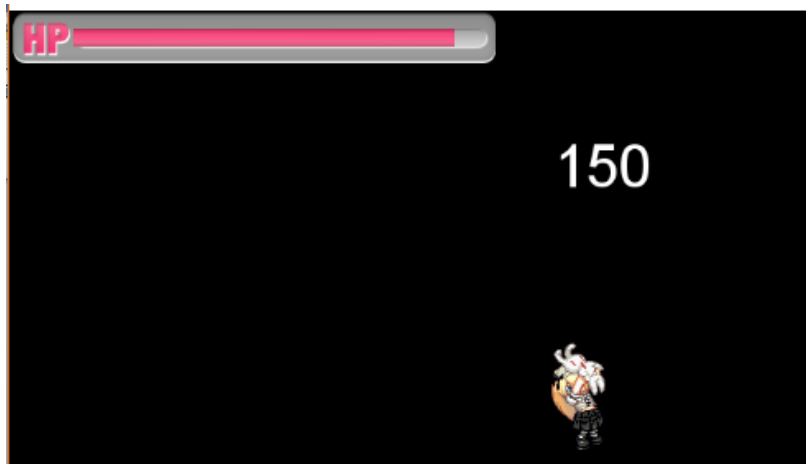




按 X 执行 dead , 血条减少



按 Y 执行 attack , 血条增加



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. 动作结束后没有回到 idle 状态

解决方案：把初始帧加到每一个动画的最后一帧

```
//恢复原来的状态  
attack.pushBack(attack.at(0));  
  
//恢复原来的状态  
dead.pushBack(dead.at(0));
```

2. 动作重叠

解决方案：添加 bool 变量 running，初始设为 false，有动作的时候改为 true，每个动作

执行先判断它的值

3. 时间倒数的飞快

解决方案：默认的 scheduleUpdate()调用频率为 1 秒 120 次，显示倒数的时候需要一秒调

用 1 次，所以只要在它等于某个值的时候调用即可。

```
void HelloWorld::update(float dt)  
{  
    timecount++;  
    if (timecount == 60) {  
        if (index < 0)  
            index = 0;  
        else {  
            index--;  
            time->setString(Value(index))  
            timecount = 0;  
        }  
    }  
}
```

五．思考与总结

这次实验进一步学习了 cocos2dx 的动画，上次作业是将已有的动画添加到游戏中，这次作业用到了帧动画，将已有的动画图片分割，create 一个 animation。另一方面是学习了调度器。覆盖重写 update 后，现在的 update 跟 unity3D 里面的 update 函数有点类似了，每帧调用。这次的 update 只是作为一个时间倒计时，不知道以后还有什么用途。