现代操作系统应用开发实验报告

学号： 15331418　　　　**班级　：** 晚上班

姓名： 郑柏川　**实验名称：Coco2d-x_UI_调度器_帧动画**

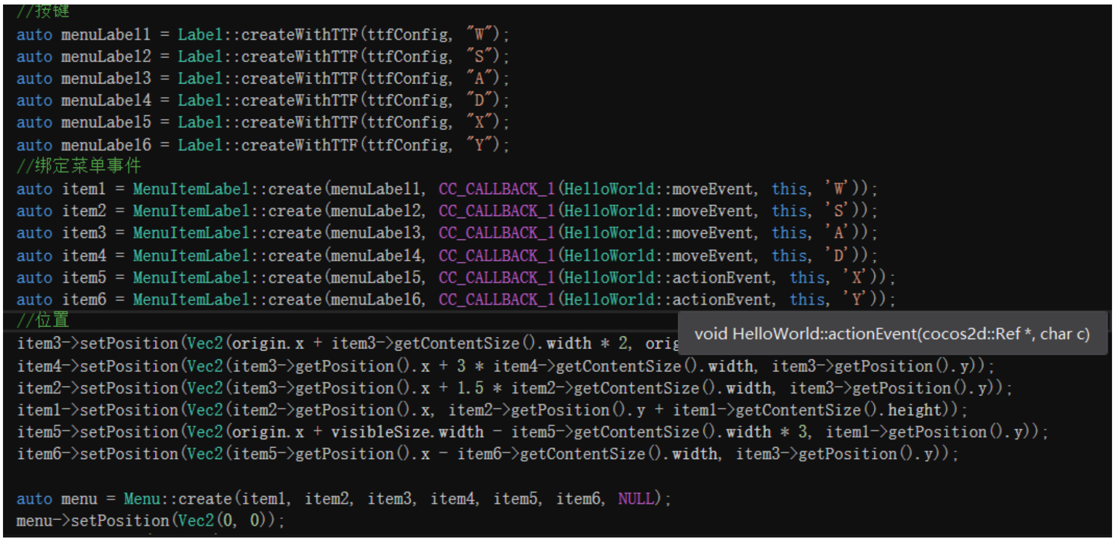
一．参考资料

http://bbs.csdn.net/topics/340143691

ppt

二．实验步骤

首先实现 6 个按键

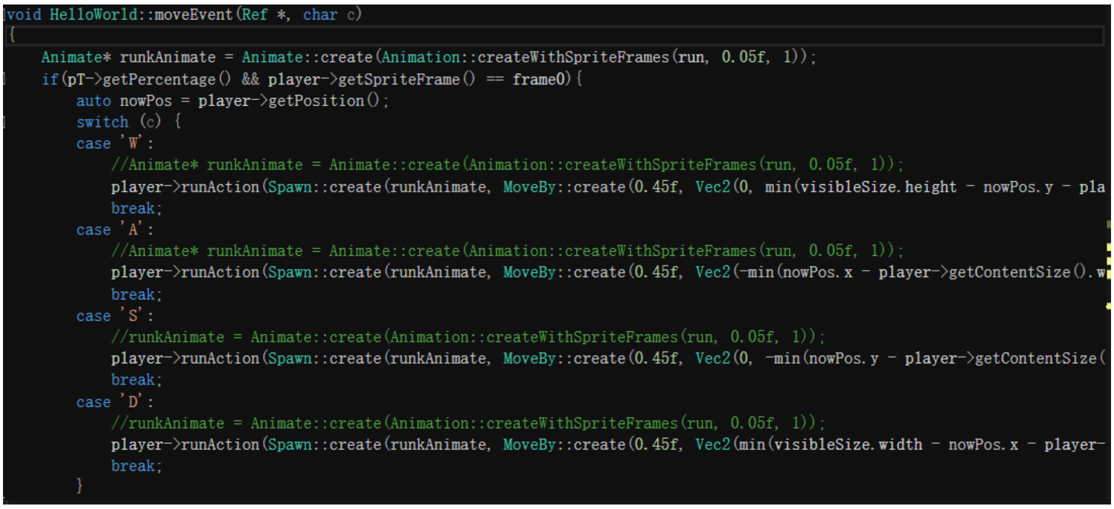


```
//按键
auto menuLabel1 = Label::createWithTTF(ttfConfig, "W");
auto menuLabel2 = Label::createWithTTF(ttfConfig, "S");
auto menuLabel3 = Label::createWithTTF(ttfConfig, "A");
auto menuLabel4 = Label::createWithTTF(ttfConfig, "D");
auto menuLabel5 = Label::createWithTTF(ttfConfig, "X");
auto menuLabel6 = Label::createWithTTF(ttfConfig, "Y");
//绑定菜单事件
auto item1 = MenuItemLabel::create(menuLabel1, CC_CALLBACK_1(HelloWorld::moveEvent, this, 'W'));
auto item2 = MenuItemLabel::create(menuLabel2, CC_CALLBACK_1(HelloWorld::moveEvent, this, 'S'));
auto item3 = MenuItemLabel::create(menuLabel3, CC_CALLBACK_1(HelloWorld::moveEvent, this, 'A'));
auto item4 = MenuItemLabel::create(menuLabel4, CC_CALLBACK_1(HelloWorld::moveEvent, this, 'D'));
auto item5 = MenuItemLabel::create(menuLabel5, CC_CALLBACK_1(HelloWorld::actionEvent, this, 'X'));
auto item6 = MenuItemLabel::create(menuLabel6, CC_CALLBACK_1(HelloWorld::actionEvent, this, 'Y'));
//位置
item3->setPosition(Vec2(origin.x + item3->getContentSize().width * 2, orig
item4->setPosition(Vec2(item3->getPosition().x + 3 * item4->getContentSize().width, item3->getPosition().y));
item2->setPosition(Vec2(item3->getPosition().x + 1.5 * item2->getContentSize().width, item3->getPosition().y));
item1->setPosition(Vec2(item2->getPosition().x, item2->getPosition().y + item1->getContentSize().height));
item5->setPosition(Vec2(origin.x + visibleSize.width - item5->getContentSize().width * 3, item1->getPosition().y));
item6->setPosition(Vec2(item5->getPosition().x - item6->getContentSize().width, item3->getPosition().y));

auto menu = Menu::create(item1, item2, item3, item4, item5, item6, NULL);
menu->setPosition(Vec2(0, 0));
```

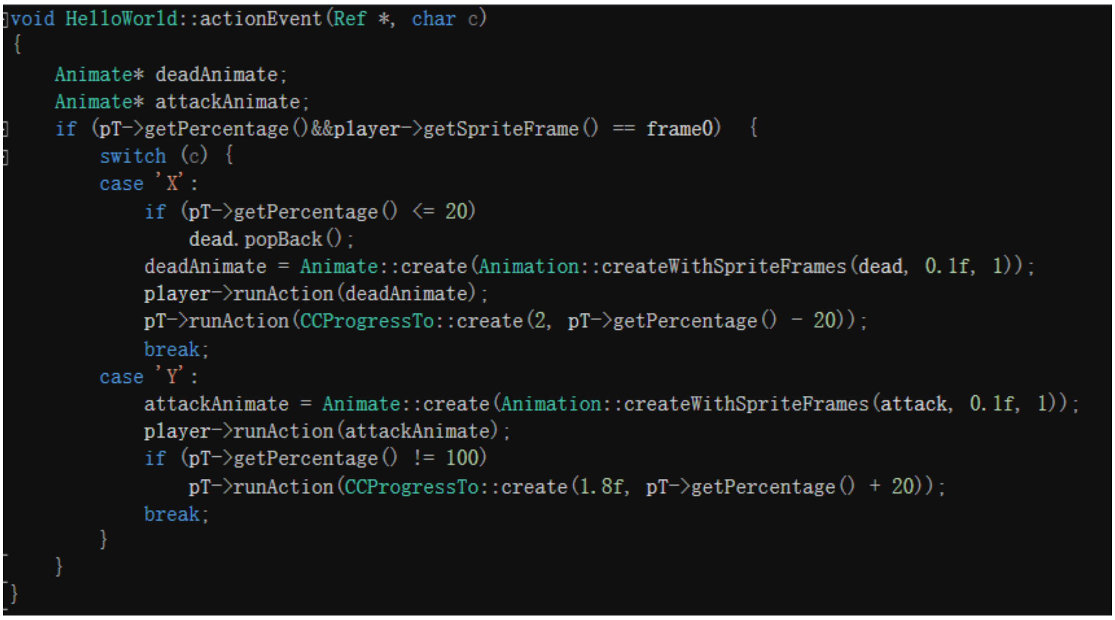
void HelloWorld::actionEvent(cocos2d::Ref *, char c)

并且实现他们的事件：

移动事件也就是当 hp 还有而且帧数回到最初的时候可以移动，然后将动画跟移动用 Spawn 组合在一起，攻击死亡事件就是让 player 与进度条都执行动画

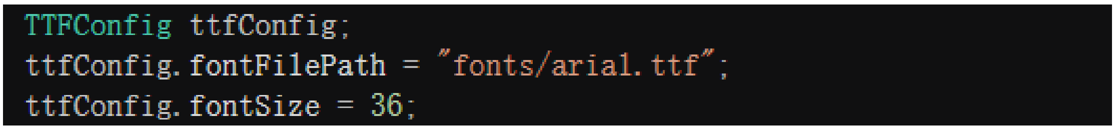


```
void HelloWorld::moveEvent(Ref *, char c)
{
    Animate* runkAnimate = Animate::create(Animation::createWithSpriteFrames(run, 0.05f, 1));
    if(pT->getPercentage() && player->getSpriteFrame() == frame0){
        auto nowPos = player->getPosition();
        switch (c) {
        case 'W':
            //Animate* runkAnimate = Animate::create(Animation::createWithSpriteFrames(run, 0.05f, 1));
            player->runAction(Spawn::create(runkAnimate, MoveBy::create(0.45f, Vec2(0, min(visibleSize.height - nowPos.y - pla
            break;
        case 'A':
            //Animate* runkAnimate = Animate::create(Animation::createWithSpriteFrames(run, 0.05f, 1));
            player->runAction(Spawn::create(runkAnimate, MoveBy::create(0.45f, Vec2(-min(nowPos.x - player->getContentSize().w
            break;
        case 'S':
            //runkAnimate = Animate::create(Animation::createWithSpriteFrames(run, 0.05f, 1));
            player->runAction(Spawn::create(runkAnimate, MoveBy::create(0.45f, Vec2(0, -min(nowPos.y - player->getContentSize(
            break;
        case 'D':
            //runkAnimate = Animate::create(Animation::createWithSpriteFrames(run, 0.05f, 1));
            player->runAction(Spawn::create(runkAnimate, MoveBy::create(0.45f, Vec2(min(visibleSize.width - nowPos.x - player-
            break;
        }
```



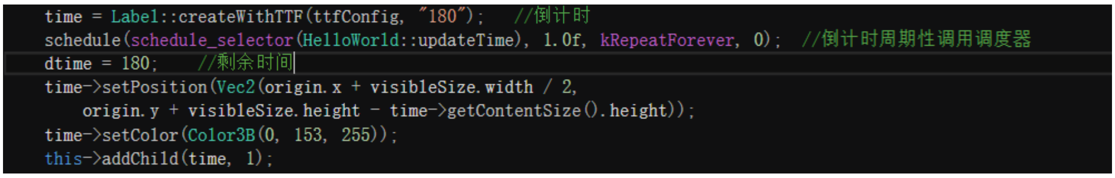
```
void HelloWorld::actionEvent(Ref *, char c)
{
    Animate* deadAnimate;
    Animate* attackAnimate;
    if (pT->getPercentage()&&player->getSpriteFrame() == frame0)  {
        switch (c) {
        case 'X':
            if (pT->getPercentage() <= 20)
                dead.popBack();
            deadAnimate = Animate::create(Animation::createWithSpriteFrames(dead, 0.1f, 1));
            player->runAction(deadAnimate);
            pT->runAction(CCProgressTo::create(2, pT->getPercentage() - 20));
            break;
        case 'Y':
            attackAnimate = Animate::create(Animation::createWithSpriteFrames(attack, 0.1f, 1));
            player->runAction(attackAnimate);
            if (pT->getPercentage() != 100)
                pT->runAction(CCProgressTo::create(1.8f, pT->getPercentage() + 20));
            break;
        }
    }
}
```

字体要求：

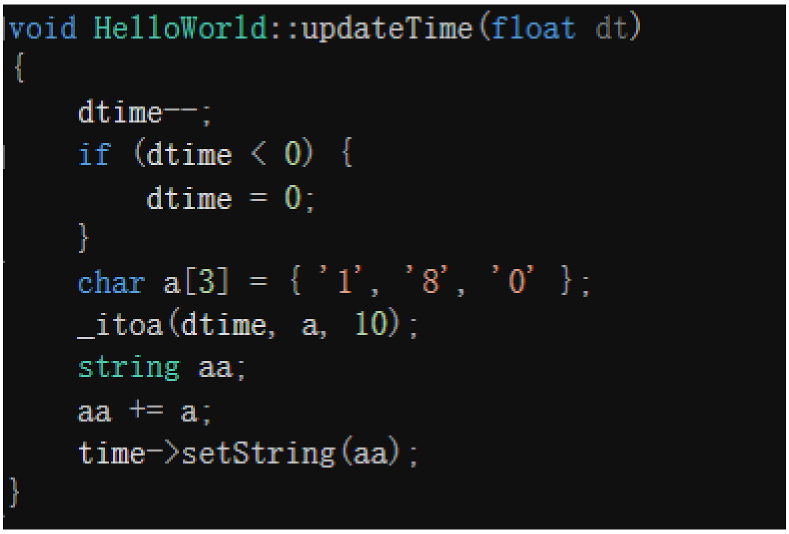


```
TTFConfig ttfConfig;
ttfConfig.fontFilePath = "fonts/arial.ttf";
ttfConfig.fontSize = 36;
```

倒计时：

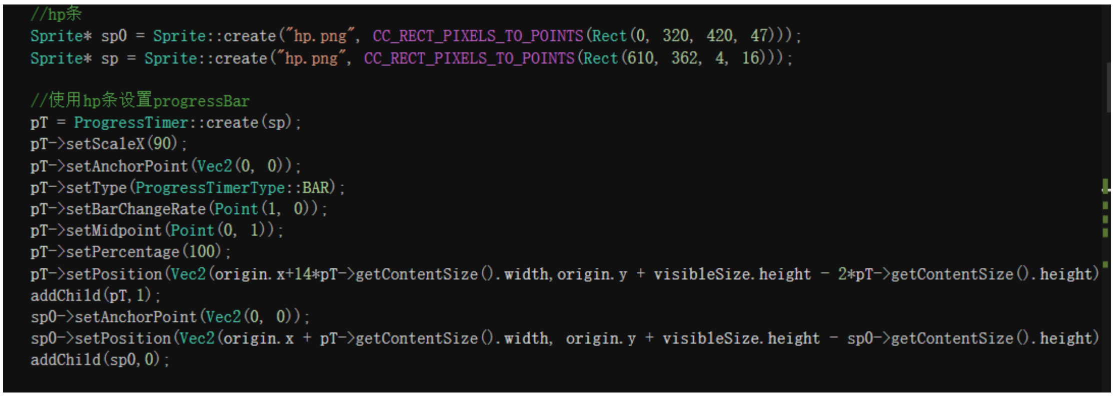


```
    time = Label::createWithTTF(ttfConfig, "180");   //倒计时
    schedule(schedule_selector(HelloWorld::updateTime), 1.0f, kRepeatForever, 0);  //倒计时周期性调用调度器
    dtime = 180;    //剩余时间
    time->setPosition(Vec2(origin.x + visibleSize.width / 2,
        origin.y + visibleSize.height - time->getContentSize().height));
    time->setColor(Color3B(0, 153, 255));
    this->addChild(time, 1);
```



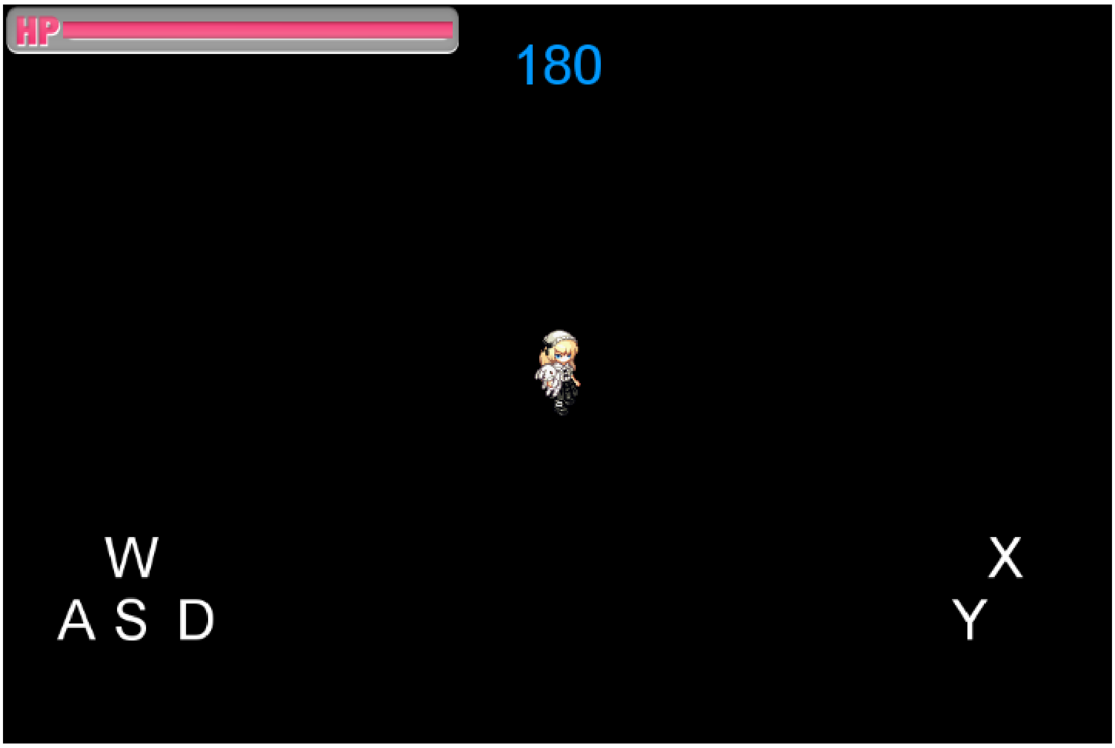
```
void HelloWorld::updateTime(float dt)
{
    dtime--;
    if (dtime < 0) {
        dtime = 0;
    }
    char a[3] = { '1', '8', '0' };
    _itoa(dtime, a, 10);
    string aa;
    aa += a;
    time->setString(aa);
}
```

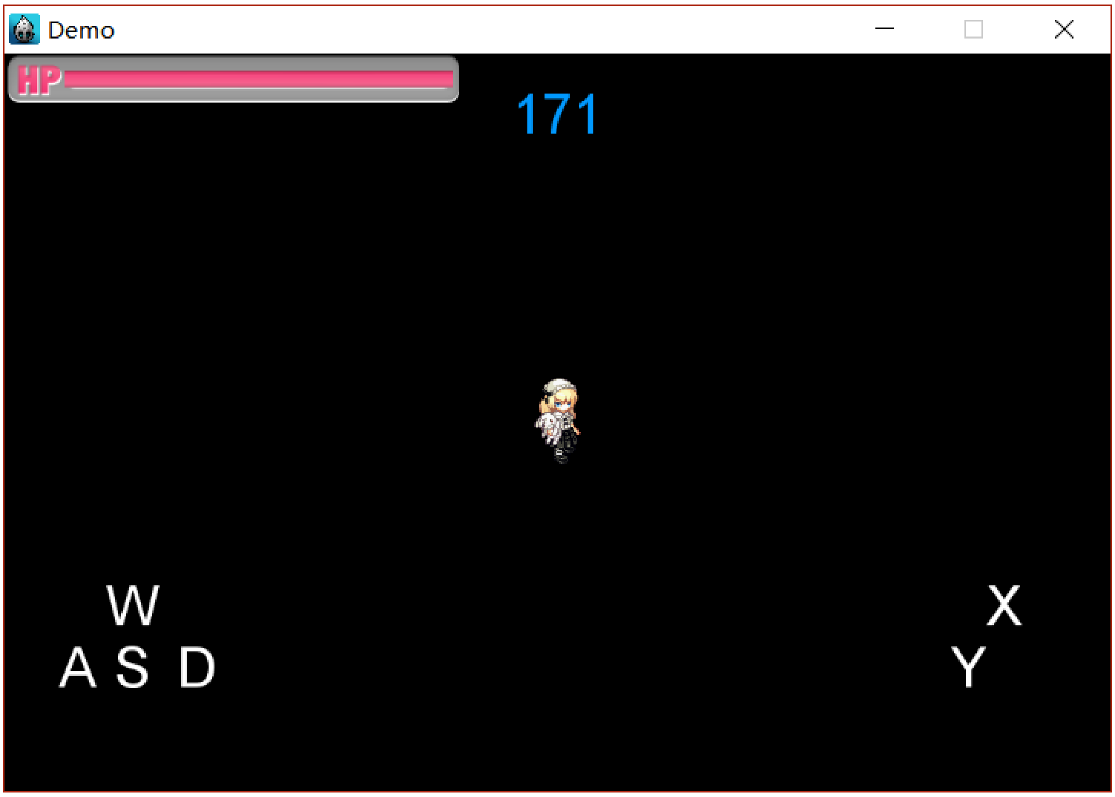
人物血条（与 Demo 一样）

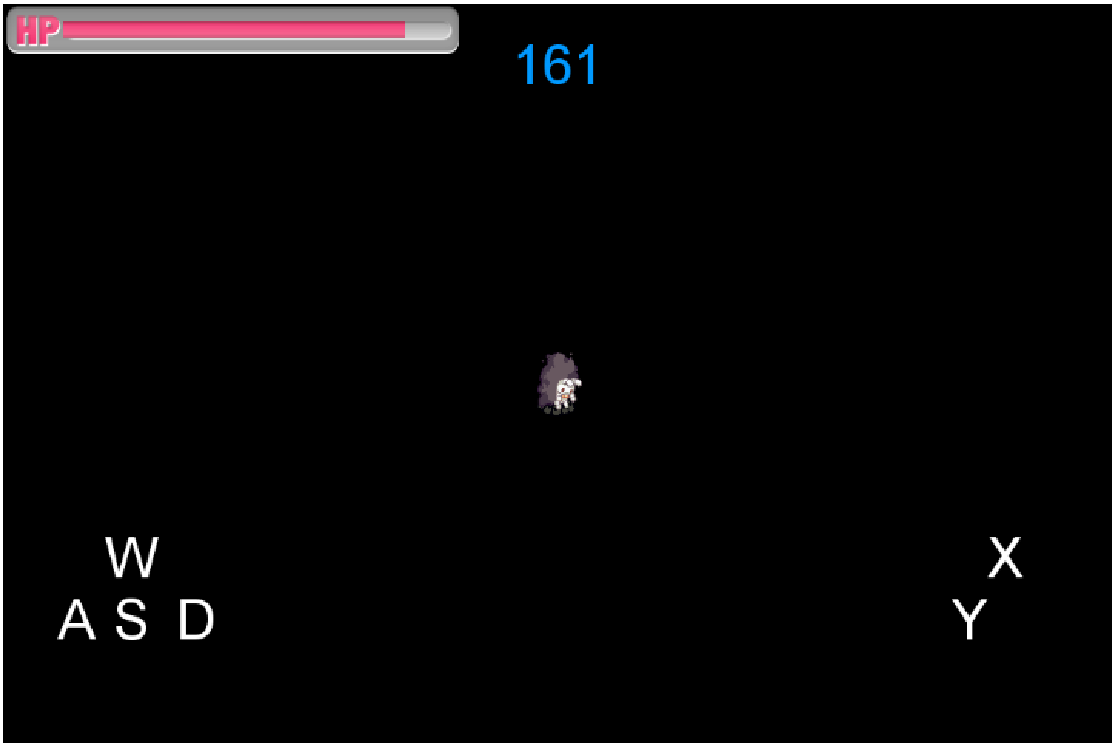


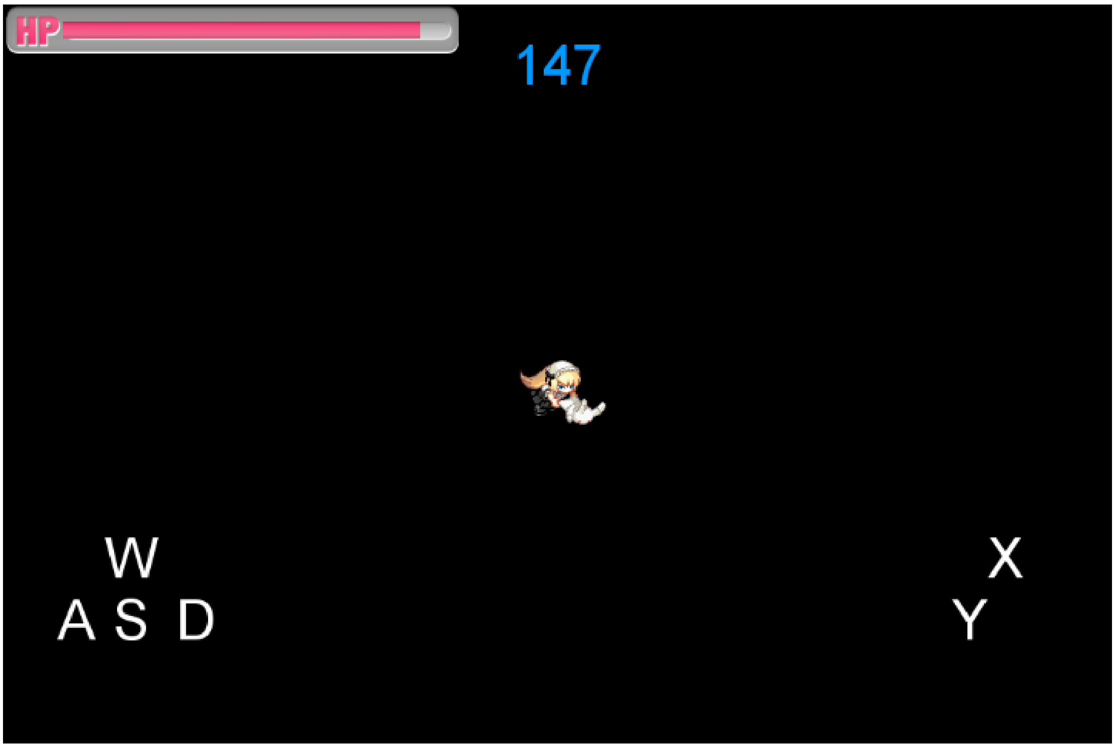
```
    //hp条
    Sprite* sp0 = Sprite::create("hp.png", CC_RECT_PIXELS_TO_POINTS(Rect(0, 320, 420, 47)));
    Sprite* sp = Sprite::create("hp.png", CC_RECT_PIXELS_TO_POINTS(Rect(610, 362, 4, 16)));

    //使用hp条设置progressBar
    pT = ProgressTimer::create(sp);
    pT->setScaleX(90);
    pT->setAnchorPoint(Vec2(0, 0));
    pT->setType(ProgressTimerType::BAR);
    pT->setBarChangeRate(Point(1, 0));
    pT->setMidpoint(Point(0, 1));
    pT->setPercentage(100);
    pT->setPosition(Vec2(origin.x+14*pT->getContentSize().width,origin.y + visibleSize.height - 2*pT->getContentSize().height)
    addChild(pT,1);
    sp0->setAnchorPoint(Vec2(0, 0));
    sp0->setPosition(Vec2(origin.x + pT->getContentSize().width, origin.y + visibleSize.height - sp0->getContentSize().height)
    addChild(sp0,0);
```

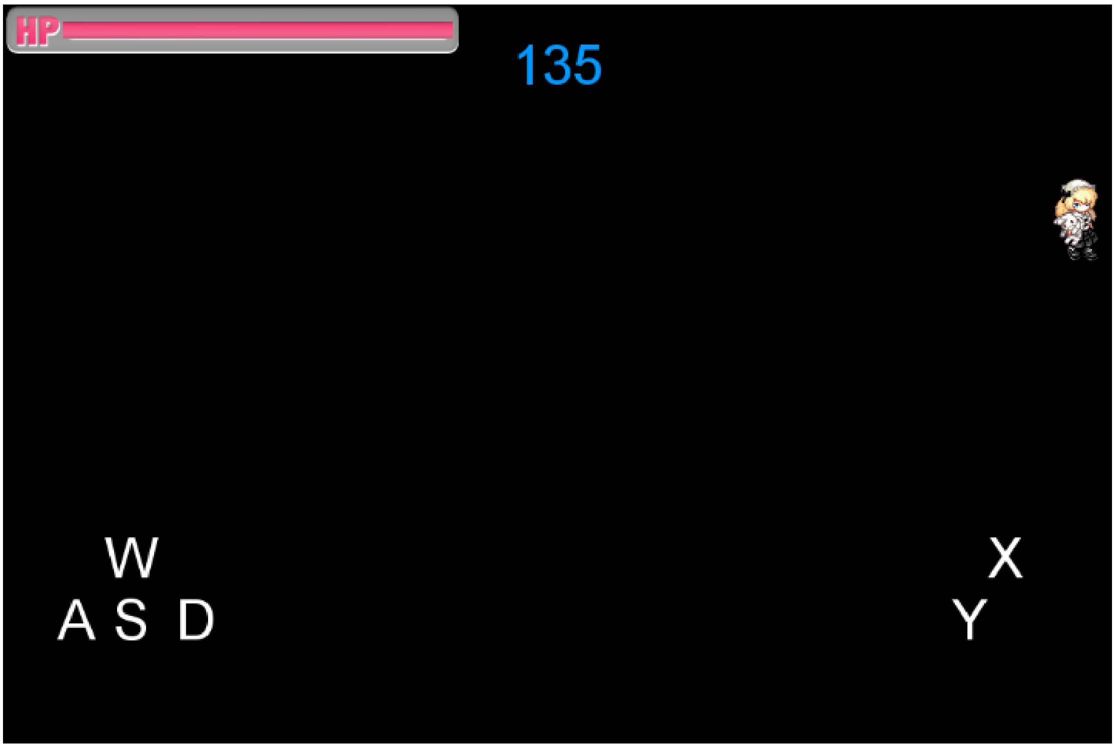
三．实验结果截图




Demo
HP
171
W
A S D
X
Y


HP
161
W
A S D
X
Y





四．实验过程遇到的问题

1. **实现不会移动到窗口外问题：**

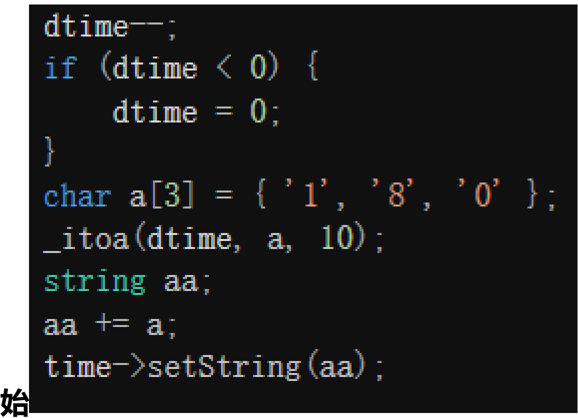
   一开始是用如果移动到窗口外就直接置于边缘，可行但是繁琐，然后

一位大神告诉我可以用 min(visibleSize.height - nowPos.y - player->getContentSize().height / 2, 50.0f)类似这样的做法来实现

2. 动画同时播放问题：

看到 Demo 中有 dead.pushBack(frame0);类似的操作，我就明白可以用第 0 帧来检测是否完成动作，所以在做动作时检测帧数即可

3. 倒计时的字符显示：



```
dtime--;
if (dtime < 0) {
    dtime = 0;
}
char a[3] = { '1', '8', '0' };
_itoa(dtime, a, 10);
string aa;
aa += a;
time->setString(aa);
```

一开始会报错，提示 a 的堆栈错误什么的，即使我初始化还是报错，百度之下发现我的写法没错而是编译器的问题：http://bbs.csdn.net/topics/340143691，

把 project->配置属性->c/c++->代码生成->基本运行时检查 为 默认值就不会报错

五．思考与总结

这次的作业还是比较难的，但是 Demo 有很多可参考的代码，基本可以实现基础代码，至于那些进阶的例如不可走出窗口外，不能同时做两个动作则是慢慢的摸索。在这过程中加深了对动画的理解，对精灵动作的理解，是序列还是同时进行，对动画帧数，都有了一定的掌握