

现代操作系统应用开发实验报告

姓名：_____刘宇庭_____

学号：_____16340158_____

实验名称：__cocos2d-x 学习__

一、参考资料

1. cocos2d-x 文件目录

① <http://blog.51cto.com/shahdza/1549803>

2. 本地坐标与世界坐标相互转换

① <http://blog.163.com/zjfto/blog/static/201429061201292193855498/>

② <http://www.cnblogs.com/lyout/p/3292702.html>

3. 调度器 Scheduler

① <https://blog.csdn.net/wlk1229/article/details/55683358>

4. Progress Timer

① <https://blog.csdn.net/u010105970/article/details/40183719>

5. 回调函数

① <https://blog.csdn.net/we000636/article/details/8438036>

② <https://blog.csdn.net/marpools/article/details/24526223>

二、实验步骤

1. Hw9

- ① cocos2d-x 安装配置
- ② 学习 cocos2d-x 文件目录
- ③ 新建 HelloCocos 项目，制作自己的 HelloWorld 界面

2. Hw10

- ① 完成菜单场景的页面布局
- ② 实现菜单场景切换到游戏场景
- ③ 完成游戏场景的页面布局和功能

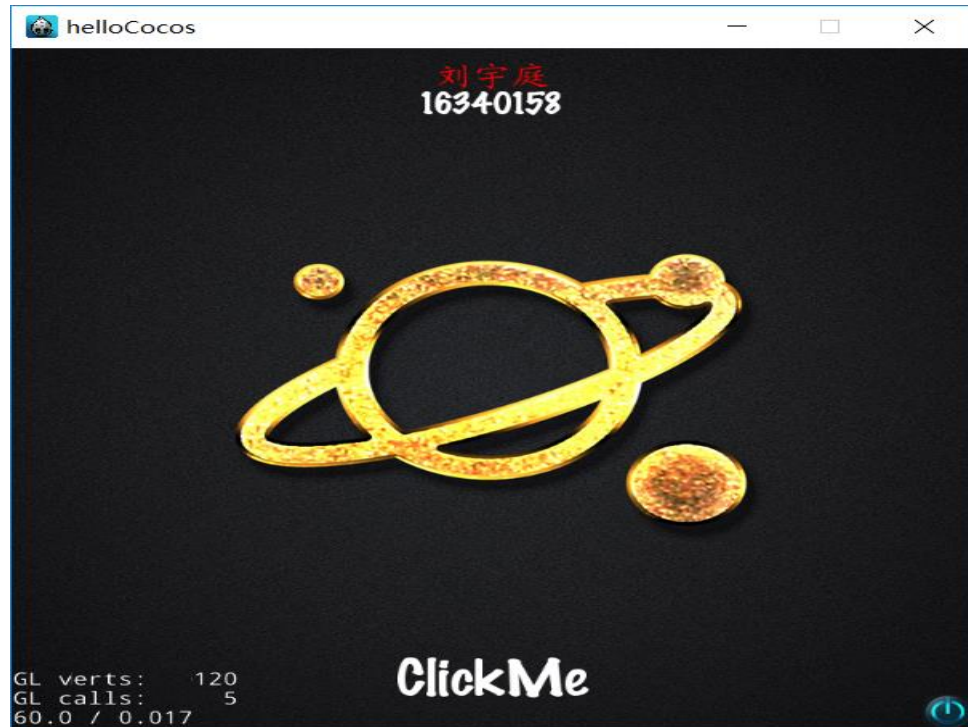
3. Hw11

- ① 学习调度器的使用，完成倒计时功能
- ② 按要求设计页面布局
- ③ 完成各个动作的实现函数

三、关键步骤截图

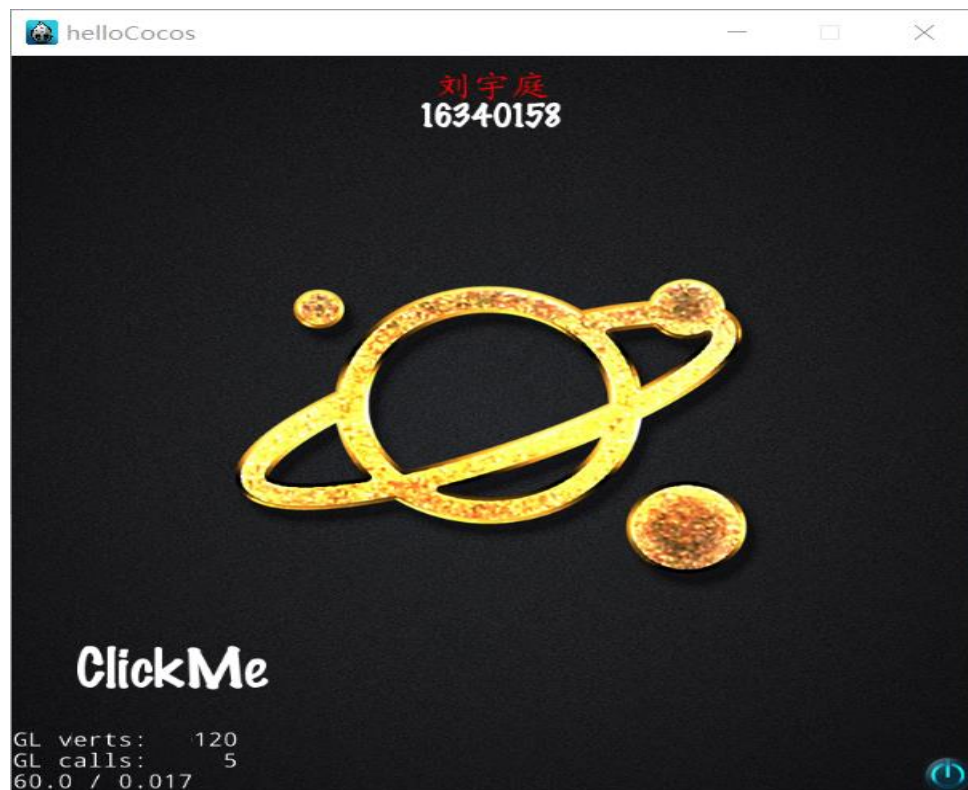
1. Hw9

- ③ 界面展示

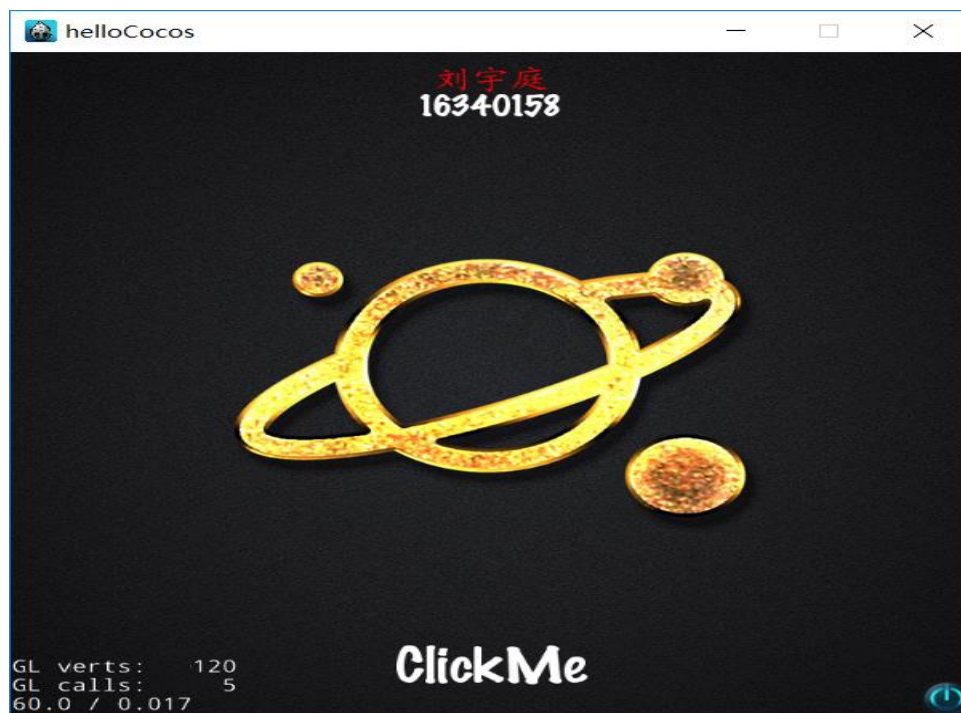


- ④ 简单的触发事件(点击 ClickMe 标签, 标签移动, 再次点击回到原位)

点击: (标签移动)



再次点击：(标签回到原位)

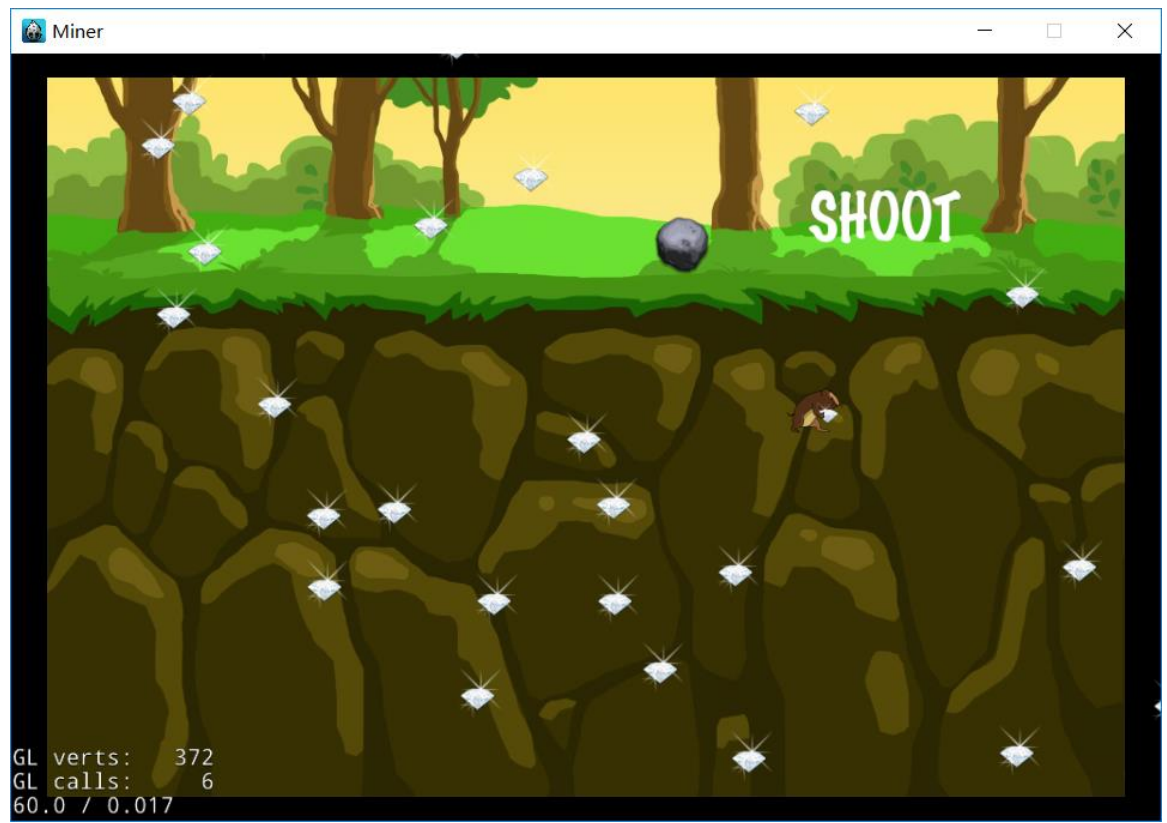
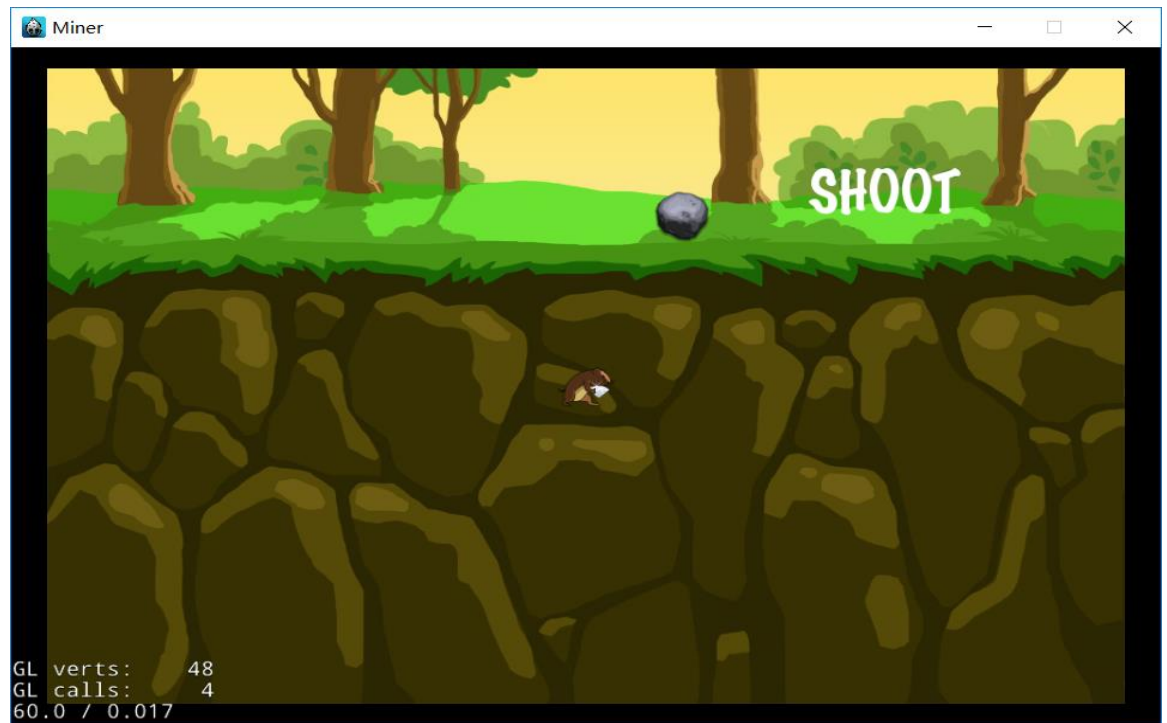


2. Hw10

① 菜单页面：(动画：矿工嘴巴、山洞灯光效果)

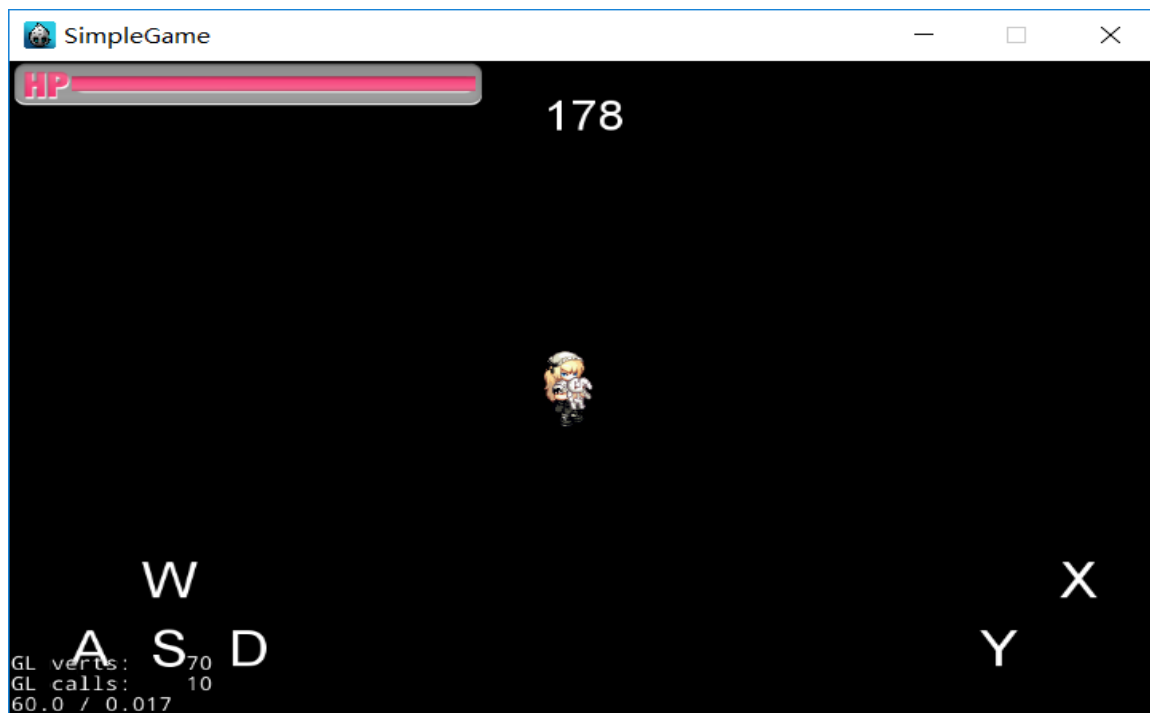


② 游戏页面（具体要求都已经实现）

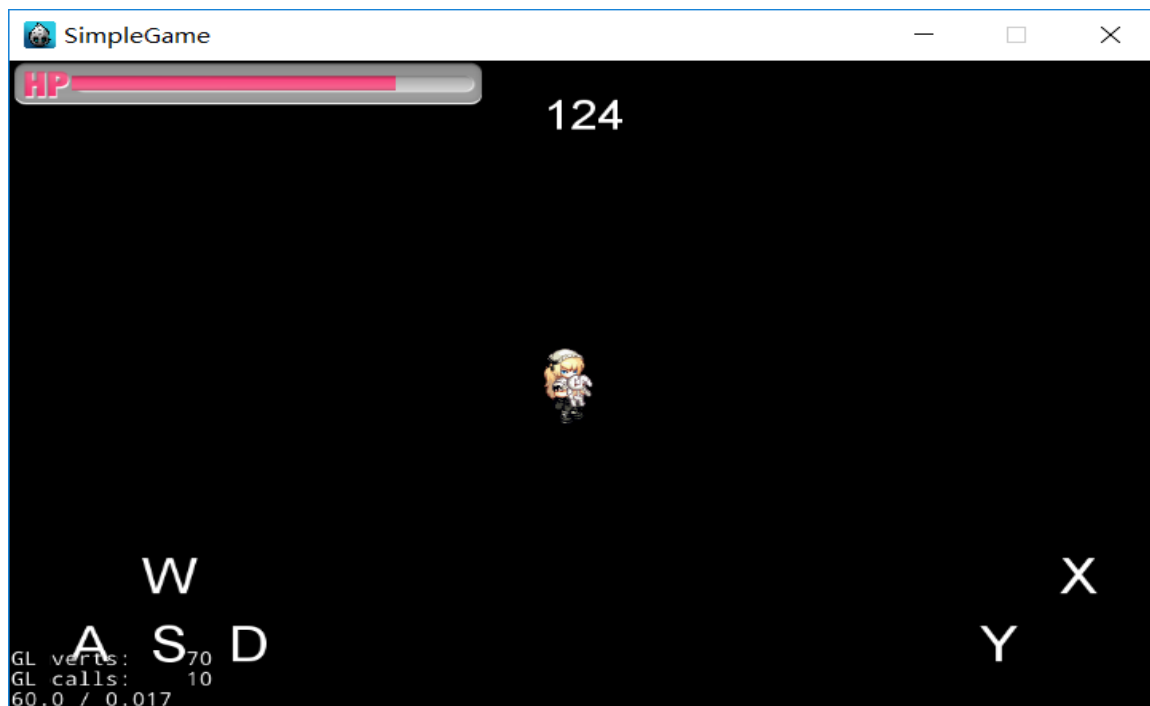


3. Hw11

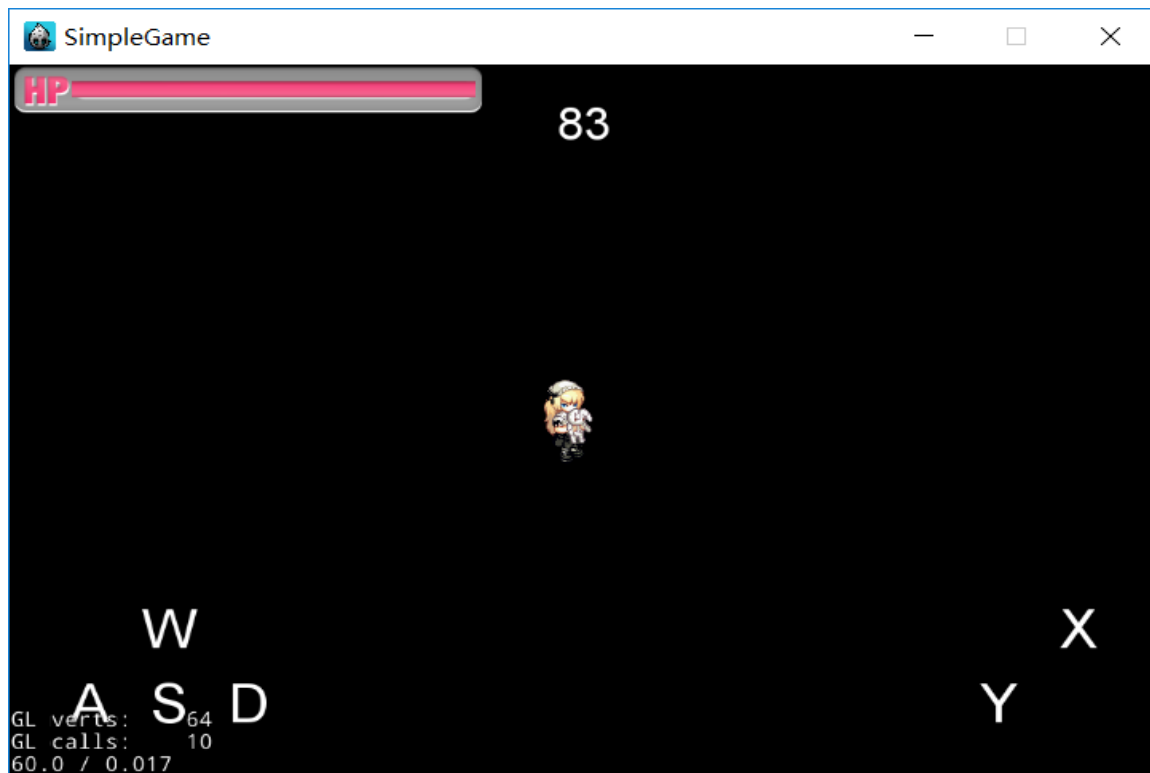
① 初始页面



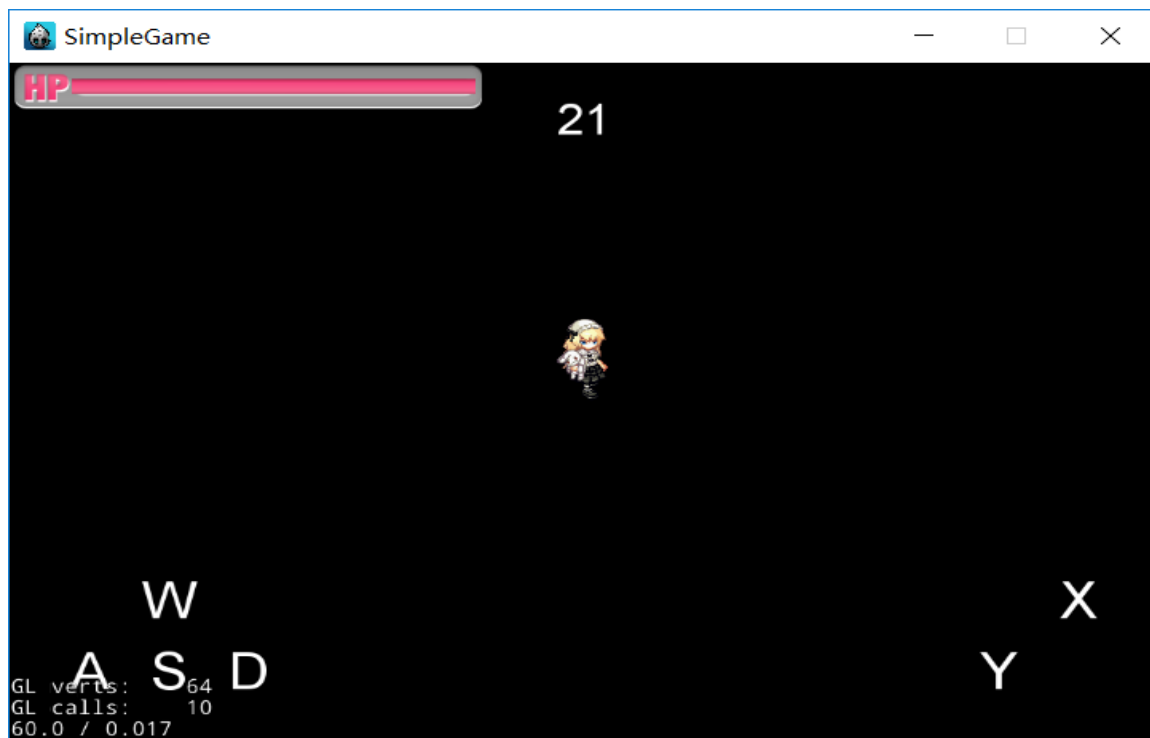
② 点击按钮 X (扣血、人物有动画)



③ 点击按钮 Y (加血、人物有动画)



④ 移动 (展示为上移, 过程有动画)



四、遇到的问题

1. cocos 中文显示不支持，hw9 中名字显示不出来

解决方法：用 Dictionary 解析 xml 文件，并且上网下载了新的字体 [GB2312](#)

主要实现代码：

```
1. CCDictionary* message = CCDictionary::createWithContentsOfFile("info.xml")
   //读取 xml 文件，文件在 Resources 目录下
2. //name
3. auto name = message->valueForKey("name");    //根据 key，获取 value
4. const char* myName = name->getCString();    //将 value 转化为字符串
5. auto nameLabel = Label::createWithTTF(myName, "fonts/GB2312.ttf", 24);
6. nameLabel->setColor(Color3B(255, 0, 0));
7. //id
8. auto id = message->valueForKey("id");    //根据 key，获取 value
9. const char* myId = id->getCString();    //将 value 转化为字符串
10. auto idLabel = Label::createWithTTF(myId, "fonts/Marker Felt.ttf", 24);
```

2. hw10 中世界坐标与本地坐标互相转换不熟悉

世界坐标本地坐标的互相转换不熟悉，四个转化函数不是很懂，经过不断测试，现在算是大概了解二者之间的转换

3. hw11 中如何获取当前标签内容并修改（计时器的实现）

主要实现代码：

```
1. // 时间倒计时
2. void HelloWorld::updateTime(float dt) {
3.     int leftTime = std::atoi(time->getString().c_str());
4.     if (leftTime > 0) {
5.         --leftTime;
6.         std::string tmp = Value(leftTime).asString();
7.         time->setString(tmp);
8.     }
9.     else
```



```
10.         time->setString("TIME OUT");
11. }
```

4. hw11 中 X、Y 两个按钮的动画不能同时播放

解决方法: 实现 X、Y 动作互斥, 设置一个判定变量 acting, 当其为 false 的时候, X、Y 动作都可以执行, 当某个动作正在进行中, acting 将被设为 true, 表示正在执行 X 或 Y 动作, 此时点击按钮 X、Y 无效, 必须等待当前动画结束后 acting 被设置为 false 以后, 二者才重新生效。

实现代码:

在类中添加一个新的成员:

```
1. bool acting;
```

X 回调函数:

```
1. // 死亡
2. void HelloWorld::Xfun(cocos2d::Ref * pSpende)
3. {
4.     if (!acting) {
5.         acting = true;
6.         Animate* deadAnimate = Animate::create(AnimationCache::getInstance(
7.             )->getAnimation("deadAnimation"));
8.         auto set = CCCallFunc::create([this]() {
9.             log("x");
10.            acting = false;
11.        });
12.        CCSequence* deadSeq = CCSequence::create(deadAnimate, set, NULL);
13.        player->runAction(deadSeq);
14.
15.        // 减血
16.        int num = pT->getPercentage();
17.        if (num > 0) {
18.            CCProgressTo* progressTo = CCProgressTo::create(1.5f, num - 20);
19.            pT->runAction(progressTo);
20.        }
21.    }
22. }
```

```
19.     }
20. }
21. }
```

Y 回调函数:

```
1. // 攻击
2. void HelloWorld::Yfun(cocos2d::Ref * pSpender)
3. {
4.     if (!acting) {
5.         acting = true;
6.         Animate* attackAnimate = Animate::create(AnimationCache::getInstance()->getAnimation("attackAnimation"));
7.         auto set = CCCallFunc::create([this]() {
8.             log("y");
9.             acting = false;
10.        });
11.        CCSequence* attackSeq = CCSequence::create(attackAnimate, set, NULL);
12.        player->runAction(attackSeq);
13.
14.
15.        // 加血
16.        int num = pT->getPercentage();
17.        if (num < 100) {
18.            CCProgressTo* progressTo = CCProgressTo::create(1.5f, num + 20);
19.            pT->runAction(progressTo);
20.        }
21.    }
22. }
```

五、思考与总结

这三周的作业总体来说还是比较简单的, bonus 三周的也都完成, 这三周主要学习了一下 cocos 的一些基本内容, 如何创建精灵、添加监听事件、设置回调函数、调度器的使用以及 cocos 的坐标系学习, 感觉做游戏还是比较有意思的吧,

同时感觉 cocos 里面还是有很多比较新的东西，有很多 cocos 内部的方法不会，
还是要继续加油吧。