

现代操作系统应用开发实验报告

学号： 14331098

班级： 周四班

姓名： 黄建武

实验名称： HW10

一 . 参考资料

ProgressBar 的操作

<http://shahdza.blog.51cto.com/2410787/1546707>

游戏素材

<http://pic.66rpg.com>

lambda 表达式的使用

<http://www.cocoachina.com/bbs/read.php?tid=195070>

键盘监听事件

http://blog.csdn.net/tonny_guan/article/details/38149367

回调函数的使用

<http://blog.csdn.net/elloop/article/details/50438032>

将 cocos-2d 项目导出为 exe

<http://www.apkbus.com/thread-165341-1-1.html>

二 . 实验步骤

- ① 阅读 Demo Classes 的源代码，理解 Demo 的代码和思路。
- ② 阅读作业要求的 PDF，将作业分成若干部分完成；

- ③ 运行 Demo 的代码发现窗口大小发生变化时，屏幕上的东西位置和大小都发生很大的变化，于是对 Demo 进行大修改，固定位置，更改计时器样式，并添加了背景；
- ④ 添加倒计时；
- ⑤ 实现点击 X,Y 播放动画；
- ⑥ 实现血槽的加减动画效果；
- ⑦ 实现点击 WASD 控制主角移动；
- ⑧ 实现角色不会移动到可视窗口外的功能；
- ⑨ 添加键盘输入监听，键盘也可控制移动播放动画；
- ⑩ 添加功能：不能同时运行多个动画，血槽为空或时间为 0 时停止一切动画，出现 gameover；
- ⑪ 进行最后的调试；

三．实验结果截图

开始界面，添加背景图片，将背景进行了缩放，适应屏幕大小，窗口大小发生改变，血槽时间精灵按键都位置都不会发生变化。



点击 WASD 控制主角移动，添加了按键输入，使用键盘上的 WASD 和个↓←→都可以控制主角移动

```
// 键盘响应事件
auto keyListener = EventListenerKeyboard::create();
keyListener->onKeyPressed = [this](EventKeyboard::KeyCode keyCode, Event* event){ ... }
this->getEventDispatcher()->addEventListenerWithSceneGraphPriority(keyListener, this);
```

点击 X 或使用键盘输入 X 播放动画



点击 Y 或使用键盘输入 Y 播放动画



角色不会移动到可视窗口外



时间到主角 HP 减到 0，显示 gameover，停止一切动画



时间为 0 时，将主角 HP 减为 0，显示 gameover，停止一切动画



四．实验过程遇到的问题

- ① 窗口放大或缩小，时间和血槽等位置会发生变动。



运行上次的黄金矿工发现不会出现这种问题，对比代码后发现项目默认设计区域没有边界，修改为 EXACT_FIT 后位置不会随窗口大小而改变。

```
// Set the design resolution
//glview->setDesignResolutionSize(designResolutionSize.width, designResolutionSize.height,
    ResolutionPolicy::NO_BORDER);
glview->setDesignResolutionSize(designResolutionSize.width, designResolutionSize.height,
    ResolutionPolicy::EXACT_FIT);
```

- ② 窗口大小改变精灵的位置虽然不会改变，可是血槽的长度却没变短



仔细阅读血槽部分的代码后才发现血槽是有一张小图片拉伸 90 倍形成的，于是改为根据屏幕大小拉伸相同的比例

```
//使用hp条设置progressBar  
auto scale = visibleSize.width / 2.4 / sp0->getContentSize().width;  
pT = ProgressTimer::create(sp);  
pT->setScaleY(scale);  
pT->setScaleX(88.8 * scale);
```

五 . 思考与总结

- ① 把程序分解成一个个小的部分，分而治之，更有效率而且更容易排错。
- ② 看似简单的一个小游戏，就几个功能，可是实现起来还是遇到很多问题，很多操作不懂，只能上网搜索，经过这次作业对帧动画和调度器也有了更深的了解。