

现代操作系统应用开发实验报告

姓名：谷雨

学号：16341005

实验名称：cocos 学习

一、参考资料

1. cocos2dx 学习笔记之菜单 Menu 和菜单项 MenuItem

<https://blog.csdn.net/ellis1970/article/details/79184357>

2. cocos2d3.10 简单引用 lambda 表达式用作回调函数

<https://blog.csdn.net/freshstraw/article/details/52672947>

3. Cocos2d-x 3.0 开发（三）点击交互的四种处理

<https://blog.csdn.net/fansongy/article/details/12716671>

二、实验步骤

请在这里简要写下你的实验过程。

第九周：

先进行 python 的下载安装。

开始安装时由于没有勾选添加路径的选择，没有 python 命令。

于是我运行第二遍安装程序才完成。

接下来我进行了示例的测试以及 helloWorld 的完成。

还完成了 bonus

1. 读取 xml

```
CCDictionary* message = CCDictionary::createWithContentsOfFile("tar.xml");    //读取xml文件, 文件在Resources目录下
auto name = message->valueForKey("name");
auto num = message->valueForKey("num");
log((name->_string + '\n' + num->_string).c_str());
Label* label = Label::createWithSystemFont(name->_string + '\n' + num->_string, "Arial", 17);
//auto label = Label::createWithTTF(b, "fonts/arial.ttf", 24);
if (label == nullptr)
{
    problemLoading("fonts/arial.ttf");
}
```

2. 更换图片

```
auto sprite = Sprite::create("atom.jpg");
sprite->setPosition(50, 50);
sprite->setAnchorPoint(Vec2(0.5, 0.5));
this->addChild(sprite); // 添加到层
auto act = ScaleTo::create(0, 0.2);
sprite->runAction(act);
```

3. Eventlistener

```
//通过 lambda 表达式 直接实现触摸事件的回调方法
listener1->onTouchBegan = [=](Touch* touch, Event* event) {
    auto target = static_cast<Sprite*>(event->getCurrentTarget());

    Point locationInNode = target->convertToNodeSpace(touch->getLocation());
    Size s = target->getContentSize();
    Rect rect = Rect(0, 0, s.width, s.height);

    if (rect.containsPoint(locationInNode))
    {
        log("sprite began... x = %f, y = %f", locationInNode.x, locationInNode.y);
        target->setPosition(((double)(rand() % 10)) / 10 * visibleSize.width + origin.x,
                            ((double)(rand() % 10)) / 10 * visibleSize.height + origin.y);
        return true;
    }
    return false;
};
//将触摸事件绑定到精灵身上
_eventDispatcher->addEventListenerWithSceneGraphPriority(listener1, sprite);
```

第十周:

黄金矿工简陋版。

由于 ppt 要求的非常详细, 基本按照 ppt 的要求一步一步完成。

新版黄金矿工游戏，共有两个界面：**主界面与游戏界面**

主界面：在demo代码基础上完善场景，添加开始按钮（MenuItem），点击进入游戏界面。

游戏界面：两个Layer，StoneLayer锚点位于左下角，坐标设为(0,0)，其上有一石头精灵，初始坐标为（560，480）；MouseLayer锚点位于左下角，坐标设为(0,屏幕高度的一半)，其上有一老鼠精灵，初始坐标为（屏幕宽度的一半，0）。有一个Label，作为shoot按钮。

游戏要求：游戏开始后，点击屏幕任意位置，在该位置添加一块奶酪，老鼠跑到该位置吃掉奶酪；点击shoot按钮，石头发射到老鼠所在的位置，老鼠跑开，留下钻石。

加分项：尝试添加一两个动画

1. 创建菜单

```
auto title = Sprite::create("gold-miner-text.png");
addChild(title);
title->setPosition(visibleSize.width / 2, visibleSize.height * 3 / 4);

auto startItem = MenuItemImage::create(
    "start-0.png",
    "start-1.png",
    CC_CALLBACK_1(MenuScene::menuStartCallBack, this));
startItem->setPosition(Vec2(origin.x + 150, origin.y + 180));
auto menu = Menu::create(startItem, NULL);
menu->setPosition(Vec2::ZERO);
gold->addChild(menu, 0);

return true;
```

2. 创建动画

```
for (int i = 0; i < totalFrames; i++)
{
    sprintf(frameName, "gem-mouse-%d.png", i);
    mouseAnimation->addSpriteFrame(SpriteFrameCache::getInstance()->getSpriteFrameByName(frameName));
}

mouseAnimation->setDelayPerUnit(0.1);
AnimationCache::getInstance()->addAnimation(mouseAnimation, "mouseAnimation");

totalFrames = 6;
Animation* diamondAnimation = Animation::create();

for (int i = 0; i < totalFrames; i++)
{
    sprintf(frameName, "pulled-diamond-%d.png", i);
    diamondAnimation->addSpriteFrame(SpriteFrameCache::getInstance()->getSpriteFrameByName(frameName));
}

diamondAnimation->setDelayPerUnit(0.1);
AnimationCache::getInstance()->addAnimation(diamondAnimation, "diamondAnimation");
```

第十一周：

本周的要求没有上一周那么详细，自己思考的地方比较多。

因为觉得某些要求实现方式有很多反而有种无从下手的感觉。

后来经过仔细看 ppt 与同学交流，确定了感觉中比较合适的方案。

将虚拟按键作为菜单项，分别绑定事件。

移动类按钮一起处理

动作类按钮一起处理。

时间使用调度器完成。

关于运动范围的限定：开始时以为要用调度器判断，但却不知道如何实现。后来直接在点击时进行范围的判断。

操作失误，作业文件丢失，从头开始。

1. 创建虚拟按钮并且绑定事件

```
auto W = Label::createWithTTF("W", "fonts/arial.ttf", 36);
auto A = Label::createWithTTF("A", "fonts/arial.ttf", 36);
auto S = Label::createWithTTF("S", "fonts/arial.ttf", 36);
auto D = Label::createWithTTF("D", "fonts/arial.ttf", 36);
auto X = Label::createWithTTF("X", "fonts/arial.ttf", 36);
auto Y = Label::createWithTTF("Y", "fonts/arial.ttf", 36);
W->setName("W");
A->setName("A");
S->setName("S");
D->setName("D");
X->setName("X");
Y->setName("Y");

auto Btn_W = MenuItemLabel::create(W, CC_CALLBACK_0(HelloWorld::RunAction, this, W));
//Btn_W->setPosition(Vec2(origin.x + W->getWidth() * 2, origin.y + W->getHeight() * 3));
Btn_W->setPosition(Vec2(origin.x + W->getContentSize().width * 2, origin.y + W->getContentSize().height * 3));
Btn_W->setName("Btn_W");
auto Btn_S = MenuItemLabel::create(S, CC_CALLBACK_0(HelloWorld::RunAction, this, S));
//Btn_W->setPosition(Vec2(origin.x + W->getWidth() * 2, origin.y + W->getHeight() * 3));
Btn_S->setPosition(Vec2(origin.x + W->getContentSize().width * 2, origin.y + W->getContentSize().height * 2));
Btn_S->setName("Btn_S");
auto Btn_A = MenuItemLabel::create(A, CC_CALLBACK_0(HelloWorld::RunAction, this, A));
//Btn_W->setPosition(Vec2(origin.x + W->getWidth() * 2, origin.y + W->getHeight() * 3));
Btn_A->setPosition(Vec2(origin.x + W->getContentSize().width, origin.y + W->getContentSize().height * 2));
Btn_A->setName("Btn_A");
auto Btn_D = MenuItemLabel::create(D, CC_CALLBACK_0(HelloWorld::RunAction, this, D));
//Btn_W->setPosition(Vec2(origin.x + W->getWidth() * 2, origin.y + W->getHeight() * 3));
Btn_D->setPosition(Vec2(origin.x + W->getContentSize().width * 3, origin.y + W->getContentSize().height * 2));
Btn_D->setName("Btn_D");
auto Btn_X = MenuItemLabel::create(X, CC_CALLBACK_0(HelloWorld::DieAction, this, X));
//Btn_W->setPosition(Vec2(origin.x + W->getWidth() * 2, origin.y + W->getHeight() * 3));
Btn_X->setPosition(Vec2(origin.x + visibleSize.width - W->getContentSize().width, origin.y + W->getContentSize().height * 3));
Btn_X->setName("Btn_X");
auto Btn_Y = MenuItemLabel::create(Y, CC_CALLBACK_0(HelloWorld::AttackAction, this, Y));
//Btn_W->setPosition(Vec2(origin.x + W->getWidth() * 2, origin.y + W->getHeight() * 3));
Btn_Y->setPosition(Vec2(origin.x + visibleSize.width - W->getContentSize().width * 2, origin.y + W->getContentSize().height * 3));
```


2. 创建序列帧动画

```
// 死亡动画(帧数: 22帧, 高: 90, 宽: 79)
auto texture2 = Director::getInstance()->getTextureCache()->addImage("$lucia_dead.png");
die.reserve(22);
for (int i = 0; i < 22; i++) {
    auto frame = SpriteFrame::createWithTexture(texture2, CC_RECT_PIXELS_TO_POINTS(Rect(79 * i, 0, 79, 90)));
    die.pushBack(frame);
}

// 运动动画(帧数: 8帧, 高: 101, 宽: 68)
auto texture3 = Director::getInstance()->getTextureCache()->addImage("$lucia_forward.png");
run.reserve(8);
for (int i = 0; i < 8; i++) {
    auto frame = SpriteFrame::createWithTexture(texture3, CC_RECT_PIXELS_TO_POINTS(Rect(68 * i, 0, 68, 101)));
    run.pushBack(frame);
}
```

3. 血条变化

```
pT->runAction(ProgressTo::create(2, pT->getPercentage() - 20));
```

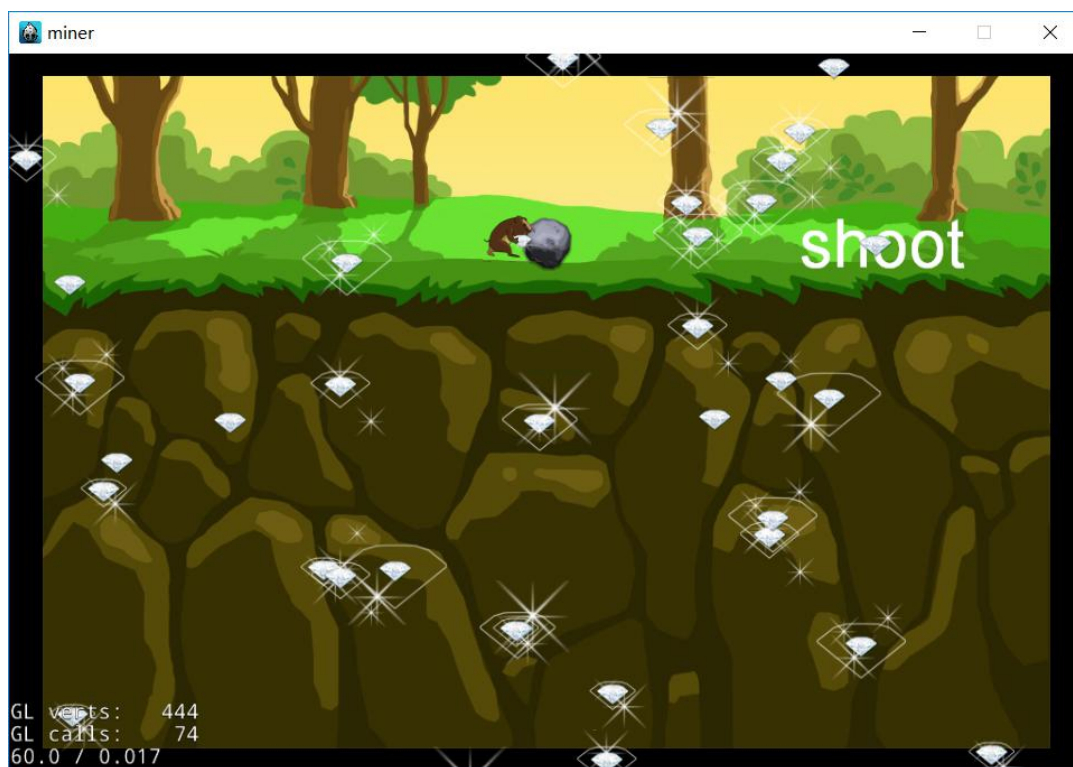
三、关键步骤截图

第九周:

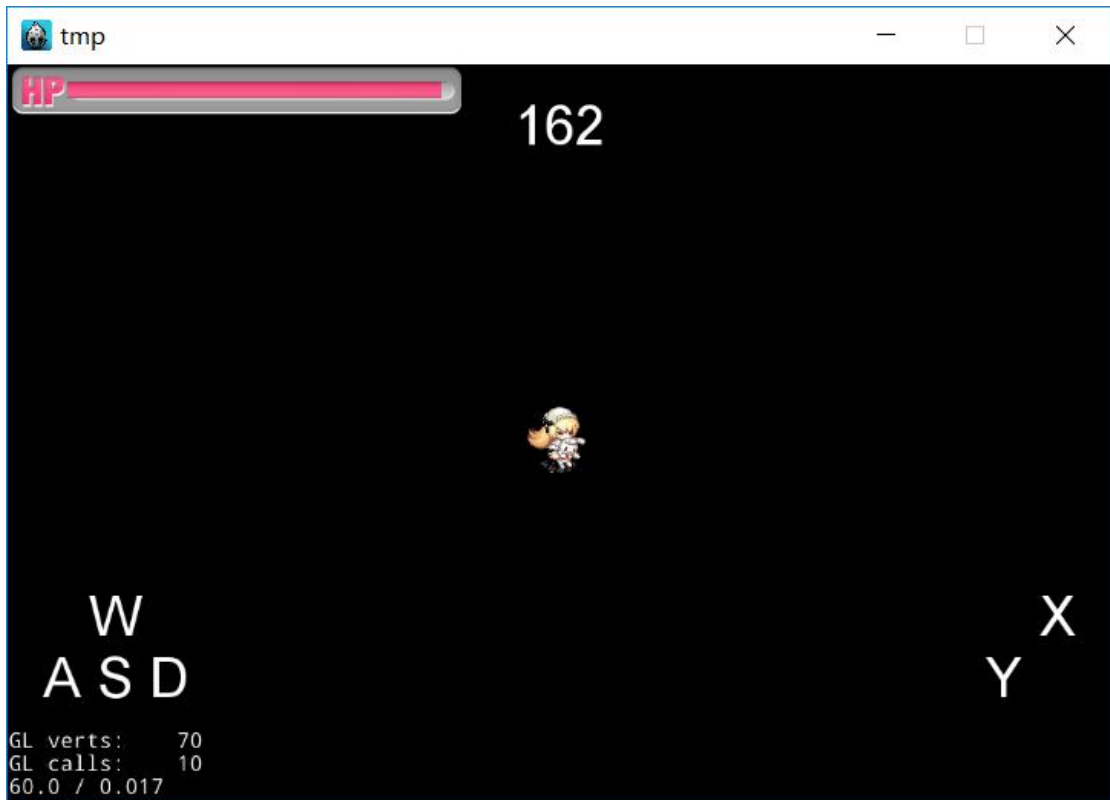




第十周:



第十一周



四、亮点与改进（可选）

实现了每一周的 bonus。

第九周：

添加一个点击后随机移动的 Eventlistener

第十周：

添加了关于钻石的动画。

第十一周：

添加了血条增减动画。

五、遇到的问题

1. 关于中文的 xml 读取。

在代码没有出问题时依然无法加载出中文。

后无意中发现文本的保存方式是 ANSI, 保存为 UTF-8 后成功加载。

2. 关于世界坐标与本地坐标的转换

由于对函数的作用理解错误，一直卡在发射石头。问过同学后才意识到原来是自己的理解出错。

3. 对 cocos2dx 的操作和设计思路不太熟悉，复杂一点的场景就有点不知所措。

六、思考与总结

1. 整个作业下来并没有多大的难度。但是由于不熟悉，本次作业还是被卡得很厉害。
2. 又温习了一下 c++ 的语法
3. 刚接触 cocos, 很多细节还不了解, 不熟练, 因此如果以后要使用到的话应该课后多加了解这方面的知识。
4. 虽然我对于 cocos 有了初步的认识。但是觉得依然有些不适应 cocos 的设计思路。感觉以后不想再使用它了……