

现代操作系统应用开发实验报告

学号： 15331046

班级： 软工一班

姓名： 陈志扬

实验名称： HW10

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。

资料：老师提供的教程 cocos2d-x 简单游戏元素、TA 师兄提供的作业要求及指导

网站：官方 API 文档 <http://api.cocos.com/> 百度 <https://www.baidu.com/>

cocos2d-x 坐标研究

http://blog.163.com/zjf_to/blog/static/201429061201292193855498/

深入理解 cocos2d-x 坐标系

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

二. 实验步骤

请在这里简要写下你的实验过程。

- ① 首先，学习课程所教内容 cocos2d-x 简单游戏元素，理解 Scene 类、Sprite 类、Action 类，学习使用导演场景、运行场景、创建精灵的三种方式、动作类的多种基本动作，以及序列动作、同步动作。深刻学习理解 cocos2d-x 坐标系，主要参考老师教程和 TA 师兄提供的网上两篇博客，区分屏幕坐标系与 cocos2d-x 坐标系、世界坐标系与本地坐标系。

简单学习 cocos2d-x 的锚点 Anchor。

- ② 接着，根据 TA 师兄的作业要求和指导逐步完善游戏功能。先从移植 Classes 和 Resources 文件夹开始，在 vs 中添加现有项解决编译错误的问题；了解游戏相对应的逻辑，逐步完成代码的编写。
- ③ 代码编写先从 MenuScene 开始，由于 TA 师兄已实现矿工 miner，且有相应的代码可参考，所以这部分比较简单，像 `gold-miner-text`、`menu-start-gold` 只需要调整好位置即可，然后添加一个 start 菜单，编写相应的代码以及回掉函数 startMenuCallback 即可。具体代码在第四部分实验过程中贴出。
- ④ 接下来编写老鼠爬行的动画，由于 TA 师兄给了矿工 miner 的腿部动画代码，可以作为参考，所以这部分也很简单，只需要小心老鼠爬行的动画不是三帧，而是八帧动画。具体代码同样在第四部分给出。
- ⑤ 接下来是实现 GameScene 部分的代码编写。添加两个 Layer，一个 stoneLayer 和一个 mouseLayer，设置好锚点和位置，然后添加两个精灵石头 stone 和老鼠 mouse，其中老鼠爬行的动画参考 MenuScene 中矿工 miner 腿部动画的代码即可成功编写。然后设置一个 Label，内容显示为 Shoot。具体代码同样在第四部分给出。
- ⑥ 再者，编写游戏逻辑。点击屏幕任意位置，可放置一块奶酪，老鼠跑到奶酪的位置“吃掉”奶酪。这个逻辑可转化为添加一个事件监听器 `EventListenerTouchOneByOne`，然后实现回掉函数 onTouchBegan 代码，具体为获取点击屏幕的位置，将其转换为本地坐标，且在该位置上添加奶酪精灵（包括渐隐的持续时间），然后通过本地坐标将老鼠移动到该

位置上，奶酪逐渐消失“被吃掉”。具体代码同样在第四部分给出。

- ⑦ 编写游戏逻辑。点击 Shoot，石头 stone 发射一个石头到老鼠的当前位置，与此同时老鼠留下钻石后跑到其他位置。这个逻辑可转化为创建一个 `MenuItemFont Label` 为 Shoot 的按钮，实现回调函数 `shootMenuCallback` 代码，具体为获取随机位置，将老鼠精灵 `mouse` 移动该位置上，然后通过老鼠的原先位置转换为世界坐标将钻石精灵 `diamond` 放到该原先位置上，同时添加 stone 从 `stoneLayer` 的起始位置发射过来，且设置其 3 秒后消失（渐隐）。具体代码同样在第四部分给出。

- ⑧ 至此，游戏基本功能已实现，再添加两个动画作为加分项。我主要在老鼠移动添加了两个动画：一个是在吃奶酪的过程中我设置老鼠到达奶酪的位置后扩大一倍然后再缩放回去；另一个是在点击 Shoot 后老鼠以顺时针方向旋转 360 度移动到其他随机位置。具体代码同样在第四部分给出。

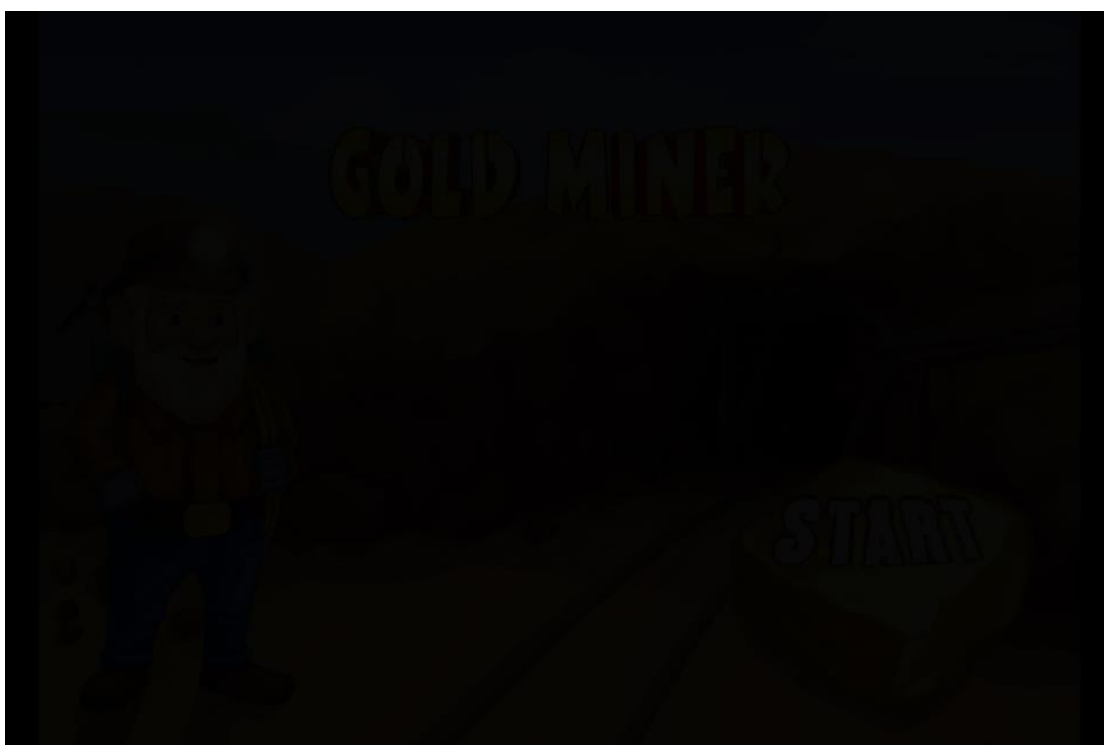
三．实验结果截图

请在这里把实验所得的运行结果截图。

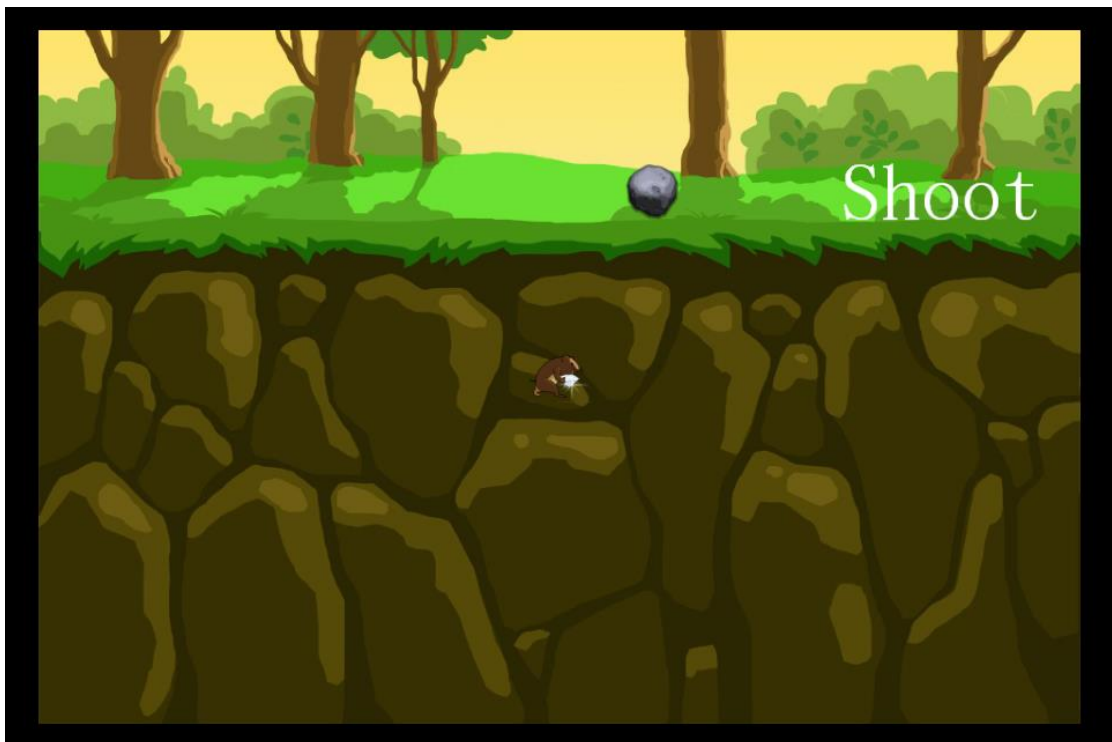
- ① 程序运行后 `MenuScene` 界面：可看到矿工在抖腿



② 点击 START 过渡动画（我设置了 0.8 秒渐隐）

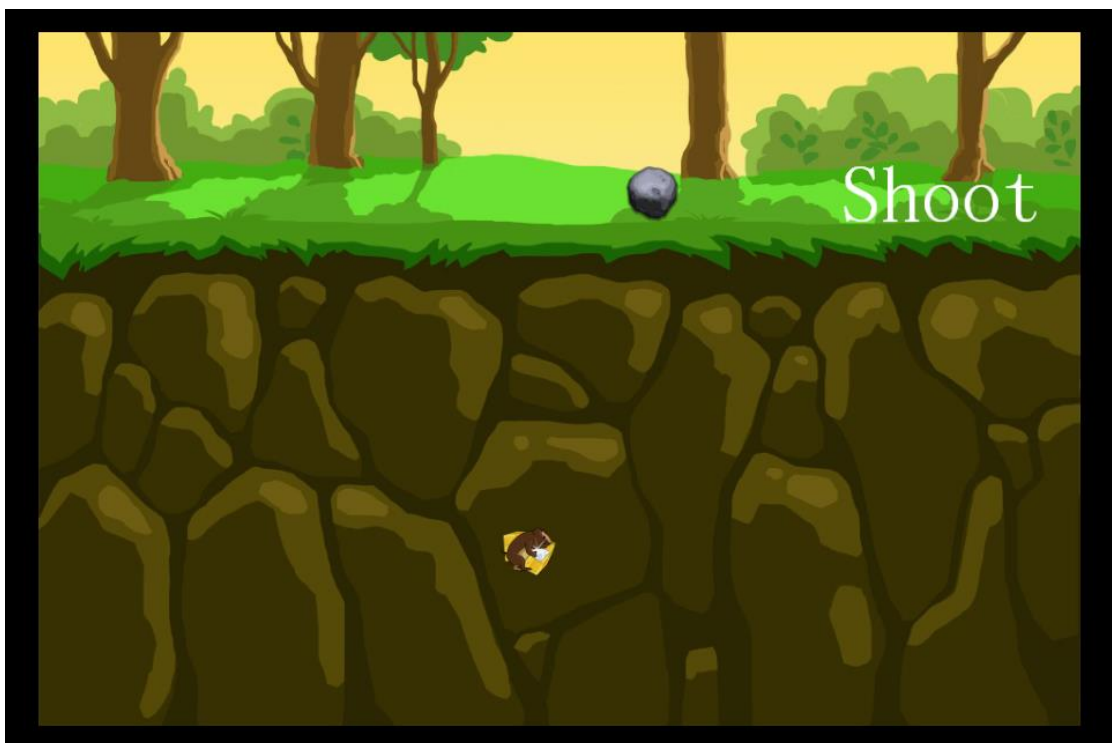


③ GameScene 界面：可看到老鼠在爬行

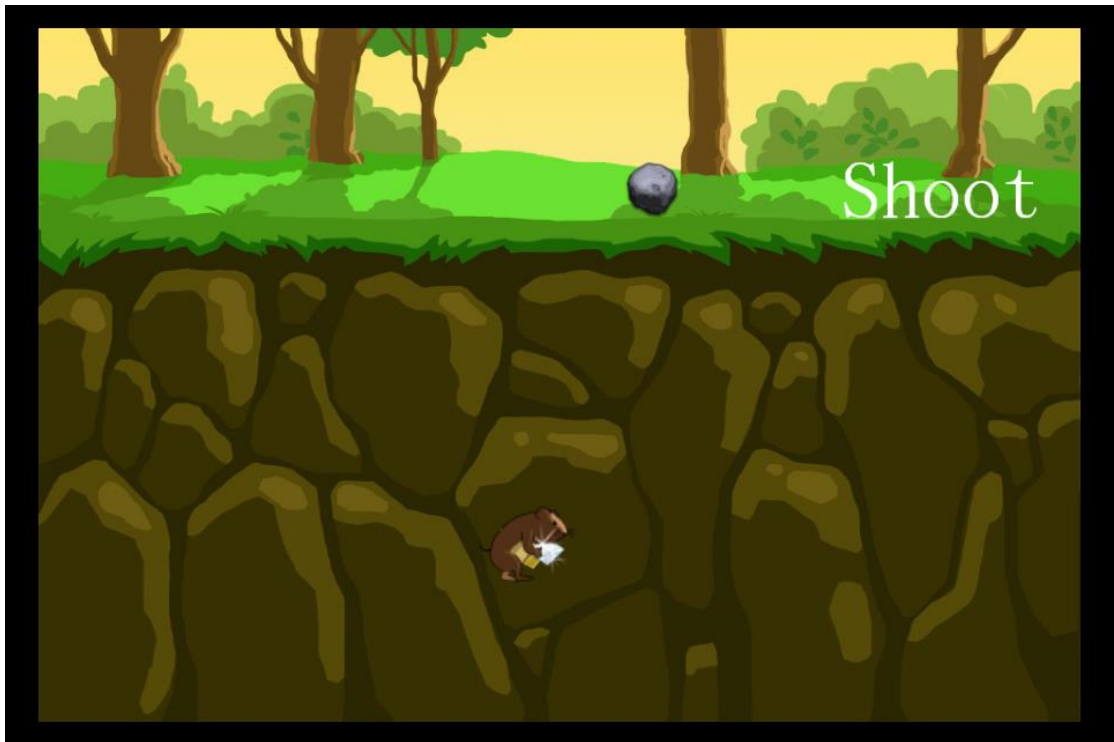


④ 点击屏幕任意位置：可看到奶酪出现，老鼠移动到奶酪位置后
先放大一倍，然后缩小到原先大小，奶酪消失不见。

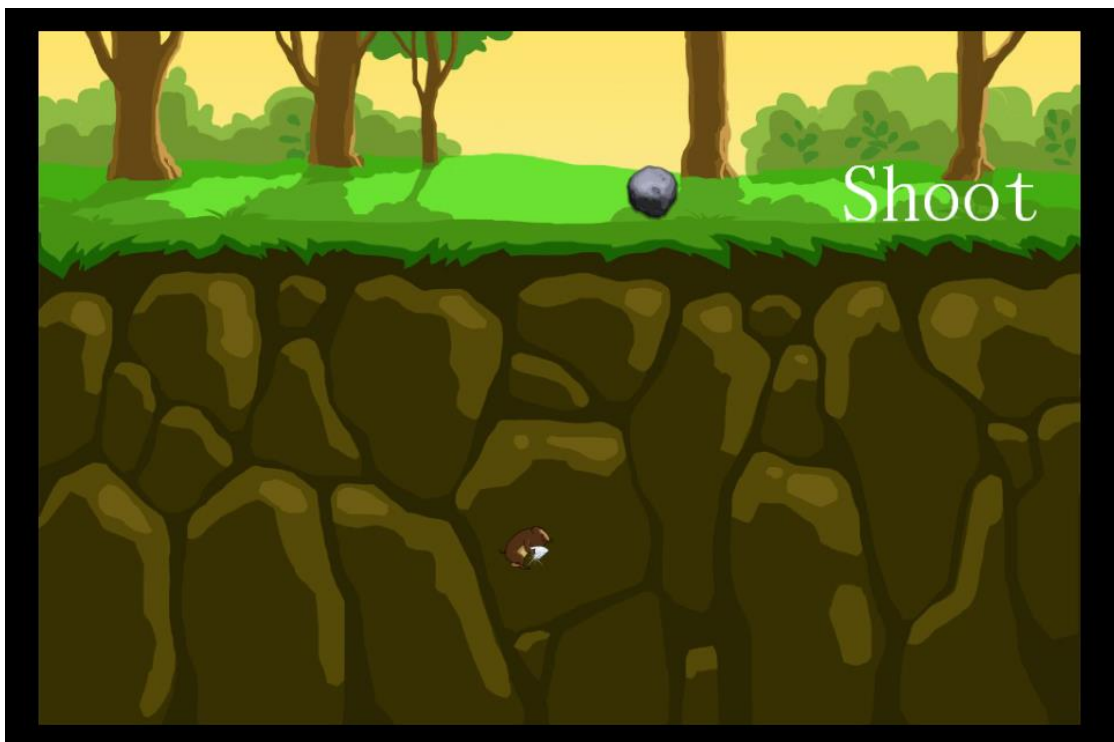
刚刚移动到奶酪位置：



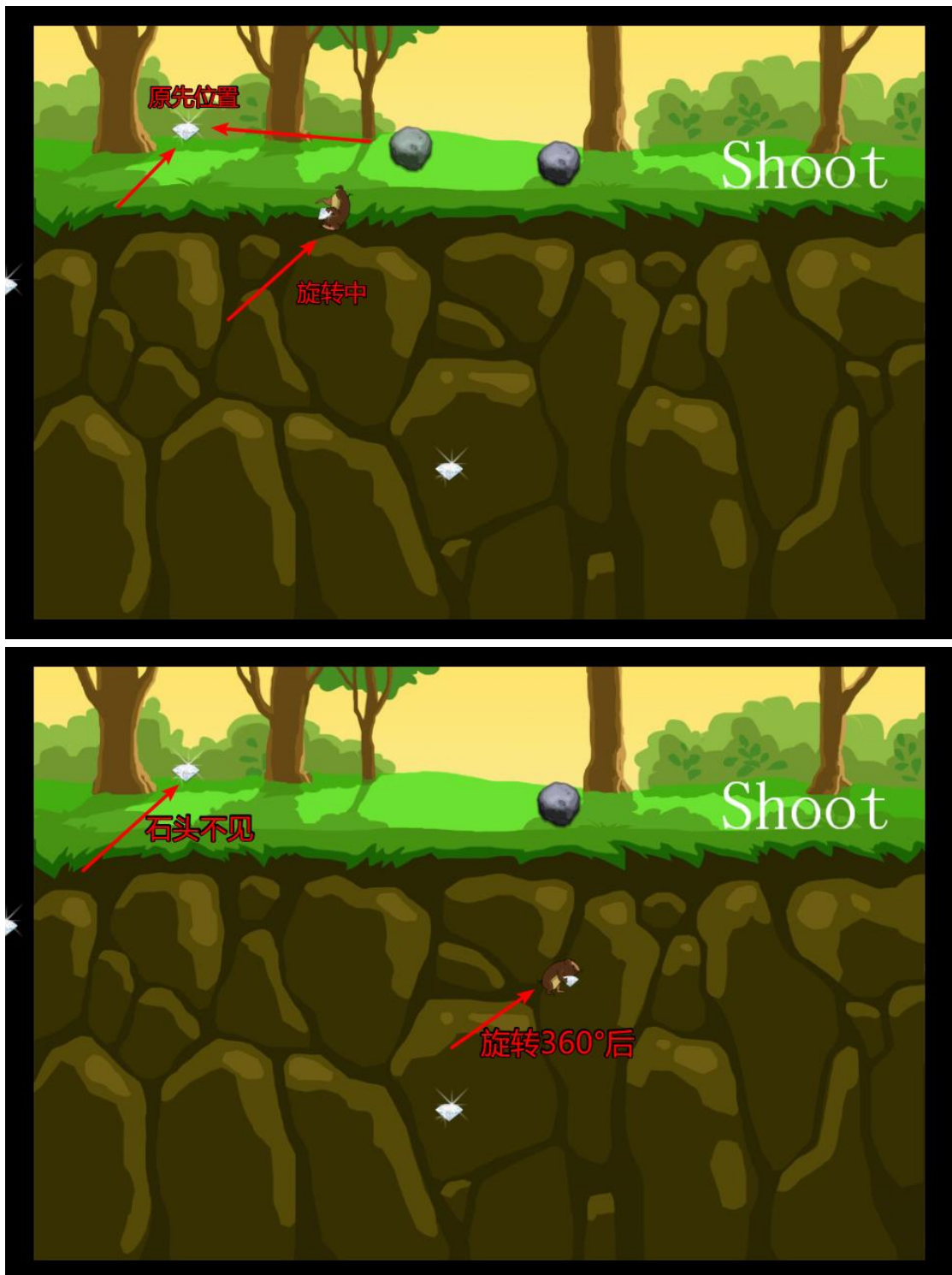
老鼠放大一倍：



奶酪被“吃掉”，老鼠缩小到原先大小：



⑤ 点击 Shoot：可看到老鼠以顺时针方向旋转 360 度移动到其
他任意位置，留下钻石在原先位置，而石头也发射到该位置上，
过了 3 秒后石头消失，钻石还在。



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。

这次实验其实比较简单 ,所以问题也比较少 ,在这里我主要贴一下我的关键代码。

① MenuScene 中回掉函数 startMenuCallback:

```

void MenuScene::startMenuCallback(cocos2d::Ref * pSender)
{
    // 渐隐效果
    auto scene = GameScene::createScene();
    Director::getInstance()->replaceScene(TransitionFade::create(0.8, scene, Color3B(0, 0, 0)));
}

```

② MenuScene 中各个精灵和 Start 菜单按钮的代码：

```

auto text = Sprite::create("gold-miner-text.png");
text->setPosition(Vec2(480 + origin.x, visibleSize.height / 2 + origin.y + 180));
this->addChild(text, 1);

auto gold = Sprite::create("menu-start-gold.png");
gold->setPosition(Vec2(750 + origin.x, visibleSize.height / 2 + origin.y - 180));
this->addChild(gold, 1);

auto start = MenuItemImage::create("start-0.png", "start-1.png", CC_CALLBACK_1(MenuScene::startMenuCallback, this));
start->setPosition(Vec2(755 + origin.x, visibleSize.height / 2 + origin.y - 130));

auto menu = Menu::create(start, NULL);
menu->setPosition(Vec2::ZERO);
this->addChild(menu, 1);

```

③ AppDelegate 中关于老鼠爬行的动画部分代码：

```

// load game resource
// the mouse's animation...
SpriteFrameCache::getInstance()->addSpriteFramesWithFile("level-sheet.plist");
int mouseFrames = 8;
char mouseName[20];
Animation* mouseAnimation = Animation::create();

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

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

```

④ GameScene 中各个精灵和 Shoot 这个 Label 的代码：

游戏背景：

```

// add the background of the game
auto level = Sprite::create("level-background-0.jpg");
level->setPosition(Vec2(visibleSize.width / 2 + origin.x, visibleSize.height / 2 + origin.y));
this->addChild(level, 0);

```

stoneLayer 和 stone 精灵：


```

// add the stone layer
stoneLayer = Layer::create();
stoneLayer->setAnchorPoint(Vec2(0, 0));
stoneLayer->setPosition(0, 0);
// add the stone sprite
stone = Sprite::create("stone.png");
stone->setPosition(Vec2(560, 480));
stoneLayer->addChild(stone);
this->addChild(stoneLayer, 1);

```

mouseLayer 和 mouse 精灵 (包括爬行动画):

```

// add the mouse layer
mouseLayer = Layer::create();
mouseLayer->setAnchorPoint(Vec2(0, 0));
mouseLayer->setPosition(Vec2(0, visibleSize.height / 2));
// add the mouse sprite
// includes its animation
mouse = Sprite::createWithSpriteFrameName("gem-mouse-0.png");
Animate* mouseAnimate = Animate::create(AnimationCache::getInstance()->getAnimation("mouseAnimation"));
mouse->runAction(RepeatForever::create(mouseAnimate));
mouse->setPosition(Vec2(visibleSize.width / 2, 0));
mouseLayer->addChild(mouse);
this->addChild(mouseLayer, 1);

```

Shoot Label 的代码和为之创建一个 menu :

```

// add the "Shoot" label with MenuItemFont Class
auto shoot = MenuItemFont::create("Shoot", CC_CALLBACK_1(GameScene::shootMenuCallback, this));
shoot->setFontSizeObj(70);
shoot->setColor(Color3B(255, 255, 255));
shoot->setPosition(Vec2(visibleSize.width / 2 + origin.x + 330, visibleSize.height / 2 + origin.y + 160));
// add the shoot menu
auto menu = Menu::create(shoot, NULL);
menu->setPosition(Vec2::ZERO);
this->addChild(menu, 1);

```

⑤ 点击屏幕的任意位置的逻辑代码 (包括创建奶酪精灵 , 奶酪 8 秒后 fade

out , 老鼠移动 , 老鼠放大和缩小代码):

```

bool GameSence::onTouchBegan(Touch *touch, Event *unused_event) {

    auto location = touch->getLocation();
    // add the cheese sprite
    auto cheese = Sprite::create("cheese.png");
    cheese->setPosition(location);
    this->addChild(cheese, 0);

    // cheese fade out after 8 seconds
    auto cheeseFadeOut = FadeOut::create(8);
    cheese->runAction(cheeseFadeOut);

    // convert the touched location to the node space coordinates
    // move the mouse to the node location
    auto location_node = mouseLayer->convertToNodeSpace(location);
    auto mouseMoveTo = MoveTo::create(2, location_node);
    // 放大和缩小
    auto mouseScaleTo = ScaleTo::create(0.5, 2);
    auto mouseScaleBack = ScaleTo::create(0.5, 1);
    // 序列动作
    auto seq = Sequence::create(mouseMoveTo, mouseScaleTo, mouseScaleBack, nullptr);
    mouse->runAction(seq);

    return true;
}

```

- ⑥ 点击 Shoot 按钮的逻辑代码(包括老鼠以顺时针方向旋转 360 度移动到随机位置、放置钻石精灵 diamond 在原先位置、发射石头精灵 stone 到原先位置且石头 3 秒后 fade out):

```

void GameScene::shootMenuCallback(Ref * pSender)
{
    Size visibleSize = Director::getInstance()->getVisibleSize();
    // when shot, the mouse moves to another location randomly
    int random_width = CCRANDOM_0_1() * visibleSize.width;
    int random_height = CCRANDOM_0_1() * visibleSize.height;
    auto location = mouseLayer->convertToNodeSpace(Vec2(random_width, random_height));
    auto mouseMoveTo = MoveTo::create(1, location);
    // rotate itself
    auto mouseRotateTo = RotateBy::create(1, 360);
    // let the two actions run at the same time
    auto spa = Spawn::createWithTwoActions(mouseMoveTo, mouseRotateTo);
    mouse->runAction(spa);

    // convert the position of mouse to world space coordinates
    auto position = mouse->getPosition();
    auto position_world = mouseLayer->convertToWorldSpace(position);

    // add the diamond sprite to the world position of the previous mouse
    auto diamond = Sprite::create("diamond.png");
    diamond->setPosition(position_world);
    this->addChild(diamond, 1);

    // add the stone and move it to the world position of the previous mouse
    auto stone = Sprite::create("stone.png");
    stone->setPosition(Vec2(560, 480));
    this->addChild(stone, 1);
    auto stoneMoveTo = MoveTo::create(1, position_world);
    stone->runAction(stoneMoveTo);
    // stone fades out after 3 seconds
    auto stoneFadeOut = FadeOut::create(3);
    stone->runAction(stoneFadeOut);
}

```

五 . 思考与总结

请在这里写下你本次试验的心得体会以及所思所想。

总的来说，这次实验还是比较简单的，原因之一是 TA 师兄提供给我们的 demo 已经实现了很多代码，而且代码也非常具有参考性，所以完成起来相对比较轻松。主要花时间的地方在于各精灵的位置设置上，但调了这么多位置，终于有点懵懵懂懂了。

做游戏是非常好玩的，虽然这只是简单游戏，但依然充满了设计者的智慧，我由衷地向游戏设计者表示敬佩，希望通过努力学习我能够取得巨大的进步！

1. 实验报告提交格式为 pdf。
2. 实验内容不允许抄袭，我们要进行代码相似度对比。如发现抄袭，按 0 分处理。