现代操作系统应用开发实验报告

学号： 14970011　　**班级：** 2015 级教务 2 班

姓名： 宋思亭　　**实验名称：** homework10

一．参考资料

作业要求文档, 课件 PPT，

官网：http://www.cocos.com/

Github：https://github.com/cocos2d/cocos2d-x

用户手册：http://www.cocos2d-x.org/wiki/Cocos2d-x

商店：http://store.cocos.com/

API: http://api.cocos.com/

二．实验步骤

1. 阅读作业需求和课件 PPT，了解 cocos-2dx 中的游戏运行方式 Scene 类、Sprite 类和 Action 类，坐标系详解以及内存管理机制。运行 demo.exe，分析黄金矿工的游戏运行机制。
2. 导入Class和Resource文件，按作业PPT给出的方法，在AppDelegate.cpp中添加动画，由循环播放的多帧图片构成。动画分别是腿部动作，老鼠移动动作，石头发射动作和钻石闪烁动作

```
// load game resource
char frameName1[25];
SpriteFrameCache::getInstance()->addSpriteFramesWithFile("general-sheet.plist");
Animation* legAnimation = Animation::create();
for (int i = 0; i < 3; i++) {
    sprintf(frameName1, "miner-leg-%d.png", i);
    legAnimation->addSpriteFrame(SpriteFrameCache::getInstance()->getSpriteFrameByName(frameName1));
}
legAnimation->setDelayPerUnit(0.1);
AnimationCache::getInstance()->addAnimation(legAnimation, "legAnimation");

char frameName2[25];
SpriteFrameCache::getInstance()->addSpriteFramesWithFile("level-sheet.plist");
Animation* mouseAnimation = Animation::create();
for (int i = 0; i < 8; i++) {
    sprintf(frameName2, "gem-mouse-%d.png", i);
    mouseAnimation->addSpriteFrame(SpriteFrameCache::getInstance()->getSpriteFrameByName(frameName2));
}
mouseAnimation->setDelayPerUnit(0.1);
AnimationCache::getInstance()->addAnimation(mouseAnimation, "mouseAnimation");

char frameName3[25];
SpriteFrameCache::getInstance()->addSpriteFramesWithFile("level-sheet.plist");
Animation* stoneAnimation = Animation::create();
for (int i = 0; i < 2; i++) {
    sprintf(frameName3, "stone-%d.png", i);
    stoneAnimation->addSpriteFrame(SpriteFrameCache::getInstance()->getSpriteFrameByName(frameName3));
}
stoneAnimation->setDelayPerUnit(0.1);
AnimationCache::getInstance()->addAnimation(stoneAnimation, "stoneAnimation");

char frameName4[25];
SpriteFrameCache::getInstance()->addSpriteFramesWithFile("level-sheet.plist");
Animation* diamondAnimation = Animation::create();
for (int i = 0; i < 7; i++) {
    sprintf(frameName4, "diamond-%d.png", i);
    diamondAnimation->addSpriteFrame(SpriteFrameCache::getInstance()->getSpriteFrameByName(frameName4));
}
```

3. 写MenuSence.cpp文件，创建开始界面

在init中设置背景图片，分天空和陆地两部分，注意设置缩放比例以适配窗口不同大小：

```
//创建背景
Size visibleSize = Director::getInstance()->getVisibleSize();
Vec2 origin = Director::getInstance()->getVisibleOrigin();
float winw = visibleSize.width; //获取屏幕宽度
auto background_sky = Sprite::create("menu-background-sky.jpg");
background_sky->setPosition(Vec2(visibleSize.width / 2 + origin.x, visibleSize.height / 2 + origin.y + 150));
float maxWidth = background_sky->getTextureRect().getMaxX();
background_sky->setScaleX(winw / maxWidth); //设置缩放比例
this->addChild(background_sky, 0);

auto background = Sprite::create("menu-background.png");
background->setPosition(Vec2(visibleSize.width / 2 + origin.x, visibleSize.height / 2 + origin.y - 50));
maxWidth = background->getTextureRect().getMaxX();
background->setScaleX(winw / maxWidth); //设置缩放比例
this->addChild(background, 0);
```

创建 Logo，人物和腿部动画效果：

```
//创建人物
auto miner = Sprite::create("menu-miner.png");
miner->setPosition(Vec2(150 + origin.x, visibleSize.height / 2 + origin.y - 60));
this->addChild(miner, 1);

//创建腿部动画
auto leg = Sprite::createWithSpriteFrameName("miner-leg-0.png");
Animate* legAnimate = Animate::create(AnimationCache::getInstance()->getAnimation("legAnimation"));
leg->runAction(RepeatForever::create(legAnimate));
leg->setPosition(110 + origin.x, origin.y + 102);
this->addChild(leg, 1);

//创建Logo
auto logo = Sprite::create("gold-miner-text.png");
logo->setPosition(Vec2(origin.x + visibleSize.width / 2, origin.y + visibleSize.height - logo->getContentSize().height - 80));
this->addChild(logo, 1);
```

创建开始按钮，点击跳转到 GameSence：

```
//创建开始按钮底部
auto gold = Sprite::create("menu-start-gold.png");
gold->setPosition(Vec2(origin.x + visibleSize.width - 200, origin.y + gold->getContentSize().height / 2));
this->addChild(gold, 1);

//创建开始按钮，点击开始按钮跳转
auto start_button = MenuItemImage::create(
    "start-0.png",
    "start-1.png",
    CC_CALLBACK_1(MenuSence::startMenuCallback, this));
start_button->setPosition(Vec2(origin.x + visibleSize.width - 200, origin.y + gold->getContentSize().height / 2 + 50));
auto start = Menu::create(start_button, NULL);
start->setPosition(Vec2::ZERO);
this->addChild(start, 2);


void MenuSence::startMenuCallback(cocos2d::Ref * pSendera)
{
    Director::getInstance()->replaceScene(TransitionCrossFade::create(1, GameSence::createScene()));
}
```

4. 写GameSence.cpp文件，创建游戏界面

在init中设置背景，注意设置缩放比例以适配窗口不同大小：

```
// 设置背景
Size visibleSize = Director::getInstance()->getVisibleSize();
Vec2 origin = Director::getInstance()->getVisibleOrigin();
float winw = visibleSize.width;  // 获取屏幕宽度
float winh = visibleSize.height; // 获取屏幕高度
auto bg = Sprite::create("level-background-0.jpg");
bg->setPosition(Vec2(winw / 2 + origin.x, winh / 2 + origin.y));
float spx = bg->getTextureRect().getMaxX();
bg->setScaleX(winw / spx);//  背景缩放
bg->setScaleY(winw / spx);
this->addChild(bg, 0);
```

设置 StoneLayer，注意 Shoot 按钮是 label 类型，Resource 文件中有图片，但背景难以对齐，显示效果不好：

```
// 设置stoneLayer
stoneLayer = Layer::create();
stoneLayer->ignoreAnchorPointForPosition(false);
stoneLayer->setAnchorPoint(Vec2::ZERO);
stoneLayer->setPosition(Vec2::ZERO);
stone = Sprite::create("stone.png");
stone->setPosition(Vec2(560, 480));
stoneLayer->addChild(stone);

// 设置shoot，并且可点击
auto label = Label::createWithTTF("Shoot", "fonts/Marker Felt.ttf", 65);
auto shootBtn = MenuItemLabel::create(label, CC_CALLBACK_1(GameSence::shootMenuCallback, this));
Menu* shoot = Menu::create(shootBtn, NULL);
shoot->setPosition(Vec2(740, 480));
stoneLayer->addChild(shoot);
```

shootMenuCallback 事件包含发射石头的动画和钻石闪烁的动画，注意设置老鼠跑开的坐标：

```
void GameSence::shootMenuCallback(Ref * pSender) {
    Vec2 mousePos = mouse->getPosition();
    Size visibleSize = Director::getInstance()->getVisibleSize();
    float winw = visibleSize.width; // 获取屏幕宽度
    float winh = visibleSize.height; // 获取屏幕高度

    // 发射石头
    auto shootstone = Sprite::createWithSpriteFrameName("stone-0.png");
    Animate* stoneAnimate = Animate::create(AnimationCache::getInstance()->getAnimation("stoneAnimation"));
    shootstone->runAction(RepeatForever::create(stoneAnimate));
    shootstone->setPosition(stone->getPosition());
    this->addChild(shootstone, 1);
    auto seq = Sequence::create(MoveTo::create(1.5, Vec2(mousePos.x, mousePos.y + winh / 2)), FadeOut::create(0.5), nullptr);
    shootstone->runAction(seq);

    // 老鼠跑开留下钻石
    auto diamond = Sprite::createWithSpriteFrameName("diamond-0.png");
    Animate* diamondAnimate = Animate::create(AnimationCache::getInstance()->getAnimation("diamondAnimation"));
    diamond->runAction(RepeatForever::create(diamondAnimate));
    diamond->setPosition(mousePos);
    mouseLayer->addChild(diamond, 1);

    toPos.x = (int)(CCRANDOM_0_1() * 960);
    toPos.y = (int)(CCRANDOM_0_1() * (420 - winh / 2));
    auto mousemoveto = MoveTo::create(1, Point(toPos.x, toPos.y));
    mouse->runAction(mousemoveto);
}
```

设置MouseLayer：

```
// 设置mouseLayer
mouseLayer = Layer::create();
mouseLayer->ignoreAnchorPointForPosition(false);
mouseLayer->setAnchorPoint(Vec2(0, 0));
mouseLayer->setPosition(Vec2(0, winh / 2));

// 设置老鼠的动作
mouse = Sprite::createWithSpriteFrameName("gem-mouse-0.png");
Animate* mouseAnimate = Animate::create(AnimationCache::getInstance()->getAnimation("mouseAnimation"));
mouse->runAction(RepeatForever::create(mouseAnimate));
mouse->setPosition(winw / 2, 0);
toPos = Vec2(winw / 2, 0);
mouseLayer->addChild(mouse, 2);
```

设置点击监听事件和触屏响应onTouchBegan：

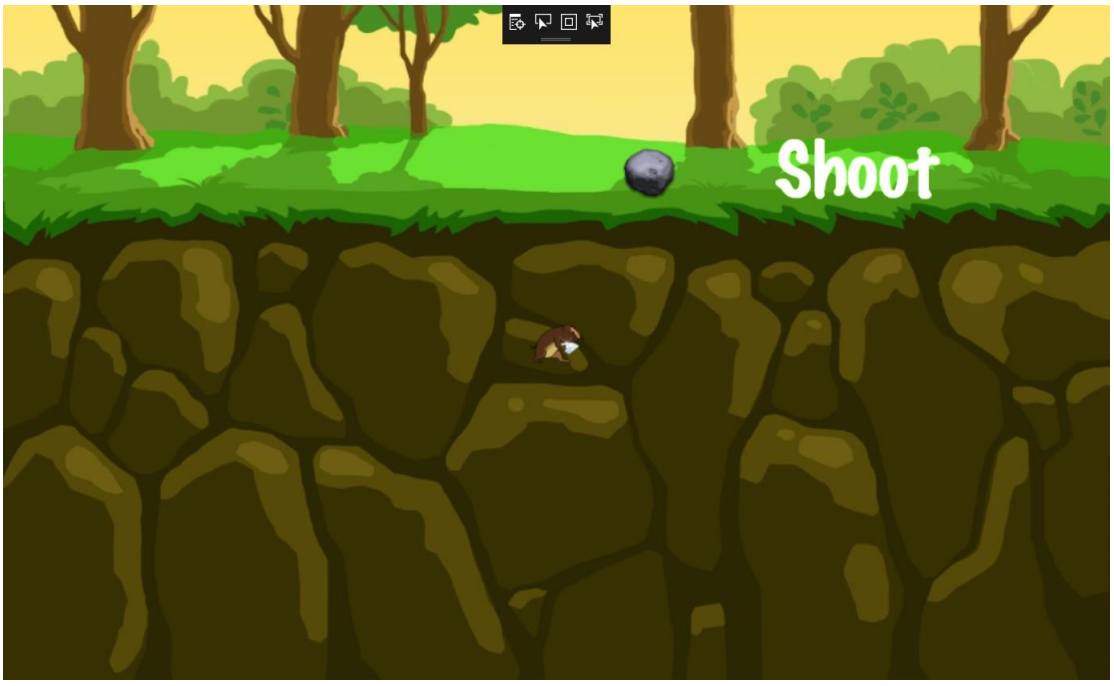
```
//设置点击监听事件
EventListenerTouchOneByOne* listener = EventListenerTouchOneByOne::create();
listener->setSwallowTouches(true);
listener->onTouchBegan = CC_CALLBACK_2(GameSence::onTouchBegan, this);
Director::getInstance()->getEventDispatcher()->addEventListenerWithSceneGraphPriority(listener, this);

bool GameSence::onTouchBegan(Touch *touch, Event *unused_event) {
    auto clickPos = touch->getLocation();  // 获得点击的坐标
    auto mousePos = mouse->getPosition();
    if (clickPos.y < 420) { // 点击石头下方才能放置奶酪
        Size visibleSize = Director::getInstance()->getVisibleSize();
        float winh = visibleSize.height;
        toPos = Vec2((int)clickPos.x, (int)(clickPos.y - winh / 2));
        // 设置奶酪
        cheese = Sprite::create("cheese.png");
        cheese->setPosition(toPos);
        mouseLayer->addChild(cheese);

        // 老鼠跑到奶酪位置并且吃掉奶酪
        auto moveto = MoveTo::create(2.0, toPos);
        mouse->runAction(moveto);
        cheese->runAction(Sequence::create(ScaleTo::create(2.0, 1.0), FadeOut::create(1.0), nullptr));
    }
    return true;
}
```

5. 调试项目

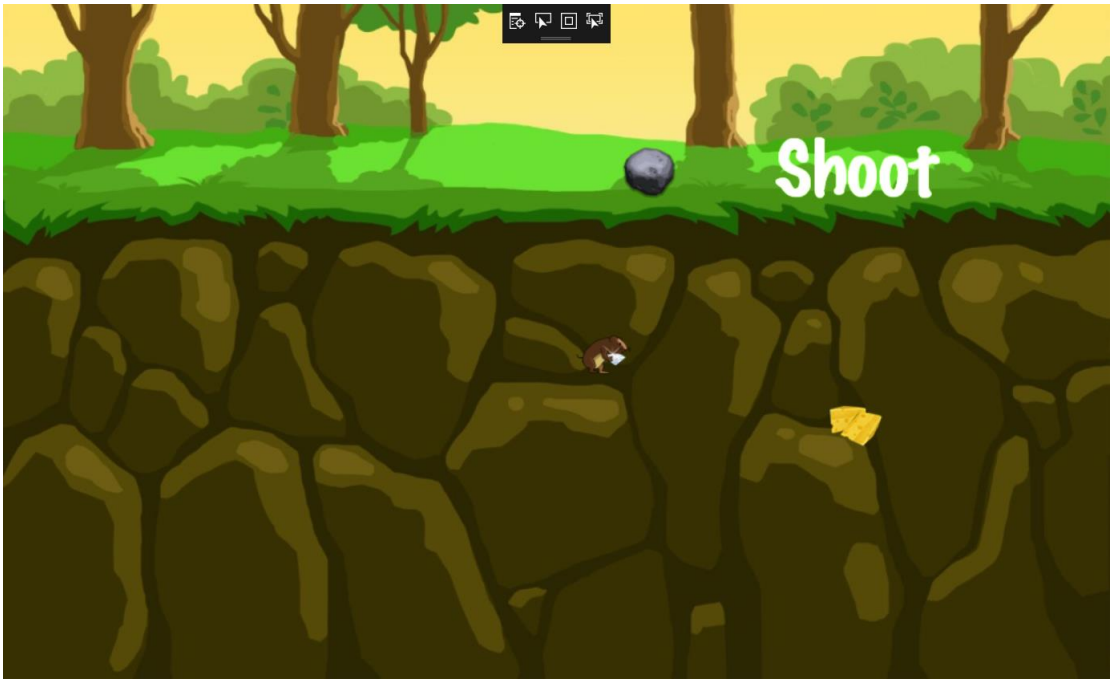
三．实验结果截图

1. 打开项目

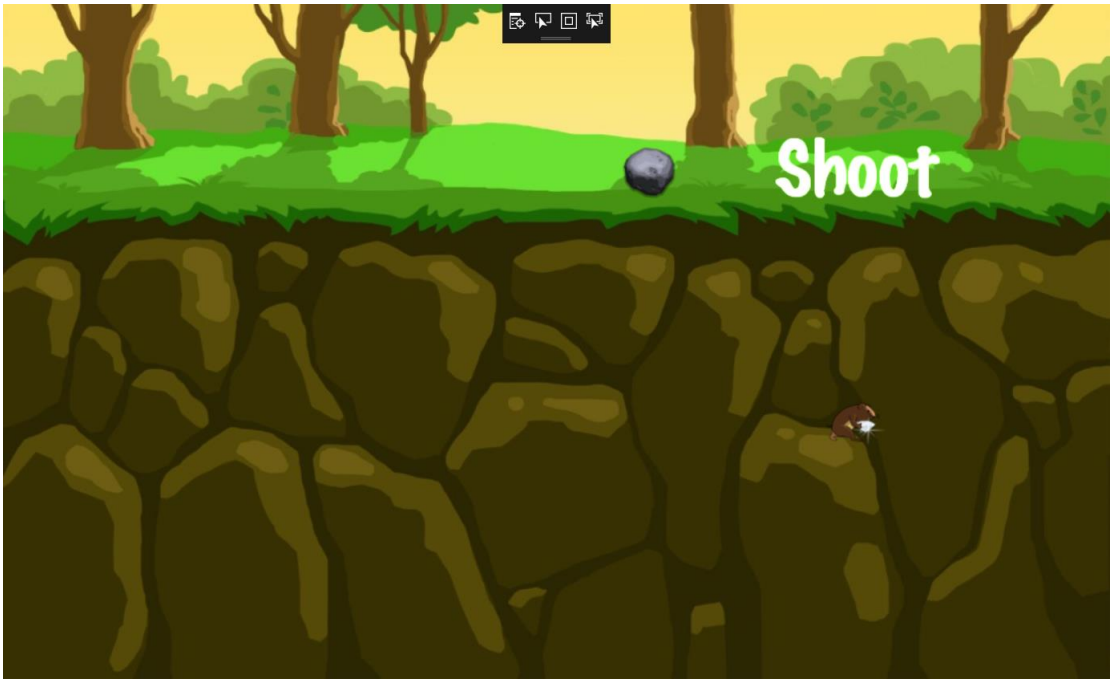
2. 点击 Start 开始游戏



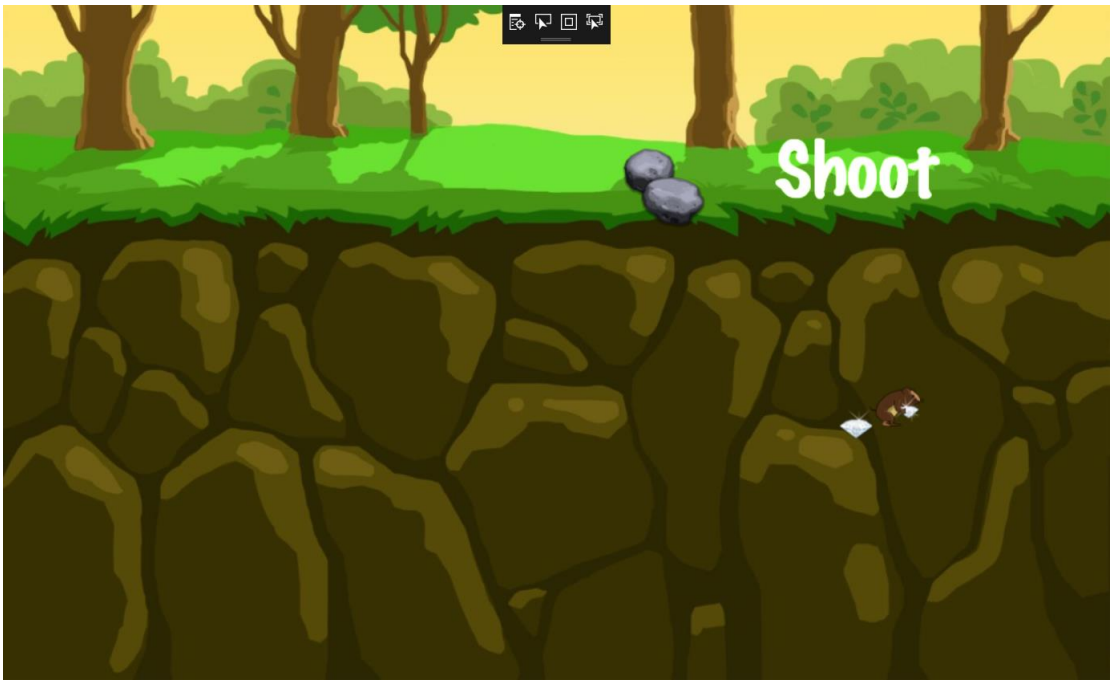
3. 点击屏幕任意位置添加奶酪

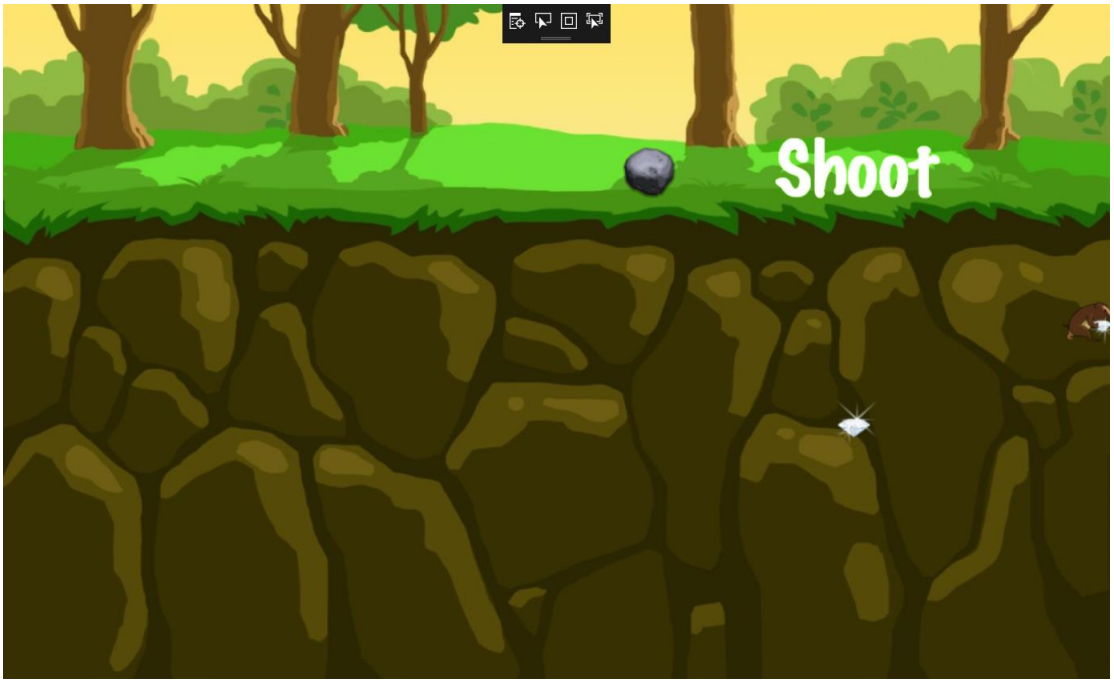


4. 老鼠跑到该位置吃掉奶酪

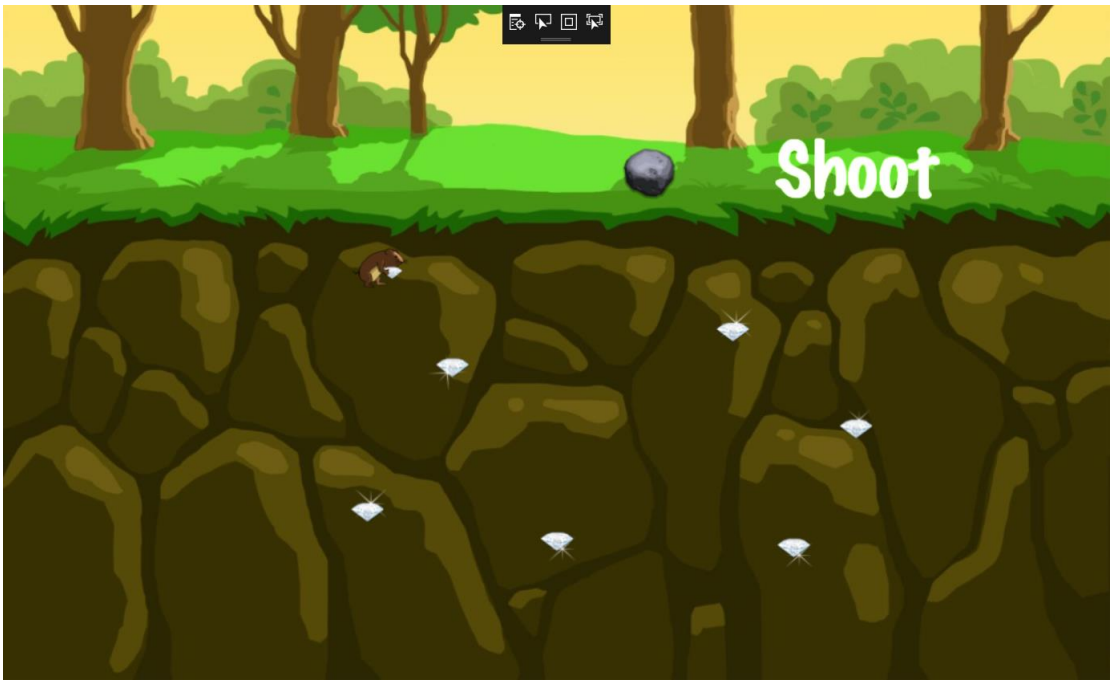


5. 点击 shoot 按钮，石头发射到老鼠所在的位置，老鼠跑开，留下钻石





6. 多次重复以上步骤



四．实验过程遇到的问题

1. 刚开始没有修改背景图片大小，界面存在黑边，将高度和宽度放大同样的比例后解决。

2. 调整各个图层位置比较麻烦， 以后编写相对复杂的项目时需要使用 cocos studio

等可以直观显示画面效果的工具。

3. 设置发射石头和钻石位置时出错，精灵的坐标是相对于父节点，将不同 layer 的坐标进行转换后解决。

五．思考与总结

1. 真正开始动手前，先分析游戏构成，理解游戏运行机制，把程序分解成一个个小的部分完成，更有效率，出错时也更容易 debug。
2. 理解开发工具的原理，比如帧动画运行原理，和 cocos2dx 坐标的表示，开发时才能更有效率地解决问题。