现代操作系统应用开发实验报告

学号： 14331388　　　　**班级　：** 教务三班

姓名： 郑泽佳　　　　**实验名称：** 实验九

一．参考资料

http://www.tairan.com/archives/5831/

http://blog.sina.com.cn/s/blog_6d193c030101h40e.html

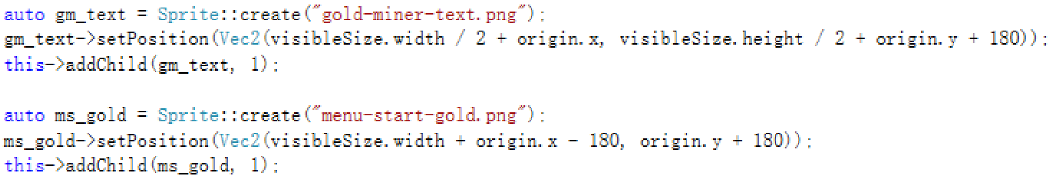
http://cocos2d-x.org/docs/api-ref/cplusplus/v3x/index.html

课件

二．实验步骤

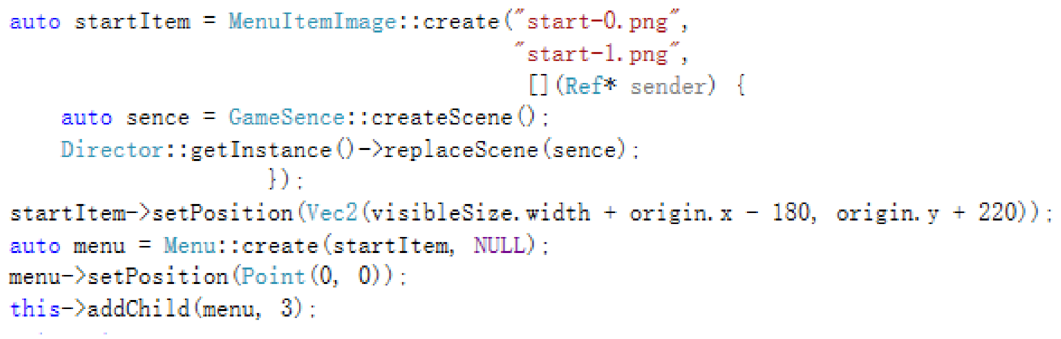
1、完善主界面，实现场景转移

（1）添加 gold-miner-text 和 menu-start-gold 精灵，调整位置，完善主界面，代码如下：



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

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

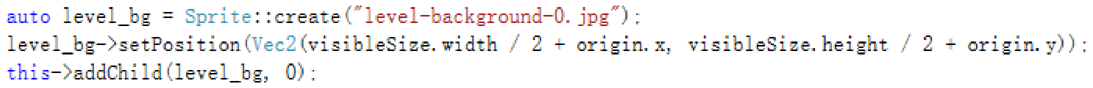
（2）添加开始按钮 MenuItemImage，调整位置，并实现其事件—场景转移，代码如下：



```
auto startItem = MenuItemImage::create("start-0.png",
                                       "start-1.png",
                                       [](Ref* sender) {
    auto sence = GameSence::createScene();
    Director::getInstance()->replaceScene(sence);
                         });
startItem->setPosition(Vec2(visibleSize.width + origin.x - 180, origin.y + 220));
auto menu = Menu::create(startItem, NULL);
menu->setPosition(Point(0, 0));
this->addChild(menu, 3);
```

2、实现游戏界面

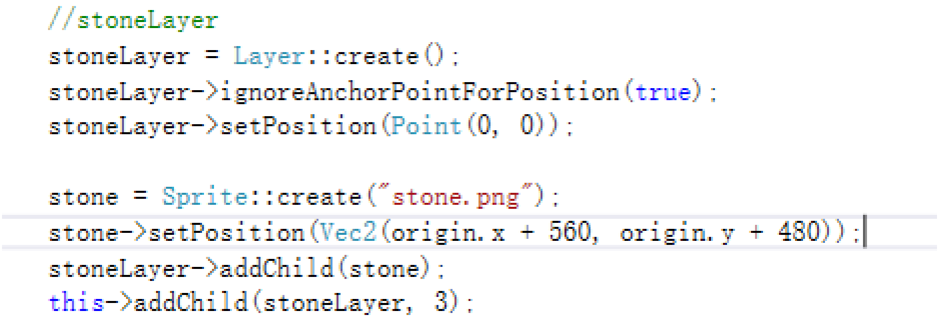
(1) 添加背景精灵，代码如下：



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

(2) 创建 Stone 精灵

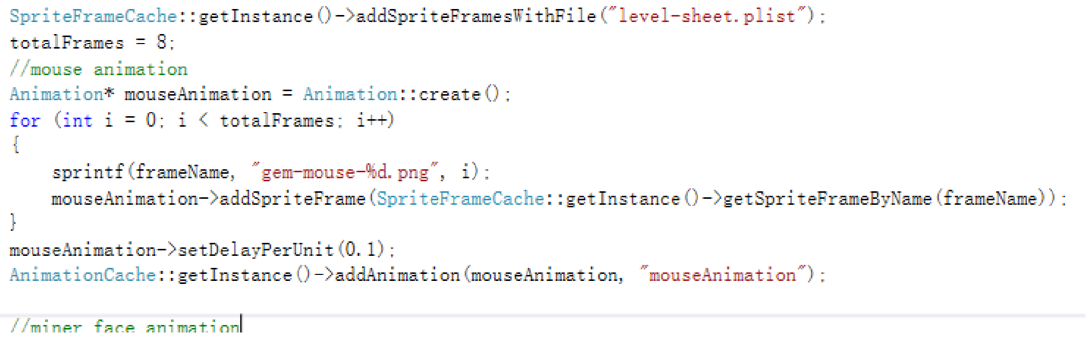
在 stoneLayer 中添加 stone 精灵，代码如下：



```
//stoneLayer
stoneLayer = Layer::create();
stoneLayer->ignoreAnchorPointForPosition(true);
stoneLayer->setPosition(Point(0, 0));

stone = Sprite::create("stone.png");
stone->setPosition(Vec2(origin.x + 560, origin.y + 480));
stoneLayer->addChild(stone);
this->addChild(stoneLayer, 3);
```

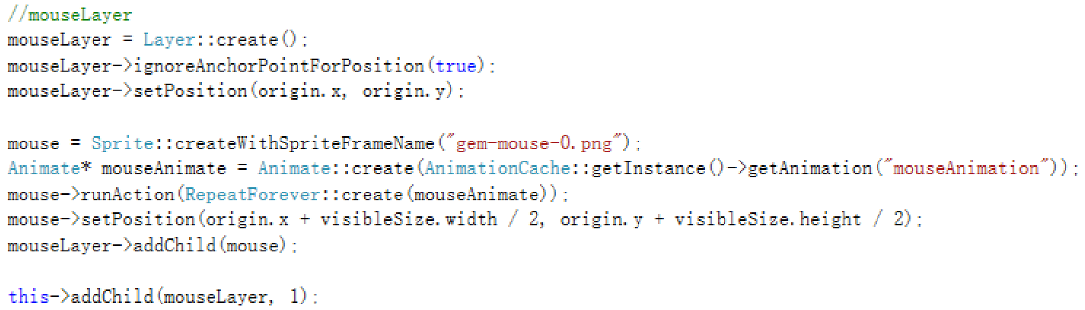
(3) 创建 mouse 精灵

在 mouseLayer 中添加 mouse 精灵，类似于 Leg 动画，在 AppDelegate.cpp 中预先加载 mouse 动画资源



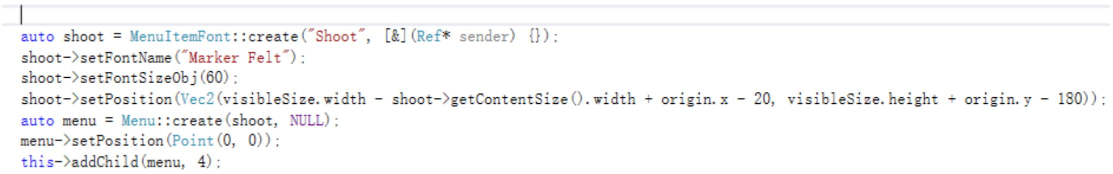
```
SpriteFrameCache::getInstance()->addSpriteFramesWithFile("level-sheet.plist");
totalFrames = 8;
//mouse animation
Animation* mouseAnimation = Animation::create();
for (int i = 0; i < totalFrames; i++)
{
    sprintf(frameName, "gem-mouse-%d.png", i);
    mouseAnimation->addSpriteFrame(SpriteFrameCache::getInstance()->getSpriteFrameByName(frameName));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

//miner face animation
```

在游戏界面中载入 mouse 动画资源



```
//mouseLayer
mouseLayer = Layer::create();
mouseLayer->ignoreAnchorPointForPosition(true);
mouseLayer->setPosition(origin.x, origin.y);

mouse = Sprite::createWithSpriteFrameName("gem-mouse-0.png");
Animate* mouseAnimate = Animate::create(AnimationCache::getInstance()->getAnimation("mouseAnimation"));
mouse->runAction(RepeatForever::create(mouseAnimate));
mouse->setPosition(origin.x + visibleSize.width / 2, origin.y + visibleSize.height / 2);
mouseLayer->addChild(mouse);

this->addChild(mouseLayer, 1);
```

(4) 创建 Shoot MenuItem

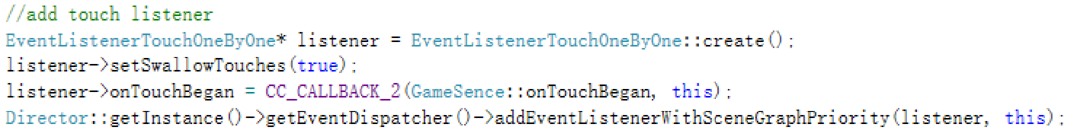


```
auto shoot = MenuItemFont::create("Shoot", [&](Ref* sender) {});
shoot->setFontName("Marker Felt");
shoot->setFontSizeObj(60);
shoot->setPosition(Vec2(visibleSize.width - shoot->getContentSize().width + origin.x - 20, visibleSize.height + origin.y - 180));
auto menu = Menu::create(shoot, NULL);
menu->setPosition(Point(0, 0));
this->addChild(menu, 4);
```

3、实现游戏效果

(1) 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

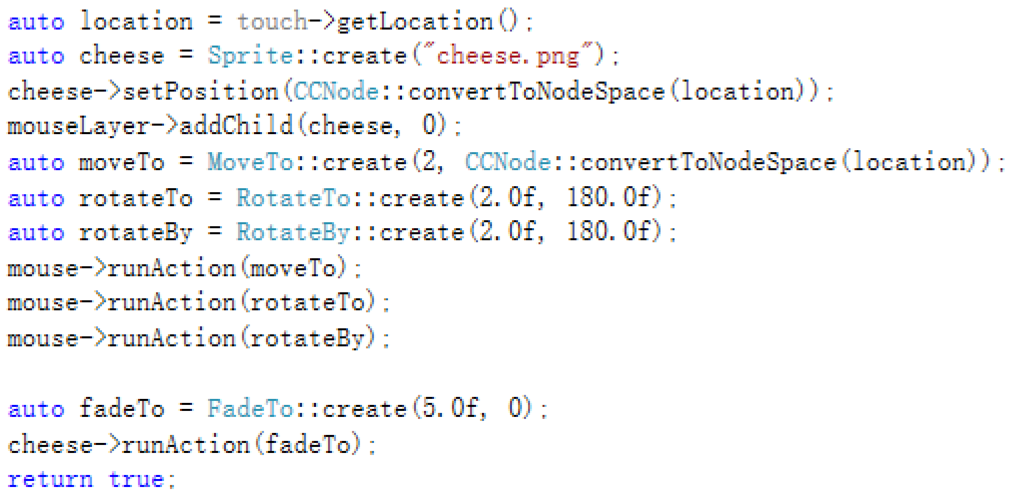
A、添加触摸事件监听器



```
//add touch listener
EventListenerTouchOneByOne* listener = EventListenerTouchOneByOne::create();
listener->setSwallowTouches(true);
listener->onTouchBegan = CC_CALLBACK_2(GameSence::onTouchBegan, this);
Director::getInstance()->getEventDispatcher()->addEventListenerWithSceneGraphPriority(listener, this);
```

B、实现 onTouchBegan 方法

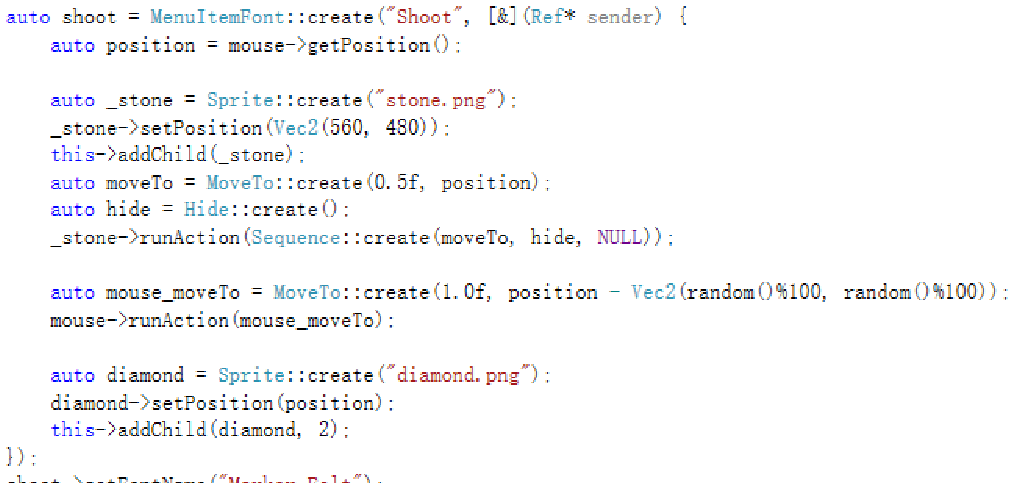
通过获取触摸监听器返回的触摸点位置，并将该触摸点的世界坐标转换为精灵所在层的本地坐标进行相应的动作操作，代码如下：



```
auto location = touch->getLocation();
auto cheese = Sprite::create("cheese.png");
cheese->setPosition(CCNode::convertToNodeSpace(location));
mouseLayer->addChild(cheese, 0);
auto moveTo = MoveTo::create(2, CCNode::convertToNodeSpace(location));
auto rotateTo = RotateTo::create(2.0f, 180.0f);
auto rotateBy = RotateBy::create(2.0f, 180.0f);
mouse->runAction(moveTo);
mouse->runAction(rotateTo);
mouse->runAction(rotateBy);

auto fadeTo = FadeTo::create(5.0f, 0);
cheese->runAction(fadeTo);
return true;
```

(2) 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。

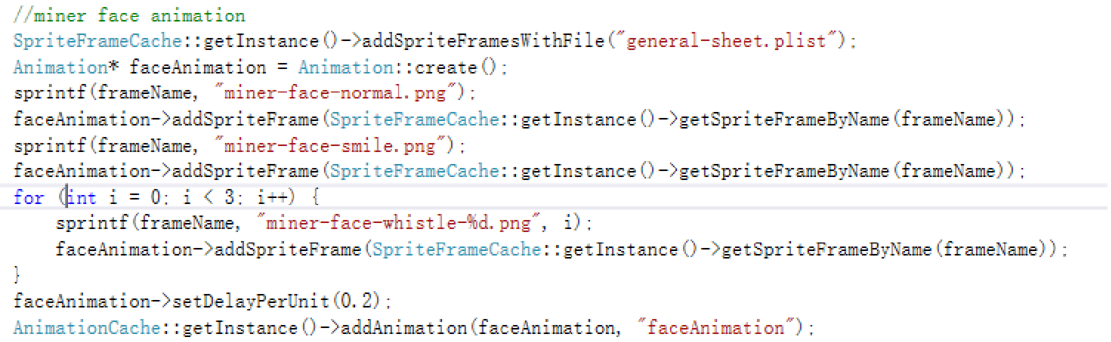
实现 shoot MenuItemImage 的触发事件



```
auto shoot = MenuItemFont::create("Shoot", [&](Ref* sender) {
    auto position = mouse->getPosition();

    auto _stone = Sprite::create("stone.png");
    _stone->setPosition(Vec2(560, 480));
    this->addChild(_stone);
    auto moveTo = MoveTo::create(0.5f, position);
    auto hide = Hide::create();
    _stone->runAction(Sequence::create(moveTo, hide, NULL));

    auto mouse_moveTo = MoveTo::create(1.0f, position - Vec2(random()%100, random()%100));
    mouse->runAction(mouse_moveTo);

    auto diamond = Sprite::create("diamond.png");
    diamond->setPosition(position);
    this->addChild(diamond, 2);
});
shoot->setFontName("Marker Felt");
```

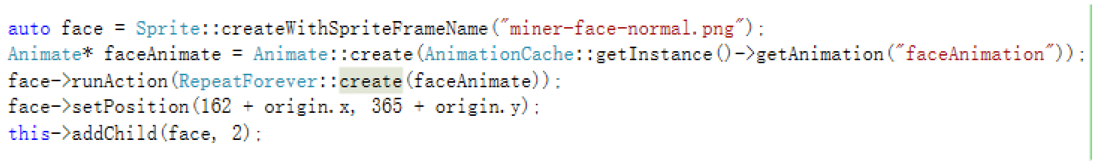
4、添加一个动画

同样类似于 leg 动画，在主界面中添加矿工脸部表情变化的动画。先在 AppDelegeate.cpp 中预先加载动画资源



```
//miner face animation
SpriteFrameCache::getInstance()->addSpriteFramesWithFile("general-sheet.plist");
Animation* faceAnimation = Animation::create();
sprintf(frameName, "miner-face-normal.png");
faceAnimation->addSpriteFrame(SpriteFrameCache::getInstance()->getSpriteFrameByName(frameName));
sprintf(frameName, "miner-face-smile.png");
faceAnimation->addSpriteFrame(SpriteFrameCache::getInstance()->getSpriteFrameByName(frameName));
for (int i = 0; i < 3; i++) {
    sprintf(frameName, "miner-face-whistle-%d.png", i);
    faceAnimation->addSpriteFrame(SpriteFrameCache::getInstance()->getSpriteFrameByName(frameName));
}
faceAnimation->setDelayPerUnit(0.2);
AnimationCache::getInstance()->addAnimation(faceAnimation, "faceAnimation");
```

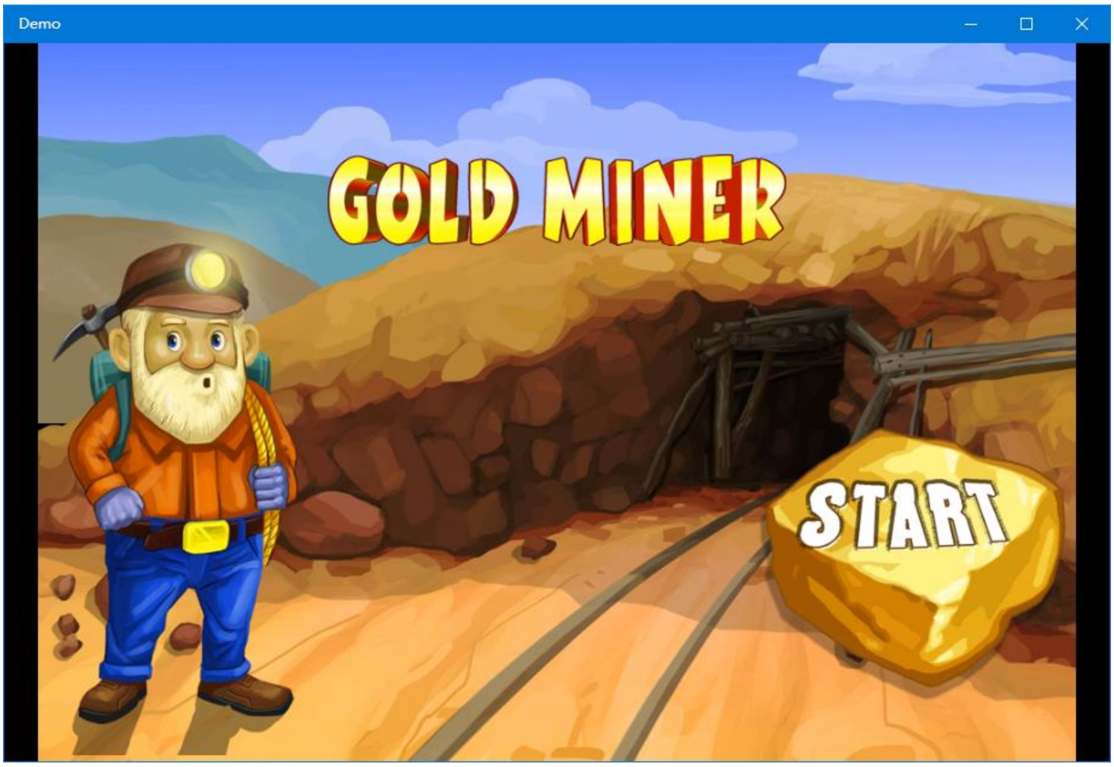
在主界面中载入动画资源，并调整位置使脸部表情与矿工对应。



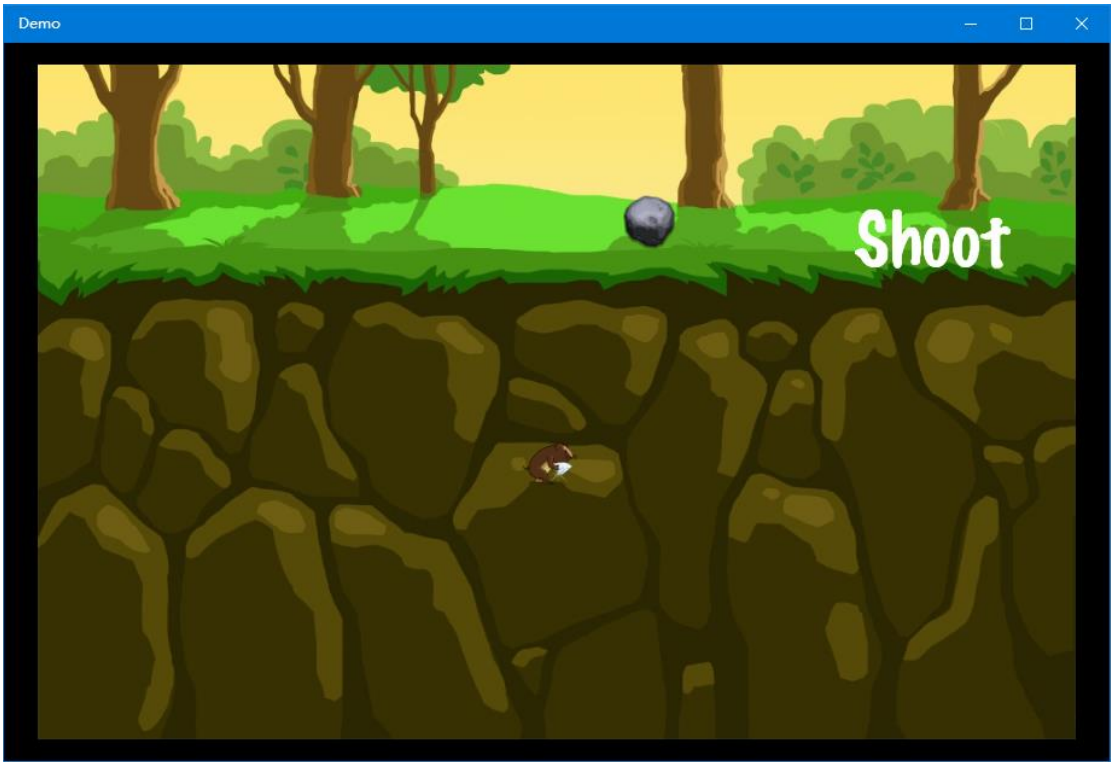
```
auto face = Sprite::createWithSpriteFrameName("miner-face-normal.png");
Animate* faceAnimate = Animate::create(AnimationCache::getInstance()->getAnimation("faceAnimation"));
face->runAction(RepeatForever::create(faceAnimate));
face->setPosition(162 + origin.x, 365 + origin.y);
this->addChild(face, 2);
```

三．实验结果截图

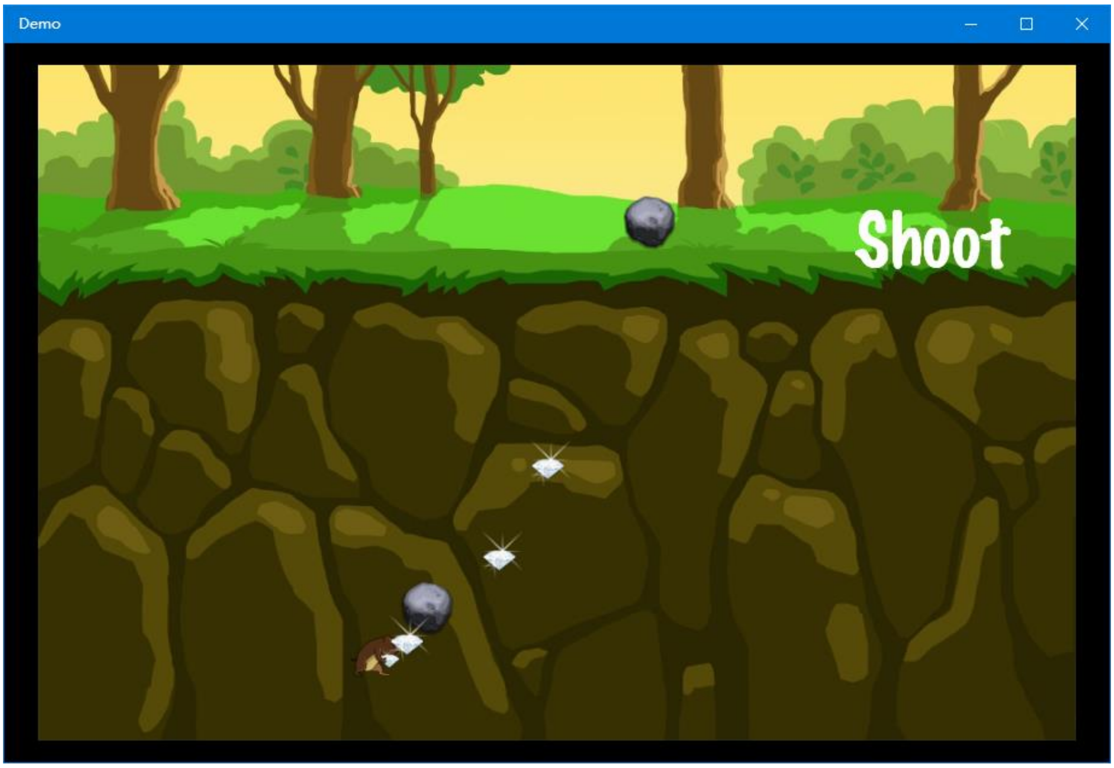
程序运行



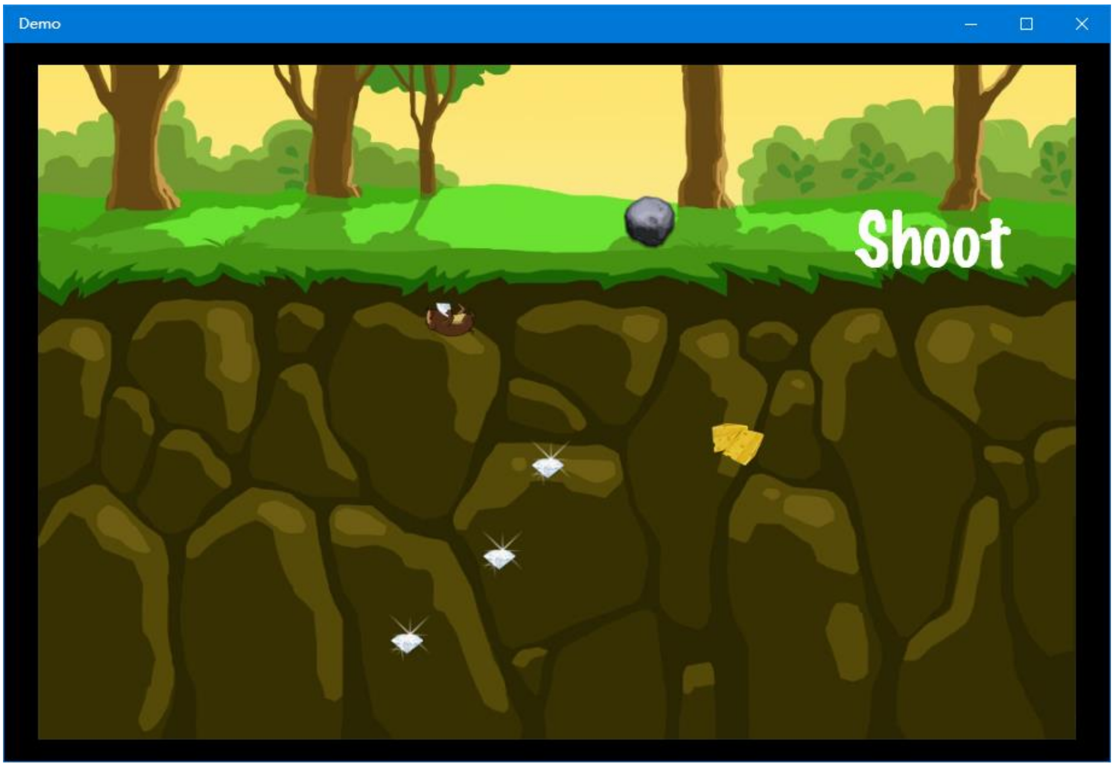
点击 Start 后进入游戏界面

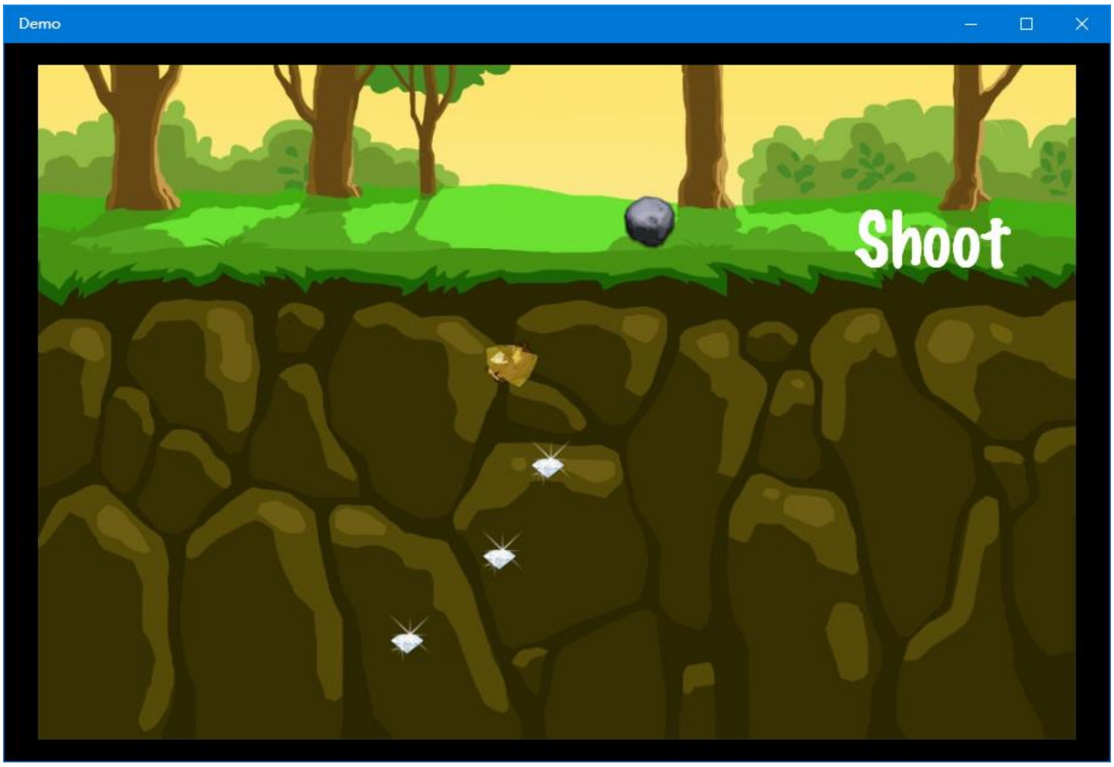


点击 Shoot



点击屏幕





四．实验过程遇到的问题

问题一：MenuItemFont 字体大小无法更改

解决过程：起初利用 setFontSize()对字体大小进行更改，可尝试多次字体依旧没改变，

通过查询得知，对于 C++要想改变字体的利用 setFontSizeObj()

问题二：点击 Shoot 时，射出的石头看不见

出错原因：设置石头位置利用了 origin.x 和 origin.y，误以为默认为 0，使得石头位置不在设想的地方。

五．思考与总结

本次实验内容运用到基本是课上讲的知识，所以相对比较轻松，不过对于坐标和锚点，还有就是界面的层次存放的顺序，个人还不是很清楚，由于没使用 cocos studio，所以在设计 UI 界面时有点麻烦，需要多次修改位置和层次重新运行程序来查看效果，加上程序运行占用较大资源，运行过慢，大大加长实验完成时间。总之，能够学以致用，收获很多。