

现代操作系统应用开发实验报告

学号： 15331242

班级： 周四早上班

姓名： 明友芬

实验名称： homework13

一、参考资料

<http://blog.csdn.net/sh15285118586/article/details/47148287>

<http://shahdza.blog.51cto.com/2410787/1553051>

<http://en.cppreference.com/w/cpp/utility/functional/placeholders>

老师上课用的 PPT 资料

二、实验步骤

- (1) 参考上次的实验内容，完成按键盘上的左右键，飞机可以左右移动的功能；
- (2) 补充完成 demo 的 fire() 函数，实现子弹的发射；
- (3) 添加背景音乐，和完成音乐的预加载函数；
- (4) 在子弹遇到陨石的时候实现陨石的爆炸的功能；
- (5) 实现附加功能，点击鼠标也可以拖动飞机飞走；
- (6) 实现添加陨石的功能，陨石不断的向下移动，并且会生成新的一行；
- (7) 正确显示子弹和陨石的数量。

三、实验结果截图

(1) 最开始的状态截图如下图 1 所示：

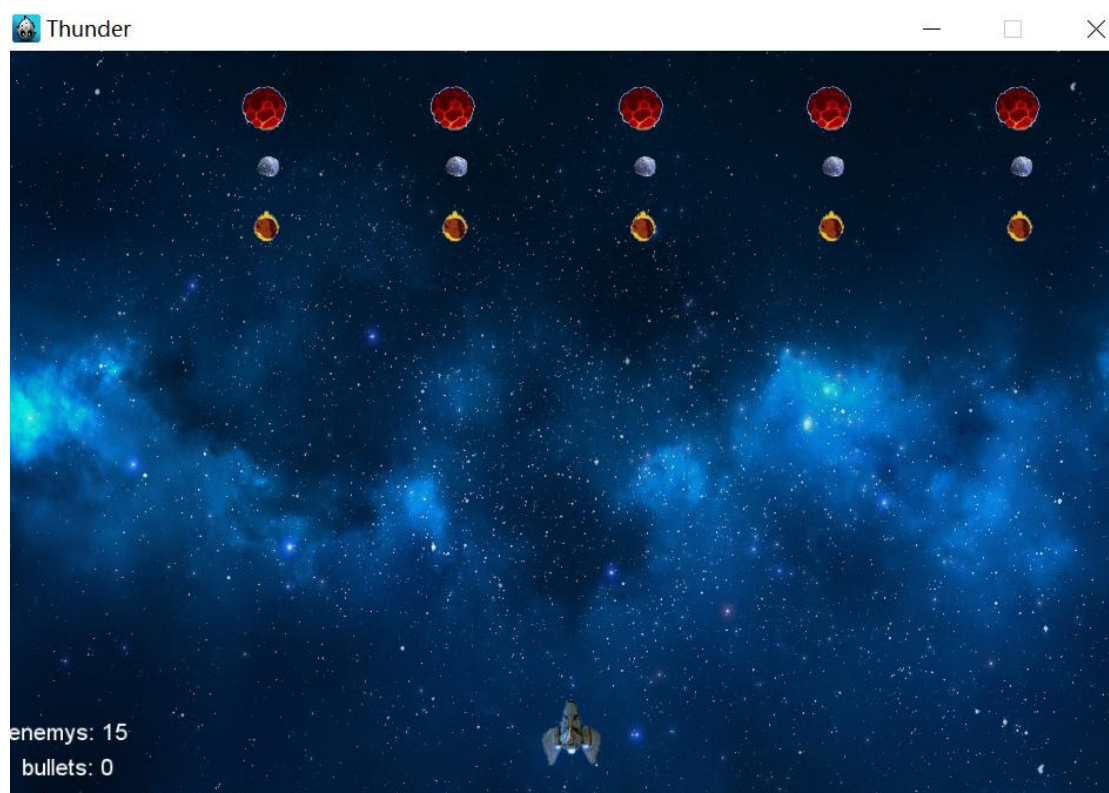


图 1

(2) 用鼠标点击飞机或者按空格键会发射子弹，实验结果如下图 2 所示：

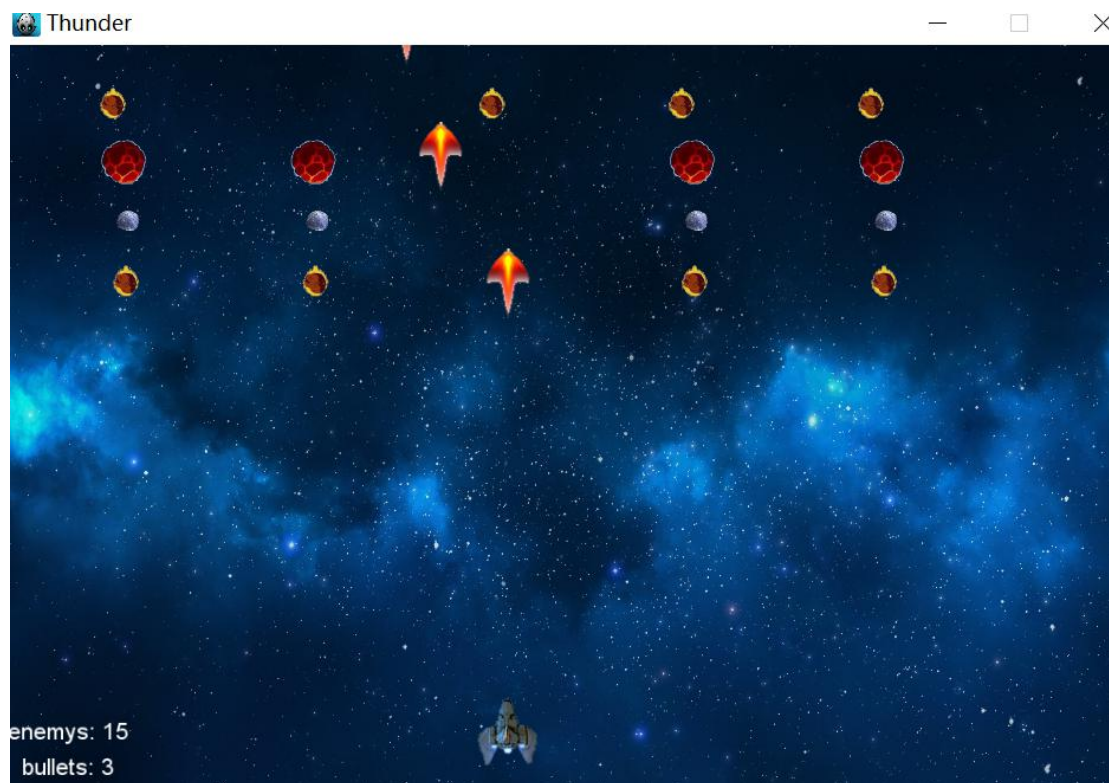


图 2

实现代码如下图 3 所示：

```
void Thunder::fire() {  
    //发射子弹  
    auto bullet = Sprite::create("bullet.png");  
    bullet->setAnchorPoint(Vec2(0.5, 0.5));  
    bullets.push_back(bullet);  
    bullet->setPosition(player->getPosition());  
    addChild(bullet, 1);  
    SimpleAudioEngine::getInstance()->playEffect("music/fire.wav", false);  
    // 移除飞出屏幕外的子弹  
    //这里将飞出屏幕的子弹移出, 并且在子弹队列里面也移除  
    bullet->runAction(Sequence::create(MoveBy::create(1.0f, Vec2(0, visibleSize.height)),  
        CallFunc::create([this, bullet]() {  
            bullet->removeFromParentAndCleanup(true);  
            this->bullets.remove(bullet);  
        })), nullptr));  
}
```

图 3

(3) 左右拖动飞机可以改变飞机的位置， 如图 3 所示：

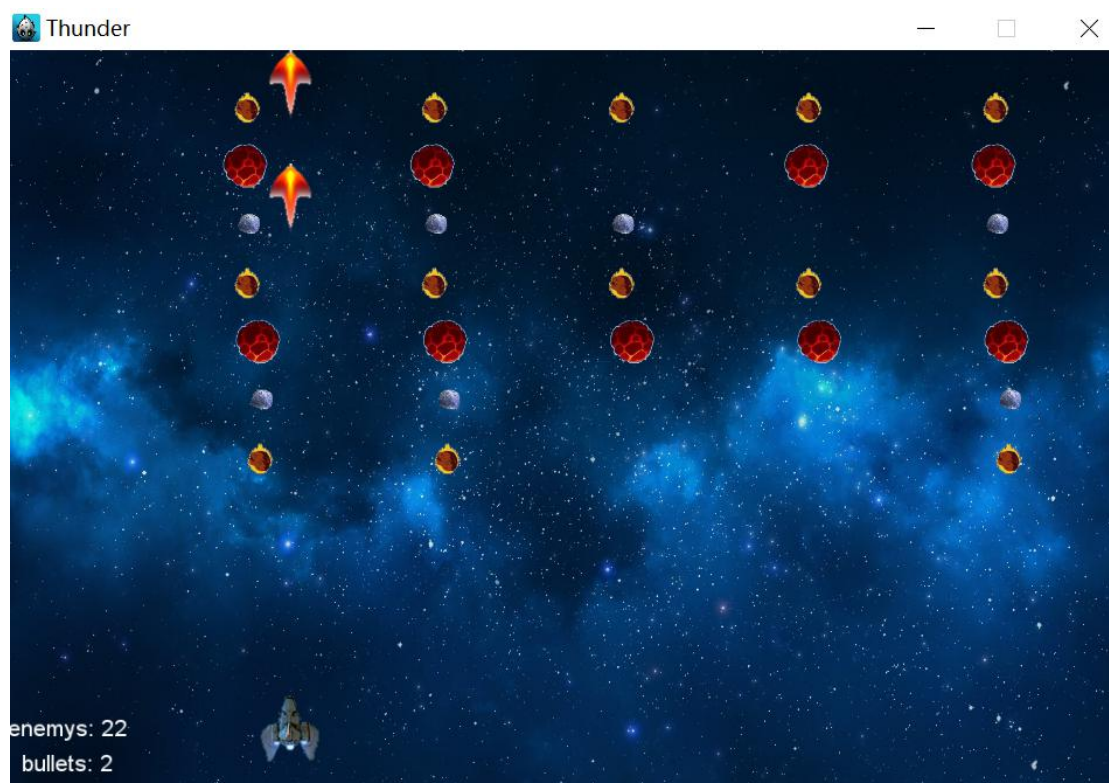


图 3

实现细节如此下图 4 所示：

```
void Thunder::addTouchListener() {
    this->setTouchEnabled(true);
    auto touchListener = EventListenerTouchOneByOne::create();
    touchListener->onTouchBegan = CC_CALLBACK_2(Thunder::onTouchBegan, this);
    touchListener->onTouchMoved = CC_CALLBACK_2(Thunder::onTouchMoved, this);
    touchListener->onTouchEnded = CC_CALLBACK_2(Thunder::onTouchEnded, this);
    this->getEventDispatcher()->addEventListenerWithSceneGraphPriority(touchListener, this);
}

// 鼠标点击发射炮弹
bool Thunder::onTouchBegan(Touch *touch, Event *event) {
    // 点击的地方只能限制在屏幕之内
    // 将精灵移动到点击的位置
    isClick = player->getBoundingBox().containsPoint(touch->getLocation());
    // 如果要在点击飞机时实现发射子弹只需要调用发射子弹的函数即可
    fire();
    return true;
}

void Thunder::onTouchEnded(Touch *touch, Event *event) {
    Vec2 pos = touch->getLocation();
    if (pos.x > origin.x + visibleSize.width)
        pos.x = origin.x + visibleSize.width;
    else if (pos.x < origin.x)
        pos.x = origin.x;
    player->setPosition(Vec2(pos.x, player->getPositionY()));
    isClick = false;
}

// 当鼠标按住飞船后可控制飞船移动（加分项）
void Thunder::onTouchMoved(Touch *touch, Event *event) {
    Vec2 pos = touch->getLocation();
    if (pos.x > origin.x + visibleSize.width)
        pos.x = origin.x + visibleSize.width;
    else if (pos.x < origin.x)
        pos.x = origin.x;
    player->setPosition(Vec2(pos.x, player->getPositionY()));
}
```

图 4

(4) 当陨石和子弹的距离小于 20 的时候会实现陨石的爆炸，实现细节如下

图 5 所示：

```
// 要确保这两个子弹是存在的
if (*begin1 && *begin2) {
    int distance = (*begin1)->getPosition().getDistance((*begin2)->getPosition());
    if (distance < 20) {
        temp = *begin1;
        // 将子弹移出屏幕
        (*begin1)->removeFromParentAndCleanup(true);
        // 实现陨石的爆炸
        (*begin1)->runAction(Sequence::create(Animate::create(Animation::createWithSpriteFrames(explore, 0.05f, 1)),
            CallFunc::create([this, temp] {
                temp->removeFromParentAndCleanup(true);
            }), nullptr));
    }
}
```

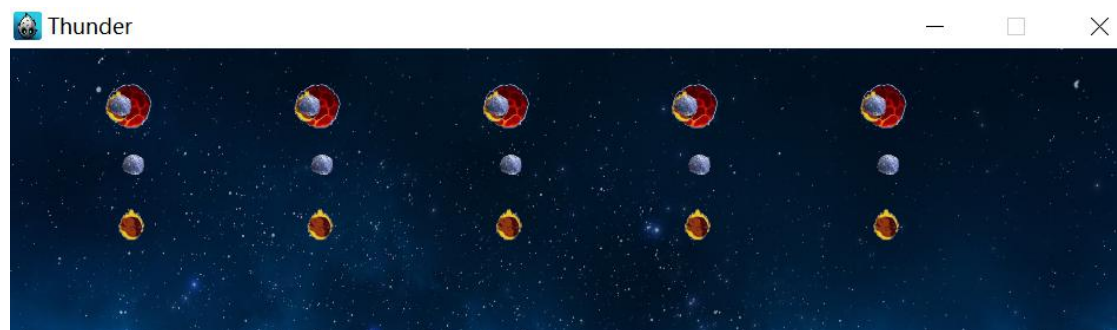
图 5

(5) 当陨石的位置低于飞机的位置的时候，游戏结束：



四、实验过程遇到的问题

在实现添加一串的陨石的时候，陨石能添加进去，但是不会实现自动的往下移动的功能：



实现的程序如下图所示：

```
void Thunder::newEnemy() {
    // 陨石向下移动并生成新的一行(加分项)
    static int stone = 1;
    char enemyPath[20];
    sprintf(enemyPath, "stone%d.png", stone);
    double width = visibleSize.width / 6.0;
    double height = visibleSize.height - 50;
    for (int j = 0; j < 5; ++j) {
        auto enemy = Sprite::create(enemyPath);
        enemy->setAnchorPoint(Vec2(0.5, 0.5));
        //设置缩放的比例, 否则的话那些石头会比较大
        enemy->setScale(0.5, 0.5);
        enemy->setPosition(width * j + 65, height);
        enemys.push_back(enemy);
        addChild(enemy, 1);
    }
    stone = stone % 3 + 1;
}
```

经过思考，才想起来我只添加了陨石，并没有实现将陨石往下移动的功能，所以将代码 更改一下之后就正确了，也就是每一次在将陨石添加上去之前将原有的陨石向下移动一 下：

```
void Thunder::newEnemy() {  
    // 陨石向下移动并生成新的一行  
    for (auto temp = enemys.begin(); temp != enemys.end(); temp++) {  
        if (*temp != NULL)  
            (*temp)->setPosition((*temp)->getPosition() + Vec2(0, -50));  
        //将原有的那些陨石不断往下移动位置实现动态的改变位置  
    }  
    static int stone = 1;  
    char enemyPath[20];  
    sprintf(enemyPath, "stone%d.png", stone);  
    double width = visibleSize.width / 6.0;  
    double height = visibleSize.height - 50;  
    for (int j = 0; j < 5; ++j) {  
        auto enemy = Sprite::create(enemyPath);  
        enemy->setAnchorPoint(Vec2(0.5, 0.5));  
    }  
}
```

五、思考与总结

在实验之前要仔细看实验要求，明确实验需要做什么，在实现功能的时候一个一个的实现，每实现一个就检测一下自己是否正确的实现了这个功能，这样也可以明确的知道自己是在哪里出错了，便于改正；

本次实验实现了一个简单的打陨石的游戏，在试实现的过程中使用了 lamda 表达式，总体而言本次实验收获挺大的。