

现代操作系统应用开发实验报告

学号： 14331098

班级： 周四班

姓名： 黄建武

实验名称： HW13

一 . 参考资料

游戏素材

<http://www.2gei.com/game2d/>

回调函数的使用

<http://blog.csdn.net/elloop/article/details/50438032>

碰撞检测

<http://www.2cto.com/kf/201408/326459.html>

二 . 实验步骤

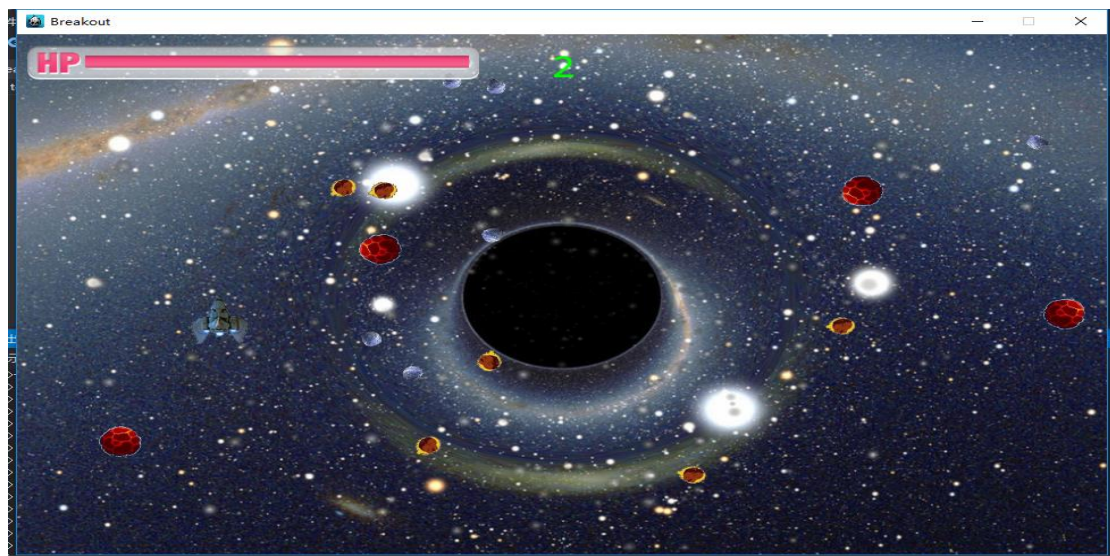
- ① 阅读 demo 的源代码，理解代码的思路。
- ② 设置合适掩码，实现碰撞检测；
- ③ 实现 onTouchBegin 函数，飞船碰到陨石播放音效和粒子特效；
- ④ 实现鼠标监听，点击飞船，按住鼠标控制飞船移动；
- ⑤ 增加 HP 条和时间，飞船碰到陨石掉血；
- ⑥ 增加计时，一定时间内血量不减为 0 则过关；
- ⑦ 实现过关页面和失败页面，过关后难度加大，每关游戏时间加 10 秒；
- ⑧ 添加额外奖励清屏炸弹；

⑨ 添加额外奖励血量恢复；

⑩ 进行最后的测试。

三．实验结果截图

开始界面



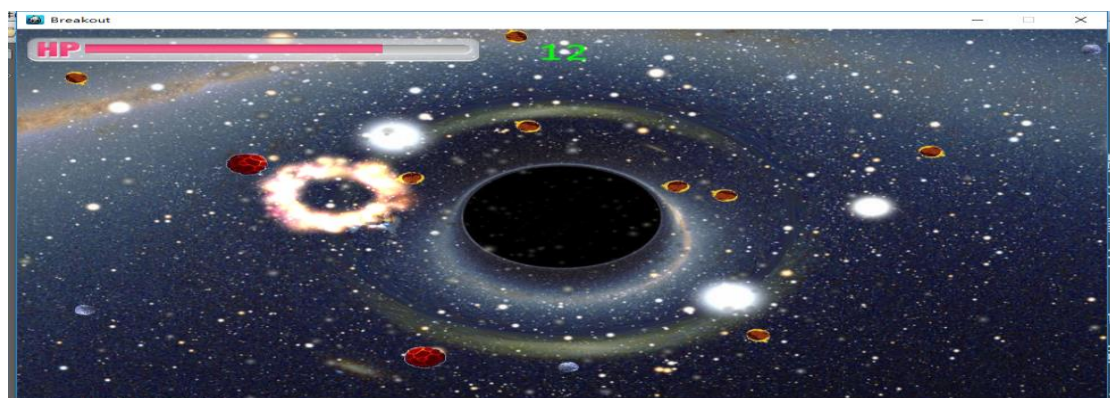
飞船掩码

```
player->getPhysicsBody()->setGroup(1);  
player->getPhysicsBody()->setCategoryBitmask(1); // 001  
player->getPhysicsBody()->setCollisionBitmask(2); // 010  
player->getPhysicsBody()->setContactTestBitmask(2); // 010
```

陨石掩码

```
re->getPhysicsBody()->setGroup(tag);  
re->getPhysicsBody()->setCategoryBitmask(2); //010  
re->getPhysicsBody()->setCollisionBitmask(3); //011  
re->getPhysicsBody()->setContactTestBitmask(1); //001
```

陨石相互碰撞反弹不爆炸，飞船碰到陨石播放音效动画并且血量减少



血量为 0，游戏结束，可继续重玩本关，点 exit 则清空关卡信息



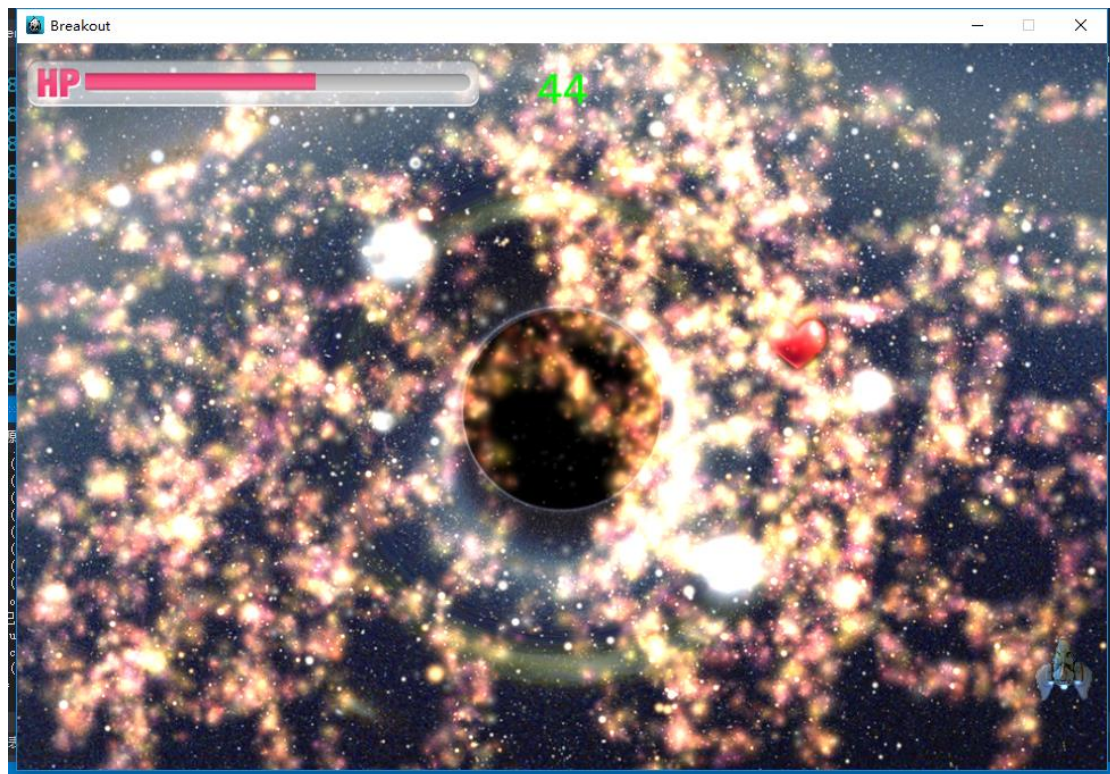
坚持一定时间则过关，可继续下一关，第一关 10 秒，第二关 20 秒，以此类推。



当陨石数量大于 30 的时候，随机出现清屏炸弹和恢复血量的道具



清屏炸弹效果，屏幕内的陨石都产生爆炸粒子



其他改进：

将存陨石的容器改为 `list`，因为会经常删除，使用 `list` 减少开销。

陨石在后期因为经过多次和边缘的碰撞会停止下来，于是将边缘的材料弹性设置为 `1`，陨石的弹性设置为 `0.7`，这样基本后期陨石速度虽有减少还是能保持运动状态。

四．实验过程遇到的问题

- ① 一开始飞船和陨石的掩码设置不合理，没设置 `ContactTestBitmask`，陨石和飞船碰撞后两者反弹，可是一直没有触发爆炸和音效的事件，仔细看了课件，才知道 `ContactTestBitmask` 是确认是否对碰撞做出反馈，添加了 `ContactTestBitmask` 后可产生爆炸效果。
可是陨石相碰却直接穿过去，总感觉不太好，仔细考虑后重新设置了飞船和陨石的掩码，终于实现飞船和陨石相碰陨石爆炸，陨石相碰反弹。
- ② 实现鼠标监听的时候，移动鼠标控制飞船移动，一开始用 `setPosition`，可是动作很不连贯，仔细看了 `demo` 中移动飞船的方法，原来是设置速度，而速度是一个向量，刚好鼠标的移动偏移量就是一个向量，于是直接将偏移量设置为速度，可是偏移量太小，将偏移量扩大 `70` 倍后飞船基本能跟上鼠标的移动。
- ③ 使用清屏炸弹时，一开始使用一个变量存储爆炸粒子，遍历陨石的容器，播放粒子特效后将陨石移除，可是程序一直崩溃，`debug` 了很久发现是爆炸粒子变量的问题，

于是修改代码，解决了问题。

```
for (auto it = enemys.begin(); it != enemys.end(); it++) {  
    auto tmp = ParticleSystemQuad::create("explode.plist");  
    tmp->setPosition((*it)->getPosition());  
    this->addChild(tmp);  
    (*it)->getNode()->removeFromParentAndCleanup(true);  
}
```

五．思考与总结

- ① 把程序分解成一个个小的部分，分而治之，更有效率而且更容易排错。
- ② 回调函数功能很强大，在动画后面执行一个回调函数很方便，不过读写外部变量会产生读写权限冲突，以后使用要多注意。