

OSLab1 引导程序与游戏

实验报告

吴澄杰

151220122

151220122@smail.nju.edu.cn

2017年3月20日

一、实验进度

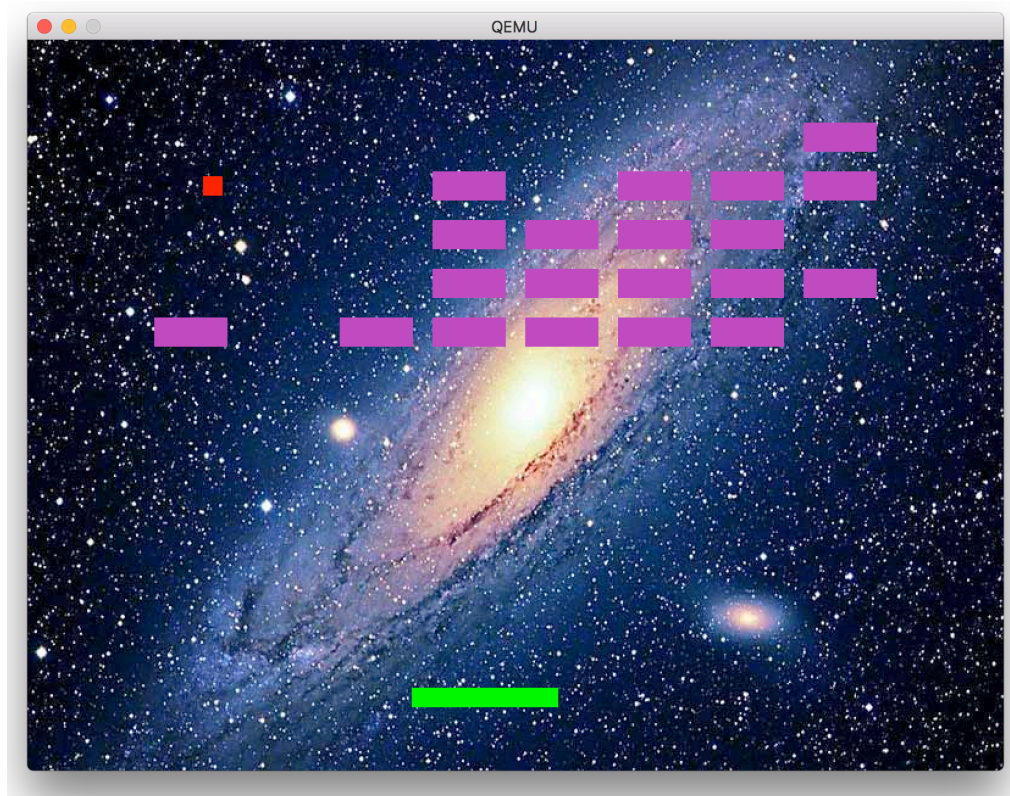
此次实验，我完成了所有内容，实现了一个直接运行在硬件上的 breakout clone（打砖块）小游戏。

二、游戏说明

1、打砖块小游戏以控制横板反弹小球，消去所有的砖块为获胜条件。Space、Enter可以控制游戏的开始、暂停、重新开始，Q可以退出游戏，左右方向键控制横板的左右移动。

2、游戏用24位真彩色输出图像，获胜与失败都有图像提示。

3、小球在砖块上的反弹，会根据碰撞的位置进行判断反弹方向，使得碰撞反应更加真实。同时，小球在横板上不同位置反弹，会给予其速度不同的反馈，这样能够增强游戏的趣味性。



（游戏过程截图）

三、实验心得

1、经过此次实验，我了解到了操作系统对硬件的抽象的重要作用和意义。此次实验中直接运行在硬件上的小游戏需要进行boot、初始化设置、直接与硬件端口交流等各项十分繁琐的工作，这对应用程序来说是极其不便的。

此次实验中的小游戏，在某种程度上发挥着操作系统的作用，其游戏逻辑才是真正的小游戏本身。无论是时钟中断、键盘中断还是24位真彩色输出，都可以看作是操作系统的功能实现了这些控制后由小游戏进行的调用。这充分说明了操作系统作为资源管理者和提供抽象的双重重要作用。

2、在这次实验中，我也真正了解到了硬件的启动机制，即一台计算机是如何从通电bios到最终用户程序得以运行的。在完成实验的过程中，需要了解这些机制的前后关系以及实现的方法。这些机制对操作系统对硬件操作的实现是十分重要的。

3、实验一开始会感觉没有头绪，对如何在硬件上直接实现一个图形化界面的游戏感到迷茫。因此，必须要一步一步进行，按照讲义上的内容首先将操作系统部分的加载、中断处理好，然后再处理游戏的逻辑。即使游戏运行在硬件上，也需要用操作系统的思维，去为游戏逻辑提供借口。

4、此次实验在复习了计算机系统基础中对计算机硬件的认识的同时，通过编写一个小游戏，体会了操作系统如何去调用与管理的基本概念与方法。

5、由于工程较大，并且考虑到后续实验的可维护性，文件的组织也是一个问题。boot、kernel和lib要划分清楚，并且要尽量减少相同文件的复制。kernel和lib的划分清楚，实际上就是操作系统进行的硬件提供的接口与用户程序使用的库函数的区分。这一点是比较重要的，区分有利于为今后实验提供便利。

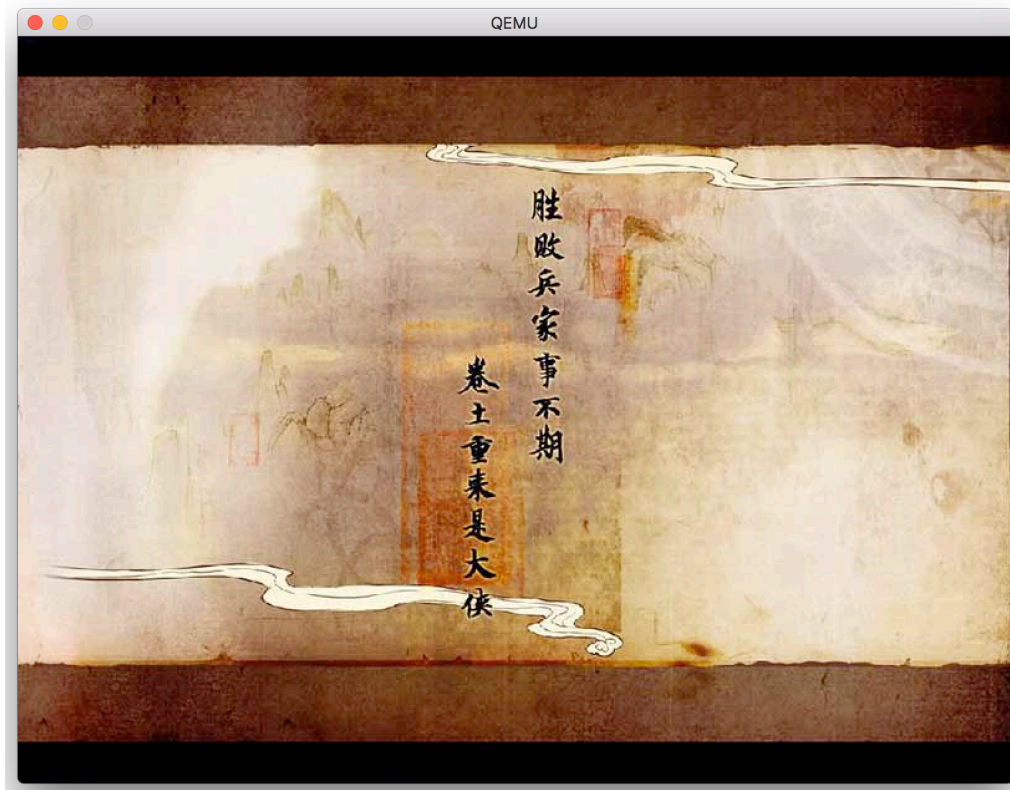
四、游戏设计相关

1、时钟中断：时钟中断到来，就更新球的位置，并且重新绘制球和砖块。同时，用判断逻辑判断球是否与砖块或横板相撞。若相撞则相应地改变速度。

2、键盘中断：在游戏的主循环中，会一直查询是否有按键输入。若检测到输入按键，则更新横板的位置，并且重新绘制横板。

3、小球与砖块的碰撞逻辑：由于小球从不同的位置，以不同的位置与砖块相撞，其反射速度应该具有不同的值，因此游戏中有专门的判断逻辑，来确定小球究竟是撞到侧面从而水平速度取负，还是撞到顶面或侧面从而竖直速度取负。具体方法是通过小球速度的斜率与小球到碰撞点的斜率进行比较得出。

4、速度改变：在游戏中，小球碰撞横板不同的位置，是会给水平速度的大小造成改变的，这样就增强了游戏的趣味性与丰富性。不这样的话，小球的运行其实是一个确定的轨迹，每次游戏都一样，就没有意思了。具体实现是，撞到横板两端1/3速度改变2个单位，中间1/3速度改变1个单位，撞到当中则水平速度不变。



(游戏失败截图)