

一个二维弹幕射击游戏

苏起冬

2017 年 11 月 25 日

目录

1 基本设计和操作说明	1
2 细节说明	2
2.1 文件结构	2
2.2 物理部分	2
2.3 非战斗画面	2
2.4 面向过程	3
2.5 struct object	3
2.6 BGM和音效	3
2.7 跟踪弹的实现	3
2.8 奇数弹和偶数弹	3
2.9 关卡脚本	3
3 存在的问题	4

1 基本设计和操作说明

说在前面 我读的书少，不是很会写文章。

设计原则 够用就行，能跑就行。

所谓的设计 要做一个二维弹幕设计游戏，这是一个不小的工程。于是我做了一个计划，大概是要实现的功能：

- 一个能动的飞机
- 一些能动的敌机
- 它们可以发射子弹
- 它们之间的互相碰撞会造成伤害
- 可以发射可以跟踪的导弹，但是这个导弹的运动要受到一定的限制，否则无法摆脱

- 敌机被打爆之后会爆出奖励
- 可以使用超级武器清空全场子弹和导弹，并对敌机造成大量伤害
- 爆出的奖励可以进行武器升级、转换为生命和超级武器
- 一些有套路的弹幕如同时向四周发射的弹幕奇数弹偶数弹
- 为了设计关卡方便地操纵敌机的行动设计一种低级的脚本
- 死了之后有一段无敌时间
- 好看一点
- **BGM和音效**

其中加粗的部分是选做内容。然后我就按照要求一个个做下来了，整体框架仍然是助教一开始提供的那几个文件和函数，并没有对框架有过多的修改。开发的时候也非常随意，想到什么就做什么了，所以可能整体的设计还是有缺乏考虑的地方。

本人还是有那么一点审美要求的，所以尽自己所能在改善画面。当然限于技术水平，并没有很好看。要不是因为要准备很多考试，我觉得我还能做更多东西。

设定和操作说明 方向键、WASD控制自机移动，z键发射子弹，x键发动超级武器攻击，shift低速模式，esc暂停。自机碰到子弹、导弹、敌机就会爆炸，爆炸之后有三秒钟的无敌时间。敌机被击毁后会掉落奖励，可以用来提升武器性能，恢复生命值和获得使用超级武器的机会。初始状态下，有3条生命和2个超级武器。

2 细节说明

2.1 文件结构

```
/
├── bin/ ..... 可执行文件
├── res/ ..... 资源文件
│   ├── script/ ..... 关卡脚本
│   ├── image/ ..... 图片文件
│   ├── fonts/ ..... 字体文件
│   ├── record/ ..... 记录
│   └── music/ ..... 声音文件
├── doc/
├── main.cpp ..... 主程序
├── object.h
├── object.cpp ..... 实现各种对象
├── script.h
├── script.cpp ..... 完成对于关卡脚本的解析
├── record.h
└── record.cpp ..... 实现对于记录的读取和写入
```

2.2 物理部分

我把原来助教写的控制自机运动的加速度、摩擦等部分去掉了，通过直接控制自机的速度而不是加速度来控制自机。因为我觉得指哪打哪，我让它动就动，让它停就停才比较爽。当然这可能是个人理念的差异。

非常简单的碰撞模型，把所有对象的碰撞检测范围设为一个圆，就判断这些圆的位置关系。此外为了照顾到游戏体验，把自机的圈做小，敌机的圈做大，给玩家一种自己很厉害的感觉。

2.3 非战斗画面

本人设计了欢迎画面、高分记录、游戏结束等非战斗画面，使程序的完整性得到了增强（让它看起来不那么像用来交作业的）。

2.4 面向过程

这个大作业并没有使用很多面向对象的特性，相反地主要采用了面向过程的风格，可能主要是个人的喜好问题，我觉得面向过程已经非常够用了，即使是SDL本身也没有使用面向对象，这并不影响它实现一些抽象数据结构。我也没有使用很多STL提供的容器。整体的数据都是靠数组和struct组织起来的，主要使用指针而不是引用。

2.5 struct object

一开始设计的时候，为了方便管理、少写一点函数，想把所有画面中的对象（自机、敌机、子弹、导弹、奖励道具）都用同一个结构体类型来表示，导致它变得非常巨大。如果当初使用面向对象的思想，搞点继承之类的，就没有那么多麻烦事了。

2.6 BGM和音效

上网学习了一下SDL_mixer的用法，实现了BGM和音效的播放。一开始程序无法正常的工作，后来自己从网上下了一个SDL2就可以了，大概是dll版本不对，我觉得非常坑爹。

具体实现起来没什么难度。SDL_mixer中的声音分为两种，一种是music，一种是chunk，前者是用于播放整体的背景音乐，后者是用来播放比较短暂的音效。程序一开始要分配一下channel的数量，chunk通过channel播放出来。

还有一些更深入的功能，但是基本功能已经实现，我也没有去进一步了解。

2.7 跟踪弹的实现

对于一枚跟踪弹来说，描述其运动状态的量有两个，一个是速度矢量，一个是加速度矢量。我设计了一种跟踪弹的运动方式：

1. 速度和加速度的模是恒定的
2. 加速度处于速度的法向，并指向自机位置

这样既起到了跟踪的作用，又限制了跟踪弹的机动能力，不至于无法摆脱。

2.8 奇数弹和偶数弹

这两种东西本质上是一种东西，就是向自机发出一系列处于一个弧面上的、均匀分布的子弹。子弹数量为奇数的称为奇数弹，数量为偶数的称为偶数弹。假设自机位置不发生改变如果发射的是奇数弹，必有一枚会击中自机，而偶数弹则不会击中。

所以放很多偶数弹也可以给玩家造成一种他水平很高的假象。默认关卡一开始会有很密的弹幕，但是如果想过关的话其实站着不动就可以了。

2.9 关卡脚本

设计了一种很原始的，体验不怎么好的用来设计关卡的脚本语言，功能不是很丰富，不过做这种小型游戏已经足够了。默认的关卡文件是res\script下的stage3.game文件，是一个文本文件，可以用任何文本编辑器编辑。

每一条指令的基本格式如下：

```
command object_id argv1 argv2 ...
```

每一个指令最多有两个整型参数和两个实型参数。

new type x y 在坐标(x,y)处新建一个敌机。敌机编号从0开始，依次递增，不因为被击毁或离开战场而减小。

setv id x y 将编号为id的战机的速度矢量设置为(x, y)

shoot id type 令编号为id的战机发射类型为type的子弹

stop id 令编号为id的战机停止设计

r id type num 令编号为id的战机向周围均匀发射num个类型为type的子弹

m id type x y 令编号为id的战机发射一枚类型为type的跟踪弹，初始速度矢量为(x, y)（这个矢量的长度是没有用的，因为速率是设置好的，只能表征方向）

aimed id num type theta 令编号为id的战机发射数量为num，种类为type，张角为theta的弧形自机狙弹幕

delay t 等着t毫秒的逝去

e 表明关卡到此准备结束。等所有敌机离开战场，关卡正式结束。

所有命令都执行完毕后，会回到最后一个e处继续执行，直到自机生命值为0或敌机全部离开战场

使用这种脚本语言，可以“方便”地设计出关卡，并且在关卡变更之后不需要重新编译程序。当然这些都只是非常简单的原型，只适用于大作业这种小型游戏里面。而且此脚本语言功能十分简陋，前期也缺乏有远见的设计，没有类似子程序的代码复用的方法，导致编写起来给人一种繁琐的感觉，

3 存在的问题

- 过度设计

- 想法很美好，但是最后实现的时候失去耐心，随便做了个能用的就草草了事。到后来代码乱写，毫无可读性。
- 游戏平衡性不佳
- 用一个结构体来概括所有的东西不是一个好主意……
- 在很多隐秘的地方隐藏了很多很难复现的bug，不管了
- 刚开始学 L^AT_EX所以report做的很难看……
- 这游戏就跟它的名字一样。