

课 程 设 计 报 告

****

设计（论文）题目： 飞机大战

学 院 名 称： 电子与信息工程学院

班 级 名 称： 网络（单）20-3

成 员 姓 名： 武旭飞 学 号 20401200102

成 员 姓 名： 郭俊文 学 号 20401200126

指 导 教 师： 茅琴娇

2021年7月2日

目录

[1 需求分析 1](#_Toc75526607)

[1.1 应用背景 1](#_Toc75526608)

[1.2 开发环境 1](#_Toc75526609)

[2 系统结构设计 1](#_Toc75526610)

[2.1 系统结构 1](#_Toc75526611)

[2.1.1 程序执行流程 1](#_Toc75526612)

[2.1.2 程序架构 2](#_Toc75526613)

[2.1.3 数据结构 3](#_Toc75526614)

[2.2 模块划分概述 3](#_Toc75526615)

[2.2.1 初始化模块 3](#_Toc75526616)

[2.2.2 定时器及BGM模块 4](#_Toc75526617)

[2.2.3 绘制画面模块 4](#_Toc75526618)

[2.2.4 游戏暂停检测模块 5](#_Toc75526619)

[2.2.5 玩家飞机功能模块 5](#_Toc75526620)

[2.2.6 子弹功能模块 6](#_Toc75526621)

[2.2.7 敌机功能模块 7](#_Toc75526622)

[3 程序模块设计思路及核心代码 8](#_Toc75526623)

[3.1 初始化模块 8](#_Toc75526624)

[3.1.1 读取图像初始化 8](#_Toc75526625)

[3.1.2 玩家飞机初始化 8](#_Toc75526626)

[3.1.3 敌机初始化 8](#_Toc75526627)

[3.1.4 子弹初始化 9](#_Toc75526628)

[3.2 定时器及BGM模块 9](#_Toc75526629)

[3.2.1 定时器 9](#_Toc75526630)

[3.2.2 BGM开始音乐 9](#_Toc75526631)

[3.2.3 BGM子弹击中音乐 10](#_Toc75526632)

[3.2.4 BGM击毁音乐 10](#_Toc75526633)

[3.3 绘制画面模块 10](#_Toc75526634)

[3.3.1 加载背景图片 10](#_Toc75526635)

[3.3.2 加载玩家飞机图片 11](#_Toc75526636)

[3.3.3 加载子弹图片 11](#_Toc75526637)

[3.3.4 加载血量(HP)和分数(SCORE) 11](#_Toc75526638)

[3.3.5 加载敌机图片 12](#_Toc75526639)

[3.4 游戏暂停检测模块 13](#_Toc75526640)

[3.5 玩家飞机功能模块 13](#_Toc75526641)

[3.5.1 玩家移动 13](#_Toc75526642)

[3.5.2 玩家飞机边界检测 14](#_Toc75526643)

[3.5.3 玩家飞机碰撞检测 14](#_Toc75526644)

[3.5.4 玩家飞机血量检测 15](#_Toc75526645)

[3.6 子弹功能模块 16](#_Toc75526646)

[3.6.1 发射子弹 16](#_Toc75526647)

[3.6.2 子弹移动 17](#_Toc75526648)

[3.6.3 子弹碰撞检测 17](#_Toc75526649)

[3.6.4 子弹边界检测 18](#_Toc75526650)

[3.7 敌机功能模块 19](#_Toc75526651)

[3.7.1 创建敌机 19](#_Toc75526652)

[3.7.2 敌机移动 20](#_Toc75526653)

[3.7.3 敌机边界检测 20](#_Toc75526654)

[3.7.4 敌机血量检测 21](#_Toc75526655)

[4 成员贡献 22](#_Toc75526656)

[5 工作日志 22](#_Toc75526657)

[6 系统使用说明书 23](#_Toc75526658)

[7 结束语 24](#_Toc75526659)

[7.1 不足与改进 24](#_Toc75526660)

[7.2 致谢 24](#_Toc75526661)

[7.3 参考文献 24](#_Toc75526662)

# 需求分析

## 应用背景

单机版单人飞机大战游戏，通过按下键盘的WSAD或者小键盘↑↓←→即可操纵飞机移动，按下空格可使飞机发射子弹，游戏上方会有源源不断的2种敌机飞下来，一种小型机，一种大型机。小型机15hp，大型机30hp。开始游戏时玩家血量为100hp。如果血量小于等于0hp或者飞机和敌机碰撞，则飞机死亡。子弹击中敌机则敌机血量-5hp。如果小敌机飞出边界，则我方飞机血量-5hp，小型机飞出边界玩家-5hp，大型机飞出边界玩家-10hp。击毁小型机增加10 score，击毁大型机增加20score。

## 开发环境

操作系统：Windows 10

使用的语言：C语言

开发工具：Visual Studio 2019

版本控制工具：Git

# 系统结构设计

## 系统结构

### 程序执行流程

程序的执行流程首先初始化，之后进入while循环,执行绘制画面模块（主要用来显示画面），然后执行暂停检测模块（检测游戏暂停），随后执行玩家飞机功能，子弹功能，敌机功能三个模块（各自有各自的处理）。

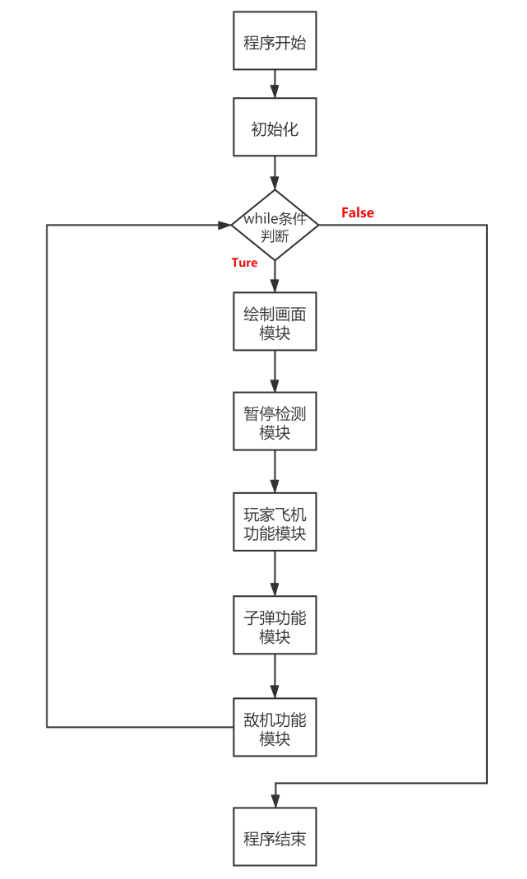


图 1 程序执行流程图

### 程序架构

应用高内聚低耦合的思想，采用三层控制模式，首先数据层由各个链表和图片，音乐组成。业务层即对对应的数据进行操作，而控制层则是人机交互的界面，由键盘和画面，音乐与人机进行交互。

### 数据结构

子弹拥有坐标属性。

敌机拥有坐标，类型属性。

玩家飞机拥有坐标，血量，分数属性。

数据结构由结构体和链表实现。有三条链表，分别是玩家飞机链表，子弹链表和敌机链表。

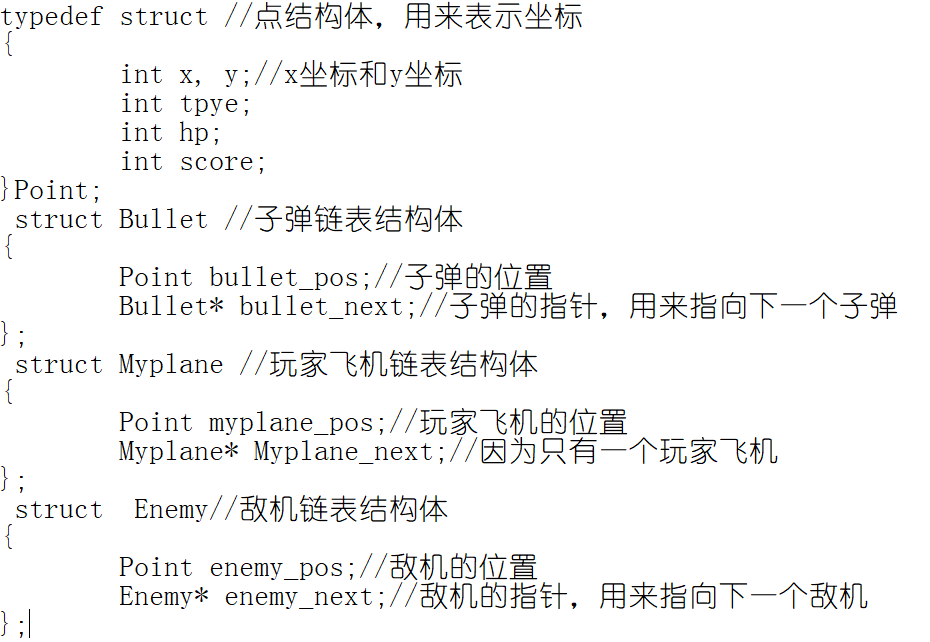


图 2 数据结构

## 模块划分概述

### 初始化模块

初始化模块用来创建玩家飞机链表头，敌机链表头，子弹链表头，并把IMAGE类对象与images文件夹下的图片关联起来。

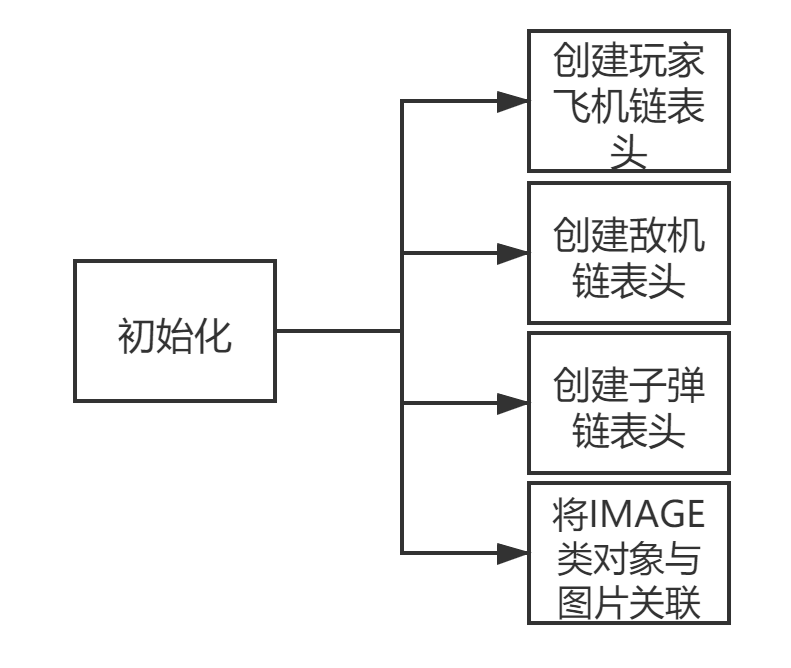


图 3 初始化

### 定时器及BGM模块

定时器及BGM模块不作为程序执行流程的一个大块，它们仅供别的模块调用。

BGM：为子弹击中，敌机死亡，游戏背景音乐三个场景来调用。

定时器：两个传入参数，分别是毫秒和id，毫秒表示指定程序运行了多少毫秒后，定时器返回true，程序通过id来区分用哪个定时器（一共有4个定时器）。这样，每隔传入的毫秒，定时器都会返回一次true，否则返回false。

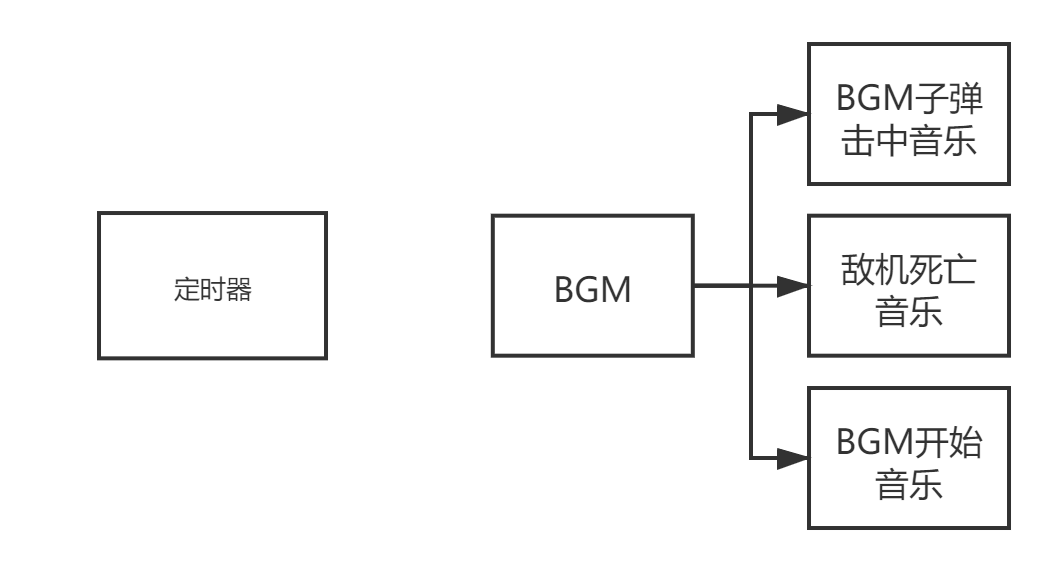


图 4 定时器及BGM模块

### 绘制画面模块

绘制画面模块需要完成游戏画面的绘制，首先绘制游戏的背景图片，之后遍历玩家飞机链表，只要链表中存在一个结点，就绘制一张玩家飞机的图片。绘制敌机和绘制子弹同理，遍历各自的链表，只要链表中存在一个结点，就通过该结点里面的坐标信息绘制到窗口的对应位置上。绘制血量和分数只需要获取玩家的血量和分数，之后再绘制到窗口上即可。

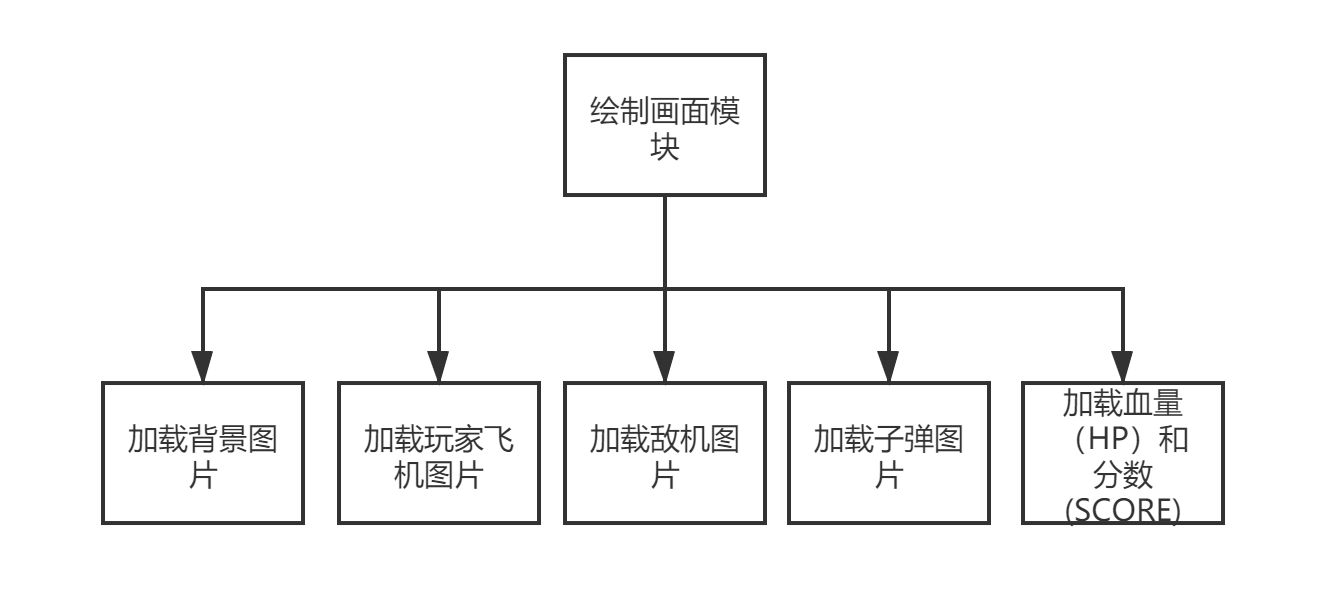


图 5 绘制画面模块

### 游戏暂停检测模块

游戏暂停采用按P检测，按下P后则暂停游戏，之后按下回车即可恢复游戏，并且暂停时暂停游戏背景音乐的播放，恢复后继续游戏背景音乐的播放。

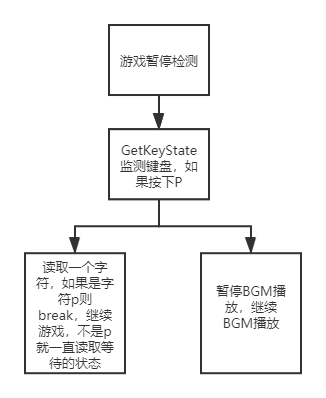


图 6 暂停模块

### 玩家飞机功能模块

玩家飞机功能模块主要有玩家移动，玩家飞机边界检测，玩家碰撞检测，玩家血量检测这4个功能。玩家移动通过检测键盘实现按WSAD或者小键盘的移动键，使得飞机的xy坐标改变。玩家边界检测是来判断飞机不要飞出窗口，如果飞出去了，强制修改坐标，使得飞机在窗口中。玩家碰撞检测用于检测玩家和敌机是否碰撞，如果碰撞了，则玩家血量置为0。玩家血量检测是通过玩家血量hp来判断，如果hp<=0则游戏结束，此时销毁玩家，敌机，子弹三条链表，并弹出对话框让用户选择重新游戏或者退出程序

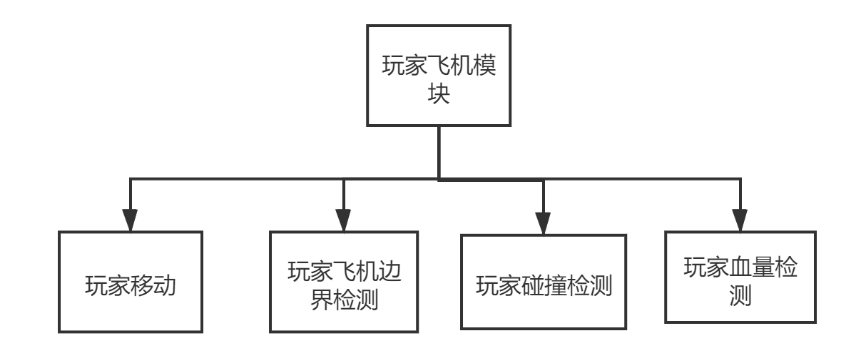


图 7 玩家飞机模块

### 子弹功能模块

子弹功能模块由发射子弹，子弹移动，子弹碰撞检测和子弹边界检测这4个功能构成。发射子弹由按下空格产生一个子弹。子弹移动是固定的时间移动固定的距离。子弹碰撞检测是用来判断有没有击中敌机，如果击中了，该子弹销毁，对应的敌机扣5点血量。子弹边检检测是用来判断子弹一直移动飞出了窗口后，则将该子弹销毁。

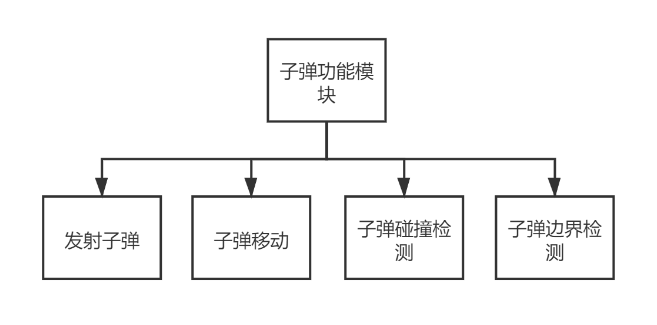


图 8 子弹功能模块

### 敌机功能模块

敌机功能模块有创建敌机，敌机移动，敌机边界检测和敌机血量检测四个功能组成。创建敌机的速度由定时器来控制。 敌机移动也由定时器来控制，每间隔一定的时间敌机坐标向下移动固定的距离。敌机边界检测是如果敌机飞出了边界，说明玩家没有把敌机打掉，所以要扣玩家的血量。敌机血量检测是子弹击中敌机会扣敌机的血，如果敌机血<=0则销毁这架敌机

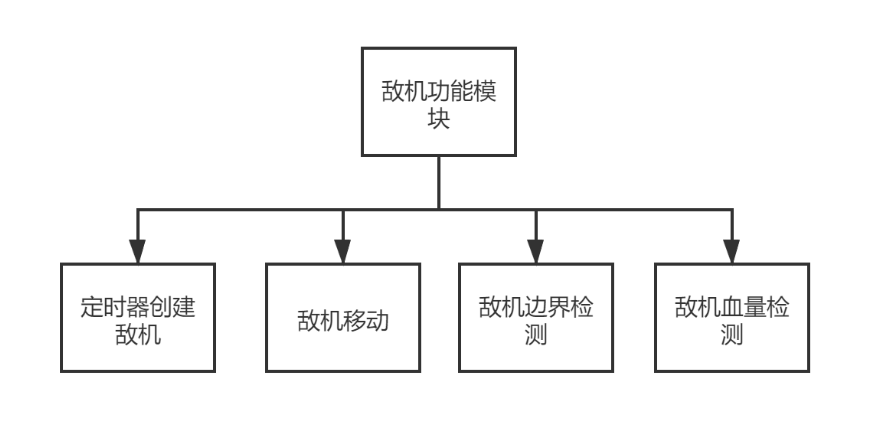


图 9 敌机功能模块

# 程序模块设计思路及核心代码

## 初始化模块

### 读取图像初始化

设计思路：将文件中的图片读入IMAGE类对象。

主要的程序代码：



### 玩家飞机初始化

设计思路：玩家飞机采用链表存储，创建玩家飞机链表头，并初始化飞机的位置，血量，分数。

主要的程序代码：

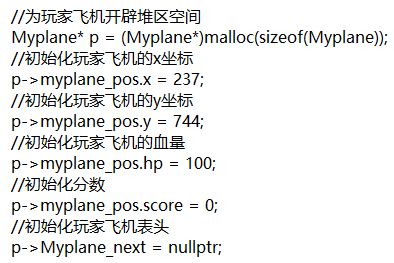


图 10 玩家飞机初始化

### 敌机初始化

设计思路：敌机飞机采用链表存储，创建敌机链表头。

主要的程序代码：

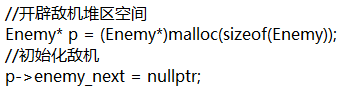


图 11 敌机初始化

### 子弹初始化

设计思路：敌机子弹采用链表存储，创建子弹链表头。

主要的程序代码：

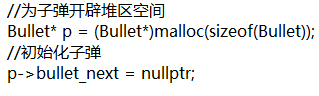


图 12 子弹初始化

## 定时器及BGM模块

### 定时器

设计思路： 利用系统的clock()返回的时间来做对比，记录上一次的clock()，下一次调用时，两个时间相减就能判断程序有没有运行超过传入的毫秒了，如果超过则返回ture，否则返回false

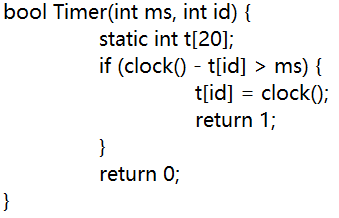


图 13 定时器

### BGM开始音乐

设计思路：开始播放背景音乐。调用windows的api函数mciSendString()函数。

主要的程序代码：



图 14 播放背景音乐

### BGM子弹击中音乐

设计思路：播放子弹击中音乐。调用windows的api函数mciSendString()函数。

主要的程序代码：



图 15 子弹击中音乐

### BGM击毁音乐

设计思路：敌机销毁时调用。调用windows的api函数mciSendString()函数。

主要的程序代码：

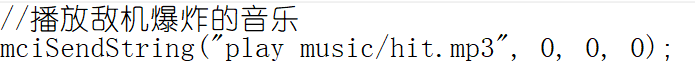


图 16 敌机击毁音乐

## 绘制画面模块

### 加载背景图片

设计思路：加载背景图片。绘制背景图片放在绘制画面函数的第一个，这是逻辑，因为这样就可以覆盖上一次绘制时窗口的画面了。

主要的程序代码：



图 17 绘制背景

### 加载玩家飞机图片

设计思路：hp>0正常图片，hp<=0爆炸图片。

主要的程序代码：

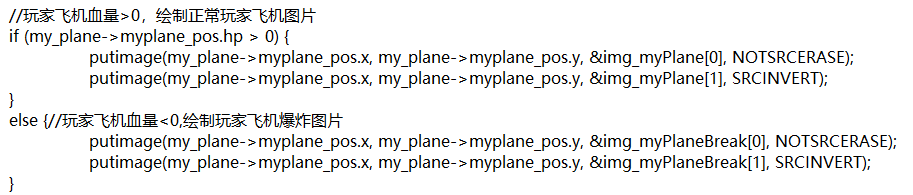


图 18 绘制玩家飞机

### 加载子弹图片

设计思路：遍历子弹链表，将所有子弹图片贴到窗口上，而贴图的位置由链表中的每个子弹属性来决定，属性中有x和y坐标，这样即可控制子弹贴图的位置

主要的程序代码：

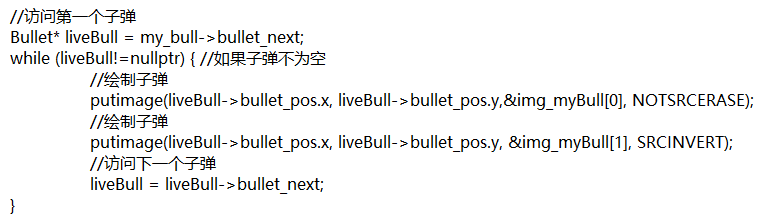


图 19 绘制子弹

### 加载血量(HP)和分数(SCORE)

设计思路：设置血量和分数文本，输出血量和分数。首先设置文字的颜色为RED，之后把文字的背景改为透明，再读取飞机的血量和分数，最后用outtextxy函数把相应文字绘制到窗口上

主要的程序代码：



图 20 绘制血量和分数

### 加载敌机图片

设计思路：遍历敌机链表，将所有敌机图片加载到窗口上。每个敌机都有自己的坐标，这样通过遍历+贴图，可以把不同敌机绘制到不同的位置上

主要的程序代码：

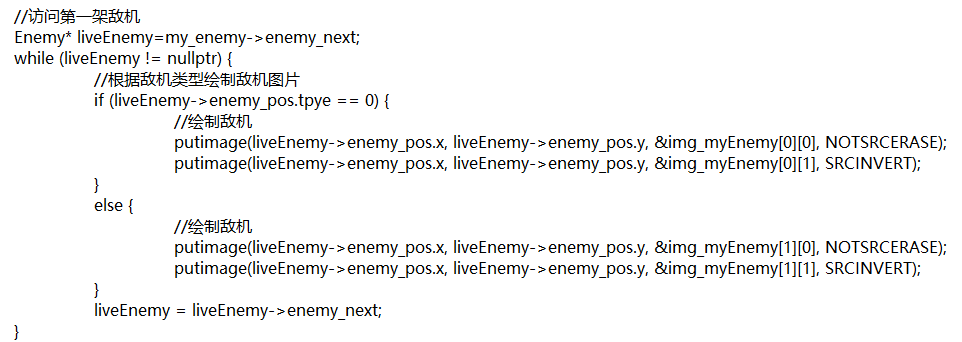


图 21 绘制敌机

## 游戏暂停检测模块

设计思路：键盘按下p时游戏暂停，再次按下p并回车，游戏继续。由GetAsyncKeyState(),判断函数调用时,确定用户当前是否按下了键盘上的一个键的函数。这样，p其实在缓冲区中，只要按下回车就会被getchar()读取，刚好取消粘贴

主要的程序代码：

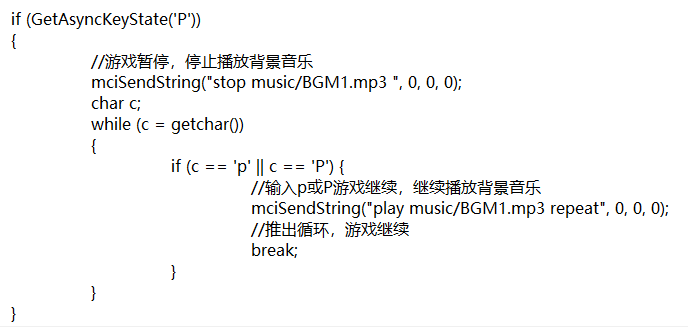


图 22 游戏粘贴

## 玩家飞机功能模块

### 玩家移动

设计思路：玩家移动由GetAsyncKeyState(),用来判断函数调用时,确定用户当前是否按下了键盘上的一个键的函数,所以玩家移动由这个函数来检测WSAD已经小键盘的上下左右,之后改对应坐标就好了

主要程序代码：

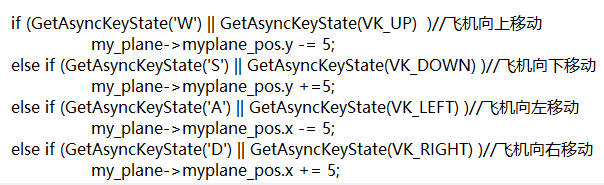


图 23 玩家移动

### 玩家飞机边界检测

设计思路：玩家飞机边界检测则是飞机一半不能飞出窗口左边界和右边界。如果超出边界，将玩家飞机的坐标设置为边界值。

主要程序代码：

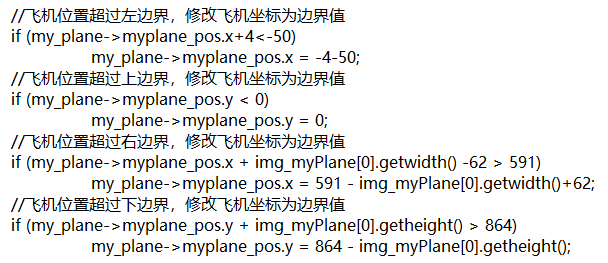


图 24玩家边界

### 玩家飞机碰撞检测

设计思路：通过玩家飞机的坐标，与所有敌机飞机的坐标做判断，根据两个坐标判断是否碰撞。通过两个链表的遍历即可完成

主要程序代码：

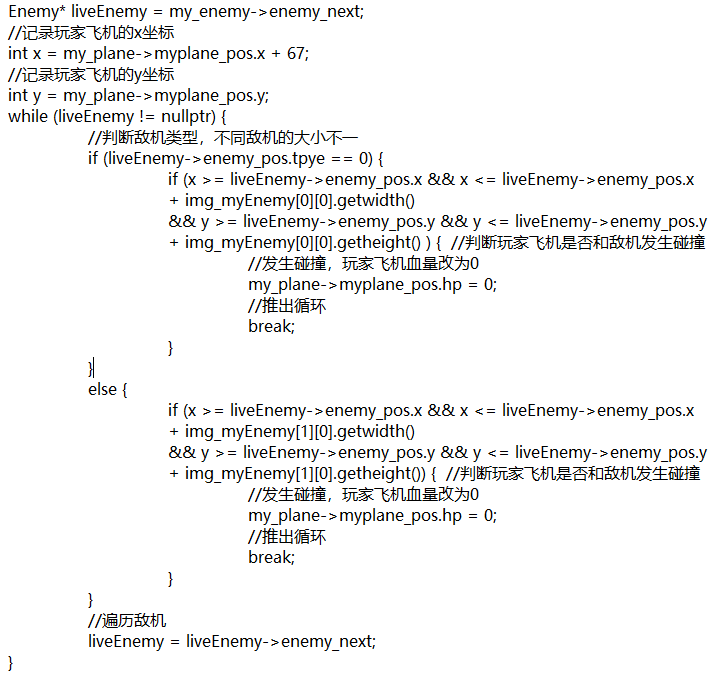


图 25 玩家碰撞检测

### 玩家飞机血量检测

设计思路：判断玩家飞机的血量，如果血量小于等于0，首先销毁飞机，子弹，敌机三个链表，随后MessBox询问玩家是否继续游戏，是则利用goto重新开始游戏，否退出游戏。

主要程序代码：

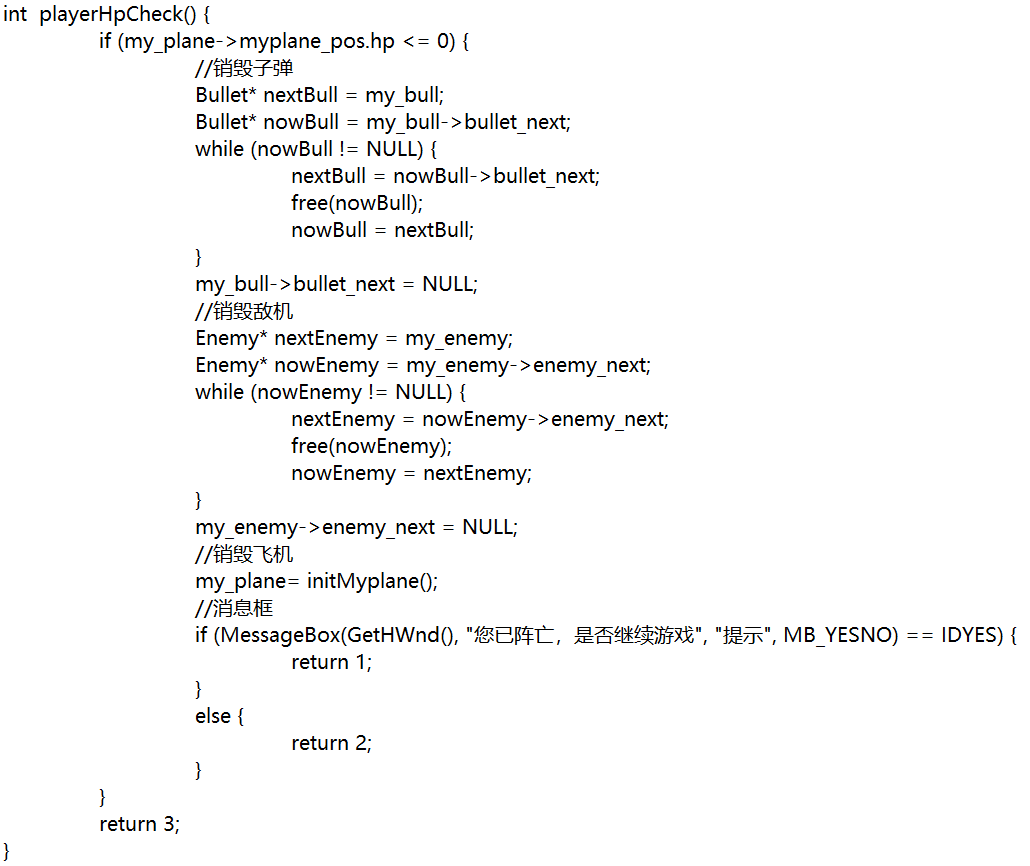


图 26 玩家血量检测

## 子弹功能模块

### 发射子弹

设计思路：监测键盘空格键是否被按下，如果空格键按下，则每隔200ms发射一颗子弹，子弹采用链表形式，在堆区开辟一块内存存放子弹，子弹采用头插法插入子弹链表。

主要程序代码：

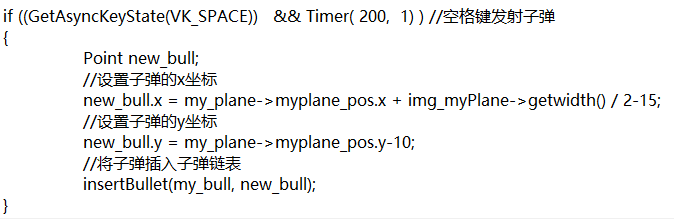


图 27 发射子弹

### 子弹移动

设计思路：Timer[4]每隔20ms改变所有子弹的位置，用子弹头链表头遍历所有所有子弹，所有子弹的y坐标-5。

主要程序代码：

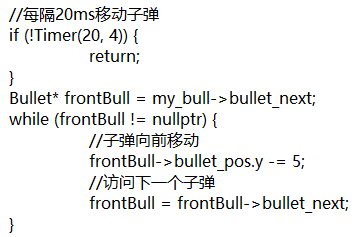


图 28 子弹移动

### 子弹碰撞检测

设计思路：如果子单链表没有遍历到表尾，并且敌机链表没有遍历到表尾，然后根据子弹和敌机各自的坐标判断子弹和敌机是否发生碰撞，如果子弹和敌机发生碰撞，播放子弹击中音乐，敌机血量-5，击中子弹的坐标调为-25，由程序之后子弹边界函数销毁子弹，遍历敌机，待所有敌机遍历完之后，子弹更新为下一个，敌机链表回到表头。

主要程序代码：

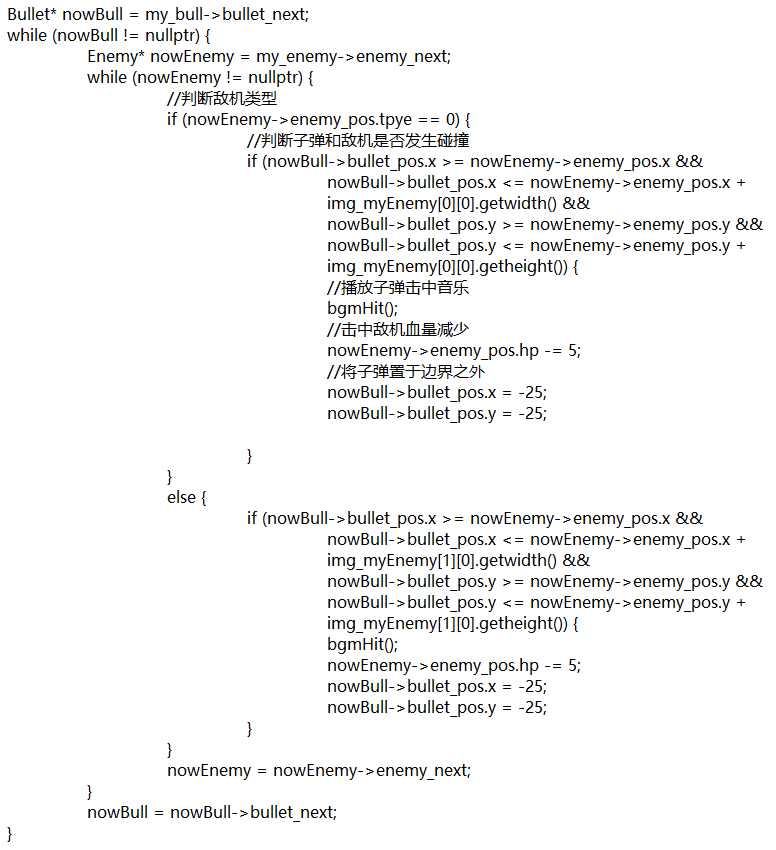


图 29 子弹碰撞检测

### 子弹边界检测

设计思路：Timer[4]每隔20ms改变所有子弹的位置，用子弹头链表头遍历所有子弹，所有子弹的y坐标-5。

主要程序代码：

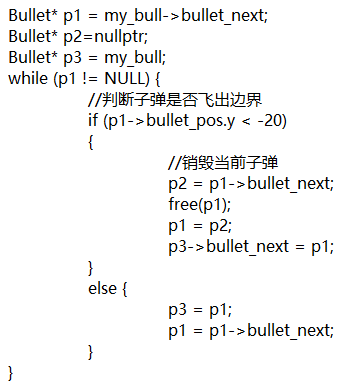


图 30 子弹边界检测

## 敌机功能模块

### 创建敌机

设计思路：根据Timer6,每隔1000ms创建一架敌机，敌机的类型采用随机数生成，初始化敌机的数值，调用insertEnemy()函数将新创建的敌机头插法插入到敌机链表中。

主要程序代码：

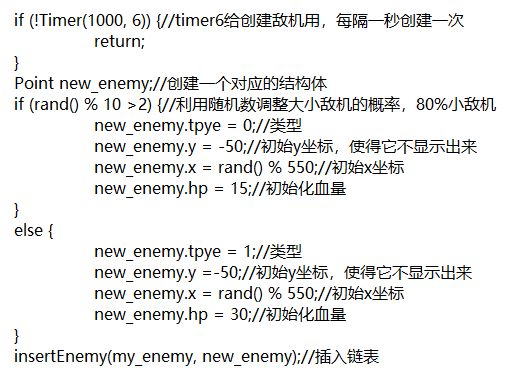


图 31 创建敌机

### 敌机移动

设计思路：根据Timer2，每隔30ms,遍历敌机链表，改变敌机的坐标，将所有敌机的y坐标+2.

主要程序代码：

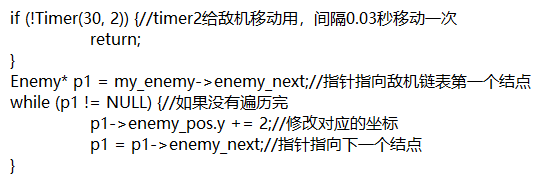


图 32 敌机移动

### 敌机边界检测

设计思路：创建敌机类型指针，遍历敌机链表，判断当前敌机的坐标，如果敌机的y坐标超出了窗口的高度（窗口高度为864），判断当前敌机的类型，如果敌机的类型为0玩家飞机的血量-5，否则玩家飞机的血量-10，销毁当前敌机，访问下一架敌机。

主要程序代码：

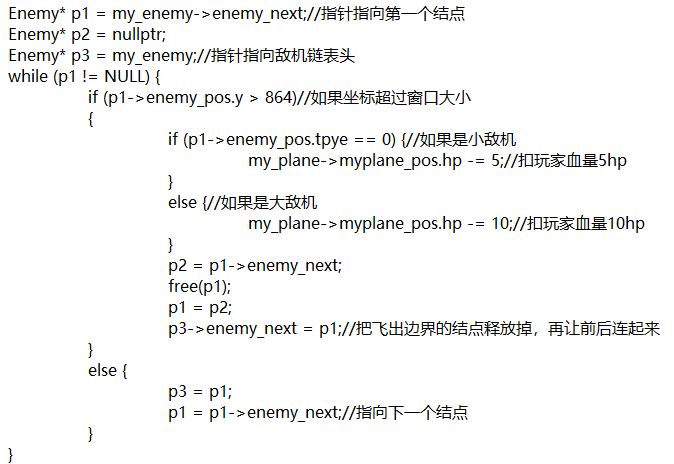


图 33 敌机边界检测

### 敌机血量检测

设计思路：创建敌机类型指针，遍历所有敌机，如果敌机的血量<0,判断敌机类型，如果敌机类型为0，玩家的分数+10,否则玩家分数+20，销毁当前敌机，访问下一架敌机。

主要程序代码：

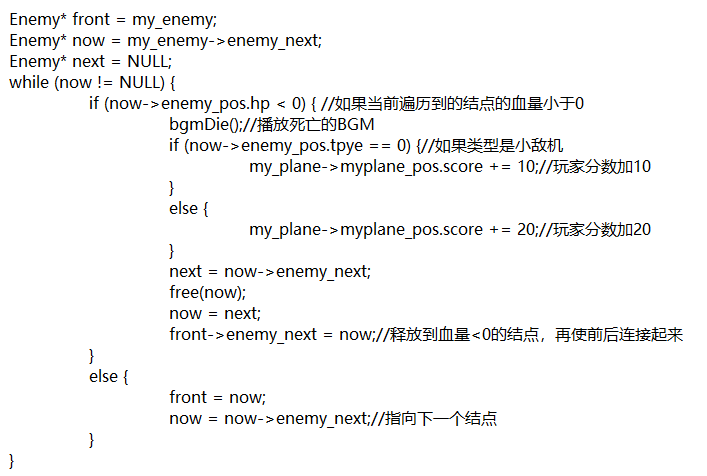


图 34 敌机血量检测

# 成员贡献

|  |  |  |
| --- | --- | --- |
| 学号 | 姓名 | 主要工作 |
| **20401200102** | **武旭飞** | 功能流程图，初始化模块，定时器及BGM模块，绘制画面模块，游戏暂停模块 |
| 20401200126 | 郭俊文 | 课设word报告，玩家飞机功能模块，子弹功能模块，敌机功能模块 |

# 工作日志

|  |  |  |
| --- | --- | --- |
| 时间 | 姓名 | 工作 |
| 2021/6/2 整天 | 武旭飞 | 完成功能流程图 |
| 2021/6/3 晚上8-12点 | 武旭飞 | 初始化模块，定时器及BGM模块，绘制画面模块，游戏暂停模块 |
| 2021/6/3 晚上8-12点 | 郭俊文 | 玩家飞机功能模块，子弹功能模块，敌机功能模块 |
| 2021/6/4 晚上6点 | 武旭飞，郭俊文 | Git合并分支，代码整合 |
| 2021/6/20 | 郭俊文 | 编写课程设计部分 |
| 2021/6/21 | 武旭飞 | 编写课程设计部分 |
| 2021/6/22 | 郭俊文 | 修改课程设计报告 |
| 2021/6/22 | 武旭飞 | 修改课程设计报告 |
| 2021/6/23 | 武旭飞 | 修改课程设计报告 |
| 2021/6/25 | 武旭飞 | 修改课程设计报告 |

# 系统使用说明书

按住WSAD或者↑↓←→可控制飞机移动，按住空格键发射子弹，暂停用P，继续游戏按Enter，死亡后重新游戏或者退出用鼠标点击弹窗即可选择，

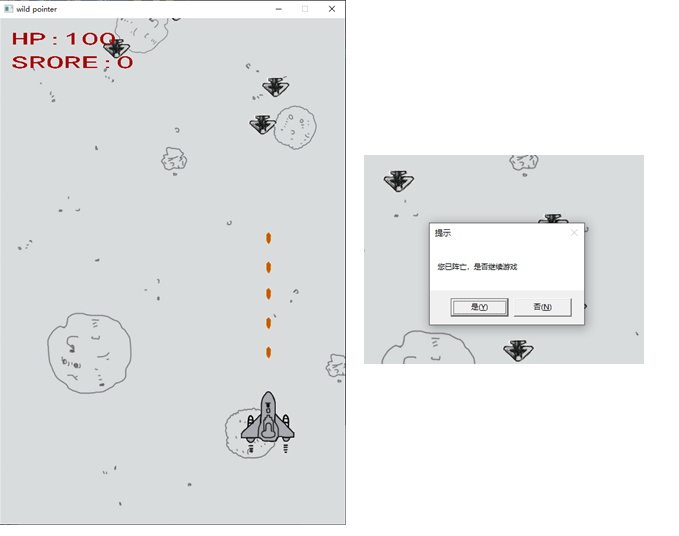


图 35 游戏界面

# 结束语

## 不足与改进

团队协作时因为任务和功能没有交代清楚导致实现和需求有差距，使用git时因为不熟悉导致汇总出了问题，希望今后git能够熟练掌握，团队分工时把任务交代清楚，整体的架构要让组员清楚明白。

## 致谢

课设项目虽然提前完成，但是其中还是有bug不知如何解决，后在老师的指导下修改bug，才有了这份成品，课设报告多次请老师指点，在此感谢茅琴娇老师的教导。

## 参考文献

[1]《C++程序设计（第三版）》，谭浩强主编，清华大学出版社  
[2]《C++ Primer Plus（第六版）中文版》，[美]Stephen Prata著 张海龙 袁国忠 译，人民邮电出版社  
[3]《数据结构与算法分析-C++语言描述(第四版)》[美]Mark Allen Weiss 著 冯舜玺 译 中国工信出版社 电子工业出版社  
[4]《C++编程思想》，[美]Bruce Eckel Chuck Allison 著，机械工业出版社  
[5]《C++程序设计语言（第4部分：标准库）》，[美]Bjarne Stroustrup 著，机械工业出版社  
[6]《C/C++学习指南》，邵发 著，清华大学出版社  
[7]《Effective C++ 改善程序与设计的55个具体做法（第三版）》 [美]Scott Meyers 著 侯捷译，电子工业出版社