**北京邮电大学软件学院**

**2016－2017学年第一学期实验报告**

**课程名称： C#程序设计实践**

**项目名称： 打飞机小游戏**

**项目完成人：**

**姓名：\_\_\_肖逸敏\_\_\_\_\_学号：\_\_2014211990\_\_\_\_\_\_**

**姓名：\_\_\_\_李银\_\_\_\_学号：\_\_\_2014211983\_\_\_\_\_**

**姓名：\_\_\_郭一卿\_\_\_\_\_学号：\_\_\_2014212060\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_**

**姓名：\_\_\_\_\_\_\_\_学号：\_\_\_\_\_\_\_\_**

**指导教师：\_\_\_\_卢本捷\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**日 期： 2016 年 12 月 27 日**

1. **实验目的**

通过这次实验熟练C#语言的使用，学会用C#语言解决现实中需要解决的问题，设计出自己想要的程序，区分C#和C语言，C++在使用上的不同，更深一步地了解C#语言。

1. **实验内容**

我们这个项目是通过c#语言完成的一个打飞机的小游戏。游戏主要有单人模式，双人竞技模式两种模式。单人模式中，玩家不限定游戏时长，直到生命值小于或等于0时游戏结束。双人竞技模式中，玩家其中之一生命值小于或等于0时游戏结束，另一方取得胜利。单人游戏中玩家的最后成绩会和之前的排行榜进行比对并添加。

1. **实验环境**

Visual Studio 2013，DirectX 9.0平台，C#语言，WinForm控件，Microsoft Visual Basic环境

1. **实验结果**

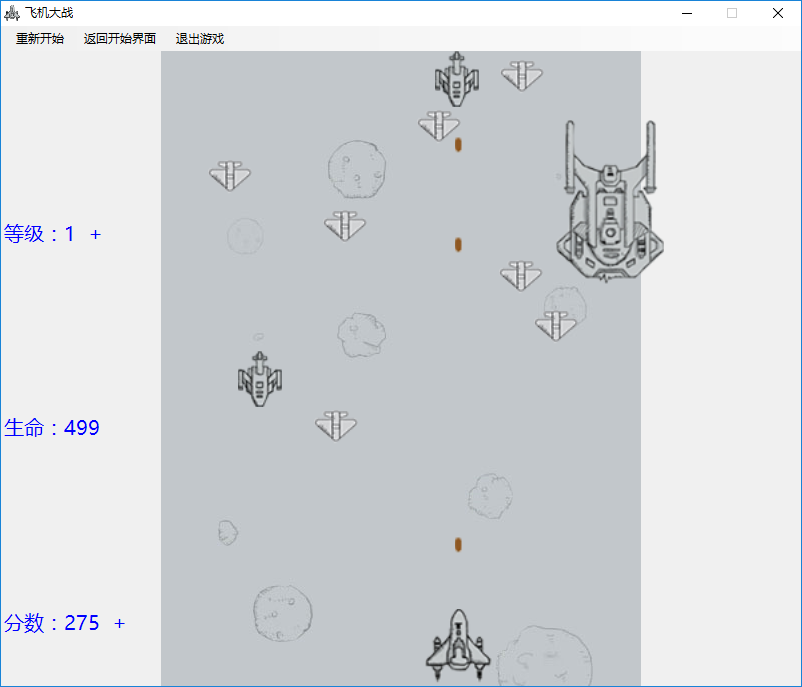
游戏开始界面：



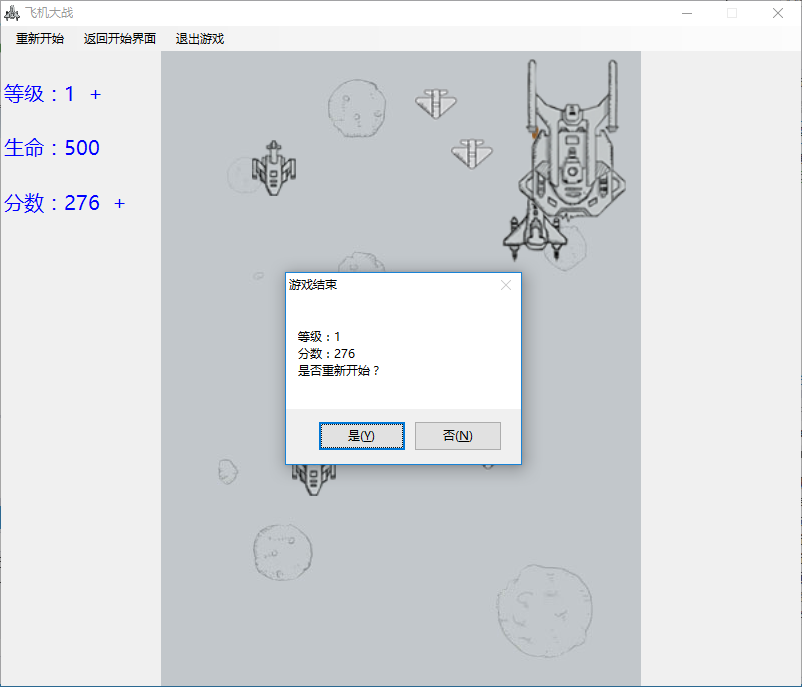
排行榜初始值：



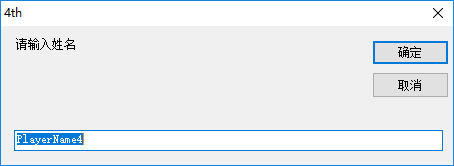
进入单人模式：



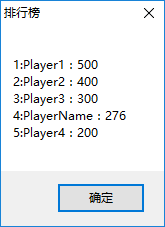
单人模式结束：



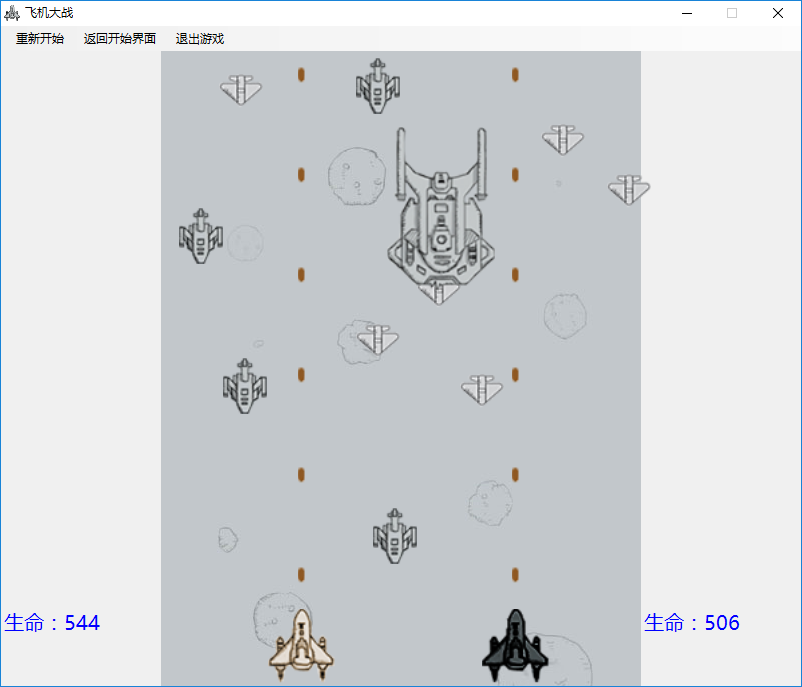
退出单人模式后分数的记录：



排行榜的变化：



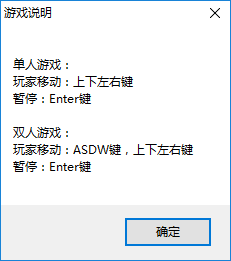
进入双人模式：



双人模式结束：



游戏规则说明：



1. **附录**

## 项目背景：

之前有一款很有名的微信打飞机小游戏，简笔画风格，十分受人欢迎。我们团队就是基于这款游戏用C#语言开发了一款打飞机小游戏，更是开发了不同的功能。

### 项目分工：

肖逸敏同学主要是面向对象和窗口编程；

李银同学主要是程序控制；

郭一卿同学主要是组件封装。

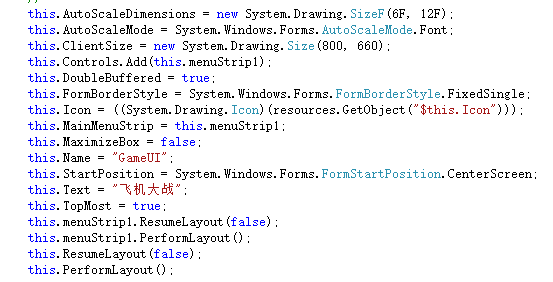
### 项目架构设计：

项目主体主要分成四个大的部分：

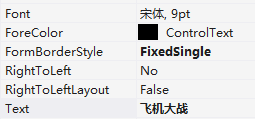
* 窗口编程（事件驱动）
* 组件封装（属性方法）
* 程序控制（语法逻辑）
* 面向对象（继承封装多态）

第一阶段：完成游戏框架的搭建，完成游戏背景功能。

* + - 1. 创建项目Game\_Fly
      2. 更改窗口名为GameUI
      3. 更改窗口属性，例如大小等

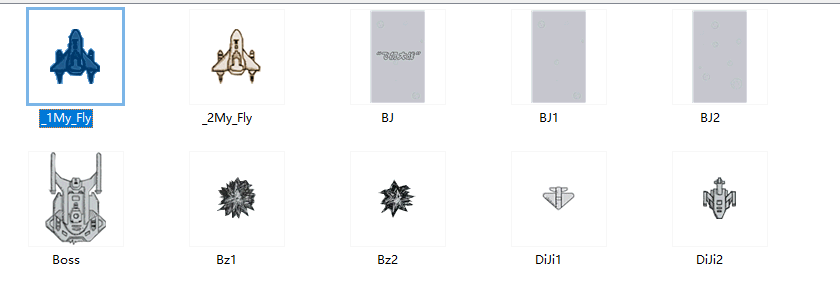


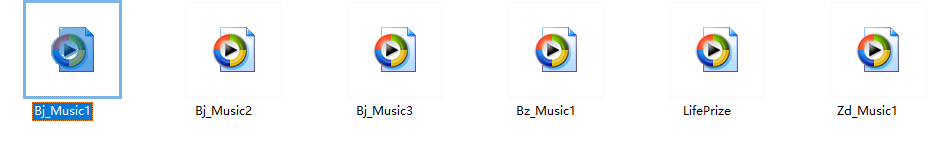
这是部分的窗口属性改变后的代码，真实的窗口改变要更简单，直接在下方改变属性即可。



这是属性改变的一部分，举例说明。

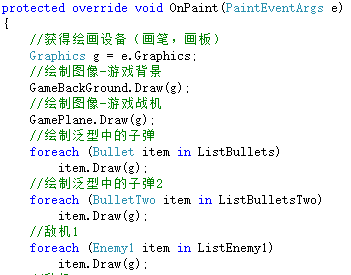
* + - 1. 添加资源，主要是音频（背景音乐，子弹飞出等），图片（各种敌机，飞机，子弹等）

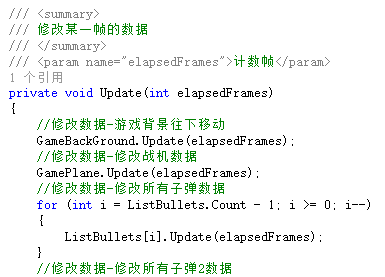




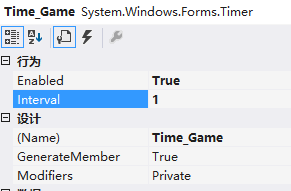
创建资源文件，直接向内部添加。

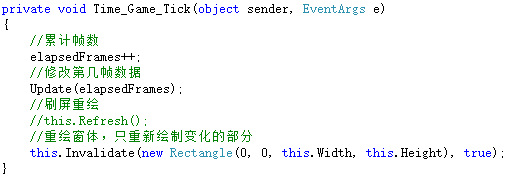
* + - 1. 添加画图功能，定时功能，背景更新功能





从工具箱中添加定时器，控制游戏的开始和结束

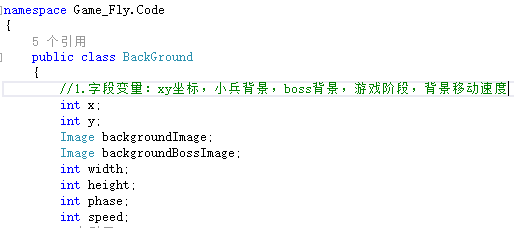




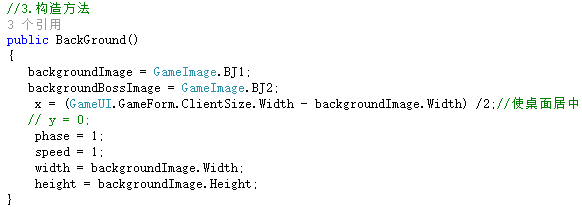
第二阶段：完成背景，飞机，子弹，敌机，奖励的封装

1.背景封装

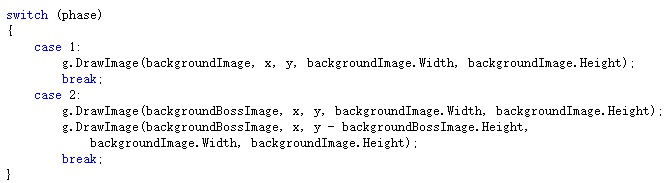
属性值：

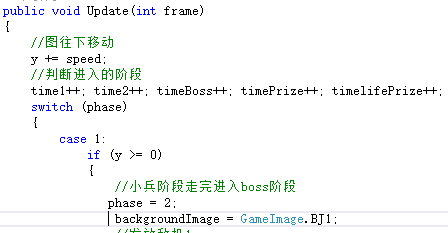


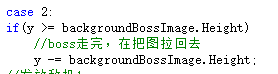
构造函数：



下面部分与GameUI相对应，保证背景的卷动，可以使游戏更有真实感。

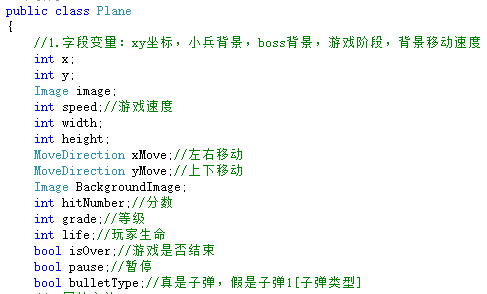




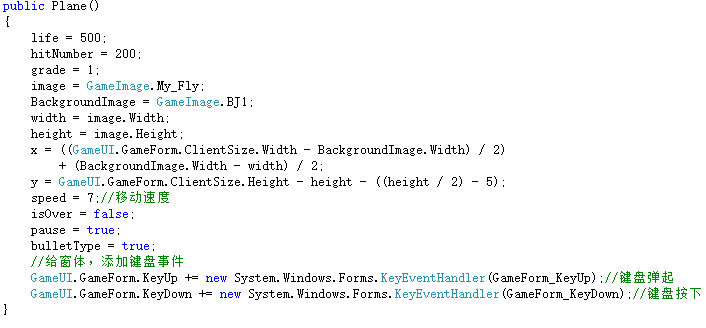


2.飞机封装

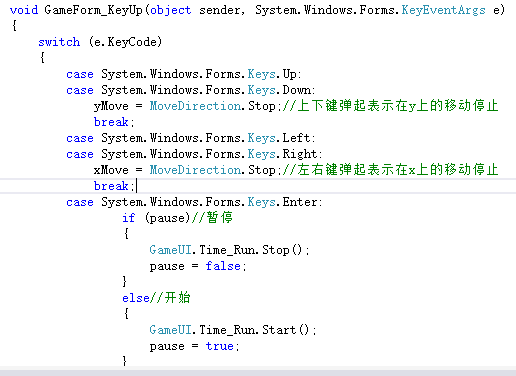
属性值：

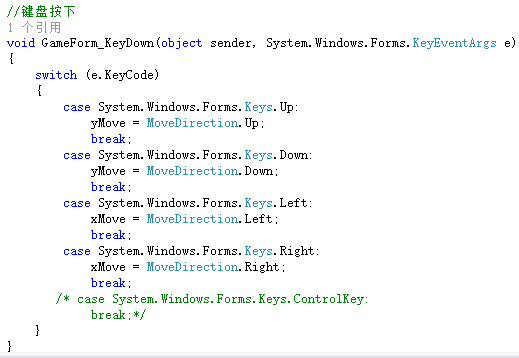


构造函数：

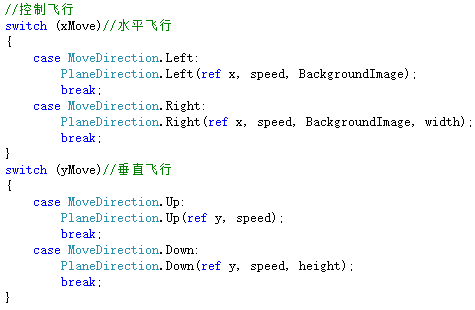


下面两个函数是使用键盘控制飞机的动作：

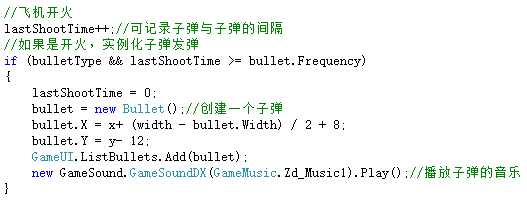




在动作函数中，还要控制飞机不飞出界面：



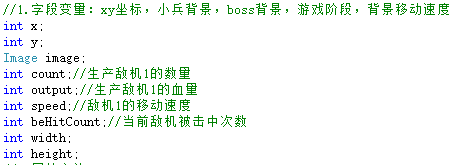
这是控制飞机开火的部分。在这部分中，实例化子弹发射，还有子弹发射的音乐。



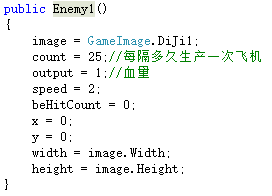
3.敌机封装：

由于本项目为了体现多样性，有三种不同敌机，这里只显示其中一种。。

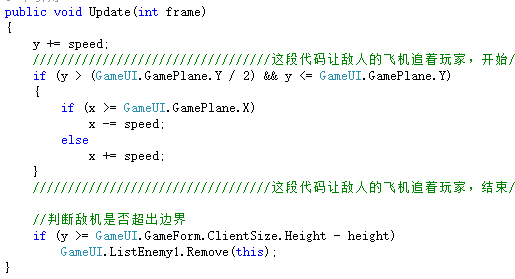
属性值：



构造函数：

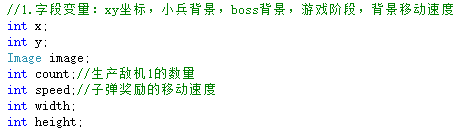


在Update中有两个函数，一个是保证敌机在一定范围内会追着飞机跑，一个是判断敌机是否飞出了y轴边界

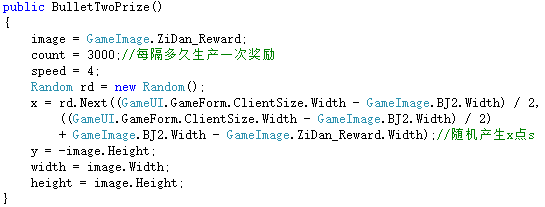


4.奖励封装

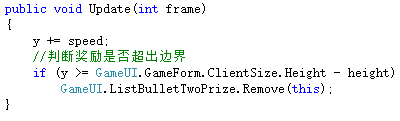
本项目中奖励有两种，一种是更新子弹为双弹的奖励，另一种是添加飞机的生命值的奖励。这里只显示其中一种。

属性值：  


构造函数：

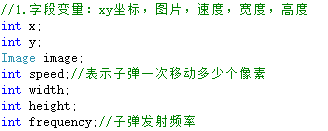


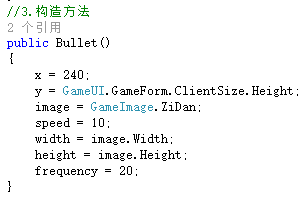
判断奖励在y轴的边界飞出：

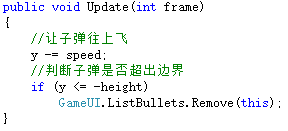


5.子弹封装：

本项目有两种子弹，双子弹只有在接收到双子弹奖励后才能出现，这里只显示其中一个

属性值：  


构造函数：  




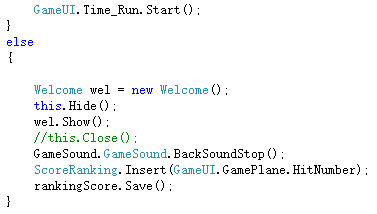
第三阶段：主界面的重构

1. 游戏开始：

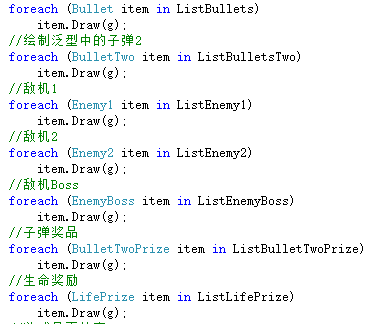


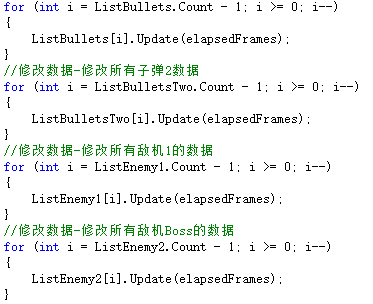
1. 游戏结束。如果想重新开始，将不记录成绩，重新开始游戏；如果放弃，则返回开始界面，记录成绩入排行榜中。





在GameUI中，使用for循环和foreach循环，可以绘制所有的敌机，子弹等和更新所有的敌机，子弹等的数据。





第四阶段：完成程序控制。

1. 判断撞击函数

子弹与敌机的碰撞：

为了显得更加真实，当敌机撞死时，会有爆炸的画面和声音。

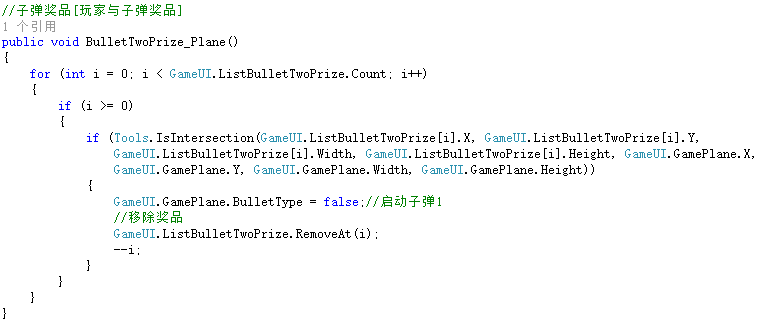


飞机与敌机的碰撞：

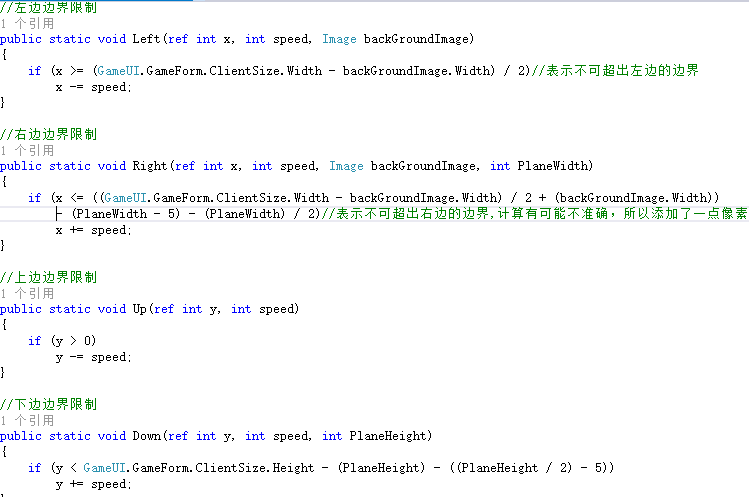
为了显得更加真实，当敌机撞死时，会有爆炸的画面和声音。



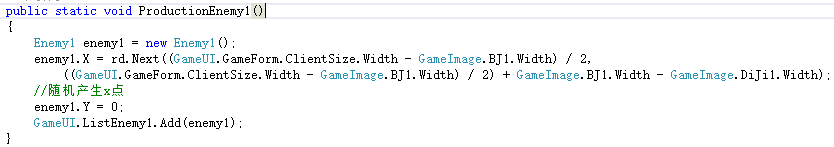
飞机与奖励的碰撞：



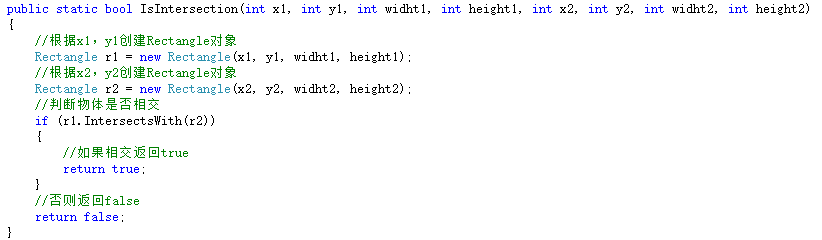
2.控制飞机不飞出界面：



1. x轴创建敌机：



1. 判断深度相交函数：



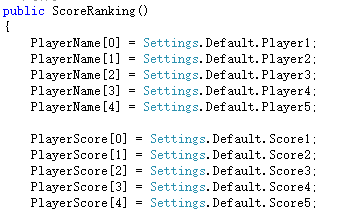
第五阶段：设计排行榜

我们将排行榜的数据放入Properties.Setting中，这样使即使项目变换位置也可以保证排行榜的存在。

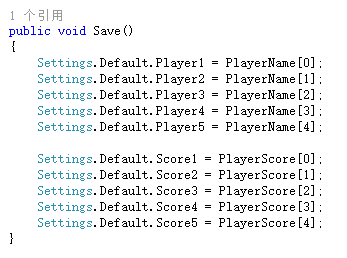


排行榜的代码类为ScoreRanking.cs，

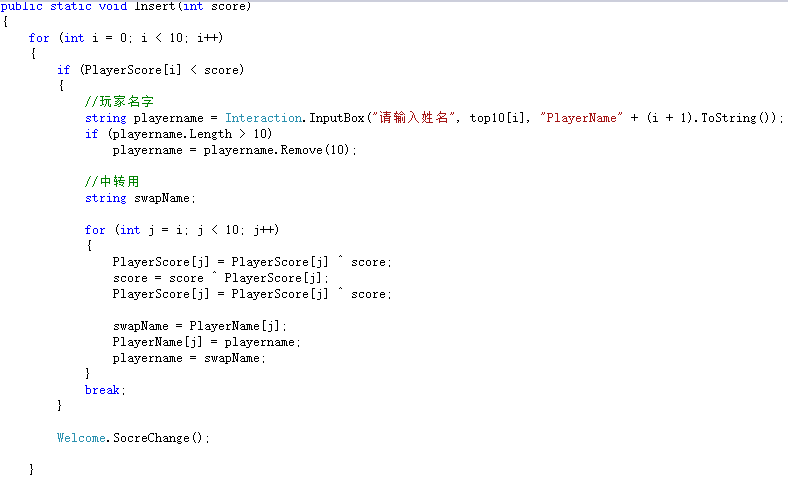
构造函数初始化：



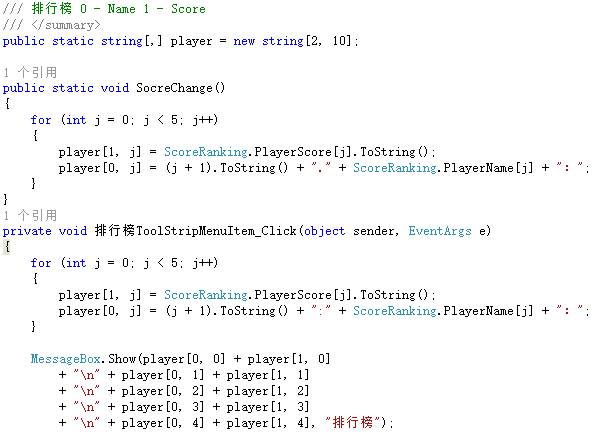
Save函数将更新的数据存入Setting中：



下面是将取得的新成绩插入Setting中：

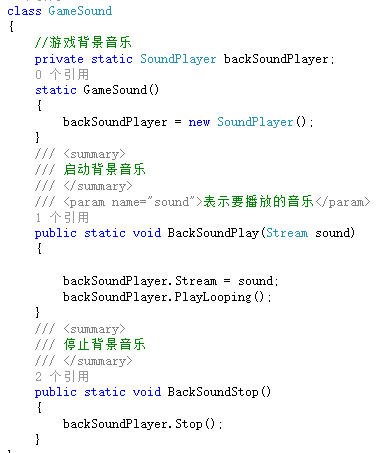


这是在开始界面的调用和显示：



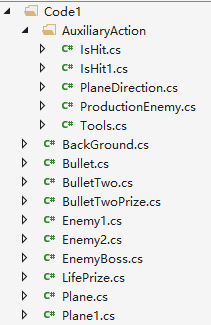
第六阶段：声音的设置

声音的设置引用了DirectXSound的方法，因此在引用处需要引用安装的DirectXSound。



第七阶段：设计双人版打飞机游戏

由于部分原因（在下面的项目错误报告中会详细提到），本小组未能实现代码的复用，因此重新创建一个包进行调用。



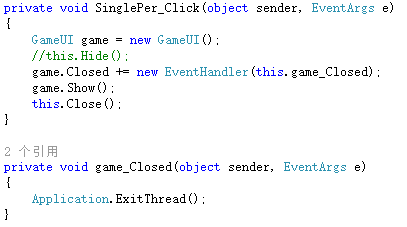
在这里，我们重写了所有包。同时为了区分两个不同的飞机，我们创建了Plane和Plane1两个类，分别封装两个飞机。IsHit类也写成两个，也是为了区分两个飞机的分数。

第八阶段：设计开始界面和单人模式，双人模式界面的按钮细节

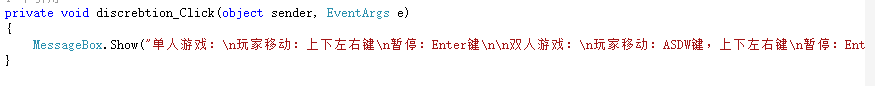
1. 开始界面：

开始界面上有五个按钮，分别是进入两种模式的两个按钮，显示游戏规则的按钮，退出游戏按钮，显示排行榜的按钮。

进入其他窗体界面函数：

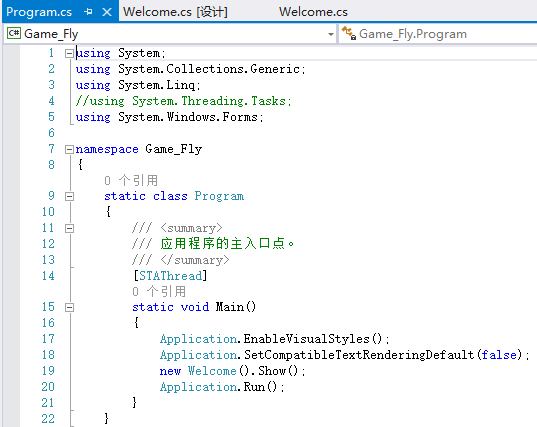


显示游戏规则按钮：



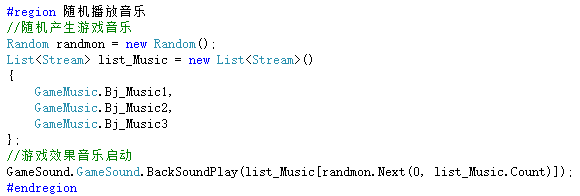
显示排行榜的按钮代码在第五阶段已经展示过，这里不赘述了。

下面是将开始界面设置为main的代码：

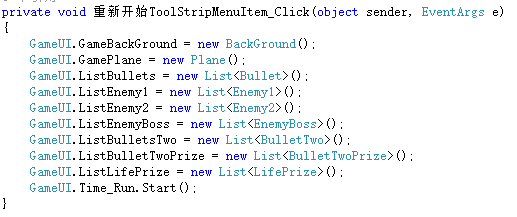


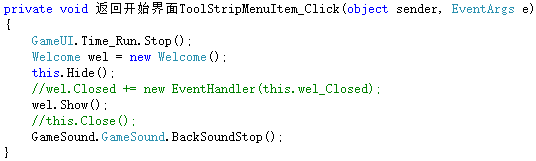
1. 单人模式界面：

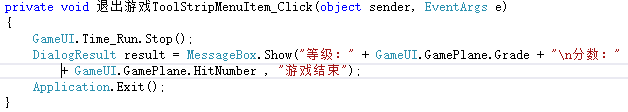
本游戏有三种背景音乐，每次打开都是随机选择音乐：



这个界面上有三个按钮，分别是重新开始，返回开始界面和退出游戏。







3.双人模式界面：

内里代码和单人模式类似，就不再赘述了。

至此，整个项目的架构设计完成。

### 项目错误记录

1. 组件封装的问题：
2. 背景帧开始只设置了一个，想的是背景帧轮流滚动，但这样写的时候中间总会有一大段的空白区。上网查过背景帧滚动时发现需要设置两个背景帧进行轮流的滚动。于是我们设置了两个背景帧进行轮流的滚动，最终完成了背景帧滚动的问题。
3. 飞机在飞行时最开始没有设置上下左右的界限，导致我们一运行就飞到找不到的地方了。因此特意负责封装的同学让负责程序控制的同学写了飞行的界限函数进行调用，完成了界线的限制。
4. 在飞机控制开火的部分，开始没有设置子弹与子弹之间的间隔，导致子弹运行时报错，后来上网进行查证以后加上了子弹之间的间隔，解决了这个问题。
5. 最开始设计敌机飞行的时候并没有添加敌机跟着飞机跑的功能，是后来我们上网试玩别人的打飞机游戏时，发现别人的游戏中的敌机可以跟着飞机跑，因此我们也商量着添加了这个功能。
6. 在游戏结束后的出现的提示信息由于使用了MessageBox，结果出现了一系列的提示信息。最后通过调整定时器停止和提示信息出现的顺序解决。
7. 程序控制的问题：
8. 在设计排行榜时，最开始我们设计的是本地文件，后来发现网上有一个例子是使用的Properties.Setting，我们商量后觉得更符合WinForm的特点，后来就将排行榜的本地目录添加入了Properties.Setting。
9. 设计排行榜时，第一次完成的时候没有初始化所有的排行榜，导致虽然排行榜已经在Setting中初始化，但直接打开时全部为Null，程序报错。
10. 在分数记录时，最开始由于结果通知消息和分数记录消息的顺序问题导致分数记录的输入框不能出现在窗体前面而是后面。后来通过调整出现顺序解决了这个问题。
11. 窗口编程的问题：
12. 在开始运行时发生了闪屏问题，后来我们添加了双缓冲，解决角色移动时的屏幕闪烁问题。
13. 在窗口跳转的时候，首先使用的是this.Close()。后来运行时发生当尝试开启游戏窗体关闭开始窗体时，会使两个窗体同时关闭，后来发现因为这是单线程，因此当一个窗体关闭，整个线程关闭，导致程序关闭。后来使用的this.Hide()，但这会导致当关闭游戏窗体开启开始窗体时游戏音乐关不掉。因此我们添加了关闭游戏音乐的函数，完成了窗口的跳转。
14. 为了避免开启游戏窗体关闭开始窗体时关闭进程的问题，我们设计的是开始窗体即使关闭，进程也不能关闭，因此如果直接关闭开始窗体，也会导致进程没办法关闭。为了解决这个问题，我们直接在开始窗体上设计了退出游戏的按键，保证能够关闭开始窗体而不连累游戏窗体。
15. 在设计双人游戏时，我们设计了很多个版本。最早是设计双人共同游戏，，其中一个阵亡后消失，剩下一个继续运行。但当写完后，发现没有办法将创建的plane消除掉。本来最开始是想用delete函数，后来发现C#中没有delete函数。最后我们没能解决这个问题，因此放弃了这个做法。之后我们设计两个窗体共同工作，相互竞争的模式，准备设计一个父窗体，里面嵌套两个子窗体，子窗体里面是两个飞机。但最后因为两个窗体没办法同时实现聚焦的问题，即两个窗体不可能同时有焦点，类似于两个浏览器不可能同时进行操作，最后也放弃了这个做法，最后决定两者进行结合，在一个窗体中进行竞争。
16. 在设计两个飞机时，由于与一个飞机是使用不同的窗体，因此在调用时不能调用相同的参数的函数，因此本项目代码复用性较差，是重新创建了一个包专门实现双人游戏而不是调用同一个包内的函数。
17. 在直接使用DirectX时出现了版本不兼容的问题，通过百度，我们的解决方法是将项目属性中的应用程序的目标框架改到4以下，生成中的目标平台，活动平台都改成x86.将框架改到4以下会报错，把报错行（不能引用的部分）注释掉就行。

### 项目总结：

在这次C#的大实验的学习中，我们学会了C#语言的使用，同时通过使用也明白了很多C#语言和C语言，C++语言之间的共通和不同。当然除了语言的使用，更多的是小组的共同工作的沟通能力，包容能力和分工能力。这个项目更像一个整体，实际上并不能很清晰地区分出每个人具体做了什么，更多地是大家一起讨论，协力做出的产物。在整个项目中，我们记录出现的问题，相互交流解决，不仅让我们学到了更多解决不同问题的方法，更是学到了和他人交流共同解决问题的能力。