**C语言项目报告**

**项目名称：\_\_\_C语言飞机大战\_\_\_\_\_\_\_\_**

**项目成员：\_\_\_刘博橦，白玉晗\_\_\_\_\_\_\_**

**填写日期：\_\_\_\_2020.9.5\_\_\_\_\_\_\_**

1. **摘要（Abstract）**

本项目是一个弹幕射击游戏，采用C和C++编写，使用Easyx作为图形库编写图形界面，采用多文件的编写方式。本项目并不涉及十分复杂的算法，但另一方面我们项目的功能模块很多，程序的整体结构和逻辑较为复杂，并且完成度很高。同时我们设计并实现了较为美观的界面以及弹幕，通过大量的动画效果使得整个游戏过程富有动感，赏心悦目。此外我们大量使用链表的形式去组织游戏中产生的大量实体，使用文件记录玩家历次游玩的数据。

1. **问题描述（Problem Statement）**

作为第一次编写较大型的C语言项目，我们选择了游戏本身逻辑并不复杂的弹幕射击游戏（stg）来编写。这个课题不涉及到复杂的算法，并且依靠C/C++我们完全可以进行项目的全部开发过程，无需借助其他语言。我们将在游戏的编写中大量应用类和链表的概念，以实现代码更好的结构性以及程序较高的运行效率。另外我们将尽可能地实现美观的界面和动态的游戏画面效果。

这其中我们需要解决的主要问题有游戏运行的可视化，游戏类、游戏各种实体的类的声明，游戏中大量实体的组织，自机函数，敌机函数，游戏主函数以及游戏数据储存（文件的使用）的编写等。

1. **组内分工（Group Division）**

本小组成员包括：刘博橦，白玉晗。

具体分工：

刘博橦：

1.构建游戏的整体框架（包括整体架构的设计、主函数的编写、各个类的声明等）

2.可视化（包括素材的准备、动画功能的实现以及图形界面的编写等）

3.关卡Boss及其弹幕的设计与编写

4.部分代码的修改

白玉晗：

1.三种自机主武器的设计与编写

2.道具系统和分数系统的设计与编写

3.所有涉及文件操作的部分（包括使用文件存储玩家数据、存储replay数据以及使用配置文件记录游戏参数设置）

4.游戏背景故事及情节设定等

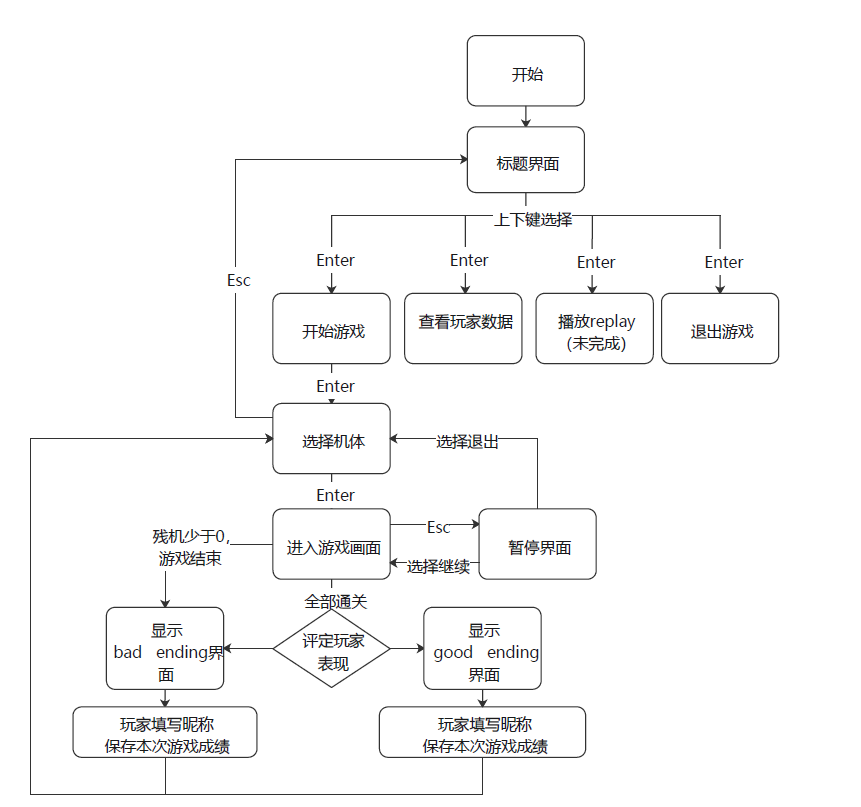
工作量所占百分比刘博橦占65%，白玉晗占35%。

1. **分析（Analysis）**

由于我们仅仅进行了一个学期的C语言学习，对图形引擎和面向对象的程序设计思想还不甚了解，因此面对这样一个对我们来说十分“庞大”的项目，只能在不断的学习和尝试中逐步推进。我们的整体思路是：先解决最基本的图形素材的加载和显示问题—构建主界面—以最简单的静态图形素材实现游戏的最基本功能即玩家（自机）的移动和出界判定（自机被限制在游戏区域内）—实现动画功能—实现自机的射击行为（涉及到链表的添加和删除操作）—实现敌机和敌机弹幕功能—实现自机和敌机之间的各种判定—实现关卡—实现文件读写玩家数据。至此游戏主要功能已构建完成，剩余工作则是向各个类中封装更多的变量，编写更多的类函数以及声明更多的子类，以此向游戏中不断封装更多其他功能。这样做能够保证每推进一点都能进行有效的调试。

1. **设计（Design）**

程序整体流程如下：



1. 最基本部分的构建：

开发项目时初步构建游戏的主要框架，首先学习了Easyx的相关函数实现图像资源的读取并实现一个标题界面。此时仅提供两个功能即开始游戏和退出，标题界面是一个死循环，只有当玩家选择退出时才会退出循环；若选择开始游戏，则调用游戏主函数Game()（后续添加功能后会先进入角色选择），其中同样是一个死循环，每帧调用各个具体的游戏函数。然后学习了C++的面向对象思想，声明一个“玩家”类，包括玩家的当前位置，并编写成员函数“显示”和“移动”。游戏通过接受键盘输入实时更新玩家的位置，并对玩家是否出界做出判定，使得玩家只能在游戏区域内移动。

1. 动画功能的实现：

以自机为例。游戏中读取的图像文件指针被存储在了一个IMAGE类型的全局数组中。在“玩家”类中添加以下成员变量：起始帧（IMAGE指针）、动画帧数、当前帧、播放速度、上一帧播放时间。绘图函数中先绘制当前帧，然后对当前时间和上一帧时间的距离进行判定，若大于播放速度，则“当前帧”指针前移，并更新“上一帧时间”；若“当前帧”超出了“帧数”的范围，则依照需求是否将播放指针归位。以此便实现了动画的播放功能。

1. 子弹（自机和敌机）功能的实现：

以敌机为例进行介绍。不同类型的敌机分别为“敌机”类衍生的不同子类，在父类中已声明虚函数“射击”即为敌机的攻击函数，而每一个不同的敌机都有着不同定义的“射击”函数，其内容是：当游戏时间达到某些节点时，使用new函数生成一个子弹对象，并将其链接到一个已经存在的链表上。该链表的头指针被声明在“关卡”类中，因此只要关卡存在，该链表便一直存在。若子弹被判定消亡（击中或出界），则将其从链表中删除。此外子弹的运动轨迹同敌机一样，是由以时间为参数的参数方程控制的，而且参数方程中也有一些其他参数决定了子弹初始方向等，而这些参数也是一些其他因素如发射波数等的函数。由此便实现了屏幕上华丽的弹幕效果。

1. 存取玩家数据功能的实现：

每次游戏结束后，将本次游戏的时间、分数、残机数等数据写入data.dat中。以后再进行游戏时，若玩家选择“玩家数据”则可在游戏中查看历次成绩，游戏将每一次的成绩读入一个结构体数组并对分数进行排序，然后将排好序的游戏数据呈现在屏幕上。

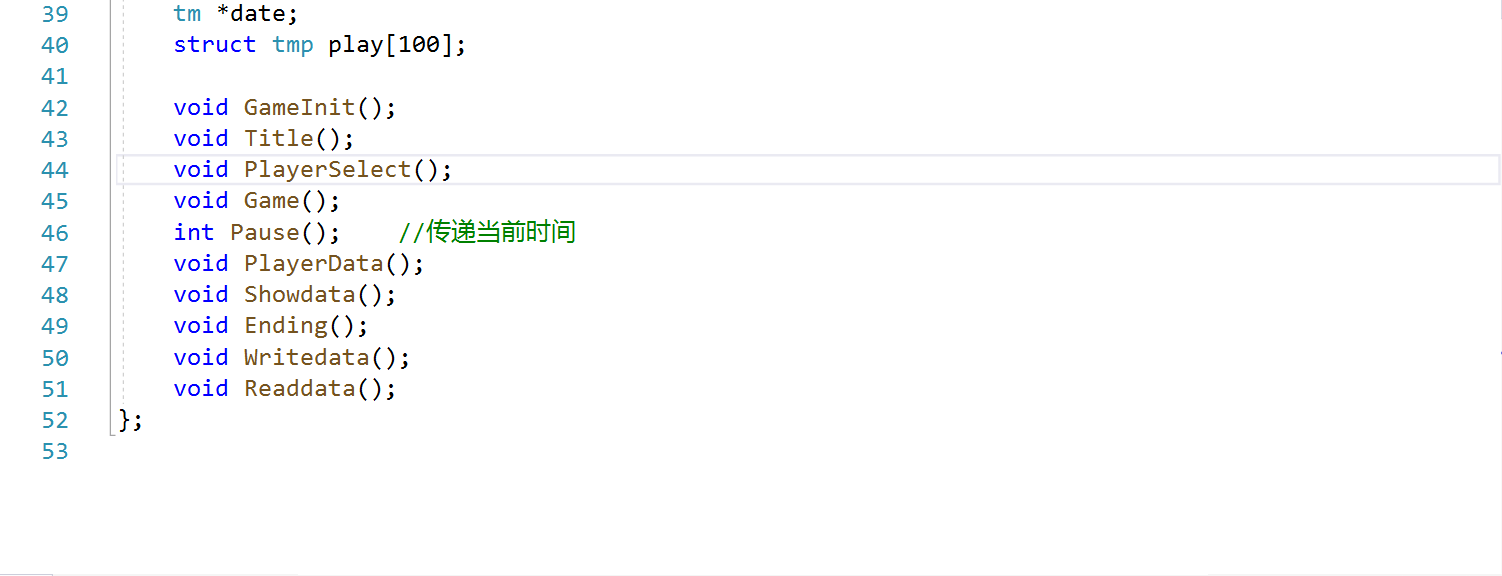
5.由于时间原因，其中播放replay功能（使用文件存储每一帧的玩家操作和随机种子）及输入玩家昵称没有完成（文本编码方式暂时存在问题）。

1. **实施（Implementation）**

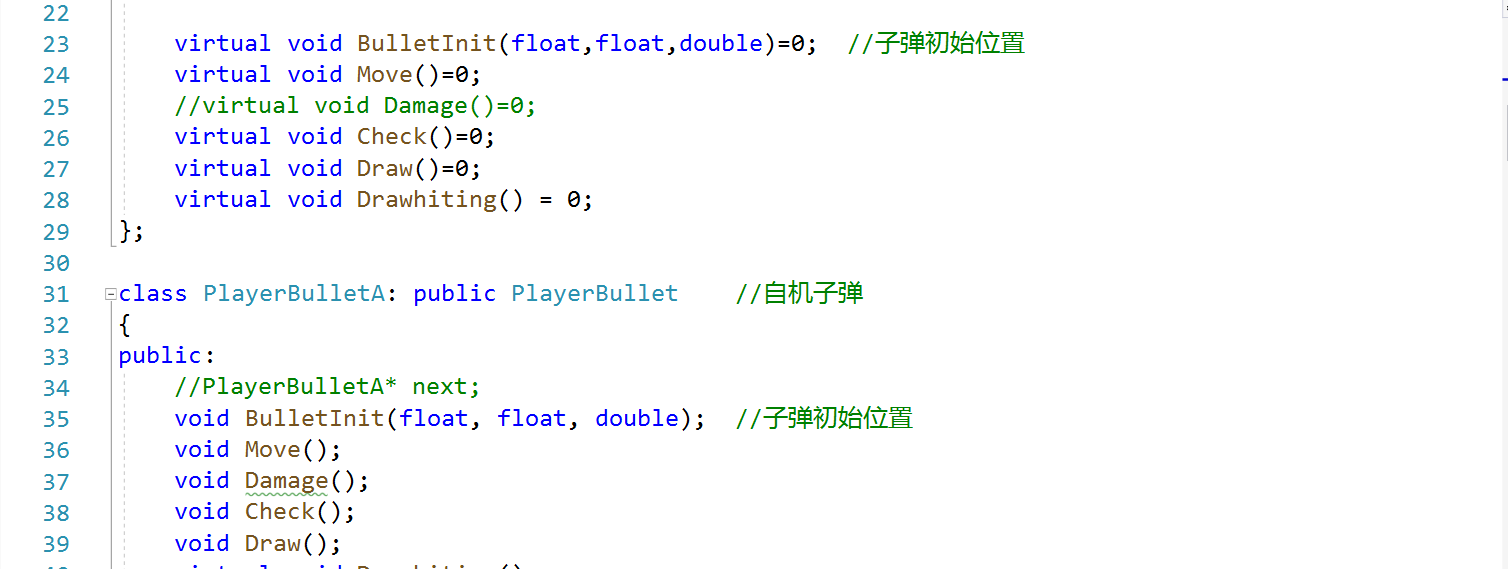
**程序入口**

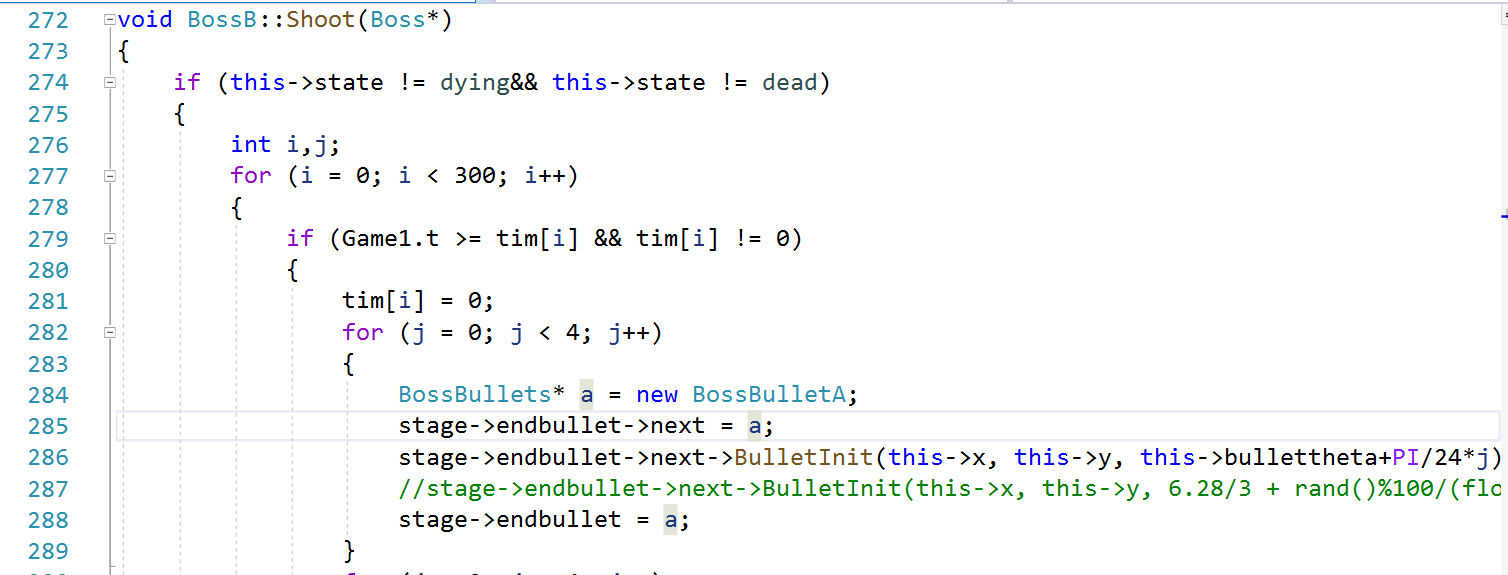


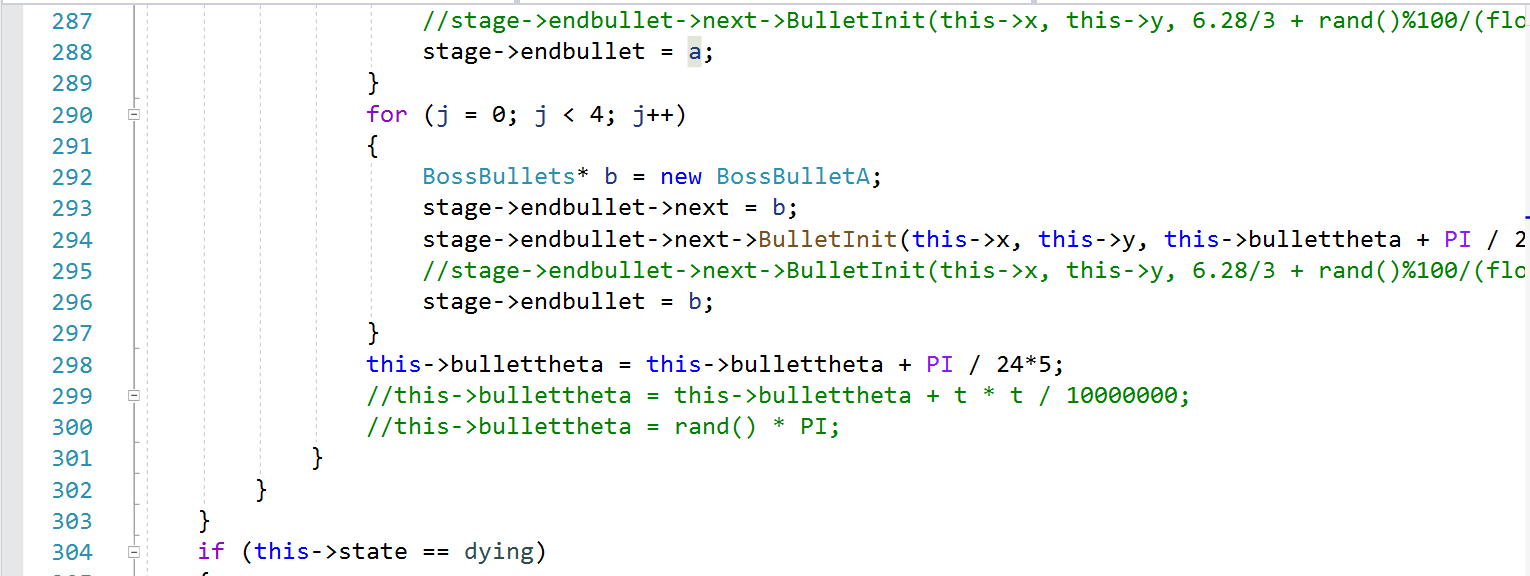
**游戏类**

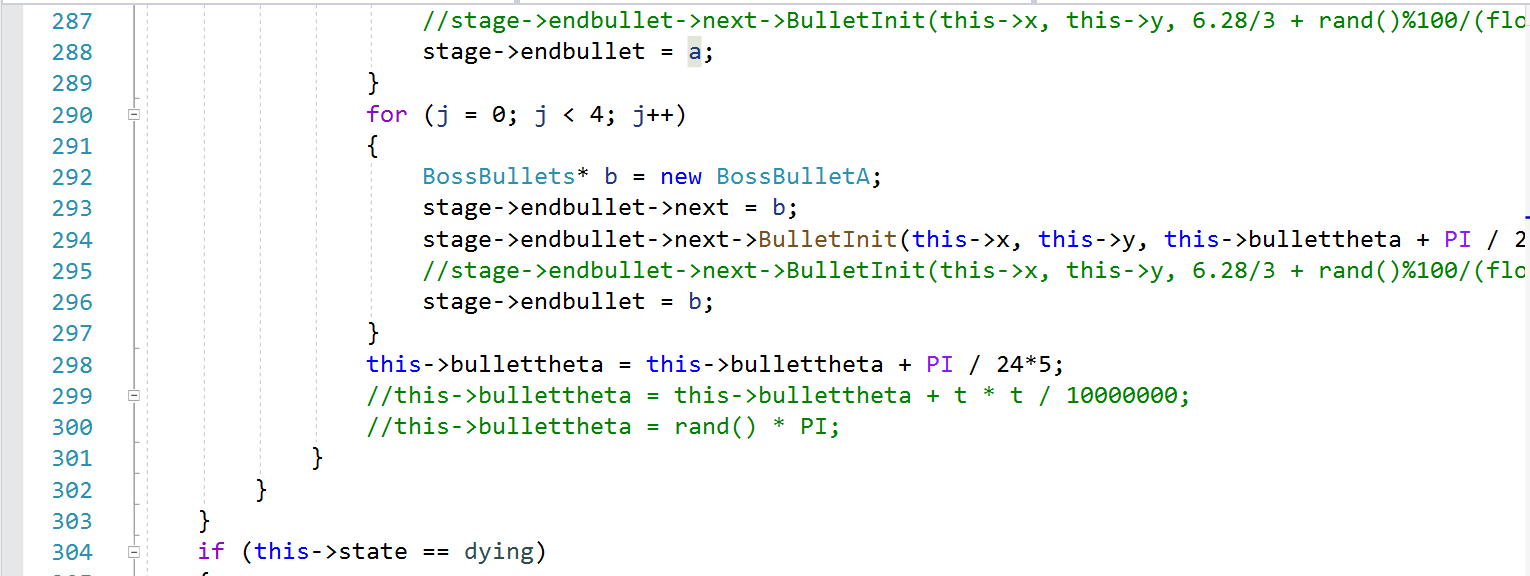


**子弹类举例**

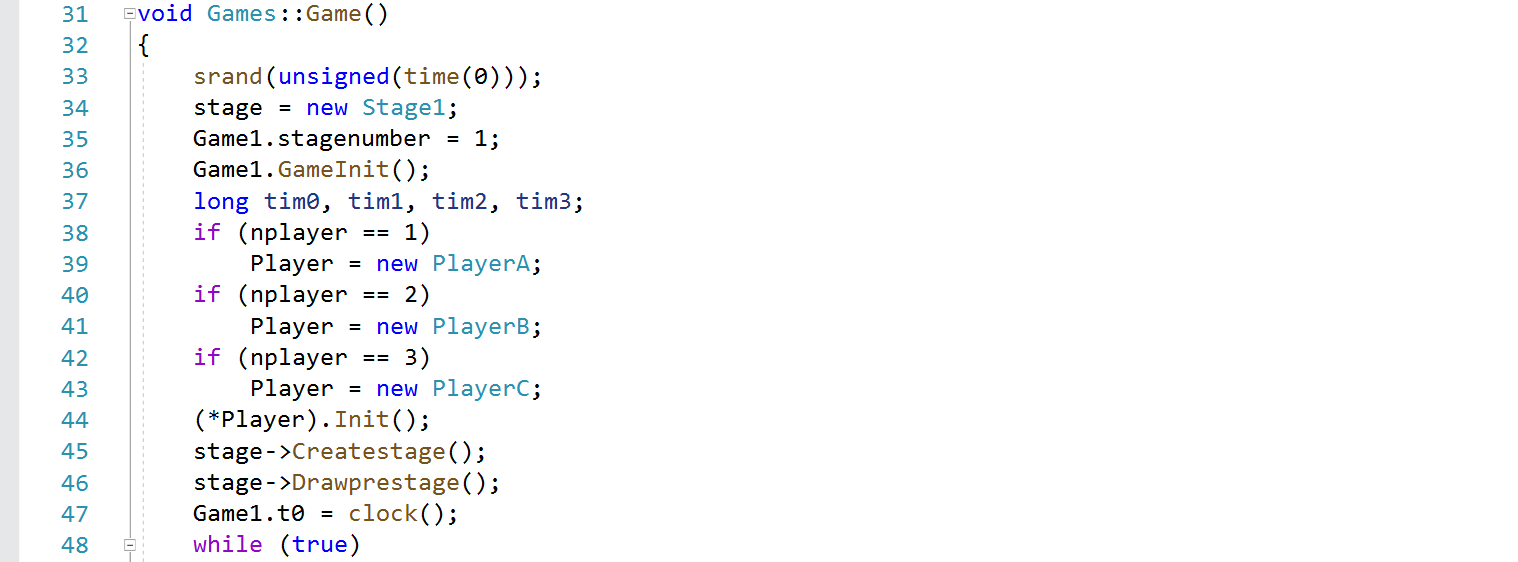
 

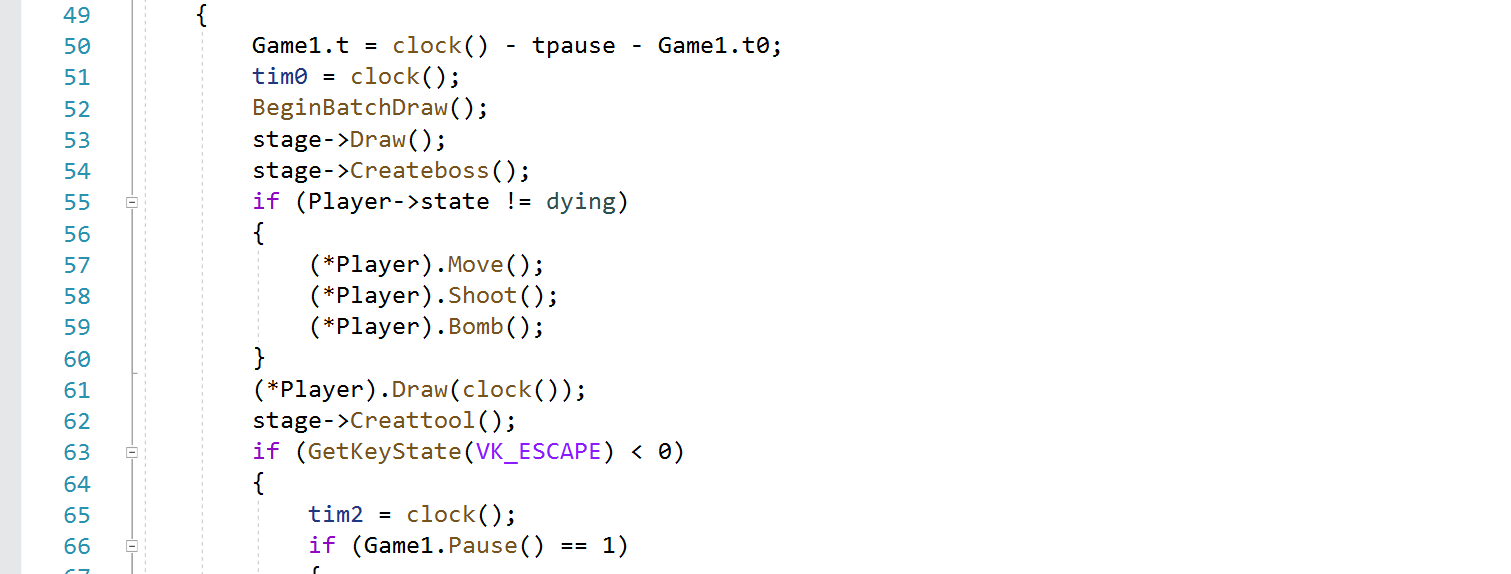
**Boss函数举例**

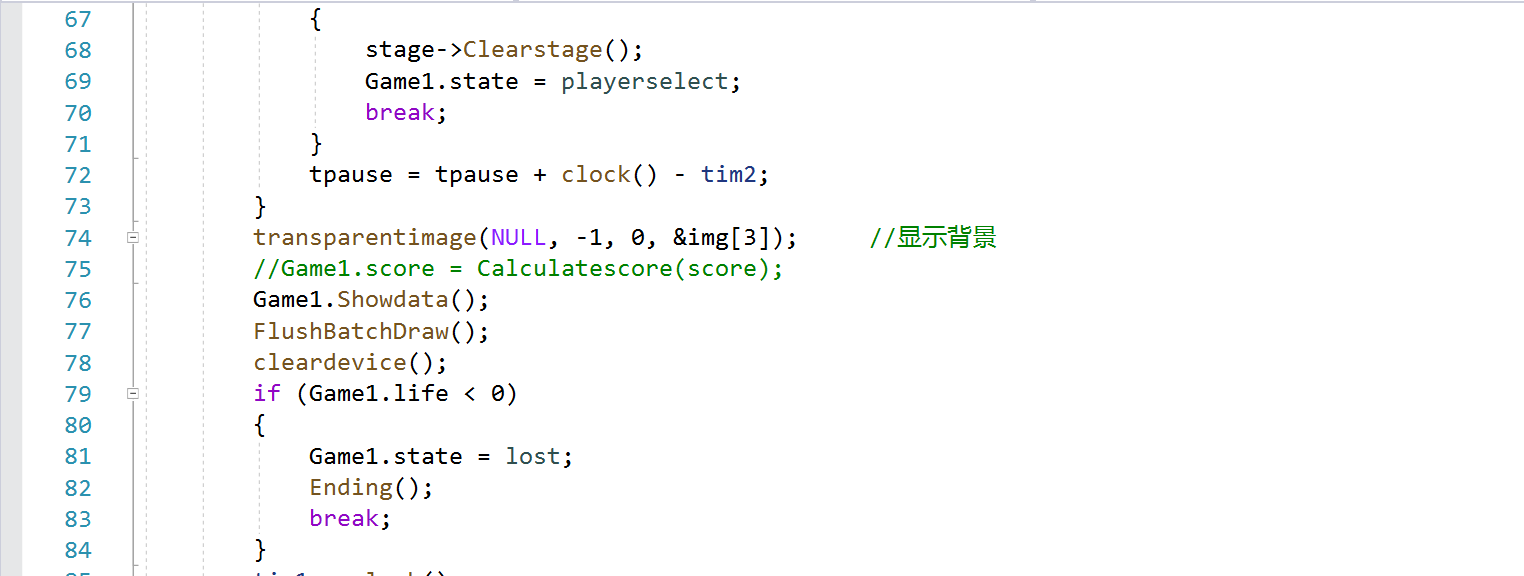
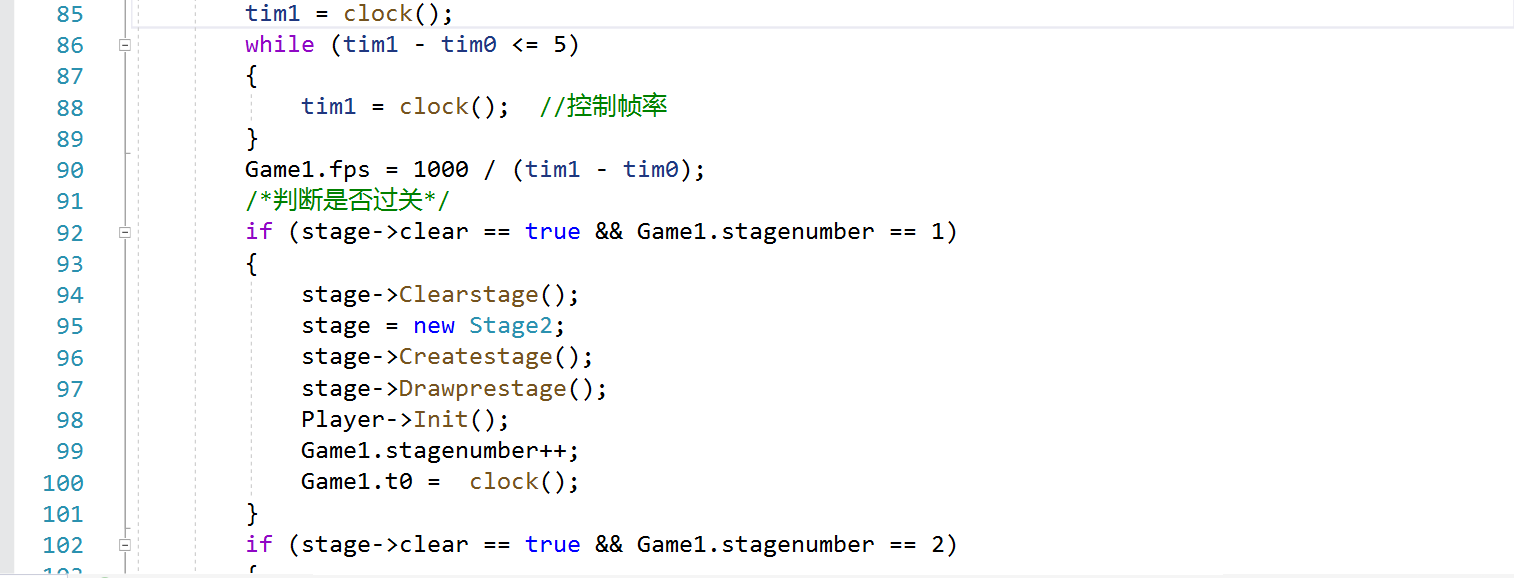


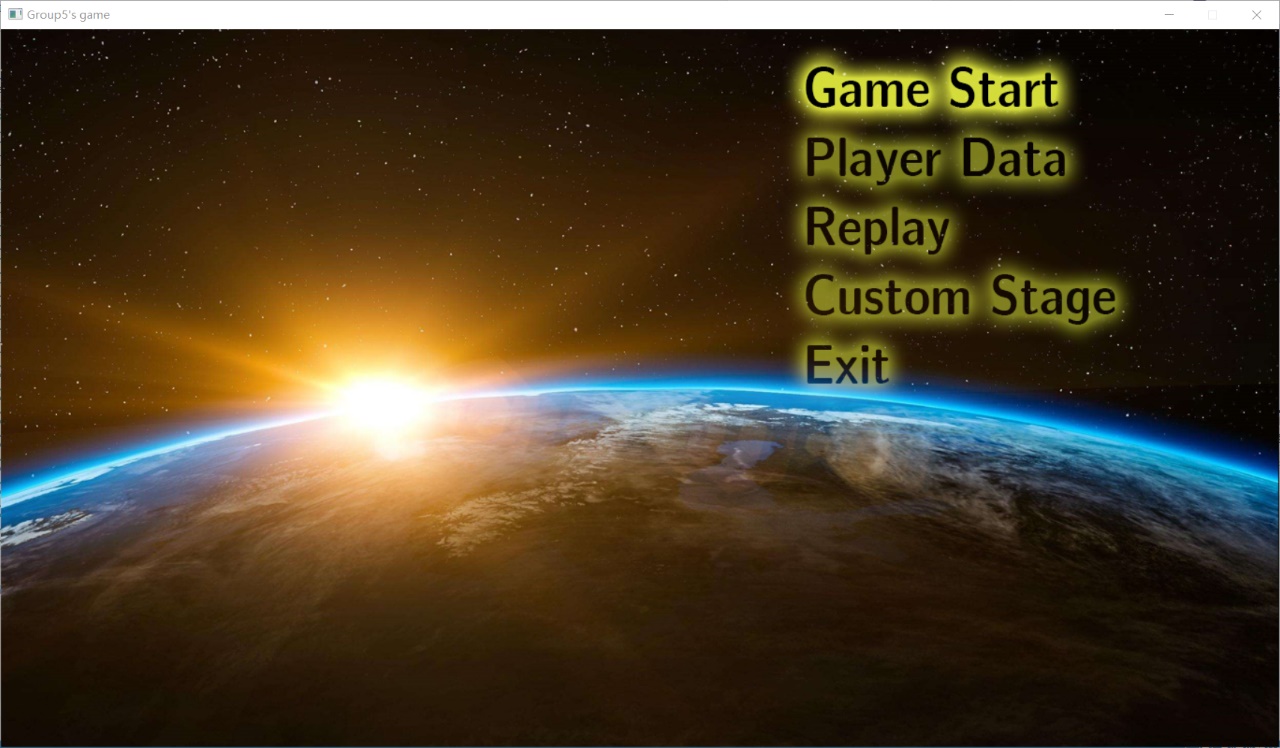
**Game主函数**



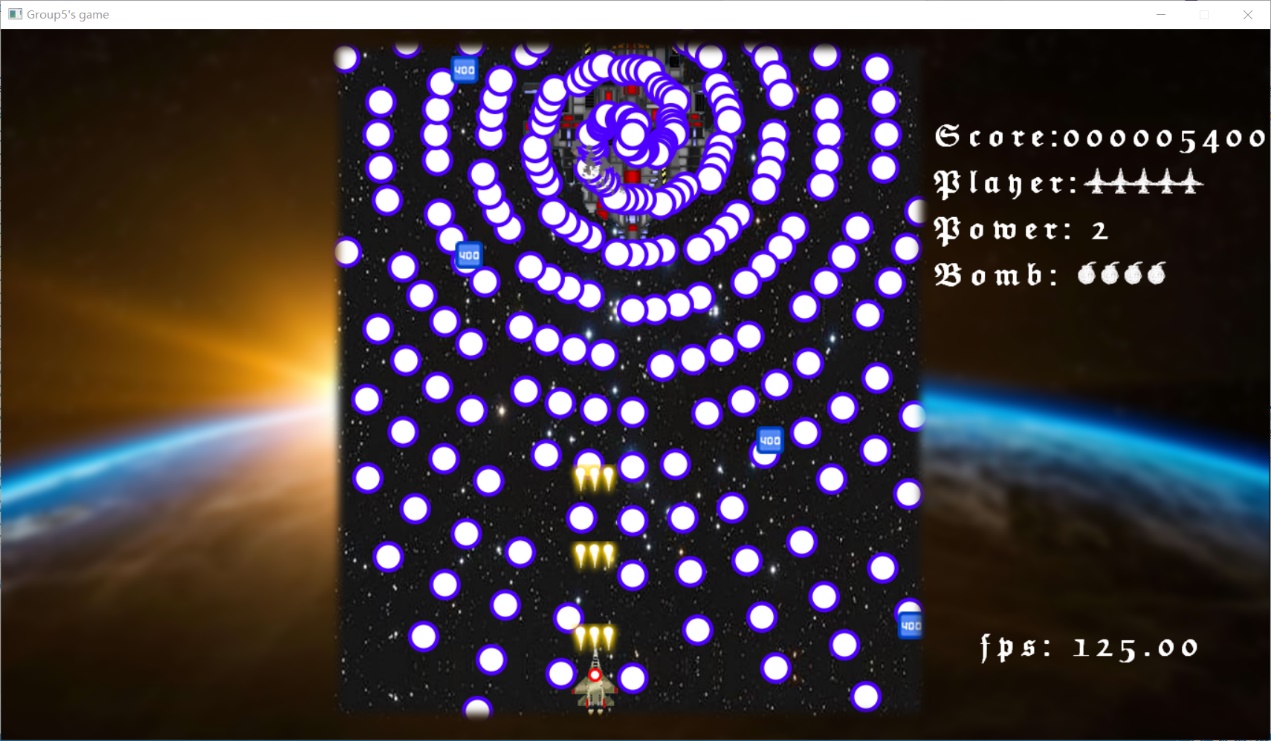


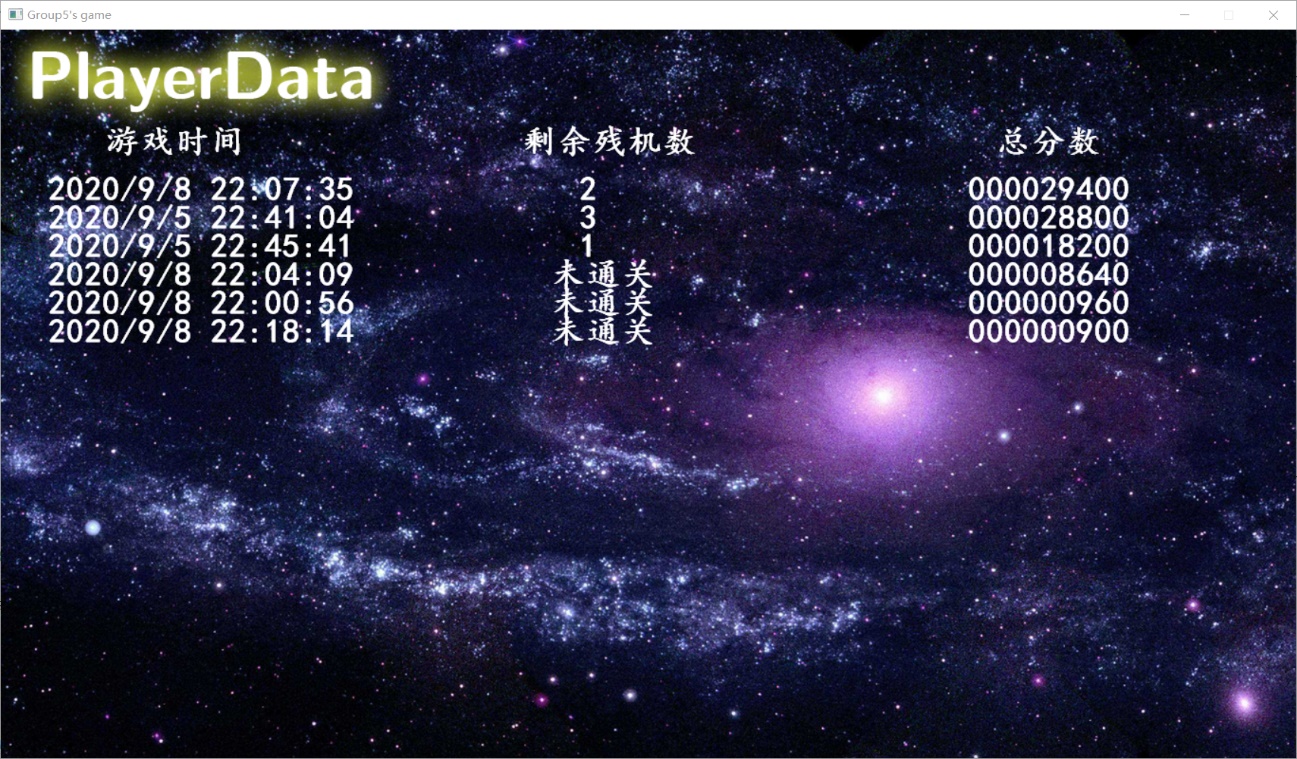
1. **测试（Test）**











1. **总结**

**由于时间原因，有部分计划中的功能未能实现，程序也还存在一些未修复的Bug，但总体来说已经是一个完整的、可以顺利游玩的游戏了。而且这是我们第一次尝试编写规模较大的C（C++）程序，C++和Easyx的知识都是边做边学的，因此代码的结构有很多不合理、有待改进的地方，而且一部分功能的实现十分笨拙。但这些问题使我们积累起了许多宝贵的经验，当以后编写其他规模可能更大的程序时，思路将会清晰得多，程序结构也将更加有条理。**