课程设计二报告

一．设计内容

设计一个基于图形界面的空战游戏，使用 qt 的 GUI 框架

下图为最终的游戏过程截图

二．设计目标（功能介绍）

（1） 加载界面
（2） 主窗口可伸缩
（3） 提供菜单界面
（4） 背景音乐（可切换）
（5） 游戏指南介绍游戏
（6） 显示我方血量、蓝量、分数
（7） 可移动我方飞机
（8） 自动打出子弹
（9） 按空格发动技能并消耗蓝
（10） 多种敌人类型
（11） 击杀特殊敌人掉落补给
（12） 可暂停游戏
（13） 暂停后弹出选项：可返回菜单、查看指南或继续游戏
（14） 子弹打中或飞机碰撞造成受到伤害
（15） 血量为 0 的物体将消失
（16） 我方血量为 0 游戏自动结束
（17） 游戏结束后弹出选项：重新游戏、查看指南、返回菜单

三．设计思路

（一） 类的设计

1. 视角类 MyView

（1） 功能：容纳场景的窗口，在该类中重定义 resizeEvent 事件处理函数可以完成窗口和场景的自适应伸缩

（2） 成员数据：无

（3） 成员函数：resizeEvent

（4） 实例化：main 中实例化为 view

2.场景类 Scene

（1）功能：提供图形化界面，一切视觉与玩家交互的控件都在场景类中进行控制。场景类也是和玩家交互的顶层接口，玩家发出的信号、或者触发的事件，都在场景类中进行第一层处理

（2）成员数据：文本显示、按钮显示、遮罩面板、飞机状态显示、图像路径、媒体音乐播放器、菜单显示、物体显示

（3）成员函数：构造函数、按钮信号的槽函数、事件触发后的处理重载函数

（4）实例化：Scene.cpp: Scene* MyScene；main 中申请空间

3.游戏运行类 ProcessGame

（1）功能：控制游戏的逻辑，对游戏的物体对象进行管理、控制游戏的运行状态、与场景类进行交互（即场景类的需求通过游戏运行类来逻辑实现，游戏运行类逻辑实现的结果通过场景类来进行体现）、按键响应

（2）成员数据：移动键按钮状态（以处理移动按键响应）、定时器（控制物体运动）、分数、游戏阶段（是否暂停、是否处于菜单界面）

（3）成员函数：构造函数、定时器的删除和设置函数、清除游戏现场函数、Scene 中按钮信号处理的槽函数的逻辑实现（图形界面的接口）、Scene 事件响应函数的逻辑实现（图形界面的接口）、玩家飞机状态更新函数

（4）实例化：ProcessGame.cpp: ProcessGame game;

4.物体类 Object

（1）功能：作为游戏中物体的顶层抽象，可以标记物体的类型和碰撞状态

（2）成员数据：type：包括玩家飞机、玩家子弹、普通敌人、红色敌人、蓝色敌人、普通敌人子弹、红色敌人子弹、红色补给、蓝色补给

（3）成员函数：构造函数、通过类型和图像路径初始化 object

（4）实例化：在需要的位置实例化其派生类

5.飞机类 Plane

（1）功能：由 object 派生，处理飞机的碰撞和移动、记录飞机状态

（2）成员数据：生命值和蓝量

（3）成员函数：处理碰撞和处理移动的函数

（4）实例化：在需要的位置实例化其派生类

6.子弹类 Bullet

（1）功能：由 object 派生，处理子弹产生和移动

（2）成员数据：子弹的方向、子弹的杀伤力

（3）成员函数：子弹发射函数、子弹整体更新位置函数、某个子弹移动函数

（4）实例化：在静态成员函数子弹发射中实例化具体对象

7.补给类 Medicine

（1）功能：由 object 派生，代表物体为补给

（2）成员数据：继承 object 的 type 标记补给的类型

（3）成员函数：通过补给类型和图像进行构造

（4）实例化：在掉落补给的位置实例化具体对象

8.我方飞机类 MyPlane

（1）功能：由 Plane 派生，处理我方飞机的移动按键响应、射击模式

（2）成员数据：长宽（防止我方飞机出界），射击模式、移动趋势、伤害

（3）成员函数：是否静止、设置移动趋势的函数

（4）实例化：MyPlane.cpp: MyPlane* mycraft,游戏开始的时候申请空间，返回菜单释放空间

9.敌方飞机类 EnemyPlane

（1）功能：由 Plane 派生，处理敌方飞机的整体位置更新、具体飞机的移动、和射击

（2）成员数据：继承 object 的类型判断其是哪种敌方飞机

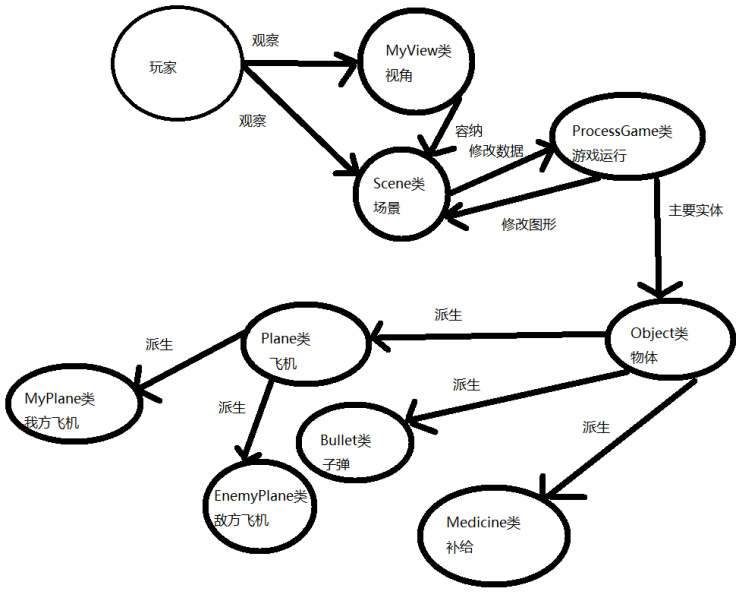
（3）成员函数：射击函数、整体位置更新函数、刷新产生函数、单个飞机移动函数

（4）实例化：在飞机刷新定时器事件中创建具体的对象

10.Config.h 配置了一些宏

（二） 类层次的关系

层次关系如下图



四．逻辑流程

1. 在 main.c 中初始化视角和场景类，然后应用进行循环 exec 状态（等待事件）
2. 刚开始只有按钮和键盘事件会被响应，通过按下按钮或键盘触发 Scene 中的事件响应函数或者槽函数
3. Scene 接收到信号或事件后将要处理的事情丢给 ProcessGame 类来处理
4. 在 ProcessGame 中通过具体的信号来处理内部的数据，Object 和被派生出的飞机、子弹、补给、键盘响应、定时器响应都是在此类中处理

五．课程设计一和二的关联和改善

主要的逻辑控制都按照设计一的思路，除了把顶层加了 ui 以外，内部也有一些改进

比如数据结构的改进（STL 的掌握）：子弹、敌方飞机从数组管理改成 vector 管理

流程上沿用：菜单<->游戏的结构

比如利用 qt 的定时器库让物体运动机制更完善，在设计一中提到，基于冷却的原理，即一个功能使用后再被申请使用的时候，只记录下申请，等待冷却结束再满足申请，在二中通过了解 qt 定时器的使用知道了这是基于定时器的物体运动机制

类的层次更加鲜明（继承的运用）：把许多物体抽象成 Object，统一的管理

增加了许多新的功能：例如音乐、例如技能、例如血槽、例如补给、例如多种敌人

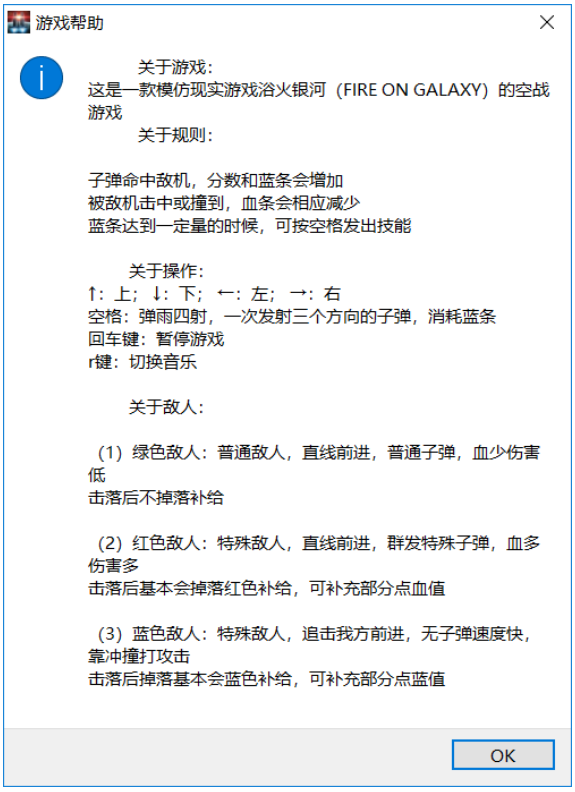
六．程序的操作方法

1.刚进入游戏看到加载界面

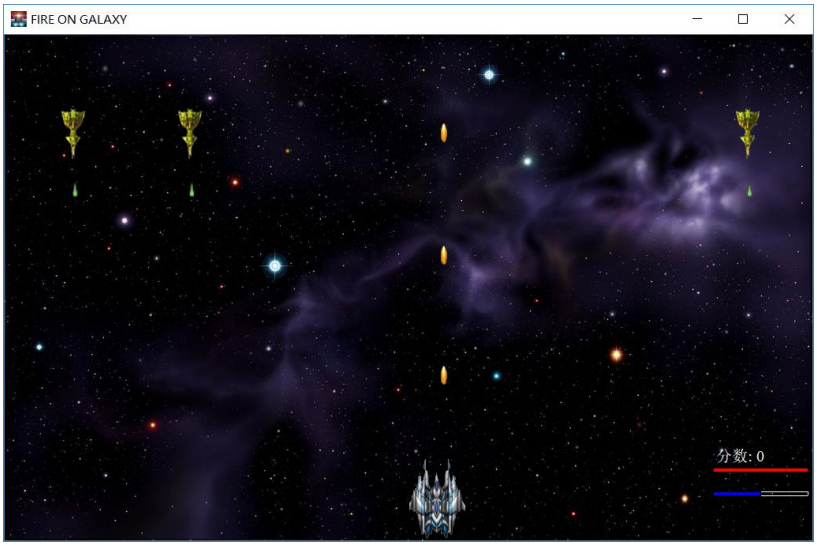
2.然后看到游戏窗口

3.听到背景音乐，可以按 R 键切换

4.点游戏指南查看游戏的介绍信息

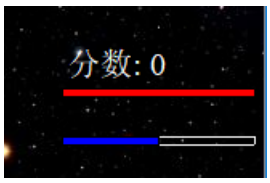

游戏帮助
关于游戏：
这是一款模仿现实游戏浴火银河（FIRE ON GALAXY）的空战游戏
关于规则：
子弹命中敌机，分数和蓝条会增加
被敌机击中或撞到，血条会相应减少
蓝条达到一定量的时候，可按空格发出技能
关于操作：
↑：上；↓：下；←：左；→：右
空格：弹雨四射，一次发射三个方向的子弹，消耗蓝条
回车键：暂停游戏
r键：切换音乐
关于敌人：
（1）绿色敌人：普通敌人，直线前进，普通子弹，血少伤害低
击落后不掉落补给
（2）红色敌人：特殊敌人，直线前进，群发特殊子弹，血多伤害多
击落后基本会掉落红色补给，可补充部分点血值
（3）蓝色敌人：特殊敌人，追击我方前进，无子弹速度快，靠冲撞打攻击
击落后掉落基本会蓝色补给，可补充部分点蓝值
OK

5. 点开始游戏进行游戏


FIRE ON GALAXY
分数: 0

6. 随时按回车键暂停

7. 按上下左右移动，空格发技能
8. 右下角查看当前状态



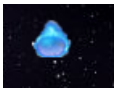
9. 绿色敌人速度慢、单子弹攻击，伤害弱、防御弱

10. 红色敌人速度慢、多子弹攻击，伤害强、防御强

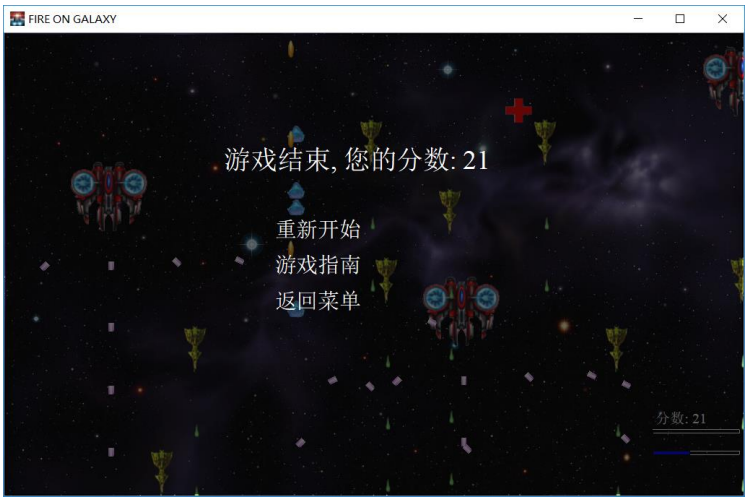
11. 蓝色敌人速度快、撞击攻击，伤害强、防御一般

12. 击杀红色敌人掉落红色补给，补充血量

13. 击杀蓝色敌人掉落蓝色补给，补充蓝量



14. 玩家死亡后弹出选项和分数



七．遇到的问题和解决方案

(1) 没有 QT 基础，为了入门 QT，通过在 bilibli 上找视频和 QT 学习之路文档快速学习

(2) 中途遇到不了解的功能再来搜索，如如何做到窗口伸缩自适应、如何播放音乐

(3) 关于键盘响应失去了底层的 vc 编译器库，但通过查阅到有键盘响应事件的重载函数

(4) 图片的截取，一开始不知道怎么把飞机的图片中白色背景去掉，后来询问同学临时学习了 ps 的基础知识

(5) 不了解如何生成图标和打包 qt，直接搜索 qt 图标 打包 发布几个关键词即可找到解决方法