

**软件学院大作业任务书**

题 目： 简易飞机大战

专 业： 软件工程

班 级： 软件工程1910班

姓 名： 赵英剑，严之寒，胡亚冰

学 号：8002119321，8002119311，8002119309

完成人数： 3人

起讫日期： 2020. 5.1 — 2020. 6.1

任课教师： 张云如

部分管主任： 夏灵林

完成时间： 2020年5月25日

**说 明**

1. 本任务书由任课教师填写后，下达到学生。
2. 任务完成后，任课教师需填写小结表。
3. 任务书、学生成绩与学生完成后的大作业（纸质和电子两种）一并报送各教学研究部审核后转教务办。
4. 另附一份全班大作业总结

|  |
| --- |
| **大作业的要求和内容：（包括题目选择范围、技术要求、递交时间、考核方法等）**  请根据本学期所学的C++语言知识，采用小团队合作方式（团队成员最多不超过3名）或者个人独立完成方式，完成一个完整可运行且具有一定实用功能的C++程序，具体程序功能可以根据需求进行自定义，但不能过于简单，不得抄袭他人代码或从网上下载代码直接充当大作业，且所设计的程序必须满足以下要求：   1. 要求应用**面向对象设计的思想，**程序中必须体现面向对象的设计； 2. 程序中要具备**有封装、继承、多态等知识点的体现**； 3. 程序不少于200行代码； 4. 程序应遵循C++编程规范和质量标准，系统实现代码规范，注释清晰；大作业文档规范，分析设计过程以及总结阐述简洁明了。； 5. 要求程序的启动界面中必须显示程序名称、版本信息，版权说明等； 6. 要求代码书写规范（注意必要的缩进、空行、空格的应用），同时代码中要有必要的注释，对应函数要有功能描述、入口及出口参数说明。   最终提交的作业，除提交完整的程序工程代码外，还必须以大作业报告的形式阐述整个程序的实现过程，要求报告内容中必须包括：   * **小组分工（团队合作完成的必须有该内容，独立完成的可以省略）**（首先要阐述小组分工的依据，给出为什么这样分工的理由；然后，详细说明小组中每一位组员承担的具体任务，要说明谁是**组长**，同时每位组员要附加**正面照片**）； * **需求分析**（项目介绍、功能需求）； * **系统分析与设计**（阐述项目中需解决的关键技术问题，同时要以程序流程图、函数关系描述等）； * **程序设计与实现**（其中内容不能只粘贴代码，首先要描述代码工程的文件组织结构，然后以文字的方式阐述代码中主要函数的设计意图、主要功能、运用到的关键知识点，对于代码实现中有特色的算法或有创新的地方，可以着重描述（这将视为大作业的加分点）。如果必须给出实现代码才能更好地说明问题时，也必须先有相关的文字叙述，然后才是代码，代码只是作为例证。）； * **小组讨论日志（团队合作完成的必须有该内容，独立完成的可以省略）**（以时间顺序记录讨论的时间、地点、讨论的内容，并附加讨论时拍摄的照片）； * **组员或个人小结**（如果是团队合作的形式，该部分必须包含小组中每一位组员自己写的个人开发小结，其中必须谈到每个人在项目中具体负责的工作、所起的作用、贡献程度、工作量、开发过程中遇到的困难以及如果克服困难、个人收获、得到的启示或教训等等；如果是独立完成的，则必须包含工作量的描述、开发过程中遇到的困难以及如果克服困难、个人收获、得到的启示或教训等等），切忌空洞无实际内容或前篇一律的敷衍文字； * **参考文献**（该部分给出整个程序从选题、需求分析、系统设计到代码实现过程中所参考的书籍、网上资料等。）。   大作业的评分点涵盖大作业从选题、需求分析、代码实现到文档撰写全过程。具体评分点及各评分点的比重如下：   * 选题 15%   评分依据：选题的难度、创新度、工作量等   * 需求分析 15%   评分依据：分析是否充分、表述是否明确、功能的实用价值等   * 文档撰写质量 30%   评分依据：结构完整性、内容充实度、格式符合度、图表规范程度等   * 代码质量 40%   评分依据：代码复杂度、功能完整性、是否运用了要求的知识点、设计或算法是否有创新等 |
| 教师小结：  成绩：  教 师 签 名：  教研部负责人：  学生姓名： |



**NANCHANG UNIVERSITY**

**C++程序设计课程设计**



**题 目:** 简易飞机大战

**学 院：** 软件学院

**专 业：** 软件工程

**班 级：** 软件工程1910班

**完成人数：** 3人

**成 员：** 赵英剑，严之寒，胡亚冰

**起讫日期：** 2020. 5.1 — 2020. 6.1

**任课教师：** 张云如

**完成时间：** 2020年5月25日

**填表日期：** 2020 年 6 月 6 日

**目 录**

一、 项目小组分工（独立完成的可以省略该内容） 1

二、 项目需求分析 1

1.1 项目介绍 1

1.2 功能需求 1

三、 系统分析与设计 1

3.1 本程序需解决的关键技术问题 1

3.2 程序流程 2

3.3 函数关系描述 2

四、 设计与实现 2

3.1 工程文件组织结构 2

3.2 程序设计 2

3.3 程序实现 2

五、 小组讨论日志（独立完成的可以省略该内容） 3

六、 个人小结 3

七、 参考文献 3

# 项目小组分工（独立完成的可以省略该内容）

1. 组长：**赵英剑**

学号：8002119321.

专业：软件工程. 班级：软件工程1910班.

**主要负责**：程序主体、算法编写，网页编写。

1. 组员：**严之寒**

学号：8002119311.

专业：软件工程. 班级：软件工程1910班.

**主要负责**：程序UI设计，项目资源搜集。

1. 组员：**胡亚冰**

学号：8002119309.

专业：软件工程. 班级：软件工程1910班.

**主要负责**：辅助UI设计，辅助资源提供。

# 项目需求分析

## **项目介绍**

**项目简介：**

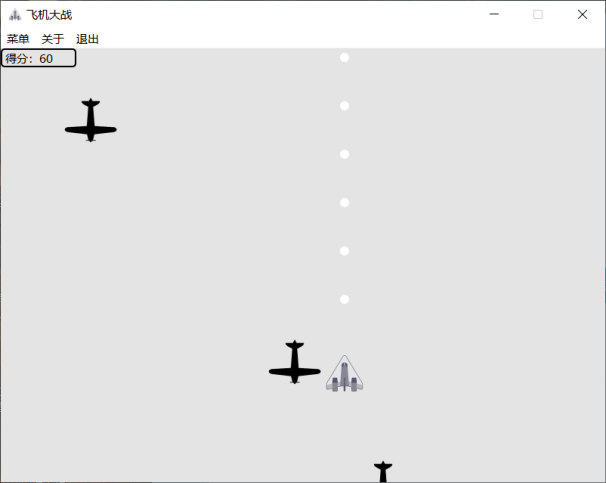
项目名称为简易飞机大战，使用者可以操控一架小飞机与敌机战斗很经典的打飞机小游戏，简约而不简单 （此版本为无敌版） 游戏目标：合理操作控制飞机消灭更多的敌人，获得更高分数。

**项目发展：**

本项目初期打算设计一个简单但有趣的小游戏，于是选择了飞机大战这个小游戏，于项目发展中期建立了一套适用于本项目的解决方案，并由此为依据，在项目的后期由本项目的负责团队用时一周开发完成。终期通过测试后，宣布完工。

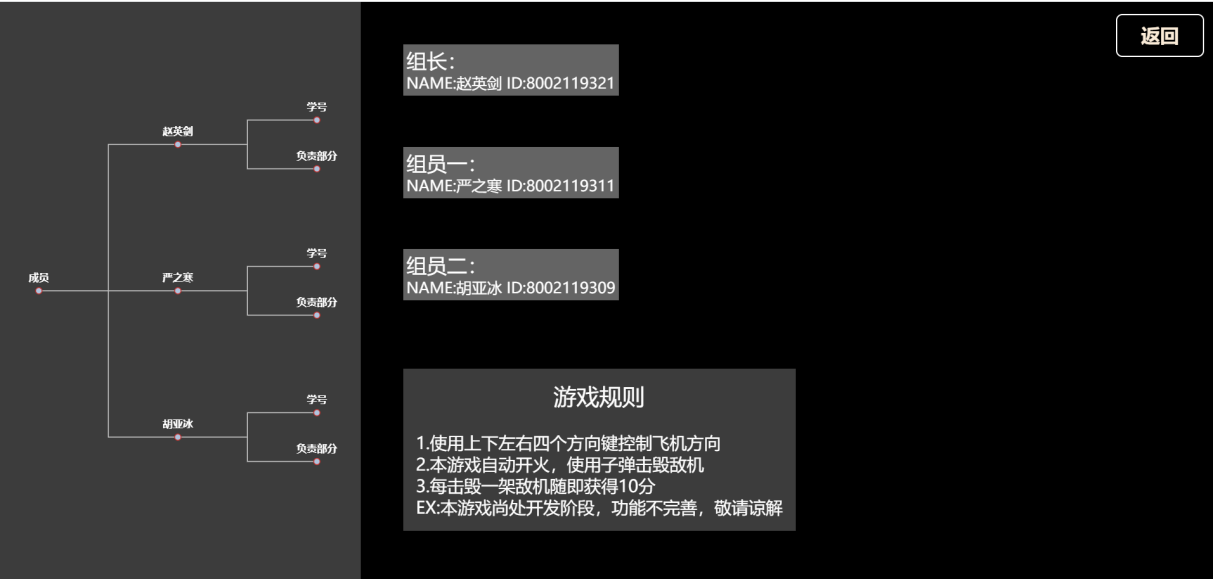
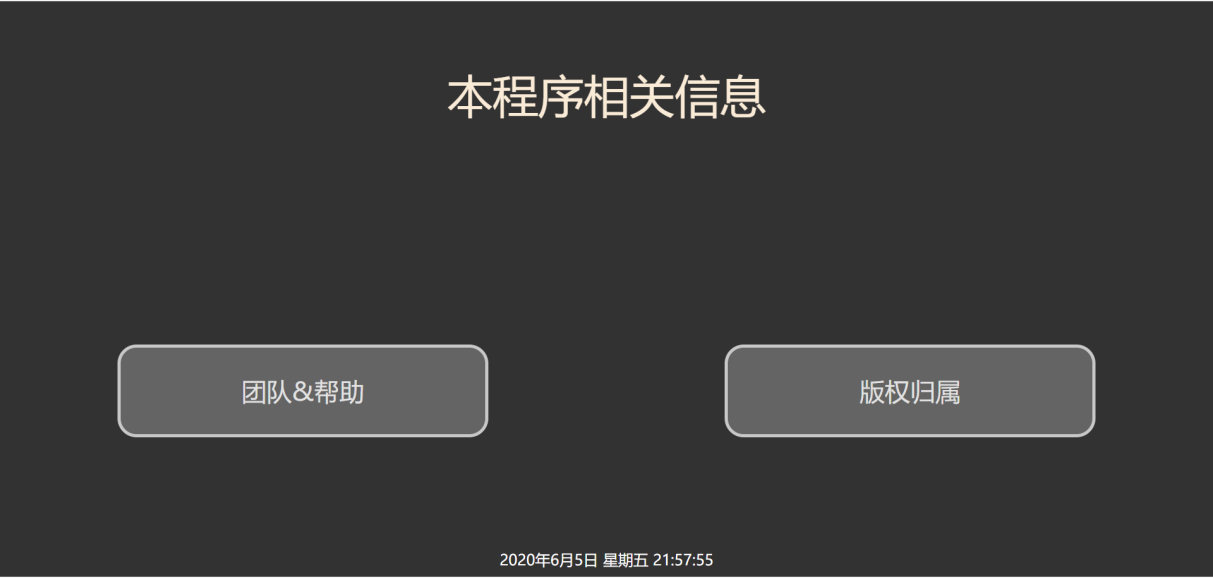
**项目基本展示：**

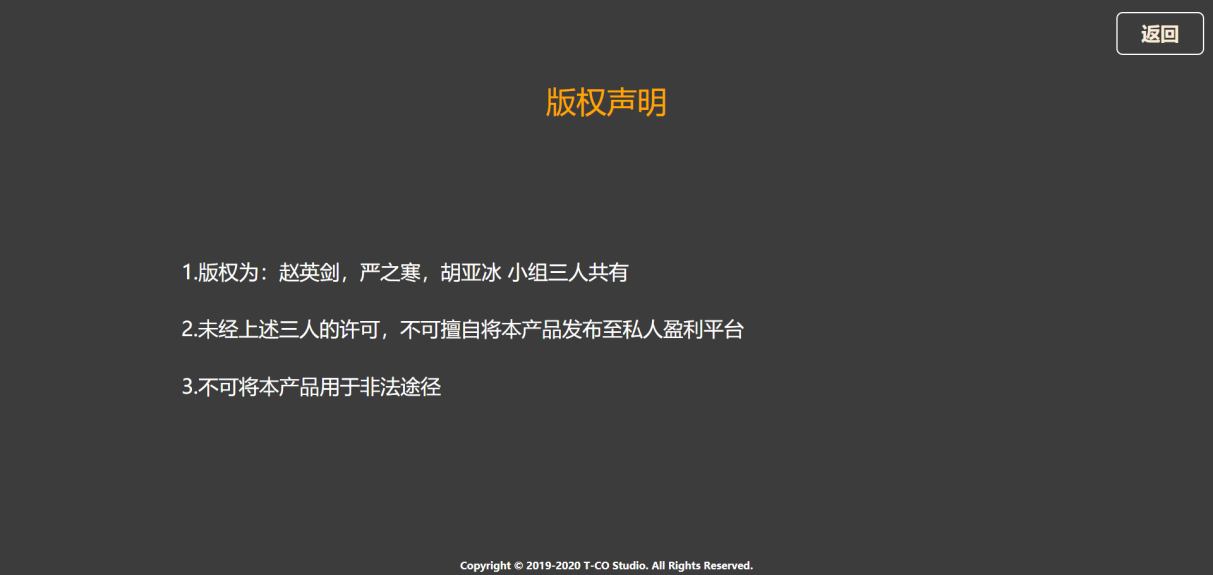
**1.首界面 2.游戏界面**



**3.网页展示**

网页初始界面

成员界面

版权界面

**本程序的静态网页链接：***[点击此处](https://virtual-memes.gitee.io/about)*

**本程序的打包文件：***[点击此处](https://gitee.com/Virtual-Memes/About/releases/v1.0)*(下载Release压缩包文件解压，运行Plane.exe）

**项目源码：***[点击此处](https://github.com/Virtual-Memes/CPP_Final_Program)*

## **功能需求**

使用者可以流畅的操控飞机与敌机战斗，且每击落一架敌机就能加十分。有重新开始及退出功能，通过点击按钮选项即可查看与本程序相关的信息的展示用静态网页。

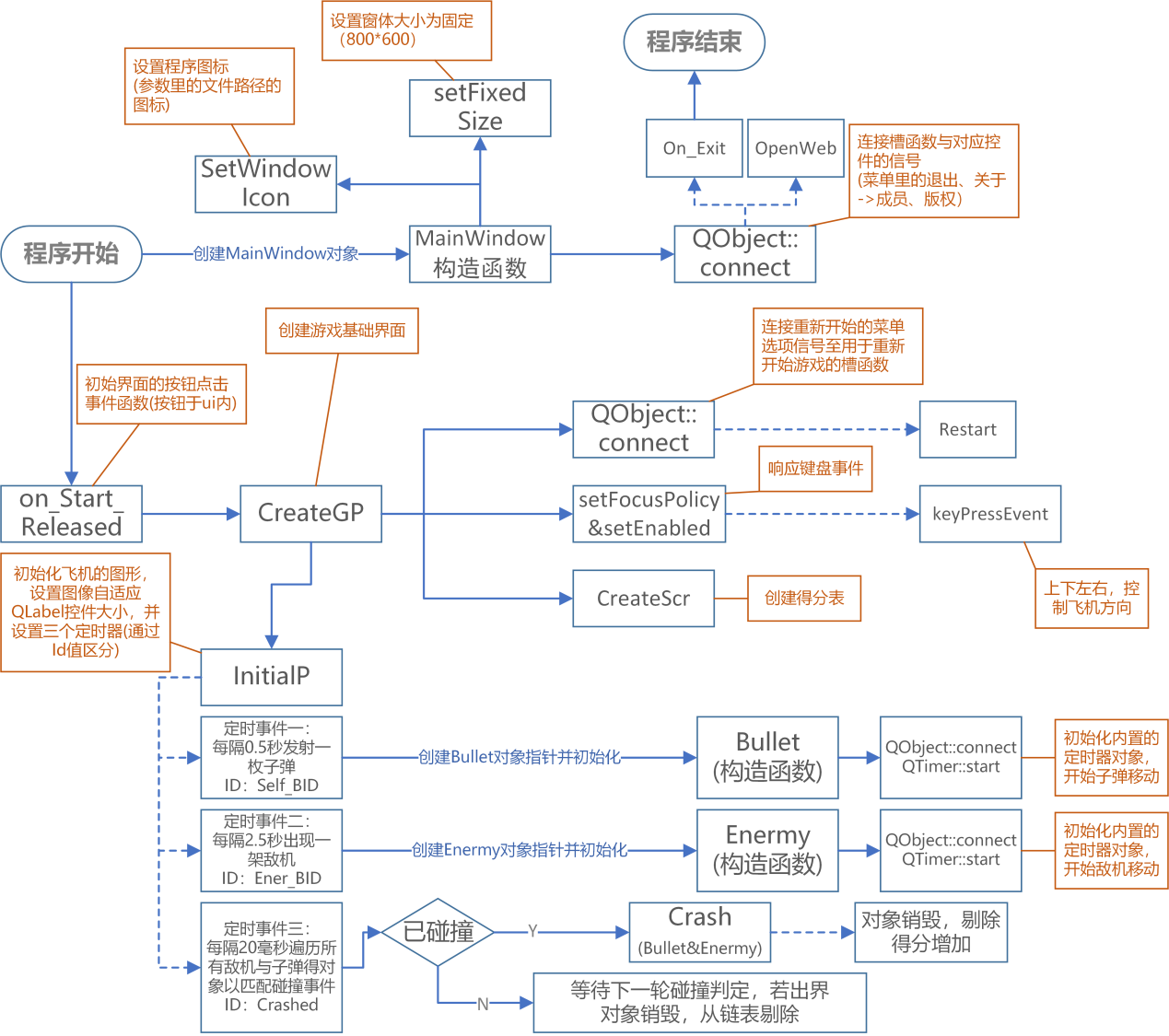
# 系统分析与设计

## **本程序需解决的关键技术问题**

在编写本次大作业项目时，所遇到的关键技术问题如下：

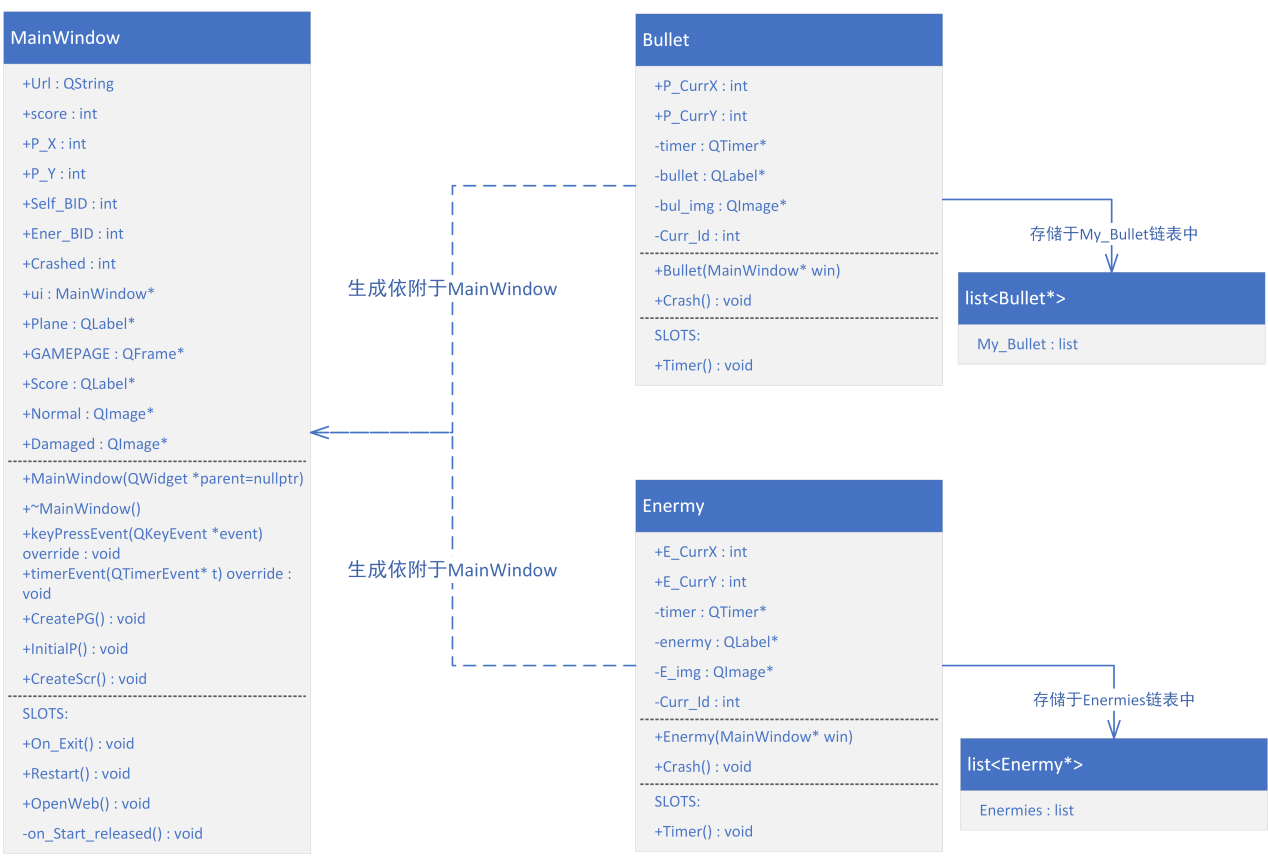
1. 在开始游戏时，如何才能正确地切换至游戏界面
2. 在于游戏界面如何才能让飞机响应键盘的事件并随之移动方向，且如何 移动，移动的速度又该如何设置等
3. 子弹的发射如何实现，以及子弹生成后又应如何移动，且子弹不同时刻 的状态又该如何设置
4. 敌机的发生如何实现，以及敌机地运动又该通过怎样地方式来实现，敌 机的不同状态又该如何实现
5. 敌机与子弹相碰撞该如何判定，碰撞后又该执行哪些步骤
6. 对于内存的管理

## **程序流程**



程序运行时各函数之间的关系正如上图所表示的，在开始程序时，即将菜单区的功能与与之对应的槽函数相连接(重新开始在游戏开始时进行连接)。开始游戏后，随即画面转变至游戏界面，在游戏过程中，每一个子弹/敌机都有属于自己的定时器，用于控制其移动以及出界后，销毁自己，并从对应链表中剔除，释放内存。而子弹与敌机的发生，都由另外的定时器控制，并且碰撞的判定也由另一个定时器判定，判定后则如上图中进行操作。

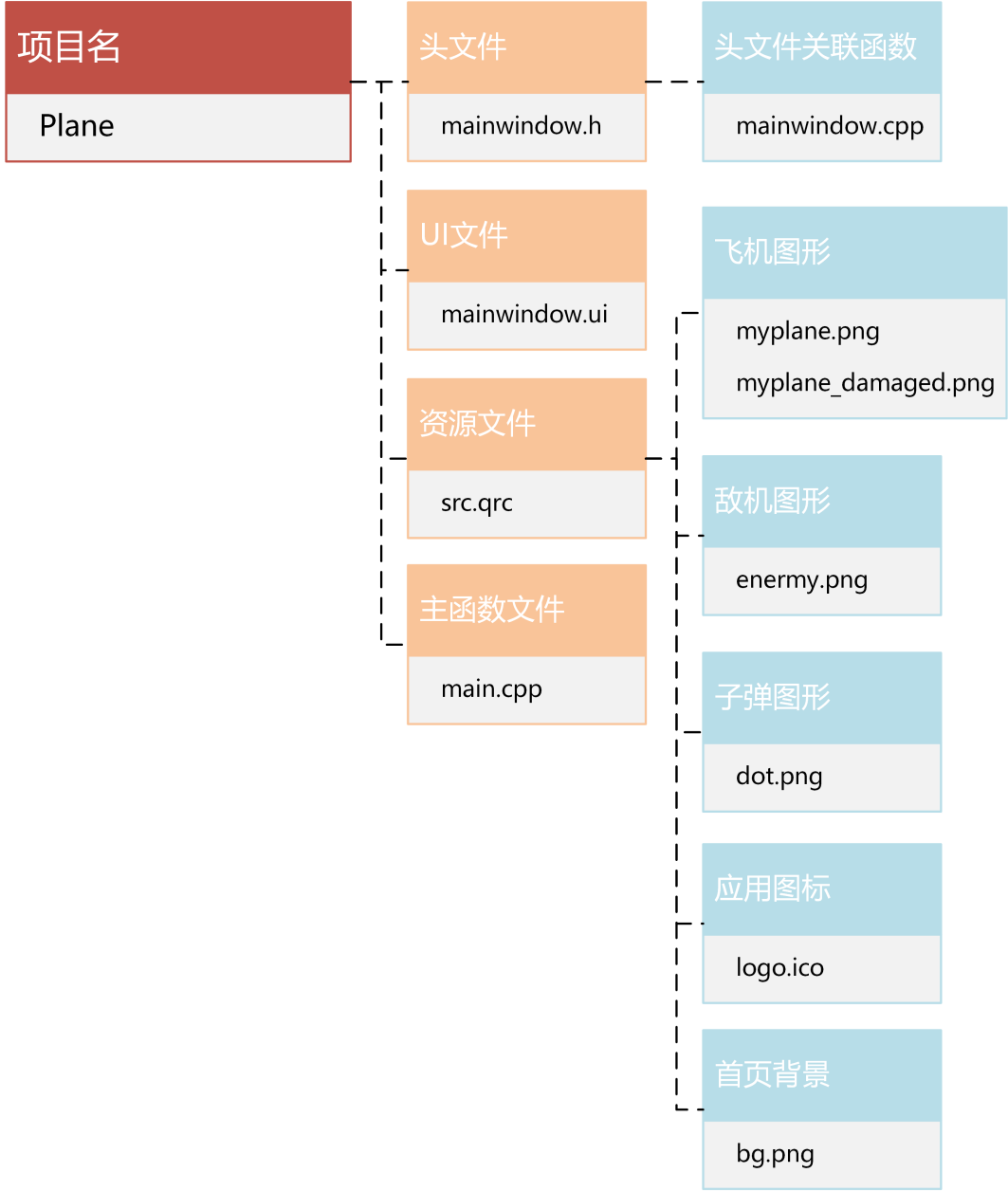
## **函数关系描述**



其运行时的函数与类的关系如上UML图与图中相应的文字描述，概括可述为：在程序开始运行时，调用主体窗口的构造函数，随即构建开始界面，开始游戏后，随即进入第二个界面，各类行为由相应的定时器函数或信号和槽函数进行控制。

# 设计与实现

## **工程文件组织结构**



本项目的文件结构如上图所示，由一个控制主体窗口包含所有类的mainwindow.h头文件和一个与之关联的mainwindow.cpp文件，再有整个程序开始的入口main.cpp文件与涵括资源文件的src.qrc文件，其中包含飞机图形，子弹图形，敌机图形及相关的图片文件。

## **程序设计**

1. MainWindow(类) 为整个程序主体窗口的构造类，通过这个类得以构造出这个程序的大体框架，以及操作的菜单，是这个图形界面得以产生的主要依赖。其中，通过在main函数中构造出这个类的对象就得以调起该类的构造函数MainWindow(构造函数)，从而通过构造函数的运行构建起整个程序框架。
2. on\_Start\_Released(成员槽函数) 响应首页的按钮点击事件，从而执行CreateGP函数，构造游戏界面。
3. CreateGP(成员函数) 创建游戏界面，初始化飞机，以及响应键盘事件。
4. InitialP(成员函数) 为飞机初始化控件，并加载飞机图片，并同时开启定时器事件(子弹发射，敌机发生，碰撞判定等)。
5. keyPressEvent(override) 为飞机的移动设置键盘事件的响应，通过键盘控制飞机移动。
6. timerEvent(override) 为子弹，敌机，碰撞判定设置定时器事件(以ID区分事件)。
7. CreateScr(成员函数) 创建得分表，记录得分。
8. Bullet(类) 定义所有的子弹为同一个类，其内定义了子弹的控件类型，子弹的图片以及每个子弹所分配的定时器。
9. Enermy(类) 定义所有的敌机为同一个类，其内定义了子弹的控件类型，子弹的图片以及每个敌机所分配的定时器。
10. My\_bullet&Enermies(list链表) 存储所有已生成的子弹对象，敌机对象，用于判定碰撞事件，若判定为已碰撞，则从中删除。
11. Timer(成员槽函数) 为每个子弹/敌机的定时器所分配的槽函数，用于控制子弹/敌机的移动，出界即自动销毁对应对象，从对应list中删除，释放内存。
12. Crash(成员函数) 若已碰撞则触发该函数，销毁对象，释放内存。
13. Restart(成员槽函数) 若点击重新开始则触发该函数，清空所有子弹，敌机，得分重置为零。
14. On\_Exit(成员槽函数) 退出程序。
15. OpenWeb(成员槽函数) 打开网页。

## **程序实现**

1. 界面的切换：

通过点击事件先销毁初始界面再调起CreateGP函数创建游戏界面，同时初始化飞机，设置响应点击事件。调起飞机初始化函数及的分表创建的函数。

1. /\*mainwindow.cpp\*/
2. //重新开始的槽函数
3. **void** MainWindow::on\_Start\_released(){
4. **delete** ui->Begin;
5. CreateGP();
6. }
7. //初始化背景窗口
8. **void** MainWindow::CreateGP(){
9. GAMEPAGE=**new** QFrame(ui->Center);
10. GAMEPAGE->setGeometry(0,0,800,600);
11. GAMEPAGE->setStyleSheet("QFrame{background-color: '#e4e4e4';};");
12. //连接重新开始按键至对应的槽函数
13. QObject::connect(ui->Replay,SIGNAL(triggered()),**this**,SLOT(Restart()));
14. //设置相应鼠标事件
15. setEnabled(**true**);
16. setFocusPolicy(Qt::StrongFocus);
17. GAMEPAGE->show();
18. //调用初始飞机函数
19. InitialP();
20. //调用初始得分表函数
21. CreateScr();
22. }
23. 飞机的移动功能的实现：
24. /\*mainwindow.h\*/
25. //重写keypressEvent设置键盘事件
26. **void** keyPressEvent(QKeyEvent \*event) override;

通过重写keyPressEvent再在CreateGP时设置响应键盘事件，即可通过按动键盘的方向键来移动飞机的位置，具体如下：

1. /\*mainwindow.cpp\*/
2. //键盘事件，控制飞机的移动，同时设置移动边界
3. **void** MainWindow::keyPressEvent(QKeyEvent \*event){
4. //判定为上方向键，向上移动
5. **if**(event->key()==Qt::Key\_Up){
6. **if**(P\_Y>=0){
7. P\_Y-=5;
8. Plane->setGeometry(P\_X,P\_Y,P\_HEIGHT,P\_WIDTH);
9. }
10. }
11. //判定为下方向键，向下移动
12. **if**(event->key()==Qt::Key\_Down){
13. **if**(P\_Y<=520){
14. P\_Y+=5;
15. Plane->setGeometry(P\_X,P\_Y,P\_HEIGHT,P\_WIDTH);
16. }
17. }
18. //判定为左方向键，向左移动
19. **if**(event->key()==Qt::Key\_Left){
20. **if**(P\_X>=0){
21. P\_X-=5;
22. Plane->setGeometry(P\_X,P\_Y,P\_HEIGHT,P\_WIDTH);
23. }
24. }
25. //判定为右方向键，向右移动
26. **if**(event->key()==Qt::Key\_Right){
27. **if**(P\_X<=750){
28. P\_X+=5;
29. Plane->setGeometry(P\_X,P\_Y,P\_HEIGHT,P\_WIDTH);
30. }
31. }
32. }
33. 子弹、敌机的产生，碰撞事件的判定：

此处重写了timerEvent事件。

1. /\*mainwindow.h\*/
2. //重写timerEvent设置游戏界面的定时器事件
3. **void** timerEvent(QTimerEvent \*t) override;

此处设置三个分别控制子弹发射，敌机产生，判定碰撞的三个定时器。

1. /\*mainwindow.cpp->InitialP()\*/
2. //随即开始主界面的定时器事件
3. Self\_BID=startTimer(BT\_GAP);
4. Ener\_BID=startTimer(5\*BT\_GAP);
5. Crashed=startTimer(20);

通过重写的timerEvent事件来创建子弹、敌机，并将其加入到各自的链表中，用于之后的碰撞判定。**在碰撞判定的事件定时器事件中每20ms对所有的子弹和敌机进行遍历判定是否发生了碰撞事件：**先遍历所有的子弹，对每一个子弹的位置去遍历敌机的位置去判定，若子弹在敌机的范围之内则已发生碰撞事件，则分别调起Crash函数销毁对象，并从对应的list中删除该对象(防止因再次遍历时访问到空指针而出错闪退)，同时得分表设置分数加10。

1. /\*mainwindow.cpp\*/
2. //游戏界面的总定时器事件
3. **void** MainWindow::timerEvent(QTimerEvent \*t){
4. //生成子弹的定时器
5. **if**(t->timerId()==Self\_BID){
6. //构建子弹类的对象指针，并加入存储子弹的链表
7. Bullet \*Bt=**new** Bullet(**this**);
8. My\_Bullet.push\_back(Bt);
9. }
10. //生成敌机的定时器
11. **if**(t->timerId()==Ener\_BID){
12. //构建敌机类的对象指针，并加入存储敌机的链表
13. Enermy \*Em=**new** Enermy(**this**);
14. Enermies.push\_back(Em);
15. }
16. //判定碰撞事件的定时器
17. **if**(t->timerId()==Crashed){
18. //判定是否已碰撞
19. **int** Csh=**false**;
20. //遍历存储子弹的链表
21. **for**(auto M\_B:My\_Bullet){
22. **int** PBx=M\_B->P\_CurrX+B\_WIDTH/2;
23. **int** PBy=M\_B->P\_CurrY+B\_HEIGHT/2;
24. //对每一个子弹对象，遍历所有敌机判定是否碰撞
25. **for**(auto E:Enermies){
26. //判定条件：若子弹与敌机的图片已重合，则视为已碰撞
27. **if**(PBx>=E->E\_CurrX&&PBx<=E->E\_CurrX+E\_WIDTH){
28. **if**(PBy>=E->E\_CurrY&&PBy<=E->E\_CurrY+E\_HEIGHT){
29. //即刻从相应的链表中删除对应的对象指针
30. My\_Bullet.remove(M\_B);
31. Enermies.remove(E);
32. //触发碰撞事件的函数
33. M\_B->Crash();
34. E->Crash();
35. //得分增加
36. score+=10;
37. Score->setText("得分："+QString::number(score));
38. //已碰撞，不再遍历，退出循环
39. Csh=**true**;
40. }
41. }
42. **if**(Csh){
43. **break**;
44. }
45. }
46. **if**(Csh){
47. **break**;
48. }
49. }
50. }
51. }
52. 子弹/敌机的移动与状态设置：

设置子弹/敌机的定时器对应连接的槽函数，并开启定时器事件

1. /\*mainwindow.cpp->Bullet::Bullet 敌机与子弹移动原理一致，这里以子弹为例\*/
2. //初始化定时器，用于子弹的移动
3. timer=**new** QTimer(**this**);
4. timer->start(20);
5. //连接定时器对象与与之对应的槽函数
6. connect(timer,&QTimer::timeout,**this**,&Bullet::Timer);

定时器每隔如上设定的事件执行如下代码操作，当判定子弹/敌机的位置超出屏幕，则销毁相应的对象释放内存，同时从list中移除以防止遍历至空指针出错，导致闪退。

1. /\*mainwindow.cpp 敌机于子弹移动原理一致，这里以子弹为例\*/
2. //子弹timer对象的槽函数，用于控制子弹的移动
3. **void** Bullet::Timer(){
4. //定时设置子弹的位置
5. P\_CurrY-=4;
6. bullet->setGeometry(P\_CurrX,P\_CurrY,B\_HEIGHT,B\_WIDTH);
7. //若子弹超出屏幕，则销毁该子弹
8. **if**(P\_CurrY<0){
9. //从存储子弹的链表中删除对应子弹的对象指针
10. My\_Bullet.remove(**this**);
11. //停止定时器
12. timer->stop();
13. //逐一销毁相关的对象指针释放内存，销毁子弹
14. **delete** bullet;
15. **delete** bul\_img;
16. **delete** timer;
17. **delete** **this**;
18. }
19. }
20. 重新开始游戏的实现：

当重新开始的按键按下，则调起如下的函数，对每一个子弹/敌机的对象逐一销毁，同时清空链表，飞机位置置为初始并且分数置零。

1. /\*mainwindow.cpp\*/
2. //重新开始的槽函数
3. **void** MainWindow::Restart(){
4. //遍历存储子弹对象指针的链表，移除所有的子弹
5. **for**(auto B:My\_Bullet){
6. My\_Bullet.remove(B);
7. B->Crash();
8. }
9. //清空链表
10. My\_Bullet.clear();
11. //遍历存储敌机对象指针的链表，移除所有敌机
12. **for**(auto E:Enermies){
13. Enermies.remove(E);
14. E->Crash();
15. }
16. //清空链表
17. Enermies.clear();
18. //得分置零
19. score=0;
20. Score->setText("得分："+QString::number(score));
21. //飞机位置置为初始位置
22. Plane->setGeometry(350,400,P\_HEIGHT,P\_WIDTH);
23. P\_X=Plane->x();
24. P\_Y=Plane->y();
25. }
26. 退出程序的实现：

点击退出调起quit函数退出整个程序。

1. /\*mainwindow.cpp\*/
2. //退出的槽函数
3. **void** MainWindow::On\_Exit(){
4. QApplication::quit();
5. }
6. 网页的打开：

点击关于下展开的一个选项，立即调起系统默认的浏览器打开如下地址

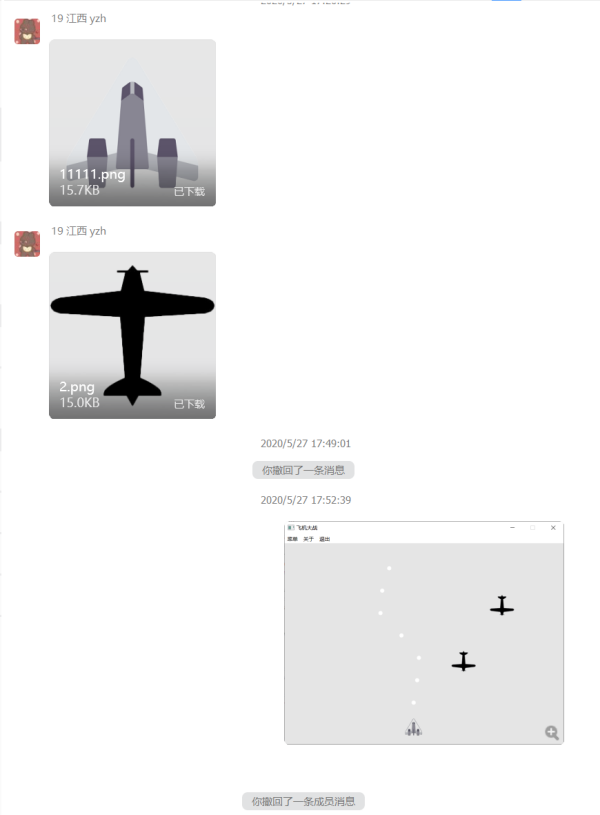
的网页。

1. /\*mainwindow.cpp\*/
2. //打开相关选项的槽函数
3. **void** MainWindow::OpenWeb(){
4. //设置指定的链接
5. Url="https://virtual-memes.gitee.io/about/";
6. //调用本地默认浏览器打开网页
7. QDesktopServices::openUrl(QUrl(Url.toLatin1()));
8. }

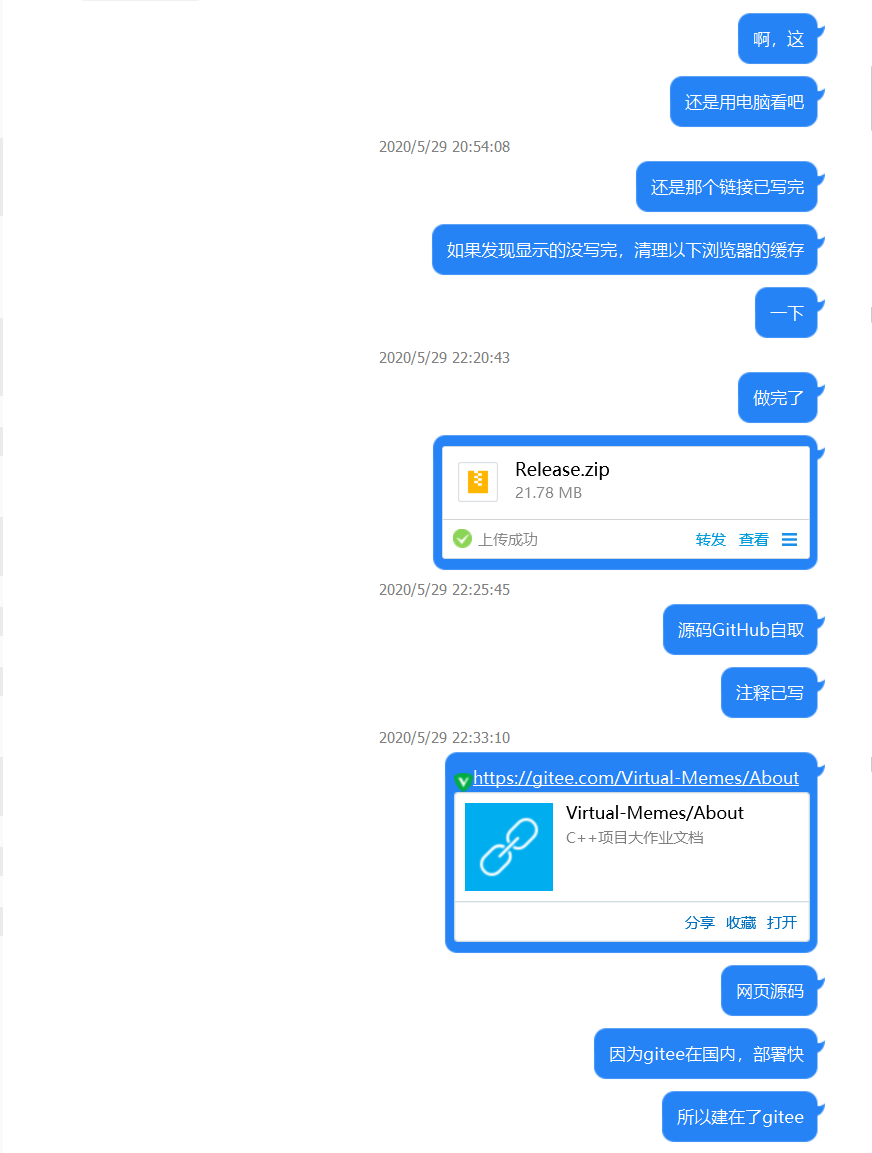
# 小组讨论日志（独立完成的可以省略该内容）

*2020/5/25-就Bug进行讨论：*

*2020/5/27-项目资源交换，UI美化：*



# *2020/5/29-就网页进行组内展示：*

*2020/6/2-就报告编撰的流程图部分的讨论：*

# procedure个人小结

**赵英剑**：在本次的C++大作业项目之中，我主要负责设计程序主体结构，碰撞判定算法与，此次作业的介绍性质的网页的编写。在本次的C++大作业项目中，主要使用的是Qt这样的一个图形界面编写软件来编写程序。在编写过程中也常常会遇到例如：子弹无法飞出，飞机动作不响应键盘事件，碰撞判定完后闪退等bug，但通过分析代码结构，程序运行过程，并及时查阅资料，最终解决了这一系列的问题并成功地完成了此次大作业项目地编写（网页编写过程便不叙述）。由此次经验，我认识到一个程序的完成就是要经历无数次的失误，而最重要的是坚持不放弃，出错时分析错误，再从中寻得答案与解决办法。同时也是一种自我提升的方式。与此同时，团队之间的密切合作亦是极为重要的。如若只是一个人独自闭门造车，不与团队成员相互交流心得，怕是也无法顺利的完成此次的大作业项目。

**严之寒**：本次我在这个项目中负责的部分是程序UI设计与项目资源收集，起到了一个完善、美化程序图形界面的作用，贡献了整个程序大概百分之二十五的工作量，虽然不是特别多，但是也多多少少遇到了一些困难，如无法熟练使用Qtcreator这个软件的功能，编译器无报错但是运行结果有问题等，我积极学习使用方法，和组内其他人分析问题并解决。通过这次大作业我学习到了很多，首先自己的工程能力需要增强，合作能力也急需在之后的学习生活中提高。

**胡亚冰**：在这次实验中我的任务主要为辅助资源文件的查取（src，qrc），如飞机图形、敌机图形等。通过这次飞机大战的编写，我将自己c++所学的东西运用了起来，熟悉了所学知识，并对面向对象程序设计有了更加清晰地认识。我不仅更加熟练c++的语法规则和算法知识，而且对c++这门计算机语言的理解更加深了，也更加喜欢程序设计了，对c++的使用得到了进一步的提高，结构设计的合理性决定了这个程序的价值。在今后的学习中我要注意这方面，使得我的编程能力能有进一步的提高。这也需要感谢我们的组长赵英剑教会了我很多的知识和方法。在本次实验中，我主要起到了辅助的作用，辅助赵英剑同学对程序的编写，所以工作量相比组长少了很多。通过这次作业，也发现了自己有非常多不会的地方，虽然在过程中学会了一些但还是留有很多不足，希望以后加油。

# 参考文献

[1] C++ Primer [M]. 北京：电子工业出版社，2013. 9  
[2] 郑莉等. C++语言程序设计[M]. 北京：清华大学出版社，2010. 7

[3] Qt官方文档

[4] CSDN博客Qt教程资源