

# 程序设计实习 QT 大作业报告

林枫 刘和金

程序设计实习 QT 大作业报告

功能介绍

设计细节

开始界面

游戏主场景

游戏规则与机制

生命值

无敌状态

炸弹

火力

分数和奖励系统

游戏难度系统

结算界面

游戏运行逻辑

分工情况

总结反思

收获

不足

## 功能介绍

本项目是一个用 QT 开发的弹幕射击游戏，玩家需要通过键盘操纵自机移动，躲避敌机发射出的子弹，并通过拾取掉落物，发射子弹击败敌机来获得分数。游戏的难度会随着时间推进而增加，玩家的目标是在敌机的进攻下存活三分钟，并在此前提下尽可能获得更多的分数。

## 设计细节

### 开始界面

开始界面使用类 `MainWindow` 实现，该类继承自 `QMainWindow` 类，而 `QMainWindow` 类又继承自 `QWidget` 类。

该类包含的内容：

- 开始界面的大小。
- 标题的大小和位置。
- 背景图和标题图的设置。
- 开始游戏按钮和退出按钮，包含了进入动画、点击动画和点击音效。
- 设置背景音乐并循环播放。

## 游戏主场景

游戏主场景使用类 `GameScence` 实现，该类继承自 `QWidget` 类。

主场景界面包含的内容：

- 界面大小、位置、标题、背景设置。
- 自机：玩家操作的对象，通过键盘移动，移动方式分为快速和慢速两种，可以发射子弹，随着火力值增加，自机周围会陆续出现四个僚机用于发射子弹，自机图片中心有判定点，判定点被敌机和敌机子弹击中后会扣血。
- 自机子弹：玩家发射的子弹，子弹有一定攻击力，击中敌人一定次数可以消灭敌人，子弹会随自机火力值增加样式和攻击力。
- 敌机：要击败的敌人，会随机从游戏界面顶端任意位置出现，移动轨迹为从上到下固定速度的直线，会发射一定数量的子弹，有一定生命值，被玩家子弹击中后会根据玩家子弹当前攻击力扣除生命值，生命值归零后会被消灭并掉落红蓝 P 点，敌机会随游戏进程增加生命值、子弹数量、子弹速度和出现数量。
- 敌机子弹：根据大小分为两种样式，每种样式有多种颜色，发射角度随机。
- 掉落物：种类有两种，红 P 点和蓝 P 点，在火力值达到上限之前击落敌机会掉落红 P 点，玩家吃到红 P 点之后增加火力值，火力值达到上限之后击落敌机会掉落蓝 P 点，玩家吃到蓝 P 点之后增加分数。
- 信息显示：当前分数、火力值、生命值、bomb 数、剩余时间。
- 音乐音效：背景音乐循环播放。当玩家释放 bomb、被击中、获得奖励、获得 P 点、敌机被击落时，会播放相应的音效。

## 游戏规则与机制

### 生命值

玩家初始生命值为 3，被击中后扣除 1 生命值并进入 3 秒无敌状态，并回到初始位置，玩家生命值会在界面左下角显示。敌人初始生命值为 100，敌人被玩家击中后会根据子弹的攻击力扣除一定生命值，敌人生命值归零后判定为被击落。

当左下角显示的生命值为零时，玩家再被击中一次视为游戏失败。

### 无敌状态

该状态下玩家撞击子弹和敌机时不会扣除生命值，同时玩家自机图片变蓝用于和通常状态进行区分。

### 炸弹

使用炸弹可以将当前视图内的所有敌机和敌机子弹清空，并进入 3 秒无敌状态。

当左下角显示的 bomb 数大于 0、并且距离上一次释放 bomb 超过 3 秒时，按下 X 键即可释放 bomb。

初始 bomb 数为 4，之后每当生命值减少时，将当前的 bomb 数重置为 4，此外还有其他的获得方式，具体将会在下面的分数和奖励系统中介绍。

### 火力

击败敌方时会掉落红 P 点，玩家每吸收一个红 P 点后可以增加 0.05 火力值，火力值的提高会增加玩家子弹的伤害。

初始火力值为 1.0，最大火力值为 5.0。当火力值分别达到 2.0，4.0 时会增加自机子弹样式。当自机被击中时会扣除 1 点火力值，但不会低于 1.0。

## 分数和奖励系统

当玩家火力值到达上限 5.0 后，击败敌方时不会掉落红色 P 点，而是掉落蓝色 P 点，玩家吸收蓝色 P 点之后左上角显示的分数会增加，击败敌人同样也会根据敌方被击败的位置获得一定分数，敌方越靠近板底分数越高。

分数每增加 5000 时，玩家可以额外获得一个 bomb，每增加 10000 时，可以额外获得 1 点生命值，游戏过程中一共可以通过这种方式获得十个 bomb 和五次生命值加成。

## 游戏难度系统

游戏时长上限为三分钟，玩家存活三分钟则游戏通关，在界面右上角设置有读秒计时器，用于提示游戏事件进程。

随着时间推进会增加游戏难度，每过半分钟会增加敌机子弹数量、速度和敌机生命值，并缩短敌机出现时间，玩家需要预留足够资源用于应对后期的弹幕。

## 结算界面

结算界面使用类 `GameOverScence` 实现，该类继承自 `QWidget` 类。

结算界面包含的内容：

- 页面标题、大小、位置。
- 背景：根据游戏通关与否设置两种背景。
- 标题：根据游戏通关与否设置两种标题图，通关显示 game clear，失败显示 game over。
- 返回主菜单按钮和退出游戏按钮。
- 分数显示：显示游戏所得分数。
- 背景音乐：两种，循环播放。

## 游戏运行逻辑

游戏运行的核心在于游戏画面的实时更新，主要由以下函数实现：

`void initialize();` 用于初始化游戏界面各参数信息，由该函数调用 `GameStart();`。

`void GameStart();` 开始游戏，启动各计时器，启动间隔为 10ms 的总计时器，利用 `connect` 函数在每次 timeout 时调用 `updatePosition()` 函数更新坐标和 `update()` 函数更新绘图，从而实现游戏画面实时更新的效果。

`void updatePosition();` 调用各个成员类对象对应的更新坐标的函数，使画面中的每一个对象的位置更新。

`void *paintEvent*(QPaintEvent *event);` 根据 `updatePosition()` 函数更新的坐标实时绘制画面中的每一个成员。

## 分工情况

林枫同学主要负责了项目创意的提出以及完善、游戏主场景的编写实现、功能测试、演示视频录制等工作。

刘和金同学主要负责了游戏开始界面与游戏结算界面的编写实现、项目报告的撰写等工作。

# 总结反思

## 收获

- 利用四个函数的嵌套实现游戏界面的不断刷新功能，该方法不仅在 stg 类游戏中适用，还可以运用到各种游戏的编写中。
- QTimer、QPainter、QLabel、QKeyEvent 等类的应用。
- 利用 git 管理项目的开发。

## 不足

- 没有设置阶段性开发目标对项目进度进行检验，导致后期工作量偏大，在以后的开发过程中应当注意。