

# 作业报告 - 《原神：星穹铁道》

## 程序功能介绍

你说的对，但是《原神：星穹铁道》是由我们组自主研发的一款全新卡牌对战游戏。游戏发生在一个被称作「七圣召唤牌桌」的幻想世界，在这里，你将扮演一位名为「棋手」的神秘角色，操控4个属性各异，能力不同的伙伴们，和他们一起击败强敌，通过深渊的考验——同时，逐步发掘「qt」的真相。

简而言之，你将拥有4个棋子，与一个boss对战，每一个角色有普攻和技能两种攻击方式，一共有5个技能点，使用技能会消耗技能点，使用普攻会回复技能点，每一个角色有不同的技能和属性，你需要灵活运用这一切可以利用的东西，打败boss，通过关卡。

## 规则解释

本游戏战斗规则的灵感来自于原神的元素反应和星铁的韧性条。

相信玩过这两款游戏的同学能很好地上手。

## 回合制

本游戏的回合制参考自星铁，是一个半回合制游戏。

## 回合

每名角色行动一次称为该角色一个回合。角色间的回合独立，互不影响。

## 行动顺序

在屏幕的左侧，显示了角色们的行动顺序。角色的行动顺序与角色的速度有关。可以粗略的理解为角色当前速度越快，行动越快，行动次数越多。

复杂版：

首先说速度对于战斗的基本原则：速度决定两次行动之间的间隔。

数值表现为：行动条长度/速度（初始行动条长度为10000）

可以理解为没过1点时间单位，行动条长度减少速度的数值。

不论是角色还是怪物都有速度，行动都依照这一公式进行回合排序。

如果将一场战斗的回合排序分作一个个点的话，战斗顺序就可以简单理解为下面这种情况。

设定一个单位的速度为100，那么该单位的下一次行动会在100点后进行。

那另一个单位速度是200，那就只间隔50点了。

在排序上，50点行动自然比100点要排序前一些。

除此之外，如果将角色加速或者减速，那么角色下次行动的时间在剩余行动条长度/当前速度

## 角色

### 敌方角色

Boss 是一个强力的敌人，他每一轮回合有且仅有一种方式进攻，就是普攻，会以 `bot.h` 中的方式进行攻击，具体是什么样子需要玩家在对战中自行探索。

与角色不同的是，boss拥有一个护盾，在护盾存在的时期可以降低角色对其造成的伤害，但是角色可以通过元素攻击击破护盾，特别的是，在护盾破碎的瞬间还会对boss造成一定的伤害。

### 玩家角色

玩家拥有四名角色，每名角色拥有两种行动方式。

- 普攻：对敌方boss造成角色对应属性伤害。如果boss身上没有元素，则挂上该元素。否则与boss身上元素产生反应。除此之外，还会回复一点战技点（上限五点）。
- 战技：战技会消耗一点战技点，因此仅能在拥有战技点时使用。每个角色的战技拥有不同的功能，下面将从左至右介绍：
  - 莫娜：回复我方全体生命值
  - 迪卢克：使我方全体加速，持续2回合
  - 神里凌华：对敌方造成冰属性伤害并使其减速，持续2回合
  - 影：使我方全体受到伤害减少，持续2回合

## 项目各模块与类设计细节

---

前端：

- `mainwindow.h/cpp`：主窗口，实现游戏刚开始的启动界面，通过槽和信号连接了游戏界面，载入界面，规则界面和退出的功能。
- `gamewindow.h/cpp`：游戏窗口，用 `QPainter` 来绘制精美的游戏画面，同时用两个 `QPushButton` 来代表普攻和技能。
- `rulewindow.h/cpp`：规则窗口，用了一张图片来简单阐述游戏规则和特性。
- `endwindow.h/cpp`：在一轮卡牌游戏结束后的一个 `GameEnd` 窗口，提供了下一关，保存退出的选项。

后端：

- `role.h/.cpp` 包含了对战系统。记录了一名角色和怪物的技能机制以及游戏运行的主进程。由于各角色和怪物之间的机制具有很强的相似性，使用多态和派生来写这部分内容使代码更加简练。具体地：所有的类都有基类 `person` 派生而来，它包含所有角色都有的元素种类、血量等信息。再由基类派生出 `player` 类和 `monster` 类，代表玩家角色和敌方怪物的基类。再有二者派生出具体的角色和怪物。
- `bot.h/.cpp` 包含了怪物的出招系统。它将根据当前局面进行贪心地出招，对玩家的操作提出了一定考验。

## 小组成员分工情况

---

郭劲豪： 前端设计和图形绘制。

杜少龙： 后端接口以及对战系统的实现。

## 项目总结与反思

---

这次通过一学期面向对象的学习，最终实现了一个依托星穹铁道和原神中七圣召唤的机制的卡牌对战游戏，而我们最终也从什么都不懂的小白到可以运用 `QPainter`，`QString`，`QWidget` 等来实现一个比较完整的GUI画面。

从起初的大作业设计报告中天马行空的构想，再到开始实现后不断阉割功能，然后重新构想代码结构，不断地通过思考和重写代码，锻炼了自己构建项目的能力吧。这次不那么成功的大作业经历也让我们知道以后，在动手实践之前，首先要把对应的项目文档写好，包括要构建什么类，类与类之间的关系等，希望在下次的大作业之中能够将这次所积累的经验运用上去。