

# QT 大作业报告

报告组名：无人起名的组 题目：单机游戏——Divide (又名除法消消乐)

报告人：杨子涵（组长）、张金涛、马天麟

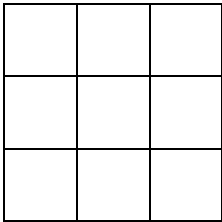
日期：8 / 7 / 2023

报告内容：

- (1)程序功能介绍
- (2)项目各模块与类设计细节
- (3)小组成员分工情况
- (4)项目总结与反思

一、程序功能介绍

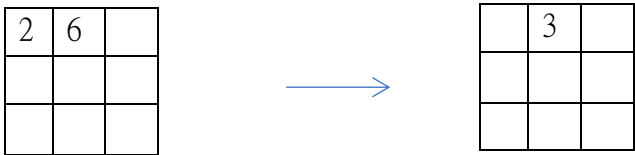
借鉴游戏 divide 消除，我们组使用 qt 程序开发框架实现了除法消除游戏的基本功能，并在规则上有一定创新。



游戏主体为一 3x3 网格，初始状态为空网格。除网格外，初始界面包括一个计分板、一个历史最高得分板、两个待给数字、一个数字缓冲区域以及重新开始按钮。游戏开始后，玩家需要将系统给出的数字方块填入网格内并进行消除操作以获取尽可能多高的分数直到网格内无法再填入新提供的方块。同时，为了游戏可以拥有更多的可玩性以及玩家可以有更多的得分空间，设置了一个缓冲方块，玩家可以将系统给出的数字方块放进缓冲方块内，先将其余方块填入网格内，然后在合适的得分时刻放出缓冲方块内的数字以获取更高的分数。整个程序设有初始界面，点击进入游戏界面。游戏过后（系统判定游戏结束）会跳出得分界面，显示该局玩家得分。可通过得分界面 retry 键再次进入游戏。

游戏规则

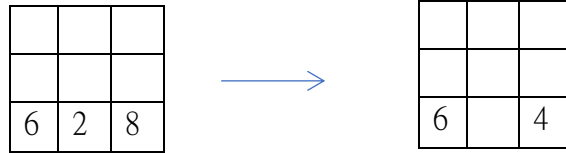
最基本的规则是：随着游戏进行，玩家需填入系统逐个提供的数字方块。如果相邻方块呈整除关系，则除数方块消失，被除数方块数字缩小为除式的商。如下图所示，如果有左上格填入 2，中上格填入 6，则左上格消除后为空，中上格变为 3。



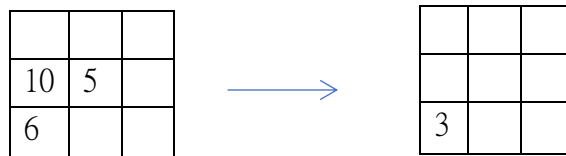
除此之外，在最基础的规则下会出现连续消除以及同时消除的情况，这些除法补充规则均在最基本的规则下加以补充完善，以下一一举例说明。

除式触发补充规则：

1. 当新放入的数字可以同时和周围多个方块进行消除时，优先跟得分高的方块进行消除。（例：如果有左下格为 6，右下格为 8，当在中下格放入 2 时，两边除式同时触发。左下不变，右下变更为 4，中下为空）



2. 允许除式连续触发，即一个除式的结果导致新除式的产生，称为 Combo。  
（例：如果有左下格为 6，左中格为 10。当在中心格放入 5 时，中心格数字消失，左中格变更为 2，并与左下格的 6 触发新除式。最终结果为左下变更为 3，左中格与中心格为空）



积分规则：

1. 每放入一个方块积一分。
2. 每次触发除式，增加积分为被除数与商之差。
3. 若形成 Combo，新除式积分在以上计算方法基础上乘以二。若继续 Combo，新除式积分乘以三，以此类推。

## 二、项目各模块与类设计细节

文件夹包括以下内容。

Headers

- ✧ mainwindow.h // 定义 MainWindow 类
- ✧ randomnumber.h // 定义有 math 类，RandomNumber 类

Sources

- ✧ main.cpp
- ✧ mainwindow.cpp

Forms

- ✧ mainwindow.ui // MainWindow 界面的 UI 设计

Resources

- ✧ res.qrc
  - BackGround :
    - GameOver.png // 游戏结束画面背景图
    - START.png // 初始界面 ClickToStart 画面背景图
  - SoundEffect :
    - devideBox.mp3 // 触发除式音效
    - over.mp3 // 游戏结束音效
    - putBox.mp3 // 放置数字方块音效

randomnumber.h 文件

```
class math{
    static int gcd(int n,int m);//求 n 和 m 的最大公因数};
class RandomNumber{
    int aval_num[101], total, difficulty, amount;
    // amount 是 get 函数的调用次数 代表出了几个数了
    RandomNumber(int _difficulty=15)//构造函数，缺省的话默认难度系数 15
    void change_difficulty(int _difficulty)// 自由更改难度系数
    int get(int num[]) //输入现下网格及缓冲格的数，根据难度系数（越大越难消）、
    amount（越大数越大）生成合适的“随机数”
};
```

MainWindow 中用到的类总结

界面部分运用了 QWidget 类搭建，并通过其成员函数进行标题和背景设置。从运行效果来看，程序一共有三个界面：初始界面、主界面（即游戏界面）、结束界面。实际上，代码实现层面，程序仅有两个界面，分别是主界面和它的子界面——结束界面。初始界面则是由主界面上一个全屏按完即关闭的按钮实现，是一个"伪界面"。各界面上的 "零件" 则是运用了 QPushButton 、QLabel 、及 QLCDNumber 类，并通过自动和手动信号槽连接，实现按键显示与界面切换。此外，程序所有按键音效由 QMediaPlayer 类实现。

MainWindow 成员函数总结

MainWindow 成员函数除了构造函数和析构函数，皆为私有函数，各功能如下表。

	void init_vector();	初始化容器	
	void update_data(int pos, int newNum);	更新九宫格数据	
	void react_to_changes(int pos, int deep);	对于数据变化触发除法	
	void as_divisor(int pos);	新填充数字充当除数	
	bool whether_end();	判断结束	
	bool setBox(int current_pos, QPushButton* box);	每回合放置数字方块后处理	
	slot:		
	void on_box_1_clicked(); // 以及相似的从 box_2 至 box_9	若在 box_1,box_2...放置数字	
	void on_box_keep_clicked();	若将数字存入 box_keep/ 若运用 box_keep 的数字	
	void on_pushButton_clicked();	主界面中 restart 游戏	
	void Retry_clicked();	游戏结束界面 restart 游戏	
	void Start_clicked();	ClickToStart 进入主界面	

调用  
RandomNumber ::  
int get(int num[])

调用  
RandomNumber ::  
int get(int num[])

注：箭头代表调用，A 指向 B 即为 A 调用了 B

### 三、小组成员分工情况

杨子涵	程序 UI 设计	负责内容包括人机交互、操作逻辑和界面美观。人机交互以按键为主，通过显示内容和播放音效，予以反馈。程序主要运用信号槽关联，搭建主界面内以及初始、结束界面与主界面之间的操作逻辑。各界面美工部分则由本人运用其他平面设计平台完成。
	后端算法与前端的整合	编写整个程序框架并设计端口，将队友负责的内容顺利接入程序之中，进行团队统筹和整合。
张金涛	除法触发处理	玩家填入给出的数字方块后，通过函数对网格内的除法触发进行判定和处理，具体通过对每一小格以及周围格的整除判定来实现，多次连续除法触发通过递归加记录次数深度实现。
	计分板积分实现	玩家填入给出的数字方块后，若存在消除行为，则按照积分规则进行积分，每放入一个方块积一分。若触发除式，则按积分规则处理触发除式及其 combo 的额外积分。其中 combo 积分倍数与连续触发除法深度相关。
马天麟	随机数的生成机制	随机数生成机制主要关注于针对目前格子中有的数进行相关的生成，关注于游戏难度随时间变化，同时保证随机数有较大的概率使游戏继续进行（不会陷入僵局）。这保证了游戏的可游玩性。在生成机制上我们也进行了多次的商酌和调整，致于寻找更合理的随机数生成。
	管理 github 上项目仓库	管理 github 则是负责上传我们的项目到 github 上托管，来更便捷的共享项目文件，同时负责总体项目的打包上传。

#### 四、项目总结与反思

本篇将就我组在集体编写程序时各方面遇到的阻碍，进行反思与学习，并与末尾处作结。

界面设计方面，本身已经意料到会有颜色统一的问题，因此提前设定了简易色表及对应色号的不同显示格式。然而在设计过程中，还是出现了各个界面颜色风格不统一的问题。主要原因包括各平台同色号颜色有色差，自制色表颜色样式不足，以及设计过程中欠缺风格统一化考量。因此，为下次处理此类问题设以下解决方案。首先，所有界面背景应采用统一平台进行设计，再以相同格式放入 QT。其次，提前设定色表时，应将同颜色的明暗色系纳入考量，以供设计时参考。最后，多个界面的背景设计应同时进行并严格按色表设计，随时进行比对，以确保风格的统一。必要时可采用素材库的形式，提升设计效率。

实现除法触发处理方面，最为困难的地方是存在很多种不同种类的玩家填入数字的方式，如何让每一种情况都可以得到正确的按照规则方式的处理，就成了我们组需要解决的难题。在这个问题上，首先我们需要确保每一种消除方式的处理都是严格按照规则实现的，这就要求我们制定完善的游戏规则，在制定规则的时候便要完善各种情况。由于我们都是第一次实现这种类型的游戏框架，在制定规则的时候，我们并没有将全部规则一口气制定出来，而是先制定最基本和最容易延申的规则，然后运行游戏在游戏过程中找出会出现错误的情况并把出问题的地方进行修改，最终得出较为完善的版本，我们的游戏也许还存在其他消除触发的漏洞，以后有机会可以继续对相关完善。由于需要对每一种情况有不一样的处理方式，所以无法找出统一的除法处理方法，加上网格主体为 3x3，我们组采用了对一个网格极其周围网格进行穷举的办法来判定除法的触发，如果可以进行连续消除，则继续触发除法并且记录倍数，采用了记录消除次数深度的办法，最终成型了除法触发处理的函数整体。积分处理则与除法触发息息相关，所以一并加入了除法触发处理的函数内。

程序整合方面，在本次项目进行过程中，充分体现了完成集体编程项目的经验不足。由于在往日的计算机编程学习当中，过于习惯于个人作业，因此在进入集体作业后，产生了各式各样的问题。当中主要包括命名、注释、理解分歧等。举例，队友曾将某音效命名为 player 并在函数处理部分大量调用。在整合时，为了修改该音效命名，需要将所有该名字出现的地方进行更改。整合过程当中同时穿插着队友对函数处理部分的多次调试翻新（显然当中音效仍命名为 player），于是又要再改。整个过程中投放了过多时间在无谓的改名上，降低了团队效率。注释的欠缺阻碍了程序上的有效沟通，规则理解分歧产生了算法理解上的误会，严重拉长了整合程序的需时。综上，集体编程项目中，我们应提前充分沟通自个不同变量的命名，协商端口并至少粗略了解各自部分的算法逻辑，以提升团队效率。

运用 github 方面，我们也学会了善用搜索来进行学习。事实上，在编写代码，使用 QT 各种资源以及 github 时，我们都遇到了不少关键的问题，而正是使用各种搜索工具，利用网络资源进行自学，才让我们攻克难关，取得项目的完工。

这次项目的完成给我们留下了深刻的印象。虽然在整个项目中，我们团队面临了许多挑战，但我们仍最终成功地完成了任务。在这个过程中，我们不管是编程还是团队方面都吸收了很多宝贵的经验和教训，并有如下几点总结。

首先，团队合作是项目成功的关键。团队中的每个人都发挥了重要的作用，各自承担了相应的责任。小组成员间良好的沟通和协调是我们能够提高效率的关键。通过及时的沟通和有效的分工，我们能够横跨紧张的期末季及时解决问题并保持项目进度。此处亦感谢老师体恤，延长了时限。我们每个人完成不同的部分，最后进行整合即可顺利完成任务，这让我们省下了大量思考和整理代码的时间，更加专注于自己负责的部分，提高项目完成质量。

其次，项目计划和时间管理非常重要。在这个项目中，我们遇到了一些紧急情况和技术难题，这导致了项目进度的延迟。通过这一次的经验，我们意识到我们应该在项目开始之前制定详细的项目计划，包括清晰的里程碑和时间安排。同时，我们也需要为预计可能出现的问题留出一些缓冲时间，以应对不可控因素的影响，比如这次项目进行过程中军训安排的各项变更。

此外，个人能力的提升也是项目反思中需要考虑的方面。通过这个项目，我意识到自己在某些技术和领域知识上的不足之处。作为每个个体，我们仍需要不断学习和提升自己的能力，以完成我们每个人负责的任务。

通过这次项目以及对其深刻的反思与总结，我们每一个组员在 QT 编程以及团队协作方面都学到了宝贵的知识与经验，促进自我成长。