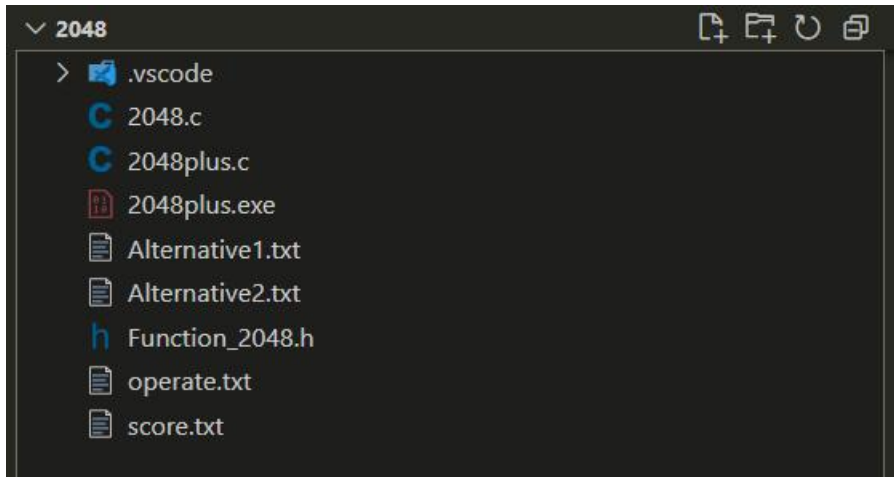


# 2048plus 开发文档

## 1: 项目文件夹介绍

### 【1】整体文件



### 【2】.vscode

包含了在 vscode 中的一些配置



### 【3】2048.c

期中 project，作为参考。

### 【4】2048plus.c

期末 project，源代码中有详细注释。

### 【5】2048plus.exe

2048plus.c 的可执行文件，需要与 Alternative1.txt 文件在同一个文件夹里方可正常运行。

### 【6】Alternative1.txt(Alternative2.txt)

包含了要替换的字符序列，可在 2048plus.c 源文件中通过更改宏定义来改变所要使用的文件。也可直接更改文档中的内容进行自定义。

### 【7】Function\_2048.h

自定义函数库，源代码中有详细注释。

### 【8】operate.txt 和 score.txt

分别保存操作和分数的文档，初始为空。

注：文件打开和保存模式都为 GBK。

## 2：游戏运行测试

此次测试采用 Alternative2.txt

```
Alternative2.txt
1 8
2 乾
3 坤
4 震
5 巽
6 坎
7 离
8 艮
9 兑
10
```

开始界面：

```
C:\Users\21714\Desktop\2048\2048plus.exe
#####Welcome to 2048#####
#   Please choose one command:   #
#       h: help                   #
#       s: start                   #
#       q: quit                    #
#####Have fun#####
_
|
```

输入 s 之后进入游戏界面，提示输入行列数。

```
C:\Users\21714\Desktop\2048\2048plus.exe
请输入行数和列数，若直接按m则设置为默认长度。
5 5
```

输入完成后进入游戏。

```

|-----|
|       |       |       |       |
|       |       |       |       |
|       |       |       |  乾  |
|       |       |       |       |
|       |       |       |  乾  |
|-----|
已连击次数: 0
当前连击奖励分数: 0
合并次数: 0
本次多合奖励分数: 0
```

下面为运行一段时间后的成功案例：  
由于采用替换方案的最高数用“兑”代替，故首次出现“兑”即胜利。

```
| 坎 | 坤 | 兑 | 坤 | 坤 |
| 乾 | 巽 | 巽 | 离 | 震 |
|   |   | 乾 | 震 | 坤 |
|   |   |   |   | 巽 |
|   | 乾 |   |   | 坤 |
-----
已连击次数: 1
当前连击奖励分数: 1
合并次数: 3
本次多合奖励分数: 7682
恭喜你，成功过关！
Press any key to continue . . .
```

按任意键后显示分数。若再按一次则返回主界面。

```
基础分数: 745231
连击奖励分数总和: 159
多合奖励分数总和: 30837
游戏总得分: 776227
Press any key to continue . . .
```

在 operate.txt 文件中显示操作数和具体操作。

```
operate.txt
1 操作数: 393
2 具体操作为:
  assadwsssdswsaadwswaaaaaddwssaaddddsswwawwwwssaaddaddssaadwsssdwsssdssssdaasssdawasaadawaddaadddsadwd
  ddsassdsssdadwssdddaddsdswswadswdwdsswwdaawdwddssawwwsaswddwwwwwswsddwaawwadswaadaawdwddaddddwaawaa
  awddsaadddwdsdaaswddddaadwsawdwssadaassdwssadsswsaaddddaswassadswaddsswswawdsddwswaddswaaaawdwd
  sssassdssssdaddsssdwswswwwwdaddddwswassdddsdwadddddsadwaaddadsswsaasdaaddw
3
```

在 score.txt 文件中显示各种分数。

```
score.txt
1 基础分数: 745231
2 连击奖励总和: 159
3 多合奖励总和: 30837
4 游戏总得分: 776227
5 最高连击数: 6
6 最高连击得分: 8
7 最高多合数: 3
8 最高多合得分: 7691
9
```

下面为运行一段时间的失败案例：

```
-----
|      震      |      坤      |
|-----|-----|
|      坤      |      乾      |
|-----|-----|
很遗憾，挑战失败。
Press any key to continue . . .
```

按任意键后显示分数。（再按任意键后返回）

```
基础分数：      0
连击奖励分数总和： 3
多合奖励分数总和： 18
游戏总得分：      0
Press any key to continue . . .
```

operate.txt 文件中继续显示第二次的操作数和具体操作。

```
operate.txt
1  操作数：      393
2  具体操作为：
   assadwsssdswsaadwswaaaaaddwssaaddddsswwawwwssaadaddssaadwsssdwsssdssdaassdawasaadawaddaadddsadwd
   ddsassdsssdadwssddadsdswswadswdwdsswwdaawdwddssawwwsaswddwwwswwsddwaawwadswaadaawdwddaddddwaawaa
   awddsaaaddwswdaaswdddaadwsawdwssadaassdwssadsswssaaddddaswassadswaddsswawdsddwswaddswaaaawdwddd
   sssassdssssdadddsdswwwwswwwwdaddddwswsssdsswswdwaddddsaadwaaddadsswswsaasdaaddw
3  操作数：      16
4  具体操作为： wwwwasawwwawwwsw
5
```

score.txt 文件中继续显示各种分数。

```
score.txt
1  基础分数：      745231
2  连击奖励总和：  159
3  多合奖励总和：  30837
4  游戏总得分：    776227
5  最高连击数：    6
6  最高连击得分：  8
7  最高多合数：    3
8  最高多合得分：  7691
9  基础分数：      0
10 连击奖励总和：  3
11 多合奖励总和：  18
12 游戏总得分：    21
13 最高连击数：    2
14 最高连击得分：  1
15 最高多合数：    1
16 最高多合得分：  15
17
```

下面为自动运行示例：  
按 g 后进入自动运行操作。

```
-----
|           |           |           坤           |
-----
|           |           |           |           |
-----
|           |           |           乾           |
-----
已连击次数:      0
当前连击奖励分数: 0
合并次数:      0
本次多合奖励分数: 0
请输入要自动随机执行的步数
_
```

输入运行步骤数。

```
-----
|           |           |           坤           |
-----
|           |           |           |           |
-----
|           |           |           乾           |
-----
已连击次数:      0
当前连击奖励分数: 0
合并次数:      0
本次多合奖励分数: 0
请输入要自动随机执行的步数
10_
```

运行结束。

```
-----
|           |           坤           |           |
-----
|           乾           |           震           |           |
-----
|           坤           |           震           |           乾           |
-----
已连击次数:      0
当前连击奖励分数: 0
合并次数:      0
本次多合奖励分数: 0
执行结束。 _
```

### 3：总结

本次 project 中涉及到了许多课上没有讲过的内容，例如如何表示汉字字符串，文件之间的通讯等。这些内容需要自行查询，学习。在这个过程中锻炼了我们的自主学习能力和搜索信息的能力。此外，程序中出现的 bug 也是数不胜数。例如 double 类型数据的运算，数据的溢出，程序进入死循环等。找到并解决这些 bug 也是程序设计中的重要一环。再者，函数的封装与调用也是尤为重要的，在面对较大的程序时，在一个 main 函数中写所有的代码已经不再合适了，这样既不利于分而治之实现问题的简化，也不利于出现错误时快速简洁地找到问题的根源。唯有设计并使用函数才可让主程序既简洁又易于理解，并且大大提高了程序的可移植性。