**共同工作**

**项目设计目标:小组认为该项目应做到游戏性，美观性，合理性的目标。**

游戏性：完成基本的象棋功能，即在完成基本的正确的象棋的移动，将军，吃子，绝杀等判定的基础上，再加入一个有特殊功能的按钮。

美观性：游戏要做到有自己的UI界面，同时有合适的，美观的棋子，棋盘，点击效果，落子效果，按钮的样式和排列以及背景图。

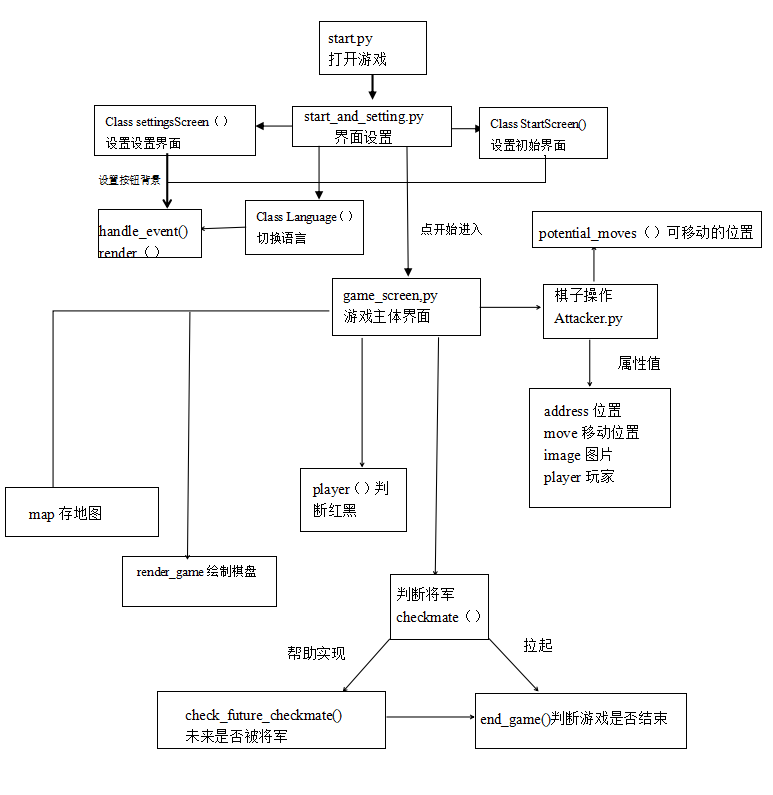
合理性：即符合当前市面上的象棋软件样式，本小组通过考察市场上象棋软件竞品以及询问象棋软件使用者来规范设计样式。使得本小组的象棋软件符合象棋软件的普遍形制。（灵感源于小组成员有参加智运会象棋项目遇上留学生象棋选手）设计有切换英文的功能，能使外国人也能适应该象棋软件。

**项目分工：**

俞乐楠：负责整体的统筹规划，制订代码基本思路，编写代码框架，进行游戏测试，Bug的调试与修复

袁昊旻：负责制订代码的编写，变量的命名规范，负责编写各个棋子对象的功能模块，以及进行UI的调整测试。

傅裕翔：负责编写棋盘显示区域以及绝杀的判定函数和“万宁象棋”功能，负责各阶段汇报的ppt制作

**代码总体框架：**

本项目由start.py, start\_and\_setting.py, game\_screen.py, attacker.py四个主要的文件组成以及image图片组成。其中这四个程序分别代表开始游戏，调设置、英汉，游戏界面，棋子移动。通过各段程序间的联动实现象棋功能，UI设计，行棋/绝杀提示以及中英文切换功能。总共代码行数约1200+行。

**第三方库介绍：**

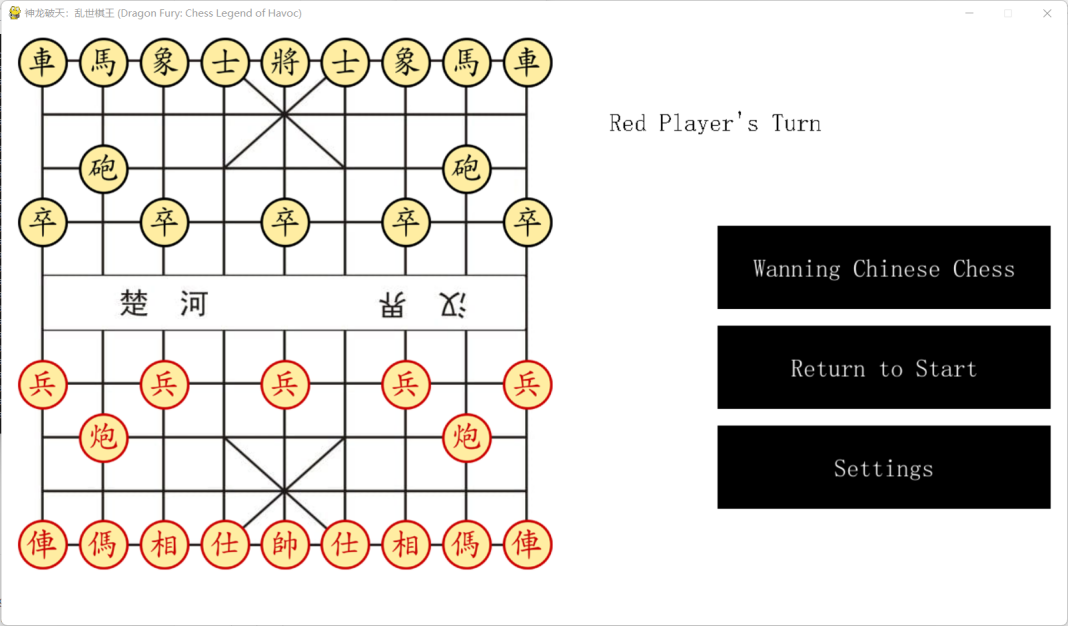
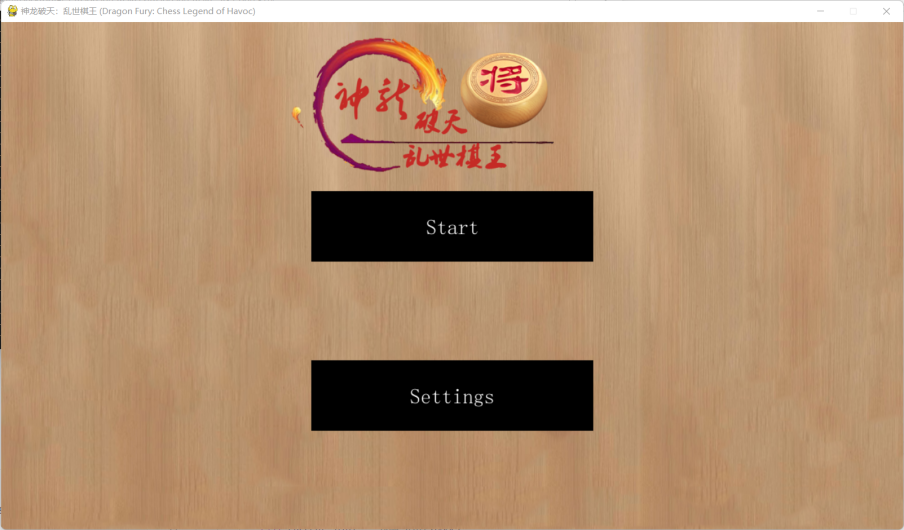
Pygame:Pygame是一个用于开发2D游戏的Python库。它建立在Simple DirectMedia Layer (SDL)库之上，提供了一系列功能丰富的工具和函数，用于创建游戏、图形和多媒体应用程序。提供了处理图像、声音、输入设备和网络等方面的功能。允许创建游戏窗口、渲染图像、处理用户输入以及播放声音和音乐。可以轻松处理碰撞检测、动画、粒子效果和物理模拟等游戏开发中常见的任务。

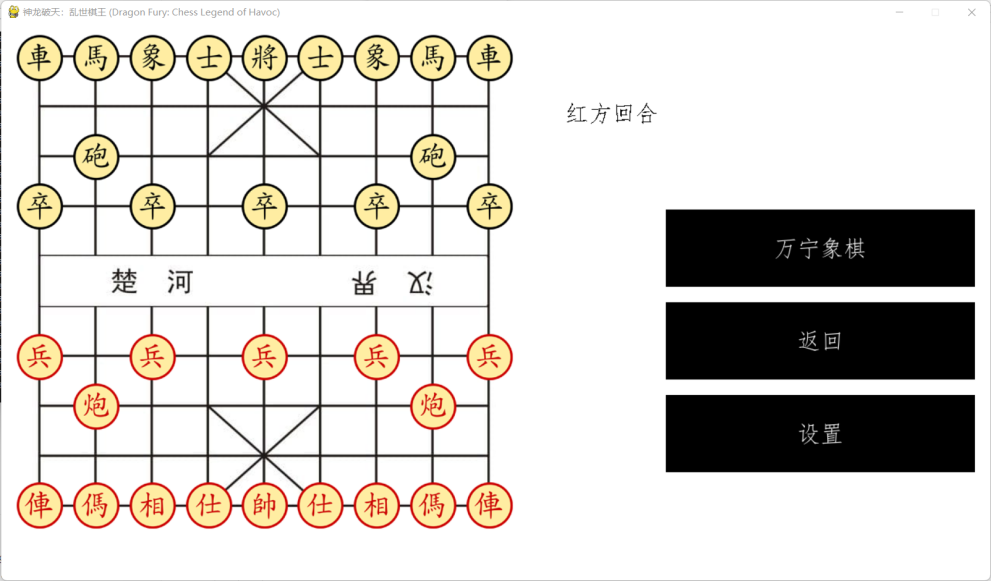
跨平台支持，可以在Windows、Mac和Linux等操作系统上运行。

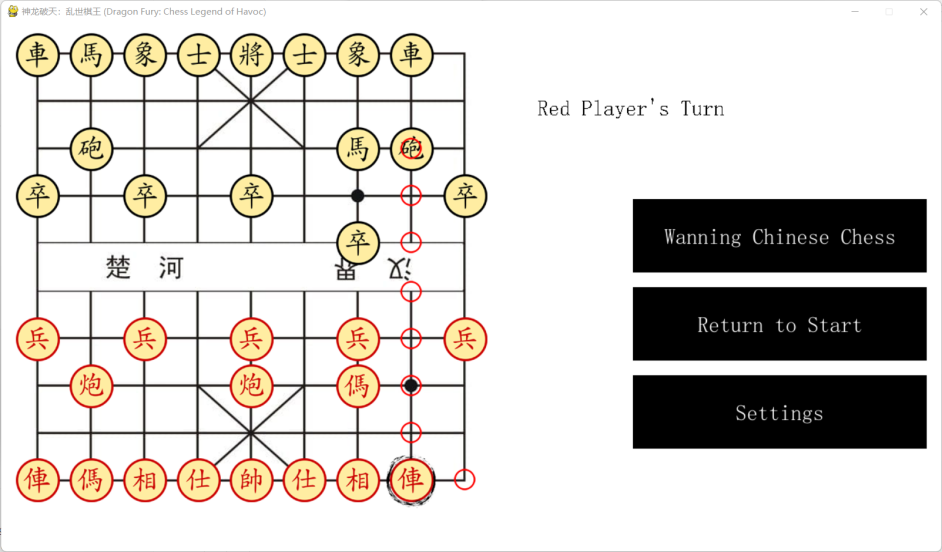
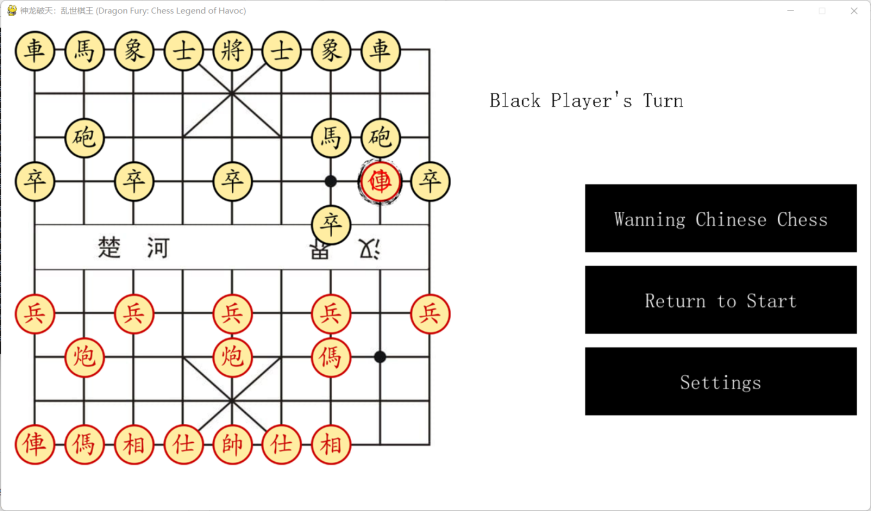
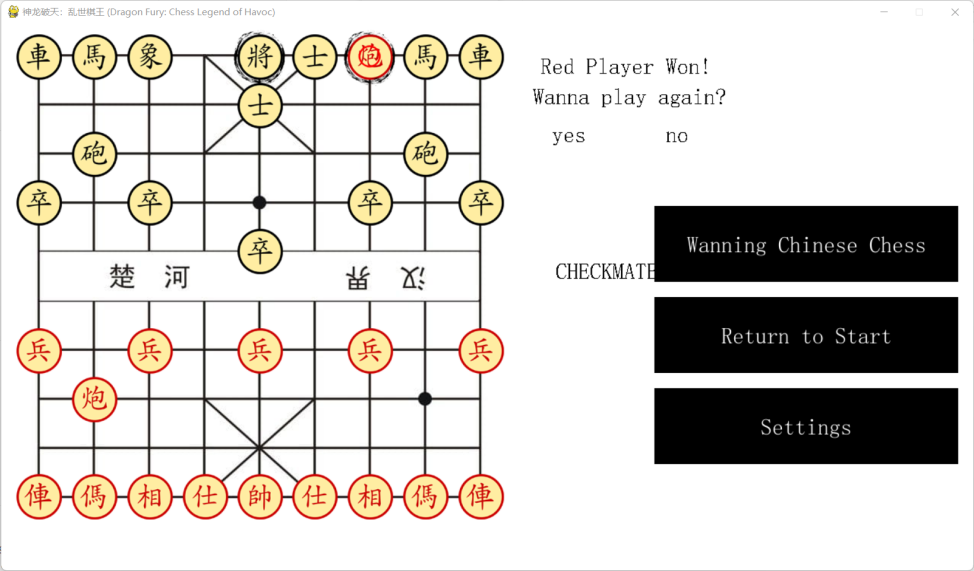
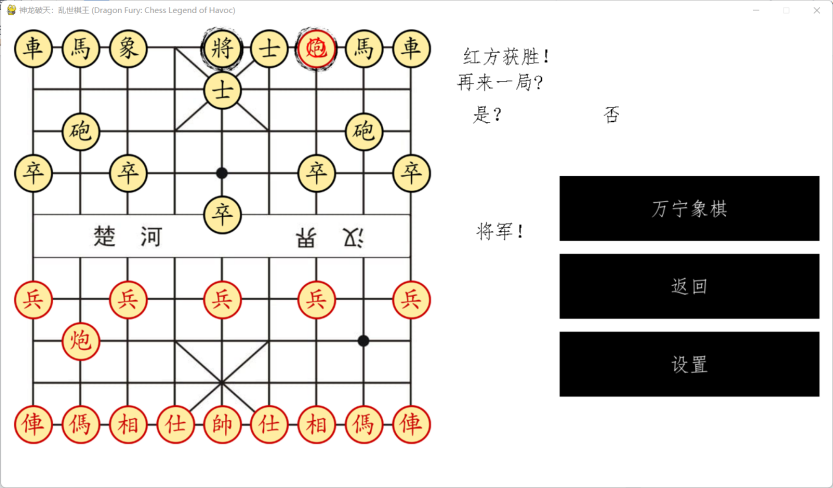
第三方库下载地址：<https://www.pygame.org/>

（不是第三方库但是需要下载的标准库）Copy：copy 是一个用于复制对象的标准库。它提供了两种不同的复制方式：浅复制和深复制。浅复制（Shallow Copy）是创建一个新对象，但是该对象的内容只是原始对象的引用。也就是说，新对象中的引用指向原始对象中的数据。当原始对象发生变化时，新对象也会受到影响。浅复制可以使用 copy.copy() 函数来实现。深复制（Deep Copy）是创建一个新对象，并且该对象的内容是原始对象的副本。也就是说，新对象中的数据是原始对象中数据的独立拷贝。当原始对象发生变化时，新对象不会受到影响。深复制可以使用 copy.deepcopy() 函数来实现。

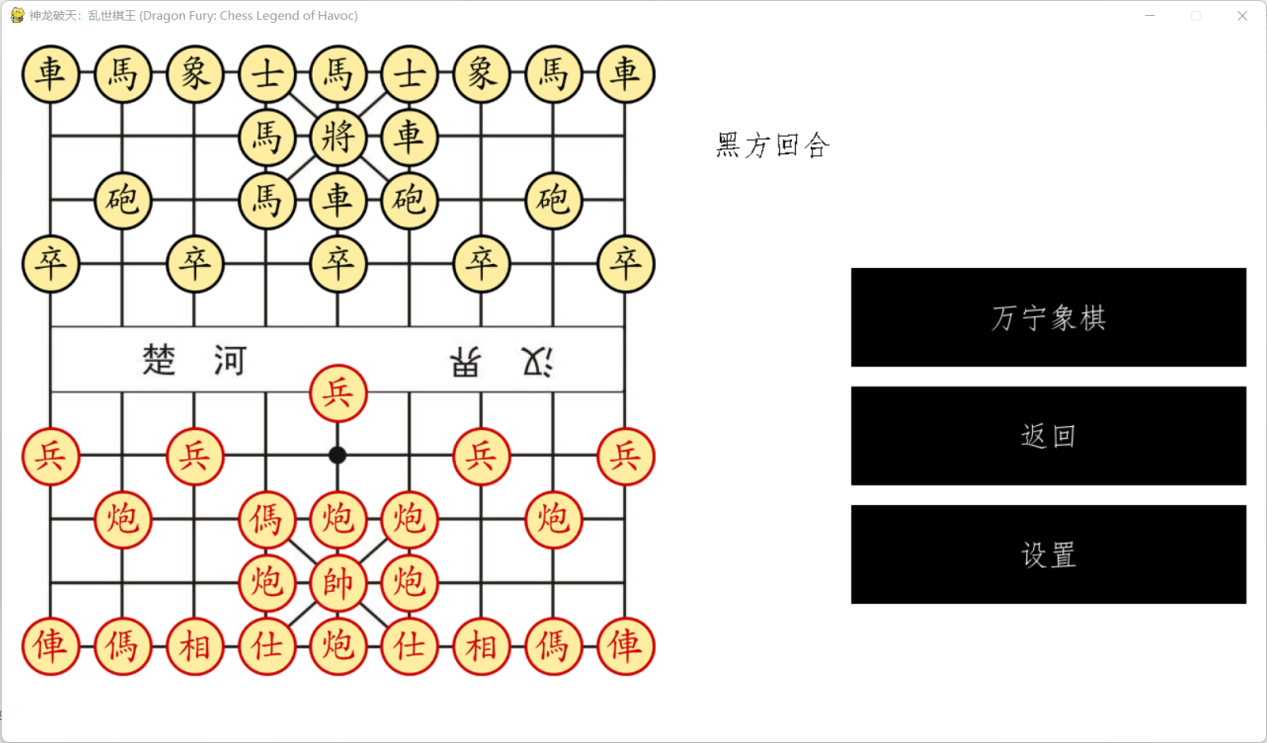
**软件运行截图：**

中英文版界面（封面参考天天象棋的封面并加上自己制作的logo，游戏界面采取简约风格）

选中，落子特效：

被绝杀样式

万宁象棋（按下后将帅旁随机出现车马炮）

**完成度评价：95%**

本项目基本完成了项目立项时期定下的目标，在小组成员的通力合作下，实现了界面切换，各个棋子的功能，棋盘显示，棋子操作以及象棋规则判定。做出了合理的UI，完成了这个面向对象的象棋项目。同时，为了使这个单纯的双人象棋对弈游戏更为有趣，加入了项目目标之外的但是在现今较为火热娱乐项目。

同时，小组通过git更新代码，加强对python使用的信息检索，合作完成这个开源项目，也给了我们很多宝贵的项目开发经验。

不足之处：小组成员在项目开发上的经验不足，处理方式略显稚嫩，例如写代码的时候易漏下self，把对象当成普通的字符串处理，以及将bug不全部修改完全便投入使用导致整个程序需要大幅度的，更费神的修改。

**个人工作**

本人首先在指定代码规范的环节，按照python公认的标准制定了代码规范和命名规范，考虑到python语言的特性和习惯，废弃了常用的驼峰命名法，详细规则如下：

1、包名：全部小写字母，中间可以由点分隔开，不推荐使用下划线。作为命名空间，包名应该具有唯一性，推荐采用公司或者组织域名的倒置，如com.apple.quicktime.v2。

2、模块名：全部小写字母，如果是多个单词构成，可以用下划线隔开，如dummy\_threading。

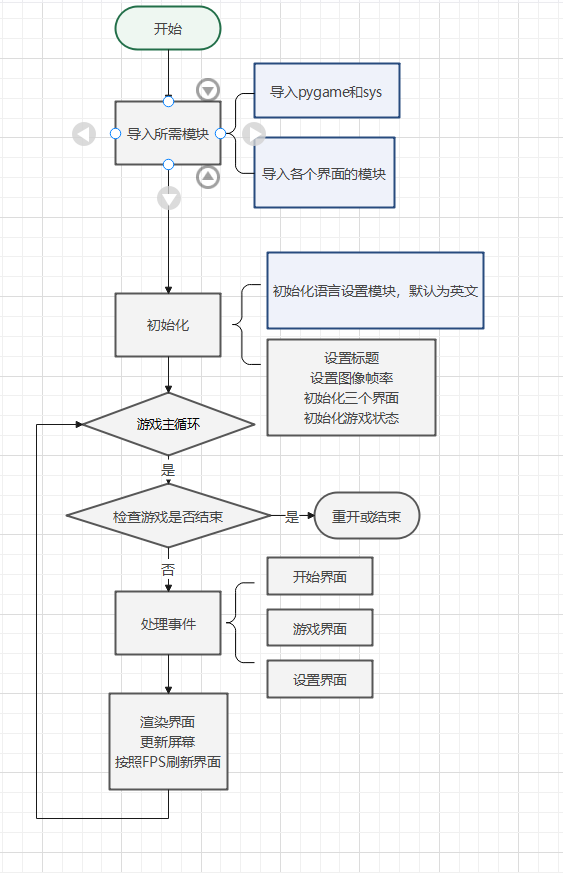
3、类名：总是使用首字母大写单词串。如MyClass。内部类可以使用额外的前导下划线。

4、变量名：变量名：全部小写，由下划线连接各个单词。如color = WHITE，this\_is\_a\_variable = 1。

5、函数名和方法名：函数名应该为小写，可以用下划线风格单词以增加可读性。如：myfunction，my\_example\_function。

6、常量名：全部大写字母，如果是多个单词构成，可以用下划线隔开，如YEAR和WEEK\_OF\_MOUTH。

接着，把start.py的文件写了，这是维持整个游戏有序运行最重要的主文件，包含了整个游戏运行的整体架构。主要实现的是按照设定的刷新率60Hz来刷新屏幕，并且实现不同界面的界面跳转与界面管理，有关该程序流程图如下。



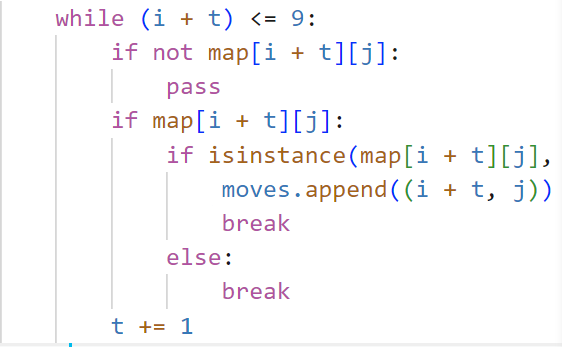
start\_and\_setting.py文件中东西不多，就涉及了这两个界面，比较有意思的是，对于中英文的切换这个功能的实现也放这里了。其实就是一个字典来实现中英文的切换。考虑到在github上进行了开源，会有一些外国友人游玩，就把默认语言设置成英文了。当然其他两个界面也没什么好说的，都不复杂。



在完成程序基本框架的搭建后，就把start\_and\_setting.py（也就是开始和设置界面也完成了），把game\_screen.py（游戏界面）的框架填充任务交给了傅裕翔，把attacker.py（棋子设计）任务交给了袁昊旻。

在此后的协作过程中，本人主要负责debug与代码的调试，下面是主要攻坚克难的debug举例：

1.这句if map[i][j]，在开始的时候是没有加入的，所以在程序运行的时候会报错，因为在map的某一坐标为空的时候，并不能使用这个类的方法。所以加入了这句代码。其实还有一个办法，就是对于Attacker基类派生一个Space的空类，这样就能使用公用方法并且不报错了。



2.在检测未来的将军的函数check\_future\_checkmate()函数中，一开始只有对于map内的每一个棋子的address进行了改变，但是并没有改变在map中每一个棋子相对应的位置，这会造成一个相当严重的bug，会导致可行棋子的位置造成偏差，对于游戏的结束也会造成误判。但是这也是被我最后一个攻克恶劣bug。

文本

描述已自动生成

3.还是上面那张图，map1=copy.deepcopy(map)这行代码，在预想中是需要拷贝一个棋子的副本以便于后面的操作不影响前面的所有棋盘。但是在运行过程中不符合我们预想的效果，报错是pygame.image.Image对象不能copy.deeepcopy。

原来是，在map中，map是列表套列表套了一个Attacker基类派生的棋子类。在棋子类的初始化中，有其中一个内部属性被初始化为pygame.image.Image类了，并且这个类不能被copy库的deepcopy()函数进行深度拷贝。要修复这个bug很简单，只要被copy库的deepcopy()函数调用的对象不包含pygame.image.Image类的对象就可以。但是对bug进行修复的过程相当复杂，必须把Attacker基类派生的所有派生类的初始化进行修改，后面把每一个类的图像的初始化改成相对地址，这样就能解决bug了。

后面再增加特殊功能的过程中，编写了娱乐功能的函数，在傅裕翔的个人文档中有所体现，再此处不多加赘述。