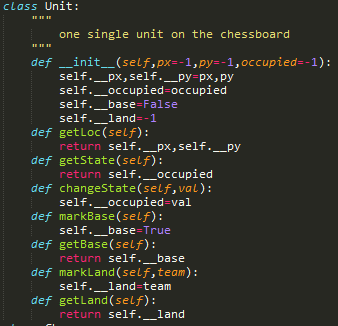
算法部分：

象棋游戏python算法部分分为基本构成和功能添加两部分。

1. 基本构成：

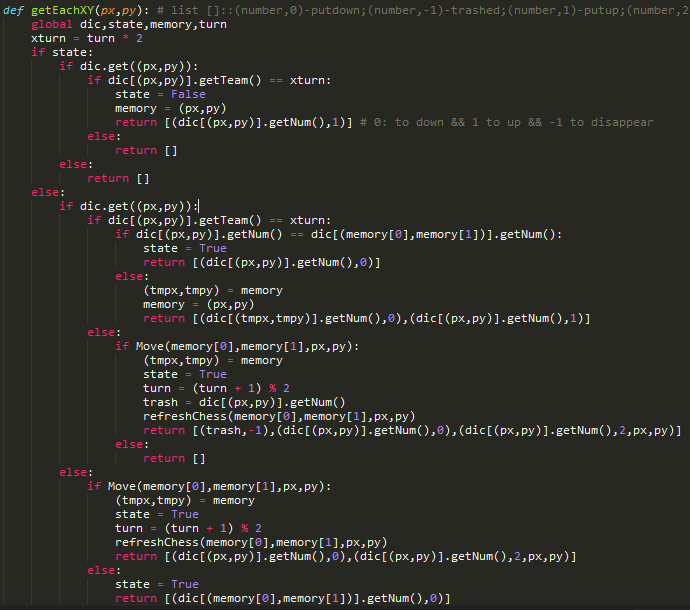
棋盘：将每一个可走点都封装成Unit类，并带有国界和大本营标记，以方便相，士，兵，帅等特殊棋子约束限制或不同区域的不同走法。同时还有标记棋盘上该点是否有子的状态量。用global的list存储，在每次开始或读档时重置更新。



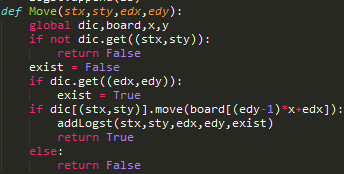
棋子：全部封装成基类Chessman，之后再根据不同种类做出不同的派生类，其中定义主要的move函数，包括各个棋子的独特走法，传入unit类变量，更加方便的进行判断，对每个类实例给予独特编号，作为接口和界面交互。以global的dic字典存储当前棋盘上所有棋子的类实例，开始新游戏或读档时以此刷新界面。



计算走法：根据传入的一对坐标值，返回对应的移动或选中是否成功。这是界面和算法部分交互的关键，因为界面传入的实际是一系列点坐标，需要判断前后相邻坐标值是否能构成一步走法，或是在不断地更新选中待走的棋子。



移动函数：在确定两对点坐标后进行对应棋子移动和消灭的函数，其中涉及到之后的日志记录和字典刷新等等，但本质上还是在调用特定类的move()函数。

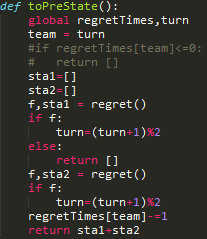


更新棋盘：在有棋子被吃掉或有棋子位置变换后及时覆盖之前的dic记录棋盘的状态，由于python类没有指针，所以每次采用复制操作，如果在C++中可以直接改变指针，但在运行效率上并无明显影响。

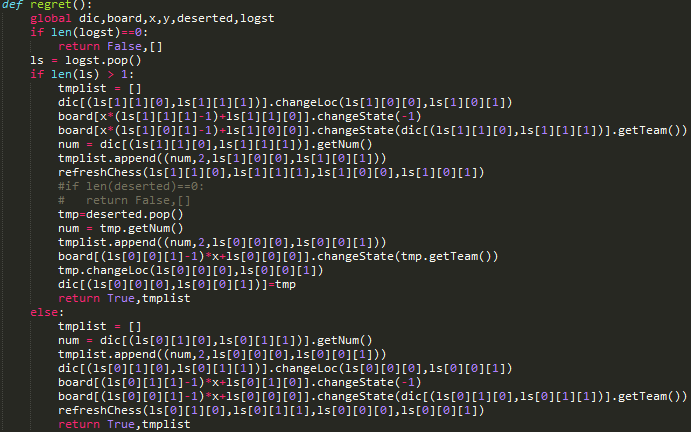


1. 功能添加

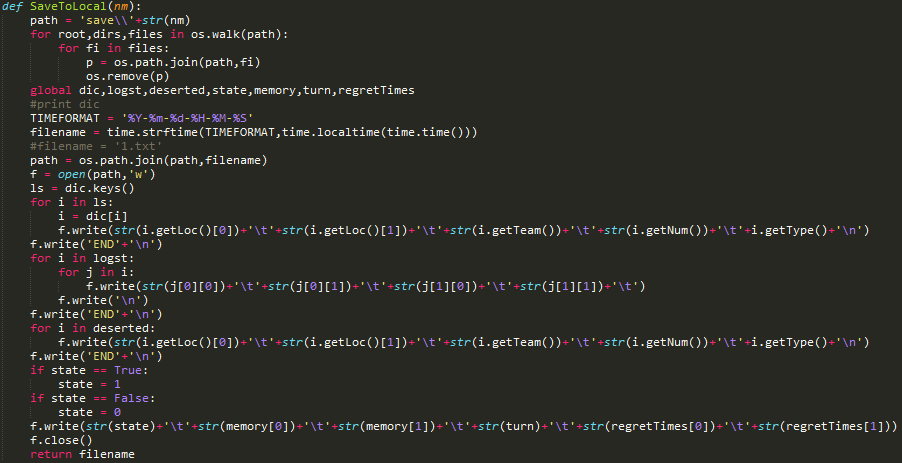
悔棋功能：在每一次成功的移动时都会修改游戏日志，添加操作步骤，由于游戏简单，所以只有两种日志条目：第一种是记录一个棋子从原位到另一空位，第二种是记录一棋子从原位到另一棋子，而被吃棋子进入trash can。在悔棋时，考虑到要回到上一次自己的turn，因此设定一次悔棋，每个人各悔一子。而且还可添加悔棋次数限制，在每一局中规定每位玩家最多的悔棋次数。



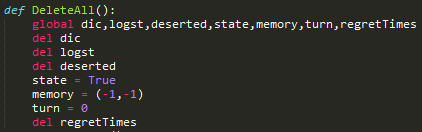
为了完成和界面的交互，每次将悔棋结果以list形式传出，list中包括待悔棋的棋子编号(由实例的getNum()获得)和目标坐标。



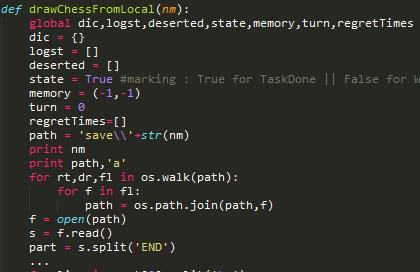
存档功能：提供复盘功能，可以存储游戏至本地，也可从本地文件中读档并继续之前的游戏，之前游戏的所有状态都将被保留还原，包括游戏日志，悔棋步数，红方或黑方走子等。



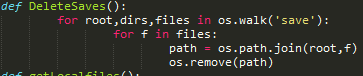
上面是存盘功能，通过向本地文件中写入dic、log等状态量的全部内容实现存档，为了合理的和界面交互，暂定有八个可选存档区域，而文档命名暂不提供输入方式，而以选定的系统时间为准。



上面是清零函数，保证每次读档前清零，不会和当前游戏冲突。



上面是读档函数，只截取一部分，以界面传来的文件编号读取指定文件中的dic、log等信息，重构全局变量。同时提供清除所有存档的功能，方便操作。



为了方便界面传入变量，将save和load函数判断放在一起，以此来进行选择。同时为方便拷入游戏，每个存档所在文件夹都会自动创建，不会引发IOexceptions。

