黑白棋设计文档2

2017.12

罗希月 徐煜凯

2017

## 简介

黑白棋，又叫翻转棋、奥赛罗棋、苹果棋或反棋。游戏通过相互翻转对方的棋子，最后以棋盘上谁的棋子多来判断胜负。它的游戏规则简单，因此上手很容易，但是它的变化又非常复杂。

由于需要每一轮棋子可下的合法位置都需要判断，其编程难度大于五子棋，故我们选择黑白棋作为我们的编程项目。

相应的黑白棋游戏规则在此略过，我们只介绍我们项目的特殊之处。

游戏由一个8×8的棋盘共64个棋格。在我们的棋盘中，出现的标记为：

“. ”代表没有落子

“0”是白棋

“@”是黑棋

“-”是可用位置，如下图所示

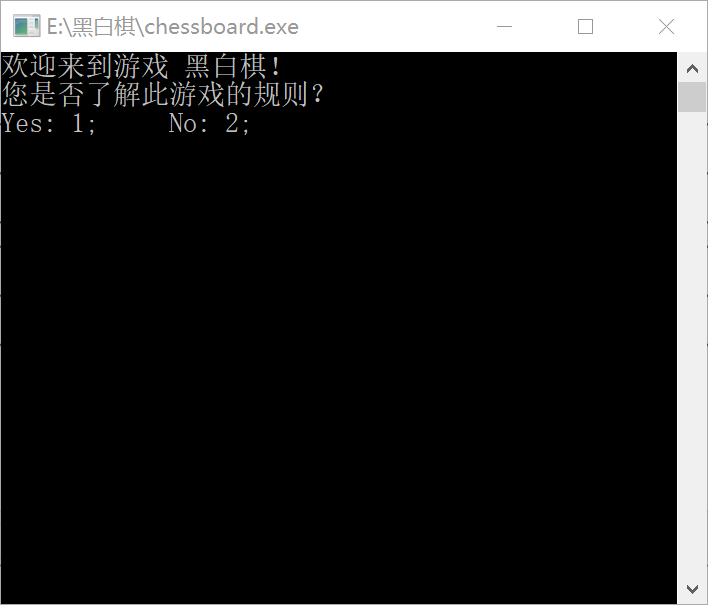
在落子的时候，要求用户输入行数[空格]列数。

例如：[3][空格][4]即下在三行四列。

在我们的游戏运行开始时，也会有相应游戏规则简介。

## 游戏效果

初始页面

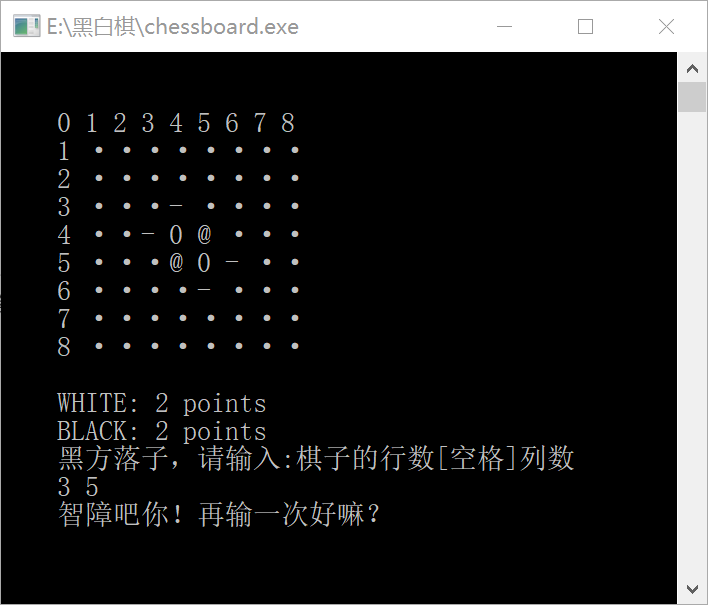


如果选择2，则有游戏规则简介

如果选择1就直接开始游戏

棋盘页面，如图为一个8×8的棋盘

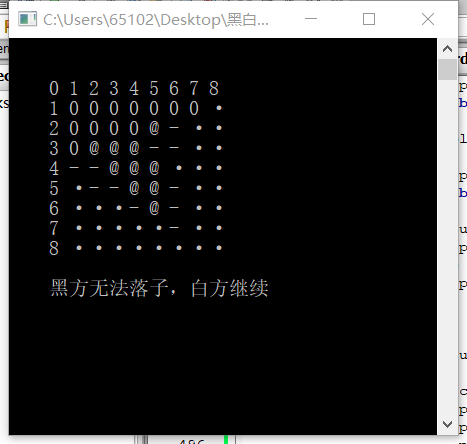
如果输入的数据合法，则会进入下一界面，如果输入不合法，则会要求重新输入



WHITE和BLACK实时记录棋盘上棋子的数量

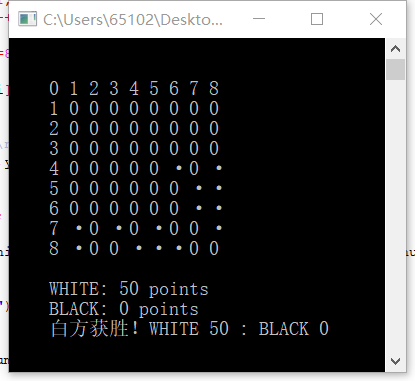
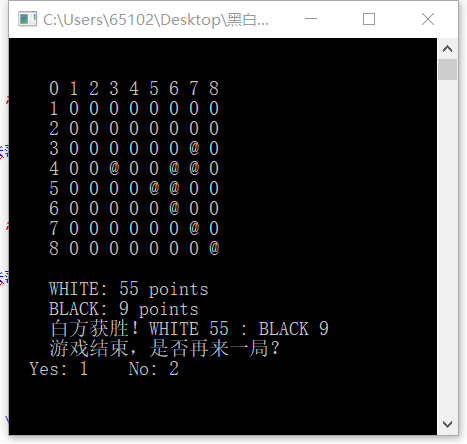
还有提示上一轮对方所下位置

如果有一方没有位置可下，则输出：白方无法落子，黑方继续



如果有一方棋子全无或棋盘被布满，则游戏结束，棋子多的一方获胜

如图为两种获胜方法



## 代码解析

主要功能函数有：

Chessboard：初始化棋盘

Print：每一轮打印棋盘

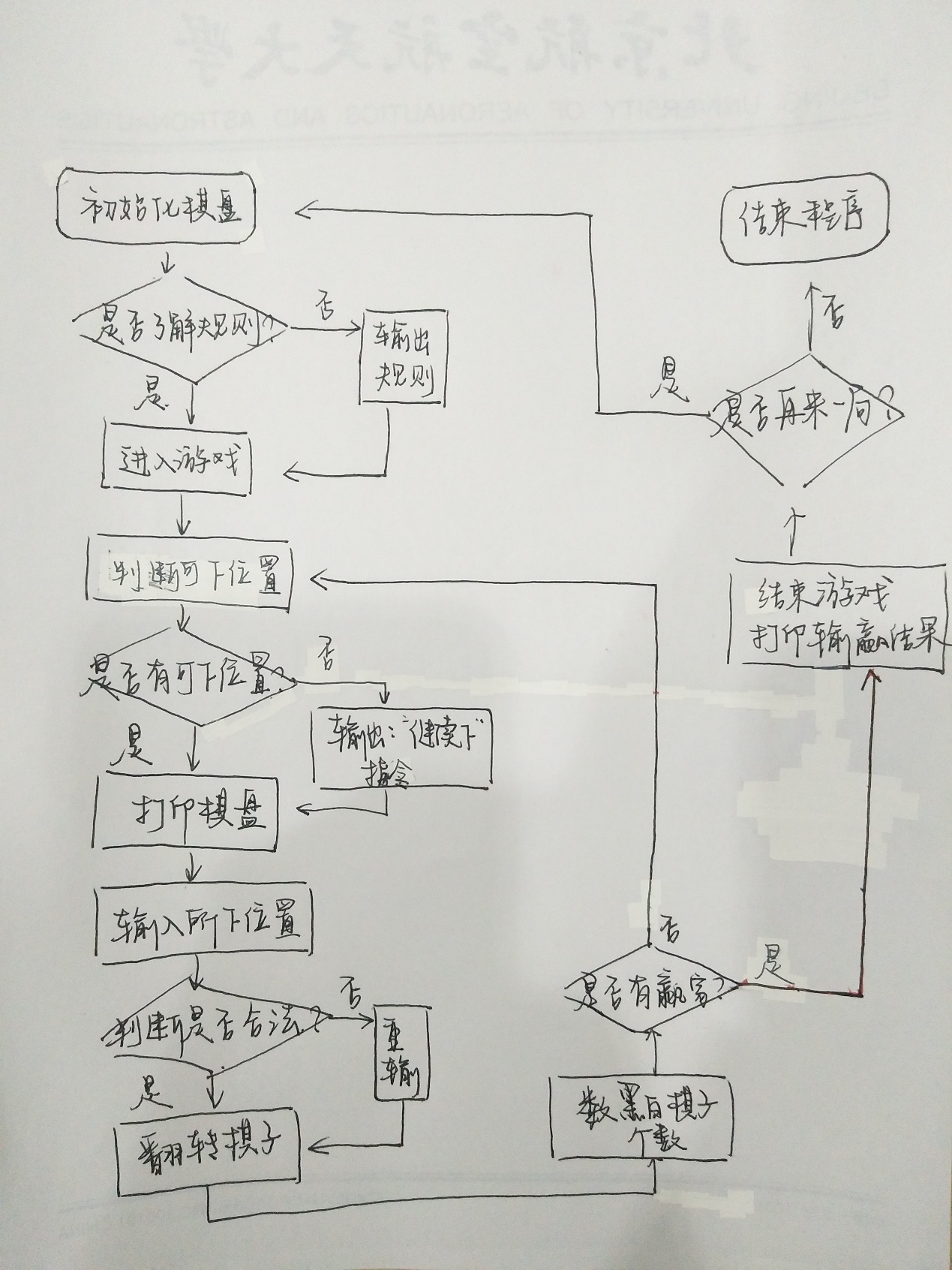
Count：每一轮数棋盘中黑子和白子的数目

Check：每一轮标记可下的合法位置

Flip：每一轮反转棋子

Main：实现循环

循环示意图：



## 成员分工

徐煜凯：完成chessboard、print、count、check函数的编写

罗希月：完成flip、main函数的编写

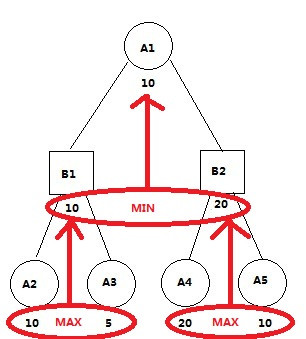
Debug由两人共同完成

## 进化！人机对战！

思想：以简易的极大极小博弈树为基础，采用目前网络上已知的各个棋格的赋值标准。如图：



由于技术原因，并没有对赋值权重进行动态调整，只是将其当作一个定值来评估当前的局势的分数

大致思路如下：穷尽每一个可以下的位置，分析当此处落子后对手让自己得分最小的情况，对手让自己的得分是再下一步自己得分最大的情况（叙述不太到位，详情见图），这就是极大极小算法。

## 出现了！新的函数！

我们添加了两个函数：

Evaluate : 我们用一个函数来计算在给定赋值下当前

局势的分值，寻找自己的棋子乘以权重并相加。

Predict ：顾名思义，对未来局势进行预测，是整

个AI棋手的重点。

## 道歉

非常遗憾地通知看这篇报告的人们，由于时间问题

和debug的难度，人机对战目前只开放人类棋手为

黑子的模式，另一模式将在周日进行进一步调试。(为这个简易版的AI先生我们已经工作了一整个星期六了)

## AI棋手的表现

表现尚可，在与4399黑白棋游戏的困难难度AI对战中能处于平分秋色的地位，在徐煜凯接手后迅速失败，可见我们的AI棋手水平是高于徐煜凯同学的。

## 成员分工

罗希月同学和徐煜凯同学各自独立地编写了一位AI选手，将二者整合后形成了现在看到的AI，两个人功不可没，看到最后的成果，二人都流出了喜悦的泪水。