Matlab实现五子棋人机对战项目报告

11912433 李亦琛

（计算机科学与技术系 指导教师：魏艳）

[摘要] 本文主要介绍了用Matlab实现的五子棋人机对战项目本项目。本项目首先实现了最基本的五子棋界面，包括五子棋初始和落子界面以及游戏结束界面；其次对完整的五子棋游戏逻辑接口进行了封装，如获得合法落子点、判断游戏是否结束及获胜方等；最后使用Minimax极小化极大算法和Alpha-Beta剪枝算法来实现五子棋AI落子。

[关键词] Minimax算法；Alpha-Beta剪枝；五子棋

1. 项目介绍

本项目为单人项目，旨在用Matlab实现的五子棋人机对战。五子棋是一种古老的策略游戏，是两个人在一个棋盘上轮流将棋子下在空位上，先在直线（横、竖、斜）五个棋子连成线的一方获胜。其中五子棋棋盘大小为15×15，先手为黑方，后手为白方，轮流在空白节点上下子，连接5枚同色棋子即为胜利。如果棋盘上所有的节点都下满了棋子，但还没有达成连线胜利条件，则为平局。

同时，该项目在实现基本的五子棋界面、游戏逻辑接口的基础上，进一步添加了使用Minimax极小化极大算法和Alpha-Beta剪枝算法实现的AI落子，使得该项目更加具有挑战性和趣味性。

1. 实现方法

2.1 五子棋界面实现

我采用了Matlab中最基本的Figure来绘画五子棋游戏界面，并通过调用imshow函数。其中五子棋中有各个界面元素，如棋盘、棋子、棋盘线等。这些元素的颜色、位置和尺寸都是根据五子棋规则和界面效果进行调整的，实现的游戏界面如图1所示。

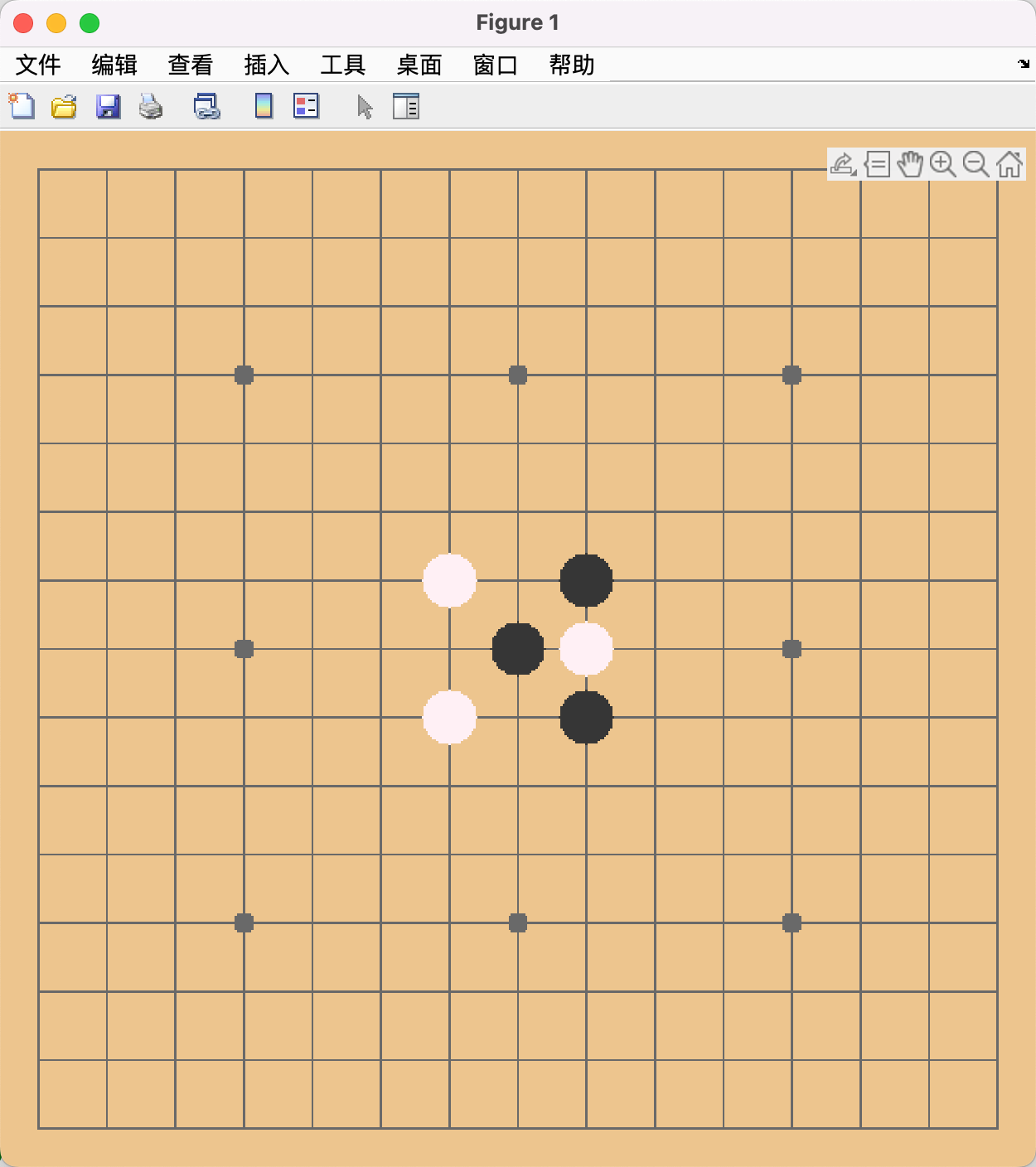


图 1 五子棋基础游戏界面

在棋盘和棋子等游戏元素的绘制完成后，还需要添加游戏结束界面。我通过调用dialog函数创建一个对话框，并使用uicontrol函数将文字和按钮等控件添加到对话框中，实现的游戏结束界面如图2所示。

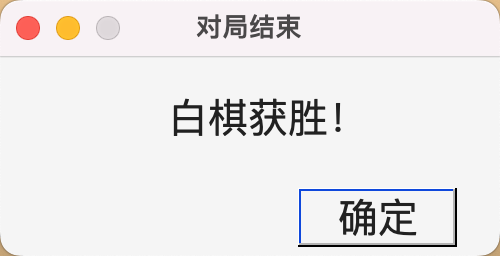


图 2 五子棋游戏结束界面

除此之外，为了实现人方下棋，还需要实现获得鼠标点击位置并在棋盘上成功落子。由于鼠标显示在Matlab库函数内置为十字光标，通过对库函数ginput\_pointer的修改，仅将十字光标改为箭头光标即可。通过调用上述ginput\_pointer函数即可获得鼠标点击位置，并对该坐标点进行判断并落子。

2.2 五子棋游戏逻辑实现

在实现完基本的游戏界面后，还需要实现内部逻辑，包括落子step函数、获得所有合法落子点get\_legal\_actions函数、判断游戏是否结束is\_done函数。

以上所有游戏逻辑函数以及游戏界面的实现即可满足人人对战、人-随机对战。通过这些对战即可对上述实现的所有代码进行测试。

2.3 五子棋AI实现

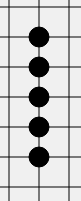
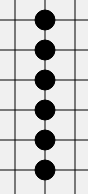
在界面绘制和游戏逻辑完成后，我还实现了五子棋的人机AI。我采用了Minimax极小化极大算法和Alpha-Beta剪枝算法，通过遍历棋盘上的合理位置，对于每个位置，模拟人机在该位置落子后的情形，并根据得分情况，评估该步走法的可行性。最后，我们通过比较各个位置的得分，确定AI应该落子的位置。 此外，我还添加了Alpha-Beta剪枝算法，这个算法是基于Minimax算法的，主要作用是减少无效搜索，并在最短时间内找到最佳的解。该算法在性能方面比Minimax算法更优秀，在游戏AI中使用较为广泛。

我参考了https://blog.csdn.net/livingsu/article/details/104539741的五子棋AI实现，下面是我实现上述算法的一些细节。

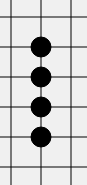
关于评估函数的设计，我对棋盘局面进行评分，即AI先找到所有可落子点，并模拟该落子，并使用该评估函数进行局面评分，如果得分越高则该落子点越好。

为了对局面有更准确的优势评估，对五子棋棋型个数进行了统计，并分为了以下8种。

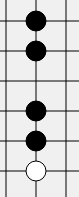
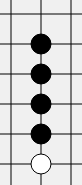
1. 连5或长连

[](https://img-blog.csdnimg.cn/20200227171241363.png)

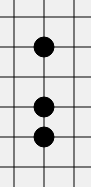
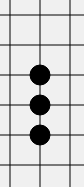
1. 活4（有两个位置可以形成连5）



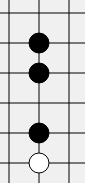
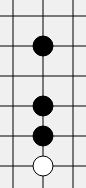
1. 冲4（有一个位置可以形成连5）



1. 活3（走一步可以形成活4）



1. 眠3（走一步可以形成冲4）



1. 活2（走一步可以形成活3）
2. 眠2（走一步可以形成眠3）
3. 活1（走一步可以形成活2）

用一个六元组来表示棋型，通过检查棋盘上所有的六元组来统计局面的得分。每个棋型的权重设计如表1所示。

表 1 各棋型权重表



此外，为了减小搜索分支数，提高搜索效率，采用了局部搜索和静态评价启发的方法。局部搜索不考虑所有空位置，只考虑那些能和棋子产生关系的空位置，因此选择了有子点周围8个方向延申3个长度的位置。静态评价启发是对于Alpha-Beta剪枝算法而言的，越早搜索到较优走法，剪枝就会越早发生。如果对可走节点的评估分数进行简单排序，就可以提高剪枝速度。因此将局部搜索得到的所有可走节点按照评估函数所获得的评分进行排序，最终只选取得分前10名的位置。

1. 实现结果

当Minimax的深度为4时，人机AI落子速度极快且表现优异，和我对打能够不相上下。当深度为6时则落子较慢。

总的来说，本项目实现了基础的五子棋界面和游戏逻辑，还添加了基于Minimax和Alpha-Beta剪枝算法的AI落子。通过该项目，我熟练掌握了Matlab编程的基本方法和技巧，此外，也能够更深入地了解Minimax和Alpha-Beta剪枝算法的基本思想和核心。

1. 参考网站

五子棋界面实现参考https://www.jb51.net/article/163316.htm

五子棋人机AI实现参考https://blog.csdn.net/livingsu/article/details/104539741