

基于 $\alpha-\beta$ 剪枝树算法的安卓五子棋程序设计与实现

宋万洋

(天津理工大学 中环信息学院, 天津 300380)

摘要: 本文设计并研发了一种基于智能算法的安卓五子棋应用程序, 程序中包括两种模式: 玩家对弈和人机对弈, 其中在人机对弈模式中, 程序一方采用 $\alpha-\beta$ 剪枝树算法实现。程序主要由界面显示及控制模块、玩家对弈模块、人机对弈模块和胜负判定模块组成。经过测试, 程序具有较高智能程度, 能够击败大多数业余选手, 并且具有较好的人机交互界面和响应速度, 兼顾了智能性与娱乐性。

关键词: 博弈论; $\alpha-\beta$ 剪枝树算法; 五子棋; 安卓程序

中图分类号: TP317; TP183

文献标识码: A

文章编号: 2096-4706 (2019) 11-0092-03

Design and Implementation of Android Gobang Program Based on $\alpha-\beta$ Pruning Tree Algorithm

SONG Wanyang

(Zhonghuan Information College, Tianjin University of Technology, Tianjin 300380, China)

Abstract: This paper designs and develops an Android Gobang application program based on intelligent algorithm. The program includes two modes: player game and man-machine game. In the man-machine game mode, the program side uses $\alpha-\beta$ pruning tree algorithm. The program is mainly composed of interface display and control module, player game module, man-machine game module and win-loss judgment module. After testing, the program has a high degree of intelligence, can beat most amateur players, and has a better human-computer interaction interface and corresponding speed, taking into account the intelligence and entertainment.

Keywords: game theory; $\alpha-\beta$ pruning tree algorithm; Gobang; Android program

0 引言

人工智能在最近几年得到了飞速的发展, 很多国家甚至把人工智能作为高科技发展计划的重点项目, 在此方面投入了巨大的资源。最优化策略是人工智能研究的一个分支, 而博弈论则是最优化策略的一个代表性算法, 博弈论的思想是穷举当前状态的所有后继情况, 计算最终收益, 以求在可接受的条件内获取尽量高的收益^[1]。这种算法的思想和棋类游戏中对抗的思想类似, 本文选用 $\alpha-\beta$ 剪枝树算法, 设计并开发一款安卓端的五子棋程序, 程序中包括两种模式: 玩家对弈和人机对弈。在玩家对弈中, 黑白棋子由两名玩家控制, 程序负责控制下棋的顺序并判断输赢; 而在人机对弈中, 由玩家选择一方, 另一方由程序自动控制实现对弈。

本文程序选用 Java 开发语言, 在 Android Studio 开发环境中编译开发, 程序中实现了游戏对弈选择、游戏流程控制、游戏胜负判断、基于 $\alpha-\beta$ 剪枝树的自动下棋等功能。程序兼顾了趣味性和智能性, 能够帮助用户丰富业余生活。

1 $\alpha-\beta$ 剪枝树算法原理

在五子棋人机对弈过程中, 考察的重点在于哪一方的“眼

光”更能超前^[2], 双方根据当前具有的完备的局面信息以及规则判断自己和对方落子的所有情况并分别作为节点, 汇总所有的节点, 就可以得到一颗基于层次的博弈树, 落子时根据对博弈树的权重进行搜索, 选择当前情况的最优解, 即为五子棋的落子位置, 但对于五子棋来说, 博弈树的每层节点数量可以达到几十甚至上百, 如果需要搜索连续的几层节点, 那么搜索的数量是指数级的, 搜索时间可能是无法接受的。因此, 需要采用一定的策略缩小搜索范围, 而 $\alpha-\beta$ 剪枝树算法就是一种比较常用的棋类对弈游戏搜索算法^[3]。

$\alpha-\beta$ 剪枝树算法是由递归实现的一种搜索树算法, 在搜索过程中主要关注两个值, 即 α 与 β , α 值用来表示在搜索过程中对自己有利的值中的最好值, β 则为最不利于对方的值, 每一个节点都可以返回一个与 $\alpha-\beta$ 相关的值, 以此来判定是否继续搜索^[4]。在整个搜索过程中, MAX 方被称为 α , MIN 方被称为 β , 双方都拥有最优的值, 在开始搜时, α 被赋初值为 $-\infty$, β 被赋初值为 $+\infty$, 而在搜索过程中, 节点 MAX 使 α 的值持续增长, 节点 MIN 则使 β 持续递减, 因此形成了一个 $[\alpha, \beta]$ 区域, 该区域被叫做窗口。该窗口中的值用来表示该节点的子节点的值的范围, 缩小窗口值的过程就是向下搜索的过程, 最终在这个窗口中会留下最优值。当节点 MAX 得到该节点的子节点返回值大于 β 的值或者节点 MIN 得到该节点的子节点返回值小

收稿日期: 2019-05-13

于 α 值，就会执行减枝。

如图 1 所示，在 MAX 层，假如当前层中的节点值是已搜索过的所有值中最大的一个，如果发现下一层 (MIN) 中的其中一个节点可能会产生比刚才搜索到的最大值还要小，即可以把该节点直接剪掉。反之，如果在 MIN 层，找到了已搜索过的所有值中最小的一个，发现下一层 (MAX) 会产生比刚才搜索到的最小值还要大，即可把该节点直接剪掉。即双方都不会走出让对方更有利的一步。

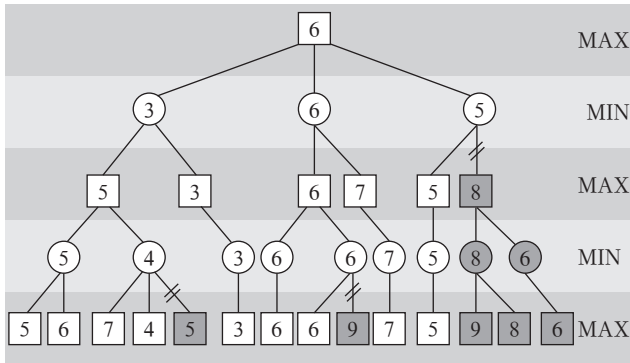


图 1 $\alpha-\beta$ 剪枝示意图

在五子棋游戏中，如果某一个落点的结果不大于 α ，那么该落点就可以被视为是很差的落点，该落点将被抛弃；如果某一个落点的结果不小于 β ，那么就作废整个节点，因为对方不希望出现这个局面，对方会有更好的落点避免这个局面出现，在判定过程中发现了一个不小于 β 的落点的结果，对该结果及其后续进行剪枝操作，不计算后续所有节点。但是当某一个落点的结果介于 α 与 β 之间的时候，那么该落点就可以被一方考虑。在搜索过程中， α 会持续增加以至于能反映新的情况，但是可能会出现计算超时的情况，因此还需添加其他限定条件，以保证搜索能够及时停止，获得一个可以接受的局部最优值，在本文中，考虑到安卓手机的性能，采取限定最大搜索 4 层的策略，来实现五子棋游戏中的人机对弈的效果。

2 程序设计与实现

五子棋应用程序需要具有良好的人机交互体验，界面要美观，下棋计算时间要快，能供用户选择进行玩家对弈或者人机对弈功能，并且能够自动判定胜负并记录得分^[5]。根据功能需求，将该程序划分成如图 2 所示的几个模块，程序的主要人机交互界面划分如图 3 所示。

页面显示及主控模块：修改配置文件 main.xml 插入背景图片，通过 TextView 字符串组件来显示菜单与结束的字符串，用 View.OnClickListener 接口中的 setOnClickListener 来监听是否有触摸，如果有触摸则运行相应的状态。程序运行过程中，自定义 onDraw 函数来绘制棋盘以及棋子，确保程序界面正常显示和触摸控制。

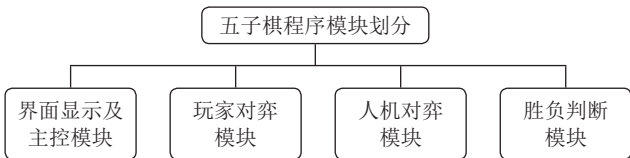


图 2 程序模块划分

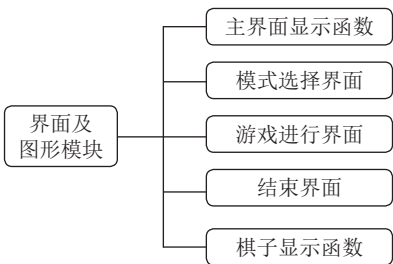


图 3 人机交互界面

游戏模式选择模块：因为本程序可以由用户选择玩家对弈还是人机对弈两种模式，所以在进入页面中提供游戏模式选择按钮，用户点击按钮时，名为 mMenuTv 的菜单的字符串组件将获取到的选择状态传递给界面初始化函数 showPopupWindow () 以及 pop_choose_gobang 配置文件来执行相应的操作。

游戏胜负判断模块：通过继承 View 并重载返回值为布尔类型的 onTouchEvent 函数来判断该触碰是否可以落子，如果对应的位置为空的话，则在该位置记录落子信息，并且判断游戏胜负。胜负判断主要通过获取该点并通过遍历它的周围来判断横向、纵向或者斜向是否有五子相连，Android 中的 View 横纵坐标与正常的 x、y 轴不同，它是正常的坐标轴顺时针旋转 90 度，所以判断五子相连的时候，横向则是 y 轴加减、纵向才是 x 轴加减。如果遍历出有五个字相连则调用 GameOver 函数中的 Toast.makeText 来显示游戏结束，否则继续游戏。胜负判断的流程图如图 4 所示。

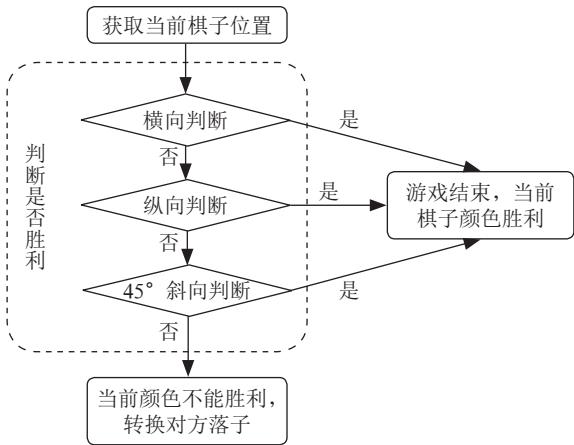


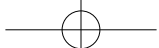
图 4 判断游戏胜负流程图

3 结论

本文实现了一款 Android 平台下的五子棋游戏程序。程序中可以实现玩家对弈和人机对弈的功能，其中人机对弈功能采用 4 层的 $\alpha-\beta$ 剪枝树搜索算法实现，经过测试，程序的智能水平已经能够达到业余中高级水平，并且程序运行速度较快，落子时间以及胜负判定时间都可以接受。程序基于 Android 6.0 开发，能够兼容市面上绝大多数手机，兼顾了趣味性和智能性。

参考文献：

[1] 董慧颖，王杨．多种搜索算法的五子棋博（下转 97 页）



在图 3 架构下通过分别对系统并发操作和单独对数据库进行测试发现,数据库中的数据量越大,系统的读功能越强,查询时间越快,而写时间稍微慢,原因是配置的双主备数据库架构的影响。

表 3 各服务器 IP 分配和软件安装表

主机名	IP 地址	任务角色	数据库
Node1	192.168.216.11 Vip192.168.216.200	Nginx_Master Keepalived	
Node2	192.168.216.12 Vip192.168.216.200	Nginx_backup Keepalived	
Node3	192.168.216.13	Tomcat1	
Node4	192.168.216.14	Tomcat2	
Node5	192.168.216.15	Mycat1 Tomcat3	MySQL1_Master
Node6	192.168.216.16	Mycat2 Tomcat4	MySQL1_backup
Node7	192.168.216.17		MySQL2_Master
Node8	192.168.216.18		MySQL2_backup

4 结 论

图 3 的 Nginx+Keepalived+Mycat 后端服务器集群架构,既解决了 Web 系统的单点故障问题,也解决了高并发问题,适于中小型高并发系统,但对于淘宝天猫这样的系统来说,这样的架构还是不适合大型电商系统,由于上下行数据都是通过 Nginx 服务器传输,当数据量很大的情况下,Nginx 服务器是整个架构的数据传输瓶颈,如何解决该问题,作者将在后续研究中再讨论。

参考文献:

[1] 百度百科. 服务器集群 [EB/OL].<https://baike.baidu.com/item/服务器集群/5309783>, 2019-05-21.
[2] Nginx. nginx documentation [EB/OL].<http://nginx.org/en/docs/>, 2019-03-31.

[3] Keepalived. What is Keepalived? [EB/OL].<http://www.keepalived.org/index.html>, 2019-05-21.
[4] 百度百科. 双机热备 [EB/OL].<https://baike.baidu.com/item/双机热备/2394182>, 2019-05-21.
[5] MyCAT 官网. 什么是 MYCAT [EB/OL].<http://www.mycat.io>, 2019-03-31.
[6] 百度百科. 单点故障 [EB/OL].<https://baike.baidu.com/item/单点故障/3570893>, 2019-05-21.
作者简介: 刘金秀 (1967.01-), 女, 汉族, 湖南湘潭人, 本科, 硕士, 高级工程师, 研究方向: 软件工程; 陈怡华 (1994.10-), 女, 汉族, 广东汕头人, 管理人员, 研究方向: 教学管理; 谷长乐 (1992.02-), 男, 汉族, 湖南衡阳人, 管理人员, 研究方向: 思想政治教育。

(上接 93 页) 弈算法研究 [J]. 沈阳理工大学学报, 2017, 36 (2): 39-43+83.
[2] 孙世文. 五子棋人工智能算法实现研究 [J]. 中国新通信, 2018, 20 (23): 143.
[3] 周洋, 邓莉, 谢煜. 一种五子棋博弈算法的分析 [J]. 现代计算机 (专业版), 2017 (10): 8-10.

[4] 毛丽民, 卢振利, 刘叔军, 等. 五子棋对弈机器人移动平台的研究 [J]. 微特电机, 2017, 45 (1): 9-14.
[5] 刘洋. 点格棋博弈中 UCT 算法的研究与实现 [D]. 合肥: 安徽大学, 2016.
作者简介: 宋万洋 (1991-), 男, 汉族, 天津人, 助教, 硕士研究生, 研究方向: 软件工程、数据挖掘。