

摘 要

本文主要介绍了基于 Android studio的五子棋游戏开发过程。该游戏的搜索算法采用了极大极小值搜索算法并使用α-β剪枝算法增加算法的效率，数据库使用sqlite实现。

五子棋游戏主要包含两大功能模块，游戏模块与回放模块。其中游戏模块是游戏的核心功能模块，采用人机对战以及单机人人对战两种游戏模式。主要包括人机对战功能、胜负提示功能、判断落子优势区功能、棋局保存功能等；回放模块主要实现了读取棋局功能、重现棋局功能、回放控制功能。在人机对战模式中的人机对战功能中使用了极大极小值搜索算法，该算法经过α-β剪枝优化能够更好地实现计算机判定优势落子位置的功能。

本游戏的研发为用户提供了一种在时间碎片中进行娱乐的方式，实现了一般的五子棋游戏的基本功能，并且具有棋局保存与回放的特色功能。

关键词：五子棋；α-β剪枝算法；深度搜索；人工智能

**Abstract**

This paper mainly introduces the development process of Gobang game based on Android studio. The search algorithm of the game adopts the minimax search algorithm and uses α-β Pruning algorithm increases the efficiency of the algorithm, and the database is implemented by SQLite.

Gobang game mainly includes two functional modules, game module and playback module. Among them, the game module is the core function module of the game, which adopts two game modes: man-machine battle and single machine everyone battle. It mainly includes man-machine combat function, victory and defeat prompt function, judgment of falling son advantage area function, chess game saving function, etc; The playback module mainly realizes the functions of reading chess games, reproducing chess games and playback control. The minimax search algorithm is used in the man-machine combat function in the man-machine combat mode, the algorithm has been α-β Pruning optimization can better realize the function of computer to determine the position of dominant seeds.

The research and development of this game provides users with a way of entertainment in time fragments, realizes the basic functions of general Gobang games, and has the characteristic functions of chess game saving and playback.

**Key words**：gobang；α-β Pruning algorithm；deep search

目 录

**[第1章 概 述 1](#_Toc101827917)**

[1.1五子棋游戏开发的目标和意义 1](#_Toc101827918)

[1.2国内外相关课题的研究 1](#_Toc101827919)

[1.3五子棋游戏功能简介 2](#_Toc101827920)

**[第2章 五子棋游戏的基础理论与相关技术 4](#_Toc101827921)**

[2.1 五子棋游戏的基础理论 4](#_Toc101827922)

[2.2 五子棋游戏相关技术 5](#_Toc101827923)

[2.3 棋局实现 6](#_Toc101827924)

[2.4 评估函数及相关算法 6](#_Toc101827925)

**[第3章 五子棋游戏总体设计 8](#_Toc101827926)**

[3.1 系统体系结构设计 8](#_Toc101827927)

[3.2 系统功能模块设计 9](#_Toc101827928)

[3.3 游戏数据库设计 10](#_Toc101827929)

**[第4章 五子棋游戏详细设计与实现 11](#_Toc101827930)**

[4.1 初始界面 11](#_Toc101827931)

[4.2 游戏模块 12](#_Toc101827932)

[4.3 棋局回放模块 17](#_Toc101827933)

[4.4 五子棋游戏的UI设计与实现 23](#_Toc101827934)

**[第5章 五子棋游戏核心算法实现 25](#_Toc101827935)**

[5.1 基本思想 25](#_Toc101827936)

[5.2 评估函数 25](#_Toc101827937)

[5.3 常用核心算法及优化 28](#_Toc101827938)

**[第6章 测 试 39](#_Toc101827939)**

**[结 论 41](#_Toc101827940)**

**[参考文献 42](#_Toc101827941)**

**[致 谢 43](#_Toc101827942)**

# 第1章 概 述

## 五子棋游戏开发的目标和意义

随着计算机及互联网在经济社会中的运用日益广泛，计算机的发展也越来越迅速，作用也越来越重要。计算机的发展催生了一门新兴的学科—人工智能。人工智能(ArtificialIntelligence)，人工智能是随着人类认知的深入和科学技术的进步而衍生出来的一项技术，主要思想是通过一些复杂的算法规则把人类的智慧赋予机器。１９５６年，人工智能的概念首次被提出。当时由于计算机性能的不足和理论基础的缺失，人工智能技术没能得到迅速的发展。

直到近年来，计算机的计算能力突破瓶颈，各种各样的深度学习和机器学习算法研究也在如雨后春笋般不断出现。伴随着这些AI技术也到达了一个全新的领域，并且已经在游戏，服务业等众多领域得到了广泛的运用，并获得社会的认同。而计算机博弈，这一项作为重要的研究部分在AI发展历程中不可或缺，一直伴随着AI的发展而日益完善。该部分在实验与总结方面易于实现，可以通过某个算法在计算机博弈中的表现总结快速平行推广至其他的行业或者应用领域。

五子棋便是一款博弈游戏。在棋类运动中，五子棋是一个重要的组成部分，又具有相对简单的规则，易于上手也易于开展深入研究。在游戏中与AI斗智斗勇，运筹帷幄，也可以让人对AI的了解更加深入，深刻。

而手机作为新时代的移动智能终端，拥有着巨量的用户，且都会随身携带。所以，将一些常见的游戏或是软件从计算机平台移植到手机平台上，一定会受到用户的好评，取得用户的簇拥。

本系统开发的意义在于通过移动智能终端的便携性，推广五子棋对弈游戏来发扬与继承优秀传统文化，并通过此方式激发广大人民群众对AI的兴趣，使人们对AI的接受度更高，也能让更多的人参与到AI的研究当中去。

## 国内外相关课题的研究

无论是中国还是西方世界，计算机博弈的研究都广泛而深入。上世纪五十年代，AI概念横空出世，但在这之前就有人想到了设计AI与棋手对弈。国内外许多知名学者和知名科研机构都涉足过这方面的研究，至此七十余年，已经取得了相当震撼的成就：1912年，西班牙人莱昂纳多·托雷斯·奎韦多的国际象棋自动机，被认为是世界上第一台真的象棋机器。两年后，托雷斯·奎韦多在巴黎索邦大学机器实验室展示了他的机电式象棋机。最终游戏机器能够用车和国王,将死人类对手的国王。1997年5月11日， IBM 深蓝电脑在六局比赛后击败了国际象棋大师卡斯帕罗夫，获得历史性的胜利：IBM深蓝赢了两局，卡斯帕罗夫赢了一局，三局平局。这证明了人工智能在某些情况下有不弱于人脑的表现。

2017年5月，AlphaGo与当时世界排名第一的选手柯洁展开了角逐。在这三场比赛中，AlphaGo以3比0获胜。此对局的比赛结果不仅仅在象棋领域，更在世界范围内引起了轰动。

而中国也在人机对弈方面取得了良好的成绩。2006年，“浪潮杯”首届中国象棋人机大战中，5位中国象棋特级大师最终败在超级计算机浪潮天梭手下。中国人发明的这项充满东方智慧的模拟战争游戏，被中国超级计算机独占鳌头。2021年7月，在交大校园内中国象棋AI“小原”与象棋特级大师孙勇征进行人机对抗实战并以孙大师投子认输宣告了“小原”的胜利。

对围棋、中国象棋、桥牌、扑克等许多种其它种类游戏博弈的研究也正在进行中。 在人机对弈历史上虽然互有胜负，但由无数人心血打磨而成的计算机总是略占上风，并且时代在发展，伴随着计算机机能的进步，人们对算法的研究更加深入，人类与计算机在棋类运动中的差距只会越来越明显。总而言之，人机对弈的研究任重而道远。

## 五子棋游戏功能简介

五子棋游戏是一款比较经典的黑白棋游戏，主要包含两大功能：游戏功能和游戏回放功能。其中游戏功能能够实现基本的游戏竞技，对于独自游玩的用户，五子棋游戏提供人机对战功能并可由用户选择难度进行游戏。对于想要与同伴对弈的用户，五子棋游戏提供单机人人对战功能。单机人人对战与人机对战共有的功能介绍如下：

1、棋局保存功能：对局结束后，用户可选择将此次对局保存，以便日后回顾。

2、胜负提示功能：系统根据棋局信息提示用户胜利者。此功能的难点在于不需要像评估函数中对全局进行判断，需要使用启发式的评估方法。

3、棋盘切换功能：用户可选择主题皮肤，现共有两种皮肤可供选择：经典、自然。

人机对战功能特有功能：判断落子优势区功能：用户无思路是可点击提示，系统提示需要注意的落子点。此功能的特点在于基于优势区域进行落子搜索对比在整个棋盘进行落子搜索节约了大量的资源。

棋局回放功能就是基于数据库存储信息实现的对用户选择存储的棋局进行回放的功能。功能介绍如下：

1、棋局回放功能：通过用户选择存储的棋局信息进行回放，此功能的难点在于需要记录棋局中的落子顺序与在数据库中对数组信息的存储与读取。

2、残局重现功能：在回放过程中，随时可以根据残局进行游戏。现仅提供重现单机人人对战。

3、回放暂停功能：在回放过程中，可以进行暂停。

在人机对战功能中，要求可以实现不同的难度，在实现时对AI的深度搜索函数的层数进行限制即可实现。要求可以实现人机对战的响应在1~10秒内，防止太快影响游戏观感或者太慢使用户等待时间过长。通过wAIt函数将AI的响应时间固定大于1s，通过判断落子优势区功能将AI搜索时间控制在10s范围内。另外还有棋局回放功能中，需要实现回放按照时间跳动，防止瞬间所有棋子落完用户无法通过回放进行回顾学习等。通过计时器设置间隔实现此功能，此功能的特点时绑定进度条并不以时间进行作为进度条的进度，而是使用棋局落子的数量作为进度，防止棋局回放之间的长度差异造成的进度条不合理跳动。

# 第2章 五子棋游戏的基础理论与相关技术

## 2.1 五子棋游戏的基础理论

### 2.1.1 公平性

五子棋游戏是一种竞技性极强的游戏。对于五子棋这种竞技性强的游戏，在竞技过程中，最基本的问题就是五子棋的公平性，首先我们需要介绍公平的定义。具体阐述如下：

定义一：明确的不公平性：如果一方胜出已经证明了，这场比赛可以说是明确不公平。例如：五子棋的一般规则就是明确不公平。

定义二：单调不公平性：如果一方已经证明不一定会赢，但也不能证明对方一定赢，这场比赛可以说是单调不公平。例如：K 子棋中 Connect(m,n,k，p，p)，可 用 策 略 盗 用 论 点 (Strategy-stealingargl) 证 明 自 方 必 然 不 会 必 胜 ；因 此 ，Connect(6，1，1)，Connect(7，l，1)，Connect(6，2，2)等皆为单调不公平的。然而，因为 Connect(8，l，1)已被证明双方平手，所以不是单调不公平的。

定义三：经验不公平性：如果在下棋的实践经验发现有一方高于另一方，这场比赛可以说是经验的不公平。例如：在早期五子棋和连珠棋的一般规则，已被普遍认定为的黑色必胜，这就是当时的对经验不公平的描述。

定义四：潜在的公平性：如果不能证明或者论证明确不公平，单调不公正，经验不公平，那么这个游戏可以作为一种潜在的公平。在此基础上进行定义，一个在此时为潜在公平的游戏，不一定在未来继续公平，如果一个游戏可以持续公平的时间越长，那么公平的机率就越高[1]。

### 2.1.2 五子棋的基本规则

五子棋与围棋一样被称为“诘棋”，对弈双方不断设下陷阱，在打断对方的必胜路线的同时将对手引入死局当中。在经过以上定义的阐述之后，很容易可以得出，没有其他规则限制的简单规则五子棋是一种不公平的游戏。因为在先手落子后，后手落子也只能追平落子数量。保持领先者一直是先手，这对于后手是不公平的。但是由于五子棋的不公平性，会出现几种无法中断的“先手必胜”套路，严重影响了五子棋游戏的竞技性。现在流行的在正规比赛中采用的规则如下：

采用固定起手或自由开局，当采用固定起手开局时，开局形状均以"星"和“月"命名。采用固定起手办法的比赛均采用26种开局，包括斜指开局的长星局、峡月局、恒星局、水月局、流星局、云月局、浦月局、岚月局、银月局、明星局、斜月局、名月局、彗星局、寒星局、溪月局、疏星局、花月局、残月局、雨月局、金星局、松月局、丘月局、新月局、瑞星局、山月局、游星局。在此种固定起手开局的情况下，通常对先手棋手有禁手规则：禁手一、先手棋手不允许一步形成多个活三。禁手二、先手棋手不允许一步形成多个活四或冲四。禁手三、先手棋手不允许通过多于五个子（如六连）的落子方式获得胜利。简称为三三禁手，四四禁手，长连禁手。后手棋手无禁手规则[2]。

当采用自由落子开局时，一般为偶数对局数，对弈双方轮流作为先手。自由落子开局一般不使用禁手规则。考虑到大众对于五子棋的认知程度更倾向于休闲娱乐，采用自由落子开局的方式实现游戏。

## 2.2 五子棋游戏相关技术

### 2.2.1 集成开发平台

1、IDEA（IntelliJIDEA）

IDEA 全称IntelliJIDEA，是java语言开发的集成环境，IntelliJ在业界被公认为最好的java开发工具之一，尤其在智能代码助手、代码自动提示、重构、J2EE支持、各类版本工具(git、svn、github等)、JUnit、CVS整合、代码分析、创新的 GUI设计等方面的功能可以说是超常的。

2、Android Studio

Android Studio是一个Android开发环境，由Google发布，运行于Linux,Mac OS X, windows 三大平台，基于IntelliJ IDEA.类似Eclipse ADT，Android Studio提供了集成的Android开发工具用于开发和调试，也是Google官方推荐安卓程序开发工具。

### 2.2.2 SQLLite数据库

SQLite是遵守ACID的关系型数据库管理系统，它包含在一个相对小的C库中。它是D.RichardHipp建立的公有领域项目[3]。

ACID，指数据库事务正确执行的四个基本要素的缩写。其中包含：原子性(Atomicity）、一致性（Consistency）、隔离性（Isolation）、持久性（Durability）。

SQLite作为一款轻型的数据库，它的设计目标是嵌入式的，而且目前已经在很多嵌入式产品中使用了它，它占用资源非常的低，在嵌入式设备中，可能只需要几百K的内存就够了。比起Mysql、PostgreSQL这两款开源世界著名的数据库管理系统来讲，它的处理速度比他们都快。

区别于常见的客户-服务器范例，SQLite引擎不是个程序与之通信的独立进程，而是连接到程序中成为它的一个主要部分。所以主要的通信协议是在编程语言内的直接API调用。这在消耗总量、延迟时间和整体简单性上有积极的作用[4]。

### 2.2.3 java

Java是一门面向对象编程语言，不仅吸收了C++语言的各种优点，还摒弃了C++里难以理解的多继承、指针等概念，因此Java语言具有功能强大和简单易用两个特征。Java语言作为静态面向对象编程语言的代表，极好地实现了面向对象理论，允许程序员以优雅的思维方式进行复杂的编程[5]。

Java编程语言具有多线程和面向对象的基本特点。利用该特性，程序员在编写程序时可以有一个更加宽松的编写环境，减轻程序员的工作压力，全面提高程序员的工作效率，所开发的软件更符合相关标准[6]。

## 2.3 棋局实现

### 2.3.1 位棋盘

在最早的国际象棋中采用的是上世纪六十年代由苏联科研人员发明出来的比特棋盘，也就是位棋盘。国际象棋棋盘的长和宽的长度和一个字节的长度相等，所以可以用一个64位的数字来表示一个棋子的位置。这个技术一般给用于64位架构的计算机时可以只使用十二个位棋盘就可以完全表示出一个国际象棋棋盘上的所有信息。由于计划五子棋在手机上实现，棋盘的设定与常见的五子棋不尽相同，并且需要记录落子的顺序以便于实现残局重现功能，本方法不予选择。

### 2.3.2 数组棋盘

对于一个平面的棋盘来说，在程序中将其重现的常用方法是使用二维数组。数组的两个维度，一个表示棋盘的长度，一个表示棋盘的宽度，而每个元素对应棋盘上的一个位置，元素的索引正好对应棋子在棋盘中所处的位置。对于五子棋，考虑到棋盘与手机屏幕的比例，选择了14\*20的棋盘样式。为了记录落子顺序，将无子的位置设为0，黑子的位置则是奇数，白字的位置为偶数。

## 2.4 评估函数及相关算法

### 2.4.1 评估函数

评估函数是通过给定或者计算机可读取的数据库对局面进行数字化的转型，并且以得到的量化的数据作为评估标准来评价当前局面是优势或是劣势。在各种不同的棋类博弈程序中，评估函数都是极其重要的一部分。可以说棋类博弈程序的智能水平很大程度上取决于评估函数的水平。

### 2.4.2 搜索技术

五子棋游戏主要采用的搜索技术有两种，分别为极大极小搜索算法与α-β剪枝算法。

极大极小搜索算法：在博弈中，双方采取的策略都是尽可能让局面对我方有利对敌方不利。极大极小搜索就是一种基于当前状态推测出使我方最有利而对方最不利的行动。

α-β剪枝算法：在博弈中思考步数的增加会让计算机思考的可能性指数增长，当数量过大时计算机无法快速运算。所以需要采取策略减少不必要思考的可能性，就是α-β剪枝算法

### 2.4.3 棋盘转换与落子识别

机器必须通过棋盘转换与落子识别才能理解棋盘信息。让计算机理解五子棋的基础就是将五子棋的棋盘信息转换为计算机可理解的形式，并且需要制定规则让AI知晓何处可落子，落子的逻辑怎样符合规则。

### 2.4.4 算法优化

对于性能较弱的移动智能终端平台，必须通过对算法进行优化才能让程序良好地运行。只简单采用极大极小搜索算法和α-β剪枝算法作为基础做出一个五子棋AI，它的运算能力一定不足以支持指数增长的落子可能，所以需要对算法进行优化，针对五子棋规则进行修正。

以上部分均在第五章算法的实现详细讲述。

# 第3章 五子棋游戏总体设计

## 3.1 系统体系结构设计

五子棋游戏系统根据核心游戏系统的结构作为基础加以修改进行设计。游戏的架构一般分为核心层（Core），引擎层（Engine），游戏类型层（Game Genre），游戏层（Game），而游戏核心架构也可以用分层的方式来设计。游戏核心架构如图3-1所示。

1、信息层：信息层担任存储外界信息（大脑部分）与感知外界信息（眼睛部分）并更新存储信息的功能。可以理解为僵尸的眼睛与记忆，与决策层、请求层构成AI的头部

2、决策层：决策层的作用是根据信息层中的世界信息决策AI的行为，利用程序规定AI的行为，并控制决策的时机。

3、请求层：请求之间的重要程度不同，请求层的作用就是处理不同重要程度的请求执行的顺序。

4、行为层：通过程序完成上层传入的请求。

5、运动层：根据行为层完成的请求匹配相对于的动画或是游戏表现。

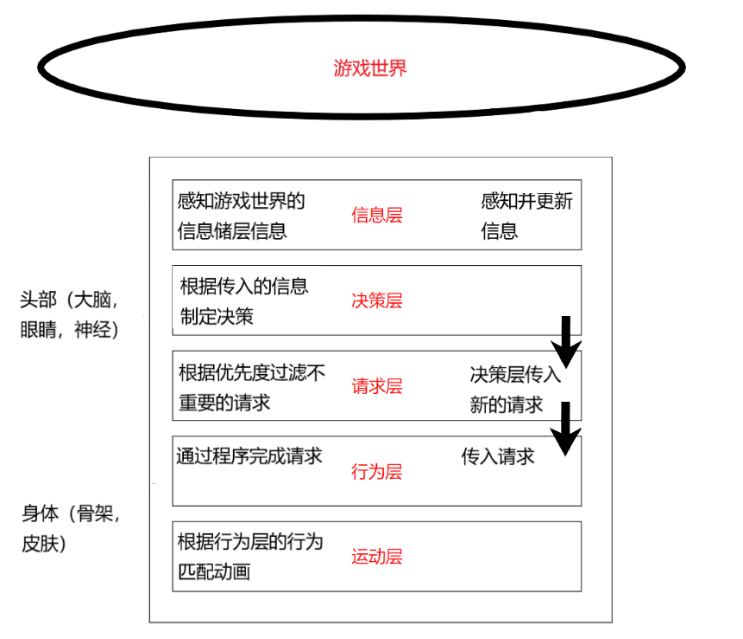


图 3-1 核心游戏系统架构

在信息层，存放游戏中AI所需要或者展示给玩家的数据，主要为两部分，首先是知识池，也就是游戏世界的数据存储。在五子棋游戏中，游戏相关的数据就是棋盘的数据，本系统采用数组棋盘作为知识池进行游戏数据的存储。第二部分是感知器，是更新游戏数据的方法，在游戏中，每次落子都会使棋盘数据刷新。

决策层则是AI的行为决策，在五子棋游戏中，AI的决策只有思考自己的最优落子位置，所以此层存放的是AI对落子位置判定的算法。

请求层的作用是过滤决策层传递发出的请求，本系统针对此层进行了部分修改。当人类棋手正在进行落子演算，需要五秒钟，这段时间内AI的决策都不能被接受，因为人类棋手方尚未落子。对AI棋手亦是如此。所以本系统以落子前另一色棋子不允许落子的方式代替请求层的作用。

行为层是获得决策层的请求后做出的行为，也就是落子行为。在虚拟棋盘上将对应的位置改变数值。

运动层针对的是具有动画效果的游戏，与行为同步进行的运动，在可见的棋盘上调用函数将对应位置放置一颗棋子。

## 3.2 系统功能模块设计

根据五子棋游戏实现的基本功能，将游戏划分为两大功能模块进行设计，分别为游戏模块和游戏回放模块。具体的功能模块如下：

1、游戏模块

游戏模块主要是玩家用户进行游玩的模块。

1. 人机对战功能：用户进入游戏后根据提示选择难度即可开始人机对战。AI执黑子先行。
2. 单机人人对战功能：用户进入游戏后根据提示选择人人对战即可开始人人对战。在五子棋规则上，人机对战与人人对战相同无禁手规则。
3. 棋局保存功能：对局结束后，用户可选择将此次对局保存，输入需要保存的信息作为棋局的名字即可存储以便日后回顾。
4. 胜负提示功能：系统根据棋局信息提示用户胜利者。
5. 棋盘切换功能：用户可选择主题皮肤，现共有两种皮肤可供选择：经典、自然。
6. 对局回放模块

对局回放模块主要是玩家用户在对局结束后回顾的模块。

1. 棋局回放功能：用户通过点击棋局的名字（由用户输入的名字与棋盘基本信息组成）选择棋局，系统通过棋局信息进行回放。
2. 残局重现功能：用户在回放过程中，随时可以根据残局进行游戏。现仅提供重现单机人人对战。
3. 回放暂停功能：在回放过程中，用户可以使用此功能让回放暂停。
4. 回放继续功能：在回放过程中，在使用了暂停功能后，用户使用继续功能回放便可继续播放。
5. 快速回放功能：在回放过程中，用户使用此功能即可快进回放迅速完成棋局回放。

该系统的功能模块如图3-2功能模块图所示。

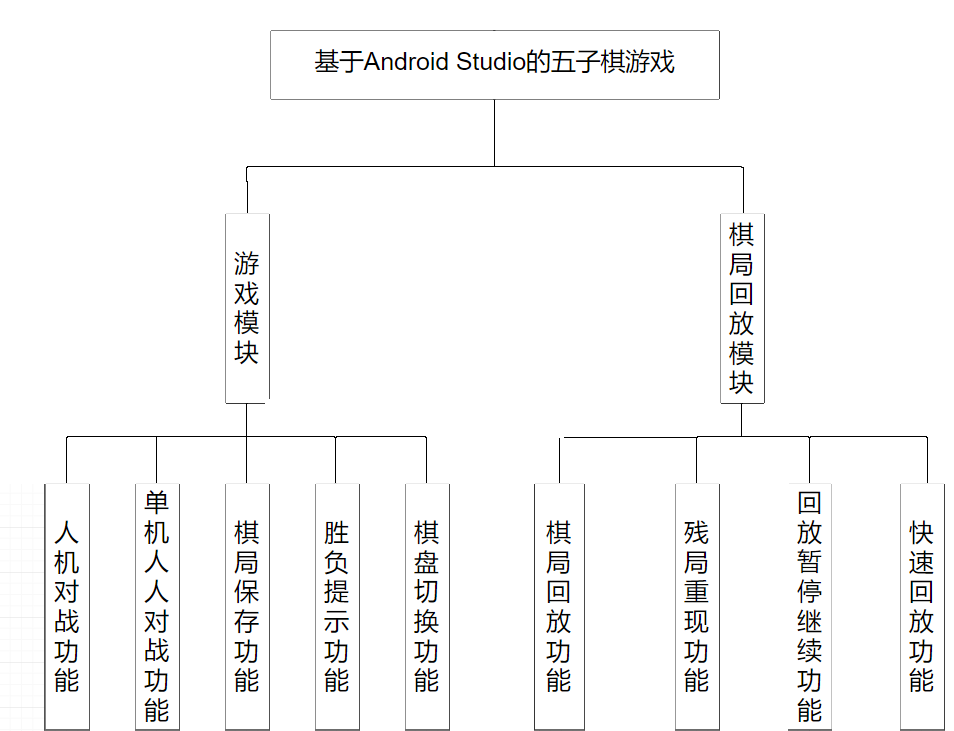


图 3-2 功能模块图

## 3.3 游戏数据库设计

实现数据库的目的是为了完成本游戏的特色功能棋局回放。

通过继承SQLiteOpenHelper类重写onCreate方法进行建表。五子棋游戏需要实现利用表存储数组棋局的信息，并且每一张表表示一个棋局。所以数据库中表的数量取决于存储的棋局数量。表结构如表3-1棋局信息表例所示。

每张表所拥有的数据数固定为280个。

表 3-1棋局信息表例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 允许空 | 主键 | 功能描述 |
| i | int | 11 | 否 | 是 | 数组棋盘的一级下标 |
| j | Int | 11 | 是 | 否 | 数组棋盘的二级下标 |
| Chuchar | Int | 11 | 是 | 否 | 数组棋盘的落子顺序 |

# 第4章 五子棋游戏详细设计与实现

## 4.1 初始界面

一款优秀的游戏，必定会拥有契合其风格的UI设计，只有在这方面做出努力，才能吸引用户进行游玩，用户在游玩时最先感受到的UI设计是初始界面的UI设计。本节详细介绍字体效果与按钮。

### 4.1.1 字体效果

字体效果是UI设计中重要的一环，根据五子棋的黑白二色，设计字体为黑白异色，并以特殊字体凸显。以下为字体效果的实现方式：

SpannableStringBuilder是一种字符串类型，与String相近，二者的区别在于SpannableStringBuilder可以使用其方法setSpan（）进行对字体效果的改动，实现各种不同风格字体的展示。五子棋的对弈双方分别持黑子与白子，为了凸显主题，并且给予游玩者新的体验。将五子棋游戏命名为《黑白五子棋》，并通过SpannableStringBuilder的setSpan（）方法对黑白二字添加颜色修改效果，使黑字为黑色，白字为白色。通过layout的setBackgroundColor方法对背景色进行修改将背景色置为灰色以此凸显标题。通过setSpan方法对五子棋三字进行拉伸填充标题。最终整体效果如图4-1整体效果图所示。



图 4-1 整体效果图

### 4.1.2 按钮的设计与实现

相较于字体，按钮更加适合展示独特的UI风格。按钮的设计思路与字体相同，利用黑白色的棋子作为按钮，体现游戏主题。以下为按钮效果的实现方式：

使用selector可以对控件的背景进行改变。在layout界面中添加各式控件时，可以通过设置控件的background属性进行控件皮肤的添加。在五子棋游戏中，五子棋的对弈双方所持的黑子与白子与按钮形式相似，实际上是调用同一个类绘制的皮肤。作为按钮来使用不仅符合主题，还能增加趣味性。在开始界面的layout begin.xml中添加四个按钮。在drawable/xxx.xml中配置android的selector，通过shape android:shape="oval，solid android:color="#fffdfd"画出白子的形状为圆形并且在begin.Xml中添加背景将二者连接，最终按钮效果如下图4-2所示。

图 4-2 按钮效果图

## 4.2 游戏模块

游戏模块是五子棋游戏的核心功能模块，是用户最看重的部分。本小节介绍主要游戏流程，对战功能的详细实现的部分与核心算法重合，在后文第五章中讲述。

### 4.2.1 人机对战功能

本五子棋实现了人机对战功能。在五子棋游戏中，用户通常会选择人机对战功能进行游玩，落子迅速不用长考。而人机对战功能的AI游戏水平决定了游戏的有趣程度。所以对此功能需要花费更多的时间进行编写并进行算法的优化。具体实现方式在5.3节中详细介绍。具体人机对战功能实现的步骤如下：

1、AI获取可落子区域：通过bestpoint函数判定有意义的落子区域并通过算法选取其中得分最高的十个点作为备选项。

2、AI进行计算得出落子位置：通过AI函数，得出此时最佳的落子位置。

3、AI落子：与人类落子不同，AI落子不需要进行确认，直接在棋盘上生成棋子即可。

4、人类落子：与前文中人人对战中的棋手落子步骤相同。

5、显示获胜者：利用安卓手机的土司提示用户胜利者。

6、是否存储对局：弹出确认按钮询问用户是否存储对局，点击取消清空棋盘，可重新进行对弈。点击确认进入到对局信息存储界面。

7、输入信息：用户输入棋局的名字，与系统自动生成的对局信息代码拼接作为棋局的名字存储。

8、游戏结束：代表本局游戏结束，关闭存储信息界面，返回到清空的棋盘。

人机对战的流程如图 4-3人机对战流程图所示。

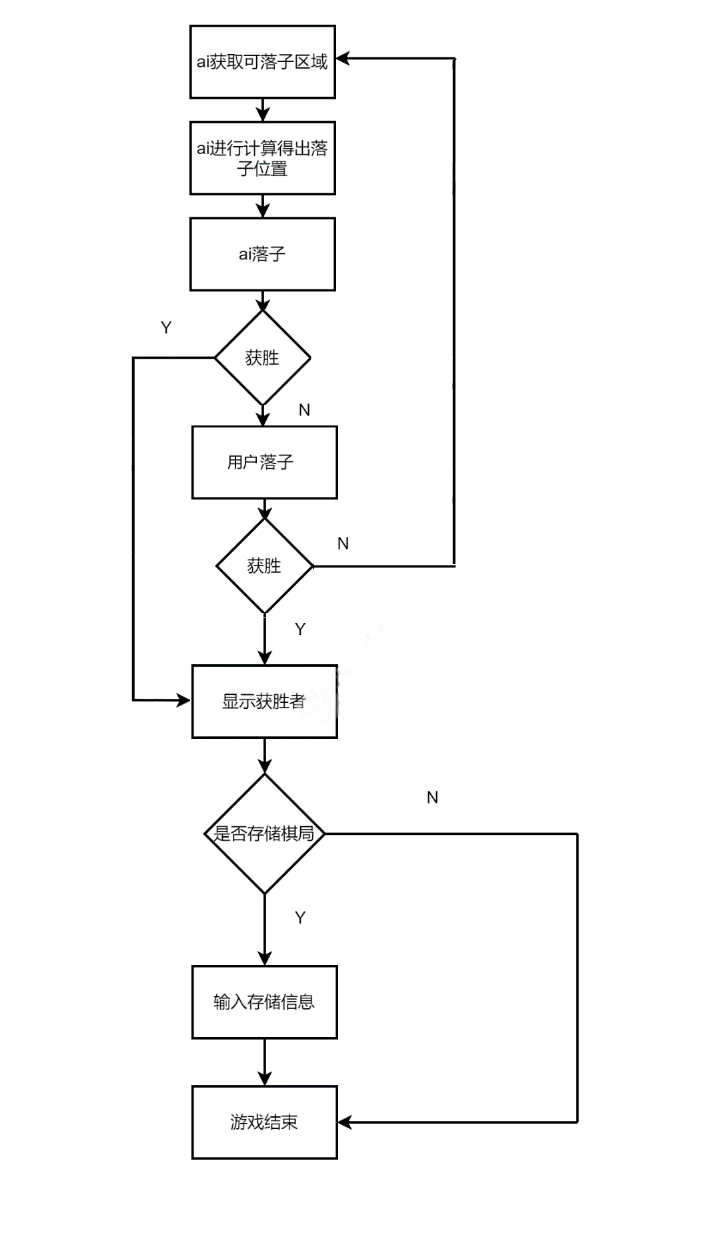


图 4- 3人机对战流程图

### 4.2.2 人人对战功能

本五子棋实现了人人对战功能，本游戏的人人对战功能流程按照现实中进行五子棋对战的流程，设计出在移动智能终端上进行人人对战的流程。在启动程序后，玩家可选择棋盘的皮肤，进入游戏后可以选择是否进行人机对战，选择否则进行单机人人对战。

1、选择游戏模式与棋盘皮肤：棋盘皮肤有两种皮肤进行选择，通过点击游戏开始界面中的“主题切换”按钮进行切换。

2、黑棋棋手落子/白棋棋手落子：游戏开始时，由黑棋棋手进行落子，落子需要确认，确认后落子权转给白棋棋手，由白棋棋手进行落子。

3、获胜：每次棋手落子进行一次判定，若有棋手达成五连则判定某方棋手获胜，显示获胜者。

4、显示获胜者：利用安卓手机的土司提示用户胜利者。

5、是否存储对局：弹出确认按钮询问用户是否存储对局，点击取消清空棋盘，可重新进行对弈。点击确认进入到对局信息存储界面。

6、输入信息：用户输入棋局的名字，与系统自动生成的对局信息代码拼接作为棋局的名字存储。

7、游戏结束：代表本局游戏结束，关闭存储信息界面，返回到清空的棋盘。

人人对战的流程如图 4-4 人人对战流程图所示。

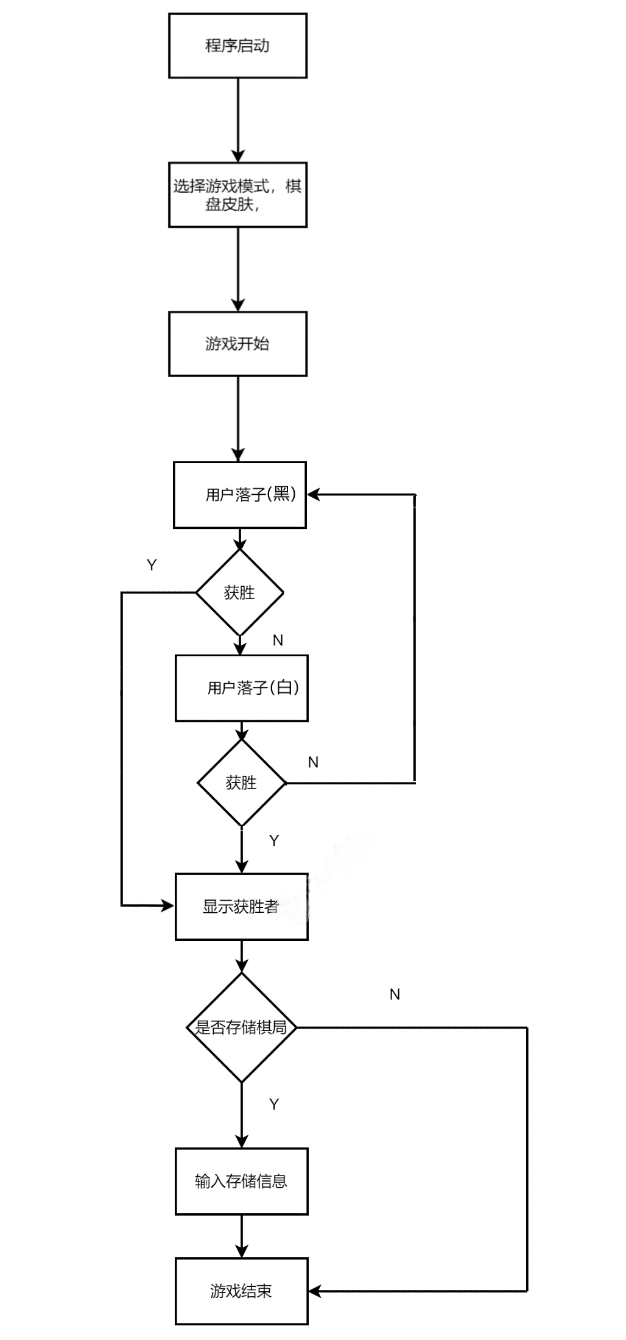


图 4- 4人人对战流程图

### 4.2.4 胜负提示功能

五子棋游戏的胜负逻辑并不复杂，当白棋或者黑棋在水平，垂直，左斜，右斜四方向上连成五连时，相应颜色的棋手就获得胜利并结束游戏弹出提示。所以可以采用启发式的逻辑进行实现。具体实现方式如下：

每一次落子都代表一次局面的刷新，所以将判断游戏是否结束的判断放于Cbtn的点击事件中，落子一次进行一次判断，通过五子棋游戏性质可以推测若连成五子，最后落下的一子一定是连成的五子中的最后一子。所以只需要对落子处的四个方向进行判断是否存在五连即可。形成五连后游戏结束，弹出提示框让用户选择是否保存本次对局，选择确定进入保存界面，点击取消退出游戏界面。游戏结束时弹出的提示界面如图4-6游戏结束所示。



图 4-6 游戏结束

### 4.2.5 棋盘切换功能

棋盘的切换功能可以让用户对棋盘皮肤进行切换。点击“主题切换”按钮即可对棋盘棋盘皮肤进行更换，现共有两款皮肤：默认，自然。

棋盘的初始皮肤使用canvas画布与paint画笔实现，具体实现方式在后文中详细介绍。默认棋盘效果如下图4-7棋盘效果（默认）图所示。

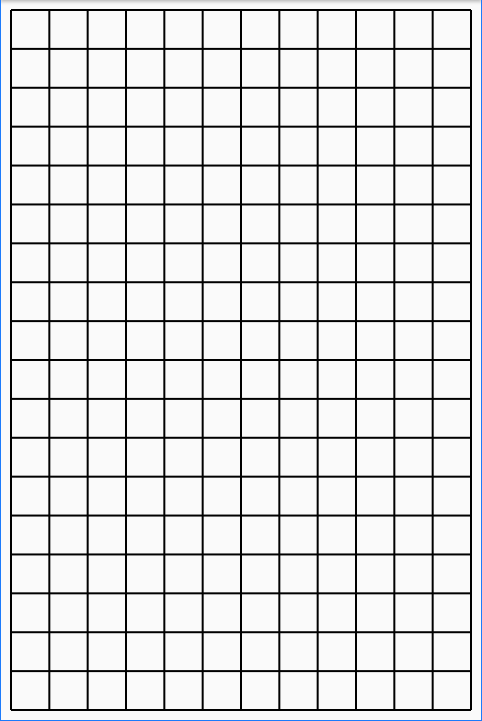


图 4-7棋盘效果图（默认）

棋盘的背景只是白底黑线有些单调，为了增加客户满意度，本五子棋游戏添加了切换棋盘背景的功能。通过getResources()方法获取到layout test.xml的背景，通过读取intent.getStringExtra传递进来的背景信息的值，若为“natrual”（英语单词自然的错拼）则将背景置换为存储在软件当中的图片。通过点击开始界面的主题设置即可切换。当进行主题设置切换后，历史对局中呈现的对局效果与游戏中的对局效果都会以切换后的效果展示。切换后的棋盘效果如图4-8棋盘效果图（自然）所示。

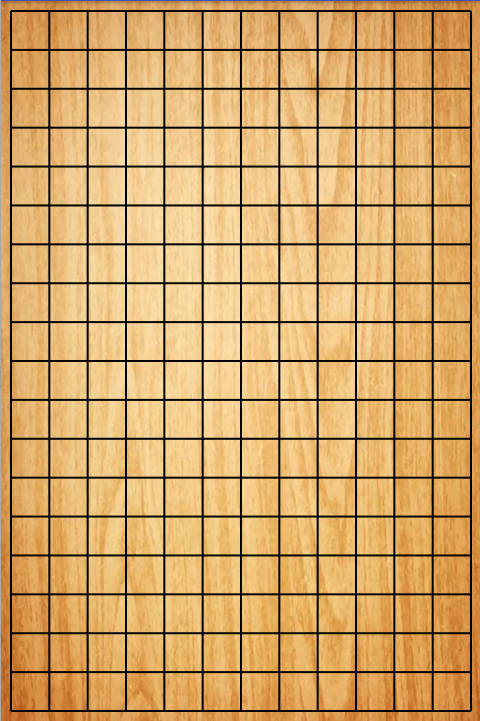


图 4-8棋盘效果图（自然）

## 4.3 棋局回放模块

棋局回放是本文所介绍的五子棋游戏的特色之一，可以将进行的游戏对局保存并在回放模块中重现，并且可以中断回放重现落子。运行程序后，点击历史对局按钮，并选择想要读取的对局（储存于手机数据库），系统会开始以0.5s落一子的速度落子，底部有进度条显示棋局进度，点击进度条可以唤醒/隐藏控制按钮，完成按钮可以跳过过程，直接把所有棋子落完，暂停按钮可以暂停落子，点击后变为继续按钮。重下按钮可根据当前落子位置生成游戏对局重新对弈。具体棋局回放功能实现的步骤如下：

1、选择历史棋局：点击游戏开始界面中文本显示为“历史棋局”的按钮，进入棋局选择界面。

2、选择想要读取的棋局：系统将list将所有存储的棋局的名字显示，用户点击想要观看的棋局名字进行回放。

3、系统按照棋局自动落子：list隐藏，绘制棋盘并根据棋局信息顺序进行落子，一秒一子。

4、点击进度条唤醒按钮：在回放界面，按钮由点击进度条进行唤醒，重复点击进度条即可隐藏按钮。

5、点击完成按钮：跳过顺序落子，直接完成棋局。

6、点击暂停按钮：落子暂停。暂停按钮转换为继续按钮。

7、点击继续按钮：落子继续。继续按钮转换为暂停按钮。

8、点击重下按钮：跳转到游戏界面，并将已经完成回放的部分落子信息进行传递。

9、创建以当前棋局信息为底的棋局对战：根据传入的落子信息进行落子构建残局游戏，使用此流程实现的游戏只能进行人人对战。

10、棋局完成（回放结束）：所有落子都已按照顺序落完，将重下与暂停按钮删除。

棋局回放的流程如图4-9棋局回放流程图所示。

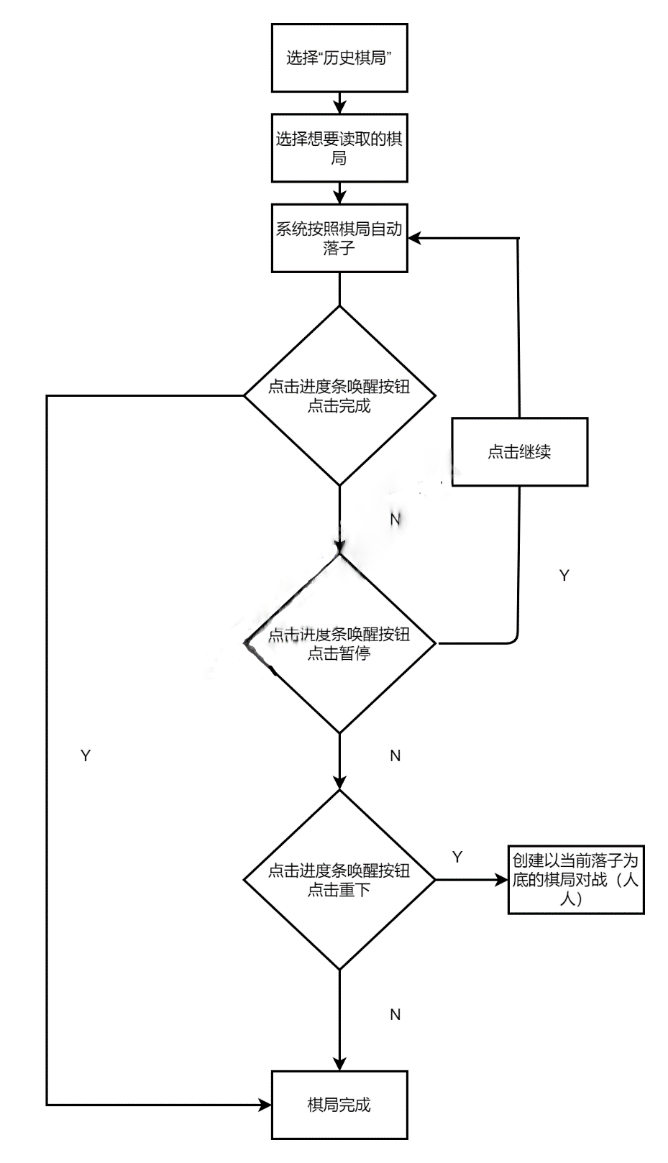


图 4-9棋局回放流程图

### 4.3.1 棋局回放功能

棋局回放功能是本五子棋游戏的特色之一，也是棋局回放模块的主要功能。如果说棋局的保存实质是通过数据库对二维数组进行存储。那棋局的读取的实质就是通过数据库对二维数组进行读取。为了五子棋的观赏性，读取的方式添加了时钟，随时间读取。点击历史对局按钮，跳转至棋局选择页面。点击需要回放的棋局名字即可开始回放。棋局选择页面如图4-10棋局列表所示。

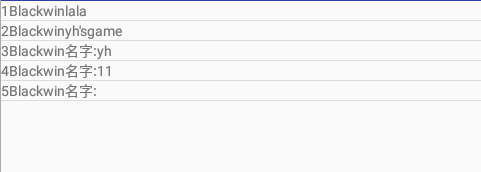


图 4-10 棋局列表

棋局回放功能的具体实现方式如下：

使用ListView控件，使用SQL helper中重写的接口方法搜索存储的表名并按顺序列出。对ListView 设置点击监听，当点击某个棋局时根据点击的表名调用SQL helper的方法搜索出表内存储的数组值并调用绘制棋盘的类进行绘制，并开始棋局回放。

在棋局回放界面，棋子会按照一秒一步的速度进行落子，进度条同步滚动。点击进度条可以唤醒四个操作按钮，分别为完成，返回，重下，暂停。按钮的功能在后文中详细介绍。按钮唤醒后与按钮唤醒前界面的对比如图4-11棋局回放\_唤醒按钮的对比所示。

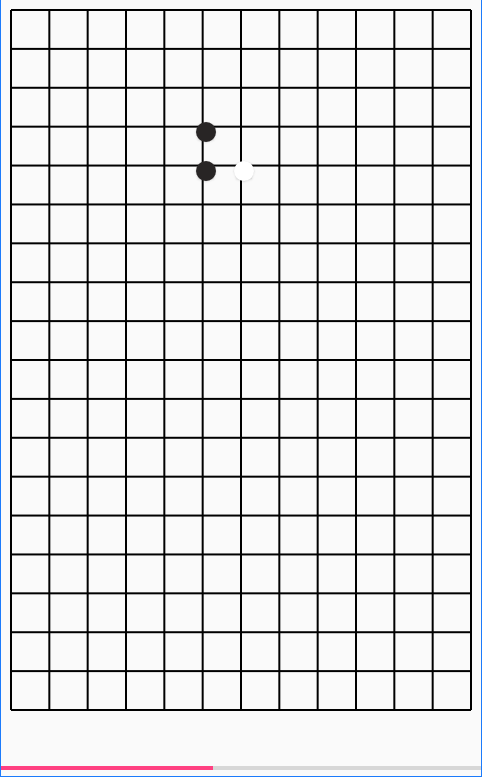
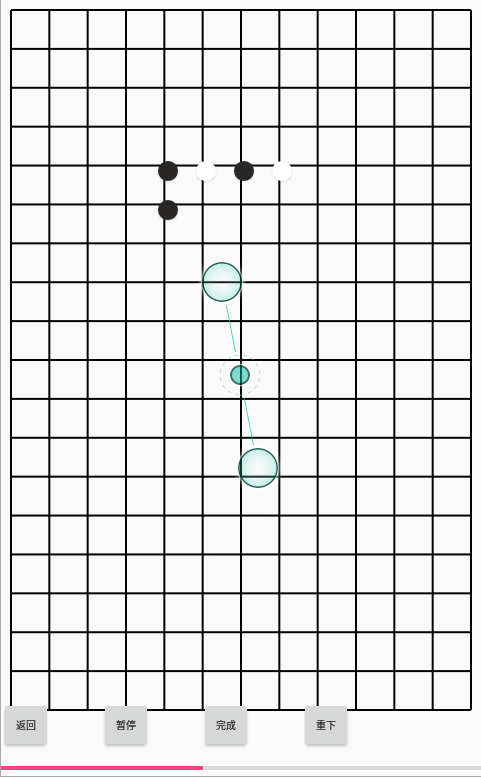
 

图 4-11 棋局回放\_唤醒按钮的对比

唤醒功能的具体实现方式如下：

在棋局回放所属的xml中放置一个计时器Chronometer ch和一个进度条ProgressBar progressBar。

在回放方面，对ch调用setBase方法，设置使用的秒数为当前时间加上落子的总数，在时间跳动监听中做一个循环，循环中在数组中搜索值等于time的元素，找到时在棋盘上的与数组相同下标出进行落子。Time初值为1，每次经历一次循环就加一， 这样就可以实现按照顺序回放落子。并且在开始回放（调用read（）方法）后开始启动，

在进度条方面，在计时器ch的时间跳动监听中使用progressBar.incrementProgressBy(progressBar.getMax() / process)，每次时间跳动，进度条progressBar就增加100/落子总数的百分比，就实现了进度条progressBar表示落子进度。

在功能按钮方面：在progressBar的点击监听中用代码添加四个按钮于progressBar上方平行排列并各自赋予显示文字，id，以及点击事件。点击进度条progressBar时，将标识use置为true，当use为true时，创建按钮，当use为false时，调用方法remove掉四个按钮。按钮功能的具体实现方式如下：

1、暂停/继续按钮：停止/继续计时器的跳动

2、完成按钮：修改计时器的时间，进度条直接进度增加至100%，设置循环直接读取棋局数组的值并落子。

3、返回按钮：调用end方法关闭此页面。

4、重下按钮：通过intent传递数组值、棋局回放标识、Curcher（已落子数）并跳转至游戏界面，。

### 4.3.2 残局重现功能

残局重现功能为本五子棋游戏的特色之一，可以将回放中的某个时刻所展现的棋局作为残局重现到棋盘上，并以此残局开始人人对战。 此功能的具体实现方式如下：

通过intent传递数组值、棋局回放标识。Curcher（已落子数）并跳转至游戏界面。游戏界面在识别棋局回放标识调用read（）函数进行落子。Read函数作用为根据传入的数组值，在棋盘的相应位置生成棋子按钮。生成棋子的方法为Wchucher和Bchucher方法，根据传入的下标在相应的位置生成黑棋或者白棋。棋局回放的界面与棋局重现功能使用后跳转的游戏界面效果如图4-12棋局回放\_重下的对比所示。

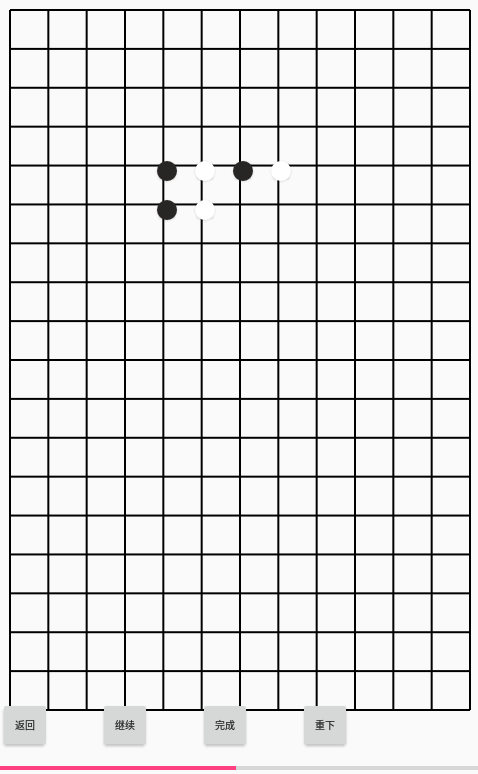
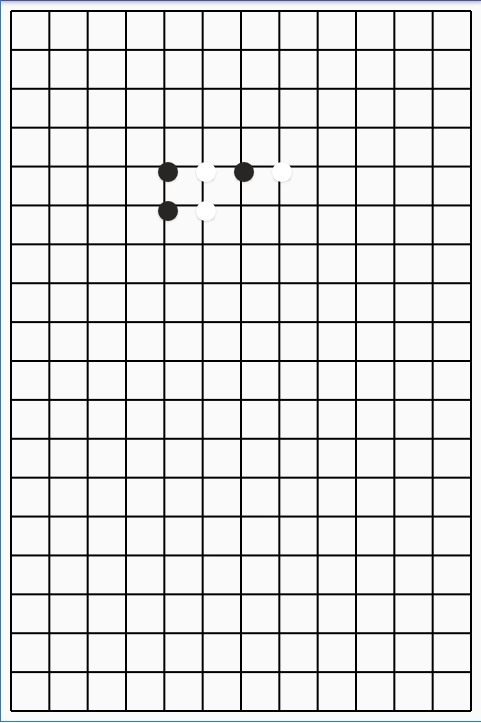
 

图 4- 12棋局回放\_重下的对比

### 4.2.3 棋局保存功能

在对局结束后，用户可以选择是否对本次游戏产生的棋局进行保存。保存的棋局可以在棋局回放功能中进行回顾。以下为棋局保存功能的具体实现方式：

五子棋的棋局进行数字化转换为二维数组，棋局的保存实质是通过数据库对二维数组进行存储。游戏结束后，通过AlertDialog设置提示框提示用户是否继续存储，选择存储则跳转至save并通过intent传递棋盘数据。Intent不支持直接传递数组，采用循环intent.putExtra ("array"+ii,chuchart[ii])的方式进行传递，跳转后的棋局页面如图4-13棋局存储所示。

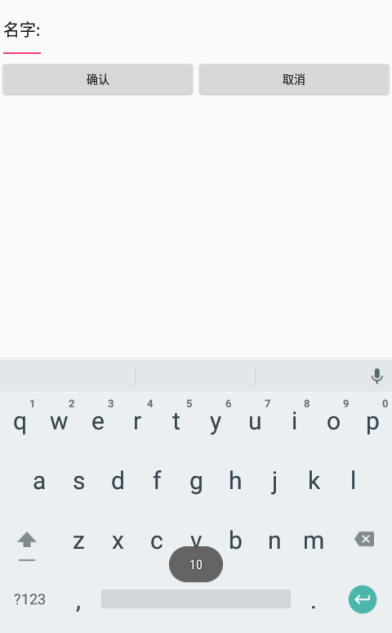


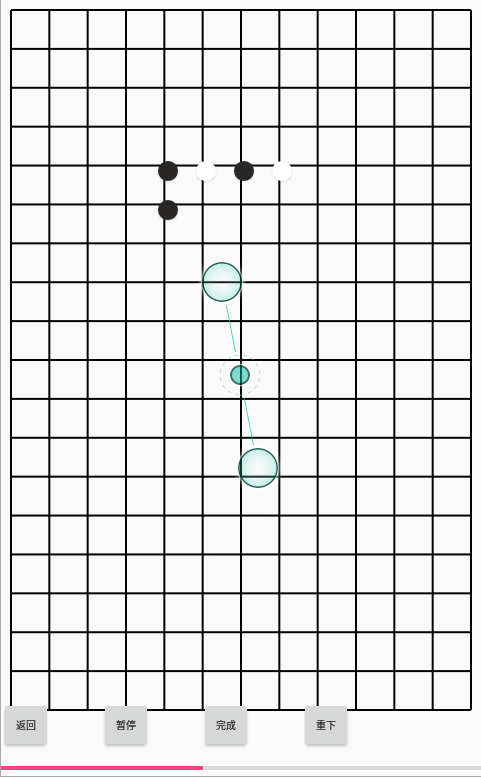
图 4- 13 棋局存储

在save的oncreate方法中，首先将传递来的数组数据通过循环重新存入二维数组当中。其次通过SQLiteOpenHelper连接数据库（创建了类sqlhelper继承），并通过sql语句insert into Data(name,x,y,chuchar) values(?,?,?,?)进行存储，存储的表单name名字由棋局总数、胜利者、用户输入的名字组成。点击确认按钮后存储完成，若点击取消按钮则回到游戏界面重新开始对局。

### 4.3.3 回放暂停、继续功能

暂停、继续按钮的功能为暂停、继续棋局回放。具体实现方式如下：

设置暂停、继续按钮的点击事件，根据棋局是否播放标识符显示文本为暂停或继续。当棋局是否播放标识符为true时，文本为暂停，点击事件为将计时器ch的时间跳动停止。文本为继续时，点击事件为将计时器ch的时间跳动开始。暂停前与暂停后的棋局回放界面效果如图4-14棋局回放\_暂停按钮的对比所示。



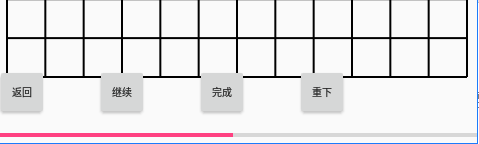
 

图 4- 14棋局回放\_暂停按钮的对比

### 4.3.4 快速回放功能

完成按钮的功能为快速回放功能，也就是跳过计时器直接展示完整棋局。实现方式如下：

将进度条的进度设为100%，并根据传入的棋局数组信息直接进行落子。将计时器关闭。使用快速回放功能后的棋局回放界面效果如图4-15棋局回放\_结束所示。

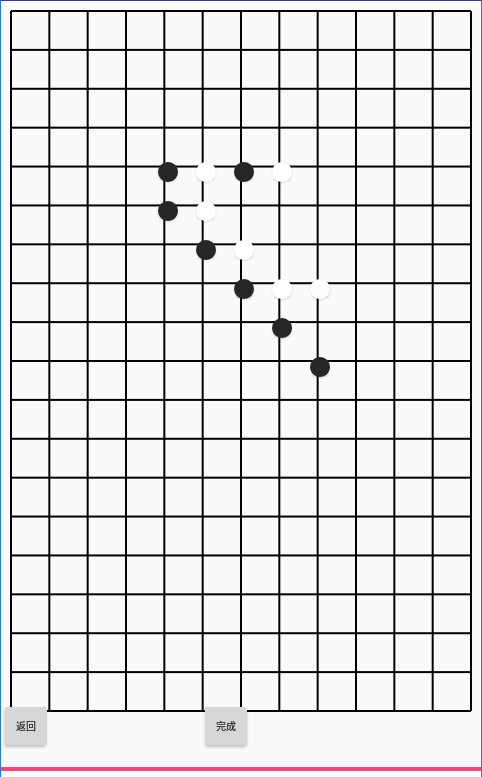


图 4- 15棋局回放\_结束

## 4.4 五子棋游戏的UI设计与实现

UI设计是游戏的外衣，没有靓丽的外表无法吸引用户的青睐，而游戏内容是游戏的本质，贫乏无趣的灵魂拥有再美丽的外衣也无法赢得用户长久的驻留。所以对UI的设计也是非常重要的。本小节介绍前文中因不具体属于某模块而省略的UI设计部分。

### 4.4.1 落子设计与实现

五子棋游戏需要用到两样器具：棋子与棋盘。五子棋的落子不同于普通的触摸游戏，只有触摸特殊区域棋盘线交界处程序才能给出反应进行落子。为了实现此功能，本文中所介绍的五子棋游戏放弃了传统的获取点击位置的方式，而是采用了给五子棋交界处放置按钮控件的方法。五子棋落子的具体实现方式如下：

在游戏内容的类中，首先通过findViewById(R.id.root)获取layout test.xml，在一个与棋盘交界处对应的双重循环中自定义button控件，将id设为棋盘下标相关的代数式i\*100+j，以此防止控件id与其他控件的id混淆。通过LayoutParams属性对其落点进行改动，定为棋盘的交界处。通过设置setAlpha使其透明不可见。在代码中添加按钮的点击事件（使用此方法需要implements View.OnTouchListener），由于棋类运动落子无悔，但手机的使用过程中往往容易误触导致用户体验严重受损，所以本五子棋游戏采用了全新的落子机制。当首次点击按钮时，程序会在点击的btn按钮处添加一个cbtn按钮。并将所有的btn按钮不可用，也就是不允许其他落子。cbtn按钮的背景是程序中存储的图片（勾号），cbtn与btn按钮的效果如图4-16有背景图的按钮cbtn与透明按钮btn所示。

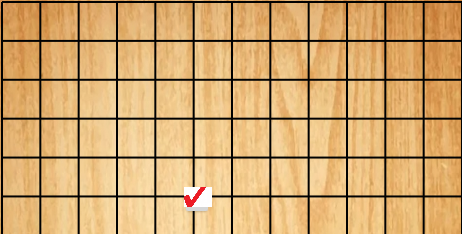
 

图 4-16 有背景图的按钮cbtn与透明按钮btn

此时若为误触只需要点击棋盘上的其他位置即可取消本次误触（通过重写on\_touch屏幕点击事件和btn的点击事件共同实现，点击btn后生成的cbtn使用的是同一个id my\_view并调用enable方法将所有的btn按钮不可用并将标识onlyid设为lala（默认值为no），on\_touch事件中若检测到onlyid不为默认值no，则将my\_view这个id的控件清楚）。若重复点击此cbtn按钮便视为在此处落子，落子的实现方式为获取当前位置，传入至 Bchequer（）或者Wchequer（）方法中，进行落子， Bchequer（）方法中将置入的棋子id设为curche+10000，curche即为这颗棋子落子的排位顺序如第一颗。第二颗。通过 drawable调用存储于程序当中的button\_bcircle\_shape.xml作为背景。落子于传入的位置，并将onlyid置为lala，curche+1。最终落子效果如图4-17落子效果图所示 。



图 4-17落子效果图

### 4.4.2 棋盘设计与实现

五子棋游戏中除了棋子外，所需要的器具就是棋盘。根据不同手机的屏幕像素数传输不同的数值进行棋盘设计。为了在具体实现中提升代码复用率，方便编写，利用Paint和Canvas对棋盘的绘制进行实现。具体实现方式如下：

建立一个java类，通过Canvas画布与Paint画笔绘制棋盘，以WAVG款虚拟机为例子，获得手机长宽的像素单位数，宽与棋盘的宽相除，长与棋盘的长度相除，设置画笔为黑色，2.0f宽度。设置一个循环以获取的间隔用画笔在画布上进行划线。即得到一个绘制棋盘的类，在游戏内容的类中先获取本身的layout的view root通过root.addView(new Myclass(this))调用此方法。即完成了五子棋游戏棋盘的绘制。具体实现代码核心如下：

首先需要对画笔进行布置。

PAInt pAInt = new PAInt();

pAInt.setColor(Color.BLACK);

pAInt.setStyle(PAInt.Style.STROKE);

pAInt.setStrokeWidth(2.0f);

pAInt.setAntiAlias(true);

在画笔准备完毕后，根据棋盘行数与列数进行绘画，间隔的大小通过手机的屏幕像素数计算获取。

for (int i=0;i<=12;i++){  
 STX=STW+i\*TLW;  
 STY=STH;  
 canvas.drawLine(STX,STY,STX,height-STH,pAInt); }

# 第5章 五子棋游戏核心算法实现

五子棋的对战模式可以被理解为对弈双方互设诘棋，AI棋手需要破除对方设下死局陷阱，同时还要做好陷阱请君入瓮。其中破除死局陷阱时AI棋手需要“眼睛”，也就是评估函数，获取棋局的信息，找出最紧要的落子点防御。而做好陷阱，AI棋手需要搜索算法来寻找最优落子点为对方棋手编制陷阱。最基础的搜索算法就是暴力搜索，但是如果采用这种搜索算法，随着搜索层数的增长，资源的占用率与时间效率的损耗都是天文数字。所以需要运用极大极小值搜索算法与α-β剪枝算法增强AI棋手的运算速度。本章会介绍构成AI棋手的主要算法。

## 5.1 基本思想

五子棋的基本目标是比对手更快的实现五子连成的线，为了这个目的，五子棋应该由进行局面评估，根据局面的分数最优选择评估分数最高的一步。当搜索到必胜路线时，AI选择的路线分数是必定最高的必胜线路。

## 5.2 评估函数

计算机不能直接识别棋盘，需要构建评估函数才能使AI可以通过评估函数的结果选择最佳的位置进行落子。评估函数的水平直接且很大长度决定了AI的智能程度与对弈水平，所以需要将人类棋手的落子思考过程以数学的方法重现，将这些部分量化。

### 5.2.1 五子棋基本棋型

五子棋中对各种棋子连成的棋型有专门的术语进行称呼，以下为五子棋棋型的基本介绍。

五连：指五个同色棋子或者更多的同色棋子在垂直、水平。左斜、右斜方向形成的“连”。需要注意在五子棋规则中，长连不能作为胜利条件。棋型如图 5-1 五连和长连。

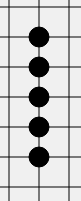
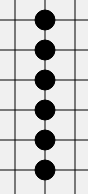
 

图 5-1 五连和长连

活四：指四个同色棋子在棋盘垂直、水平。左斜、右斜方向的连接，且在四子连接的方向上两端各有一个无子的点。棋型如图 5-2 活四。

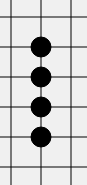


图 5-2 活四

冲四：指四个同色棋子在棋盘垂直、水平。左斜、右斜方向的连接，且在四子连接的方向上两端紧密连接的只有一个点可以落子。或者说同色棋手再落子一次可以形成五连的棋形称之为冲四。棋型如图 5-3 冲四。

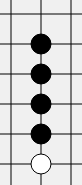


图 5-3 冲四

活三：指三个同色棋子在棋盘垂直、水平。左斜、右斜方向的连接，且在三子连接的方向上两端各有一个无子的点。

跳三：指同色棋手再落子一次可以形成活四的棋形称之为活三。棋型如图 5-4 活三和跳三。

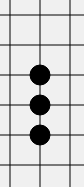
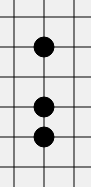
 

图 5-4 活三和跳三

冲三：指三个同色棋子在棋盘垂直、水平。左斜、右斜方向的连接，且在三子连接的方向上两端紧密连接的只有一个点可以落子。或者说同色棋手再落子一次可以形成冲四的棋形称之为冲三。棋型如图 5-5 冲三。

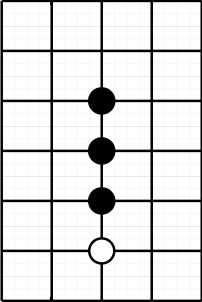


图 5-5 冲三

活二：指两个同色棋子在棋盘垂直、水平。左斜、右斜方向的连接，且在两子连接的方向上两端各有一个无子的点。

冲二：指两个同色棋子在棋盘垂直、水平。左斜、右斜方向的连接，且在两子连接的方向上两端紧密连接的只有一个点可以落子。或者说同色棋手再落子一次可以形成冲三的棋形称之为冲二。

活一：独立的一子，在棋盘垂直、水平。左斜、右斜方向上无紧密连接的子。

冲一：在棋盘垂直、水平。左斜、右斜方向上两端紧密连接的只有一个点可以落子，另外一端为异色子。

死棋：指没有可能连成五连的棋型，对于计算机而言，死棋容易被误认为活二，活三。棋型如图 4-6死棋，易被识别为活二。

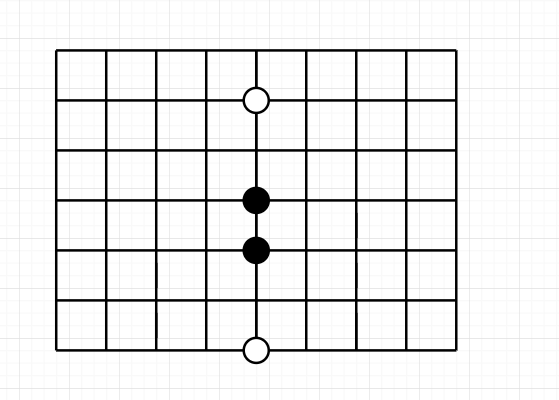


图 5-6死棋效果图

### 5.2.2 棋型分数评估

五子棋AI需要的棋局信息除了原始的棋盘信息，还需要棋盘信息中对应的各个棋型所代表的威胁程度。在统计完所有五子棋中出现的基本棋型后，只需要将每个棋型赋予对应的分数既可以表示出此棋型的威胁程度。

那么就需要根据五子棋的规则对每个棋型赋予合理的分数。例如五连即游戏胜利，所赋予的分数应当是最高的，且其他棋型的累加也不能超过。活四意味着极其接近胜利，所赋予的分数很高，但应当小于对方的冲四或者活四。所以有一下几个基本规则：

1、优先度的优先度顺序由高到低为：五连，活四，冲四，活三，冲三，活二，冲二，活一，冲一，死棋。

2、落子时对方的同优先度棋型的分数比己方的高

3、高优先度的棋型的分数要大于多个低优先度棋型的分数

由这些规则构建的分数如表5-1棋型分数表

表 5-1 棋型分数表

|  |  |  |
| --- | --- | --- |
| 棋型 | 己方分数 | 敌方分数 |
| 五连 | 1000000 | -1000000 |
| 活四 | 10000 | -10000 |
| 冲四 | 400 | -10000 |
| 活三 | 400 | -7000 |
| 冲三 | 40 | -50 |
| 活二 | 40 | -50 |
| 冲二 | 10 | -30 |
| 活一 | 10 | -30 |
| 冲一 | 1 | -10 |
| 死棋 | 0 | -0 |

## 5.3 常用核心算法及优化

### 5.3.1 博弈树简介

在各种博弈中，将己方准备做出选择的时间点作为一个节点，当决定完成后，整体状况发生改变。对于一个确定的状态，选择的可能性有很多种，每个选择都会产生一种新的情况，将准备做出选择的状态称之为父节点，选择后的状态称之为子节点，将子节点和父节点连接，子节点又作为父节点产生新的子节点，一直不断重复直至博弈结束，于是就形成了一颗树，称之为博弈树。博弈树的例图如图5-7所示。

而五子棋的下棋过程就如同从博弈树的父节点走到其中一个子节点，再由对手选择下一个字节点。每一个字节点都可能会严重影响游戏的结果。双方都想要通过更加好的选择来使自己胜利的可能性增加并且降低对手的胜利可能性。博弈树算法中的极大极小搜索算法就是这种情况的算法表现。

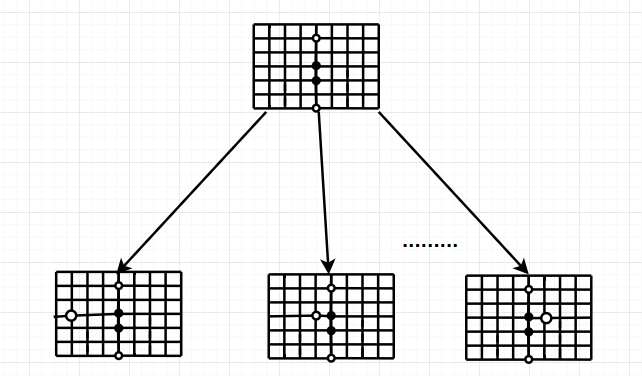


图 5-7 博弈树

### 5.3.2 极大极小搜索算法

极大极小算法是计算机博弈理论的经典算法，它是一种找到失败的最大可能性中的最小值的算法，也就是基于对方的最优选择来最小化对手的收益，通常通过递归来进行实现[7]。举一个例子，你的老板给了三个工作地点选项让你进行选择，但是他选择工作。第一个工作地点中有清洁工待遇的工作与首席执行官待遇的工作，第二个工作地点中只有白领工作。老板是资本家，会尽可能选择支出最少的工作，在这种情况下，想要获得更高利益的你不可能选择第一个工作地点，因为这样只有可能干执行官的活，领清洁工的薪水。所以最合理的情况就是选择第二个袋子，至少可以是白领工作白领待遇。

在五子棋游戏思想中也是如此，AI的目标就是在有限的搜索次数与深度内选择最终获利最多的选项。在对局中，人类棋手与AI都会试图在自己的回合内最大化自己的棋盘优势，同时尽可能削弱对手的棋盘优势。而在思考深度增加，思考对手落子的可能性时，会替对手选择最优的落子位置进行考虑，对手会落子于使自己分数最低的点，而己方则需要在这种前提下选择最优的点进行落子。目前，规则简单的棋类用到的搜索算法多数是在该算法的基础上进行改进［8］。

将博弈树的思想转变为数字型式，那极大极小值搜索算法就能在其中取得有效的应用：

将偶数层设为敌方，奇数层设为己方。己方优先考虑极大值的点，敌方考虑极小值的点

通过评估函数获取取得极大或者极小值的点。

对弈双方根据规则交替选择并返回父节点最后返回极大值，最后可以取得极大值的路线则是最优的路线。

极大极小值搜索算法在博弈树中的运用如图 5-8 所示。

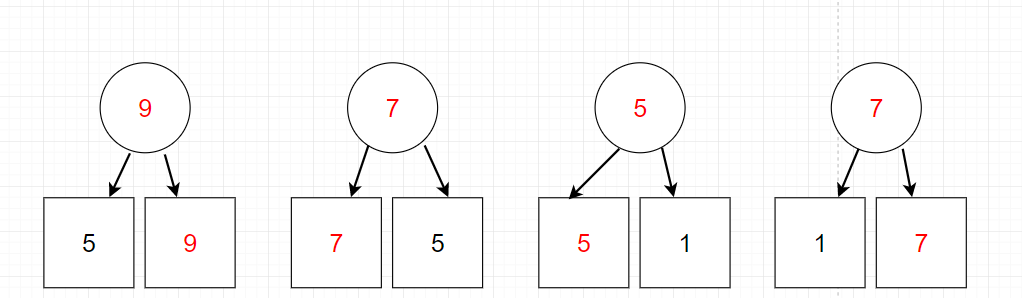


图 5-8-1 第一层max层（取大值，实际为最深的底层）

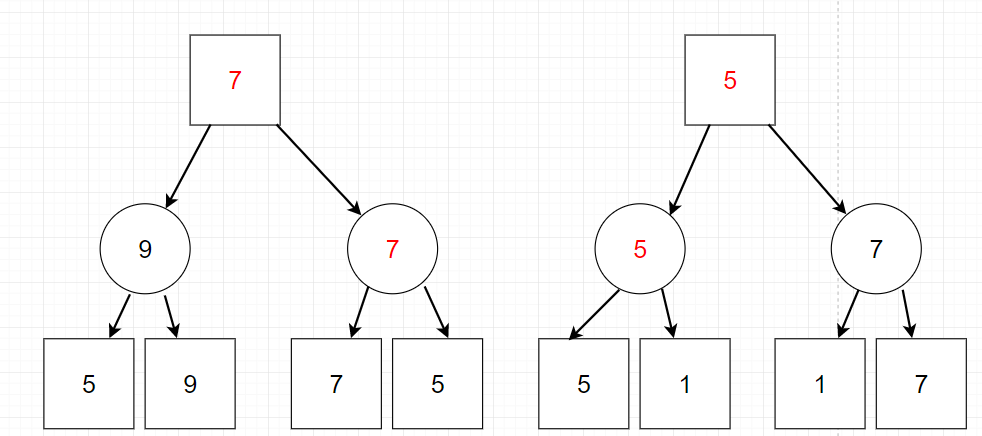


图 5-8-2 第二层min层（取小值）

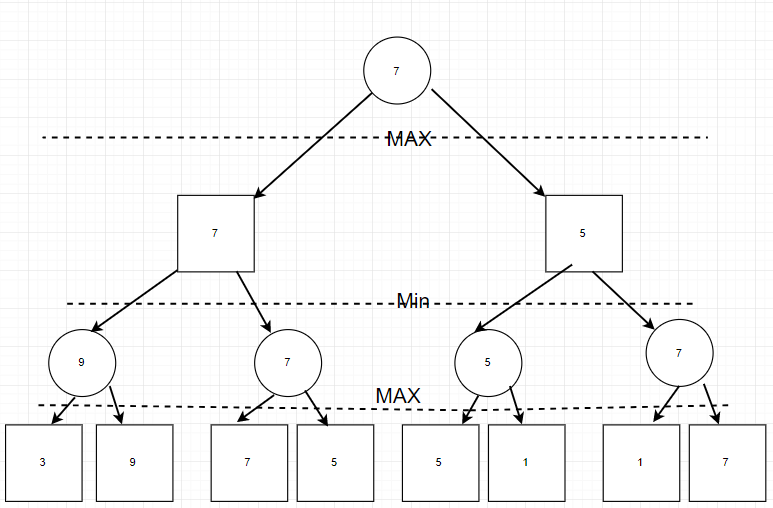


图 5-8-3 极大极小值搜索算法在博弈树中的运用

如图5-8-1所示，当子节点位于MAX层时，向上传递子结点中最小的值如5-7传递7，子节点在max层时向上传递子结点中最大的值如3-9传递9。

本文中算法的主要是采用递归的方式实现，该算法的优点在于代码量少，整体简洁，在树的遍历算法中具有良好的效果，可以很好的实现极大极小值搜索算法所呈现功能。算法的基本实现过程如下：

以下为本游戏实现极大极小值搜索算法所使用的代码主要部分，除去了一些不相干的函数如α-β剪枝算法的部分。其中bestpoints()方法是生成最优可落子点的函数，在后文中详细介绍。

调用函数时传入的参数分别为tmpcurche[][]：棋盘、depth ：搜索深度、tmpi：搜索进行次数 返回为得分。

public int AI(int tmpcurche[][],int depth,int alpha,int beta,int tmpi){

在极大极小值搜索算法中，当递归运算到最深的层数时，也就是博弈树中最深的子节点时，计算出当前棋局的分数并且返回。

if(depth==0) {//搜索进行到底部时返回得分开始回退  
 point []best=bestpoints(tmpcurche);  
 tmpcurche[best[0].x][best[0].y]=curche+tmpi;  
 int sumgrade=sum(tmpcurche);  
 tmpcurche[best[0].x][best[0].y]=0;  
 return sumgrade;  
 }

当递归运算计算到奇数层时，意味着AI站在AI的立场上对棋局进行判断，对同一颗子树下的所有节点进行展开（递归），并且在伪棋盘上落子，在落子结束后清除落子。

else if(tmpi%2==1){//AIceng  
 point []best=bestpoints(tmpcurche);  
 for(int j=0;j<best.length;j++)  
 {  
 tmpcurche[best[j].x][best[j].y]=curche+tmpi;  
 int nextAI=AI(tmpcurche,depth-1,alpha,beta,tmpi+1);  
 tmpcurche[best[j].x][best[j].y]=0;  
 }

当递归运算计算到奇数层时，意味着AI站在人类棋手的立场上对棋局进行判断，对同一颗子树下的所有节点进行展开（递归），并且在伪棋盘上落子，在落子结束后清除落子。

需要注意的是在实际应用当中，尤其在性能良莠不齐的移动智能终端当中需要注重极大极小值带来的数据冗余，浪费的空间和多余的运算不仅会影响程序运行的速度，还会影响手机的空间安全。例如当AI在已经发现一条必胜路线时是不需要再对后面的博弈树子树进行推演的，本无需搜索的节点就形成了数据冗余，因为极大极小算法属于静态搜索，依然会每个节点进行评估，展开所有的子树进行回退比较，这样会导致资源大量的浪费。所以需要采取措施减少不必要的节点搜索。

### 5.3.3 α-β剪枝算法

假设我们的搜索深度为n，m为每一深度所搜索的节点数，那搜索算法的时间复杂度就是O（mn），可以意识到当深度和节点数增大时，搜索时间将成指数级增长，这将会是一个恐怖的数字。为了减少计算量，可以对树进行剪枝。α-β剪枝算法是一种深度优先搜索算法，但是加入了剪枝操作。选择这种这种算法的目的在前文中已经提过，为了减少不必要节点的搜索，提高搜索的效率。这种算法的基本思想时在子节点搜索到的时候，与之前生成博弈树时计算评估父节点的兄弟节点的值进行比较，可以得出哪些子节点是没有意义的，可以停止拓展的。这也就是所谓的剪枝[9]。具体案例如图5-9所示：

最开始传入的α，β分别为负无穷与正无穷，代表着上界与下界。根据图 5-9-2的第二层左节点的 α，β分别为9与7，α>β所以未触发α-β剪枝。此处注意，在算法讲解中max层与min层触发α-β剪枝的条件不同，min层为α>β，max层为α<β。但在算法实现时，通过给予α，β负值的方式可以只使用α>β作为α-β剪枝的触发条件。

根据图 5-9-1的第三层右子树的左节点，此时第二层的右节点正在遍历其子节点的值进行比较，首先进行比较的就是第三层右子树的左节点，这个左节点递归进行子节点的比较得出α=5并返回自身的α值作为节点的值。

返回后，根据图 5-9-1第二层的右节点的值进行比较后，α的值变为5，此时α<β触发剪枝条件，进行剪枝。

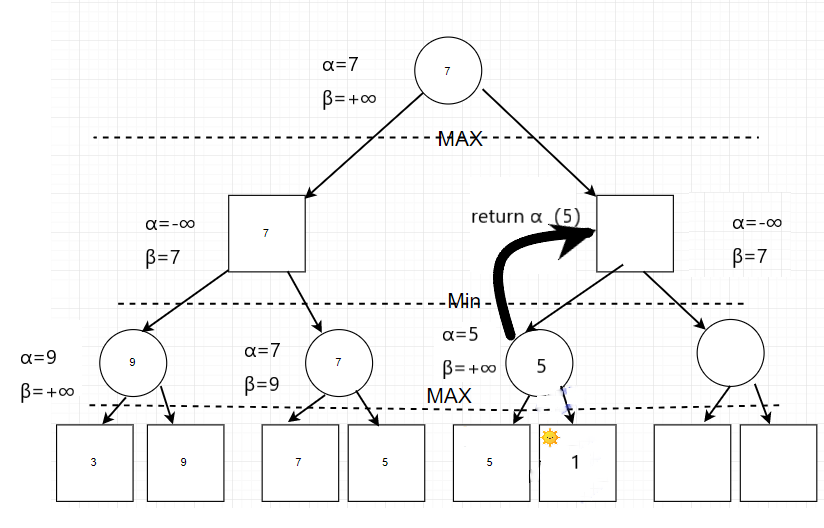


图 5-9-1 α-β剪枝算法在极大极小值搜索算法中的运用

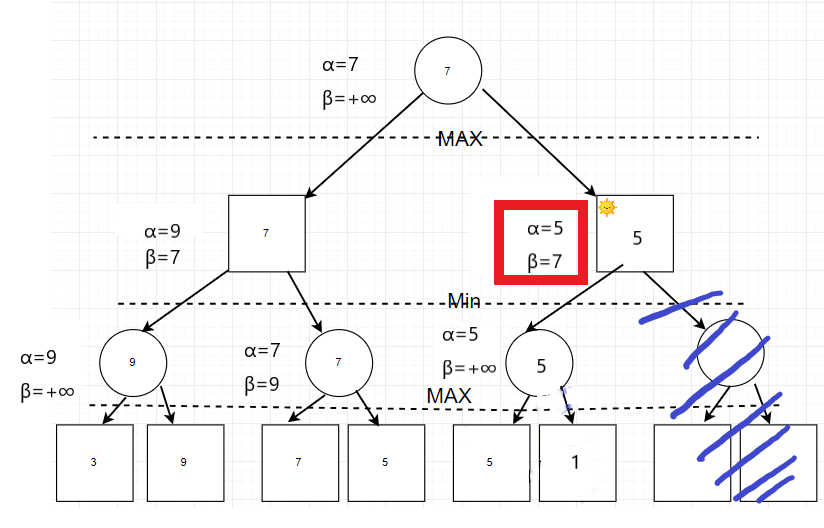


图 5-9-2 α-β剪枝算法在极大极小值搜索算法中的运用

需要注意的是，α-β剪枝算法对于搜索的顺序有较高的要求，在最坏的情况下，α-β剪枝算法在在极大极小值搜索算法中的运用作用为0。将搜索树的右子树部分视为完整的树来看待，以这个子树作为具体案例如图5-10-1及图5-10-2所示：在搜索第二层右节点时，进行子树值的遍历比较。在对第二层右节点的左子节点后返回值，第二层右节点α=1.β=5，此时并未触发α-β剪枝如图5-10-1所示。

在对第二层右节点的右子节点后返回值，α=7.β=5。此时触发α-β剪枝，若后续还有节点需要比较则可全部剪枝。但此时此子节点已经是最后的子节点，剪枝效果为0。若1、7子节点的顺序调换，剪枝的效果就增加了，所以α-β剪枝算法依赖于顺序。

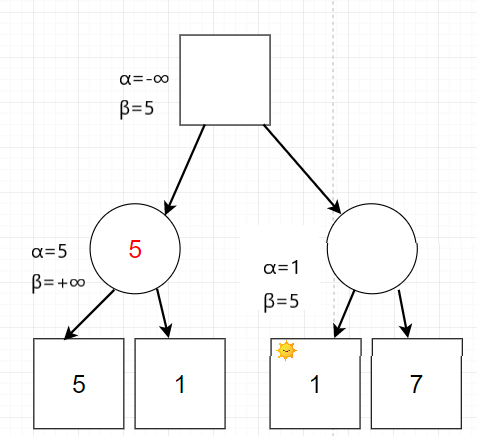


图 5-10-1 α-β剪枝算法在极大极小值搜索算法中的运用（效果差异）

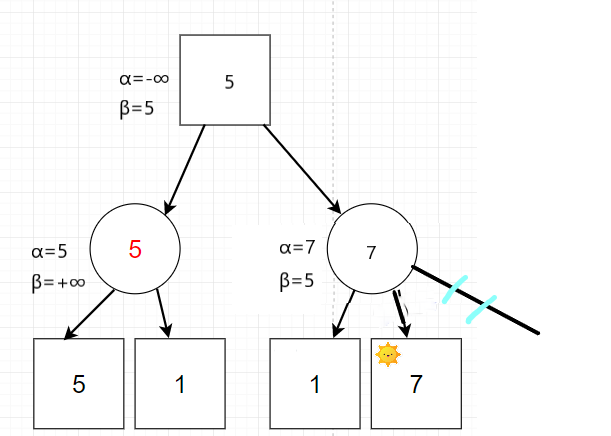


图 5-10-2 α-β剪枝算法在极大极小值搜索算法中的运用2

以下为AI算法中添加的α-β剪枝算法的实现方式

这一步是极大极小值的优化版本，通过给予负值实现max层min层的交替出现。

int nextAI=-AI(tmpcurche,depth-1,-beta,-alpha,tmpi+1);//此处αβ的顺序应当是相反的

在AI层时，我们选取的是子节点中最大的值，和这一颗子树下所有的子节点的值进行比较，若是比本身大就将其作为新的alpha。总的来说，就是获取最大值。

if(nextAI>alpha){  
 alpha=nextAI;

当深度达到预定传入的深度时，例如本难度中则是思考五步，那么当计算的深度为5时，利用全局变量记录准备传出的落子点。也就是下一步落子的最佳位置，代码如下。  
 if(depth==5)  
 {  
 AIbestpoint=best[j];  
 }

若此处可以判断与当前的alpha值相比，此处的值更小，则意味着此子节点以及其兄弟子树不必要进行搜索，即可通过break退出。

### 5.3.3 算法优化

前文有提及，若只是使用简单极大极小搜索算法和α-β剪枝算法，做出的AI棋手不但运算速度缓慢，还会影响手机空间。所以需要在极大极小值搜索算法的函数中对算法进行优化。有以下几种思路：

1、对移动顺序进行规划：

前文提及当移动搜索顺序不佳时，α-β剪枝算法的效率会降低，以至于不同的搜索顺序会导致不同的剪枝，导致可以剪枝的部分没有被剪枝。所以为了α-β剪枝算法可以取得良好的效果，可以优先检查每个子节点，对子节点按照MAX层MIN层的规则进行排序，这样可以使搜索从最优移动至最差，增加算法的效率。

2、最小窗口搜索

在α-β过程中,窗口越窄剪枝的可能性越大。一个性质为α=β-1 的搜索窗口叫作最小窗口。因为它是最窄的可能窗口，许多人认为，运用最小的窗口搜索可以进一步提高搜索效率。一些α-β的改进如 Nega Scout 和 MTD（f），均来源与最小窗口搜索。对于具有浓密的树一些博弈，它们具有重要的优势。[10]

3、前瞻性剪枝：

通过对于棋局的判断[11]，在进行α-β剪枝前提前剪去一些分枝，以减少博弈树的分支数量，在呈指数增长的搜索次数上，这种优化可以极大程度地减少搜索次数。以本文介绍的五子棋游戏举例，棋盘为14\*20，仅使用极大极小值搜索算法当搜索深度为4时，搜索次数为5,886,955,800次。若使用前瞻性剪枝，划定有意义落子范围，优化落子选择为15个，当搜索深度为4时，搜索次数为50,625次。减少的次数一目了然。所以本次算法优化采取了前瞻性剪枝的优化方式，也就是前文提到的bestpoints()方法，以下为核心代码部分，首先将棋盘数组转换成简便运算的全为1与0的组成方式。  
 for(int i=0;i<13;++i) {  
 for (int j = 0; j < 19; ++j) {  
 if(tmpchuchart[i][j]%2==1)  
 chuchart1[i][j]=1;  
 else if(tmpchuchart[i][j]%2==0&&tmpchuchart[i][j]!=0)  
 chuchart1[i][j]=2;  
 else { chuchart1[i][j]=0；}}  
 然后将每一个棋盘上有子的位置每个方向向外延伸3个位置，将这些位置记录为落子有意义位置。每个位置都进行模拟落子并记录落子分数，记录于worth数组，核心代码如下。  
 for(int i=0;i<13;++i){  
 for(int j=0;j<19;++j){  
 worth[i][j]=-1000000;  
 if(B[i][j]==true){  
 chuchart1[i][j]=1;  
 worth[i][j]=sum(chuchart1);  
 chuchart1[i][j]=0;}}}  
 最后，按照worth数组中的数组大小顺序将最大得分点塞入bestpoint数组中，并传递出去，核心代码如下：  
 int w;  
 for(int k=0;k<10;++k){  
 w=-1000000;  
 for(int i=0;i<13;++i){  
 for(int j=0;j<19;++j){  
 if(worth[i][j]>w){  
 w=worth[i][j];  
 point tmp=new point(i,j);  
 bestpoint[k]=tmp;}}}

为了不重复计算此点的分数，需要将已经进入bestpoint数组的点的分数清空，清除掉上一点，计算下一点的位置和分数，核心代码如下。  
 worth[bestpoint[k].x][bestpoint[k].y]=-1000000;  
 return bestpoint;

需要注意的是，调用此函数是传入的初始alpha与beta的值应当为正无穷与负无穷，本函数中采用的是一个相对大的值进行替代。

### 5.3.4 评估函数

评估函数的实质是将棋局的形势进行数字化处理，转化为计算机可以理解的形式让计算机得以理解局面。这是AI棋手的“眼睛”，是核心，最接近人类的部分。对于棋局的评估函数一般采用棋型统计方式，也就是根据5.2.2中给出的分数表进行打分。常见的统计方式为将不同的棋型转换为五元组或六元组常量输入程序，对棋盘进行扫描，出现棋型则统计分数并不再扫描棋型内的棋子。本五子棋游戏软件采用了新式的评估函数，从水平，垂直，左斜，右斜四个方向读取整个棋盘，当程序读取到未落子处或者异色棋子时进行一次判断，之前若有读取到棋子，则将读取到的棋子颜色，棋子个数，棋子位置传入评分函数，评分函数根据传入的信息分析之前的棋型的分数并返回回去，就完成了局面评估。以下为评估函数（水平）的代码的核心部分：

public int gethorizontalgrade(int tmpcurche[][],int origin){

int grade=0;  
for(int j=0;j<tmpcurche[0].length;j++){

算法的整体逻辑如下，当根据方向顺序读取落子时，遇到未落子处，与上一枚落子颜色不同的落子处，棋盘尽头处，则进行一次评分判断。设定两个布尔变量用于记录这次读取头方向是不是有空位与尾方向是不是有空位，以此判断此处棋型为活棋或者冲棋。设定颜色变量记录此次读取的棋型颜色与用于判断是否进行读取。设定数量变量记录读取的棋型为几连，此部分逻辑核心代码如下：  
 boolean isStartStem = false;  
 boolean isEndStem = false;  
 int colori=0,colorj=0;   
 int secounter=0,color=-1;   
 for(int i=0;i<tmpcurche.length;i++){

遇到有子位置且未开始棋型读取（未读取到棋子）时，开始读取棋型，将有子处的颜色赋给颜色变量，由于前方尚未开始读取棋型，说明前方无子，核心代码如下：  
 if(tmpcurche[i][j]!=origin&&color==-1&&j!=0)   
 {  
 Toast.makeText(context,"change", Toast.LENGTH\_SHORT).show();  
 secounter++;  
 isStartStem=false;  
 color=tmpcurche[i][j]%2;  
 colori=i;  
 colorj=j;  
 continue;  
 }

遇到无子位置或与上一枚落子颜色不同的落子处，根据不同的情况进行结算，本处展示部分结算，结算函数都使用同一个函数getprority（）。不同的是各处传入的前空后空的值与连续数的不同，核心代码如下。  
 if(tmpcurche[i][j]==origin||tmpcurche[i][j]%2!=color){   
 if(tmpcurche[i][j]%2!=color&&tmpcurche[i][j]!=origin)   
 { if(tmpcurche[i][j]%2==0){  
 isEndStem=true;  
 grade+=getpriority(isStartStem,isEndStem,secounter,colori,colori,2,tmpcurche,color);   
 color=0;  
 colori=i;  
 colorj=j;  
 secounter=1;   
 }  
 ……

评分函数的功能是根据评分表以及传入的信息返回局面分数。根据前文中提到的评估函数的分值表进行分数的计算与返回。以下为评分函数的具体实现：  
 public int getpriority(boolean isStartStem,boolean isEndStem,int secounter,int i,int

j,int direction,int tmpchuche[][],int color){  
 int priority = 0;

if (secounter == 5) {  
 priority = 100000;

若传入的前堵后堵都为真值时，说明传入的棋型在这个方向上已经没有价值，为死棋，代码如下：  
 if (isStartStem && isEndStem) {  
 Toast.LENGTH\_SHORT).show();  
 priority = 0;

若传入的前堵后堵都为假值时，说明传入的棋型为活棋，根据活棋的连数进行分数的返回。需要注意其中有特殊的情况，尽管前后皆为空，但是头尾实际上已经被另外的颜色隔开，实际上是已经无价值的死棋。此处以水平方向上活二与死二的差别进行介绍，核心代码如下：  
 if (isStartStem == isEndStem) {  
 if (secounter == 4) {  
 priority = 10000;  
 } else if (secounter == 2) {  
 boolean islive=true;  
 switch (direction){  
 case 2:{……  
 当方向是水平时，根据水平方向上前后是否有其他颜色的落子或者边界进行判定，若前后都有则可以视为死棋，此部分核心代码如下：  
 if(i-3<0||i+2>tmpchuche.length-1) {  
 priority = 0;  
 islive=false;  
 break； }  
 else if((tmpchuche[i-3][j]%2!=color)&&(tmpchuche[i+2][j]%2!=color))  
 { if(color==1){if(tmpchuche[i-3][j]==0||tmpchuche[i+2][j]==0){  
 break;}}  
 priority = 0;  
 islive=false;  
 }

# 第6章 测 试

6.1 测试平台

以下是本系统使用到的开发环境环境与工具：

1、Intel(R) Core(TM) i5-8300H处理器，8G内存

2、Windows 10 家庭中文版

3、Android Studio

4、5.1 WVGA API 25虚拟机

6.2 评价指标

棋类游戏无法用单一量化的指标对系统的价值做评价[15]。应该由以下几点作为评价的根据：

1、系统响应速度

2、AI智能程度

3、功能完善度

6.3 功能测试

功能测试从人机对战，人人对战，对局回放三个角度进行测试。主要目的是为了测试系统是否满足五子棋游戏的基本要求。游戏界面的人人对战部分基本功能完善，系统响应迅速。人机对战部分基本功能完善，系统响应速度较慢，在层数为9以上时游戏会短暂卡顿。原因为算法优化不足与虚拟机平台资源不足。AI智能程度方面，中等难度与两名男性大学生测试者的对弈中胜率为30%。测试界面如图6-1对战测试图（人人，人机）所示。

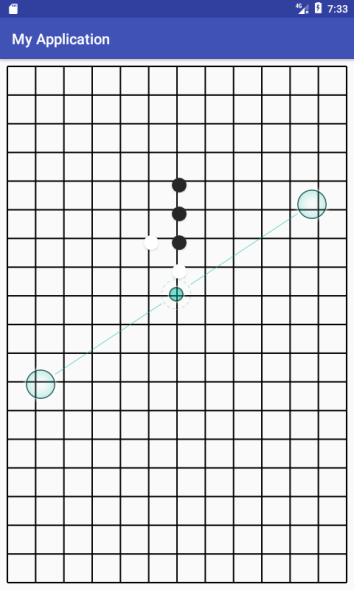
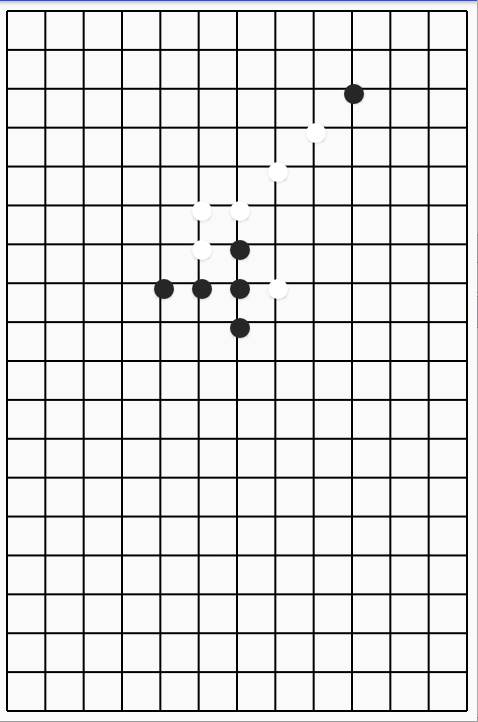


图 6-1 对战测试图（人人，人机）

棋局回放界面的功能完善，棋局的记录与回放在测试中没有出错，回放过程中系统响应迅速。基本功能完善。测试界面如图6-2棋局回放测试图所示。

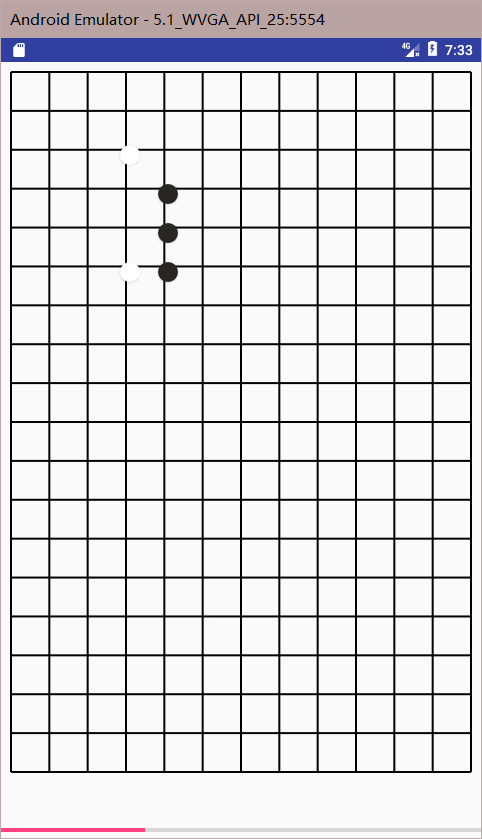


图 6-2 棋局回放效果图

测试用例：进行人机对战普通模式，游戏结束后保存棋局信息，名字为1，进入棋局回放进行回放。回放过程中选择重下。

测试结果：各项功能均正常运行，如图6-2与图6-1所示。

# 结 论

基于 Android Studio 的五子棋游戏灵活的使用了极大极小值搜索算法与α-β剪枝算法，实现了搜索算法并完成AI人机对战。系统采用sqlite作为数据库实现棋局重现功能。该游戏具体实现的功能如下：

1、五子棋游戏模块：

（1）切换棋盘皮肤，用户可在默认，自然两款皮肤选择。

（2）人机对战，具有胜负提示、棋局保存等功能。

（3）人人对战，具有胜负提示、AI提示落子区域、棋局保存、难度选择等功能。

2、棋局回放模块：

（1）棋局选择，用户可在按照存储的时间排序的棋局列表中选择棋局进行回放。

（2）回放暂停与继续，用户可通过按钮点击的方式控制回放的继续或者是暂停。

（3）残局重现，用户可通过按钮点击的方式将回放中未完成的棋局作为进行了一部分的游戏进行游玩。

（4）快速回放，用户可通过按钮点击的方式快速回放完成棋局。

本软件为用户提供了一个游戏平台，让更多的人感受到传统游戏的魅力，并在移动智能终端中让传统游戏焕发新的活力，也使游戏的操作过程更加便捷。

# 参考文献

1. 李果.六子棋计算机博弈及其系统的研究与实现[D].重庆大学硕士学位论文.2010
2. 毛丽民,卢振利,刘叔军,彭伟伟,李亮.五子棋对弈机器人移动平台的研究[J]. 微特电机.2017
3. 陈琦.试论嵌入式系统中数据库信息管理技术的应用[J].中小企业管理与科技(中旬刊).2020
4. 马获蕾,汤海凤. Android系统中SQLite数据库的研究[J].电脑知识与技术. 2013(28)
5. 卢凌.基于Java编程语言下计算机软件开发技术的思考[J]. 数字技术与应用. 2018
6. 武岳.关于计算机软件开发中JAVA编程语言的运用[J].信息记录材料. 2021
7. 戴传飞.嵌入式数据库SQLite研究与可视化工具设计[D].南京邮电大学硕士学位论文.2018
8. 姜勇. 五子棋人机对战系统设计[D]. 电子科技大学硕士学位论文, 2010
9. Alexander Pletzer. David Fillmore Conservative interpolation of edge and face data on n dimensional structured grids using differential forms[J] .Journal of Computational Physics.2015
10. M. Filali. J. Galindo Approximable WAP - and LUC -interpolation sets[J].Advances in Mathematics.2013
11. Eun Ji Lim; Dong Keun Kim. Educational Game Contents System using Inter-working between KINECT Camera, PC, and Android Platform based Humanoid Robot[J].Journal of The Korean Society for Computer Game.2015
12. 罗文浩. 五子棋对弈平台的设计与实现 [D]. 西安电子科技大学硕士学位论文, 2015
13. 李昊. 五子棋人机博弈算法优化研究与实现[D]. 大连海事大学硕士学位论文, 2020
14. 孙世文.五子棋人工智能算法实现研究[J].中国新通信. 2018(23)
15. 李云飞. 基于C语言的地棋游戏的设计研究 [D]. 内蒙古大学硕士学位论文, 2015

# 致 谢

凌晨的城市如此安静，远处的灯火一闪一闪，像课上打瞌睡的人头一顿一顿。静谧的夜晚总是会让我失神，回想起曾经一个个夜不能寐的日子。转眼间，论文终稿即将修改完毕，这段时间就像我的大学时光一样传瞬即逝，截下来一段品味有酸有苦，像黑咖啡，将它融在一起饮下，香气就会在每段时间的缝隙间渗透出来，一切都是那么的美好。

感谢我的导师杜娟。在我对着课题无从下手时，是杜娟老师给予我指引和帮助。当我出现困惑时，杜娟老师也会迅速回复帮助我攻克难关。借此机会，我谨向杜娟老师致以深深的谢意。

感谢我的同学，朋友们。四年来，是你们陪在我身边留给了我美好的回忆，也是你们的有益的建议完善了我的毕业设计，谢谢你们的陪伴与帮助。

感谢的人，感谢的话语还有很多，但也难以表达出我的感激之意。少年郎也该穿起新时裳，走出学校，或是步入社会成为世界运转所需的齿轮，或是醉心于学术向无穷的知识投身而去。毕业不是歌剧的终章，而是人生的序幕。在此，祝各位同学踏上新的征程，以梦为马，不负韶华。