

移动终端开发技术

课程设计报告

题 目	五子棋小游戏
班 级	2018软件技术二班
学 号	201825040205
姓 名	韩晓冰
小组成员	李登云, 贾世杰, 李宏伟, 高一凡
指导教师	陈媛媛

2019 年 7 月 2 日

目 录

1. 需求分析	1
1.1 总体分析.....	1
1.2 初始化.....	1
1.3 主循环控制模块.....	1
1.4 玩家下子.....	1
1.5 盘面分析填写棋型表.....	2
1.6 对方下子.....	2
1.7 胜负判断.....	2
2. 设计思路	3
2.1 五子棋的起源与发展.....	3
2.2 五子棋的基本术语及规则.....	3
2.2.1 五子棋的基本术语:	3
2.2.2 五子棋的规则.....	4
2.3 功能描述.....	5
3. 详细设计	6
4. 运行调试与分析讨论	8
4.1 运行结果.....	8
4.2 分析讨论.....	12
5. 项目总结	13

摘要

游戏设计已经成为现在社会的一种热门行业。五子棋是一种受大众广泛喜爱的游戏，其规则简单，变化多端，只需两人即可进行，非常富有趣味性和消遣性。这里设计和实现了一个双人对下的黑白五子棋单机游戏。主要应用了Android ,和java编程来完成这个游戏的设计的。本文主要介绍了本游戏开发环境，使用当前优秀的开发工具Android studio编程。包含了五子棋程序的棋盘初始化、游戏规则、胜负判断方法，及帮助等。

序 言

五子棋是起源于中国古代的传统黑白棋种之一，是一种训练人逻辑思维严密性的游戏。该设计以五子棋游戏业务为背景，设计、开发一套“五子棋游戏”软件。

通过该题目的设计过程，可以提高基本编程能力和游戏开发的技巧，熟悉C 语言图形模式下的编程。本程序中涉及很多图形函数，包括：坐标定位；颜色设置（背景色和文本颜色设置），线条颜色、填充颜色；直线风格、文本风格、填充风格；画线、画圆、画矩形函数；图形系统的初始化和关闭；图形屏幕和文字屏幕的消除；从键盘获取键值等。通过本程序的训练，可以使我们加深对 C 语言的更深刻的了解，掌握五子棋游戏开发的基本原理，为开发出高质量的游戏软件打下坚实的基础。

1. 需求分析

1.1 总体分析

软件需求分析是软件开发周期的第一个阶段，也是关系到软件开发的关键一步。对于任何一个软件而言，需求分析工作都是至关重要的一步。只有通过软件需求分析，才能把软件的功能和性能由总体的概念性描述转化为具体的规格说明，进而建立软件开发的基础。实践表明，需求分析工作进行得好坏，在很大程度上决定了软件开发的成败。

软件需求分析的任务是：让客户和开发者共同明确将要开发的是怎样一个软件。具体而言，就是通过对问题和环境的理解、分析和综合，建立逻辑模型，完成新软件的逻辑方案设计。

基于本游戏，首先得为整个棋盘建立一张表格用以记录棋子信息，我们使用一个 15*15 的二维数组 `table[15][15]`（15*15 是五子棋棋盘的大小），数组的每一个元素对应棋盘上的一个交叉点，用‘0’表示空位、‘1’表示自己的子，‘2’表示对方的子；这张表也是今后分析的基础。在此之后还要为两个玩家双方各建立一张棋型表 `Computer[15][15][4]`和 `Player[15][15][4]`，用来存放棋型数据。

1.2 初始化

首先，建立盘面数组 `Table[15][15]`、对战双方的棋形标 `Computer[15][15][4]` 和 `Player[15][15][4]`并将它们清零以备使用；然后初始化显示器、键盘、鼠标等输入设备并在屏幕上画出棋盘（棋盘可以不显示）。

1.3 主循环控制模块

控制下期顺序，当轮到某方下子时，负责将程序转到相应的模块中去，主要担当一个调度者的角色。

1.4 玩家下子

当轮到玩家下时，您通过键盘或鼠标在棋盘上落子，程序会根据该点的位置，在 `Table[15][15]`数组的相应地方记录‘2’，以表明该子是玩家下的。

1.5 盘面分析填写模型表

您在下五子棋时，一定会先根据棋盘上的情况，找出当前最重要的一些位点，如“活三”、“冲四”等；然后再在其中选择落子点。先来分析己方的模型，我们从棋盘左上角出发，向右逐行搜索，当遇到一个空白点时，以它为中心向左挨个查找，如果遇到己方的子则记录然后继续，如果遇到对方的子、空白点或边界就停止查找。左边完成后再向右进行同样的操作；然后把左右两边的记录合起来，得到的数据就是该点横向上的模型，然后把模型的编号填入到 `Computer[x][y][n]` 中就就行了（`x`、`y` 代表作标，`n=0、1、2、3` 分别表示横、竖、左斜、右斜四个方向）。而其他三个方向的模型也可用同样的方法得到，当搜索完整张棋盘后，己方的模型表也就填写完毕了。然后再用同样的方法填写对方模型表。

注意：所有模型的编号都要事先定义好，越重要的号数越大！

1.6 对方下子

有了上面填写的两张模型表，就是遍历模型表 `Computer[15][15][4]` 和 `Player[15][15][4]` 找出其中数值最大的一点，在该点下子即可。但这种算法的弱点非常明显，只顾眼前利益，不能顾全大局，这就和许多五子棋初学者一样犯了“目光短浅”的毛病。如果在这儿下子将会形成对手不得不防守的模型（例如‘冲四’、‘活三’）；那么下一步对手就会照您的思路下子来防守您，如此一来便完成了第一步的预测。这时再调用模块四对预测后的模型进行分析，如果出现‘四三’、‘双三’或‘双四’等制胜点，那么己方就可以获胜了（当然，对黑棋而言‘双三’或‘双四’是禁手，另当别论）；否则照同样的方法向下分析，就可预测出第二步、第三步……

等一等，要是盘面上没有对手必须防守的模型，那该怎么办呢？进攻不成的就得考虑防守了，将自己和对手调换一下位置，然后用上面的方法 何乐而不为呢！

1.7 胜负判断

无需多言，某方形成五子连即获胜；若黑棋走出‘双三’、‘双四’或长连即以禁手判负。

2. 设计思路

2.1 五子棋的起源与发展

五子棋是起源于中国古代的传统黑白棋种之一。五子棋不仅能增强思维能力，提高智力，而且富有哲理，有助于修身养性。它简单易学的特征，为人民群众所喜闻乐见，又有深奥的技巧和高水平的国际性比赛；它的棋文化源远流长，且具有东方的神秘和西方的直观；既有“场的”概念，亦有“点”的连接。它是中西文化的交流点，是古今哲理的结晶。

五子棋起源于古代中国，发展于日本，风靡于欧洲。据日本史料文献记载介绍，中国古代的五子棋经由高丽（朝鲜），于1688年至1704年的日本元禄时代传到日本。到去日本明治32年（公元1899年），经过公开征名，连珠这一名称才被正式确定下来。从此，连珠活动经过了不断改良，主要是规则变化（即限制黑棋一方）。例如，1899年规定，禁止黑白双方走“双三”；1903年规定，只禁止黑方走“双三”；1912年规定黑方被迫走“双三”亦算输；1916年规定，黑方不准走“长连”；1918年规定，黑方不准走“四，三，三”；1931年规定，黑方不准走“双四”，并规定将19*19的围棋棋盘改为15*15的连珠专用棋盘。本世纪初五子棋传入欧洲并迅速风靡全欧。通过一系列的变化，使五子棋这一简单游戏复杂化、规范化，而最终成为今天的职业连珠五子棋，同时也成为国际性比赛。

2.2 五子棋的基本术语及规则

2.2.1 五子棋的基本术语

- (1) 先手：进攻的着法，也指对弈双方执黑棋先在棋盘上落子的一方；
- (2) 四三：指同时具备两个先手，其中一个四，另一个是活三；
- (3) 活三（包括连三和跳三）：[1]连三：紧紧相连的同色三子[2]跳三：中间间隔一子的活三；
- (4) 四四：一子落下形成两个“四”的棋形；
- (5) 三三：一子落下形成两个活三的棋形；

- (6) 长连：相同颜色的连续六子或六子以上；
- (7) 禁手：对局中禁止使用的战术或被判为负的行棋手段；
- (8) 胜局：对局的一方获胜，成为“胜局”。
- (9) 和棋：不分胜负的对局或双方同意平局。

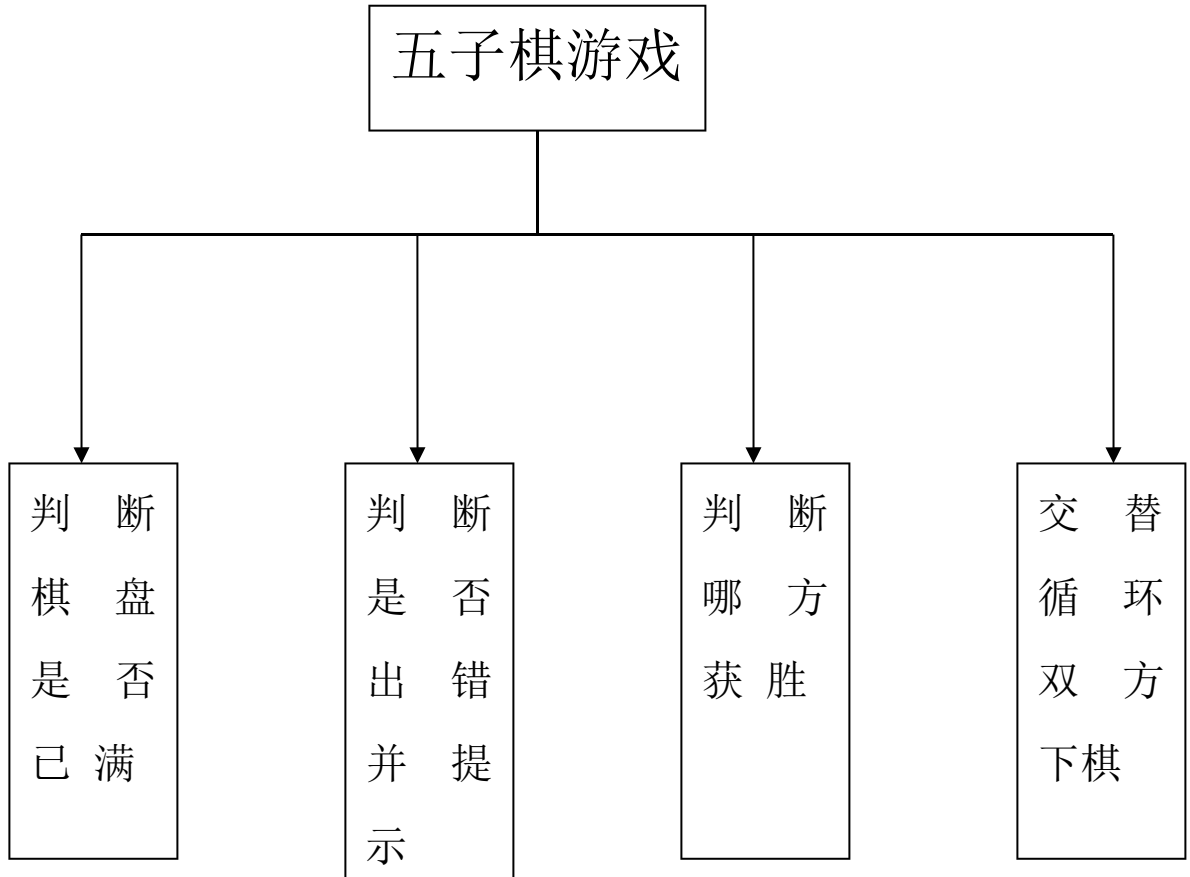
2.2.2 五子棋的规则

五子棋游戏的一个特点是先行的一方优势很大，因此在职业比赛中对黑方做了种种限制，以利公平竞争。五子棋的比赛规则如下：

- (1) 黑棋先手、白棋后手，从天元开始互相顺序落子；
- (2) 最先在棋盘横向、纵向或斜向形成连续的同色五子的一方为胜；
- (3) 黑棋禁手判负、白棋无禁手。黑棋禁手有“三，三”、“四，四”和“长连”，包括“四，三，三”和“四，四，三”。黑棋只能以“四，三”取胜；
- (4) 如分不出胜负，则以平局结束；
- (5) 五连与禁手同时形成，判胜；
- (6) 黑方禁手形成时，白方应立即指出。若白方未发现或发现后不立即指出，反而继续落子，则禁手失败，不再判黑方负。

五子棋是黑白双方或两个人之间的竞技活动，由于对黑白双方规则不同，黑棋必须先行。

2.3 功能描述



功能说明

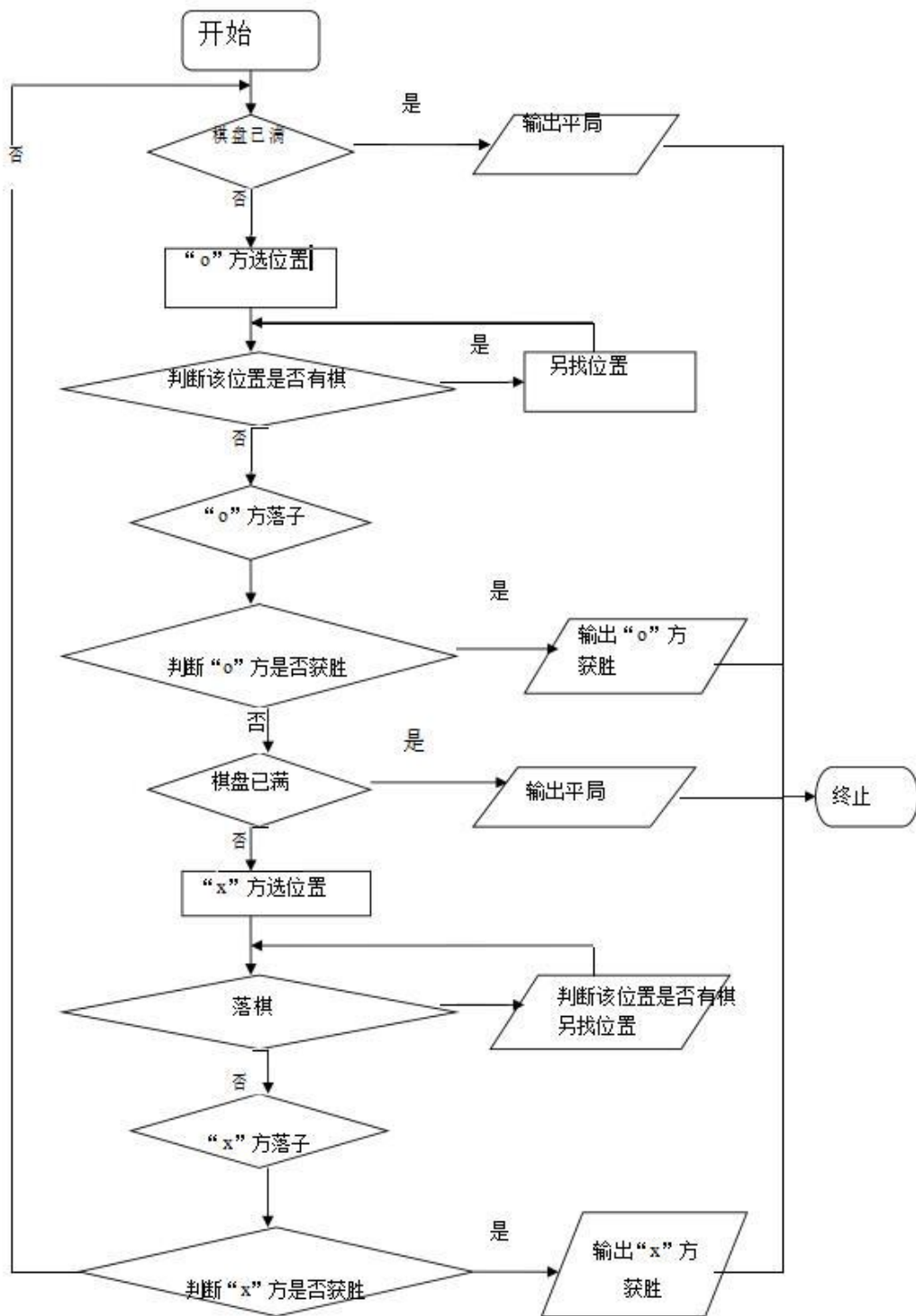
该五子棋游戏基本上实现了五子棋的游戏功能，有双方下棋的界面及最终判定结果的界面。同时该游戏采用二维坐标实现，明了易懂，方便玩家在游戏过程中的基本操作，使游戏更加简便。在细节方面，该系统提供实时存储功能，随时记录以完成的游戏，使用户可以很好的处理以外中断的情况。该游戏基本实现了游戏的一些要求和特征。在游戏的源程序和文档方面，我们也严格遵守软件工程的思想，立足实验要求，确定任务，需求分析，设计和编码，每个步骤都力求清晰易懂。源代码注释详尽，各功能模块分明，可移植性强。当然该系用也有不足的地方，第一次进行独立的课程设计，也有许多细节方面是考虑不到的，这款游戏也是在不断的调试和修改中产生和完善的。希望老师能够指出不足，帮助我们不断提高。

3. 详细设计

3.1 流程图

本程序定义了各种操作函数、各种状态判定，思想明确，思路清晰。各个判断选择了不同路径，因此继续进行或输出结果。程序中，“循环”的利用非常直接和清晰，双方交替下棋，因此循环往复。最终决出胜负或最终平局。分析时，也考虑了许多情况，针对各个情况均作出了相应措施和解决方案。

程序采用循环进行双方交替下棋，并进行了很多判断。首先判断棋盘是否已满，若棋盘已满，则输出平局，结束游戏；若棋盘未满，则继续进行。然后判断“o”方是否胜出，若“o”方胜出，则输出“o”方获胜，若未获胜，则继续判断棋盘是否已满，若已满则输出平局，结束游戏；若未满，则继续游戏。由“x”方选择位置，继续前面的操作，



4. 运行调试与分析讨论

4.1 运行结果

图 1 运行结果初始图



4.2 分析讨论

该五子棋游戏经过多次调试运行,能够顺利进行二人对弈。但由于其能力有限,不能够制作出拥有美观界面的五子棋游戏。该游戏的棋盘构成,应此,可以先让一方将棋子落在(7,7)这个空格上,然后另一方可以在起周围进行落子对弈。此程序由Android进行编译,应此在Android的环境下运行可以显现出较好的界面来进行游戏的操作。虽然该游戏的界面不太完美,但易于操作,能够完好运行,带给玩家欢乐!

5. 项目总结

随着计算机的普及，人们对游戏的要求也越来越高，像五子棋这种休闲游戏丰富了人们的业余生活。至此，对用Android来实现手机小游戏有了一个比较全面的了解。

通过本课题的研究及开发,基于 android的五子棋游戏软件开发完毕，实现了五子棋的二人对战。本课题按软件工程的要求设计与实现了整个系统，即需求分析、系统总体分详细设计与编码实现等步骤，最后进行了系统测试，并通过了系统测试。

通过此次课设，让我更加熟悉了对做一个项目的不足，游戏界面不美观。也通过这次的合作意识到开发应用团队的重要性，也认识到自己许多地方的不足。

附件一 源代码

```
package com.example.hxb02.myapplication;
import java.util.ArrayList;
import java.util.List;
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Picture;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnClickListener;

public class ChessFiveView extends View implements OnClickListener{
    int maxRows=12;
    int maxColx=14;
    int x0=14;
    int y0=100;
    int D=76;
    Bitmap start, stop;

    int chessX, chessY;
    int chessType=1; //1表示黑子, 2表示白子
    int chess[][]=new int[maxRows][maxColx]; //0表示空白, 1表示黑子, 2表示白子
    int winner=0;
    List<StepInfo> lstStep=new ArrayList<StepInfo>();

    public ChessFiveView(Context context) {
        super(context);
```

```

setOnTouchListener(this);
start=BitmapFactory.decodeResource(getResources(), R.drawable.start1);
stop=BitmapFactory.decodeResource(getResources(), R.drawable.stop2);

}

@Override
protected void onDraw(Canvas canvas) {
    // TODO Auto-generated method stub
    super.onDraw(canvas);

    Paint paint1=new Paint();
    paint1.setColor(Color.BLACK);

    for(int i=0;i<maxRows;i++){
        canvas.drawLine(x0, y0+i*D, x0+(maxColx-1)*D, y0+i*D, paint1);
    }
    for(int i=0;i<maxColx;i++){
        canvas.drawLine(x0+i*D, y0, x0+i*D, y0+(maxRows-1)*D, paint1);
    }

    canvas.drawBitmap(stop, 500, 1000, paint1);
    paint1.setColor(Color.CYAN);
    paint1.setTextSize(50);
    canvas.drawText("我的五子棋", 400, 80, paint1);

    /*if(chessType==1) {
    paint1.setColor(Color.BLACK);
    }else {
        paint1.setColor(Color.WHITE);
    }
    canvas.drawCircle(x0+chessX*D, y0+chessY*D, 20, paint1);
    //canvas.drawCircle(x0+chessX*D, y0+chessY*D, 20, paint1);
    chessType=chessType==1?2:1;

    */
    for(int i=0;i<maxRows;i++){
        for(int j=0;j<maxColx;j++){
            if(chess[i][j]>0) { //判断棋盘是否被占用
                if(chess[i][j]==1) {
                    paint1.setColor(Color.BLACK);
                }else {

```

```

        paint1.setColor(Color.WHITE);
    }
    canvas.drawCircle(x0+j*D, y0+i*D, 20, paint1);

    }
}

if(winner>0) {
    paint1.setColor(Color.RED);
    paint1.setTextSize(70);
    String s=winner==1?"黑子赢了":"白子赢了";
    canvas.drawText(s, 100, 300, paint1);
}
//显示我要悔棋
paint1.setTextSize(50);
paint1.setColor(Color.BLUE);
canvas.drawText("悔棋", 350, 1050, paint1);

canvas.drawBitmap(start, 100, 1000, paint1); //开始游戏

}

@Override
public boolean onTouch(View v, MotionEvent event) {
    // TODO Auto-generated method stub
    float x=(float)event.getX();
    float y=(float)event.getY();
    //判断悔棋
    if(x>=300&&x<=450&&y>=1000&&y<=1100) {
        goBack();
        return false; //结束onTouch()方法的执行
    }
    //开始游戏
    if(x>=100&&x<200&&y>=1000&&y<=1100) {
        rePlay();
        return false;
    }

    System.out.println(chessX);
    System.out.println(chessY);
    chessX=(int)((x-x0)/D+0.5); //如果超过一半, 则取另一行, 表示列
    chessY=(int)((y-y0)/D+0.5); //表示行
    //判断是否越界
    if(chessX<0 || chessX>=maxColx || chessY<0 || chessY>=maxRows) {

```



```

        return false;//结束OnTouch方法的执行
    }

    if(chess[chessY][chessX]>0){ //判断棋盘上该位置是否已经被占用
        return false;//结束OnTouch方法的执行
    }
    chess[chessY][chessX]=chessType; //用对应的棋子占据该位置

    StepInfo step=new StepInfo(chessY, chessX, chessType);
    lstStep.add(step);
    //更换棋子类型
    chessType=chessType==1?2:1;
    checkWinner();//进行胜负的判断
    postInvalidate(); //在改变了坐标之后立即重新绘制界面

    return false;
}

public void goBack(){
    //1. 从lstStep数组中取出最上面的一个元素
    //2. 获得位置行和列的信息, 并将对应位置的chess[row][col]的值设为0
    //3. 从新画图, 把lstStep中最上面的元素去掉

    int n=lstStep.size();
    if(n<=0){
        return ;//退出goBack()方法的执行
    }
    StepInfo step=lstStep.get(n-1);
    int row=step.getRow();
    int col=step.getCol();
    chessType=step.getChessType();

    chess[row][col]=0;
    postInvalidate();//从新画图
    lstStep.remove(n-1);
}

public void rePlay(){
    for(int i=0;i<maxRows;i++){
        for(int j=0;j<maxColx;j++){
            chess[i][j]=0;
        }
    }
    lstStep.clear();
    chessType=1;
    winner=0;
    //从新画图
    postInvalidate();
}

```

```
}
```

```
private void checkWinner() {  
    // 从上到下, 从左到右 依次搜索  
    for(int i=0;i<maxRows;i++){  
        for(int j=0;j<maxColx;j++){  
            if(chess[i][j] == 0)  
                continue; //后面的语句不在执行, 继续执行下一次循环  
  
            //以该位置为基准, 搜索4个方向 右 , 下, 右上, 右下  
            //判断是否有五个同类型的棋子  
  
            // 向右方向  
            int count=1;  
            for(int k=1;k<=4 && (j+k<maxColx);k++){  
                if(chess[i][j+k] != chess[i][j]) break;//在向右搜索  
                的时候, 如果遇到空格就不在搜索, 跳出当前的循环  
                else  
                    count++;  
            }  
            if(count >= 5){  
                winner = chess[i][j];  
                return ;//跳出checkWinner()方法的执行  
            }  
            // 向下方向  
            count=1;  
            for(int k=1;k<=4 && (i+k<maxRows);k++){  
                if(chess[i+k][j] != chess[i][j]) break;  
                else  
                    count++;  
            }  
            if(count >= 5){  
                winner = chess[i][j];  
                return ;//跳出checkWinner()方法的执行  
            }  
  
            //斜上  
            count=1;  
            for(int k=1;k<=4 && (j+k<maxColx) && (i-k>0);k++){  
                if(chess[i-k][j+k] != chess[i][j]) break;  
                else  
                    count++;  
            }  
            if(count >= 5){  
                winner = chess[i][j];  
                return ;//跳出checkWinner()方法的执行  
            }  
        }  
    }  
}
```

```

    }

    //斜下
    count=1;
    for(int k=1;k<=4 && (j+k<maxColx) && (i+k<maxRows);k++){
        if(chess[i+k][j+k] != chess[i][j]) break;
        else
            count++;
    }
    if(count >= 5){
        winner = chess[i][j];
        return ;//跳出checkWinner()方法的执行
    }

}

```

