一个象棋手机游戏的源代码

原文：http://www.easylib.org/question/game/11708.html

作者：DLUT_608

最近看到了一些五子棋，扫雷的代码讲解的文章，我也就写了个手机的象棋游戏，写的不是太全面，但还是能实现基本功能，共享出来供大家交流交流。

先介绍一下我的大体思路吧，我采用 canvas 让手机自己画出棋盘和棋子，而不是采用调用图片，虽然麻烦，但能锻炼自己的编程能力还能让算法简单，同时还能节省空间。具体的细节在代码中在说吧。首先要有一个主程序 Game，这里比较简单，大家一看就能明白，我就不多说了.

```
import javax.microedition.lcdui.Display;
import javax.microedition.midlet.MIDlet;
public class Game extends MIDlet {
    GameCanvas game;//定义游戏界面的 Canvas 类 GameCanvas 的对象 gobang
    public Game() {
        super();
        game=new GameCanvas(this);//生成 GameCanvas 类的对象 game
    }
    protected void startApp(){
        Display.getDisplay(this).setCurrent(game);
          //在屏幕上绘出游戏见面 game
    }
    protected void pauseApp(){
    }
    protected void destroyApp(boolean arg0){
    }
}
```

然后就是程序的主题部分了——GameCanvas，这里实现了从画棋盘棋子一直到判断和输出.

我的主题思想是把棋盘初始化为一个 2 维数组，在有棋子的地方初始化为非 0 数，其他的都初始化为 0;

大家可在代码中看到，在图象输出和棋子移动也都是基于这个数组进行的。

```
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;
public class GameCanvas extends Canvas implements CommandListener
{
    protected Game game;
    protected int empty;//屏幕右侧留的空间
    protected int x;//棋盘输出的坐标
    protected int cellWidth;//每个棋格的边长
    protected int mapWidth,canvasW;//棋盘的宽度和画布的宽度
    protected int a,b,c,d;//这是画炮下面的那几个折线，没什么用
    protected int chessR;//棋子的半径
    protected int selectedX,selectedY;//选择框在棋盘格局上的 x,y 位置
    protected static int i,j;
    protected int m,n,p;//记住开始的 selectedX,selectedY 和
point[selectedX][selectedY]
    protected String q;//记住 word[selectedX][selectedY]
    protected int guard,guard1,guard2,g,g1;//标记 FIRE 被按了多少次,g
是用来判断走直线时前后的棋子,中间是否有其他棋子的累加器
    protected static int g2,isRedWin,isWhiteWin;//g2 表示该谁走了，后
面那俩顾名思义了
    protected Command exitCmd;

    protected int point[][]={{1,2,3,4,5,6,7,8,9},//初始化 INT 数组
        {0,0,0,0,0,0,0,0,0},
              {0,10,0,0,0,0,0,11,0},
        {12,0,13,0,14,0,15,0,16},
        {0,0,0,0,0,0,0,0,0},
        {0,0,0,0,0,0,0,0,0},
        {28,0,29,0,30,0,31,0,32},
        {0,26,0,0,0,0,0,27,0},
        {0,0,0,0,0,0,0,0,0},
        {17,18,19,20,21,22,23,24,25}};

    protected String[][] word;
    public GameCanvas(){};
    public GameCanvas(Game game)//构造函数
    {
       this.game=game;

       empty=getWidth()/6;
       x=empty*1/3;
 canvasW=getWidth()-empty;
       mapWidth=canvasW-canvasW%8;
       cellWidth=mapWidth/8;
       a=cellWidth*2/5;
       b=cellWidth/8;
       c=cellWidth-a;
       d=cellWidth-b;
       chessR=cellWidth*2/5;
       selectedX=0;
       selectedY=0;
       guard=0;
       guard1=selectedX;guard2=selectedY;
```

```
     m=guard1;n=guard2;
     word=new String[10][9];
     g2=1;
     for(i=0;i<10;i++)//初始化字符数组
     {
         for(j=0;j<9;j++)
  {
if(i==0)
{
if(j==0){word[i][j]="车";}
if(j==1){word[i][j]="马";}
if(j==2){word[i][j]="相";}
if(j==3){word[i][j]="士";}
if(j==4){word[i][j]="帅";}
if(j==8){word[i][j]="车";}
if(j==7){word[i][j]="马";}
if(j==6){word[i][j]="相";}
if(j==5){word[i][j]="士";}
}
if(i==1){word[i][j]="空";}
if(i==2){
 if((j!=1)&(j!=7)){word[i][j]="空";}
 if(j==1){word[i][j]="炮";}
 if(j==7){word[i][j]="炮";}
 }
if(i==3){
 if(j%2==0){word[i][j]="卒";}
 if(j%2==1){word[i][j]="空";}
 }
if(i==4){word[i][j]="空";}
if(i==5){word[i][j]="空";}
if(i==6){
 if(j%2==0){word[i][j]="卒";}
 if(j%2==1){word[i][j]="空";}
 }
if(i==7){
 if((j!=1)&(j!=7)){word[i][j]="空";}
 if(j==1){word[i][j]="炮";}
 if(j==7){word[i][j]="炮";}
 }
if(i==8){word[i][j]="空";}
if(i==9)
{
if(j==0){word[i][j]="车";}
```

```
  if(j==1){word[i][j]="马";}
  if(j==2){word[i][j]="相";}
  if(j==3){word[i][j]="士";}
  if(j==4){word[i][j]="帅";}
  if(j==8){word[i][j]="车";}
  if(j==7){word[i][j]="马";}
  if(j==6){word[i][j]="相";}
  if(j==5){word[i][j]="士";}
  }

    }
       }
 exitCmd = new Command("退出", Command.EXIT, 0);

        addCommand(exitCmd);
        setCommandListener(this);
    }
    protected void paintMapa(Graphics g)//画河的上半部分的棋盘
    {
        for(int q=0;q<4;q++)
          {
           for(int w=0;w<8;w++)
              {
                g.setColor(128,128,128);
  g.drawRect(x+w*cellWidth,x+q*cellWidth,cellWidth,cellWidth);
              }
          }
         g.setColor(128,128,128);
  g.drawLine(x+3*cellWidth,x,x+5*cellWidth,x+2*cellWidth);
         g.drawLine(x+5*cellWidth,x,x+3*cellWidth,x+2*cellWidth);

         //画左上方的炮
         g.drawLine(x+d,x+cellWidth+c,x+d,x+cellWidth+d);//左上竖
         g.drawLine(x+c,x+cellWidth+d,x+d,x+cellWidth+d);//左上横

         g.drawLine(x+d+2*b,x+cellWidth+c,x+d+2*b,x+cellWidth+d);
//右上竖
         g.drawLine(x+cellWidth+b,x+cellWidth+d,x+cellWidth+a,x+c
ellWidth+d);//右上横

         g.drawLine(x+d,x+2*cellWidth+b,x+d,x+2*cellWidth+a);//左
下竖
         g.drawLine(x+c,x+cellWidth+d+2*b,x+d,x+cellWidth+d+2*b);
//左下横
```

```
         g.drawLine(x+d+2*b,x+2*cellWidth+b,x+d+2*b,x+2*cellWidth
+a);//右下竖
         g.drawLine(x+cellWidth+b,x+cellWidth+d+2*b,x+cellWidth+a
,x+cellWidth+d+2*b);//右下横

         //画右上方的炮

        g.drawLine(x+d+6*cellWidth,x+cellWidth+c,x+d+6*cellWidth,
x+cellWidth+d);
        g.drawLine(x+c+6*cellWidth,x+cellWidth+d,x+d+6*cellWidth,
x+cellWidth+d);


        g.drawLine(x+d+2*b+6*cellWidth,x+cellWidth+c,x+d+2*b+6*ce
llWidth,x+cellWidth+13+9);
        g.drawLine(x+cellWidth+b+6*cellWidth,x+cellWidth+d,x+cell
Width+a+6*cellWidth,x+cellWidth+d);

        g.drawLine(x+d+6*cellWidth,x+2*cellWidth+b,x+d+6*cellWidt
h,x+2*cellWidth+a);
        g.drawLine(x+c+6*cellWidth,x+cellWidth+d+2*b,x+d+6*cellWi
dth,x+cellWidth+d+2*b);

        g.drawLine(x+d+2*b+6*cellWidth,x+2*cellWidth+b,x+d+2*b+6*
cellWidth,x+2*cellWidth+a);
        g.drawLine(x+cellWidth+b+6*cellWidth,x+cellWidth+d+2*b,x+
cellWidth+a+6*cellWidth,x+cellWidth+d+2*b);
    }

    protected void paintMapb(Graphics g)//画那条河--楚河，哈哈
    {
       g.setColor(128,128,128);
       g.drawRect(x,x+4*cellWidth,mapWidth,cellWidth);
       g.setFont(Font.getFont(Font.FACE_PROPORTIONAL,Font.STYLE_B
OLD
                                                       ,Font.SIZE
_LARGE));
       g.drawString("楚河       汉界",getWidth()/2,x+4*cellWidth+
                                    cellWidth*3/4,Graphics.HCENTE
R|Graphics.BASELINE);
    }
    protected void paintMapc(Graphics g)//画河的下半部分的棋盘
    {
        for(int q=0;q<4;q++)
```

```
          {
           for(int w=0;w<8;w++)
              {
                g.setColor(128,128,128);
  g.drawRect(x+w*cellWidth,x+(q+5)*cellWidth,cellWidth,cellWidth);
              }
          }
         g.setColor(128,128,128);
  g.drawLine(x+3*cellWidth,x+7*cellWidth,x+5*cellWidth,x+9*cellWi
dth);
         g.drawLine(x+5*cellWidth,x+7*cellWidth,x+3*cellWidth,x+9
*cellWidth);

         //画左上方的炮
         g.drawLine(x+d,x+6*cellWidth+c,x+d,x+6*cellWidth+d);//左
上竖
         g.drawLine(x+c,x+6*cellWidth+d,x+d,x+6*cellWidth+d);//左
上横

         g.drawLine(x+d+2*b,x+6*cellWidth+c,x+d+2*b,x+6*cellWidth
+d);//右上竖
         g.drawLine(x+cellWidth+b,x+6*cellWidth+d,x+cellWidth+a,x
+6*cellWidth+d);//右上横

         g.drawLine(x+d,x+7*cellWidth+b,x+d,x+7*cellWidth+a);//左
下竖
         g.drawLine(x+c,x+6*cellWidth+d+2*b,x+d,x+6*cellWidth+d+2
*b);//左下横

         g.drawLine(x+d+2*b,x+7*cellWidth+b,x+d+2*b,x+7*cellWidth
+a);//右下竖
         g.drawLine(x+cellWidth+b,x+6*cellWidth+d+2*b,x+cellWidth
+a,x+6*cellWidth+d+2*b);//右下横

         //画右上方的炮

        g.drawLine(x+d+6*cellWidth,x+6*cellWidth+c,x+d+6*cellWidt
h,x+6*cellWidth+d);
        g.drawLine(x+c+6*cellWidth,x+6*cellWidth+d,x+d+6*cellWidt
h,x+6*cellWidth+d);


        g.drawLine(x+d+2*b+6*cellWidth,x+6*cellWidth+c,x+d+2*b+6*
cellWidth,x+6*cellWidth+d);
        g.drawLine(x+cellWidth+b+6*cellWidth,x+6*cellWidth+d,x+ce
llWidth+a+6*cellWidth,x+6*cellWidth+d);

        g.drawLine(x+d+6*cellWidth,x+7*cellWidth+b,x+d+6*cellWidt
h,x+7*cellWidth+a);
        g.drawLine(x+c+6*cellWidth,x+6*cellWidth+d+2*b,x+d+6*cell
Width,x+6*cellWidth+d+2*b);

        g.drawLine(x+d+2*b+6*cellWidth,x+7*cellWidth+b,x+d+2*b+6*
cellWidth,x+7*cellWidth+a);
        g.drawLine(x+cellWidth+b+6*cellWidth,x+6*cellWidth+d+2*b,
x+cellWidth+a+6*cellWidth,x+6*cellWidth+d+2*b);
    }

    protected void paintAllChess(Graphics g)//画出所有的棋子
    {
      for(i=0;i<10;i++)
      {
         for(j=0;j<9;j++)
  {
  if(point[i][j]!=0)
                {   if(point[i][j]<17){g.setColor(255,0,0);}
      else{g.setColor(255,255,255);}
  g.fillArc(x-chessR+j*cellWidth,x-chessR+i*cellWidth,2*chessR,2*
chessR,0,360);
  g.setColor(0x00000000);
  g.setFont(Font.getFont(Font.FACE_PROPORTIONAL,Font.STYLE_BOLD
                                                         ,Font.SIZE_
LARGE));
  g.drawString(word[i][j],x+j*cellWidth,x+chessR+i*cellWidth,Grap
hics.HCENTER|Graphics.BOTTOM);
             }

  }
       }
   }

   protected void chooseChess(Graphics g)//选定棋子，实现的原理就是如果
选择了就再按照指定的颜色
   {    m=guard1;n=guard2;                //再重新单独输出一个棋子
 if(point[guard2][guard1]!=0)
 {
   if(g2%2==1)
   {
  if(point[guard2][guard1]<=16)
  {
  g.setColor(255,255,0);
  g.fillArc(x-chessR+guard1*cellWidth,x-chessR+guard2*cellWidth,2
*chessR,2*chessR,0,360);
           g.setColor(0x00000000);
    g.setFont(Font.getFont(Font.FACE_PROPORTIONAL,Font.STYLE_BOLD
                                                     ,Font.SIZE_L
ARGE));
    g.drawString(word[guard2][guard1],x+guard1*cellWidth,x+chessR
+guard2*cellWidth,Graphics.HCENTER|Graphics.BOTTOM);
  }
   }
   if(g2%2==0)
   {
  if(point[guard2][guard1]>16)
  {
  g.setColor(0,255,0);
  g.fillArc(x-chessR+guard1*cellWidth,x-chessR+guard2*cellWidth,2
*chessR,2*chessR,0,360);
           g.setColor(0x00000000);
    g.setFont(Font.getFont(Font.FACE_PROPORTIONAL,Font.STYLE_BOLD
                                                     ,Font.SIZE_L
ARGE));
    g.drawString(word[guard2][guard1],x+guard1*cellWidth,x+chessR
+guard2*cellWidth,Graphics.HCENTER|Graphics.BOTTOM);
  }
   }
 }

   }
   protected void whoIsGoing(Graphics g)//判断该谁走了
   {
 checkWin();
 g.setFont(Font.getFont(Font.FACE_PROPORTIONAL,Font.STYLE_BOLD
                                                     ,Font.SIZE_L
ARGE));
        if(isRedWin!=0)
 {
  if(g2%2==1){
  g.setColor(255,0,0);
  g.drawString("该红方走了
",x,x+chessR+10*cellWidth,Graphics.LEFT|Graphics.BOTTOM);
  }
 }
 else{   g.setColor(255,255,255);
  g.drawString("白方胜利
",x,x+chessR+10*cellWidth,Graphics.LEFT|Graphics.BOTTOM);}
 if(isWhiteWin!=0)
 {
  if(g2%2==0){
  g.setColor(255,255,255);
  g.drawString("该白方走了
",x,x+chessR+10*cellWidth,Graphics.LEFT|Graphics.BOTTOM);
  }
 }
 else{   g.setColor(255,0,0);
  g.drawString("红方胜利
",x,x+chessR+10*cellWidth,Graphics.LEFT|Graphics.BOTTOM);}
   }
   protected void checkWin()//判断输赢
   { isRedWin=0;isWhiteWin=0;
 for(i=0;i<3;i++)
 {
  for(j=0;j<3;j++){if(point[0+i][3+j]==5){isRedWin++;}}
 }
 for(i=0;i<3;i++)
 {
  for(j=0;j<3;j++){if(point[7+i][3+j]==21){isWhiteWin++;}}
 }
   }

   protected void paintSelected(Graphics g)//画选择框
   {
        g.setColor(0,0,255);
        g.drawRect(x-chessR+selectedX*cellWidth,x-chessR+selected
Y*cellWidth,2*chessR,2*chessR);

   }

    protected void paint(Graphics g)
    {
        g.setColor(0x00000000);

        g.fillRect(0, 0, getWidth(), getHeight());

        paintMapa(g);
        paintMapb(g);
        paintMapc(g);
 paintAllChess(g);

 if(guard%2==1)
        {
   chooseChess(g);
 }

        paintSelected(g);
 whoIsGoing(g);
    }
    protected void changTwoChessNum(int m,int n,int selectedX,int
 selectedY)//改变两个格子的值
    {
 g2++;
 p=point[selectedY][selectedX];
        point[selectedY][selectedX]=point[n][m];
        point[n][m]=0;
        q=word[selectedY][selectedX];
        word[selectedY][selectedX]=word[n][m];
 word[n][m]="空";
    }
    protected void theRuleOfChe(int m,int n,int selectedX,int sel
ectedY)//车的规则
    {
                            g=0;
       if(m==selectedX)
       {
     if(n>selectedY)
     {
      for(i=1;i      {
       if(point[selectedY+i][m]!=0){g++;}
      }
     }
     else
     {
      for(i=1;i      {
       if(point[n+i][m]!=0){g++;}
      }
     }
     if(g==0){changTwoChessNum(m,n,selectedX,selectedY);}

       }
       if(n==selectedY)
       {
```

```
      if(m>selectedX)
     {
      for(i=1;i      {
       if(point[n][i+selectedX]!=0){g++;}
      }
     }
     else
     {
      for(i=1;i      {
       if(point[n][m+i]!=0){g++;}
      }
     }
     if(g==0){changTwoChessNum(m,n,selectedX,selectedY);}

       }
    }
    protected void theRuleOfMa(int m,int n,int selectedX,int sele
ctedY)//马的规则
    {
  if(n<9){
    if(point[n+1][m]==0)
    {
     if(selectedX-m==1){if(selectedY-n==2){changTwoChessNum(m,n,s
electedX,selectedY);}}
    }
  }
  if(n>0){
    if(point[n-1][m]==0)
    {
    if(m-selectedX==1){if(n-selectedY==2){changTwoChessNum(m,n,se
lectedX,selectedY);}}
    }
  }
  if(n<9){
    if(point[n+1][m]==0)
    {
     if(selectedX-m==-1){if(selectedY-n==2){changTwoChessNum(m,n,
selectedX,selectedY);}}
    }
  }
  if(n>0){
    if(point[n-1][m]==0)
    {
    if(m-selectedX==-1){if(n-selectedY==2){changTwoChessNum(m,n,s
electedX,selectedY);}}
    }
  }
        if(m<8){
    if(point[n][m+1]==0)
    {
     if(selectedX-m==2){if(selectedY-n==1){changTwoChessNum(m,n,selectedX,selectedY);}}
    }
  }
  if(m>0){
    if(point[n][m-1]==0)
    {
    if(m-selectedX==2){if(n-selectedY==1){changTwoChessNum(m,n,selectedX,selectedY);}}
    }
  }
  if(m<8){
    if(point[n][m+1]==0)
    {
     if(selectedX-m==2){if(selectedY-n==-1){changTwoChessNum(m,n,selectedX,selectedY);}}
    }
  }
  if(m>0){
    if(point[n][m-1]==0)
    {
    if(m-selectedX==2){if(n-selectedY==-1){changTwoChessNum(m,n,selectedX,selectedY);}}
    }
  }
    }
    protected void theRuleOfPao(int m,int n,int selectedX,int selectedY,int g1)//炮的规则
    {
      g=0;
        if(m==selectedX)
        {
       if(n>selectedY)
       {
         for(i=1;i         {
          if(point[selectedY+i][m]!=0){g++;}
         }
       }
       else
       {
         for(i=1;i         {
          if(point[n+i][m]!=0){g++;}
         }
       }
       if(g==g1){changTwoChessNum(m,n,selectedX,selectedY);}

       }
       if(n==selectedY)
       {
        if(m>selectedX)
       {
         for(i=1;i         {
          if(point[n][i+selectedX]!=0){g++;}
         }
       }
       else
       {
         for(i=1;i         {
          if(point[n][m+i]!=0){g++;}
         }
       }
       if(g==g1){changTwoChessNum(m,n,selectedX,selectedY);}

       }
    }
    protected void theRuleOfXiang(int m,int n,int selectedX,int selectedY)//相的规则
    {
  if(n<9&m<8){if(point[n+1][m+1]==0){if((selectedX-m==2)&(selectedY-n==2)){changTwoChessNum(m,n,selectedX,selectedY);}}}
  if(n>0&m<8){if(point[n-1][m+1]==0){if((selectedX-m==2)&(selectedY-n==-2)){changTwoChessNum(m,n,selectedX,selectedY);}}}
  if(n<9&m>0){if(point[n+1][m-1]==0){if((selectedX-m==-2)&(selectedY-n==2)){changTwoChessNum(m,n,selectedX,selectedY);}}}
  if(n>0&m>0){if(point[n-1][m-1]==0){if((selectedX-m==-2)&(selectedY-n==-2)){changTwoChessNum(m,n,selectedX,selectedY);}}}
    }
    protected void theRuleOfShi(int m,int n,int selectedX,int selectedY)//士的规则
    {
  if((m>2&m<6)&(selectedX>2&selectedX<6)&(n>=7&n<=9)&(selectedY>=7&selectedY<=9))
  {

   if((selectedX-m==1)&(selectedY-n==1)){changTwoChessNum(m,n,selectedX,selectedY);}

   if((selectedX-m==1)&(selectedY-n==-1)){changTwoChessNum(m,n,selectedX,selectedY);}
   if((selectedX-m==-1)&(selectedY-n==1)){changTwoChessNum(m,n,selectedX,selectedY);}
   if((selectedX-m==-1)&(selectedY-n==-1)){changTwoChessNum(m,n,selectedX,selectedY);}
  }
  if((m>2&m<6)&(selectedX>2&selectedX<6)&(n>=0&n<3)&(selectedY>=0&selectedY<3))
  {
   if((selectedX-m==1)&(selectedY-n==1)){changTwoChessNum(m,n,selectedX,selectedY);}
   if((selectedX-m==1)&(selectedY-n==-1)){changTwoChessNum(m,n,selectedX,selectedY);}
   if((selectedX-m==-1)&(selectedY-n==1)){changTwoChessNum(m,n,selectedX,selectedY);}
   if((selectedX-m==-1)&(selectedY-n==-1)){changTwoChessNum(m,n,selectedX,selectedY);}
  }
    }
    protected void theRuleOfShuai(int m,int n,int selectedX,int selectedY)//帅的规则
    {
  if((m>2&m<6)&(selectedX>2&selectedX<6)&(n>=7&n<=9)&(selectedY>=7&selectedY<=9))
  {
   if((selectedX-m==1)&(selectedY-n==0)){changTwoChessNum(m,n,selectedX,selectedY);}
   if((selectedX-m==-1)&(selectedY-n==0)){changTwoChessNum(m,n,selectedX,selectedY);}
   if((selectedX-m==0)&(selectedY-n==1)){changTwoChessNum(m,n,selectedX,selectedY);}
   if((selectedX-m==0)&(selectedY-n==-1)){changTwoChessNum(m,n,selectedX,selectedY);}
  }
  if((m>2&m<6)&(selectedX>2&selectedX<6)&(n>=0&n<3)&(selectedY>=0&selectedY<3))
  {
   if((selectedX-m==1)&(selectedY-n==0)){changTwoChessNum(m,n,sel
ectedX,selectedY);}
   if((selectedX-m==-1)&(selectedY-n==0)){changTwoChessNum(m,n,se
lectedX,selectedY);}
   if((selectedX-m==0)&(selectedY-n==1)){changTwoChessNum(m,n,sel
ectedX,selectedY);}
   if((selectedX-m==0)&(selectedY-n==-1)){changTwoChessNum(m,n,se
lectedX,selectedY);}
  }
    }
    protected void theRuleOfZu(int m,int n,int selectedX,int sele
ctedY)//卒的规则
    {
 if(point[n][m]<17)
  {
    if(selectedY>=n)
    {
      if(n<5)
      {
      if((selectedY-n==1)&(selectedX-m==0)){changTwoChessNum(m,n,
selectedX,selectedY);}
      }
      else
      {
      if((selectedY-n==1)&(selectedX-m==0)){changTwoChessNum(m,n,
selectedX,selectedY);}
      if((selectedY-n==0)&(selectedX-m==1)){changTwoChessNum(m,n,
selectedX,selectedY);}
      if((selectedY-n==0)&(selectedX-m==-1)){changTwoChessNum(m,n
,selectedX,selectedY);}
      }
    }
  }
  else
  {
    if(selectedY<=n)
    {
      if(n>4)
      {
      if((selectedY-n==-1)&(selectedX-m==0)){changTwoChessNum(m,n
,selectedX,selectedY);}
      }
      else
      {
      if((selectedY-n==-1)&(selectedX-m==0)){changTwoChessNum(m,n
,selectedX,selectedY);}
      if((selectedY-n==0)&(selectedX-m==1)){changTwoChessNum(m,n,
selectedX,selectedY);}
      if((selectedY-n==0)&(selectedX-m==-1)){changTwoChessNum(m,n
,selectedX,selectedY);}
      }
    }
  }

    }
       public void commandAction(Command c, Displayable d)
 {
         if (c == exitCmd) {
             game.destroyApp(false);
             game.notifyDestroyed();
         }
  }
    protected synchronized void keyPressed(int keyCode) //处理按键
    {
        int action = getGameAction(keyCode);

        if (action == Canvas.LEFT )
      {
            selectedX=(--selectedX+8+1)%(8+1);
        }
        else if (action == Canvas.RIGHT)
        {
              selectedX=(++selectedX)%(8+1);
        }
        else if (action == Canvas.UP)
        {
               selectedY=(--selectedY+9+1)%(9+1);
        }
        else if (action == Canvas.DOWN)
        {
            selectedY=(++selectedY)%(9+1);
        }
        else if (action == Canvas.FIRE)//这里的FIRE键我分成了两种情
况：一是选种棋子，
        {                              //二是当选择了棋子后，让棋子走到
下面选择的位置

                guard=guard+1;//每按下FIRE一次，GUARD就加一，用来判断
FIRE是被选种还是选种后走下不棋
```

```
    if(guard%2==1)  //这时是当选种某一个棋子时，调用 choosChess 函数，选择
棋子
    {
      if(point[selectedY][selectedX]!=0)
      {
      guard1=selectedX;
      guard2=selectedY;
      }

    }

    if(guard%2==0)//这种情况是当棋子被选种后
          {
                     if(point[selectedY][selectedX]!=point[n][m])
//当走的下一步不是自身，也就是玩家选过
         {                                            //一个棋子，又不
想选了，这只需什么都不做
                         if((point[n][m]==1)|(point[n][m]==9)|(po
int[n][m]==17)|(point[n][m]==25))//当选定的棋子是车的时候
            {                                       //repaint 就 OK
了
         if(point[selectedY][selectedX]==0)//当下一步走的是空格，则改变
选种的格子和下一步所
           {                                  //走的格子的 point[][]
和 word[][]的植，然后 repaint 就 OK
            theRuleOfChe(m,n,selectedX,selectedY);
         }
        else//当下一步是想吃对方的子的，则把下一步格子的值变为刚才选定的格子
的值，而
        {                               //刚才选定的格子的值则便为零
               if((point[selectedY][selectedX]/17)!=(point[n][m]/
17))//当然，想吃的子不能是自己的
                 {
            theRuleOfChe(m,n,selectedX,selectedY);


                      }
              }
            }
      if((point[n][m]==2)|(point[n][m]==8)|(point[n][m]==18)|(poi
nt[n][m]==24))//当选定的棋子是马的时候
      {
    if(point[selectedY][selectedX]==0)
        {
             theRuleOfMa(m,n,selectedX,selectedY);
        }
        else
        {
              if((point[selectedY][selectedX]/17)!=(point[n][m]/1
7))//当然，想吃的子不能是自己的
                  {
            theRuleOfMa(m,n,selectedX,selectedY);

                       }
        }
      }
      if((point[n][m]==10)|(point[n][m]==11)|(point[n][m]==26)|(p
oint[n][m]==27))//当选定的棋子是炮的时候
      {
        if(point[selectedY][selectedX]==0)
        {
     g1=0;
              theRuleOfPao(m,n,selectedX,selectedY,g1);
        }
        else
        {
     g1=1;
              if((point[selectedY][selectedX]/17)!=(point[n][m]/1
7))//当然，想吃的子不能是自己的
                  {
             theRuleOfPao(m,n,selectedX,selectedY,g1);

                       }
        }
      }
      if((point[n][m]==3)|(point[n][m]==7)|(point[n][m]==19)|(poi
nt[n][m]==23))//当选定的棋子是相的时候
      {
    if(point[selectedY][selectedX]==0)
        {
              theRuleOfXiang(m,n,selectedX,selectedY);
        }
        else
        {
              if((point[selectedY][selectedX]/17)!=(point[n][m]/1
7))//当然，想吃的子不能是自己的
                  {
            theRuleOfXiang(m,n,selectedX,selectedY);
```

```
                        }
            }
        }
        if((point[n][m]==4)|(point[n][m]==6)|(point[n][m]==20)|(poi
nt[n][m]==22))//当选定的棋子是士的时候
        {
    if(point[selectedY][selectedX]==0)
            {
                  theRuleOfShi(m,n,selectedX,selectedY);
            }
            else
            {
                  if((point[selectedY][selectedX]/17)!=(point[n][m]/1
7))//当然，想吃的子不能是自己的
                      {
                theRuleOfShi(m,n,selectedX,selectedY);

                        }
            }
        }
        if((point[n][m]==5)|(point[n][m]==21))//当选定的棋子是帅的时候
        {
    if(point[selectedY][selectedX]==0)
            {
                  theRuleOfShuai(m,n,selectedX,selectedY);
            }
            else
            {
                  if((point[selectedY][selectedX]/17)!=(point[n][m]/1
7))//当然，想吃的子不能是自己的
                      {
                 theRuleOfShuai(m,n,selectedX,selectedY);

                        }
            }
        }
        if((point[n][m]>11&point[n][m]<17))//当选定的棋子是红方卒的时候
        {
    if(point[selectedY][selectedX]==0)
            {
                  theRuleOfZu(m,n,selectedX,selectedY);
            }
            else
           {
                  if((point[selectedY][selectedX]/17)!=(point[n][m]/17))//当然，想吃的子不能是自己的
                      {
                theRuleOfZu(m,n,selectedX,selectedY);

                        }
            }
        }
        if(point[n][m]>27)//当选定的棋子是白方卒的时候
        {
    if(point[selectedY][selectedX]==0)
            {
                  theRuleOfZu(m,n,selectedX,selectedY);
            }
            else
            {
                  if((point[selectedY][selectedX]/17)!=(point[n][m]/17))//当然，想吃的子不能是自己的
                      {
                 theRuleOfZu(m,n,selectedX,selectedY);

                        }
            }
        }
           }
            }

         }

         repaint();

}
```

代码程序就 是这些，看起来很麻烦吧，如果仔细看了的话，其实很简单，只不过象棋不比五子棋，规则太多了，但是我的代码里有很多缺陷，例如由于采用了 canvas 而不 是 gamecanvas，所以在 paint()函数里面就会有很多的输出，这样在一些功能不强大的手机里会有闪屏的现象，我现在正在用 gamecanvas 写这个程序，大家先凑合看这个吧. 还有，由于象棋游戏的自身因素，所以导致了只能在大屏幕手机上使用，我这个是针对ＷＴＫ２２的默认 的彩色模拟器编的，而且都实现了，感觉效果还行，但是在一些其他的小屏手机上就不理想了，这个也没办法.

最后声明一下，大家可以拿我的代码转载，但毕竟是我的原创，请大家转载是把我的名字挂上-----dlut_608_#4,呵呵还有，代码太长了，没好好整理就发上来了，希望大家能耐心。

这个程序有很多需要赶紧的地方，希望大家看了以后多提宝贵意见.