第 2 卷◆第 3 期◆版本 1.0◆2024 年
文章类型：论文|刊号（ISSN）：2972-4236(P) / 2972-4244(O)

基于 Java 海战棋玩家与人工智能对战游戏程序

朱炫睿
南京一中
DOI:10.12238/acair.v2i3.8657

[摘　要] 本文介绍了基于Java语言开发的一款海战棋游戏系统的设计与实现,该系统是基于玩家与人工智能(AI)对战模式。游戏设计中采用了面向对象编程思想,实现了游戏界面的图形化显示、游戏规则的逻辑处理、以及基于简单策略的AI算法。本程序旨在通过实战演练,提升玩家的逻辑思维与策略布局能力,同时也展示了Java在游戏开发领域的应用潜力。
[关键词] Java编程；海战棋；人工智能；策略算法；面向对象编程
中图分类号：TP18　文献标识码：A

Based on the Java sea battle chess players and artificial intelligence battle game program

Xuanrui Zhu
Nanjing No.1 Middle School

[Abstract] This paper introduces the design and implementation of a sea battle chess game system developed based on Java language, which is based on the battle mode between players and artificial intelligence (AI). The object-oriented programming idea is adopted in the game design, realizing the graphical display of the game interface, the logical processing of the game rules, and the AI algorithm based on simple strategies. This program aims to improve the player's logical thinking and strategy layout ability through practical drills, and also demonstrates the application potential of Java in the game development field.
[Key words] Java programming; sea battle chess; artificial intelligence; strategy algorithm; object-oriented programming

引言

海战棋作为一款运行很久的策略游戏,以其简单的规则和丰富的策略性吸引了众多玩家。随着计算机技术的发展,将海战棋移植到数字平台,特别是引入人工智能对战,成为了新的发展趋势。

JAVA是一种简单易操作的面向对象程序设计语言,它继承了C++语言面向对象技术的核心,具有强大的功能,同时简单易懂,利于软件工程师遇到复杂编程问题得以开拓思维轻松实现,是一种应用广泛的、重要的网络编程语言。

广度优先遍历(BFS,Breadth-first Search)是一种图遍历算法。其从一个给定的节点出发,逐层扩展,先访问起始节点的相邻节点,再访问相邻节点的相邻节点,以此类推,遍历完所有的可达节点。基于该思想,可以实现简易的人工智能,使其可以在与玩家的对战中展现人类的思维,发现有目标后可以锁定,而不是随机落步棋盘[1]。

海战棋游戏运用广度优先遍历(BFS)遍历算法,通过玩家按程序指令生成玩家舰船,电脑按程序指令生成电脑AI舰船,初始化完成后,双方进入回合制攻击。先将对方所有船只击中者胜[2]。

1 系统架构

1.1模块

1.1.1用户界面(UI)模块

负责游戏的视觉呈现,包括游戏菜单、棋盘面布局、玩家信息显示、消息提示框等。可以使用Java Swing或JavaFX实现图形界面,以实现动态、交互友好的用户界面[3]。

1.1.2游戏逻辑模块

实现海战棋的核心游戏规则,包括舰船的放置规则、射击规则(命中、擦过失、命中、沉没)、游戏结束条件等[4]。

1.1.3数据持久化模块

实现游戏进度的保存与读取功能,允许玩家中断后继续游戏。

1.1.4人工智能(AI)模块

设计AI对手的决策算法,如基于规则的简单策略(随机射击、沿边角射击)、或更高级的搜索算法(如Minimax、Alpha-Beta剪枝)来决定射击位置[5]。

1.2实现

1.2.1船只的生成与管理

文章类型：论文|刊号（ISSN）：2972-4236(P) / 2972-4244(O)

用继承关系,三种船只cruiser,destroyer,airship 继承父类Ship。

```
public class Ship {
private String type;
private int times;
public Ship(String t,int a) {
 type = t;
 times=a;
}
public String getship() {
 return type;
}
public int hit() {
 times--;
 return times;
}
public int checkremain() {
 return times;
}
}
```

子类一：

```
public class Cruiser extends Ship{
 public Cruiser(String t,int a) {
      super(t,a);
 }
}
```

子类二：

```
public class Destroyer extends Ship{
public Destroyer(String d,int a) {
 super(d,a);
}
}
```

子类三：

```
public class Aircraft extends Ship{
public Aircraft(String a,int b) {
 super(a,b);
}
}
```

1.2.2玩家生成船只：

用Scanner类接受玩家输入,并且if… else 语句检查是否有不符合要求的情况。没有则正常执行,有的话重新执行当前方法,不影响之前船只的摆放。代码如下：

```
public void setcru1(){
//让玩家输入放置船的坐标和方向,并检查是否可以这样放会不会超出边界
//重合以放置的船
Scanner input = new Scanner(System.in);
/*
接受玩家输入的代码
*/
if(row<0 || row>=10 || col<0 || col>=10) {
System.out.println("out of bound");
setcru1();
//检查是否超出边界
} else {
 System.out.println("the direction to put cru1? left?right?up?down?");
String direction = inputstring.nextLine();
}
if(direction.equals("left")) {
//当方向为向左时执行的语句
}
else if(direction.equals("up")) {
//当方向为向上时执行的语句
}
 else if(direction.equals("right")) {
//当方向为向右时执行的语句
}
 else if(direction.equals("down")) {
//当方向为向下时执行的语句
}
else {
System.out.println("wrong direction");
setcru1();
//检查玩家输入是否正确
}
}
}
```

1.2.3电脑的攻击方法

用BFS遍历所有可能条件,如果没有任何船只击中,则随机攻击其余为打击的格子。如果打中任意船只,那接下来的攻击会落在该船只可能出现的线路上,以此排除试过的,不是想要结果的线路。

具体代码：

```
public void activeattack() {
if(direction.size()>0) {
if(row1!=-1) {
if(player[row1][col1]!=null) {
if(player[row1][col1].checkremain()==0) {
row1=-1;col1=-1;
reset();
activeattack();
```

```
    }
    else {
    if(f1==false) {
    //此时判断当前位置是否在地图边缘,是的话去除向地图边界的方向
    }
    String t =direction.get(0);
    if(t=="left") {
    if(col1-1>=0) {
    if(playerf[row1][col1-1]=="nullf") {
    //射击过此格子,去除当前方向,重新执行次方法
    }
    else
if(col1==0&&player[row1][col1].checkremain()!=0) {//打到了边界,但是没击沉,此时修改col1的值,使//得下次攻击击中剩余部分(在此条件下应当在右边)
    }
    else {
    String get = attack(row1,col1-1);
    f1=true;
    if(get.equals("empty")) {
    //打空了
    direction.remove(0);
    } else {
    //打中了,但不是同一艘船
    //row2,cow3,row4谁的值为-1,便将其值设定为此时打空的坐标,并去除不可能的方向
    } else {
    //打中了同一艘船
    row1=row1;
    col1=col1-1;
    direction.remove("up");direction.remove("down");
    }
    }
    }
    } else {
    if(player[row1][col1].checkremain()!=0) {//打到了边界,但是没击沉,此时修改col1的值,使//得下次攻击击中剩余部分
    }else {
    //超出了地图边界,去除此方向,重新执行
    direction.remove("left");
    activeattack();
    }
    }
    } else if(t=="up") {
    //方向为上的时候执行的语句
    } else if(t=="right") {
    //方向为右的时候执行的语句
    } else{
    //方向为右的时候执行的语句
    }
    else if(row2!=-1){
    ……
    }
    else if(row3!=-1){
    ……
    }
    else if(row4!=-1){
    ……
    }
    else{
    //当目前没有任何船只被击中,便随机射击地图上一个未射击过的格子
    }
```

2 结论

在本论文中,成功设计并实现了基于Java语言的海战棋游戏系统,该系统集成了玩家与人工智能(AI)对战的完整功能。通过精心设计的游戏架构,不仅实现了经典海战棋游戏的基本规则和流程,还创新性地融入了自适应难度的人工智能对手,极大地丰富了单人游戏模式的可玩性和挑战性。未来工作可以进一步优化AI算法,增加更多样化的游戏模式,以及探索云计算和大数据技术在游戏性能优化和用户体验个性化方面的应用,以适应更广泛的游戏市场和玩家需求。

[参考文献]

[1]朱金波.Java编程语言在计算机软件开发中的应用优势分析[J].信息记录材料,2023,24(5):68-70.

[2]唐莹.JAVA连连看游戏设计流程.内江科技[J].内江科技,2017,(10):38-39.

作者简介：

朱炫睿(2006--),男,汉族,江苏人,南京一中。