人机五子棋算法

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 课程名称 | Web程序设计 |  |  |  |
| 系 别 | 电信学院 |  | 实验报告日期 | 2018 年 11 月 2日 |
| 专业班号 | 电类815 |  |  |  |
| 姓 名 | 齐泽坤 |  | 学 号 | 2181311697 |
| 同 组 人 | 王瑞琪 |  |  |  |

机器博弈其实就是在打分，再把棋子下到分高的地方，比如下到一个眠四处，如果没有棋子，应给一个极高分，因为再下一个白棋子就赢了，但是如果下了一个黑棋子，那么，上面的第一种赢法无论如何也不可能了，所以应该直接设置为零分。所以在设置两个一位数组，记录每种赢法的得分，再查找棋盘上哪些位置可以实现这种赢法，给这些位置加分，最后从所有位置中，找出分最高的落子即可。而且，一种赢法上的落子越多，就越接近获胜，所以分数应该越高。

程序分为四个部分：棋盘绘制、赢法数组、分数统计、悔棋与撤销悔棋

1. **棋盘绘制部分**

采用canvas画布来设置布局

<style type='text/css'>

canvas {

display: block;

margin: 60px auto;

box-shadow: -2px -2px 2px #efefef, 5px 5px 5px #b9b9b9;

cursor: pointer;

}

.btn-wrap {

display: flex;

flex-direction: row;

justify-content:center;

}

.btn-wrap div {

margin: 0 10px;

}

div>span {

display: inline-block;

padding: 10px 20px;

color: #fff;

background-color: #6496ED;

border-radius: 5px;

cursor: pointer;

}

div.unable span {

background: #D6D6D4;

color: #adacaa;

}

#result-wrap {text-align: center;}

</style>

//绘画棋盘

var drawChessBoard = function() {

for(var i = 0; i < 15; i++) {

context.moveTo(15 + i \* 30, 15);

context.lineTo(15 + i \* 30, 435);

context.stroke();

context.moveTo(15, 15 + i \* 30);

context.lineTo(435, 15 + i \* 30);

context.stroke();

}

}

//画棋子

var oneStep=function(i, j, me) {

context.beginPath();

context.arc(15 + i \* 30, 15 + j \* 30, 13, 0, 2 \* Math.PI); // 画圆

context.closePath();

//渐变

var gradient=context.createRadialGradient(15 + i \* 30 + 2, 15 + j \* 30 - 2, 13, 15 + i \* 30 + 2, 15 + j \* 30 - 2, 0);

if(me) {

gradient.addColorStop(0, '#0a0a0a');

gradient.addColorStop(1, '#636766');

}

else {

gradient.addColorStop(0, '#d1d1d1');

gradient.addColorStop(1, '#f9f9f9');

}

context.fillStyle=gradient;

context.fill();

}

//销毁棋子

var minusStep=function(i, j) {

//擦除该圆

context.clearRect((i) \* 30, (j) \* 30, 30, 30);

// 重画该圆周围的格子

context.beginPath();

context.moveTo(15 + i \* 30, j \* 30);

context.lineTo(15 + i \* 30, j \* 30 + 30);

context.moveTo(i \* 30, j \* 30 + 15);

context.lineTo((i + 1) \* 30, j \* 30 + 15);

context.stroke();

}

1. **赢法数组**

//赢法的统计数组

var myWin = [];

var computerWin = [];

//赢法数组

var wins = [];

for(var i = 0; i < 15; i++) {

wins[i] = [];

for(var j = 0; j < 15; j++) {

wins[i][j] = [];

}

}

var count = 0; //赢法总数

//横线赢法

for(var i = 0; i < 15; i++) {

for(var j = 0; j < 11; j++) {

for(var k = 0; k < 5; k++) {

wins[i][j + k][count] = true;

}

count++;

}

}

//竖线赢法

for(var i = 0; i < 15; i++) {

for(var j = 0; j < 11; j++) {

for(var k = 0; k < 5; k++) {

wins[j + k][i][count] = true;

}

count++;

}

}

//正斜线赢法

for(var i = 0; i < 11; i++) {

for(var j = 0; j < 11; j++) {

for(var k = 0; k < 5; k++) {

wins[i + k][j + k][count] = true;

}

count++;

}

}

//反斜线赢法

for(var i = 0; i < 11; i++) {

for(var j = 14; j > 3; j--) {

for(var k = 0; k < 5; k++) {

wins[i + k][j - k][count] = true;

}

count++;

}

}

**三、分数统计**

// 计算机下棋

var computerAI = function() {

var myScore = [];

var computerScore = [];

var max = 0;

var u = 0,v = 0;

for(var i = 0; i < 15; i++) {

myScore[i] = [];

computerScore[i] = [];

for(var j = 0; j < 15; j++) {

myScore[i][j] = 0;

computerScore[i][j] = 0;

}

}

for(var i = 0; i < 15; i++) {

for(var j = 0; j < 15; j++) {

if(chressBord[i][j] == 0) {

for(var k = 0; k < count; k++) {

if(wins[i][j][k]) {

if(myWin[k] == 1) {

myScore[i][j] += 200;

} else if(myWin[k] == 2) {

myScore[i][j] += 400;

} else if(myWin[k] == 3) {

myScore[i][j] += 2000;

} else if(myWin[k] == 4) {

myScore[i][j] += 10000;

}

if(computerWin[k] == 1) {

computerScore[i][j] += 220;

} else if(computerWin[k] == 2) {

computerScore[i][j] += 420;

} else if(computerWin[k] == 3) {

computerScore[i][j] += 2100;

} else if(computerWin[k] == 4) {

computerScore[i][j] += 20000;

}

}

}

if(myScore[i][j] > max) {

max = myScore[i][j];

u = i;

v = j;

} else if(myScore[i][j] == max) {

if(computerScore[i][j] > computerScore[u][v]) {

u = i;

v = j;

}

}

if(computerScore[i][j] > max) {

max = computerScore[i][j];

u = i;

v = j;

} else if(computerScore[i][j] == max) {

if(myScore[i][j] > myScore[u][v]) {

u = i;

v = j;

}

}

}

}

}

\_compi = u;

\_compj = v;

oneStep(u, v, false);

chressBord[u][v] = 2;//计算机占据位置

for(var k = 0; k < count; k++) {

if(wins[u][v][k]) {

computerWin[k]++;

\_myWin[k] = myWin[k];

myWin[k] = 6; //这个位置对方不可能赢了

if(computerWin[k] == 5) {

resultTxt.innerHTML = '遗憾，计算机赢了，继续加油哦！';

over = true;

}

}

}

if(!over) {

me = !me;

}

backAble = true;

returnAble = false;

var hasClass = new RegExp('unable').test(' ' + returnbtn.className + ' ');

if(!hasClass) {

returnbtn.className+=' '+'unable';

}

}

1. **悔棋与撤销悔棋**

// 悔棋

backbtn.onclick=function(e) {

if(!backAble) {

return;

}

over=false;

me=true;

returnbtn.className=returnbtn.className.replace(new RegExp("(\\s|^)unable(\\s|$)"), " ");

// 我，悔棋

chressBord[\_nowi][\_nowj]=0; //我，已占位置 还原

minusStep(\_nowi, \_nowj); //销毁棋子

for(var k = 0; k < count; k++) {

// 将可能赢的情况都减1

if(wins[\_nowi][\_nowj][k]) {

myWin[k]--;

computerWin[k] = \_compWin[k]; //这个位置对方可能赢

}

}// 计算机相应的悔棋

chressBord[\_compi][\_compj]=0; //计算机，已占位置 还原

minusStep(\_compi, \_compj); //销毁棋子

for(var k = 0; k < count; k++) {

// 将可能赢的情况都减1

if(wins[\_compi][\_compj][k]) {

computerWin[k]--;

myWin[k] = \_myWin[i]; //这个位置对方可能赢

}

}

resultTxt.innerHTML ='--人机五子棋--';

returnAble=true;

backAble=false;

}

// 撤销悔棋

returnbtn.onclick=function(e) {

if(!returnAble) {

return;

}

// 我，撤销悔棋

chressBord[\_nowi][\_nowj]=1; //我，已占位置

oneStep(\_nowi, \_nowj, me);

for(var k = 0; k < count; k++) {

if(wins[\_nowi][\_nowj][k]) {

myWin[k]++;

\_compWin[k] = computerWin[k];

computerWin[k] = 6; //这个位置对方不可能赢

}

if(myWin[k] == 5) {

resultTxt.innerHTML = '恭喜，你赢了！';

over = true;

}

}// 计算机撤销相应的悔棋

**附上总代码及分析**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>五子棋</title>

<style type='text/css'>

canvas {

display: block;

margin: 60px auto;

box-shadow: -2px -2px 2px #efefef, 5px 5px 5px #b9b9b9;

cursor: pointer;

}

.btn-wrap {

display: flex;

flex-direction: row;

justify-content:center;

}

.btn-wrap div {

margin: 0 10px;

}

div>span {

display: inline-block;

padding: 10px 20px;

color: #fff;

background-color: #6496ED;

border-radius: 5px;

cursor: pointer;

}

div.unable span {

background: #D6D6D4;

color: #adacaa;

}

#result-wrap {text-align: center;}

</style>

</head>

<body>

<h3 id="result-wrap">-----人机五子棋-----</h3>

<canvas id="chess" width="450px" height="450px" ></canvas>

<div class="btn-wrap">

<div id='restart' class="restart">

<span>重新开始</span>

</div>

<div id='goback' class="goback unable">

<span>悔棋</span>

</div>

<div id='return' class="return unable">

<span>撤销悔棋</span>

</div>

</div>

<script type="text/javascript" charset="utf-8">

var over=false;

var me=true; //我

var \_nowi=0,

\_nowj=0; //记录自己下棋的坐标

var \_compi=0,

\_compj=0; //记录计算机当前下棋的坐标

var \_myWin=[],

\_compWin=[]; //记录我，计算机赢的情况

var backAble=false,

returnAble=false;

var resultTxt=document.getElementById('result-wrap');

var chressBord=[]; //棋盘

for(var i = 0; i < 15; i++) {

chressBord[i] = [];

for(var j = 0; j < 15; j++) {

chressBord[i][j] = 0;

}

}

//赢法的统计数组

var myWin = [];

var computerWin = [];

//赢法数组

var wins = [];

for(var i = 0; i < 15; i++) {

wins[i] = [];

for(var j = 0; j < 15; j++) {

wins[i][j] = [];

}

}

var count = 0; //赢法总数

//横线赢法

for(var i = 0; i < 15; i++) {

for(var j = 0; j < 11; j++) {

for(var k = 0; k < 5; k++) {

wins[i][j + k][count] = true;

}

count++;

}

}

//竖线赢法

for(var i = 0; i < 15; i++) {

for(var j = 0; j < 11; j++) {

for(var k = 0; k < 5; k++) {

wins[j + k][i][count] = true;

}

count++;

}

}

//正斜线赢法

for(var i = 0; i < 11; i++) {

for(var j = 0; j < 11; j++) {

for(var k = 0; k < 5; k++) {

wins[i + k][j + k][count] = true;

}

count++;

}

}

//反斜线赢法

for(var i = 0; i < 11; i++) {

for(var j = 14; j > 3; j--) {

for(var k = 0; k < 5; k++) {

wins[i + k][j - k][count] = true;

}

count++;

}

}

// debugger;

for(var i = 0; i < count; i++) {

myWin[i] = 0;

\_myWin[i] = 0;

computerWin[i] = 0;

\_compWin[i] = 0;

}

var chess = document.getElementById("chess");

var context = chess.getContext('2d');

context.strokeStyle = '#bfbfbf';

//边框颜色

var backbtn = document.getElementById("goback");

var returnbtn = document.getElementById("return");

window.onload = function() {

drawChessBoard();

// 画棋盘

}

document.getElementById("restart").onclick = function() {

window.location.reload();

}

// 我，下棋

chess.onclick = function(e) {

if(over) {

return;

}

if(!me) {

return;

}

// 悔棋功能可用

backbtn.className=backbtn.className.replace(new RegExp("(\\s|^)unable(\\s|$)"), " ");

var x=e.offsetX;

var y=e.offsetY;

var i=Math.floor(x / 30);

var j=Math.floor(y / 30);

\_nowi=i;

\_nowj=j;

if(chressBord[i][j]==0) {

oneStep(i, j, me);

chressBord[i][j]=1; //我，已占位置

for(var k = 0; k < count; k++) {

// 将可能赢的情况都加1

if(wins[i][j][k]) {

// debugger;

myWin[k]++;

\_compWin[k] = computerWin[k];

computerWin[k] = 6; //这个位置对方不可能赢了

if(myWin[k] == 5) {

// window.alert('你赢了');

resultTxt.innerHTML = '恭喜，你赢了！';

over = true;

}

}

}

if(!over) {

me = !me;

computerAI();

}

}

}

// 悔棋

backbtn.onclick=function(e) {

if(!backAble) {

return;

}

over=false;

me=true;

returnbtn.className=returnbtn.className.replace(new RegExp("(\\s|^)unable(\\s|$)"), " ");

// 我，悔棋

chressBord[\_nowi][\_nowj]=0; //我，已占位置 还原

minusStep(\_nowi, \_nowj); //销毁棋子

for(var k = 0; k < count; k++) {

// 将可能赢的情况都减1

if(wins[\_nowi][\_nowj][k]) {

myWin[k]--;

computerWin[k] = \_compWin[k]; //这个位置对方可能赢

}

}// 计算机相应的悔棋

chressBord[\_compi][\_compj]=0; //计算机，已占位置 还原

minusStep(\_compi, \_compj); //销毁棋子

for(var k = 0; k < count; k++) {

// 将可能赢的情况都减1

if(wins[\_compi][\_compj][k]) {

computerWin[k]--;

myWin[k] = \_myWin[i]; //这个位置对方可能赢

}

}

resultTxt.innerHTML ='--人机五子棋--';

returnAble=true;

backAble=false;

}

// 撤销悔棋

returnbtn.onclick=function(e) {

if(!returnAble) {

return;

}

// 我，撤销悔棋

chressBord[\_nowi][\_nowj]=1; //我，已占位置

oneStep(\_nowi, \_nowj, me);

for(var k = 0; k < count; k++) {

if(wins[\_nowi][\_nowj][k]) {

myWin[k]++;

\_compWin[k] = computerWin[k];

computerWin[k] = 6; //这个位置对方不可能赢

}

if(myWin[k] == 5) {

resultTxt.innerHTML = '恭喜，你赢了！';

over = true;

}

}// 计算机撤销相应的悔棋

chressBord[\_compi][\_compj]=2; //计算机，已占位置

oneStep(\_compi, \_compj, false);

for(var k = 0; k < count; k++) {

// 将可能赢的情况都减1

if(wins[\_compi][\_compj][k]) {

computerWin[k]++;

\_myWin[k] = myWin[k];

myWin[k] = 6; //这个位置对方不可能赢

}

if(computerWin[k] == 5) {

resultTxt.innerHTML = '遗憾，计算机赢了，继续加油哦！';

over = true;

}

}

returnbtn.className +=' '+'unable';

returnAble=false;

backAble=true;

}

// 计算机下棋

var computerAI = function() {

var myScore = [];

var computerScore = [];

var max = 0;

var u = 0,v = 0;

for(var i = 0; i < 15; i++) {

myScore[i] = [];

computerScore[i] = [];

for(var j = 0; j < 15; j++) {

myScore[i][j] = 0;

computerScore[i][j] = 0;

}

}

for(var i = 0; i < 15; i++) {

for(var j = 0; j < 15; j++) {

if(chressBord[i][j] == 0) {

for(var k = 0; k < count; k++) {

if(wins[i][j][k]) {

if(myWin[k] == 1) {

myScore[i][j] += 200;

} else if(myWin[k] == 2) {

myScore[i][j] += 400;

} else if(myWin[k] == 3) {

myScore[i][j] += 2000;

} else if(myWin[k] == 4) {

myScore[i][j] += 10000;

}

if(computerWin[k] == 1) {

computerScore[i][j] += 220;

} else if(computerWin[k] == 2) {

computerScore[i][j] += 420;

} else if(computerWin[k] == 3) {

computerScore[i][j] += 2100;

} else if(computerWin[k] == 4) {

computerScore[i][j] += 20000;

}

}

}

if(myScore[i][j] > max) {

max = myScore[i][j];

u = i;

v = j;

} else if(myScore[i][j] == max) {

if(computerScore[i][j] > computerScore[u][v]) {

u = i;

v = j;

}

}

if(computerScore[i][j] > max) {

max = computerScore[i][j];

u = i;

v = j;

} else if(computerScore[i][j] == max) {

if(myScore[i][j] > myScore[u][v]) {

u = i;

v = j;

}

}

}

}

}

\_compi = u;

\_compj = v;

oneStep(u, v, false);

chressBord[u][v] = 2;//计算机占据位置

for(var k = 0; k < count; k++) {

if(wins[u][v][k]) {

computerWin[k]++;

\_myWin[k] = myWin[k];

myWin[k] = 6; //这个位置对方不可能赢了

if(computerWin[k] == 5) {

resultTxt.innerHTML = '遗憾，计算机赢了，继续加油哦！';

over = true;

}

}

}

if(!over) {

me = !me;

}

backAble = true;

returnAble = false;

var hasClass = new RegExp('unable').test(' ' + returnbtn.className + ' ');

if(!hasClass) {

returnbtn.className+=' '+'unable';

}

}

//绘画棋盘

var drawChessBoard = function() {

for(var i = 0; i < 15; i++) {

context.moveTo(15 + i \* 30, 15);

context.lineTo(15 + i \* 30, 435);

context.stroke();

context.moveTo(15, 15 + i \* 30);

context.lineTo(435, 15 + i \* 30);

context.stroke();

}

}

//画棋子

var oneStep=function(i, j, me) {

context.beginPath();

context.arc(15 + i \* 30, 15 + j \* 30, 13, 0, 2 \* Math.PI); // 画圆

context.closePath();

//渐变

var gradient=context.createRadialGradient(15 + i \* 30 + 2, 15 + j \* 30 - 2, 13, 15 + i \* 30 + 2, 15 + j \* 30 - 2, 0);

if(me) {

gradient.addColorStop(0, '#0a0a0a');

gradient.addColorStop(1, '#636766');

}

else {

gradient.addColorStop(0, '#d1d1d1');

gradient.addColorStop(1, '#f9f9f9');

}

context.fillStyle=gradient;

context.fill();

}

//销毁棋子

var minusStep=function(i, j) {

//擦除该圆

context.clearRect((i) \* 30, (j) \* 30, 30, 30);

// 重画该圆周围的格子

context.beginPath();

context.moveTo(15 + i \* 30, j \* 30);

context.lineTo(15 + i \* 30, j \* 30 + 30);

context.moveTo(i \* 30, j \* 30 + 15);

context.lineTo((i + 1) \* 30, j \* 30 + 15);

context.stroke();

}

</script>

</body>

</html>