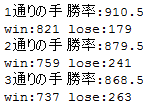
**ゲーム情報学特論　第六回　レポート**

## ○×ゲームを考える。先手は３通りの手があるが、それぞれ1000シミュレーションして、勝率を求めよ。

## 各シミュレーション中は着手がランダムに選択し、勝てば1勝、引き分けは0.5勝と考える。



２通りの手と３通りの手は勝率がだいたい同じである。

１通りの手はいちばん勝ちやすい。

ソースコード：

**package** WinCount;

**import** java.awt.Point;

**import** java.util.HashSet;

**import** java.util.Random;

**import** java.util.Set;

/\*\*

\* Make the random chess

\* **@author** s1710260

\*

\*/

**public** **class** RandomChess {

**public** **static** **int** *MaruChess* = 1;

**public** **static** **int** *BatsuChess* = 0;

**public** **static** **int** *COLS* = 3;

**public** **static** **int** *ROMS* = 3;

**public** **int**[][] chess = **new** **int**[*COLS*][*ROMS*];

**public** Set<Point> SameChess = **new** HashSet<Point>();

**public** Point ChessPoint = **new** Point();

/\*\*

\* make the random chess

\* **@param** type 3通りの手

\*/

**public** **void** getRandomChess(**int** type) {

// chess of random

Random rand = **new** Random();

**int** count\_batsu = 0;

**int** count\_maru = 0;

**int**[] temp = **new** **int**[*COLS* \* *ROMS* - 1];

**for** (**int** i = 0; i < *COLS* \* *ROMS* - 1; i++) {

temp[i] = rand.nextInt(2);

// System.out.println(temp[i]);

**if** (count\_maru == 4) {

temp[i] = 0;

}

**if** (count\_batsu == 4) {

temp[i] = 1;

}

**if** (temp[i] == 1) {

count\_maru++;

}

**if** (temp[i] == 0) {

count\_batsu++;

}

// System.out.println(temp[i]);

}

// ３通り

**switch** (type) {

**case** 1:

chess[0][0] = 1;

**int** count1 = 0;

**for** (**int** a = 0; a < *COLS*; a++) {

**for** (**int** b = 0; b < *ROMS*; b++) {

**if** (a == 0 && b == 0) {

**continue**;

} **else** {

chess[a][b] = temp[count1];

count1++;

}

}

}

**break**;

**case** 2:

chess[1][0] = 1;

**int** count2 = 0;

**for** (**int** a = 0; a < *COLS*; a++) {

**for** (**int** b = 0; b < *ROMS*; b++) {

**if** (a == 1 && b == 0) {

**continue**;

} **else** {

chess[a][b] = temp[count2];

count2++;

}

}

}

**break**;

**case** 3:

chess[1][1] = 1;

**int** count3 = 0;

**for** (**int** a = 0; a < *COLS*; a++) {

**for** (**int** b = 0; b < *ROMS*; b++) {

**if** (a == 1 && b == 0) {

**continue**;

} **else** {

chess[a][b] = temp[count3];

count3++;

}

}

}

**break**;

}

}

**public** **double** judgeChess(**int** x, **int** y) {

**if** (chess[x][y] == 1) {

// right direction

**int** count\_right = 0;

**for** (**int** i = 0; i < *COLS*; i++) {

**if** (chess[i][y] == chess[x][y]) {

count\_right++;

}

}

**if** (count\_right == 3) {

**return** 1;

}

// dip direction

**int** count\_dip = 0;

**for** (**int** i = 0; i < *COLS*; i++) {

**if** (x + i <= 2 && y + i <= 2) {

**if** (chess[x + i][y + i] == chess[x][y]) {

count\_dip++;

}

}

}

**if** (count\_dip == 3) {

**return** 1;

}

// down direction

**int** count\_down = 0;

**for** (**int** i = 0; i < *COLS*; i++) {

**if** (chess[x][i] == chess[x][y]) {

count\_down++;

}

}

**if** (count\_down == 3) {

**return** 1;

}

} **else** {

// right direction

**int** count\_right = 0;

**for** (**int** i = 0; i < *COLS*; i++) {

**if** (chess[i][y] == chess[x][y]) {

count\_right++;

}

}

**if** (count\_right == 3) {

**return** 0.5;

}

// dip direction

**int** count\_dip = 0;

**for** (**int** i = 0; i < *COLS*; i++) {

**if** (x + i <= 2 && y + i <= 2) {

**if** (chess[x + i][y + i] == chess[x][y]) {

count\_dip++;

}

}

}

**if** (count\_dip == 3) {

**return** 0.5;

}

// down direction

**int** count\_down = 0;

**for** (**int** i = 0; i < *COLS*; i++) {

**if** (chess[x][i] == chess[x][y]) {

count\_down++;

}

}

**if** (count\_down == 3) {

**return** 0.5;

}

}

**return** 0;

}

}

**package** WinCount;

**public** **class** CountNum {

**public** **static** **int** *TYPE\_1* = 1;

**public** **static** **int** *TYPE\_2* = 2;

**public** **static** **int** *TYPE\_3* = 3;

**public** **static** **void** main(String[] args) {

RandomChess rc= **new** RandomChess();

**double** win1 = 0;

**double** win2 = 0;

**double** win3 = 0;

**int** count1 = 0;

**int** count2 = 0;

**int** count3 = 0;

**int** count\_win1=0;

**int** count\_lose1=0;

**int** count\_win2=0;

**int** count\_lose2=0;

**int** count\_win3=0;

**int** count\_lose3=0;

**while**(count1<1000)

{

rc.getRandomChess(*TYPE\_1*);

**int** i = 0;

**while**(i<3){

**if**(rc.judgeChess(i, 0)==1||rc.judgeChess(0, i)==1){

win1+=1;

count\_win1++;

**break**;

}

i++;

}

**if**(i==3){

win1+=0.5;

count\_lose1++;

}

count1++;

};

System.*out*.println("1通りの手 勝率:"+win1);

System.*out*.println("win:"+count\_win1+" lose:"+count\_lose1);

**while**(count2<1000)

{

rc.getRandomChess(*TYPE\_2*);

**int** i = 0;

**while**(i<3){

**if**(rc.judgeChess(i, 0)==1||rc.judgeChess(0, i)==1){

win2+=1;

count\_win2++;

**break**;

}

i++;

}

**if**(i==3){

win2+=0.5;

count\_lose2++;

}

count2++;

};

System.*out*.println("2通りの手 勝率:"+win2);

System.*out*.println("win:"+count\_win2+" lose:"+count\_lose2);

**while**(count3<1000)

{

rc.getRandomChess(*TYPE\_3*);

**int** i = 0;

**while**(i<3){

**if**(rc.judgeChess(i, 0)==1||rc.judgeChess(0, i)==1){

win3+=1;

count\_win3++;

**break**;

}

i++;

}

**if**(i==3){

win3+=0.5;

count\_lose3++;

}

count3++;

};

System.*out*.println("3通りの手 勝率:"+win3);

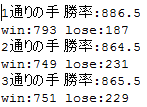
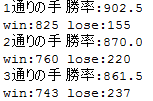
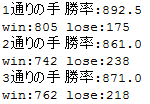
System.*out*.println("win:"+count\_win3+" lose:"+count\_lose3);

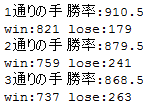
}

}

## 上記のプレーヤ同士を戦わせてみよ。シミュレーション数を20に減らしたプレーヤが弱くなるか確認せよ。

３回のシミュレーションの結果：



前と比べると

１通りの手と２通りの手はだいたい同じであるが、３通りの手は強くなる。

シミュレーションが多ければ多いほど、プレーヤが弱くなる。

## “シミュレーション中に、勝てる手があれば打ち、相手の勝ちを防ぐ手があれば打つ”ような「現実的なシミュレーション」を実装してみよ。そのようなプレーヤが強くなる確認せよ。