Projet battle Tron

Pseudo codingame : remii

Github : https://github.com/remi-dct/Projet.git

Code :

import java.util.\*;

import java.io.\*;

import java.math.\*;

/\*\*

 \* Auto-generated code below aims at helping you parse

 \* the standard input according to the problem statement.

 \*\*/

class Player {

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

**Scanner** in = new Scanner(**System**.**in**);

        int map[][] = new int[32][22];

        for(int c = 0; c<22; c++){ //boucle qui permet de délimiter les murs de la map

            map[0][c]=1;

            map[31][c]=1;

        }

        for(int r = 0; r<32; r++){ //boucle qui permet de délimiter les murs de la map

            map[r][0]=1;

            map[r][21]=1;

        }

        // game loop

        while (true) {

            int N = in.nextInt(); // total number of players (2 to 4).

            int P = in.nextInt(); // your player number (0 to 3).

            int positionX[] = new int[N];

            int positionY[] = new int[N];

            for (int i = 0; i < N; i++) {

                int X0 = in.nextInt(); // starting X coordinate of lightcycle (or -1)

                int Y0 = in.nextInt(); // starting Y coordinate of lightcycle (or -1)

                int X1 = in.nextInt(); // starting X coordinate of lightcycle (can be the same as X0 if you play before this player)

                int Y1 = in.nextInt(); // starting Y coordinate of lightcycle (can be the same as Y0 if you play before this player)

                map[X1+1][Y1+1]=1; //insère dans la map un "1" dans la case où se trouve la tête du serpent

                positionX[i]=X1;   //permet de récupérer la position de

                positionY[i]=Y1;   //la tête du serpent

            }

            boolean direction[] = new boolean[4]; // tableau de booléen qui va nous aider à determiner quelle direction prendre

            for (int n = 0; n<4; n++){

                direction[n] = true;

            }

            if(map[positionX[P]+1][positionY[P]]==1) //si la case du haut est égale à 1

                direction[0] = false;                 // alors direction[0] = false

            if(map[positionX[P]][positionY[P]+1]==1) // si la case de gauche est égale à 1

                direction[1] = false;                 // alors direction[1] = false

            if(map[positionX[P]+1][positionY[P]+2]==1) // si la case en dessous est égale à 1

                direction[2] = false;                   // alors direction[2] = false

            if(map[positionX[P]+2][positionY[P]+1]==1) // si la case à droite est égale à 1

                direction[3] = false;                   // alors direction [3] = false

            boolean jouer = true; // varibale booleen qui permet de déterminer si le joueur à jouer son tour (vrai = il n'a pas encore joué, faux il a joué)

            //conditions qui permettent d'orienter le serpent

            if(direction[0]&&jouer)

            {

**System**.**out**.println("UP");

                jouer = false;

            }

            if(direction[1]&&jouer)

            {

**System**.**out**.println("LEFT");

                jouer = false;

            }

            if(direction[2]&&jouer)

            {

**System**.**out**.println("DOWN");

                jouer = false;

            }

            if(direction[3]&&jouer)

            {

**System**.**out**.println("RIGHT");

                jouer = false;

            }

        }

    }

}