



Facultad de Ingeniería, Departamento de Sistemas e Industrial
Universidad Nacional de Colombia - Sede Bogotá

Búsqueda entre adversarios

Cuadrato

Alex Jose Alberto Barreto Cajicá
Daniel Arturo Moreno Rincón

6 de julio de 2020



Análisis del problema
Juego del cuadrado

Implementación
Primera parte
Segunda parte

Análisis del problema

Juego del cuadro



No es necesario dar como acción a jugar la línea número cuatro para completar el cuadro ni marcar el cuadro.

Para no dar ventaja al rival la acción que retornemos en principio solo debe dejar al cuadro elegido con máximo dos lados, al igual que al cuadro adyacente al lado elegido.

El juego debe dividirse en dos etapas, la primera cuando pueden elegirse líneas sin dar oportunidad al rival de marcar un cuadro, la segunda cuando ya por obligación la línea elegida dará oportunidad al rival de marcar cuadros en este punto se buscará darle la oportunidad de que marque la menor cantidad.



En esta parte también tenemos dos momentos importantes, elegir la posible acción a elegir y tratar de percibir las jugadas realizadas por el adversario.

De forma aleatoria se elige un cuadro y se evalúan sus cuatro lados para ver en cuales es posibles marcar una línea además se evalúan los cuadros adyacentes a los lados a donde se es posible poner una línea para no crear en el cuadro del lado tres lados.

```
if(l<2) {
    if(((String)p.getAttribute(i+":"+j+": "+Squares.LEFT)).equals(Squares.FALSE)) {
        if(j!=0) {
            int l2 = 0;
            if(cuadro.containsKey((String)(i+","+(j-1)))) {
                l2 = cuadro.get((String)(i+","+(j-1)));
            }
            if(l2<2) {
                v.add(Squares.LEFT);
            }
        }
    }
    else {
        laux++;
    }
}
```



Si en el cuadro elegido no se tiene el mismo número de opciones indica que el adversario marcó una línea en el cuadro con lo que se procede a evaluar los lados donde hay líneas y los cuadros adyacentes al lado para actualizar la información que tenemos de dichos cuadros.

```
if(l<laux ) {  
    if(((String)p.getAttribute(i+":j+":Squares.LEFT)).equals(Squares.TRUE)) {  
        if(j!=0) {  
            int l2 = 0;  
            int l3 = 0;  
            int iaux=i;  
            int jaux=j-1;  
            if(((String)p.getAttribute(iaux+":jaux+":Squares.LEFT)).equals(Squares.TRUE))  
                l3++;  
            if(((String)p.getAttribute(iaux+":jaux+":Squares.TOP)).equals(Squares.TRUE))  
                l3++;  
            if(((String)p.getAttribute(iaux+":jaux+":Squares.BOTTOM)).equals(Squares.TRUE))  
                l3++;  
            if(((String)p.getAttribute(iaux+":jaux+":Squares.RIGHT)).equals(Squares.TRUE))  
                l3++;  
        }  
    }  
}
```



Cuando ya en el tablero se tiene un caso cercano al punto de no poder elegir ninguna línea sin dar oportunidad al rival de que marque cuadros se evalúa de forma aleatoria lugares donde se puede elegir colocar una línea y los cuadros que se pueden marcar a partir de dicha línea.

```
String move = v.get((int)(Math.random()*v.size()));

evaluar(p, i, j, move);
if((Squares.LEFT).equals(move)) {
    evaluar(p, i, (j-1), Squares.RIGHT);
}
if((Squares.TOP).equals(move)) {
    evaluar(p, (i-1), j, Squares.BOTTOM);
}
if((Squares.BOTTOM).equals(move)) {
    evaluar(p, (i+1), j, Squares.TOP);
}
if((Squares.RIGHT).equals(move)) {
    evaluar(p, i, (j+1), Squares.LEFT);
}

jugadas.put((String)(i+", "+j+", "+move), afectados);
```

Implementación

Segunda parte



6

La función de evaluación parte del cuadro elegido obteniendo el número de líneas que sabemos tiene y evaluando si luego este se puede marcar continuamos evaluando en los otros tres lados y mientras se cumpla la condición de que con la adición de una línea se dan las condiciones para marcarlo se continúa y se va aumentando en uno los cuadros que son afectados y eliminando el cuadro del tablero de opciones.

```
public void evaluar(Percept p, int i, int j, String move) {
    String llave = i+" "+j;
    int l = 0;
    if(cuadroAux.containsKey(llave)) {
        l = cuadroAux.get(llave);
        l = l+1;
        cuadroAux.remove(llave);
        cuadroAux.put(llave,l);
        if(l>2) {
            cuadroAux.remove(llave);
            afectados = afectados+1;
            if(((String)p.getAttribute(i+" "+j+" "+Squares.LEFT)).equals(Squares.FALSE) && !(Squares.LEFT).equals(move)) {
                evaluar(p, i, (j-1), Squares.RIGHT);
            }
            if(((String)p.getAttribute(i+" "+j+" "+Squares.TOP)).equals(Squares.FALSE) && !(Squares.TOP).equals(move)) {
                evaluar(p, (i-1), j, Squares.BOTTOM);
            }
            if(((String)p.getAttribute(i+" "+j+" "+Squares.BOTTOM)).equals(Squares.FALSE) && !(Squares.BOTTOM).equals(move)) {
                evaluar(p, (i+1), j, Squares.TOP);
            }
            if(((String)p.getAttribute(i+" "+j+" "+Squares.RIGHT)).equals(Squares.FALSE) && !(Squares.RIGHT).equals(move)) {
                evaluar(p, i, (j+1), Squares.LEFT);
            }
        }
    }
}
```



Muchas Gracias!