<Àngel Contreras Rafanell>

<Dijous de 12:30 a 14:30>

<TQS-Practica-Buscaminas--ngel-Contreras-Rafanell-1633357>

Funcionalitat: Inicialitza valors, envia al tauler de valor, la primera posició, per així poder crear el tauler.

Localització: /BuscaMines/src/application/BuscaminesModel.java, BuscaminesModel, inicialitzaMatvalors

Test: /BuscaMines/src/application/BuscaminesModelTest.java, Model i inicialitzaMatvalors.

S'ha fet test de caixa negra, ja que li entren 2 paràmetres, i hem fet amb partitions equivalents, i pairwise testing (ja hi havia casos fets, així que no s'han repetit, pero si mencionats)

També de caixa blanca, si fem statement coverage, i contem els asserts com a condicions també hem fet decision i condition coverage.

```
public void inicialitzaMatvalors(int i, int j) {
    invariants();

    assert(i>=0 && this.llargada>i);
    assert(j>=0 && this.amplitud>j);
    TaulerV.initMat(i,j);

    invariants();
}
```

He aplicat DbyC.

I he utilitzat dos mocks objectes propis, MockTaulerDisp i MockTaulerVals. Que sub planten les classes taulerDisponibilitat i taulerValors.

Funcionalitat: GetValor retorna un array, d'arrays amb tres pocisions ([valor,fila,columna]), la cual la truca el controlador cuan vol descobrir una posició,i si s'envia 0, retornara mes d'una posició.

Localització: /BuscaMines/src/application/BuscaminesModel.java, Model, getValosr

Test: /BuscaMines/src/application/BuscaminesModelTest.java, Model i inicialitzaMatvalors.

Hem provat caixa negra, amb particions equivalents. I hem provat alguns casos d'ús de la clase, ex. com ha de repondre si es clica dos cops la mateixa posició, posicions invalides, pocions on el valor es 0 i posicions on el valor es 1.

També he fet caixa blanca, statement, condition i decision coverage.(Els ultims asserts son postcondicions de DbyC, i no s'ha testejat el cas perquè fallin).

He aplicat DbyC.

I he utilitzat dos mocks objectes propis, MockTaulerDisp i MockTaulerVals. Que sub planten les classes taulerDisponibilitat i taulerValors.

```java
public int[][] getValosr(int i, int j){
    invariants();
    assert(i>=0 && this.llargada>i);
    assert(j>=0 && this.amplitud>j);

    int[][] matV=TaulerV.getMat();
    boolean[][] matDisp = TaulerD.getMartrix();
    int[][] v = {{}};
    if(matDisp[i][j] == false) {

        TaulerD.posDescoberta(i, j);
        ArrayList<int[]> arrayBusca = new ArrayList<>();//es guarda tots el zeros, ja que es on es fara la cerca.
        ArrayList<int[]> arrayValors = new ArrayList<>();

        arrayValors.add(new int[] {matV[i][j],i,j});

        if(matV[i][j]==0) {//nomes farem la busqueda d'altres pocisions, si la inicial es 0

            arrayBusca.add(new int[] {i,j});
        }

        while(arrayBusca.size()>0 ) {
            //System.out.print(arrayBusca.size());
            int posi = arrayBusca.get(0)[0];
            int posj = arrayBusca.get(0)[1];

            arrayBusca.remove(0);
            //System.out.print(arrayBusca.size());
            for(int k=-1;k<2;k++) {
                //System.out.print("sssjjjjjj");
                for(int l=-1;l<2;l++) {
                    //System.out.print("lllllllllllllll");
                    int auxPosi = posi + k;
                    int auxPosj = posj + l;

                    if(auxPosi>=0 && this.llargada>auxPosi && auxPosj>=0 && this.amplitud>auxPosj) {

                        int val = matV[auxPosi][auxPosj];
                        int[] objectiu = new int[] {val,auxPosi,auxPosj};

                        boolean trobat = false;
                        for (int[] fila : arrayValors) {
                            if (Arrays.equals(fila, objectiu)) {
                            trobat = true;
                            break;
                            }
                        }
                        //System.out.println(val);
                        if(!trobat) {
                            TaulerD.posDescoberta(auxPosi,auxPosj);
                            arrayValors.add(objectiu);
                            if(val==0) {
                                arrayBusca.add(new int[] {auxPosi,auxPosj});
                            }
```

```
                            }

                        }


                    }

                }
            }
        }

        v = new int[arrayValors.size()][3];

        for(int k=0;k<arrayValors.size();k++) {
            for(int l=0;l<3;l++) {
                v[k][l] = arrayValors.get(k)[l];
            }
        }
    }

    invariants();

    if(v[0].length>0) {

        for(int k=0;k<v.length;k++) {
            //DbyC postCondicions
            assert(v[k][0]>-1 && v[k][0]<9 );
            assert(v[k][1]>=0 && this.llargada>v[k][1]);
            assert(v[k][2]>=0 && this.amplitud>v[k][2]);

        }

    }

    return v;
}
```

Funcionalitat: IsBomba, pregunta al tauler si aquella posició es una bomba.

Localització: /BuscaMines/src/application/BuscaminesModel.java, Model, isBomba

Test: /BuscaMines/src/application/BuscaminesModelTest.java, Model i testIsBomba.

Hem provat caixa negra, amb particions equivalents. I hem provat alguns casos d'ús de la clase, ex. seleccionar posició sense mina, amb mina, i posicions incorrectes.

També he fet caixa blanca, statement, condition i decision coverage.

He aplicat DbyC.

I he utilitzat dos mocks objectes propis, MockTaulerDisp i MockTaulerVals. Que sub planten les classes taulerDisponibilitat i taulerValors.

```
public boolean isBomba(int i, int j) {
    invariants();
    assert(i>=0 && this.llargada>i);
    assert(j>=0 && this.amplitud>j);

    int[][] matV=TaulerV.getMat();

    boolean res = (matV[i][j]==-1);

    invariants();
    return res; //valor temporal
}
```

Funcionalitat: posDescoberta, es truca cuan una pocisio ja esta descoberta. I genera la matriu si encara no estava generada.

Localització:/BuscaMines/src/application/BuscaminesTaulerDisp.java, BuscaminesTaulerDisp, posDescoberta.

Test: /BuscaMines/src/application/BuscaminesTaulerDispTest.java, BuscaminesTaulerDispTest i testPosDescoberta .

Hem provat caixa negra, amb particions equivalents, pairwise testing.

També he fet caixa blanca, statement, condition i decision coverage

He aplicat DbyC.

```
    @Override
    public void posDescoberta(int i, int j) {
        assert(i>=0 && j>=0 && i<this.Amplada && j<this.Llargada );

        if(!this.matGen) {
            this.matGen=true;
             this.mat = new boolean[this.Amplada][this.Llargada];
             for(int k=0;k<this.Amplada;k++) {
                 for(int l=0;l<this.Llargada;l++) {
                     this.mat[k][l]=false;
                 }
             }

        }


        this.mat[i][j]=true;
        // TODO Auto-generated method stub

    }
```

Funcionalitat: getMartrix retorna la matriu, si esta generada, i la genera si encara no existeix

Localització: /BuscaMines/src/application/BuscaminesTaulerDisp.java, BuscaminesTaulerDisp, getMartrix.

Test: /BuscaMines/src/application/BuscaminesTaulerDispTest.java, BuscaminesTaulerDispTest i testGetMartrix.

Hem provat caixa negra, amb particions equivalents, pairwise testing.

També he fet caixa blanca, statement, condition, decision coverage, loop testing(aniuats)

He aplicat DbyC.

```java
@Override
public boolean[][] getMartrix() {
    assert(this.Llargada>0 && this.Amplada>0);

    if(!this.matGen) {
        this.matGen=true;
         this.mat = new boolean[this.Amplada][this.Llargada];
         for(int i=0;i<this.Amplada;i++) {
             for(int j=0;j<this.Llargada;j++) {
                 this.mat[i][j]=false;
             }
         }
```

Funcionalitat: InitMat, inicialitza la matriu de valors donat una pocisio, i apartir de les mines, llargada i amplada donts en els setters.

Localització:/BuscaMines/src/application/BuscaminesTaulerValors.java, BuscaminesTaulerValors i initMat

Test:/BuscaMines/src/application/BuscaminesTaulerDispTest.java, BuscaminesTaulerValorsTest i testInitMat.

Hem provat caixa negra, amb particions equivalents, pairwise testing.

També he fet caixa blanca, statement, condition, decision coverage, loop testing: dos test de loop simples i un de anidat.

He aplicat DbyC.

I he utilitzat dos mocks objectes de mockito, de la clase Random. I així poder tenir les dades que necesito.

```java
@Override
public void initMat(int i, int j) {

    assert(this.Llargada>0);
    assert(this.Amplada>0);
    assert(this.mines>0);
    assert(i>=0 && j>=0 && i<this.Amplada && j<this.Llargada );
    //Thread.dumpStack();
    if(!matGen) {
        this.matGen=true;

        this.mat=new int[this.Amplada][this.Llargada];
        for(int t=0;t<this.Amplada;t++) {
            for(int y=0;y<this.Llargada;y++) {
                this.mat[t][y]=0;
            }
        }

        int[][] posMins = new int[this.mines][2];
        int k=0;

        while(k<this.mines) {//distribuim les mines

            int[] pos=this.Rand.Random(this.Amplada, this.Llargada);

            if(!isPosInArray(posMins,pos) && ((pos[0]!=i) || (pos[1]!=j))) {
                posMins[k][0]=pos[0];
                posMins[k][1]=pos[1];
                this.mat[pos[0]][pos[1]]=-1;

                k++;

            }

        }

        for(int l=0;l<this.Amplada;l++) {
            for(int h=0;h<this.Llargada;h++) {
                if(this.mat[l][h]==-1) {//com aquest pocico es una mina, el seu voltant li sumen 1
                    for(int t=(l-1);t<(l+2);t++) {
                        for(int y=(h-1);y<(h+2);y++) {
                            if(t>=0 && t<this.Amplada && y>=0 && y<this.Llargada) {
                                if(this.mat[t][y]!=-1 ) {
                                    this.mat[t][y] += 1;

                                }

                            }
                        }
                    }
                }
            }
        }

    }

}

private boolean isPosInArray(int[][] arr,int[] pos) {
    for(int i=0;i<arr.length;i++) {
        if(arr[i][0]==pos[0] && arr[i][1]==pos[1]) {
            return true;
        }
    }
    return false;
}
```


(Aquesta es un metode privat, que nomes el truca Init mat, i fa un loop simple)

Funcionalitat: Jugar, conecta amb el model, la vista i el teclat, i decideix que fer, depennt dels inputs del tecla, i de l'estat en què està amb les dades del model.

Localització:/BuscaMines/src/application/Controlador.java, Controlador i jugar().

Test: /BuscaMines/src/application/ControladorTest.java, ControladorTest, testJugar

De caixa negra he fet alguns casos d'ús. On es pot veure diferents situacions, tant portar-lo en un estat on guanya i perdi. També donar li valors invalids. I ficar li banderas,

També he fet caixa blanca, statement, condition, decision coverage.

He aplicat DbyC.

I he utilitzat dos mocks objectes de mockito, de la clase Random. I així poder tenir les dades que necesito.

```
27    public void jugar() {
28
29        vistaPlatilla();
30
31        while(true) {
32            int[] pos=getPos();
33
34
35            int accio = pos[0];
36            int fila = pos[1];
37            int columna = pos[2];
38
39            if(accio==0) {
40                //System.out.print(fila + " " + columna);
41                if(this.desc==0) {
42                    this.mod.inicialitzaMatvalors(fila, columna);
43                }
44                if(mat[fila][columna]==-3) {
45                    vis.mostrarError("No pots descobrir una cel·la marcada amb una bandera.");
46                }else if(mod.isBomba(fila, columna)) {
47                    vis.mostrarPerdut();
48                    vis.mostrarTauler(mod.getMatVals());
49
50                    break;
51                }else {
52
53                    int[][] valors = mod.getValosr(fila, columna);
54                    actualitzarMatriu(valors);
55
56                    if(haGuanyat()) {
57                        this.vis.mostrarGuanyat();
58                        break;
59                    }else {
60                        vistaPlatilla();
61                    }
62                }
63
64                }else {
65                    if(mat[fila][columna]==-3) {
66                        mat[fila][columna]=-2;
67                    }else {
68                        mat[fila][columna]=-3;
69                    }
70
71                    vistaPlatilla();
72                }
73            }
74
75        }
```

```java
    private int[] getPos(){
        int[] pos;

        while(true) {
            pos = this.tec.getPosTeclat();
            if(pos.length==3) {
                if(pos[0]>1 ||pos[0]<0||pos[1]>this.mod.getAmplada()-1 || pos[1]<0 ||pos[2]>this.mod.getLlargada()-1 || pos[2]<0 ) {
                    String msgError="El format ha de ser: Descobrir(0)/Bandera(1) fila columna. Ex 1 2.";
                    String msgError2="La fila ha de ser un numero del 0 al " + (mod.getAmplada()-1) ;
                    String msgError3="La columna ha de ser un numero del 0 al " + (mod.getLlargada()-1) ;
                    this.vis.flushNumLins(2);
                    this.vis.mostrarError(msgError);
                    this.vis.mostrarError(msgError2);
                    this.vis.mostrarError(msgError3);
                    this.vis.demanrPos();
                }else {
                    return pos;
                }

            }

        }
    }

    private void actualitzarMatriu(int[][] valors) {

        for (int i=0;i<valors.length;i++) {
            if(valors[i].length==3) {
                int fila = valors[i][1];
                int columna = valors[i][2];
                int valorCelda = valors[i][0];

                assert fila >= 0 && fila < this.mat.length ;
                assert columna >= 0 && columna < this.mat[0].length;
                if(this.mat[fila][columna]!=valorCelda) {
                    this.mat[fila][columna] = valorCelda;
                    this.desc++;
                }


            }

        }


}
```