

Problema Llums apagats. Extracte de solució

(Problema de l'examen de MTP-gestió de juny-2003)

1 Fitxer *tauler.txt*

Classe: TaulerLlums

Objectiu: Modelitza un tauler per jugar al joc de les llums apagades. El tauler pot tenir un màxim de 100 fileres i 100 columnes.

Operacions:

- `TaulerLlums();`
`TaulerLlums t;`
Post: Retorna un tauler de 100*100 i amb totes les caselles apagades.
- `TaulerLlums(int nfil, int ncol);`
`TaulerLlums t(nfil,ncol);`
Post: Retorna un tauler de nfil fileres *ncol columnes i amb totes les caselles apagades. Si nfil>100 o ncol>100 es comporta com la constructora buida.
- `void inserirLlum(int f, int c, bool& err);`
`t1.inserirLlum(f,c,err);`
Post: Marca la casella (f,c) (filera f i columna c) de t1 com a encesa i err=fals
Si aquesta casella no existeix al tauler, t1=t1' i err=cert
- `void ferClick(int f, int c);`
`t1.ferClick(f,c);`
Post: Canvia l'estat de la casella (f,c) (filera f i columna c) de t1 i de les caselles veïnes, si existeixen.
Si la casella (f,c) no existeix al tauler, t1=t1' i no fa res més (no dona cap avís d'error)
- `int getEstat(int f, int c);`
Feu-la com a exercici. Noteu que retorna -1 si (f,c) no existeix.
- `int getNFils();`
Feu-la com a exercici.
- `int getNCols();`
Feu-la com a exercici.

2 Fitxer *tauler.h*

```
const int MAXCOL=100;

const int MAXFIL=100;

class TaulerLlums{

    bool tauler[MAXFIL][MAXCOL];

    //tauler[i][j] =0: Posicio apagada
    //              =1: P0sicio encesa

    int nfiles;
    int ncolumns;

    void canvi(int f, int c);

public:

    TaulerLlums();
    TaulerLlums(int nfil, int ncol);
    void inserirLlum(int f, int c, bool& err);

    void ferClick(int f, int c);
    int getEstat(int f, int c);
    int getNFils();
    int getNCols();
};
```

3 Fitxer *tauler.cpp*

```
#include "tauler.h"

TaulerLlums::TaulerLlums(){//Exercici}

TaulerLlums::TaulerLlums(int nfil, int ncol){
    int i,j;

    if (nfil>MAXFIL || ncol>MAXCOL){
        nfil=MAXFIL; ncol=MAXCOL;
    }
    nfiles=nfil;
    ncolumns=ncol;
    for (i=0;i<nfiles;i++){
        for (j=0;j<ncolumns;j++){
            tauler[i][j]=0;
        }
    }
}
```

```

void TaulerLlums::inserirLlum(int f, int c, bool& err)
{
    if (f>=nfiles || c>=ncolumns || c<0 || f<0)
    {
        err=true;
    }
    else{
        err=false;
        tauler[f][c]=1;
    }
}

void TaulerLlums::ferClick(int f, int c)
{

    canvi(f,c);
    canvi(f+1,c);
    canvi(f-1,c);
    canvi(f,c+1);
    canvi(f,c-1);
}

void TaulerLlums::canvi(int f, int c)
{
    if (f<nfiles && c<ncolumns && c>=0 && f>=0)
    {
        tauler[f][c]=(tauler[f][c]+1)%2;
    }
}

int TaulerLlums::getEstat(int f, int c)
{
    if (f<nfiles && c<ncolumns && c>=0 && f>=0)
    {
        return tauler[f][c];
    }
    else return -1;
}

int TaulerLlums::getNFils()
{
    return nfiles;
}

int TaulerLlums::getNCols()
{
    return ncolumns;
}

```

4 Fitxer *tauler.cpp*

```
#include <iostream>
#include "tauler.h"

void obtNovaCasella(int& f,int& c,bool& final);
void obtenirTaulellIni(TaulerLlums& t1);
void marcarCanvis(TaulerLlums& t1);
void visualitzarTaulell(TaulerLlums& t1);

int main()
{
    int nfil, ncol;

    cin>>nfil;
    cin>>ncol;

    TaulerLlums t1(nfil,ncol);

    obtenirTaulellIni(t1);
    marcarCanvis(t1);
    visualitzarTaulell(t1);
    return 0;
}

void obtenirTaulellIni(TaulerLlums& t1)
{
    int f, c;
    bool final,er;

    obtNovaCasella(f,c,final);

    while (!final){
        t1.inserirLlum(f,c,er);
        obtNovaCasella(f,c,final);
    }
}

void marcarCanvis(TaulerLlums& t1)
{
    int f,c;
    bool final;

    obtNovaCasella(f,c,final);
    while (!final){
        t1.ferClick(f,c);
        obtNovaCasella(f,c,final);
    }
}

void visualitzarTaulell(TaulerLlums& t1)
```

```

{
    bool algunaEncesa=false;
    int i,j;
    for (i=0; i<t1.getNFils();i++){
        for (j=0;j<t1.getNCols();j++){

            if (t1.getEstat(i,j)==1) {
                cout<<"Casella ("<<i
                    <<" " <<j<<" encesa"<<endl;
                algunaEncesa=true;
            }
        }
    }

    if (!algunaEncesa)
        {cout<<"Felicitats, has guanyat"<<endl;}
}

void obtNovaCasella(int& f,int& c,bool& final)
{
    cin>>f;
    if (f!=999){
        cin>>c;
        final=false;
    }
    else final=true;
}

```