

Programmazione Orientata agli Oggetti

Il Gioco della vita di John Conway

Libero Nigro

Un programma genetico

- Si considera una matrice **nxm** di caratteri rappresentante un foglio quadrettato (universo o mondo virtuale). Ogni quadretto può essere occupato o meno da un organismo (carattere '*' per la presenza, '.' per l'assenza). Partendo da una configurazione iniziale di organismi, essa evolve nel tempo (definizione di generazioni successive) in accordo alle seguenti regole genetiche:
 - un organismo sopravvive nella generazione successiva, se ha due o tre vicini
 - un organismo muore, cioè lascia la cella vuota nella generazione successiva, se ha più di tre o meno di due vicini
 - un organismo nasce in una cella precedentemente vuota, se la cella è circondata da tre organismi vicini.

Si vuole scrivere un programma che simuli il gioco della vita utilizzando una classe istanziabile **GiocoDellaVita** che mantiene il mondo virtuale su una matrice **mappa** (variabile di istanza) ed ammette (almeno) i seguenti metodi:

- costruttore, che riceve n ed m (nr righe e nr colonne della matrice di char) ed inizializza a vuoto tutte le celle
 - *public void aggiungiOrganismo(int i, int j)* che aggiunge un organismo ('*') nella cella <i,j>
 - *public void configuraRandom()* che assegna randomicamente il contenuto delle celle
 - *public void prossimaGenerazione()* che genera la prossima generazione a partire da quella attuale
 - *private int contaVicini(int i, int j)* che conta il numero degli organismi presenti nell'intorno della cella <i,j>
 - *public String toString()* che ritorna sotto forma di stringa il contenuto di mappa
- La classe GiocoDellaVita è opportuno che introduca una seconda variabile di istanza **nuovaMappa**, matrice nxm di caratteri, il cui contenuto è definito a partire da mappa a cura di prossimaGenerazione(). L'uso di nuovaMappa serve a garantire il sincronismo (simultaneità) nella definizione della prossima generazione: il futuro dipende strettamente dal presente e non dal futuro stesso! nuovaMappa, a fine metodo prossimaGenerazione(), deve diventare mappa (il futuro diventa il nuovo presente)

Gioco della vita

.	.	*	.	.	*	*
*
.	.	.	*	.	*	.
*	.	.	.	*	.	.
.

Conf Iniziale

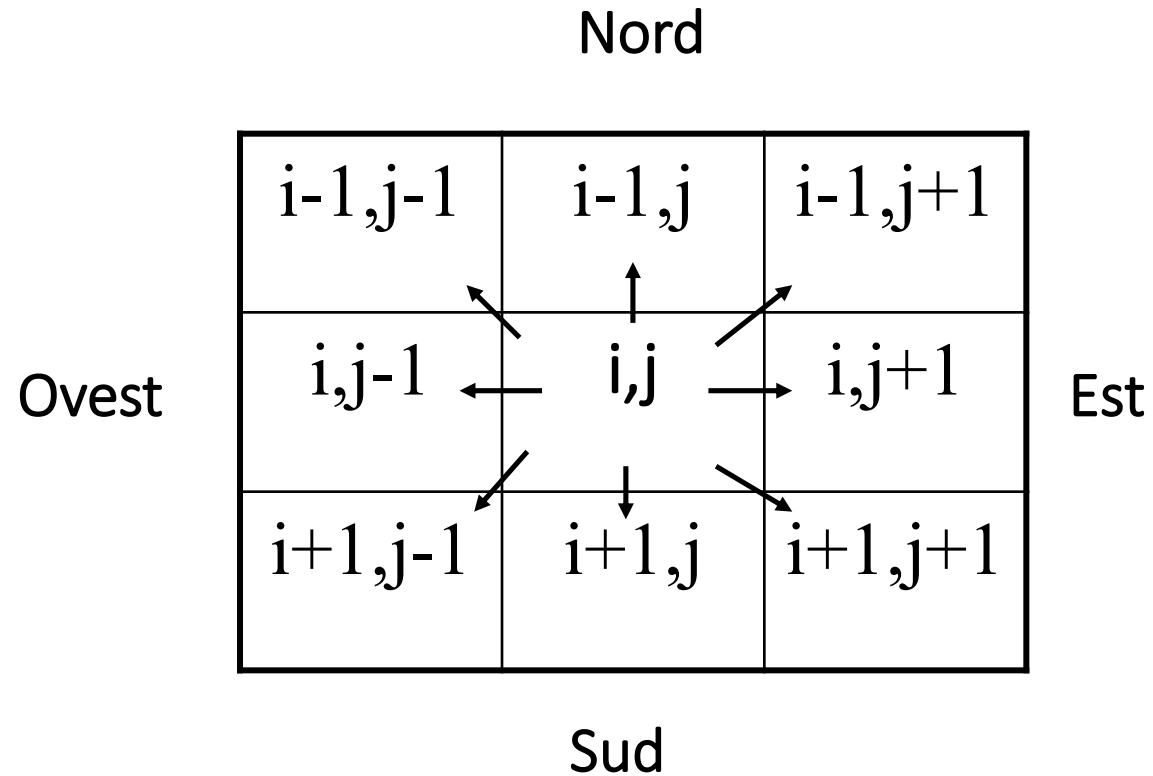
.
.	.	.	.	*	*	*
.	.	.	.	*	.	.
.	.	.	.	*	.	.
.

1 Generazione

.	*	.
.	.	.	.	*	*	.
.	.	.	*	*	.	.
.
.

2 Generazione

Calcolo dei vicini di $\langle i,j \rangle$



Occorre prestare particolare attenzione lungo i bordi

Codice Java

```
public class GiocoDellaVita{  
    private char [][]mappa;  
    private char [][]nuovaMappa;  
    private int n, m;  
    public GiocoDellaVita( int n, int m ){  
        if( n<1 || m<1 ) throw new IllegalArgumentException();  
        this.n=n; this.m=m;  
        mappa=new char[n][m];  
        nuovaMappa=new char[n][m];  
        for(int i=0; i<n; i++)  
            for(int j=0; j<m; j++)  
                mappa[i][j]='.';  
    }  
}
```

```
public void aggiungiOrganismo( int i, int j ){  
    if( i<0 || i>=n || j<0 || j>=m ) throw new IllegalArgumentException();  
    mappa[i][j]='*';  
} //aggiungiOrganismo
```

```
public void configuraRandom() {  
    for( int i=0; i<n; ++i )  
        for( int j=0; j<m; ++j ) {  
            if( Math.random()<0.5 ) mappa[i][j]='.';  
            else mappa[i][j]='*';  
        }  
} //configuraRandom
```

```

private int vicini( int i, int j ){
    int cont=0;
    if( i>0 && mappa[i-1][j]=='*' ) cont++; //NORD
    if( i>0 && j<m-1 && mappa[i-1][j+1]=='*' ) cont++; //NE
    if( j<m-1 && mappa[i][j+1]=='*' ) cont++; //EST
    if( i<n-1 && j<m-1 && mappa[i+1][j+1]=='*' ) cont++; //SE
    if( i<n-1 && mappa[i+1][j]=='*' ) cont++; //SUD
    if( i<n-1 && j>0 && mappa[i+1][j-1]=='*' ) cont++; //SO
    if( j>0 && mappa[i][j-1]=='*' ) cont++; //OVEST
    if( i>0 && j>0 && mappa[i-1][j-1]=='*' ) cont++; //NO
    return cont;
} //vicini

```



```

public void prossimaGenerazione() {
    for( int i=0; i<n; i++ )
        for( int j=0; j<m; j++ ){
            int v=vicini( i, j );
            if( mappa[i][j]=='*' )
                nuovaMappa[i][j]=( v==2 || v==3 ) ? '*' : '.';
            else
                nuovaMappa[i][j]=( v==3 ) ? '*' : '.';
        }

    //scambia mappa e nuovaMappa
    char[][] tmp=mappa; mappa=nuovaMappa; nuovaMappa=tmp;

} //prossimaGenerazione

```

```
public String toString(){  
    String s="";  
    for( int i=0; i<n; i++ ){  
        s=s+String.valueOf(mappa[i]);  
        s=s+"\n";  
    }  
    return s;  
} //toString
```

```
public static void main( String[] args ){  
    GiocoDellaVita gol=new GiocoDellaVita(10,15);  
    gol.configuraRandom();  
    for( int i=0; i<50; ++i ) {  
        System.out.println(gol);  
        gol.prossimaGenerazione();  
    }  
} //main  
} //GiocoDellaVita
```