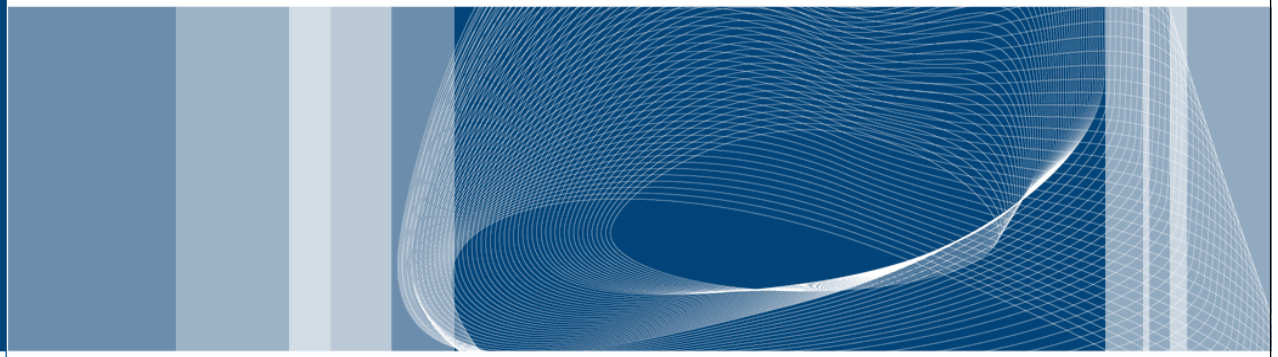
POLITECNICO DI MILANO



Fondamenti di Informatica 2013-2014

Game of Life

Paola Mussida
Area Servizi ICT

Esercizio F_21 - Game of Life

- Il gioco della vita mostra l'evoluzione di un mondo molto semplice costituito da cellule che possono essere vive o morte.
- Il campo di gioco è una matrice le cui celle possono contenere ciascuna un'unica cellula.
- Le regole dell'evoluzione sono le seguenti:
  - una cellula muore per **sovraffollamento** nel caso in cui abbia un numero di cellule vicine compreso tra 4 e 8;
  - una cellula muore per "**desertificazione**" nel caso in cui abbia 0 o 1 cellula vicina
  - una nuova cellula nasce se ha tre cellule vicine
  - una cellula rimane nel suo stato precedente nel caso abbia due cellule vicine

Esercizio F_21 - Game of Life

- Si sviluppi un programma che mostri all'utente l'evoluzione del gioco ad ogni passo.
- Si supponga che l'utente definisca lo stato iniziale del gioco:
  - indicando la dimensione del campo di gioco
  - identificando il numero e la posizione delle cellule vive
- ...e che possa controllare l'avanzamento del gioco e la sua terminazione inserendo da tastiera opportuni comandi

Esercizio F_21 - Game of Life

A. Inizializza lo scenario del gioco.

B. Assegna le celle che devono essere inizialmente vive nella posizione indicata dall'utente.

C. Visualizza lo scenario.

D. Calcola il nuovo scenario in base alle regole del gioco.

E. Chiedi all'utente se vuole visualizzare il nuovo scenario calcolato. In caso di risposta affermativa vai al punto C, altrimenti termina il programma.

Esercizio F_21 - Game of Life

```c
#include <stdio.h>

#define RIGHE 10
#define COLONNE 10

typedef enum {false, true} boolean;

typedef struct{
  int righe;
  int colonne;
  boolean matrice[RIGHE][COLONNE];
}scenario;
```

Esercizio F_21 - Game of Life - prototipi

Intestazioni delle funzioni individuate

```
void inizializza(scenario *scen);

void assegnaViteIniziali(scenario *scen, int viteIniziali);

void visualizza(scenario scen);

void aggiorna(scenario *scen);

int contaVicini(scenario scen, int riga, int colonna);
```

Esercizio F_21 - Game of Life

```
void main()
{
  int viteIniziali;
  char scelta;
  scenario mioScenario;
  boolean continua=true;

  inizializza(&mioScenario);
  printf("\nInserisci il numero delle vite iniziali: ");
  scanf("%d",&viteIniziali);
  assegnaViteIniziali(&mioScenario,viteIniziali);

                    /* ... */
```

```
/* ... */

  while (continua==true)
  {
     visualizza(mioScenario);
     aggiorna(&mioScenario);

     printf("\nVuoi continuare? (N per terminare) ");
     fflush(stdin);
     scanf("%c", &scelta);

     if (scelta=='n' || scelta=='N')
         continua=false;
   }
}
```

Esercizio F_21 - Game of Life


```
void inizializza(scenario *scen) {
  int i,j, righe, colonne;
  do{
          printf("\nMassimo numero di righe: %d ", RIGHE);
          printf("\nInserisci il numero delle righe: ");
          scanf("%d",&righe);
  }while (righe>RIGHE || righe <= 0) ;
  do{
          printf("\nMassimo numero di colonne: %d ", COLONNE);
          printf("\nInserisci il numero delle colonne: ");
          scanf("%d",&colonne);
  }while (colonne>COLONNE || colonne <= 0);
  scen->righe = righe;
  scen->colonne = colonne;

  for (i=0; i<scen->righe; i++)
          for (j=0; j<scen->colonne; j++)
             scen->matrice[i][j] = false;
}
```

Esercizio F_21 - Game of Life

```
void assegnaViteIniziali(scenario *scen, int viteIniziali) {

      int i,x,y;

      for (i=1; i<=viteIniziali; i++){
         do{
            printf("\nInserisci la cordinata X della %d
                                                  cellula: ",i);
            scanf("%d",&x);
         } while (x<0 || x>=scen->righe);
         do{
            printf("\nInserisci la cordinata Y della %d
                                                  cellula: ",i);
            scanf("%d",&y);
         } while (y<0 || y>=scen->colonne);

         scen->matrice[y][x] = true;
      }
}
```

Esercizio F_21 - Game of Life

```
void visualizza(scenario scen){
        int i,j;
        printf("\n");
        for (i=0; i<scen.righe; i++){
                for (j=0; j<=scen.colonne*4;j++)
                   printf("-");
                printf("\n|");
                for (j=0; j<scen.colonne; j++){
                   if (scen.matrice[i][j] == true)
                         printf(" * |");
                   else
                         printf ("   |");
                }
                printf("\n");
        }
        for (j=0; j<=scen.colonne*4;j++)
                printf("-");
        printf("\n");
}
```

Esercizio F_21 - Game of Life

```
int contaVicini(scenario s, int riga, int colonna) {

     int i,j;
     int viciniContati=0;

     for (i=riga-1; i<=riga+1; i++)
         for (j=colonna-1; j<=colonna+1; j++)
             if ( (i>=0) && (j>=0) && (i<s.righe)
                    && (j<s.colonne) )
                      if (s.matrice[i][j]==true)
                             viciniContati++;

     if (s.matrice[riga][colonna]==true)
             viciniContati--;

     return viciniContati;

}
```

Esercizio F_21 - Game of Life

```
void aggiorna(scenario *scen) {

      int i,j;
      scenario temp;
      int vicini;
      for (i=0; i<scen->righe; i++)
         for (j=0; j<scen->colonne; j++){
             vicini = contaVicini(*scen,i,j);
             if(vicini > 3 || vicini <2)
                  temp.matrice[i][j]=false;
             else if (vicini == 3)
                  temp.matrice[i][j]=true;
             else
                  temp.matrice[i][j]=scen->matrice[i][j];
         }
      for (i=0; i<scen->righe; i++)
         for (j=0; j<scen->colonne; j++)
             scen->matrice[i][j]=temp.matrice[i][j];

}
```