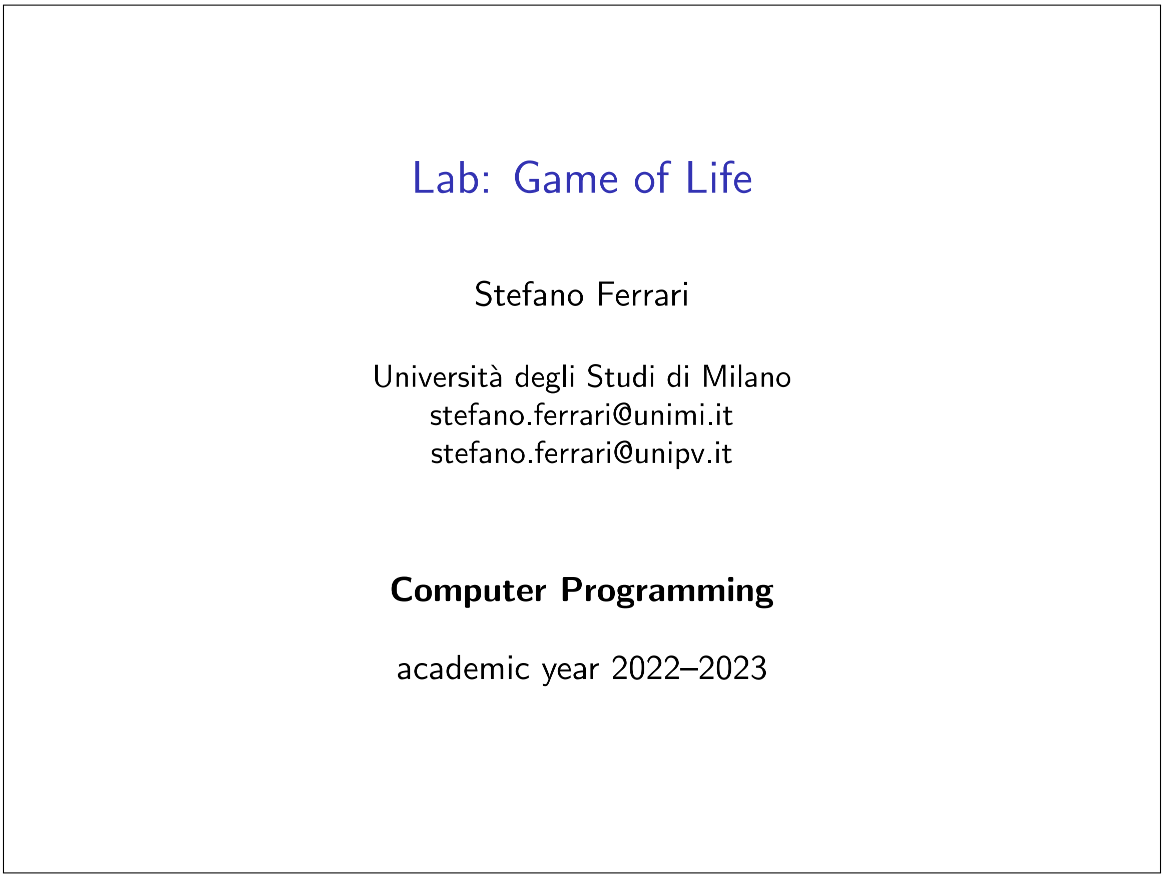


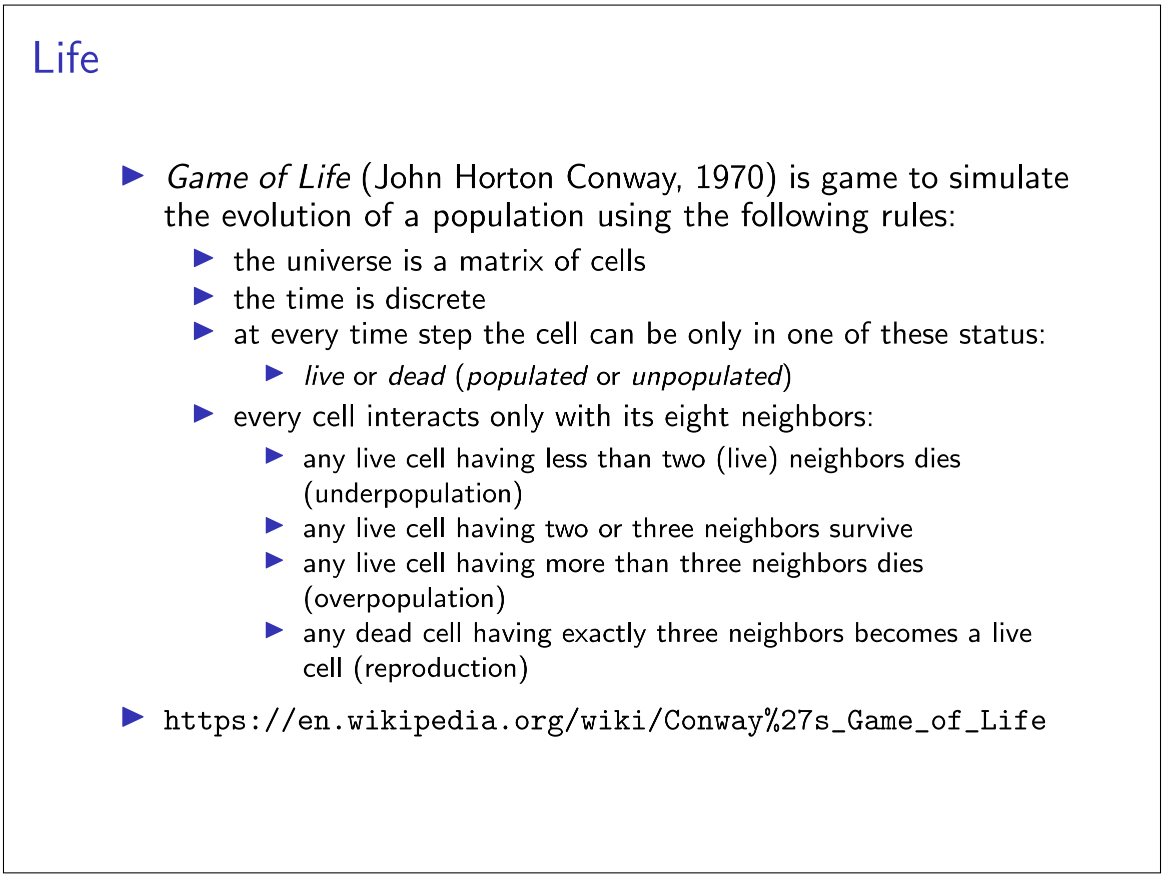
Lab: Game of Life

Stefano Ferrari

Università degli Studi di Milano
stefano.ferrari@unimi.it
stefano.ferrari@unipv.it

Computer Programming

academic year 2022–2023



Life

- *Game of Life* (John Horton Conway, 1970) is game to simulate the evolution of a population using the following rules:
  - the universe is a matrix of cells
  - the time is discrete
  - at every time step the cell can be only in one of these status:
    - *live* or *dead* (*populated* or *unpopulated*)
  - every cell interacts only with its eight neighbors:
    - any live cell having less than two (live) neighbors dies (underpopulation)
    - any live cell having two or three neighbors survive
    - any live cell having more than three neighbors dies (overpopulation)
    - any dead cell having exactly three neighbors becomes a live cell (reproduction)
- `https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life`

Development steps

- Realize the program in imperative style
  - the matrix is the main data structure
  - a function for the visualization
  - a function for the initialization
  - a function for the update
- Refactoring in OOP
  - the matrix becomes a class
  - the functions becomes methods
- Graphics
  - use `turtle` for the visualization
  - add a class for managing the graphics
  - a class for the cell?
    - the status
    - the list of the neighbors
    - the position on the screen

numpy

- Use `numpy.array` to represent the matrix
- Generate a random matrix:

```
import numpy as np
N = 100 status = [0, 1]
p = [0.8, 0.2]
np.random.choice(status, N*N, p).reshape(N, N)
```

  - `N` is the size of the matrix
  - `status` are the possible status of the cells
  - `p` are the probability of the status