

```

function NewBoard = Evolve(OldBoard,Rule)
%Evolve la board secondo la regola fissata, con condizioni periodiche
  al
%contorno

switch Rule
  case 'Life'
    Birth = 3;
    %Number of neighboring cells required for a dead cell to turn
      alive
    Survival = [2 3];
    %Number of neighboring cells required for an alive cell to
      survive
  case 'Fredkin' %Chaotic rule, fill percentage converges to 50%
    Birth = 1:2:7;
    Survival = 0:2:8;
end

PeriodicBoard = ...
  [OldBoard(end,end), OldBoard(end,:),OldBoard(end,1)
  OldBoard(:,end),OldBoard,OldBoard(:,1)
  OldBoard(1,end), OldBoard(1,:), OldBoard(1,1)];
%Last column will be counted as neighboring the first column, etc.

MooreNeighborhood = [1 1 1; 1 0 1; 1 1 1];
%Neighbors are positions reachable by a King
Neighbors = conv2(PeriodicBoard, MooreNeighborhood, 'same');
%Gives number of alive neighbors of each cell
NewBoard = ...
  ismember(Neighbors,Birth).*(1-PeriodicBoard)+...
  ismember(Neighbors,Survival).*PeriodicBoard;
%Modify a cell state according to how many alive neighbors it has

NewBoard = NewBoard(2:end-1,2:end-1);
%Removes the boundary copies
end

```