Conway’s game of life pseudocode

* Import modules (pygame, copy, sys, time)
* Make constants: W (width), H (height), ALIVE (living cell color), DEAD (dead cell color)
* Create a draw\_cells function that takes a dict of the cells as an argument
  + For each cell in the cells:
    - If it is alive, draw an alive cell colored square.
    - If it is dead, draw a dead cell colored square.
* Create a calculate\_next\_cells function that takes a dict of the cells as an argument
  + For each cell in the cells:
    - Get the neighboring coordinates of the position even if they wrap around the edge
    - Set the number of neighbors to 0
    - Count the number of living neighbors
    - Set cells based on rules
      * Each cell with one or no neighbors dies, as if by solitude.
      * Each cell with four or more neighbors dies, as if by overpopulation.
      * Each cell with two or three neighbors survives.
      * Each empty space with three neighbors will spawn a new cell.
* Initiate the Pygame window
* Repeat these steps:
  + Event handling loop:
    - Quit on exit button or esc key
    - Toggle alive/dead cell when cell is clicked
    - Exit the loop and start the simulation with space key
    - Draw the cells
  + Make a “next\_cells” dict[[1]](#footnote-1) and set it to the cells selected in the previous loop
  + Main simulation loop:
    - Quit on exit button or esc key
    - Copy the next\_cells dict into the cells dict[[2]](#footnote-2)
    - Draw the cells
    - Calculate the next step’s cells

1. dictionary [↑](#footnote-ref-1)
2. Using deepcopy [↑](#footnote-ref-2)