CA - Game of Life

Kate Cendrowski
kcendro2@pdx.edu

Research Experiences for Undergraduates (REU)
Computational Modeling   Serving the City
Cellular Automata - Game of Life

- Modified ca_example.py from day four
  - Randomized input
  - From three cases to one (handling edge cases)
  - 1D to 2D
  - Implemented Game of Life rules

Not mine, just from Google!

https://upload.wikimedia.org/wikipedia/commons/thumb/e/e0/Game_of_life_glider_gun.svg/610px-Game_of_life_glider_gun.svg.png
```python
import random

def ca():
    """ Cellular automata with Python - K. Hong - modified by K. Cendrowski"
    # 64 Boolean - Alive('1') / '0'
    # Rule - the status of current cell value is Alive
    # if only it is alive and two or three of it's Moore
    # neighborhood neighbors are Alive('1') OR it is dead
    # and exactly three of it's neighbors are alive
    # otherwise, the current cell status is dead('0')

    height = 4
    width = 5

    # list representing the current status of height x width cells
    ca = [[random.randint(0,1) for i in range(0,width)] for j in range(0,height)]

    # new Cellular values
    ca_new = ca

    # dictionary maps the cell value to a symbol
    dic = {'0': ' ', '1': 'x'}

    # initial draw = step 0
    for i in range(0,height):
        joined = 'x
        for j in range(0,width):
            joined += dic[ca_new[i][j]]
        print(joined)

    # additional 31 steps
    step = 1
    totalit = 8

    # inside cells - check the neighbor cell state
    for j in range(0,width):
        alive = 0
        # alive neighbors
        if(ca[i-1][j] + ca[i][j+1] + ca[i][j-1] + ca[i+1][j] == 2):
            alive += 1
        # upper left neighbor
        if(ca[i-1][j+1] == 1):
            alive += 1
        # upper right neighbor
        if(ca[i-1][j-1] == 1):
            alive += 1
        # upper neighbor
        if(ca[i-1][j] == 1):
            alive += 1
        # right neighbor
        if(ca[i][j+1] == 1):
            alive += 1
        # lower right neighbor
        if(ca[i][j-1] == 1):
            alive += 1
        # lower neighbor
        if(ca[i+1][j] == 1):
            alive += 1
        # lower left neighbor
        if(ca[i+1][j+1] == 1):
            alive += 1
        # center neighbor
        if(ca[i+1][j] == 1):
            alive += 1

    # less than two neighbors alive, cell dies
    if(alive < 2):
        ca_new[i][j] = 0
    # present cell and two or three neighbors alive, cell alive
    elif(alive == 2 or alive == 3):
        ca_new[i][j] = 1
    # more than three neighbors alive, cell dies
    else:
        ca_new[i][j] = 0
```

Output

8 x 8 over 4 iterations
Initial

Final

4 x 5 over 12 iterations
Initial

Final