The Game of Life

Rules:

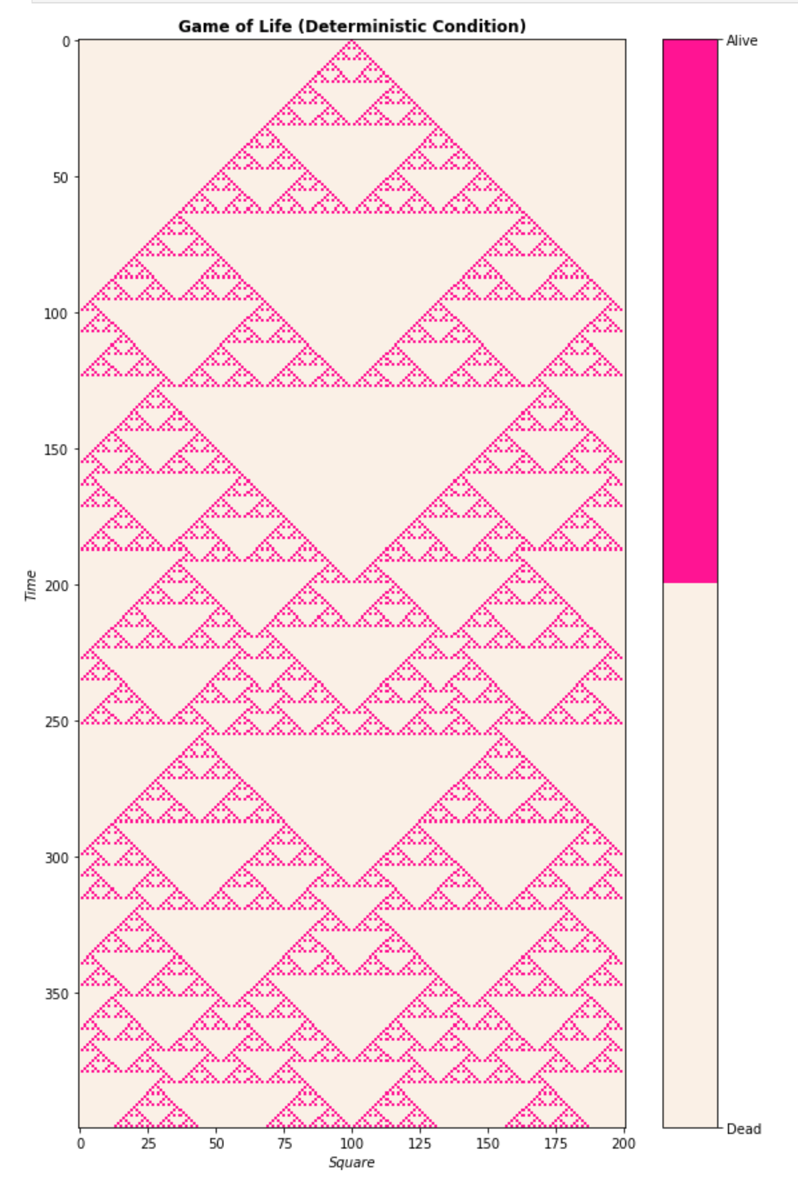
1. If the box has only one living neighbor, then the box in the next row (in that same column) lives.
2. If the box has no living neighbors, that is, it has 2 dead neighbors, then the box in the next row (in that same column) dies.
3. If the square has 2 living neighbors, then the square in the next row (in that same column) dies.

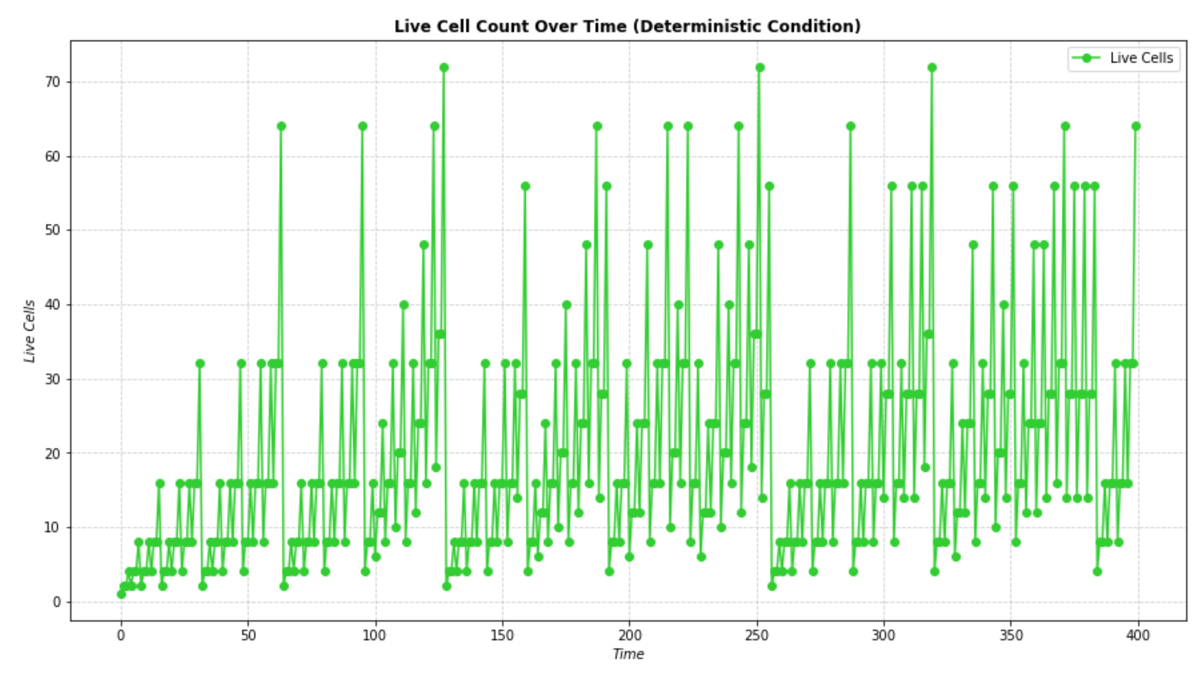
Solution:

- Computational solution to Conway's game of life (the 1D version) on a 200-square board with 400 time steps. Considering two options:
  - a) Deterministic initial conditions.
  - b) Random initial conditions.

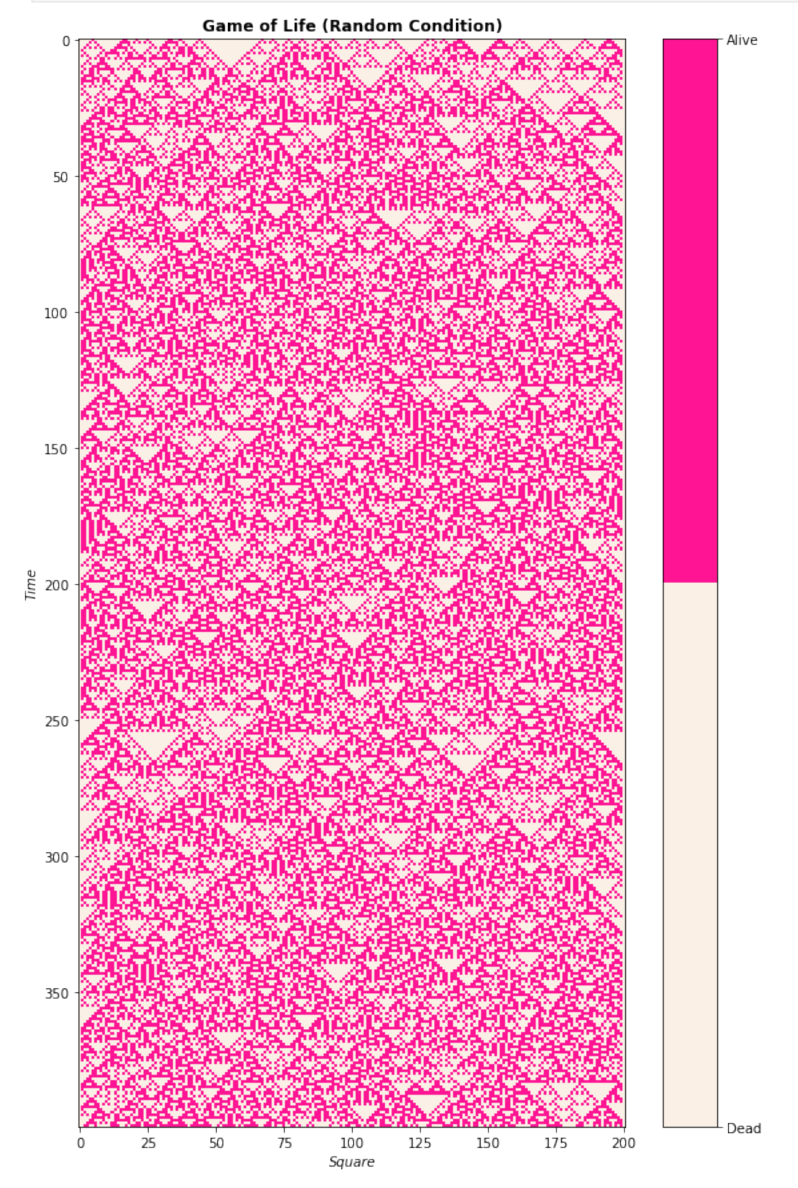
In [ ]:
```python
# Import the necessary libraries
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap

# Function that receives a board size and vector of cells with initial conditions and returns a graph simulating the game of life
def gameOfLife(n_columns, n_rows, cells, condition):
    count = [np.sum(cells[0])] # List to store the live cell count in each row
    for i in range(1, n_rows):
        for j in range(1, n_columns - 1):
            neighbors = [cells[i-1, j-1], cells[i-1, j+1]]
            if any(v == 1 for v in neighbors) and any(v == 0 for v in neighbors):
                cells[i,j] = 1
        count.append(np.sum(cells[i])) # We count live cells and add them to the counting list

    # Generate the graphics of our game
    plt.figure(figsize=(10, 15))
    plt.imshow(cells, interpolation='nearest', cmap=ListedColormap(['linen', 'deeppink']))  # Custom colormap to represent "alive" and "dead"
    plt.xlabel('Square', fontstyle='italic')
    plt.ylabel('Time', fontstyle='italic')
    plt.title('Game of Life (%s Condition)' %condition, fontweight='bold')
    plt.colorbar(ticks=[0, 1]).set_ticklabels(['Dead', 'Alive'])

    # Generate the graph of the live cell count in each row
    plt.figure(figsize=(15, 8))
    plt.plot(range(0, n_rows), count, marker='o', linestyle='-', color='limegreen', label='Live Cells')
    plt.xlabel('Time', fontstyle='italic')
    plt.ylabel('Live Cells', fontstyle='italic')
    plt.title('Live Cell Count Over Time (%s Condition)' %condition, fontweight='bold')
    plt.grid(True, linestyle='--', alpha=0.6)
    plt.legend(loc='best')

    plt.show()
```

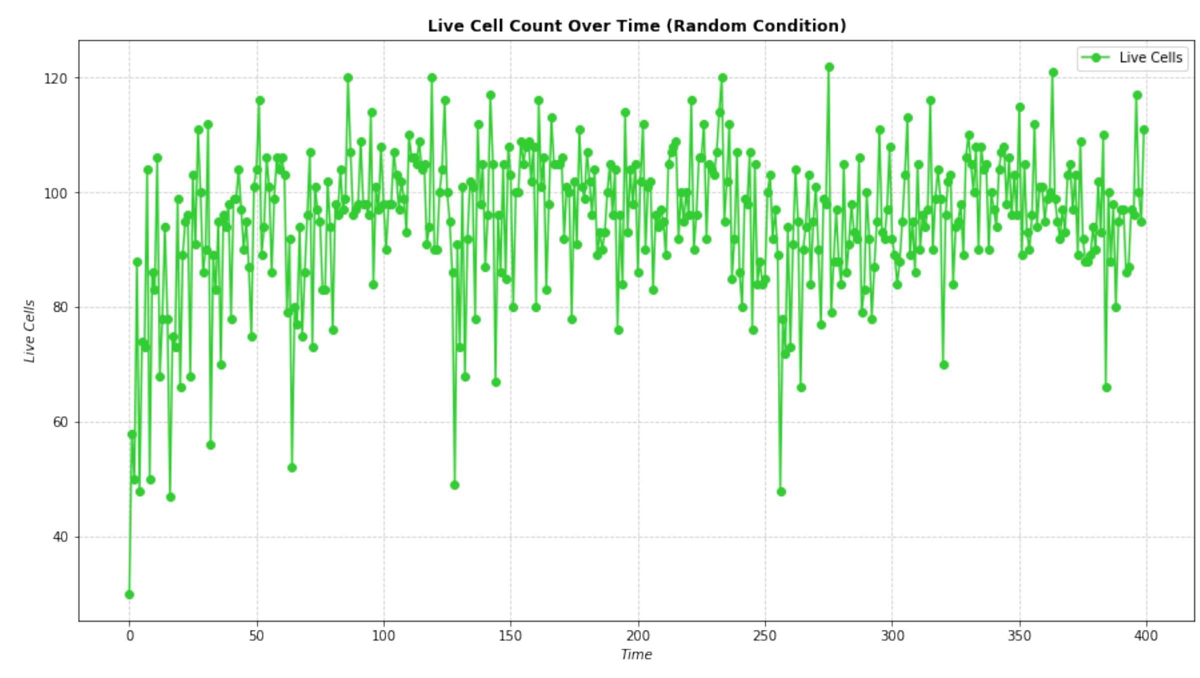
In [ ]:
```python
# Define the size of our board
n_columns, n_rows = 201, 400

# Matrix with 0's representing all dead cells
cells = np.zeros((n_rows, n_columns))

# Define the first live box (deterministic condition)
cells[0, int((n_columns-1)/2)] = 1

# Play, according to the rules, for each line.
gameOfLife(n_columns, n_rows, cells, "Deterministic")
```





In [ ]:
```python
# Define the size of our board
n_columns, n_rows = 201, 400

# Matrix with 0's representing all dead cells
celulas = np.zeros((n_rows, n_columns))

# Define the first 30 live cells (random condition)
celulas[0,np.random.choice(n_columns, 30, replace=False)] = 1

# Play, according to the rules, for each line
gameOfLife(n_columns, n_rows, celulas, "Random")
```





Sierpinski Triangle

1. How many complete triangles are there in iteration k?

$$N_k = 3^k$$

with k = {0,1,2,3,...,n}, where k=0 represents our initial equilateral triangle.

Let us note that when k tends to ∞, we have:

$$lim_{k\to\infty} N_k = \infty$$

2. What is the area of each complete triangle at iteration k?
Let a_0 be the first equilateral triangle at k=0, the area of each filled triangle is given by:

$$A_k = (\frac{1}{4})^k \cdot a_0$$

since, if we look at the k=1 case, we have 3 complete triangles and 1 empty one; so each one represents 1/4 of the total area.

Let us note that when k tends to ∞, we have:

$$lim_{k\to\infty} A_k = 0$$

3. What is the total area of the object at iteration k?
To find the expression, simply take the product of each triangle in k iterations by the number of total triangles.

$$A_{Total}(k) = N \cdot A_k = (\frac{3}{4})^k \cdot a_0$$

Let us note that when k tends to ∞, we have:

$$lim_{k\to\infty} A_{Total}(k) = 0$$

Koch's Flake

1. How many sides does the object have at iteration k?

$$N_k = 3 \cdot 4^k$$

with k = {0,1,2,3,...,n}, where k=0 represents our initial equilateral triangle.

Let us note that when k tends to ∞, we have:

$$lim_{k\to\infty} N_k = \infty$$

2. What is the perimeter of the object at iteration k?
Let "ℓ" be the original length of each side of the equilateral triangle at k=0, so the length of each side at iteration k is given by:

$$L_k = \frac{\ell}{3^k}$$

Therefore, the perimeter of the Koch flake at iteration k is:

$$P_k = N_k \cdot L_k = 3 \cdot \ell \cdot (\frac{4}{3})^k = P_0 \cdot (\frac{4}{3})^k$$

Let us note that when k tends to ∞, we have:

$$lim_{k\to\infty} P_k = \infty$$

3. What is the area of the object at iteration k?
Let a_0 be the first equilateral triangle at k=0, note that we can divide it into 9 identical equilateral triangles by the relationship of similarity between their areas. Therefore, the area of each triangle in iteration k is given by:

$$a_k = \frac{1}{9^k} \cdot a_0$$

So, to calculate the area added by all the triangles in a certain iteration k, simply calculate the product of the area of each triangle by the total number of triangles, or its equivalent, the total number of sides of the triangle:

$$A_k = N_k \cdot a_n = 3 \cdot 4^{k-1} \cdot \frac{1}{9^k} \cdot a_0 = \frac{3}{4} \cdot (\frac{4}{9})^k \cdot a_0$$

Finally, to calculate the total area of the Koch Flake we can approximate it by adding all the triangles in each iteration, such that:

$$A_{Total}(k) = a_0 + \sum_{i=1}^{k} A_i \; = a_0 + \sum_{i=1}^{k} \frac{3}{4} \cdot a_0 \cdot (\frac{4}{9})^i$$

Where

$$\sum_{i=1}^{k} \frac{3}{4} \cdot a_0 \cdot (\frac{4}{9})^i$$

is the geometric series, such that:

$$\sum_{i=0}^{n} ar^i = a \cdot \frac{1-r^{n+1}}{1-r}$$

So, we solve:

$$A_{Total}(k) = a_0 + \frac{3}{4} \cdot a_0 \cdot (\frac{1-\frac{4}{9}^{k+1}}{1-\frac{4}{9}} - 1) = a_0 \cdot (1 - \frac{3}{4} + \frac{3}{4} \cdot (\frac{1-\frac{4}{9}^{k+1}}{\frac{5}{9}}))$$

Note that our expression goes up to k, so we look at the case when k tends to infinity, such that:

$$lim_{k\to\infty} A_{Total}(k) = a_0 \cdot (1 - \frac{3}{4} + \frac{3}{4} \cdot (\frac{1-0}{\frac{5}{9}})) = a_0 \cdot (\frac{1}{4} + \frac{3}{4} \cdot \frac{9}{5})$$

$$lim_{k\to\infty} A_{Total}(k) = \frac{8}{5} \cdot a_0$$