Game Of Life

Alan Kantserov

November 20, 2015

The Game of Life, also known simply as Life, is a cellular automaton devised by the British mathematician John Horton Conway in 1970. Basically, this game consists of classes. However, each class has functions inside itself, which make game alive. Rules for the alive cells:

- Cells with less than 2 alive neighbours die
- Cells with more than 3 alive neighbours die
- Cells with 2 or 3 alive neighbours stay alive

Rules for the dead cells:

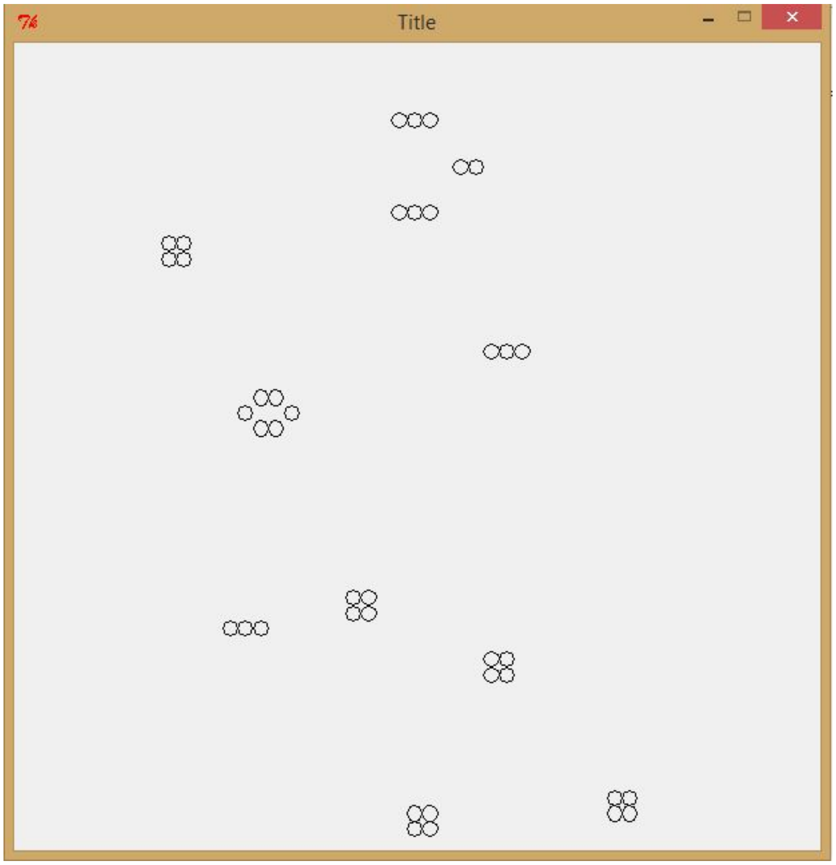
- Cells 3 alive neighbours reborn and become alive

Figure 1: Not fun game of life

PYTHON CODE

```
import random #random number generator library
from graphics import * #graphics library

#this function creates an NxN array filled with zeros
def empty(N): #class empty with value N
    a=[] #empty arra
    for i in range(N): #loop N times
        b=[] #empty array
        for j in range(N): #loop N times
            b=b+[0] #add zeros to an array
        a=a+[b]
    return a

#this function fills the array a with a portion p of live cells
def fill(a,p):
    N=len(a)
    for i in range(N):
        for j in range(N):
            if random.uniform(0,1)<p:
                a[i][j]=1

def update(A,B): #this function will update our game.
    N=len(A)
    for i in range(N):
        for j in range(N):
            neigh=A[(i-1)%N][(j-1)%N]+A[(i-1)%N][j]+A[(i-1)%N][(j+1)%N]+
            A[i][(j-1)%N]+A[i][(j+1)%N]+A[(i+1)%N][(j-1)%N]+A[(i+1)%N][j]+
            A[(i+1)%N][(j+1)%N]
            if A[i][j]==0:
                if neigh==3:
                    B[i][j]=1
                else:
                    B[i][j]=0
            else:
                if neigh==2 or neigh==3:
                    B[i][j]=1
                else:
                    B[i][j]=0



def gen2Dgraphic(N): #this generates our layout
    a=[]
    for i in range(N):
        b=[]
        for j in range(N):
            b=b+[Circle(Point(i,j),.49)]
        a=a+[b]
    return a

def slider(a): #
    a[1][0]=1
    a[0][1]=1
    a[0][2]=1
    a[1][2]=1
    a[2][2]=1

def push(B,A): # this will push updated version
    N=len(A)
    for i in range(N):
        for j in range(N):
            A[i][j]=B[i][j]


def drawArray(A,a,window):
#A is the array of 0,1 values representing the state of the game
#a is an array of Circle objects
#window is the GraphWin in which we will draw the circles
    N=len(A)
    for i in range(N):
        for j in range(N):
            if A[i][j]==1:
                a[i][j].undraw()
                a[i][j].draw(window)
            if A[i][j]==0:
                a[i][j].undraw()

N=50
win = GraphWin("Title",600,600)
win.setCoords(-1,-1,N+1,N+1)
grid=empty(N)
grid2=empty(N)
circles=gen2Dgraphic(N)
fill(grid,0.1)

while True: #we want to run this program all the time
    drawArray(grid,circles,win)
    update(grid,grid2)
    push(grid2,grid)
```

OUTPUT:



Thats is what it is going to be like. However, you might have something completely different and it updates with the time