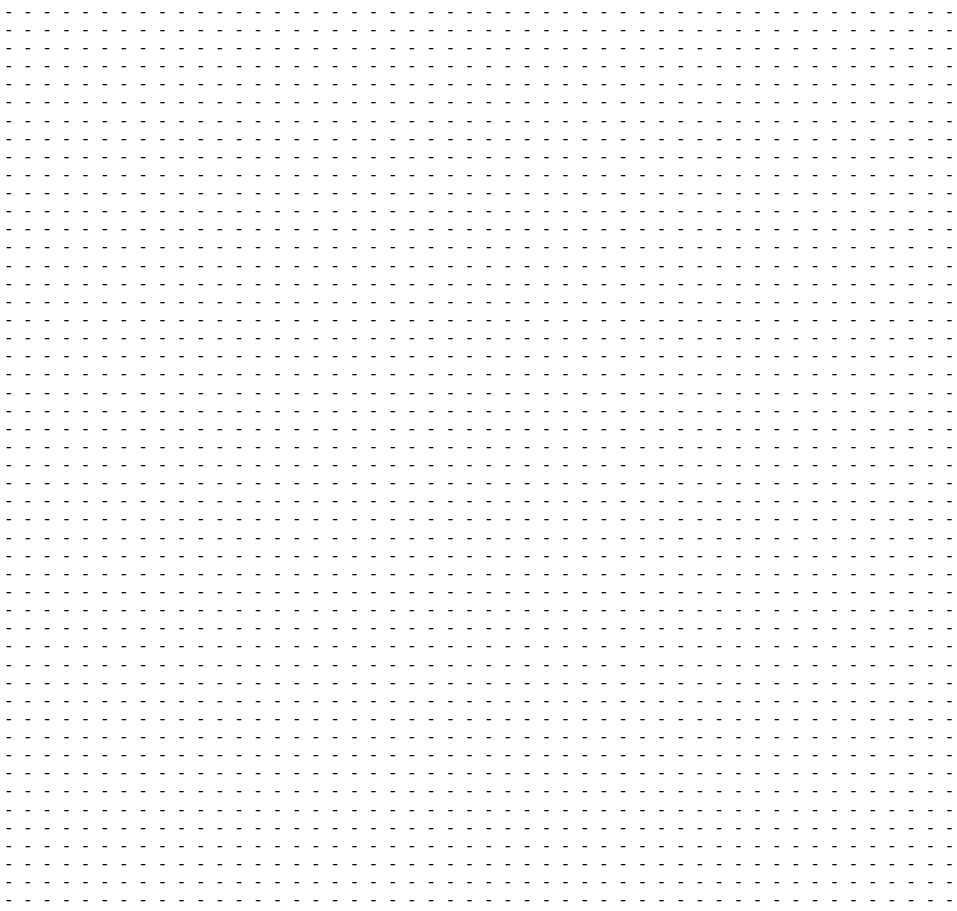
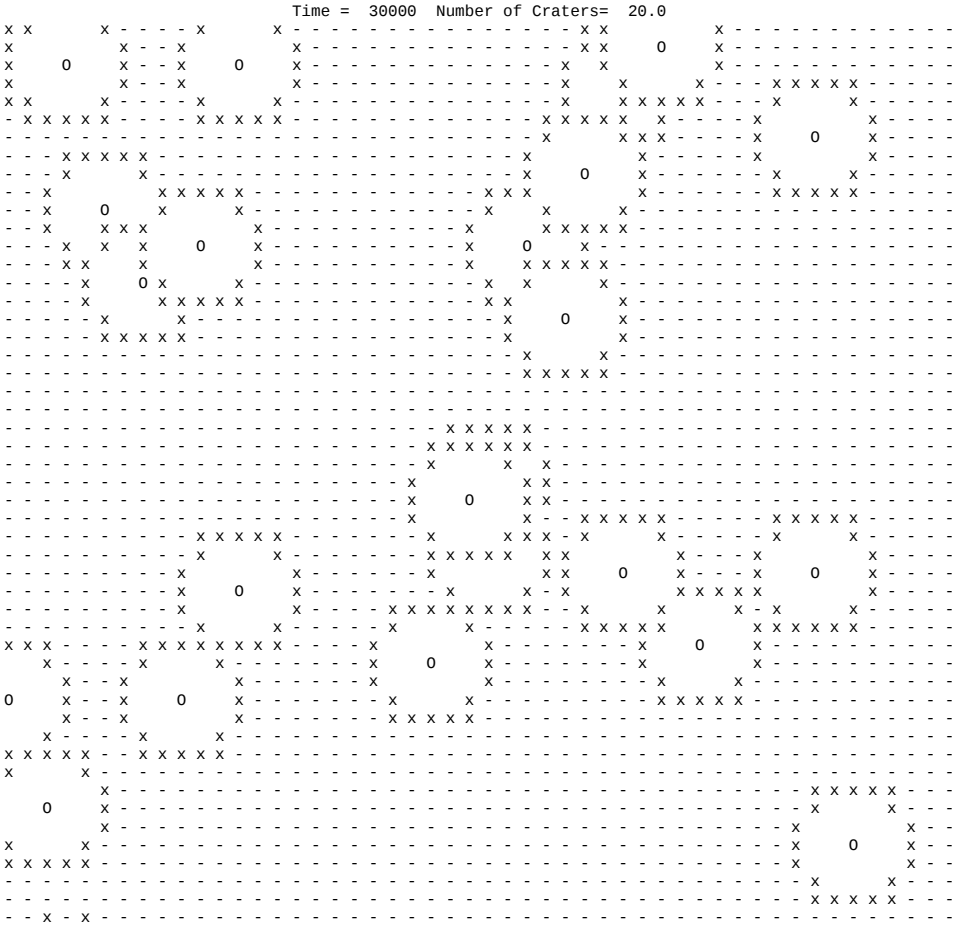
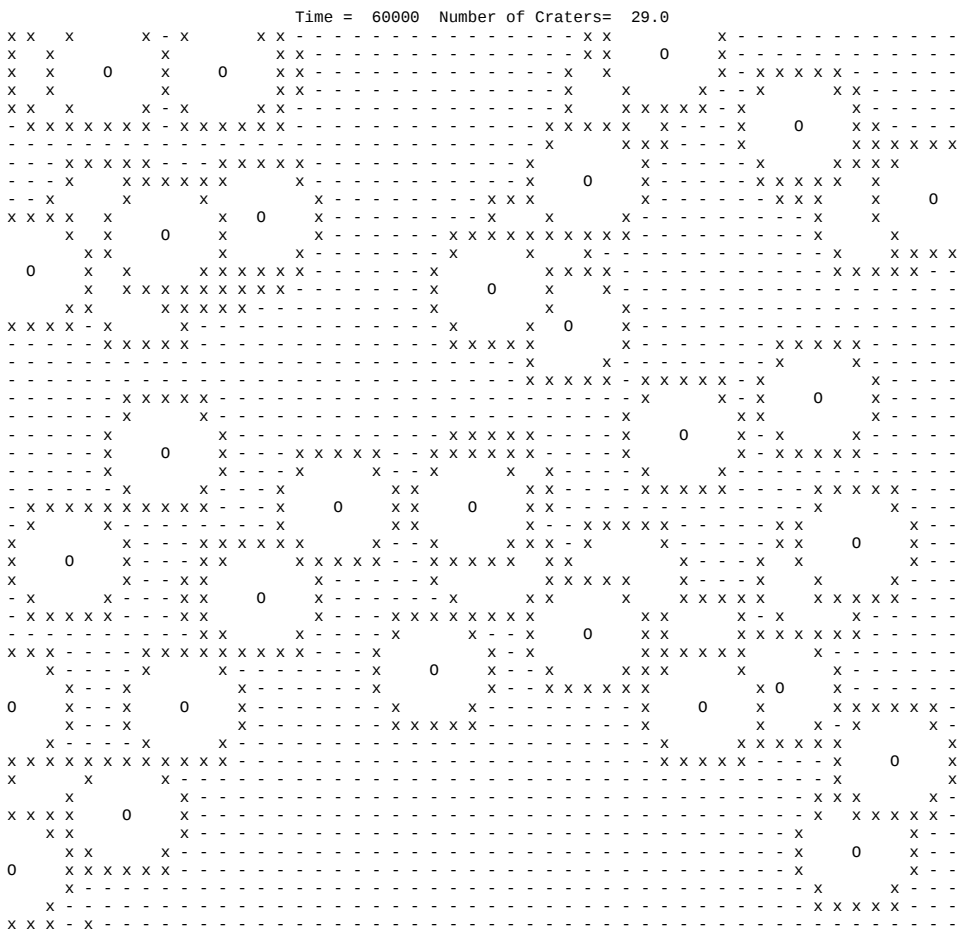
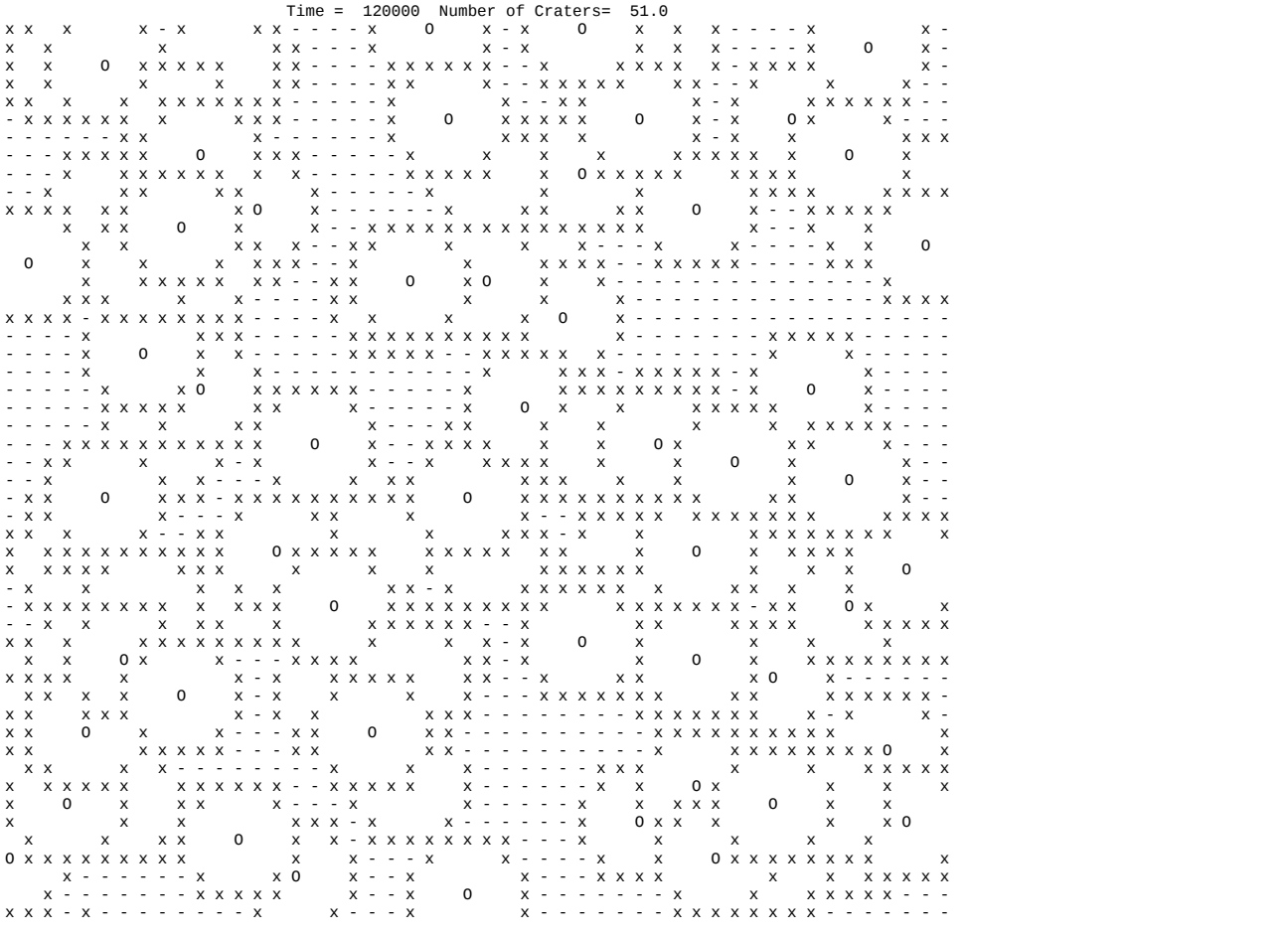
Time = 0 Number of Craters= 0

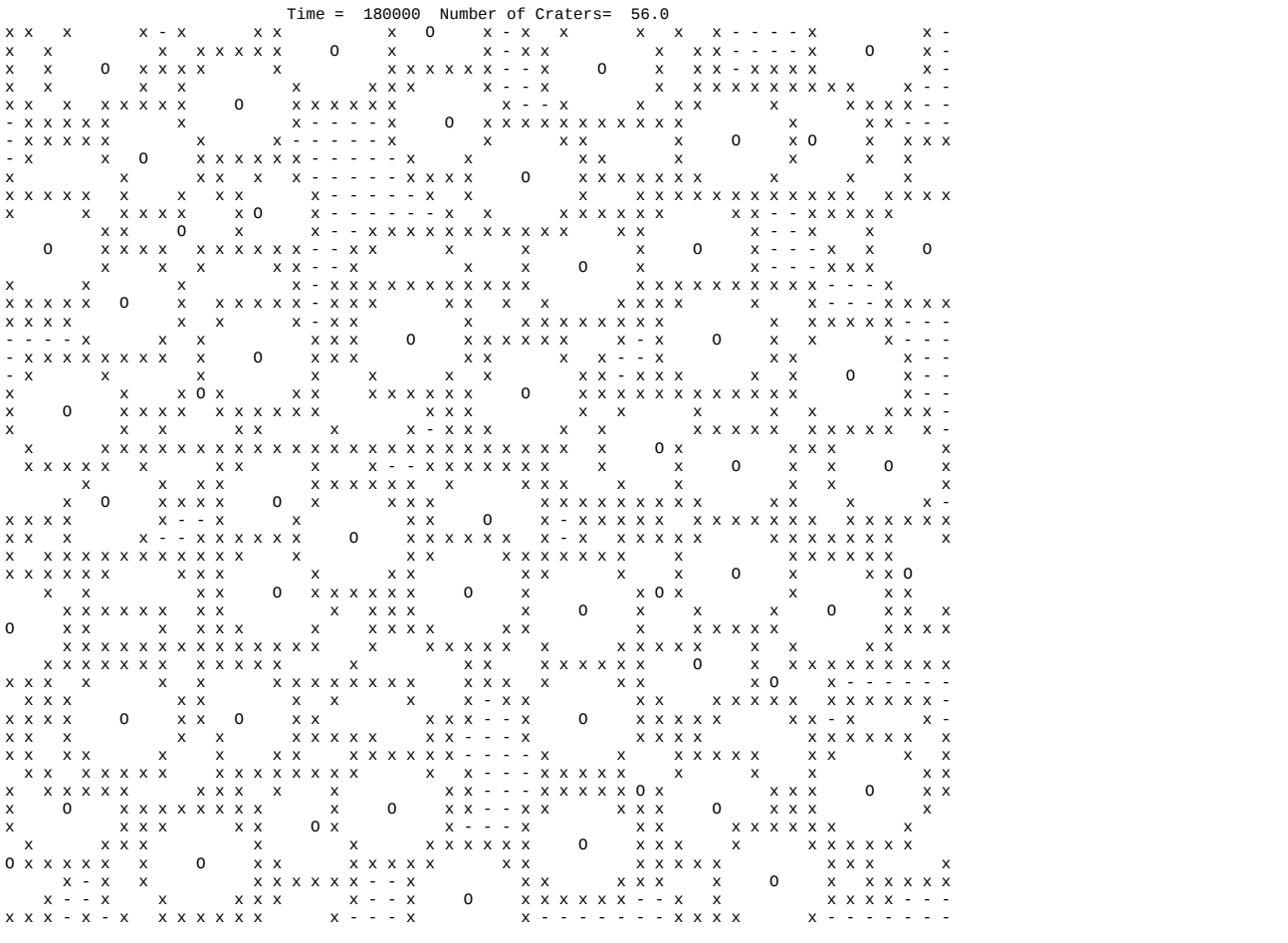


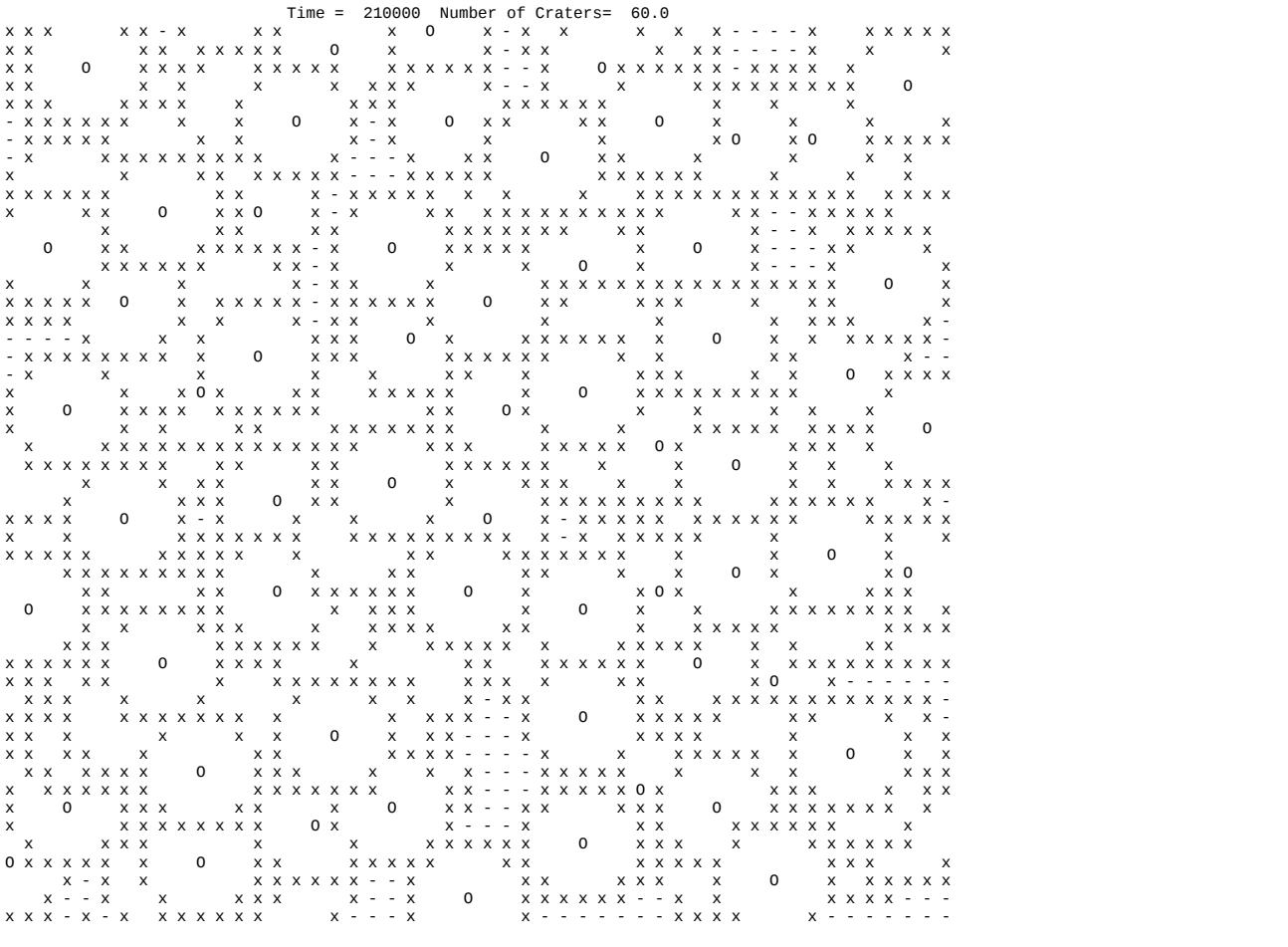
Time = 30000 Number of Craters= 20.0

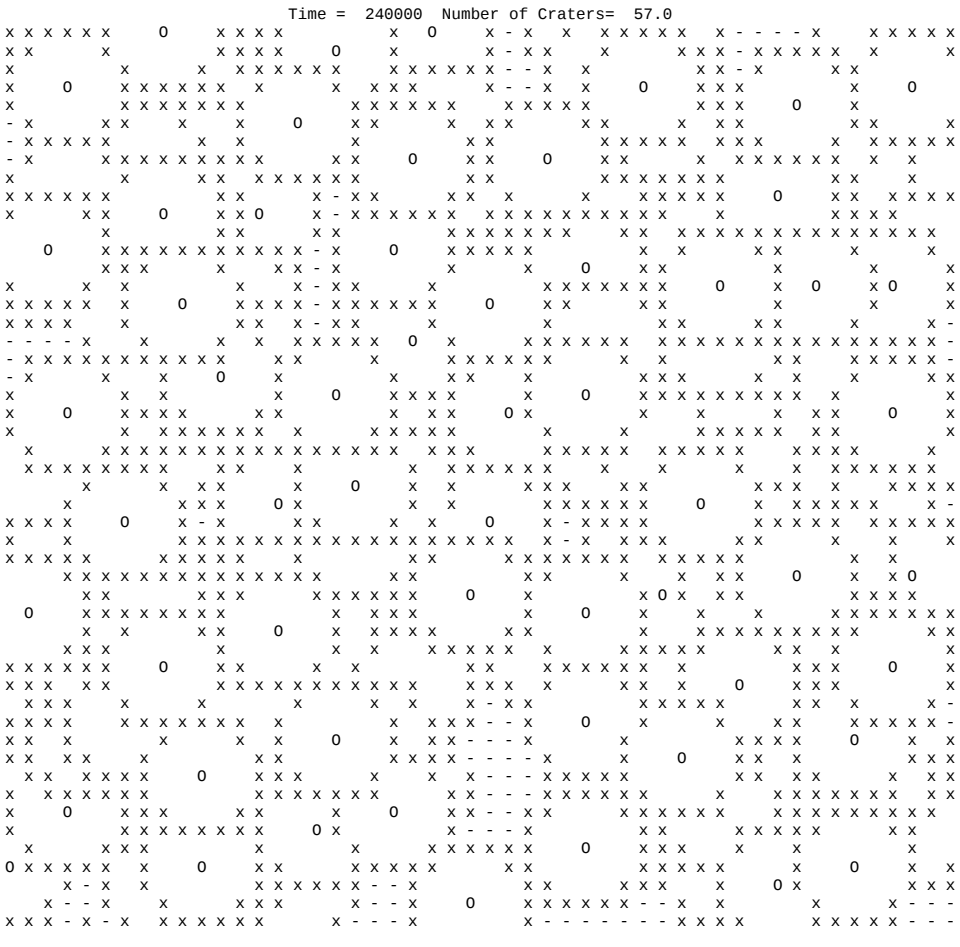


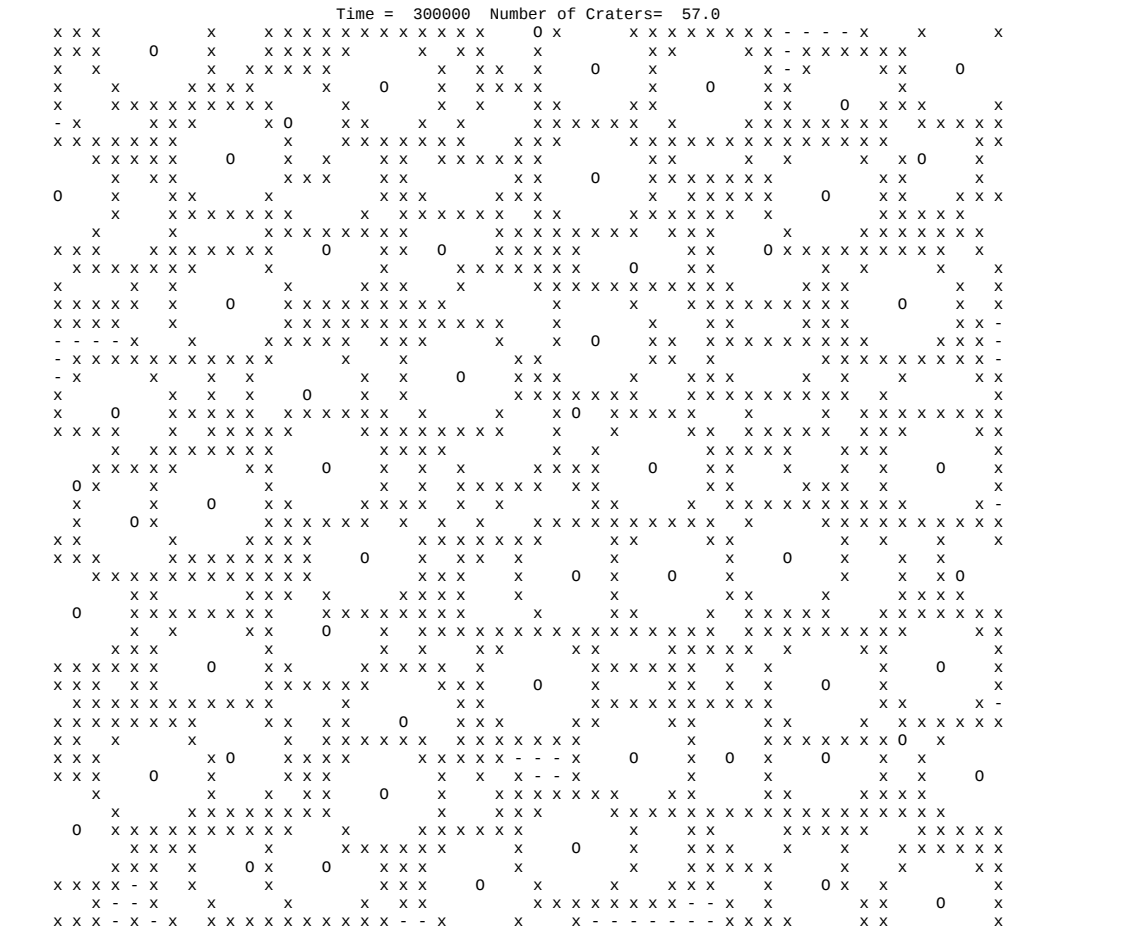

Time = 60000 Number of Craters= 29.0
Time = 90000 Number of Craters= 49.0


Time = 120000 Number of Craters= 51.0
Time = 150000 Number of Craters= 52.0


Time = 180000 Number of Craters= 56.0


Time = 210000 Number of Craters= 60.0


Time = 240000 Number of Craters= 57.0
Time = 270000 Number of Craters= 56.0



results

Number of Craters: 58.0 Years to Saturation: 319965

```
Time    |  Number of Craters
0       |     0
10000   |  **********   10.0
20000   |  ***************   15.0
30000   |  ********************   20.0
40000   |  *************************   25.0
50000   |  ***************************   27.0
60000   |  ******************************   29.0
70000   |  ********************************   33.0
80000   |  ****************************************   43.0
90000   |  ************************************************   49.0
100000  |  ***************************************************   51.0
110000  |  ******************************************************   54.0
120000  |  ***************************************************   51.0
130000  |  ***************************************************   51.0
140000  |  ************************************************   49.0
150000  |  ***************************************************   52.0
160000  |  ********************************************************   56.0
170000  |  *********************************************************   57.0
180000  |  ********************************************************   56.0
190000  |  *******************************************************   55.0
200000  |  **********************************************************   59.0
210000  |  ***********************************************************   60.0
220000  |  *********************************************************   58.0
230000  |  ********************************************************   56.0
240000  |  *********************************************************   57.0
250000  |  ***********************************************************   59.0
260000  |  *********************************************************   57.0
270000  |  ********************************************************   56.0
280000  |  *********************************************************   57.0
290000  |  ***********************************************************   60.0
300000  |  *********************************************************   57.0
310000  |  *********************************************************   58.0
```

Code:

```
#!/usr/bin/python

import random

# area is 500km x 500 km but is broken into a 10km grid
# therefore we can use a 50 x 50 area and have craters be 5 units large rather than 50
SIZE = 50
craterSize = SIZE/10
craters = [0]
time = 0
crater = 0.0
EMPTY = '-'
CENTER = 'O'
CRATER = ' '
DEBRIS = 'x'

# initialize empty, uncratered grid
grid = [[EMPTY for x in xrange(SIZE)] for x in xrange(SIZE)]

# function that goes through the grid and counts the number of crater centers
def numCraters(grid):
    crater = 0
    for i in range(SIZE-1):
        for j in range(SIZE-1):
            if(grid[i][j] == CENTER):
```

```
                crater += 1.0
    return crater

# Randomly pick a spot on the grid and make a crater centering at this location.
def makeImpact(grid):
    imp_i = random.randrange(0, SIZE-1)
    imp_j = random.randrange(0, SIZE-1)

    for i in range((craterSize/2)+2):
        for j in range(((craterSize/2)+2)-i/2):
            # must ensure that you won't write outside of the grid, if the center is near
an edge of the grid

            # create circle of debris first, as this is the largest circle
            if(imp_i+i >= SIZE):
                if(imp_j+j >= SIZE):
                    grid[imp_i][imp_j-j] = DEBRIS
                    grid[imp_i-i][imp_j-j] = DEBRIS
                    grid[imp_i-i][imp_j] = DEBRIS
                if(imp_j-j <= 0):
                    grid[imp_i][imp_j+j] = DEBRIS
                    grid[imp_i-i][imp_j] = DEBRIS
                    grid[imp_i-i][imp_j+j] = DEBRIS
                if(imp_j+j < SIZE and imp_j-j > 0):
                    if(imp_j+j >= SIZE): print SIZE
                    grid[imp_i][imp_j-j] = DEBRIS
                    grid[imp_i][imp_j+j] = DEBRIS
                    grid[imp_i-i][imp_j-j] = DEBRIS
                    grid[imp_i-i][imp_j+j] = DEBRIS

            if(imp_i - i < 0):
                if(imp_j+j >= SIZE):
                    grid[imp_i][imp_j-j] = DEBRIS
                    grid[imp_i+i][imp_j-j] = DEBRIS
                    grid[imp_i+i][imp_j] = DEBRIS
                if(imp_j-j < 0):
                    grid[imp_i][imp_j+j] = DEBRIS
                    grid[imp_i+i][imp_j] = DEBRIS
                    grid[imp_i+i][imp_j+j] = DEBRIS
                if(imp_j+j < SIZE and imp_j-j>0):
                    if(imp_j+j >= SIZE): print SIZE
                    grid[imp_i][imp_j-j] = DEBRIS
                    grid[imp_i][imp_j+j] = DEBRIS
                    grid[imp_i+i][imp_j-j] = DEBRIS
                    grid[imp_i+i][imp_j+j] = DEBRIS

            if(imp_j+j >= SIZE):
                if(imp_i+i >= SIZE):
                    grid[imp_i-i][imp_j] = DEBRIS
                    grid[imp_i-i][imp_j-j] = DEBRIS
                    grid[imp_i][imp_j-j] = DEBRIS

                if(imp_i-i <= 0):
                    grid[imp_i-i][imp_j] = DEBRIS
                    grid[imp_i-i][imp_j-j] = DEBRIS
                    grid[imp_i][imp_j-j] = DEBRIS

                if((imp_i+i < SIZE) and imp_i-i >0):
                    grid[imp_i-i][imp_j] = DEBRIS
                    grid[imp_i+i][imp_j] = DEBRIS
                    grid[imp_i-i][imp_j-j] = DEBRIS
                    grid[imp_i+i][imp_j-j] = DEBRIS
```

```
        if(imp_j-j <= 0):
            if(imp_i+i >= SIZE):
                grid[imp_i-i][imp_j] = DEBRIS
                grid[imp_i-i][imp_j+j] = DEBRIS
                grid[imp_i][imp_j+j] = DEBRIS
            if(imp_i-i <= 0):
                grid[imp_i-i][imp_j] = DEBRIS
                grid[imp_i-i][imp_j+j] = DEBRIS
                grid[imp_i][imp_j+j] = DEBRIS
            if(imp_i+i < SIZE and imp_i-i>0):
                grid[imp_i-i][imp_j] = DEBRIS
                grid[imp_i+i][imp_j] = DEBRIS
                grid[imp_i-i][imp_j+j] = DEBRIS
                grid[imp_i+i][imp_j+j] = DEBRIS

        if(imp_j+j < SIZE and imp_j-j >=0 and imp_i +i < SIZE and imp_i - i >=0):
            grid[imp_i-i][imp_j-j] = DEBRIS
            grid[imp_i+i][imp_j-j] = DEBRIS
            grid[imp_i-i][imp_j +j] = DEBRIS
            grid[imp_i+i][imp_j +j] = DEBRIS

    # make a smaller circle within the debris circle for the actual crater
    for i in range((craterSize/2)+1):
        for j in range(((craterSize/2)+1)-i/2):
            if(imp_i+i >= SIZE):
                if(imp_j+j >= SIZE):
                    grid[imp_i][imp_j-j] = CRATER
                    grid[imp_i-i][imp_j-j] = CRATER
                    grid[imp_i-i][imp_j] = CRATER
                if(imp_j-j <= 0):
                    grid[imp_i][imp_j+j] = CRATER
                    grid[imp_i-i][imp_j] = CRATER
                    grid[imp_i-i][imp_j+j] = CRATER
                if(imp_j+j < SIZE and imp_j-j>0):
                    grid[imp_i][imp_j-j] = CRATER
                    grid[imp_i][imp_j+j] = CRATER
                    grid[imp_i-i][imp_j-j] = CRATER
                    grid[imp_i-i][imp_j+j] = CRATER

            if(imp_i - i < 0):
                if(imp_j+j >= SIZE):
                    grid[imp_i][imp_j-j] = CRATER
                    grid[imp_i+i][imp_j-j] = CRATER
                    grid[imp_i+i][imp_j] = CRATER
                if(imp_j-j < 0):
                    grid[imp_i][imp_j+j] = CRATER
                    grid[imp_i+i][imp_j] = CRATER
                    grid[imp_i+i][imp_j+j] = CRATER
                if(imp_j+j < SIZE and imp_j-j>0):
                    grid[imp_i][imp_j-j] = CRATER
                    grid[imp_i][imp_j+j] = CRATER
                    grid[imp_i+i][imp_j-j] = CRATER
                    grid[imp_i+i][imp_j+j] = CRATER

            if(imp_j+j >= SIZE):
                if(imp_i+i >= SIZE):
                    grid[imp_i-i][imp_j] = CRATER
                    grid[imp_i-i][imp_j-j] = CRATER
                    grid[imp_i][imp_j-j] = CRATER

                if(imp_i-i <= 0):
                    grid[imp_i-i][imp_j] = CRATER
                    grid[imp_i-i][imp_j-j] = CRATER
```

```
                    grid[imp_i][imp_j-j] = CRATER

                if((imp_i+i < SIZE) and imp_i-i >0):
                    grid[imp_i-i][imp_j] = CRATER
                    grid[imp_i+i][imp_j] = CRATER
                    grid[imp_i-i][imp_j-j] = CRATER
                    grid[imp_i+i][imp_j-j] = CRATER

            if(imp_j-j <= 0):
                if(imp_i+i >= SIZE):
                    grid[imp_i-i][imp_j] = CRATER
                    grid[imp_i-i][imp_j+j] = CRATER
                    grid[imp_i][imp_j+j] = CRATER

                if(imp_i-i <= 0):
                    grid[imp_i-i][imp_j] = CRATER
                    grid[imp_i-i][imp_j+j] = CRATER
                    grid[imp_i][imp_j+j] = CRATER

                if(imp_i+i < SIZE and imp_i-i>0):
                    grid[imp_i-i][imp_j] = CRATER
                    grid[imp_i+i][imp_j] = CRATER
                    grid[imp_i-i][imp_j+j] = CRATER
                    grid[imp_i+i][imp_j+j] = CRATER

            if(imp_j+j < SIZE and imp_j-j >=0 and imp_i +i < SIZE and imp_i - i >=0):
                grid[imp_i-i][imp_j-j] = CRATER
                grid[imp_i+i][imp_j-j] = CRATER
                grid[imp_i-i][imp_j +j] = CRATER
                grid[imp_i+i][imp_j +j] = CRATER

        grid[imp_i][imp_j] = CENTER   #finally place the center at the center of the
impact
    return grid


#returns true if there is saturation (ie less that 5% change when time doubles)
def checkSat(craters, time):
    if(time==0):
        curAvg = 0.0
        oldAvg = 0.0
    else:
        if((time/2)==0):
            curAvg = craters[time] / time
            oldAvg = 0.0
        else:
            curAvg = craters[time] / time
            oldAvg = craters[time/2] / (time/2)
    # if the number of craters changes by less than 5% when time is doubled
    # we have reached saturation
    if(abs(craters[time] - craters[time/2]) < craters[time/2]*0.05):
        # if there are less than 5 craters we definitely have not reached saturation
        # but test could still pass
        if(craters[time] < 5):
            return False
        else:
            return True
    else:
        return False

# while the grid is not saturated with craters continue randomly generating impacts and
increasing time
while not checkSat(craters, time):
```

```
    # cratering rate is 1 impact every 1000 years, therefore every year chance of impact
is 1/1000
    # generate a number between 1 and 1000
    impact = random.randrange(1,1000)

    # if the random number is some specific number make an impact
    if(impact == 129):
        grid = makeImpact(grid)
    craters.append(numCraters(grid))
    if(time % 30000 == 0):
        print "\nTime = ", time, " Number of Craters= ", craters[time]
        for x in grid:
            print(' '.join(x))
    time += 1

print "\n------------------------------------------------------------
results-------------------------------------------------------------"
for x in grid:
    print(' '.join(x))
print "Number of Craters: ", numCraters(grid)
print "Years to Saturation: ", time
print "Time  |  Number of Craters"
for i in range(len(craters)):
   if(i % 10000 == 0):
        toprint = ""
        for x in range(int(craters[i])):
            toprint += "*"
        print i, " | ", toprint, " ", craters[i]
```

We are wanting to simulate a 500km x 500km area but we have divided it into 10km cells meaning we can do a simulation for a 50x50 area instead, which will run more quickly. Each year there is a 1/1000 chance an impact will occur. As long as we have not reached saturation, meaning the number of craters has changed by less than 5% when the time doubles, we continue allowing impacts to form.