In [2]:

```
#================MONTE CARLO=========================
import numpy as np

def func(x):
    return 1/x

# MC integration between a and b of function
list_int = []
def mc(func, a, b, n = 1000):
    for j in range(n):                        #loops to put all integrals value in a
        values = np.random.uniform(a, b, n)
        y = [func(i) for i in values]
        y_mean = np.sum(y)/n                   #sum of y values in list divided by n
        integral = (b-a) * y_mean              #integral equation
        list_int.append(integral)            #appending to list

    return integral

print(f"Monte Carlo: {mc(func, 1, 2.71, 1000): .4f}") #a = 1, b = 2.71 ~ e
```

```
Monte Carlo:  0.9865
```

In [5]:

```
#Write to file the integral values after each iteration
with open('monte_carlo.txt', 'w') as f:
    for item in list_int:
        f.write("%s\n" % item)

n_list = []                             #list of numbers from 1-1000 (for plot)
for j in range(1000):
    n_list.append(j)
with open('n.txt', 'w') as f:
    for item in n_list:
        f.write("%s\n" % item)
```

In [6]:

```
#=============Riemann Sums===============

a = 1
b = 2.71
n = 1000
h = (b - a) / (n)
x = np.linspace(a, b, n)  #Linspace Returns evenly spaced numbers over the spec
f = 1/(x)

Integral_Left = h * sum(f[:n-1])                      # [:1]takes nothing but ke
Integral_Right = h * sum(f[1:])                       # [1:]takes everything but
Integral_mid = h * sum(np.sin((x[:n-1] + x[1:])/2))   #general: #list[initial :

print(Integral_Left)
print(Integral_Right)
print(Integral_mid)
```

```
0.9964913988076767
0.9954123951176397
1.4471545459261332
```

In [ ]:

```

```

In [1]:

```
#PART 1: Out of 2 children, one who is a boy, what is probability that the othe
#Solution: 1/3 that the other will be a boy

import random
import time

win = 0
lose = 0
counter = 0

number = int(input("Number of simulations: "))

for i in range(0, number):
    combination = ["BB", "BG", "GB"]
    counter += 1
    #print("Simulation #: " +str(counter))
    random.shuffle(combination)          #Random Generator for Python in this c
    n = random.randrange(3)              #returns a randomly selected element i


    if combination[n] == "BG" or combination[n]=="GB":  #so if the other child
        lose += 1
        print("you have child2 = GIRL--lose")
    if combination[n] == "BB":
        win += 1
        print("you have child2 = BOY--win")

s = (win/number)*100
g = (lose/number)*100

print("probability of boy (win): %s" %(s))
print("probability of girl (lose): %g" %(g))

#With high enough trials, the answer approximates to 1/3 for boy, and 2/3 for g
```

```
Number of simulations: 20
you have child2 = GIRL--lose
you have child2 = GIRL--lose
you have child2 = GIRL--lose
you have child2 = GIRL--lose
you have child2 = GIRL--lose
you have child2 = BOY--win
you have child2 = GIRL--lose
you have child2 = BOY--win
you have child2 = GIRL--lose
you have child2 = GIRL--lose
you have child2 = BOY--win
you have child2 = GIRL--lose
you have child2 = GIRL--lose
you have child2 = BOY--win
you have child2 = GIRL--lose
you have child2 = BOY--win
you have child2 = GIRL--lose
you have child2 = GIRL--lose
you have child2 = GIRL--lose
you have child2 = BOY--win
probability of boy (win): 30.0
probability of girl (lose): 70
```

In [2]:

```
#PART 2:
#"I have two children. One is a boy born on a Tuesday. What is the probability
#Solution: Probability of a two boys = 13/27, and probability of boy-girl combo

import itertools      #this module is for creating permutations and combinatio
from itertools import permutations

#This part will give you list of special boy-- tuesday-boy
boy = ["boy"]
tues = [2]
list_0 = [(x,y) for x in boy for y in tues]

#This will output 14 options: ('boy', 1), ('girl', 1) etc.--boy born for each d
gender = ["boy", "girl"]
day = [1, 2, 3, 4, 5, 6, 7]
list_1 = [(x,y) for x in gender for y in day]  #combines the two lists

#This will make a permutation of the list_1 with length of 2
list1_permutations = itertools.permutations(list_1, len(list_0))

#del list ###use to re-set zip in case it doesn't work (callback error)

#Create empty list that will have ALL cases that we want so (B2-G1),(B2-G2)...
together = []

#Loops through the list1_permutations and adds ('boy',2) to each (gender, day)
for each_permutation in list1_permutations:
    zipped_1 = list(zip(each_permutation, list_0))  #we need two zips to get (
    zipped_2 = list(zip(list_0, each_permutation))
    together.append(zipped_1)
    together.append(zipped_2)

del(together[2])  #must delete the second ('boy',2) since we have two from the

print(together)
print(len(together))
```

```
[[(('boy', 1), ('boy', 2))], [(('boy', 2), ('boy', 1))], [(('boy', 2), ('boy',
2))], [(('boy', 3), ('boy', 2))], [(('boy', 2), ('boy', 3))], [(('boy', 4),
('boy', 2))], [(('boy', 2), ('boy', 4))], [(('boy', 5), ('boy', 2))], [(('bo
y', 2), ('boy', 5))], [(('boy', 6), ('boy', 2))], [(('boy', 2), ('boy', 6))],
[(('boy', 7), ('boy', 2))], [(('boy', 2), ('boy', 7))], [(('girl', 1), ('boy',
2))], [(('boy', 2), ('girl', 1))], [(('girl', 2), ('boy', 2))], [(('boy', 2),
('girl', 2))], [(('girl', 3), ('boy', 2))], [(('boy', 2), ('girl', 3))], [(('g
irl', 4), ('boy', 2))], [(('boy', 2), ('girl', 4))], [(('girl', 5), ('boy',
2))], [(('boy', 2), ('girl', 5))], [(('girl', 6), ('boy', 2))], [(('boy', 2),
('girl', 6))], [(('girl', 7), ('boy', 2))], [(('boy', 2), ('girl', 7))]]
27
```

In [3]:

```
#Now for the simulation

import random   #must have the random library

win = 0
lose = 0
ctr = 0

number = int(input("How many simulations do you want?"))

for i in range(0, number):
    ctr += 1
    print("Simulation #: " +str(ctr))

    x = random.choice(together)     #Random Generator for Python in this case;
                                    #Randomly selects one of the objects in tog


    #1st [0] is the first '()'; 2nd [] accesses either ('boy',2) or ('girl',3);
    if x[0][0][0] == 'boy' and x[0][1][0] == 'boy':      #checking to see if bo
        win += 1                                         #if so we add to win
        print("You win--i.e. have two boys")
    else:
        lose += 1
        print("You lose--i.e. have boy-girl")

#Creating probabilities
s = (win/number)*100
g = (lose/number)*100

print("probability of two-boys (win): %s" %(s))
print("probability of girl (lose): %g" %(g))

#With higher trials, the percent will be aroun 48% for wins (i.e. 2 boys)= 13/2
```

```
How many simulations do you want?20
Simulation #: 1
You win--i.e. have two boys
Simulation #: 2
You win--i.e. have two boys
Simulation #: 3
You lose--i.e. have boy-girl
Simulation #: 4
You win--i.e. have two boys
Simulation #: 5
You win--i.e. have two boys
Simulation #: 6
You lose--i.e. have boy-girl
Simulation #: 7
You win--i.e. have two boys
Simulation #: 8
You win--i.e. have two boys
Simulation #: 9
You win--i.e. have two boys
Simulation #: 10
You win--i.e. have two boys
Simulation #: 11
You lose--i.e. have boy-girl
Simulation #: 12
You lose--i.e. have boy-girl
Simulation #: 13
You lose--i.e. have boy-girl
Simulation #: 14
You win--i.e. have two boys
Simulation #: 15
You win--i.e. have two boys
Simulation #: 16
You win--i.e. have two boys
Simulation #: 17
You lose--i.e. have boy-girl
Simulation #: 18
You lose--i.e. have boy-girl
Simulation #: 19
You lose--i.e. have boy-girl
Simulation #: 20
You lose--i.e. have boy-girl
probability of two-boys (win): 55.00000000000001
probability of girl (lose): 45
```

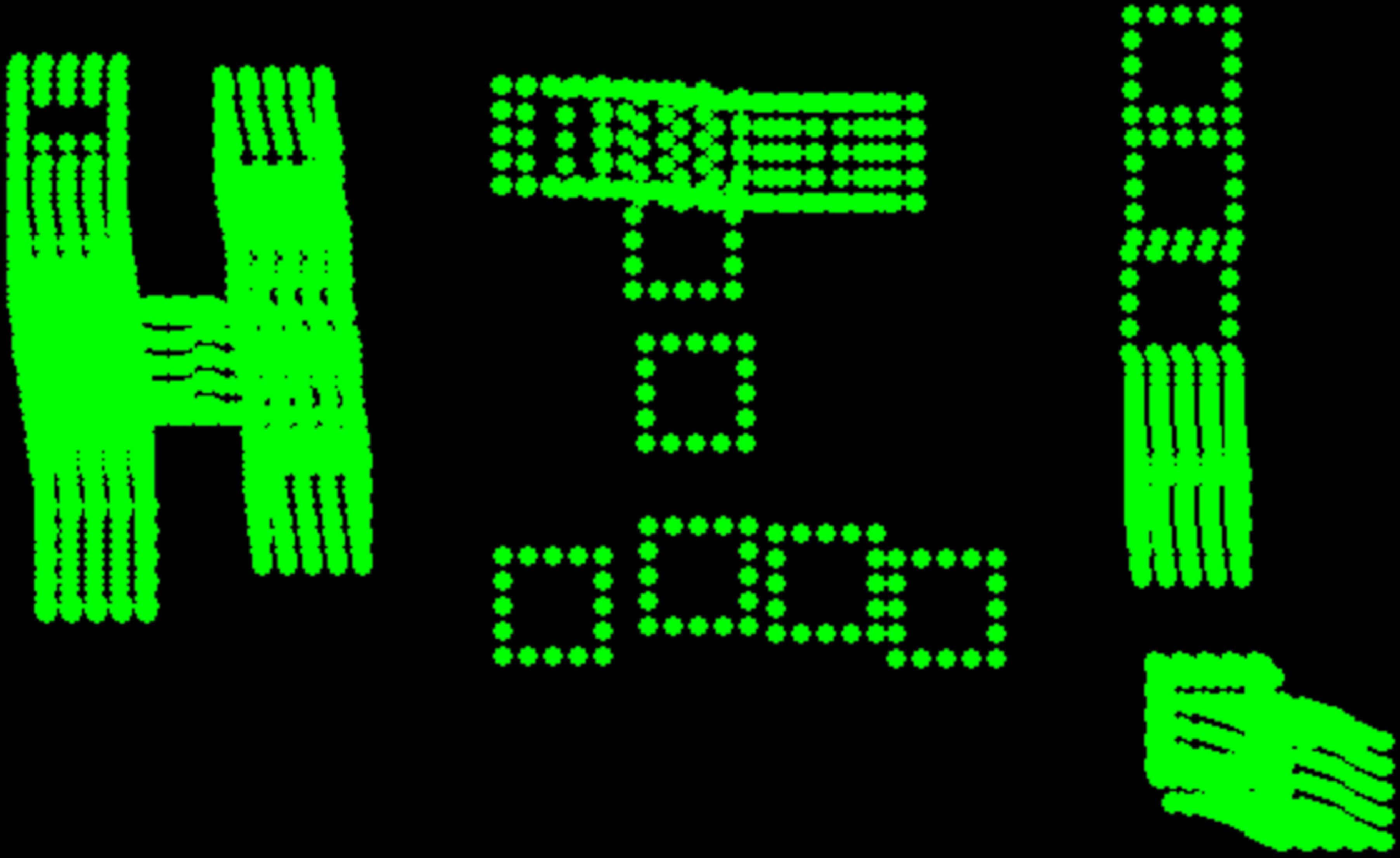
In [ ]:

In [3]:

```
# File: g (Python 3.2)

from pygame import *
init()
size = (800, 600)
(width, height) = (800, 600)
green = (0, 255, 0)
red = (255, 0, 0)
screen = display.set_mode(size)

def pattern(screen, x, y):
    for dx in range(-20, 21, 10):
        draw.circle(screen, green, (x + dx, y + 20), 4)
        draw.circle(screen, green, (x + dx, y - 20), 4)

    for dy in range(-20, 21, 10):
        draw.circle(screen, green, (x + 20, y + dy), 4)
        draw.circle(screen, green, (x - 20, y - dy), 4)



def drawScene(screen):
    if mouse.get_pressed()[0] == 1:
        (mx, my) = mouse.get_pos()
        pattern(screen, mx, my)
    display.flip()

running = True
myClock = time.Clock()
while running:
    for evnt in event.get():
        if evnt.type == QUIT:
            running = False
            continue
    drawScene(screen)
    myClock.tick(60)
quit()
```

```
pygame 2.1.2 (SDL 2.0.18, Python 3.9.12)
Hello from the pygame community. https://www.pygame.org/contribute.html
```

In [ ]:

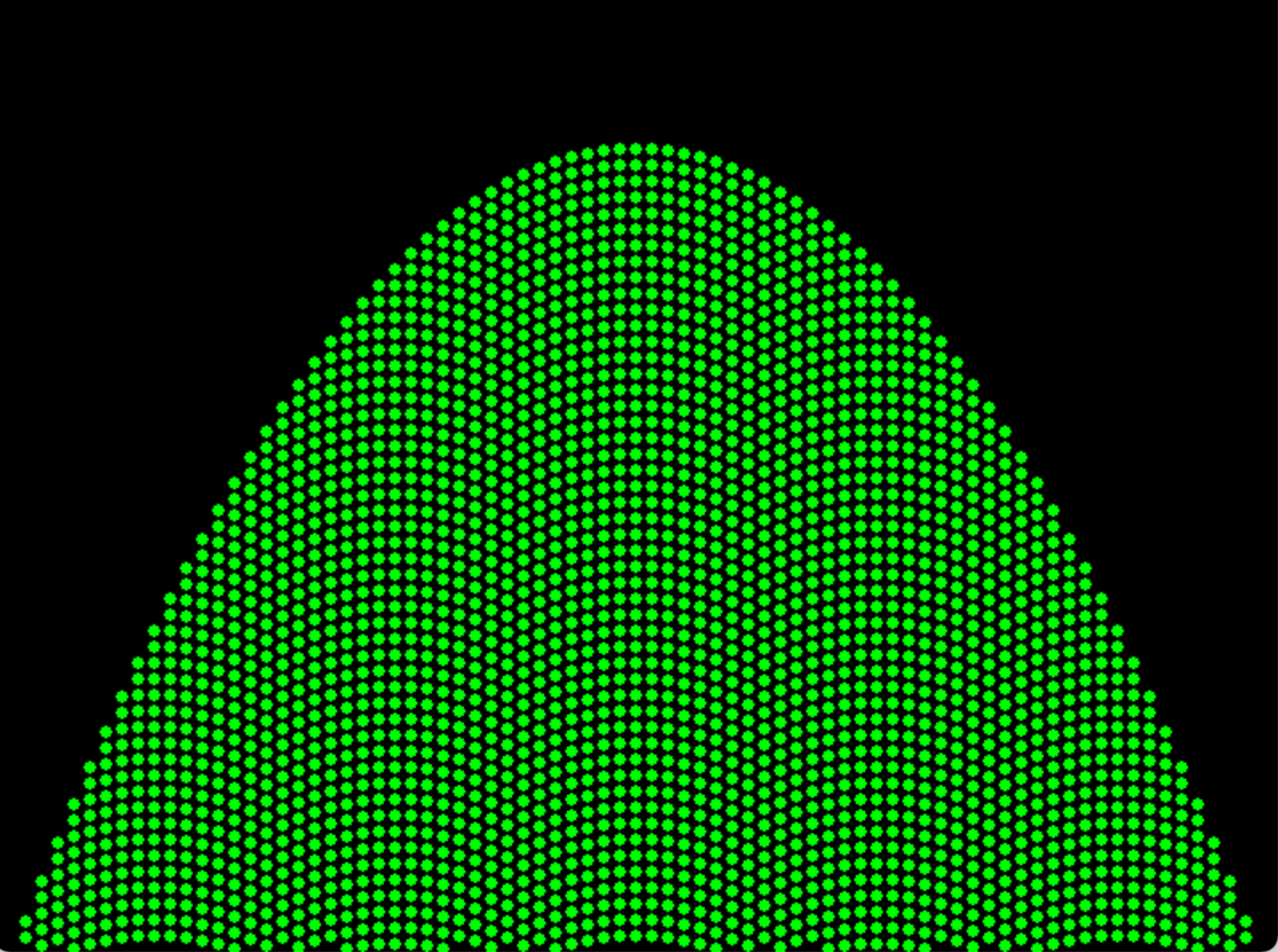


In [1]:

```
# File: g (Python 3.2)

from pygame import *
from random import *
init()
size = (800, 600)
(width, height) = (800, 600)
green = (0, 255, 0)
red = (255, 0, 0)
screen = display.set_mode(size)

def yVal(x):
    return (x - 400) ** 2 // 300 + 100


def line(x, y):
    for py in range(y, height, 10):
        draw.circle(screen, green, (x, py), 4)



def drawScene(screen):
    for x in range(0, width, 10):
        line(x, yVal(x))

    display.flip()

drawScene(screen)
running = True
while running:
    for evnt in event.get():
        if evnt.type == QUIT:
            running = False
            continue
quit()
```

```
pygame 2.1.2 (SDL 2.0.18, Python 3.9.12)
Hello from the pygame community. https://www.pygame.org/contribute.html
```

In [ ]:

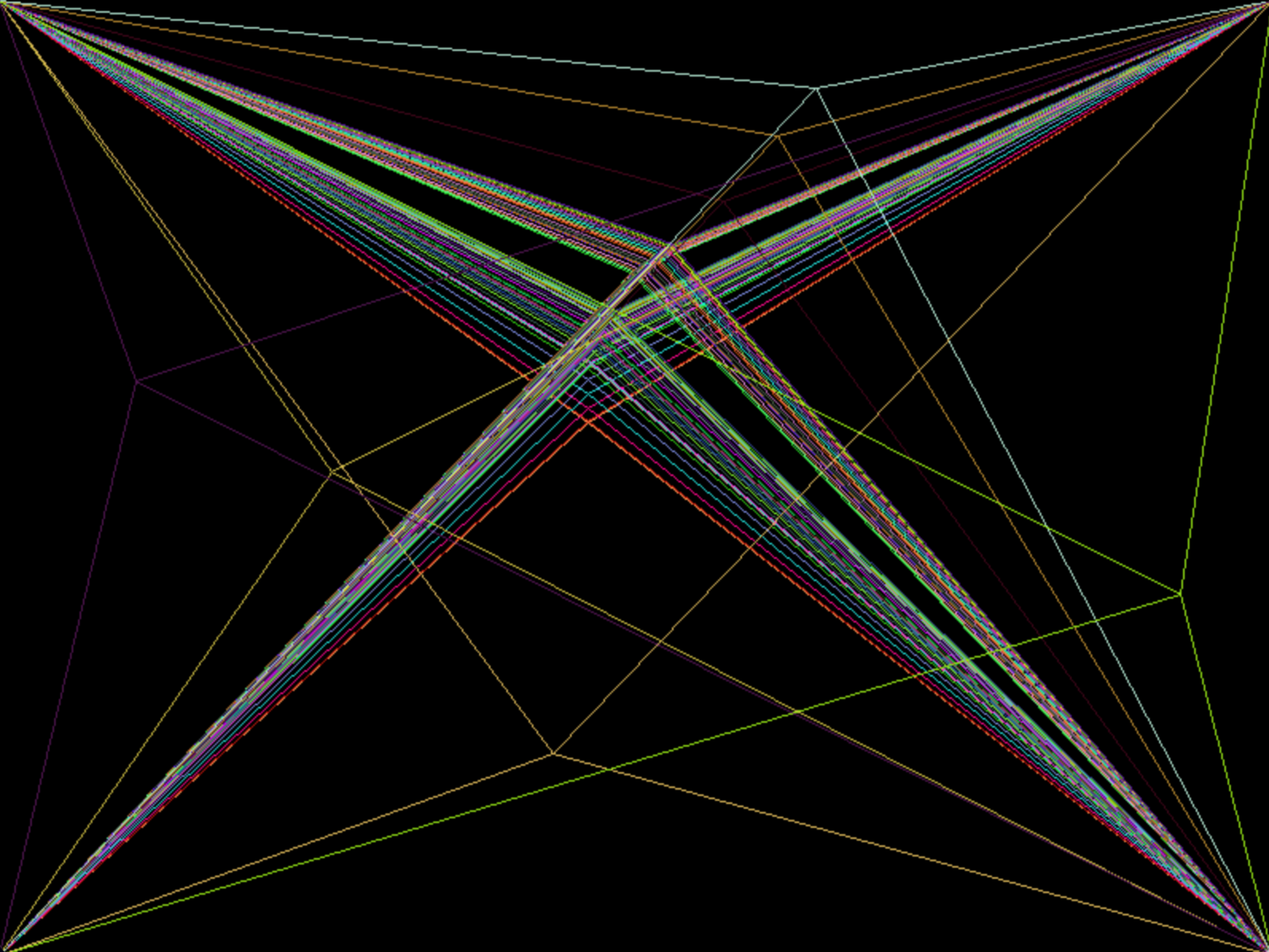


In [1]:

```
# File: g (Python 3.2)

from pygame import *
from random import *
init()
size = (800, 600)
(width, height) = (800, 600)
green = (0, 255, 0)
red = (255, 0, 0)
screen = display.set_mode(size)

def randCol():
    return (randint(0, 255), randint(0, 255), randint(0, 255))


def pattern(screen, x, y):
    c = randCol()
    draw.line(screen, c, (0, 0), (x, y))
    draw.line(screen, c, (width, 0), (x, y))
    draw.line(screen, c, (width, height), (x, y))
    draw.line(screen, c, (0, height), (x, y))


def drawScene(screen):
    (mx, my) = mouse.get_pos()
    if mouse.get_pressed()[0] == 1:
        pattern(screen, mx, my)
    if mouse.get_pressed()[2] == 1:
        screen.fill((0, 0, 0))
    display.flip()

running = True
myClock = time.Clock()
while running:
    for evnt in event.get():
        if evnt.type == QUIT:
            running = False
            continue
    drawScene(screen)
    myClock.tick(60)
quit()
```

```
pygame 2.1.2 (SDL 2.0.18, Python 3.9.12)
Hello from the pygame community. https://www.pygame.org/contribute.html
```

In [ ]:

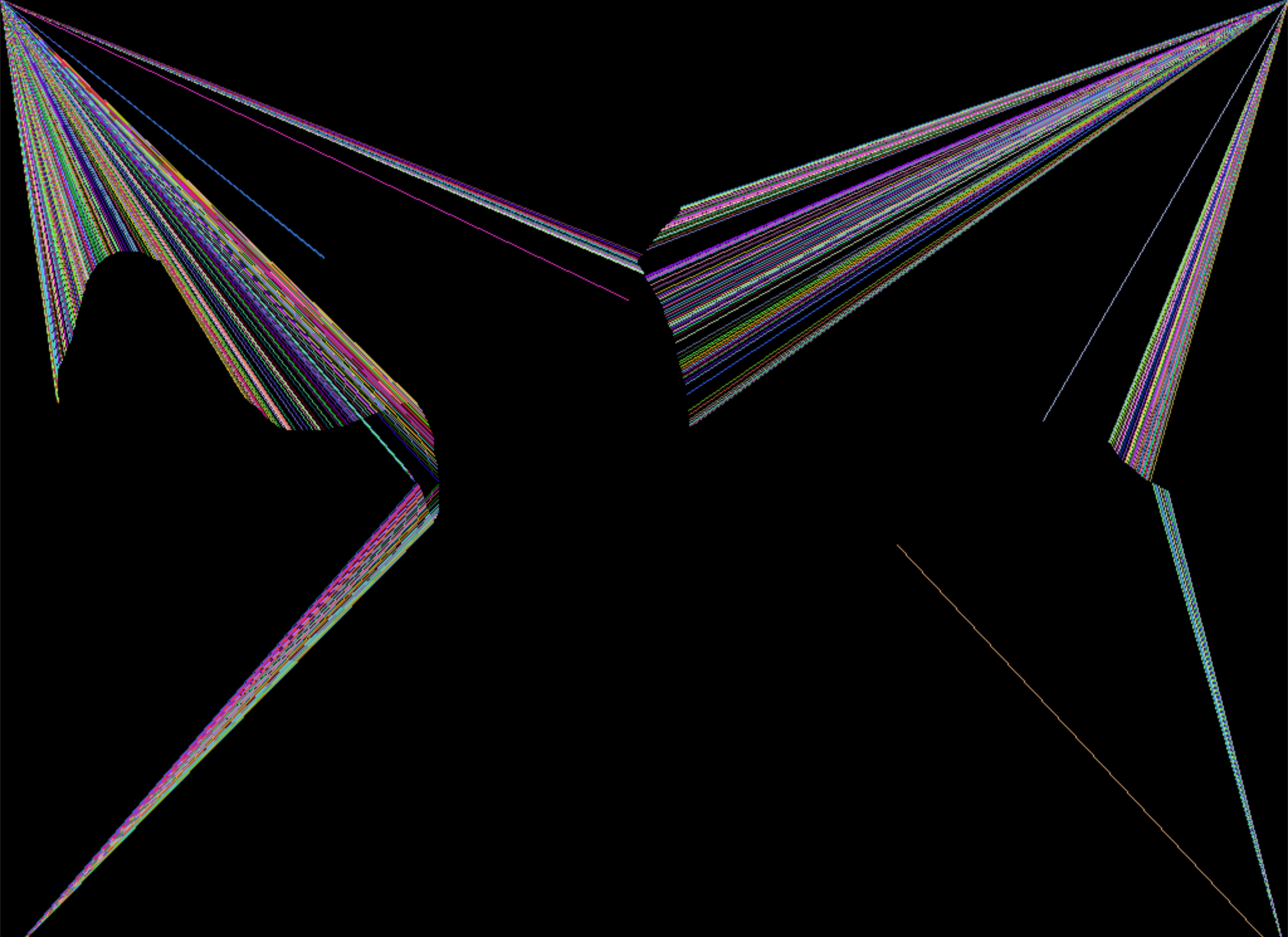


In [1]:

```
# File: g (Python 3.2)

from pygame import *
from random import *
init()
size = (1024, 768)
(width, height) = (1024, 768)
midx = width / 2
midy = height / 2
green = (0, 255, 0)
red = (255, 0, 0)
screen = display.set_mode(size)

def randCol():
    return (randint(0, 255), randint(0, 255), randint(0, 255))


def drawScene(screen):
    (mx, my) = mouse.get_pos()
    if mouse.get_pressed()[0] == 1:
        if mx < midx and my < midy:
            draw.line(screen, randCol(), (0, 0), (mx, my))
        if mx > midx and my < midy:
            draw.line(screen, randCol(), (width, 0), (mx, my))
        if mx < midx and my > midy:
            draw.line(screen, randCol(), (0, height), (mx, my))
        if mx > midx and my > midy:
            draw.line(screen, randCol(), (width, height), (mx, my))

    display.flip()

running = True
myClock = time.Clock()
while running:
    for evnt in event.get():
        if evnt.type == QUIT:
            running = False
            continue
    drawScene(screen)
    myClock.tick(60)
quit()
```

```
pygame 2.1.2 (SDL 2.0.18, Python 3.9.12)
Hello from the pygame community. https://www.pygame.org/contribute.html
```

In [ ]:



In [1]:

```
# File: g (Python 3.2)

from pygame import *
from random import *
init()
size = (800, 600)
(width, height) = (800, 600)
green = (0, 255, 0)
red = (255, 0, 0)
screen = display.set_mode(size)

def randCol():
    return (randint(0, 255), randint(0, 255), randint(0, 255))


def drawScene(screen, cols):
    for x in range(20):
        draw.circle(screen, cols[x], (x * 40 + 20, 260), 19)
        draw.circle(screen, cols[x], (x * 40 + 20, 300), 19)
        draw.circle(screen, cols[x], (x * 40 + 20, 340), 19)

    display.flip()

cols = []
for i in range(20):
    cols.append(randCol())

running = True
myClock = time.Clock()
while running:
    for evnt in event.get():
        if evnt.type == QUIT:
            running = False
            continue
    drawScene(screen, cols)
    c = cols.pop()
    cols = [
        c] + cols
    myClock.tick(10)
quit()
```

```
pygame 2.1.2 (SDL 2.0.18, Python 3.9.12)
Hello from the pygame community. https://www.pygame.org/contribute.html
```

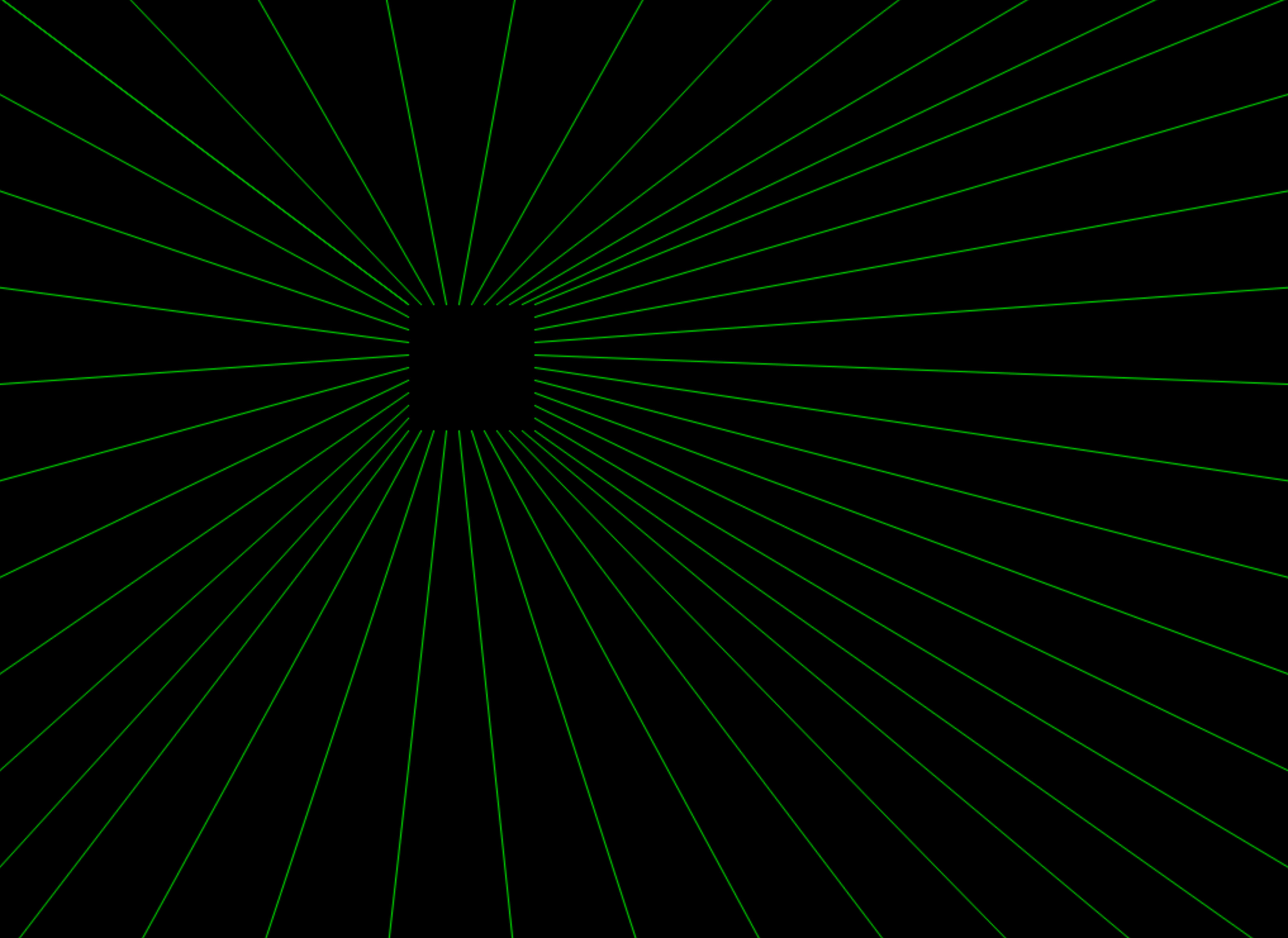
In [ ]:

In [1]:

```python
# File: g (Python 3.2)

from pygame import *
from random import *
init()
size = (1024, 768)
(width, height) = (1024, 768)
midx = width / 2
midy = height / 2
green = (0, 255, 0)
red = (255, 0, 0)
screen = display.set_mode(size)

def randCol():
    return (randint(0, 255), randint(0, 255), randint(0, 255))


def pattern(screen, mx, my):
    bigDy = height / 10
    bigDx = width / 10
    for y in range(10):
        draw.aaline(screen, green, (0, bigDy * y), (mx - 50, my + y * 10 - 50))
        draw.aaline(screen, green, (width, bigDy * y), (mx + 50, my + y * 10 -

    for x in range(10):
        draw.aaline(screen, green, (bigDx * x, 0), (mx + x * 10 - 50, my - 50))

    for x in range(11):
        draw.aaline(screen, green, (bigDx * x, height), (mx + x * 10 - 50, my +



def drawScene(screen):
    (mx, my) = mouse.get_pos()
    screen.fill((0, 0, 0))
    pattern(screen, mx, my)
    display.flip()

running = True
myClock = time.Clock()
while running:
    for evnt in event.get():
        if evnt.type == QUIT:
            running = False
            continue
    drawScene(screen)
    myClock.tick(60)
quit()
```

```
pygame 2.1.2 (SDL 2.0.18, Python 3.9.12)
Hello from the pygame community. https://www.pygame.org/contribute.html
```

In [ ]:



In [1]:

```
# File: g (Python 3.2)

from pygame import *
init()
size = (800, 600)
(width, height) = (800, 600)
green = (0, 255, 0)
red = (255, 0, 0)
screen = display.set_mode(size)
mmode = 'up'

def pattern(screen, x, y):
    for dx in range(-20, 21, 10):
        draw.circle(screen, green, (x + dx, y + 20), 4)
        draw.circle(screen, green, (x + dx, y - 20), 4)

    for dy in range(-20, 21, 10):
        draw.circle(screen, green, (x + 20, y + dy), 4)
        draw.circle(screen, green, (x - 20, y - dy), 4)



def drawScene(screen):
    if mmode == 'down':
        screen.blit(screenBuff, (0, 0))
        (mx, my) = mouse.get_pos()
        pattern(screen, mx, my)
    display.flip()

running = True
myClock = time.Clock()
while running:
    for evnt in event.get():
        if evnt.type == QUIT:
            running = False
            continue
    if mmode == 'up' and mouse.get_pressed()[0] == 1:
        screenBuff = screen.copy()
        mmode = 'down'
        mouse.set_visible(False)
    if mmode == 'down' and mouse.get_pressed()[0] == 0:
        mmode = 'up'
        mouse.set_visible(True)
    if mouse.get_pressed()[2] == 1:
        screen.fill((0, 0, 0))
    drawScene(screen)
    myClock.tick(60)
quit()
```

```
pygame 2.1.2 (SDL 2.0.18, Python 3.9.12)
Hello from the pygame community. https://www.pygame.org/contribute.html
```

In [ ]: