

```
import random
import math
import sys
import pygame

W, H = 1920, 1080
HW, HH = W / 2, H / 2
AREA = W * H

class circle:
    def __init__(self, x, y, id):
        self.x = x
        self.y = y
        self.radius = 2
#id is required to know the position of the new circle with respect to a circle that already exists, this also makes sure that we dont repeat.
        self.id = id
        self.active = True
#this is for making sure the circle is still growing as we want to make sure when it collides with another circle it should stop growing.

circles = list([])
find_space_attempts = 0
max_find_space_attempts = 100000
exit = False
gap = 3

area_covered = 0.0
percentage_covered = 0
last_reported_percentage = -1

while True:
    while True:
        x = random.randint(0, W - 1)
        y = random.randint(0, H - 1)

        found_space = True
        for c in circles:
            distance = math.hypot(c.x - x, c.y - y)
            if distance <= c.radius + gap:
#if distance between the centres is less than the existent circle's radius than the point is going to lie within.
                found_space = False
                break
        if found_space: break
        find_space_attempts += 1
#our finding space attempts is going to end the programme.
        if find_space_attempts >= max_find_space_attempts:
            exit = True
            break
```

```

    if exit: break
    circles.append(circle(x, y, len(circles)))
#if space is found we are going to add the point to our list circles.
#here we let our circle to grow and the which is going to stop it untill it encounters the other circle.
#we are iteratively checking for all of the circle existing in the list
    for c in circles:
        if not c.active: continue
        for C in circles:
            if c.id == C.id: continue
#here is how creating the id have helped us to avoid recounting.
#formulla to check whether the circles overalap or will colide after increasing the radius.
    distance_between_circles = math.hypot(c.x - C.x, c.y - C.y)
    combined_radius = c.radius + C.radius

    if distance_between_circles - combined_radius <= gap:
        c.active = False #so when the circles are sufficiently close we are going to make them inactive i.e we will stop them growing.
        if C.active:
            area_covered += (C.radius ** 2) * math.pi
        C.active = False

    area_covered += (c.radius ** 2) * math.pi
    percentage_covered = int((area_covered / AREA) * 100) #this will report the unique percentages covered.
    if last_reported_percentage != percentage_covered:
        print(percentage_covered)
        last_reported_percentage = percentage_covered
    break

    if c.active: c.radius += 1 #growing the circle.
image = pygame.Surface((W, H))
#drawing the circles using pygame.
for c in circles:
    pygame.draw.circle(image, (255, 255, 255), (c.x, c.y), c.radius, 1)
pygame.image.save(image, "circle.png")

```

```

0
1
2
3
4
5
6
7
8
9
11
12
13
14
15
16
17
18

```

19
20
22
23
24
25
26
27
28
29
30
31
32
33
34
35
37
38
39
40
41
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64

```
from google.colab import files
files.download( "circle.png" )
```

✓ 0s completed at 10:55 PM

