Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

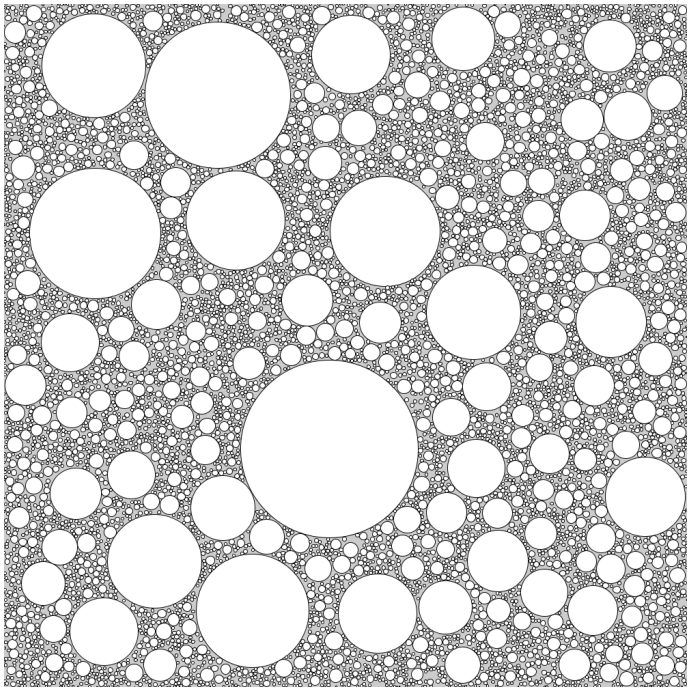
Disciplina: Programação com Interfaces Gráficas

AP3 1^o **semestre de 2018.**

Nome -

Assinatura -

Questão 1 (10,0 pontos) Empacotamento de círculos é um método pelo qual se tenta preencher um retângulo com círculos de tamanhos variados, sem que estes se intersectem. Um método simples consiste em gerar pontos aleatórios dentro do retângulo e computar a distância d ao círculo mais próximo. Se o ponto cair dentro de algum círculo gerado anteriormente, é ignorado. Caso contrário, é gerado um círculo naquela posição com raio menor que d. O código Python listado parcialmente a seguir implementa essa idéia, e uma saída gerada pelo programa é mostrada na figura abaixo.



```
from random import random
from math import *
from tkinter import *
import sys

wsize = 1000

def ponto_aleatorio(xmax,ymax):
    """Retorna uma tupla da forma (x,y) representando um ponto
    aleatório sujeito a 0 <= x <= xmax e 0 <= y <= ymax."""
    ...

def raio_aleatorio (x, y, xmax, ymax, rmin, rmax):
    """Retorna um raio aleatório entre rmin e rmax para um círculo
    centrado em x,y de forma que o círculo esteja todo contido dentro
    do retângulo dado por 0 <= x <= xmax e 0 <= y <= ymax."""
    ...

def distancia_lista(p,l):
    """Retorna a distância entre um ponto p dado por uma tupla
    da forma (x,y) e o círculo mais próximo dentre os círculos
    armazenados na lista l. Cada círculo em l é representado
    por uma tupla da forma (x,y,r), onde x,y são as coordenadas do centro
    e r é o raio. Se p está dentro de algum círculo de r, a função
    retorna 0. Se l está vazia, retorna um número grande."""
    ...

## Lista de círculos dados como (x,y,r).
circulos = []

def draw():
    """Cria um círculo aleatório e o desenha
       caso seu raio seja maior do que 2."""

    global canvas
    p = ponto_aleatorio(wsize,wsize)
    d = distancia_lista(p,circulos)
    if d > 2:
        x,y = p[0],p[1]
        r = raio_aleatorio(x,y,wsize,wsize,2,d)
        canvas.create_oval(x-r,y-r,x+r,y+r)
        circulos.append((x,y,r))

def poll():
     """Continua empacotando círculos, após um certo intervalo de tempo."""

     global root
     draw()
     root.after(33,poll)
```

```
def main():
    """Cria uma canvas e fica desenhando novos círculos a uma taxa de
       aproximadamente 30 círculos por segundo."""

    global canvas, root
    root = Tk()
    canvas = Canvas(root, width=wsize, height=wsize)
    canvas.pack(fill=BOTH, expand=YES)
    poll()
    root.mainloop()

if __name__=='__main__':
    sys.exit(main())
```

Pede-se escrever o código das funções `ponto_aleatorio()`, `raio_aleatorio()` e `distancia_lista()`.

Algumas observações:

1. Lembre-se que a distância entre dois pontos com coordenadas (x_1, y_1) e (x_2, y_2) é: $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.
2. A função `random()` retorna um número aleatório em ponto flutuante entre 0.0 e 1.0.
3. Embora não seja necessário lembrar das funções do tkinter para resolver a questão, você pode achar útil a seguinte recapitulação:
   (a) A função `poll()` é chamada repetidamente pelo tkinter a cada 33ms.
   (b) A função `canvas(root,w,h)` especifica que a janela de desenho tem largura `w` e altura `h`.
   (c) A função `draw()` é chamada automaticamente aproximadamente 30 vezes por segundo.
   (d) A função `create_oval(x0,y0,x1,y1)` desenha uma elipse contida no retângulo com cantos [`(x0,y0)`,`(x1,y1)`].