



Features

Business

Explore

Marketplace

Pricing

This repository Search

Sign in or Sign up

jbenavidesv87 / FlujoRedes

Watch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Insights

Branch: master FlujoRedes / ejemplos / 02Nodos /

Create new file Find file History

jbenavidesv87 Reporte del ejemplo 4. Latest commit 0c7fa26 Mar 5, 2018

..		
grafo.eps	Imágenes en eps	Mar 5, 2018
grafo.gnu	Imágenes en eps	Mar 5, 2018
grafo.png	-Controlada la eventualidad de que un usuario nombrara a un grafo con...	Mar 1, 2018
main.py	Reporte del ejemplo 4.	Mar 5, 2018
readme.md	Reporte del Ejemplo 3, terminado.	Mar 1, 2018

readme.md

Ejemplo 2. Nodos

En este ejemplo se crearán nodos y se modificarán sus propiedades iniciales manualmente y luego de manera aleatoria mediante el uso de las funciones `random` y `randint` de la librería `random` de `python`. Al usar `random()` se genera un número decimal aleatorio entre 0 y 1, ambos incluidos; y `randint(a, b)` genera un número entero entre `a` y `b`, ambos incluidos.

```
from Grafo import Grafo
from Nodo import Nodo
from random import random, randint # librerías de random
```

Primero se define un número total de nodos a generar, `N`, y se define un grafo `G` que los contendrá.

```
N = 20 # Nodos totales

G = Grafo()
```

Enseguida, se creará un nodo `n` al que se le modificarán sus propiedades manualmente. Se borrará su identificador para que aparezca sin etiqueta, se colocará en el origen del eje de coordenadas, su radio que se igualará a 0.2 y se hará rojo.

```

n = Nodo() # Crea un nodo almacenado en n
n.id = "" # Esto permite eliminar la etiqueta del nodo
n.posicion = (0, 0) # Centro del nodo en la coordenada (0, 0)
n.radio = 0.2 # Radio del nodo en 0.2 puntos
n.Color( # Color con valores de 0 (mínimo) a 255 (máximo):
    255, # rojo
    0, # verde
    0, # azul
    0 # alfa (transparencia mínima, es decir, color sólido)
)
G.AgregarNodo(n) # Agrega el nodo al grafo

```

Posteriormente, se crearán N nodos con la ayuda de la función `range(N)`, la cual devuelve una lista de enteros de 0 hasta $N - 1$. A cada uno de estos $i \in [0, 1, 2, \dots, N - 1]$ nodos se les modificarán sus propiedades iniciales:

- Identificador: i
- Posición: Par ordenado con valores al azar entre 0 y 1 por componente
- Radio: Al azar entre 0.05 y 0.1
- Color: Componentes rojo, verde y azul con valores al azar; transparencia media, 128 (mediana del rango $[0, 1, \dots, 255]$)

Después se agregarán al grafo. Estas acciones se realizan con el siguiente código:

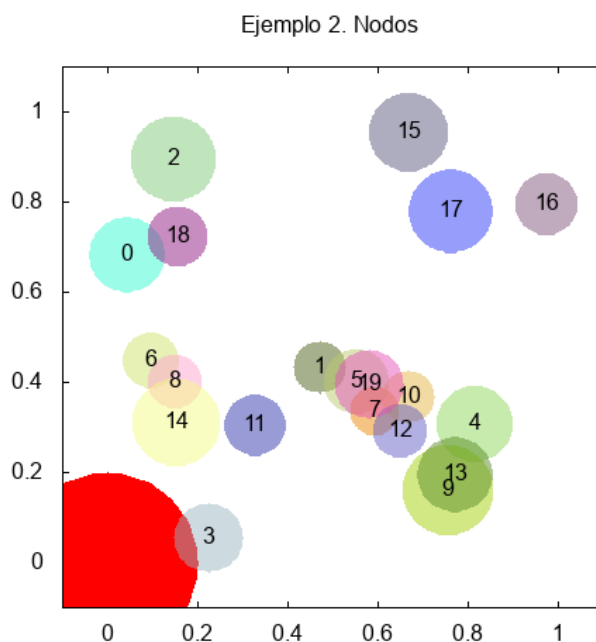
```

[... ]
for i in range(N): # Para todo i en [0, 1, ..., N - 1]
    n = Nodo() # Se crea el nodo i
    n.id = i # Se le asigna el identificador i
    n.posicion = (random(), random()) # Una posición al azar
    n.radio = 0.05 + 0.05 * random() # Un radio entre 0.5 y 1
    n.Color( # Color con componentes:
        randint(0, 255), # rojo (R)
        randint(0, 255), # verde (G)
        randint(0, 255), # azul (B)
        128, # alfa (A); transparencia
    )
    G.AgregarNodo(n) # Se agrega el nodo i al grafo

```

Al final, se desplegará este grafo en un eje de coordenadas que lleve por título **Ejemplo 2. Nodos**, almacenado en una imagen PNG nombrada **grafo** por el nombre por defecto con que se genera el grafo con la instrucción `G.DibujarGrafo("Ejemplo 2. Nodos")`.

Una de las imágenes resultantes es:



© 2018 GitHub, Inc.

[Terms](#)
[Privacy](#)
[Security](#)
[Status](#)
[Help](#)



[Contact GitHub](#)
[API](#)
[Training](#)
[Shop](#)
[Blog](#)
[About](#)