Université Polytechnique des Hauts-De-France
Institut des Sciences et Techniques ISTV

Licence 1 : TP d'Algorithmique Semestre2 - 2018
Rosaces

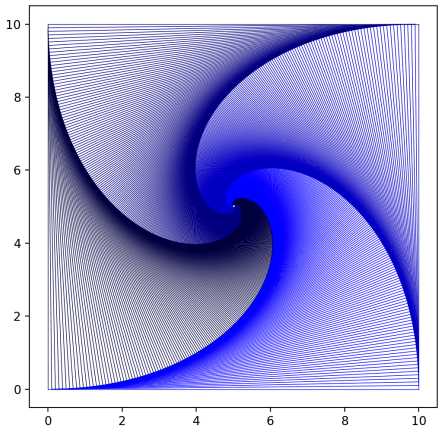
L'objectif est de suivre le mouvement de personnes attirées vers d'autres.

Un Personne a une position (x,y) sur un plan; est attirée vers une autre personne (son 'flash'); et sera identifiée par une couleur :

```
class Personne:
    x=0
    y=0
    flash = None
    couleur = (0,0,0)
```

Par exemple, ci-dessous, 4 personnes sont dans le plan. Initialement, P1 est en (0,0), P2 est en (10,0), P3 est en (10,10) et P4 est en (0,10). P1 a flashé sur P2, P2 sur P3, P3 sur P4 et P4 sur P1.

Chaque personne regarde la personne objet de son flash, et avance vers elle, doucement. On dessine à chaque étape le chemin qui sépare une personne de l'être visé.



Le dessin se fera avec la librairie `matplotlib.pyplot` (résumé par `plt`) dédiée plutôt aux dessins de courbes (2D ou 3D), d'histogrammes…
La fonction `plot` prend en entrée l'ensemble des coordonnées x des points et l'ensemble des coordonnées y.
Ainsi, tracer une ligne entre deux points `(x1,y1)` et `(x2,y2)` s'écrit : `plt.plot([x1,x2],[y1,y2])`

Après avoir poser les traits, pour afficher le dessin, on ajoute ensuite : `plt.show()`

Voici ce qu'il faut placer en tête du fichier python :

```
from numpy import zeros, array
import matplotlib.pyplot as plt
from typing import Iterable

##definition des types :
class Personne:
    x=0
    y=0
    flash = None
    couleur = (0,0,0)
TabPersonnes = Iterable[Personne]
Couleur = (float, float, float)
```

Inclure la fonction **`creer_personne(x:float, y:float)`** crée une Personne et lui affecte ses coordonnées et la retourne :

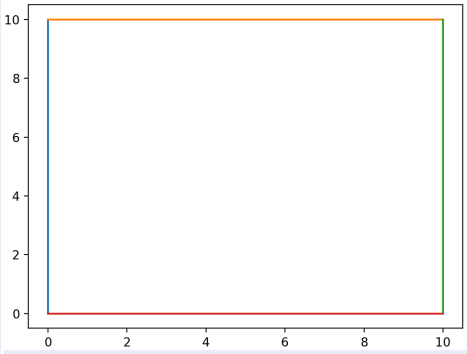
```
def creer_personne(x:float, y:float)->Personne:
    p:Personne = Personne()
    p.x = x
    p.y = y
    return p
```

---

1. Définir la fonction **`ajouter_personne(x:float, y:float, tab:TabPersonnes, n:int)->int:`** qui crée une personne de coordonnées (x,y) et l'ajoute en position n du tableau tab. La fonction retourne (n+1).
2. Définir la procédure **`flasher_personnes(tab:TabPersonnes, n:int)`** qui prend un tableau de n personnes en paramètre et qui, pour chaque personne en position i, affecte à la valeur **`flash`** la personne suivante dans le tableau (i+1). La dernière personne du tableau flashe sur la première. *N.B. pour affecter `p2` au flash de `p1`, on écrit : `p1.flash = p2`*
3. Définir la procédure **`dessiner_regards(tab:TabPersonnes, n:int)`** qui trace une ligne entre le point (x,y) de chaque personne p, et le point (x,y) de la personne *flashée* par p *(donc pour une personne p, il faut tracer un trait entre (p.x, p.y) et (p.flash.x, p.flash.y)), donc en entrée de la fonction plot on passe comme paramètre [p.x, p.flash.x] et [p.y, p.flaxh.y].*
4. Tester les algos précédents à l'aide de cette fonction :

Méthodes de test	***Figure obtenue***

```
def test_algos():
    #creation d'un tableau pour 10 personnes :
    tab:TabPersonnes = zeros(10,Personne)
   #ajout de 4 personnes :
    n=0
    n = ajouter_personne(0,0,tab,n)
    n = ajouter_personne(10,0,tab,n)
    n = ajouter_personne(10,10,tab,n)
    n = ajouter_personne(0,10,tab,n)
    #mise en relation des personnes
    flasher_personnes(tab,n)
    #test du dessin
    dessiner_regards(tab, n)
    #lancer l'affichage du graphique
    plt.show()
```



5. Définir la procédure **`avancer_personne(tab:TabPersonnes, n:int)`** qui fait avancer chaque personne i du tableau vers l'objet de son flash de 5% de la distance qui les sépare.
   Aide: pour que p1(x,y) avance de 5% vers p2(x,y), on ajoute à p1.x 5% de la différence en x entre p2 et p1, idem pour p2.y.
6. Définir la procédure **`rosace()`** qui reprend le code de test_algos, mais qui remplace l'appel à la procédure **`dessiner_regards(tab, n)`** par 100 appels à **`dessiner_regards(tab, n)`** et **`avancer_personne(tab, n)`**. Tester la fonction.. La rosace apparaît avec différentes couleurs.

---

7. *Dégradé de couleur.* Une couleur est définie par 3 teintes **R**ouge, **V**ert, **B**leu. En Python, (1,1,1) correspond au blanc (100% de rouge, de vert de de bleu); (0,0,0) correspond au noir; (1,0,0) correspond au rouge; (0,1,0) au vert et (0,0,1) au bleu. Jouer sur ces paramètres permet d'utiliser un nuancier de couleurs.
   Définir la fonction **`dessiner_regards_colores(tab:TabPersonnes, n:int)`** qui qui trace une ligne colorée entre chaque personne p, et son flash, en utilisant la couleur de p. La procédure de dessin s'appelle ainsi : **`plt.plot([p.x, p.y], [p.elu.x, p.elu.y], color=p.couleur)`**.
8. Définir la procédure **`rosace_coloree()`** qui reprend **`rosace()`** mais en affectant la couleur 'bleu' (0,0,1) à la première personne du tableau et un bleu de plus en plus sombre aux autres personnes du tableau.
   On écrit ainsi pour la 1ère personne :
   `c:Couleur = (0,0,1)`
   `tab[0].couleur = c`
   Accéder à la 3e valeur de **`c`** s'écrit **`c[2]`**. Ainsi **`c[2] = c[2] - 0.25`** donnera un bleu plus sombre (plus proche de 0).
   Créer 8 personnes en (0,0), (5,0), (10,0), (10,5), (10,10), (5,10), (0,10) et (0,5); avec un dégradé de bleu.
   Faire 500 appels à **`dessiner_regards_colores(tab, n)`** et à **`avancer_personne(tab, n)`.**

Pour améliorer l'affichage, il est possible d'ajouter le paramètre **`linewidth=0.5`** dans l'appel à **`plt.plot(…)`** et de ralentir l'approche dans la procédure **`avancer_personne`** .