"""

INFO-F101 - Programmation

Projet 2 : Vasarely doit se retourner dans sa tombe

Programme qui dessine un pavage d'hexagones, divisés

en 3 losanges de couleurs, auquel on applique une déformation

"""

\_\_author\_\_ **=** 'Charlotte Nachtegael - 000425654 - Groupe TP 1'

\_\_date\_\_ **=**'21 octobre 2015'

# -\*-coding:Utf-8 -\*

**import** turtle

inf\_gauche **=** int**(**input**(**"Borne inférieure gauche : "**))**

sup\_droit **=** int**(**input**(**"Borne supérieure droite : "**))**

longueur **=** int**(**input**(**"Longueur du rayon de l'hexagone : "**))**

col1 **=** input**(**"Couleur 1 de l'hexagone : "**)**

col2 **=** input**(**"Couleur 2 de l'hexagone : "**)**

col3 **=** input**(**"Couleur 3 de l'hexagone : "**)**

x\_centre **=** int**(**input**(**"Abscisse du centre de la déformation : "**))**

y\_centre **=** int**(**input**(**"Ordonnée du centre de la déformation : "**))**

z\_centre **=** int**(**input**(**"Hauteur du centre de la déformation : "**))**

centre **=** **(**x\_centre**,** y\_centre**,** z\_centre**)**

r **=** int**(**input**(**"Longueur du rayon de la déformation : "**))**

EPS **=** 1.0e-5

turtle**.**title**(**"Vasarely doit se retourner dans sa tombe"**)**

**def** deformation**(**p**,**c**,**r**):**

"""reçoit le point p =(x\_p,y\_p,z\_p), le centre c=(x\_c,y\_c,z\_c),

et le rayon r tous les trois avant déformation,

renvoit un point p2 après déformation

"""

**def** distance**(**p**,**q**):**

"""distance entre les 2 points p et q"""

**return** **((**q**[**0**]-**p**[**0**])\*\***2 **+** **(**q**[**1**]-**p**[**1**])\*\***2 **+** **(**q**[**2**]-**p**[**2**])\*\***2**)\*\***0.5

d **=** distance**(**p**,**c**)**

**if** d **>=** r**:**

res **=** p

**else:**

**if** d **>** EPS**:**

facteur **=** r**/**d

**else:**

facteur **=** 1

x2 **=** c**[**0**]** **+** **(**p**[**0**]-**c**[**0**])** **\*** facteur

y2 **=** c**[**1**]** **+** **(**p**[**1**]-**c**[**1**])** **\*** facteur

z2 **=** c**[**2**]** **+** **(**p**[**2**]-**c**[**2**])** **\*** facteur

res **=** **(**x2**,**y2**,**z2**)**

**return** res

deform **=** **lambda** p**:** deformation**(**p**,**centre**,**r**)** # définition de la fonction deform envoyée à pavage

**def** list\_pos**(**c**,** longueur**,** vas**):**

"""Reçoit le centre c avant déformation et donne la liste des

coordonnées des points de l'hexagone de rayon longueur sans

déformation

"""

res **=** **[]**

vas**.**penup**()**

vas**.**hideturtle**()**

vas**.**goto**(**c**)**

vas**.**forward**(**longueur**)**

vas**.**left**(**120**)**

**for** i **in** range**(**6**):**

vas**.**forward**(**longueur**)**

vas**.**left**(**60**)**

res**.**append**(**vas**.**pos**())**

**return** res

**def** hexagone**(**vas**,** c**,** longueur**,**col1**,** col2**,** col3**,** deform**):**

"""Reçoit la coordonnée du centre avant déformation et dessine

un hexagone de rayon longueur autour de celui-ci, dont les

points et le centre ont été déformés selon deform.

L'hexagone est composé de trois losanges de couleurs

(col1, col2 et col3)

"""

z **=** 0

list\_positions **=** list\_pos**(**c**,** longueur**,** vas**)**

c\_new **=** deformation**((**c**[**0**],** c**[**1**],** z**),** centre**,** r**)**

p1 **=** deformation**((**list\_positions**[**0**][**0**],** list\_positions**[**0**][**1**],** z**),** centre**,** r**)**

p2 **=** deformation**((**list\_positions**[**1**][**0**],** list\_positions**[**1**][**1**],** z**),** centre**,** r**)**

p3 **=** deformation**((**list\_positions**[**2**][**0**],** list\_positions**[**2**][**1**],** z**),** centre**,** r**)**

p4 **=** deformation**((**list\_positions**[**3**][**0**],** list\_positions**[**3**][**1**],** z**),** centre**,** r**)**

p5 **=** deformation**((**list\_positions**[**4**][**0**],** list\_positions**[**4**][**1**],** z**),** centre**,** r**)**

p6 **=** deformation**((**list\_positions**[**5**][**0**],** list\_positions**[**5**][**1**],** z**),** centre**,** r**)**

vas**.**showturtle**()**

vas**.**penup**()**

vas**.**goto**(**c\_new**[**0**],** c\_new**[**1**])**

vas**.**pendown**()**

vas**.**color**(**col1**,** col1**)**

vas**.**begin\_fill**()**

vas**.**goto**(**p6**[**0**],** p6**[**1**])**

vas**.**goto**(**p1**[**0**],** p1**[**1**])**

vas**.**goto**(**p2**[**0**],** p2**[**1**])**

vas**.**goto**(**c\_new**[**0**],** c\_new**[**1**])**

vas**.**end\_fill**()**

vas**.**color**(**col2**,** col2**)**

vas**.**begin\_fill**()**

vas**.**goto**(**p4**[**0**],** p4**[**1**])**

vas**.**goto**(**p5**[**0**],** p5**[**1**])**

vas**.**goto**(**p6**[**0**],** p6**[**1**])**

vas**.**goto**(**c\_new**[**0**],** c\_new**[**1**])**

vas**.**end\_fill**()**

vas**.**color**(**col3**,** col3**)**

vas**.**begin\_fill**()**

vas**.**goto**(**p2**[**0**],** p2**[**1**])**

vas**.**goto**(**p3**[**0**],** p3**[**1**])**

vas**.**goto**(**p4**[**0**],** p4**[**1**])**

vas**.**goto**(**c\_new**[**0**],** c\_new**[**1**])**

vas**.**end\_fill**()**

vas**.**setheading**(**0**)** # met la tortue Vasarely en place pour le prochain hexagone

**def** pavage**(**vas**,** inf\_gauche**,** sup\_droit**,** longueur**,** col1**,** col2**,** col3**,** deform**):**

"""Trace un pavage composé d'hexagones dont le coin inférieur

est donné par inf\_gauche et le coin supérieur droit par sup\_droit.

"""

vas**.**speed**(**1000**)**

centre\_inf\_gauche **=** **(**inf\_gauche**,** inf\_gauche**)**

centre\_sup\_droit **=** **(**sup\_droit**,** sup\_droit**)**

i **=** 0

x\_c **=** centre\_inf\_gauche**[**0**]**

y\_c **=** centre\_inf\_gauche**[**1**]**

**while** x\_c **<=** centre\_sup\_droit**[**0**]** **and** y\_c **<=** centre\_sup\_droit**[**1**]:**

# dessine jusqu'à atteindre le coin sup droit

hexagone**(**vas**,** **(**x\_c**,** y\_c**),** longueur**,** col1**,** col2**,** col3**,** deform**)**

x\_c **+=** 3 **\*** longueur

**if** x\_c **>** centre\_sup\_droit**[**0**]** **and** i **==** 0**:**

x\_c **=** centre\_inf\_gauche**[**0**]** **+** 1.5 **\*** longueur

y\_c **+=** **(**3**\*\***0.5**/**2**)** **\*** longueur

i **=** 1

**elif** x\_c **>** centre\_sup\_droit**[**0**]** **and** i **==** 1**:**

x\_c **=** centre\_inf\_gauche**[**0**]**

y\_c **+=** **(**3**\*\***0.5**/**2**)** **\*** longueur

i **=** 0

vas**.**exitonclick**()**

pavage**(**turtle**,** inf\_gauche**,** sup\_droit**,** longueur**,** col1**,** col2**,** col3**,** deform**)**