

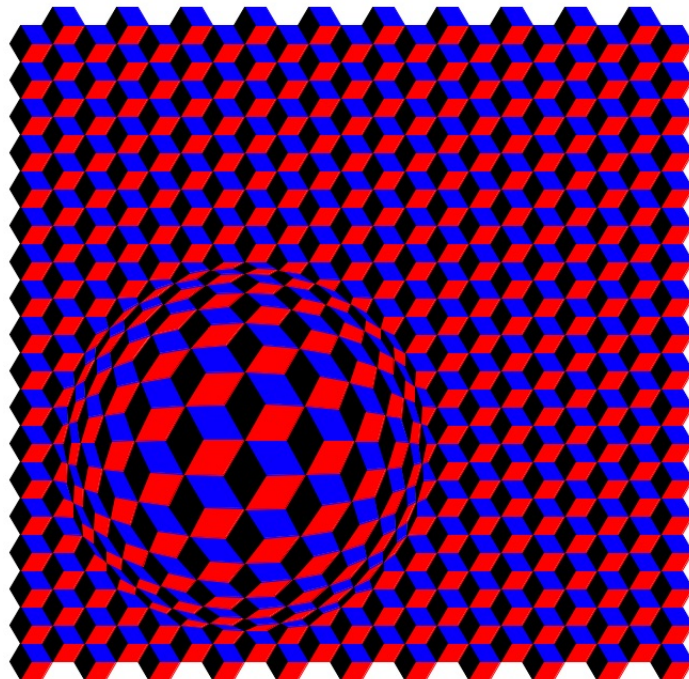
INFO-F101 – Programmation
Projet 2
"Vasarely doit se retourner dans sa tombe"
version 16 octobre 2015

Année académique 2015–2016

Victor Vasarely, né Győző Vásárhelyi le 9 avril 1906 à Pécs (Hongrie) et mort le 15 mars 1997 à Paris, est un plasticien hongrois, naturalisé français en 1961, reconnu comme étant le père de l'art optique. Wikipedia.

Projet

Dans ce deuxième projet de programmation, il vous est demandé de réaliser un programme en Python produisant une imitation d'un tableau de Victor Vasarely.



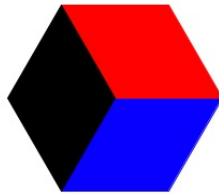
Ce tableau représente un pavage hexagonal, formé avec des losanges de couleurs différentes, déformé par une boule. Votre programme utilisera le module turtle pour peindre vos tableaux. En pratique on vous demande.

1. d'écrire une fonction `hexagone(t, c, longueur, col1, col2, col3, deform)` qui reçoit
 - la tortue `t`

- le centre c (avant déformation) où l'hexagone doit être peint
- la distance (avant déformation) longueur entre le centre et n'importe quel coin de l'hexagone
- les trois couleurs $col1$, $col2$, $col3$
- la fonction `deform` qui reçoit en paramètre un point $p = (x, y, z)$ et renvoie un point $p2 = (x2, y2, z2)$ après déformation,

La fonction `peint` un hexagone déformé en traçant des lignes droites entre le centre et les coins dont la position est calculée avec la fonction `deform`.

Conseil : utilisez la méthode `goto(x,y)` de `turtle`.



hexagone sans déformation

2. d'écrire une fonction `pavage(t, inf_gauche, sup_droit, longueur, col1, col2, col3, deform)`, qui reçoit
- la tortue `t` ;
 - des valeurs `inf_gauche` et `sup_droit` donnant une fenêtre dont le coin inférieur gauche est `(inf_gauche, inf_gauche)` et le coin supérieur droit est `(sup_droit, sup_droit)` (on représente uniquement les axes des x et des y , la hauteur du pavage avant transformation étant égale à 0 ;
 - la longueur `longueur` entre le centre et n'importe quel coin de l'hexagone avant déformation ;
 - les trois couleurs `col1`, `col2`, `col3` ;
 - la fonction `deform` telle que décrite plus haut ;

La fonction `pavage` peint les hexagones déformés dont les centres, avant déformation, se trouvent dans la fenêtre (**les bords inclus**) avec l'hexagone au bas à gauche, avant déformation, centré sur le point `(inf_gauche, inf_gauche)`. Pour cela, elle utilise la fonction `hexagone`.



première ligne inférieure peinte



deux premières lignes inférieures peintes

Le tableau au début de l'énoncé a été généré avec les paramètres `inf_gauche=-300`, `sup_droit=300`, `longueur=20`, `col1='red'`, `col2='blue'`, `col3='black'`, `centre=(-100,-100,-100)`, `r=200`

Pour vous aider, nous vous donnons le code de la fonction deform

```
EPS = 1.0e-5 # à mettre au début du programme
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 tout les trois avant déformation,
    revoie 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
```

Remarques :

- Vous pouvez utiliser la librairie math
- Les calculs se font sur des points en trois dimensions. Le pavage avant déformation se trouve sur le plan avec $z = 0$. Par contre turtle ne tient compte que des deux premières dimensions. La valeur de z n'étant pas envoyée par exemple aux `goto(x,y)`.

Consignes pour la remise du projet

Les consignes pour la remise du projet sont disponibles en ligne sur la page du cours sur l'Université Virtuelle.

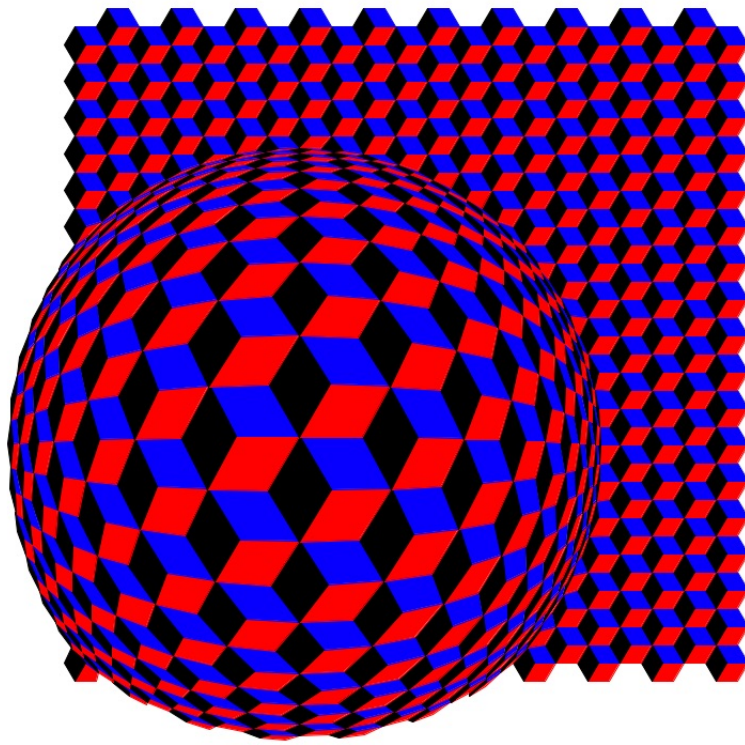
Ces consignes sont à respecter *scrupuleusement* ; relisez-les attentivement avant la remise !

Le projet est à remettre en version électronique et version papier.

Pour toute question concernant l'énoncé du projet adressez-vous à Thierry Massart.

E-mail : tmassart@ulb.ac.be

Date limite de remise. Le lundi 26 octobre 2015 à 13h.



même exemple qu'au début, mis à part le rayon=300