TP info 5° : **à la manière de Vasarely - 02** (arts et maths)

Remarque : pour ce TP, on va contrôler la couleur des objets pleins.
Quand on choisit une couleur pour un objet, il s'agit de la couleur de son contour.
Mais quand on a rendu un objet plein (en utilisant le CaRScript SetFilled), si on met un contour "épais" ("thick" en anglais et dans CaRMetal) à l'objet, le contour s'étend à tout l'objet, qui prend alors la couleur du contour.

Le CaRScript pour modifier le contour est SetThickness.

Pour créer un disque rouge à partir d'un cercle rouge contenu dans c, on écrira donc :

```
SetFilled(c,true) ;
SetThickness(c,"thick") ;
```

---

On repart d'un carré 11 × 11 de disques cyan de rayon 0,44.

Tester :

```
P = new Array() ;
var cp = 0 ;
C = new Array() ;
var cc = 0 ;
for (j=-5 ; j<6 ;j++) {
      for (i=-5 ; i<6 ; i++) {
            P[cp] = Point(i,j) ;
            C[cc]=FixedCircle(P[cp],0.44) ;
            SetFilled(C[cc],true) ;
            SetColor(C[cc],"cyan") ;
            cp++ ;
            cc++ ;
            }
      }
```

Pour l'instant, les disque sont transparents (l'intérieur semble plus clair), on réglera cela plus tard.

Les points ne sont pas très esthétiques, on commence par les masquer avec le CaRScript **SetHide**(..., true).
Modifier le script et tester.

On va ensuite tracer des disques noirs par dessus. Ces disques seront de plus en plus petits à mesure que l'on s'éloigne du centre.
On va donc utiliser les variables de boucle i et j pour définir leur rayon.
Si on enlève leur signe, i et j augmentent en même temps que la distance du centre.

Math.abs(i) donne i sans son signe. Math.abs(j) donne j sans son signe.

Pour le rayon des disques noirs, on va prendre $0,44 - Math.abs(i) \times 0,04 - Math.abs(j) \times 0,04$.

**Exercice 1 :**

Construire le script qui affiche les disques cyans puis les disques noirs avec le rayon diminué.

Les disques noirs sont créés dans une nouvelle double boucle séparée (on duplique et modifie la précédente).

On souhaite maintenant décaler les disques noirs vers le centre. On va donc modifier le centre des disques noirs.
Dans la double boucle, pour les centres des disques noirs, au lieu de construire le point aux coordonnées (i,j), on va enlever 2% à chaque coordonnées.

**Exercice 2 :**

Construire le script. Tester.

On va maintenant rendre les disques opaques.

Modifier le script en utilisant le CaRScript SetThickness (comme expliqué plus haut). Tester.

Si on pouvait choisir précisément les couleurs des disques, ce serait mieux.
Eh bien, c'est possible, et on va le faire.

**Gestion de la couleur**

Comment gère-t-on la couleur dans un logiciel de photo ou de dessin ?
On utilise le modèle RVB (rouge, vert, bleu), autrement dit le modèle RGB (red, green, blue) en anglais.

Une couleur est obtenue en mélangeant du rouge, du bleu et du vert.
Ces trois quantités peuvent varier entre 0 et 255 :
255 de rouge, 0 de bleu et 0 de vert donne du rouge pur.

Il s'agit d'un modèle de « couleurs additives » : on va du noir au blanc.
0 de rouge, 0 de bleu et 0 de vert donne du noir pur.
255 de rouge, 255 de bleu et 255 de vert donne du blanc pur.

Actuellement, on a un cyan (mélange de bleu et de vert) $\approx$ RGB(0,124,124) et un noir pur RGB(0,0,0).

On va choisir des couleurs personnaliséees.
Comment faire ? On va aller sur un site de nuanciers et choisir deux jolies couleurs.

Aller sur le site `kuler.adobe.com`, choisir un nuancier, cliquer sur le bouton avec des curseurs...
On obtient le code RGB des couleurs. Choisir deux couleurs et relever leur code RGB.

Pour modifier la couleur d'un objet X, on utilise le CaRScript SetRGBColor(X,a,b,c) où :
a est la quantité de rouge, b est la quantité de vert, et c la quantité de bleu.

**Exercice 3 :**
Modifier le script pour faire afficher les couleurs choisies. Tester.

On va maintenant tracer un disque de disques. Comment faire ?
On va partir du script précédent en $11 \times 11$, mais on va modifier ce script pour tracer le disque de la double boucle uniquement quand la distance du centre à l'origine est inférieure à 5.

Si x et y contiennent des points, le CaRScript Distance(x,y) donne la distance de x à y.

**Exercice 4 :**
Construire le script. Tester. Ajuster les paramètres pour l'esthétique...

Remarque : Pour mettre une partie de code Javascript en commentaire, on peut :

- utiliser // pour faire un commentaire court (tout ce qui se trouve après sur la ligne est en commentaire)
- encadrer une partie par /* et */ pour faire un commentaire long (tout ce qui est encadré est en commentaire)

**Exercice 5 :**
Figure libre...

Remarque : Pour construire un polygone, on utilise le CaRScript Polygon avec une syntaxe un peu particulière :

Si s, t, u, v sont des points, l'instruction z = Polygon("_s,_t,_u,_v") ; construit le polygone stuv et le met dans z.