Algorithmique et programmation
TP 7 : Le langage Java, de jolis dessins

1 Introduction

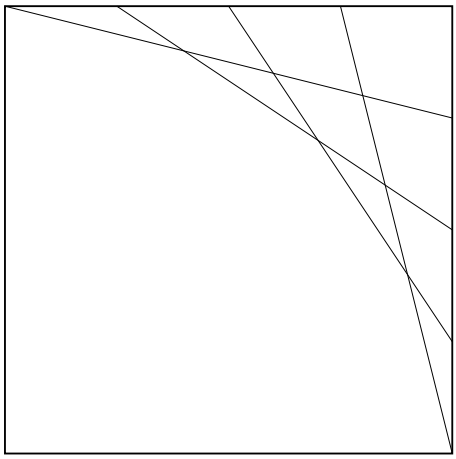
Jusqu'à présent, toutes les entrées/sorties était faite en mode texte dans le terminal. Pour les exercices proposés ci-dessous, il s'agit de dessiner des choses plus ou moins compliquées dans une nouvelle fenêtre. Vous utiliserez pour cela la petite bibliothèque graphique disponible en ligne à l'adresse `http://info.iut-bm.univ-fcomte.fr/staff/giersch/enseignement/Permanent/AlgoProg/tp/graph/`.

Vous consulterez la documentation et les quelques exemples disponibles à la même adresse, afin de comprendre l'utilisation de la bibliothèque.

2 Exercices

2.1 Des lignes qui se croisent

En utilisant la méthode drawLine(**int**, **int**, **int**, **int**), écrire un programme reproduisant la figure suivante.



La figure est composée d'un ensemble de lignes qui se croisent. Il n'y a que 4 lignes sur l'exemple, mais la solution devra bien évidemment fonctionner pour un nombre quelconque de lignes ! Soit L la largeur de la fenêtre, H sa hauteur, et deux entiers d_x et d_y. Les lignes ont alors les coordonnées suivantes :

- $(0, 0)$ à $(L - 1, d_y)$
- $(d_x, 0)$ à $(L - 1, 2d_y)$
- $(2d_x, 0)$ à $(L - 1, 3d_y)$
- $(3d_x, 0)$ à $(L - 1, 4d_y)$
- etc., jusqu'à atteindre l'autre extrémité de la fenêtre.

Rappel : c'est le coin en haut à gauche qui porte les coordonnées $(0, 0)$.

2.2 Un joli dégradé

Écrire un programme qui :

1. demande deux couleurs c_1 et c_2 à l'utilisateur, sous forme de triplets (rouge, vert, bleu) ;
2. colorie la fenêtre à l'aide de lignes obliques (haut-gauche vers bas-droit), en faisant varier les couleurs de chaque ligne, de manière à obtenir un dégradé de couleurs de c_1 vers c_2.

Indication : la méthode `setColor(float, float, float)` permet de changer la couleur de tracé en indiquant les composantes rouge, vert et bleu de la couleur choisie (chaque composante est comprise entre 0 et 1 inclus).

2.3 Une balle rebondissante

Écrire un programme qui déplace une balle dans la fenêtre en la faisant rebondir lorsqu'elle atteint les bords.

Indications :

- pour dessiner la balle, vous pourrez utiliser la méthode `fillCircle (int, int, int)` ;
- il est possible de ralentir l'exécution du programme en utilisant la méthode `msleep(long)`.

2.4 Tracé de fonction

On dispose d'une fonction f, par exemple :

```
static double f(double x)
{
    return Math.sin(x);
}
```

Écrire un programme permettant de tracer la fonction f dans une fenêtre délimitée par $X_{\min}$, $X_{\max}$ en abscisse, et $Y_{\min}$, $Y_{\max}$ en ordonnée.

Indications On veut tracer une fonction de la forme $y = f(x)$ dans l'intervalle $[X_{\min}, X_{\max}] \times [Y_{\min}, Y_{\max}]$ dans une fenêtre de taille $L \times H$, où $X_{\min}$, $X_{\max}$, $Y_{\min}$, $Y_{\max}$, L et H sont les données. La difficulté vient du fait que, pour dessiner dans la fenêtre, un autre repère que celui du graphique est utilisé. Il faut être capable de faire des conversion entre les coordonnées-fenêtre (pixels) et les coordonnées-graphique, sachant que :

(i) les coordonnées-fenêtre vont de $(0, 0)$ en haut à gauche, à $(L-1, H-1)$ en bas à droite ;

(ii) les coordonnées-graphique vont de $(X_{\min}, Y_{\min})$ en bas à gauche, à $(X_{\max}, Y_{\max})$ en haut à droite.

NB. Faire un dessin pour bien se représenter les choses.

L'idée de l'algorithme de tracé est alors de tracer un point de la courbe sur chacune des colonnes de pixels, en procédant de la manière suivante :

pour x_f **de** 0 **à** $L-1$ **faire**

1. trouver x_g (en fonction de x_f)
2. calculer $y_g = f(x_g)$
3. trouver y_f (en fonction de y_g)
4. tracer le pixel (x_f, y_f)

fpour

Pour les conversions de coordonnées, on pourra utiliser les formules suivantes. Soit (x_g, y_g) un point du graphique et (x_f, y_f) le pixel correspondant dans la fenêtre. On a la relation :

$$\begin{cases} x_g = X_{\min} + x_f \times (X_{\max} - X_{\min})/(L-1) \\ y_g = Y_{\max} - y_f \times (Y_{\max} - Y_{\min})/(H-1) \end{cases}$$

ou bien :

$$\begin{cases} x_f = (x_g - X_{\min}) \times (L-1)/(X_{\max} - X_{\min}) \\ y_f = (Y_{\max} - y_g) \times (H-1)/(Y_{\max} - Y_{\min}) \end{cases}$$