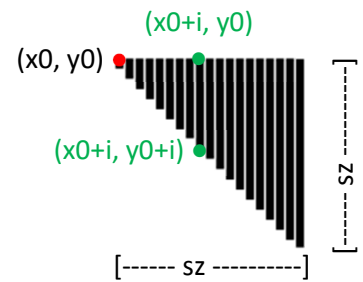


Sujet : dessin d'un triangle rectangle

Même si on possédait une méthode pour le dessin d'un triangle rempli (méthode *fillPolygone(...)*) nous entreprenons de faire ce dessin à l'aide des moyens du bord, c.-à-d. avec les méthodes prévues dans le programme. Effectivement, un triangle peut être composé d'une suite de lignes parallèles :



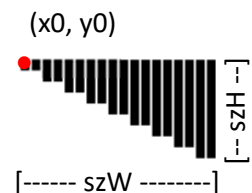
Soit **(x0,y0)** le point supérieur à gauche du triangle et **sz** la largeur et en même temps sa hauteur. Pour faire ce dessin, il faudra donc parcourir sa largeur et tracer en tout **sz** lignes.

```
for (int i=0;i<sz;i++) g.drawLine(x0+i, y0, x0+i, y0+i);
```

La dernière ligne (pour $i=sz-1$) a une hauteur de sz pixels, de $y0$ à $y0+(sz-1)$.

Or la formule devient donc un peu plus compliquée quand la hauteur est différente de la largeur. Il faut alors transmettre cette relation entre hauteur et largeur sur la relation entre la coordonnée x et la longueur de la ligne :

```
for (int i=0;i<szW;i++)  
    g.drawLine(x0+i, y0, x0+i, y0+i*(szH-1)/(szW-1));
```



La dernière ligne (pour $i=szW-1$) a une hauteur de szH pixels, de $y0$ à $y0+(szW-1)*(szH-1)/(szW-1)$

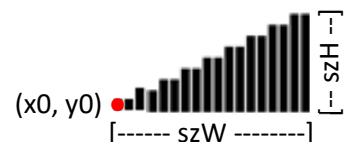
Notez qu'il est important de multiplier avant de diviser, pour ne pas avoir des problèmes à cause de la division entière.

(Pour vérifier, testez avec $i/(szW-1)*(szH-1)$ au lieu de $i*(szH-1)/(szW-1)$.)

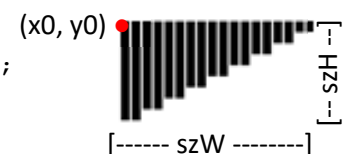
Pour compléter, voici le code pour des triangles orientés autrement :

Notez la position du point **(x0,y0)** qui se trouve toujours à l'origine de la ligne droite horizontale du triangle.

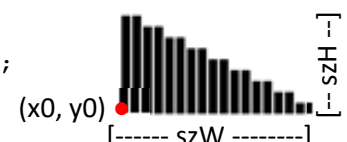
```
for (int i=0;i<szW;i++)  
    g.drawLine(x0+i, y0, x0+i, y0-i*(szH-1)/(szW-1));
```



```
for (int i=0;i<szW;i++)  
    g.drawLine(x0+i, y0, x0+i, y0+(szH-1)-i*(szH-1)/(szW-1));
```



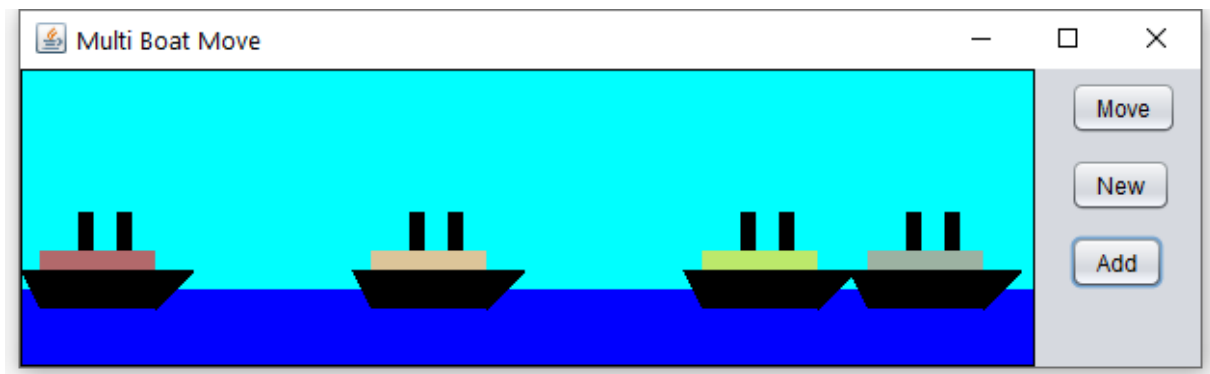
```
for (int i=0;i<szW;i++)  
    g.drawLine(x0+i, y0, x0+i, y0-(szH-1)+i*(szH-1)/(szW-1));
```



Pour aller un peu plus loin:

- Nous avons tracé des lignes verticales. On peut bien sûr dessiner les mêmes rectangles à l'aide de lignes horizontales. C'est alors le 3^e paramètre de *drawLine* qui définit son orientation.
- Un triangle qui n'est pas un triangle droit mais qui a une base parallèle à l'une des axes peut être divisé en deux triangles droits.
- Un losange est composé de 4 triangles rectangles.

Exercice : En partant de l'exercice MultiCarMove, et conforme à l'exemple type (sur le site), développez le projet ci-dessous :



- La hauteur totale du bateau est de 50 pixels, sa largeur totale de 90 pixels.
- Le triangle derrière a une largeur de 10, celui de devant une largeur de 20 pixels.
- Une cheminée a les dimensions 20x8 pixels. La mer a toujours une hauteur de 40 pixels et la distance du bateau un bord est de 30 pixels. Elle se trouve en bas du drawPanel, n'importe sa taille.
- La couleur est aléatoire. La vitesse est différente pour chaque bateau (mettez un attribut vitesse et additionnez cet attribut au lieu de '5' pour un « move »).

Personnalisez votre projet (drapeau ou hublots ou vapeur sortant des cheminées, etc.)