

Manuel d'utilisation de TextDraw

1. Présentation

Le programme **TextDraw** permet de créer des images aux formats **.SVG** et **.PPM** à partir de commandes écrites dans un fichier texte.

2. Pré-requis

Nous recommandons l'utilisation du logiciel **InkScape** pour profiter au maximum de l'image obtenue.

Voici un lien pour le télécharger : <https://inkscape.org/fr/telechargement/>

3. Utilisation

Écrivez vos commandes dans un fichier texte que vous donnerez en argument lors de l'exécution de **TextDraw**.

Vous pouvez exécuter **TextDraw** ainsi dans un terminal de commande :

```
java Main fichier_instruction.txt
```

Donnez ensuite le nom désiré pour votre image en écrivant la commande suivante dans votre fichier texte :

```
write nom_du_fichier.svg
```

Une fois le programme exécuté, vous retrouverez le résultat final dans le fichier "[nom_de_fichier].svg" ou "[nom_de_fichier].ppm" suivant l'extension choisie.

*Prenez en compte le fait que **TextDraw** dessine dans l'ordre des instructions.*

4. Les commandes

Toutes les commandes doivent se terminer par un point virgule.

4.1. La fonction DRAW

Cette fonction sert à dessiner des formes géométriques.
L'utilisation des courbes de Bézier permet d'obtenir diverses figures.

4.1.1. Les polygones

Pour dessiner un polygone, vous devez ajouter le jeton "**polygon**" après "**draw**" suivi des coordonnées des points que vous désirez encadrés par des parenthèses.

Vous pouvez obtenir un triangle, dont les sommets sont les points donnés, avec l'écriture suivante.

```
draw polygon {<x1,y1> <x2,y2> <x3,y3>};
```

4.1.2. Les ellipses

Pour dessiner une ellipse, vous devez ajouter le jeton "**ellipse**" après "**draw**" suivi des coordonnées du centre de votre ellipse, un rayon vertical et horizontal que vous désirez encadrer par des parenthèses.

L'écriture standard est la suivante :

```
draw ellipse {<x,y> (ray_vert) (ray_hori) };
```

Vous pouvez également dessiner un simple cercle, en remplaçant le jeton "**ellipse**" par le jeton "**circle**" et en donnant le centre ainsi qu'un seul rayon, avec :

```
draw circle {<x,y> (ray)};
```

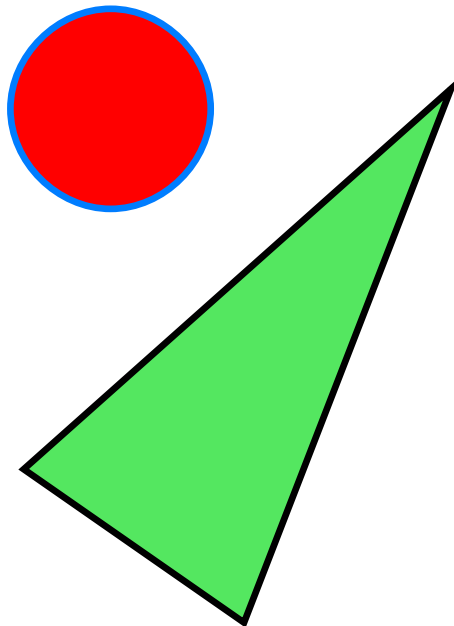
4.1.3. La couleur

Vous pouvez également colorier le contour et le fond de votre figure en complétant votre ligne de commande avec les champs "**color{R,G,B}**".

La première couleur spécifiée désigne l'intérieur et la seconde le contour de votre figure.

Exemple

```
1 draw circle {<25,10> (15) color{255,0,0} color{0,125,255}};  
2 draw polygon {<45,87> <12,64> <76,7> color{84,231,96} } ;  
3 write test.svg;
```



Les couleurs se donnent dans leurs repères **RGB** mais des couleurs, telles que le rouge, le vert ou le bleu, sont déjà enregistrées et définies comme des variables .

Listes des variables couleurs enregistrées

COULEUR	Variable 1	Variable 2	RGB
Rouge	\$rouge	\$red	{255, 0, 0}
Vert	\$vert	\$green	{0, 255, 0}
Bleu	\$bleu	\$blue	{0, 0, 255}
Jaune	\$jaune	\$yellow	{255, 255, 0}
Orange	\$orange	/	{237, 127, 16}
Rose	\$rose	\$pink	{253, 108, 158}
Violet	\$violet	\$purple	{102, 0, 153}
Marron	\$marron	\$brown	{91, 60, 17}
Noir	\$noir	\$black	{0, 0, 0}
Blanc	\$blanc	\$white	{255, 255, 255}
Gris	\$gris	\$grey	{96, 96, 96}

Voici un lien pour obtenir d'autres valeurs RGB :
<http://www.toutes-les-couleurs.com/code-couleur-rvb.php>

Ne pas préciser les champs couleurs mènera à l'utilisation des couleurs de bases qui sont le blanc pour l'intérieur et le noir pour le contour.

Vous pouvez enregistrer vos propres couleurs préférées en utilisant la méthode "def".

4.2. La fonction DEF

Cette fonction permet de définir une variable, qui portera un nom, vous permettant de la réutiliser plus tard dans votre fichier contenant les commandes textuelles.

Toutes les variables doivent commencer par le symbole '\$' devant leur nom afin de les reconnaître en tant que telles.

L'écriture d'une variable se fait ainsi :

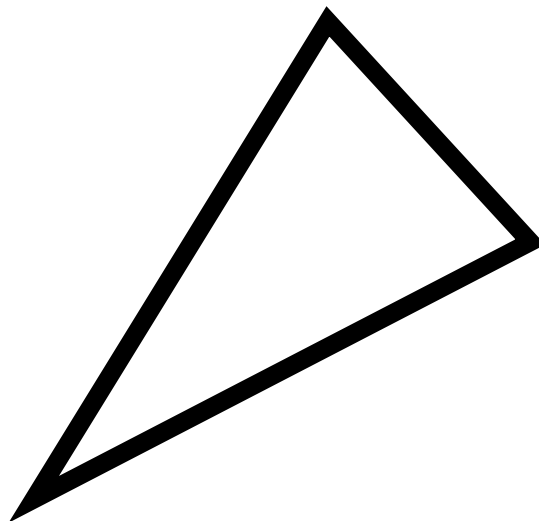
```
def $tri polygon {<15,48> <68,34> <72,22>};
```

Nous avons donc un triangle nommé "\$tri" .

Attention, le triangle "\$tri" n'est pas dessiné.

Pour cela, vous devez écrire :

```
draw $tri;
```



Vous pouvez enregistrer autant de variables différentes que vous voulez, que se soit des figures, des coordonnées ainsi que des valeurs.

Attention, réécrire une variable efface complètement les informations précédentes. Y compris lors des transformations.

4.3. Les transformations

TextDraw permet d'exécuter des transformations géométriques sur vos figures sauvegardées dans des variables.

Attention, effectuer une transformation sur une variable autre qu'une figure causera une erreur.

Une transformation modifie les informations de votre figure , prenez-en compte lors de votre dessin.

4.3.1. La translation

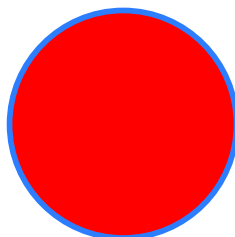
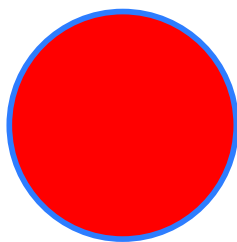
Pour effectuer une translation, donnez les coordonnées du point vers lequel le premier point de votre figure sera déplacé.

Son écriture se fait ainsi :

```
move $figure at <x,y> ;
```

Exemple

```
1 def $test circle {<25,10> (20) color{255,0,0} color{45,125,255}};  
2 draw $test;  
3 move $test at <45,67>;  
4 draw $test;  
5 write test.svg;
```



4.3.2. La symétrie

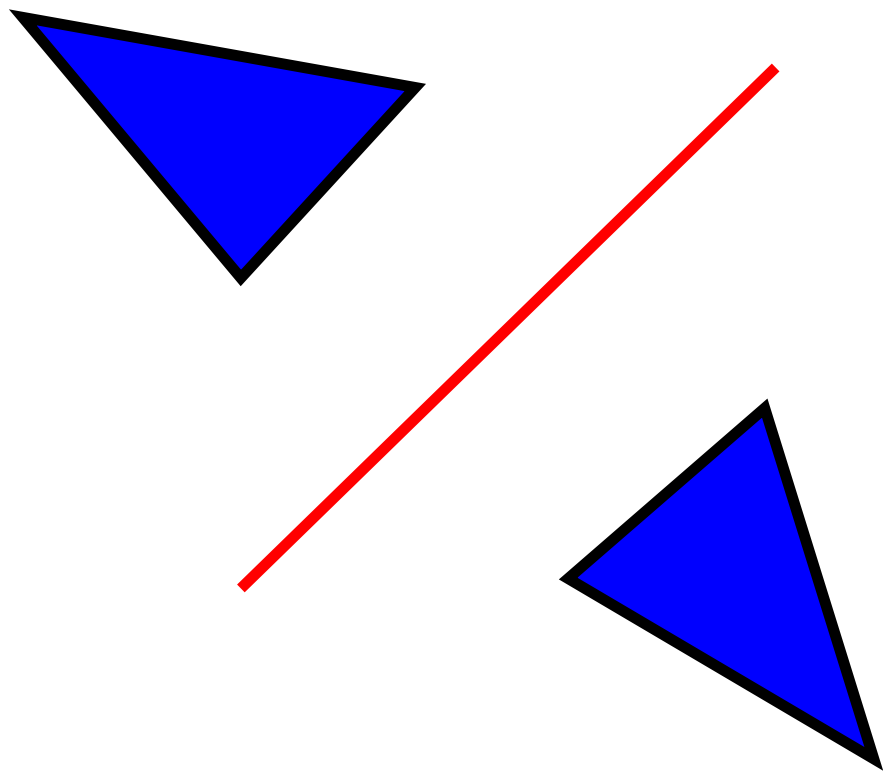
Pour effectuer une symétrie axiale, donnez les coordonnées des points de l'axe de symétrie vers lequel les points de votre figure seront déplacés.

Son écriture se fait ainsi :

```
reflect $figure from <x1, y1> <x2, y2> ;
```

Exemple

```
1  def $test polygon {<25,10> <45,36> <61,17> $bleu};  
2  draw $test;  
3  draw polygon {<45,67> <94,15> $white $red};  
4  reflect $test from <45,67> <94,15>;  
5  draw $test;  
6  write test.svg;
```



4.3.3. La rotation

Pour effectuer une rotation, donnez l'angle de la rotation, en degré désiré ainsi que les coordonnées du point duquel vous voulez exécuter votre rotation. La rotation se fait dans le sens des aiguilles d'une montre.

Son écriture se fait ainsi :

```
rotate $figure of (d°) around <x,y> ;
```

Exemple

```
1 def $test polygon {<61,10> <45,98> <51,29> $orange};  
2 draw $test;  
3 rotate $test of (45) around <45,67>;  
4 draw $test;  
5 write test.svg;
```



Vous pouvez également effectuer une rotation de la figure sur elle-même à partir de son barycentre en ne donnant que l'angle.

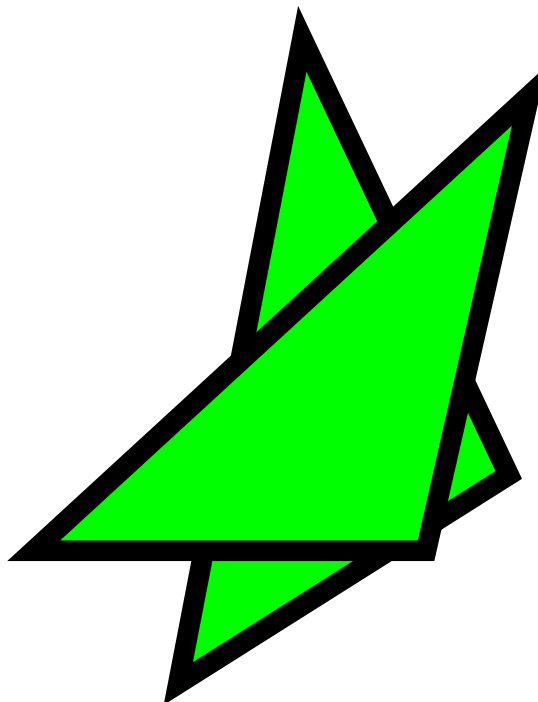
Son écriture se fait ainsi :

```
spin $figure of (d°) ;
```

Vous ne pouvez pas appliquer la méthode "spin" à une figure du type "ellipse" ou "circle".

Exemple

```
1 def $test polygon {<61,87> <45,98> <51,64> $vert};  
2 draw $test;  
3 spin $test of (35);  
4 draw $test;  
5 write test.svg;
```



4.3.4. L'homothétie

Pour effectuer une homothétie, donnez le facteur k pour agrandir ou diminuer votre figure et un point d'homothétie.

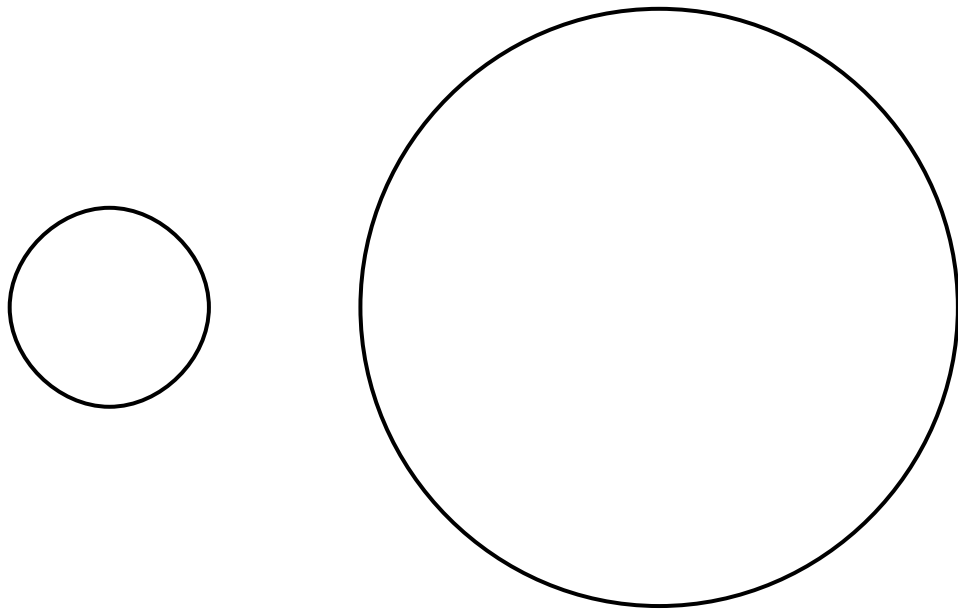
Son écriture se fait ainsi :

```
extend $figure of k from <x,y> ;
```

Vous pouvez effectuer une diminution en donnant à la variable k une valeur entre 0 et 1.

Exemple

```
1 def $test circle {<61,87> (25)};  
2 draw $test;  
3 extend $test of (3) from <15,87>;  
4 draw $test;  
5 write test.svg;
```



Vous pouvez également effectuer un agrandissement de la figure à partir de son barycentre avec l'écriture suivante.

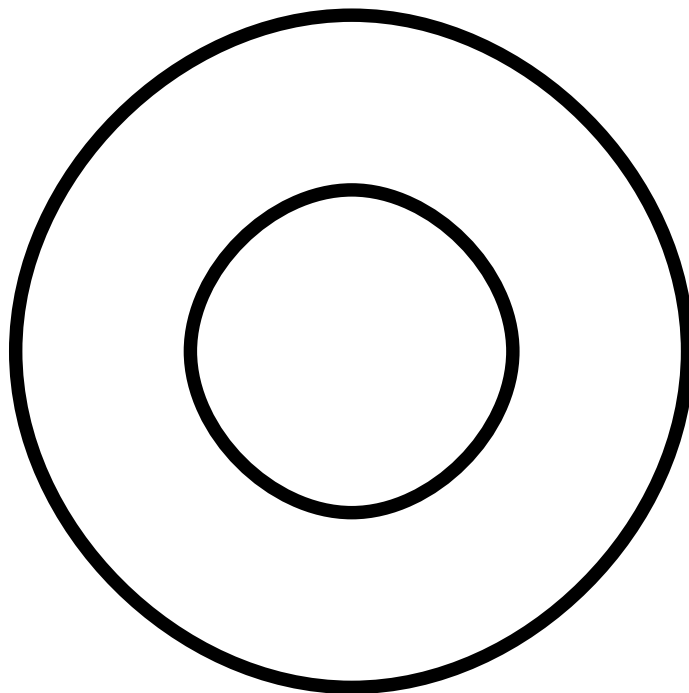
Son écriture se fait ainsi :

```
grow $figure of (d°) ;
```

Vous ne pouvez pas appliquer la méthode "spin" à une figure du type "ellipse" ou "circle".

Exemple

```
1  def $test circle {<61,87> (25)};  
2  draw $test;  
3  grow $test of (0.5);  
4  draw $test;  
5  write test.svg;
```



4.3.5. Changement de couleurs

Vous pouvez changer la couleur de contour et de remplissage d'une figure.

Son écriture se fait ainsi :

```
recolor $figure in color{255,0,0} and $blue ;
```

Vous pouvez effectuer une diminution en donnant à la variable k une valeur entre 0 et 1.

5. Exemples

Vous pouvez retrouver des exemples plus complexes dans le fichier Exemple.

6. Informations Complémentaires

TextDraw est un programme d'éditeur d'image textuel écrit en JAVA.

Il a été réalisé en 4 mois par notre groupe de 3 étudiants, dans le cadre de l'UE PI4 pour le semestre 4 de la Licence Informatique de l'Université Paris 7 Denis Diderot.

Auteurs : Pierrick JAQUETTE, Quentin PERRACHON & Samuel ELBEZ