

# Tortue graphique

proposé par Arnaud Sangnier ([sangnier@liafa.univ-paris-diderot.fr](mailto:sangnier@liafa.univ-paris-diderot.fr))

Projet de Programmation 2012/2013 - 51IF2IK3 - 20 septembre 2012

L'attribution des sujets se fait lors de la première séance. Elle apparaît ensuite sur `didel` et n'est plus modifiable.

## 1 La tortue graphique

Une "tortue graphique" désigne un terme générique pour caractériser un langage de programmation graphique utilisant un curseur se déplaçant sur un plan. La tortue graphique était un élément essentiel du langage de programmation Logo.

Vous pouvez aller jeter un coup d'oeil sur les pages suivantes :

[http://en.wikipedia.org/wiki/Turtle\\_graphics](http://en.wikipedia.org/wiki/Turtle_graphics)  
[http://en.wikipedia.org/wiki/Logo\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/Logo_(programming_language))

La tortue, qui caractérise le curseur se déplaçant dans le plan pour effectuer les dessins a trois attributs principaux :

1. sa position (typiquement, abscisse et ordonnée dans le plan)
2. son orientation (typiquement un angle en degrés ou radian)
3. son "crayon" (qui contient les informations sur la couleur et éventuellement l'épaisseur du trait)

La tortue se déplace grâce à des commandes par rapport à sa position courante et selon son orientation. Par exemple, on peut lui dire "déplace à toi en avant de 10 espaces" ou encore "tourne à gauche de 45 degrés". Pendant son déplacement, le programmeur pourra également activer ou de désactiver le crayon et choisir les attributs du crayon, ainsi si le crayon est activé, la tortue laissera une trace dans la fenêtre correspondant à son déplacement. Pour réaliser un dessin, le programmeur doit par conséquent imaginer les mouvements de la tortue le long des traits du dessin.

## 2 Description du projet

### 2.1 But

Le but de ce projet est de développer une application permettant de recevoir des instructions similaires à celle de la tortue graphique et de dessiner dans une fenêtre le graphique correspondant. Dans un premier temps, les instructions pourront être tapées dans le terminal mais idéalement il faudrait que votre programme puisse à la fin prendre en argument un fichier contenant un programme de la tortue graphique et ensuite l'afficher.

Nous fournissons plus tard quelques exemples d'instructions que votre langage de tortue graphique devra accepter, libre à vous de l'étendre au gré de votre imagination en vous inspirant des instructions Logo de la tortue graphique. Dans le cahier des charges à rendre au bout de la première semaine, vous devrez fournir une liste des instructions que vous souhaitez mettre dans votre langage en ayant pris soin d'en ajouter quelques-unes à la liste fournie ci-dessous et en expliquant quelle sera leur rôle.

Il vous faudra également écrire des programmes exemple traçant des figures particulières et utilisant votre ensemble d'instructions. L'idéal serait de faire des dessins un peu élaborés.

## 2.2 Un langage de base

Voilà quelques exemples d'instructions que vous pouvez reprendre ou dont vous pouvez vous inspirer pour définir votre ensemble d'instructions :

- BACKWARD 10 la tortue va en avant de 10 unités
- FORWARD 5 la tortue va en arrière de 5 unités
- LEFT 40 la tortue se tourne de 40 degrés vers la gauche
- RIGHT 30 la tortue se tourne de 40 degrés vers la droite
- PENUP le crayon de la tortue est désactivé
- PENDOWN le crayon de la tortue est activé
- SETCOLOR ROUGE la couleur du crayon est rouge
- SETCOLOR WHITE la couleur du crayon est rouge

## 2.3 Quelques conseils

Comme vous l'aurez compris, il est important dans un premier temps de définir de façon extrêmement précise votre ensemble d'instructions, il ne faut pas laisser de place à l'ambiguïté sur les instructions car sinon vous vous retrouverez dans l'impossibilité de parcourir le fichier contenant les programmes.

Il vous faudra également réfléchir à des règles de conception, par exemple quelle est la taille de la fenêtre, est-elle prédéfinie, l'utilisateur peut-il la changer ? Ou comment représenter la tortue dans la fenêtre, par une croix, un triangle, un rond ? Pensez à des problèmes qui peuvent se poser si la tortue sort de la fenêtre, comment gérer une telle situation ? Également si l'on souhaite voir le curseur se déplaçant ou seulement le dessin se traçant instantanément (les deux solutions devraient être possibles, mais du coup réfléchir à quelle "vitesse" le curseur doit se déplacer ?). Pensez également qu'il vous faudra renvoyer des messages d'erreur si l'utilisateur utilise de façon incorrecte une de vos instructions.

En bref, dans votre cahier des charges, il vous faudra penser au maximum à comment fonctionnera votre projet.

## 2.4 Extensions possibles

Idéalement, vous pourrez étendre votre langage de programmation avec des instructions plus complexes, comme par exemple l'introduction de variables entières ou l'utilisation de la commande REPEAT (voir la page *Wikipedia* du langage Logo). Vous pourrez aussi raffiner les différentes options offertes à l'utilisateur, pourquoi pas également un contrôle du curseur à la souris ou au clavier... bref à vous de choisir ce qui vous semble le plus judicieux.

# 3 Documents à rendre

Tout au long du semestre vous aurez à rendre les documents suivants :

- spécification fonctionnelle / cahier des charges — semaine du 24 septembre 2012,
- spécification interne — semaine du 1er octobre,
- mode d'emploi — semaine du 10 décembre,
- javadoc — au fur et à mesure de la production de code.

Ces documents sont décrits dans la présentation générale de l'enseignement distribuée en début de semestre et disponible sur `didel` :

`http://didel.script.univ-paris-diderot.fr/`

Après vous être connectés une première fois à `http://usvn.script.univ-paris-diderot.fr/login` avec vos identifiant et mot de passe ENT et avoir été rattaché à votre projet, les diverses versions de votre code seront à gérer *via* le serveur `svn` fourni par le script :

|                                       |   |
|---------------------------------------|---|
| Initialisation d'une copie locale     | <code>svn checkout --username <i>login</i> http://<i>url</i></code> |
| Mise à jour de la copie locale        | <code>svn update</code>   |
| Intégrer un fichier à subversion      | <code>svn add <i>nom_du_fichier</i></code>                          |
| Publication des modifications locales | <code>svn commit -m "<i>description des modifications</i>"</code>   |

|              |   |
|--------------|---|
| <i>login</i> | votre login ENT   |
| <i>url</i>   | <code>http://usvn.script.univ-paris-diderot.fr/svn/<i>groupe</i>/trunk</code> |

Les projets seront à rendre pour le lundi 17 décembre 2012 à 20h au plus tard. Les soutenances auront lieu pendant la session d'examen. Le détail des dates seront donnés sur le site `didel` de la matière.