

GéoTortue

irem de Paris-Nord

mini-guide par Alain Brobecker, <http://abrobecker.free.fr>



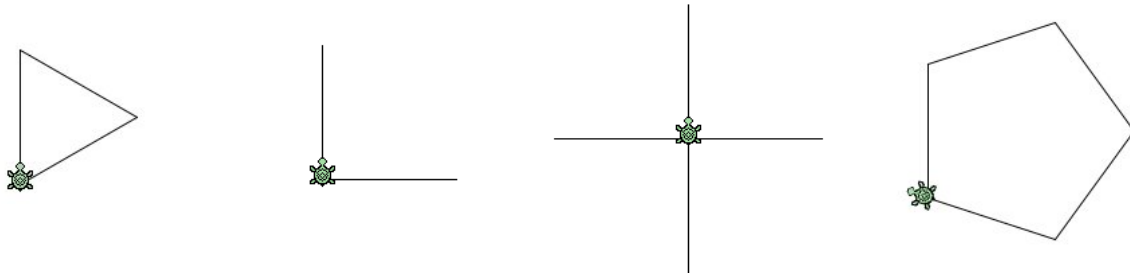
1) Prise en main, commandes de base

Lancer GéoTortue. Dans le "Bac à sable", rentrer les instructions situées dans le cadre ci-dessous les unes après les autres. Observer notamment la position de la tortue au début du programme et à la fin du programme.

```
> av 100
> td 90
> av 100
> td 90
> av 100
> td 90
> av 100
```

vg	vide graphique (réinitialisation)
td A	tourner à droite de A degrés
tg A	pour tourner à gauche de A degrés
av N	avancer de N unités (pixels?)
re N	reculer de N unités (pixels?)
lc/bc	lever/baisser le crayon
ct/mt	cache/montre la tortue

Muni de ces connaissances, réalise les quatre figures suivantes:



2) Instruction de répétition

L'instruction **rep N [...]** permet de répéter N fois le bloc contenu entre crochets. Si on met plusieurs instructions il faut les séparer par un **point-virgule** (ou un retour à la ligne). Observe ce que fait le programme ci-contre.

```
> vg
> rep 3 [ av 100; td 120 ]
```

Utilise l'instruction **rep** pour tracer un carré, puis pour tracer un pentagone régulier et enfin pour tracer un pentagone étoilé.

3) Procédure

Passer en mode "Pupitre". Ecrire dans la "Fenêtre de commande" un programme qui réalise la figure ci-contre (le carré fait 50 unités de côté, et est commencé à 50 unités du point de départ).



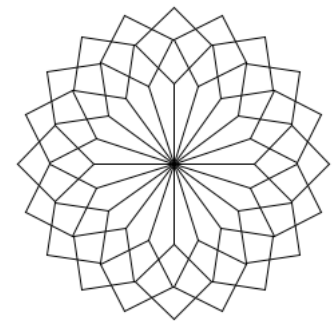
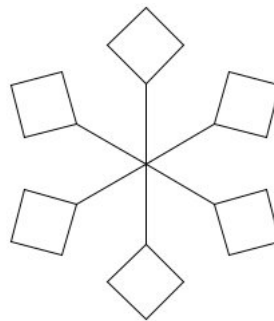
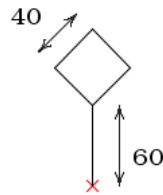
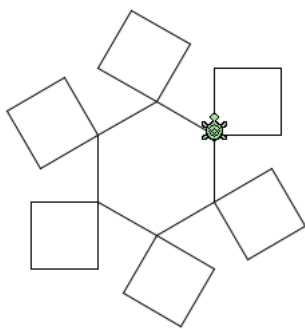
Appuie sur l'icône "ampoule". Nomme ta procédure **Drapeau**. Dans la fenêtre "Procédures" tu dois obtenir le code suivant. Tu disposes maintenant entre les instructions **pour** et **fin** d'un bout de programme qui peut être appelé quand tu veux, juste en donnant son nom dans la fenêtre de commande, ou dans une autre procédure.

```
1 > pour Drapeau
2 > av 100
...
11 > re 50
12 > bc
13 > fin
```

Dans la fenêtre de commande, entrer maintenant **vg ; rep 12 [Drapeau; td 30]** et observer le résultat.

Attention: ne pas confondre **pour** avec l'instruction de boucle **for(i;1;100)** des calculatrices TI...

En écrivant des procédures réalise les trois figures suivantes (les deux dernières seront réalisées avec une même procédure dessinant le drapeau décrit juste avant les figures).



4) Variables / affectation

GéoTortue permet d'utiliser des variables, autrement dit des cases mémoires qui contiennent une valeur pouvant être changée. On *affecte* une valeur à l'aide de la commande

NomDeLaVariable := valeur.

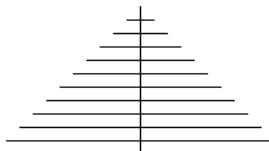
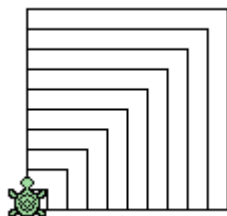
La partie de droite peut être un calcul, et peut contenir des noms de variables, la valeur de ces variables seront cherchées en mémoire lorsque l'instruction devra être exécutée.

Ecris la procédure ci-contre, exécute la et observe le résultat. Il te faudra peut-être un peu de réflexion pour vérifier que le résultat observé correspond bien à la procédure.

```

1> pour spirale
2> longueur:=10
3> rep 20 [
4> av longueur
5> td 90
6> longueur:=longueur+10
7> ]
8> fin
    
```

Ecris des procédures pour réaliser les deux figures ci-dessous:



```

9> pour Carre cote
10> rep 4 [ av cote; td 90 ]
11> fin
    
```

Une procédure peut aussi être appelée avec un ou plusieurs paramètres, il suffit de les nommer dans la définition de la procédure. Lors de l'appel de la procédure il suffit alors de leur attribuer une valeur (qui peut être contenue dans une autre variable ou dans une expression). Ci-dessus la procédure Carre crée un carré avec une taille donnée, appelle la par exemple avec les instructions **vg ; Carre 50**.

On peut utiliser cette procédure Carre dans la procédure qui traçait les 10 carrés imbriqués. Essaie de modifier ton ancienne procédure pour qu'elle utilise maintenant la procédure Carre.

Enfin, écris une procédure qui trace un polygone régulier à N côtés, avec $N > 2$, et dont les côtés mesurent 100 unités. Modifie ensuite cette procédure pour que l'utilisateur donne aussi la taille des côtés. Tu peux, si tu le veux, réfléchir à comment faire un polygone étoilé (il faudra utiliser la notion de pgcd)...

Attention: Remarque que sur les calculatrices TI l'affectation se fait "dans l'autre sens", c'est à dire en mettant la variable après l'expression, valeur \rightarrow NomDeLaVariable.

5) Instructions conditionnelles, tirage aléatoire

Les instructions conditionnelles sont de la forme:

si (condition) alors [instructions_si_vrai] sinon [instructions_si_faux]

Le premier bloc d'instruction est exécuté si la condition est vraie. Le deuxième bloc est optionnel, il est exécuté si la condition est fautive. Pour donner un exemple un peu intéressant et pas trop compliqué, nous allons aussi utiliser l'instruction **alea(n)** qui renvoie un nombre entier au hasard compris entre 1 et n.

Ainsi notre exemple choisit aléatoirement le nombre 1 ou le nombre 2. Si le nombre 1 a été choisi, la tortue avance, si c'est le nombre 2 la tortue tourne un peu à droite.

```
1> pour DeplacementAleatoire
2> rep 100 [
3> x:=alea(2)
4> si ( x==1 ) alors [ av 10 ] sinon [ td 10 ]
5> ]
6> fin
```

Ecris une procédure `DeplacementAleatoire2` qui répète 100 fois les actions suivantes: tirer un nombre aléatoire entre 1 et 3. Si ce nombre est 1, faire avancer la tortue. Si c'est 2, faire tourner la tortue à droite de 90°. Si c'est trois, faire tourner la tortue à gauche de 90°.

Attention: Tu remarqueras que l'on doit mettre `==` pour tester une égalité. Pour tester une inégalité on utilise `>`, `<`, `>=` ou `<=`. Enfin on utilise `!=` pour tester une différence.

Remarque: Si l'on veut effectuer plusieurs tests en une seule fois, et sans imbriquer les instructions **si**, on peut utiliser les opérateurs logiques `||` (OU inclusif) et `&&` (ET). Par exemple avec l'instruction **si ((x>0) && (x<200)) [...]** le bloc d'instructions sera exécuté lorsque la variable x sera plus grande que 0 et qu'elle sera aussi plus petite que 200.

6) La répétition conditionnelle "tant que"

La structure de répétition **rep N [...]** suppose que l'on connaisse le nombre de fois ou le bloc doit être répété. Dans certains cas, on veut répéter le bloc un nombre de fois indéterminé, et cesser la répétition lorsqu'une condition est vérifiée. On utilise alors l'instruction:

tant_que (condition) [...]

La procédure ci-contre commence par initialiser à 0 la variable **y**, qui sera un indicateur de position verticale de la tortue. Ensuite il effectue le bloc d'instruction tant que cette variable ne dépasse pas 200. Dans ce bloc d'instruction, on choisit aléatoirement entre un déplacement vertical, on actualise alors y, ou un déplacement horizontal.

```
1> pour MonteDe200
2> y:=0
3> tant_que (y<200) [
4> a:=alea(2)
5> si (a==1) alors [ av 10; y:=y+10 ]
6> si (a==2) alors [ td 90; av 10; tg 90 ]
7> ]
8> fin
```

Ecrire une procédure `TourComplet` qui choisit aléatoirement entre avancer de 10 unités ou tourner à droite de 5°, et qui s'arrête quand la tortue a tourné de 360° au total.

7) D'autres notions, d'autres géométries...

Il existe encore beaucoup de notions à découvrir mais les instructions que tu connais maintenant permettent déjà de faire beaucoup de programmes. Une des particularités de GéoTortue est de permettre de travailler dans des géométries différentes: en géométrie sphérique, sur un tore (appelé "quotients rectangulaires du plan"), et aussi la géométrie 3D. Pour cette dernière tu auras besoin des commandes `pvh` (pivote vers le haut), `pvb` (pivote vers le bas), `pvd` (pivote sur la droite) et `pvg` (pivote sur la gauche).

Bonne exploration...

SOLUTIONS

Triangle équilatéral:

```
rep 3 [ av 100; td 120 ]
```

La figure en forme de L:

```
av 100; lc; re 100; td 90; av 100;
bc ; re 100; tg 90
```

La croix:

```
lc; av 100; bc; re 200; lc; av 100; td 90;
av 100; bc ; re 200; lc; av 100; tg 90
```

Pentagone régulier:

```
rep 5 [ av 100; td 72 ]
```

Pentagone étoilé:

```
rep 5 [ av 100; td 144 ]
```

Procédure Drapeau:

```
pour Drapeau
av 100
rep 3 [ td 90; av 50 ]
td 90
lc
re 50
bc
fin
```

Dessiner l'hexagone avec des carrés:

```
vg
rep 6 [ Drapeau; lc; av 50; tg 60; bc ]
lc ; av 50; bc
```

Le Drapeau avec un losange:

```
pour Drapeau2
av 60
td 45
rep 3 [ av 40; tg 90 ]
av 40
tg 135
lc
re 60
bc
fin
```

L'étoile avec Drapeau2:

```
vg; rep 6 [ Drapeau2; td 60 ]
```

La rosace:

```
vg; rep 30 [ Drapeau2; td 12 ]
```

Les 10 carrés imbriqués:

```
pour CarresGrandissants
cote:=10
rep 10 [
rep 4 [ av cote; td 90 ]
cote:=cote+10
]
fin
```

Le sapin:

```
pour Sapin
taille:=100
rep 10 [
av 10
td 90
av taille
lc
re taille
bc
re taille
lc
av taille
bc
tg 90
taille:=taille-10
]
av 10
fin
```

Les 10 carrés imbriqués, version utilisant la routine Carre:

```
pour Carre cote
rep 4 [ av cote; td 90 ]
fin

pour CarresGrandissants2
x:=10
rep 10 [ Carre x; x:=x+10 ]
fin
```

Procédure tracant un polygone régulier à N côtés:

```
pour PolygoneRegulier N C
Angle:=180-((N*180-360)/N)
rep N [ av C ; td Angle ]
fin
```

Procédure DeplacementAleatoire2:

```
pour DeplacementAleatoire2
rep 100 [
x :=alea(3)
si ( x==1 ) alors [ av 10 ]
si ( x==2 ) alors [ td 45 ]
si ( x==3 ) alors [ tg 45 ]
]
fin
```

Procédure TourComplet:

```
pour TourComplet
a:=0
tant_que (a<360) [
r:=alea(2)
si (r==1) alors [ td 5; a:=a+5 ]
si (r==2) alors [ av 10 ]
]
fin
```