❉ Chapitre 26 ❉

Utilisation possible du logiciel

I. La tortue

Avec Python, il est possible de faire des dessins. Pour cela, on utilise le mode tortue.
Dans EduPython, si vous souhaitez réaliser une nouvelle figure à l'aide de la tortue, cliquer sur le bouton **Nouveau** puis **Fichier** et choisissez le mode **Tortue**.
Un nouveau programme comportant déjà quelques lignes de code est généré :

```
# Cree par ..., le ... avec EduPython
from lycee import *
import turtle as tortue

tortue.mainloop()
```

Exemple 1:

À vous de programmer en recopiant le programme ci-dessous :

```
# Cree par Bastien, le 05/03/2017 avec EduPython
from lycee import *
import turtle as tortue

i=0
while i<36:
        tortue.forward(400)
        tortue.left(170)
        i=i+1

tortue.mainloop()
```

Quelques instructions utiles :

- `tortue.forward(50)` et `tortue.back(50)` font respectivement avancer et reculer la tortue dans la direction où elle regarde de 50 pixels (l'argument pouvant être entier ou non).

- `tortue.left(40)` et `tortue.right(40)` font respectivement tourner la tortue vers la gauche ou la droite de 40 degrés.
  Par défaut, le crayon est orienté vers l'est : l'angle est de 0 °(ou 360 °).
  Aucun tracé n'est effectué, juste une rotation de la tête.

- `tortue.circle(rayon)` ou `tortue.circle(rayon,angle)` :
  — Si *rayon* > 0 : Trace un cercle de rayon *rayon* à partir de la position de la tortue et en tournant dans le sens contraire des aiguilles d'une montre.
  — Si *rayon* < 0 : Trace un cercle de rayon (-*rayon*) à partir de la position de la tortue et en tournant dans le sens des aiguilles d'une montre.
  — Si *angle* est précisé, trace un arc de cercle avec une ouverture de *angle* (en degré).

- `tortue.showturtle()` `tortue.hideturtle()` a pour effet de respectivement cacher ou montrer la tortue à l'écran.
  Pour des questions d'esthétique, on peut par exemple vouloir cacher la tortue en fin de tracé.

- `tortue.speed(5)` permet de régler la vitesse de la tortue à 5. La vitesse est un nombre entier entre 1 et 10, 1 étant la vitesse la plus lente et 10 la plus rapide.

- `tortue.up()` et `tortue.down()` ont pour effet de respectivement lever et baisser le crayon.

- `tortue.pencolor('blue')` ou `tortue.pencolor(0.45,0.36,0.28)` permettent de définir la couleur du crayon.
  — On peut entrer un texte entre guillemets parmi (entre autres) : 'aqua', 'beige', 'black', 'blue', 'brown', 'chocolate', 'fuchsia', 'gold', 'gray', 'green', 'indigo', 'khaki', 'maroon', 'orange', 'red', 'white', ...
  — On peut aussi définir sa propre couleur en paramétrant les composantes rouge, vert et bleu de la couleur (chaque composante étant un nombre entre 0 et 1).
- `tortue.bgcolor()` permet de colorier le fond de l'image.
-
  ```
  torute.begin_fill()
  tortue.color('red','blue')
  ...
  tortue.end_fill()
  ```
  permettent de colorier le contour d'une figure avec la couleur rouge et de colorier l'intérieur avec la couleur bleue.
- `tortue.reset()` efface l'écran et repositionne la tortue dans sa position initiale.
- `tortue.clear()` efface l'écran mais la position du crayon reste inchangée.
- `tortue.write('chaussette')` affiche le texte *chaussette* à l'emplacement de la tortue. Celle-ci ne se déplace pas lors de l'affichage.

Exercice 1 :

Dessiner les différentes formes géométriques suivantes :

1. un carré de coté 100 pixels ;
2. un triangle équilatéral de côté 100 pixels de couleur bleu ;
3. un pentagone régulier de coté 60 pixels de couleur verte ;
4. un hexagone régulier de côté 50 pixels.

Exercice 2 :

Recopier l'algorithme ci-dessous sur Python :

```python
#Cree par Bastien, le 05/03/2017 avec EduPython
from lycee import *
import turtle as tortue

tortue.begin_fill()
tortue.color('red','blue')
for i in range(0,6):
        tortue.forward(60)
        tortue.left(60)
tortue.end_fill()

tortue.mainloop()
```

1. Que permet de dessiner cet algorithme ?
2. Faire un dessin de votre choix avec une boucle Pour.

Exercice 3 :

Recopier l'algorithme ci-dessous sur Python :

```python
#Cree par Bastien, le 05/03/2017 avec EduPython
from lycee import *
import turtle as tortue

for i in range(0,4):
        tortue.forward(200)
        tortue.right(90)
tortue.left(60)
for i in range(0,3):
        tortue.forward(200)
        tortue.right(120)
tortue.up()
tortue.right(150)
tortue.forward(200)
tortue.left(90)
tortue.forward(90)
tortue.down()
tortue.begin_fill()
tortue.color('brown','black')
tortue.left(90)
tortue.forward(40)
tortue.right(90)
tortue.forward(30)
tortue.right(90)
tortue.forward(40)
tortue.end_fill()

tortue.mainloop()
```

1. Que permet de dessiner cet algorithme ?
2. Faire un dessin de votre choix.

II. La dichotomie

Quand on ne peut pas résoudre une équation par les méthodes usuelles, on peut tout de même trouver une valeur approchée de la solution éventuelle en utilisant l'algorithme de dichotomie. Le mot dichotomie vient du grec *dicho* qui veut dire « en deux » et *tomê* qui veut dire « couper ».

Soit f une fonction monotone sur un intervalle $[a\ ;\ b]$. Si $f(x)$ change de signe sur l'intervalle $[a\ ;\ b]$, alors l'équation $f(x) = 0$ admet une unique solution c dans l'intervalle $[a\ ;\ b]$: on a donc un encadrement d'amplitude $b-a$ de c.

En calculant $m = \dfrac{a+b}{2}$ puis en évaluant le signe de $f(m)$, on peut dire si c est dans l'intervalle $[a\ ;\ m]$ ou dans l'intervalle $[m\ ;\ b]$: on obtient alors un encadrement de c d'amplitude deux fois plus petite.

En itérant ce principe, on obtient un encadrement de plus en plus fin de c, en divisant l'amplitude de l'encadrement par deux à chaque étape.

L'algorithme de dichotomie pour un nombre d'itération donné :

Variables :	a, b, m des réels ; N un entier
Entrée :	Saisir a, b, N
Traitement :	**Pour** i allant de 1 à N
	Affecter à m la valeur $\dfrac{a+b}{2}$
	Si $f(a) \times f(m) \leqslant 0$
	alors Affecter à b la valeur de m
	Sinon
	Affecter à a la valeur de m
	Fin Si
	Fin Pour
Sortie :	Afficher a, b

Remarque :

Pour tester si $f(a)$ et $f(m)$ sont ou non de même signe, on évalue le signe du produit $f(a) \times f(m)$.

Exercice 1 :

On considère la fonction f définie sur $\mathbb{R}$ par $f(x) = x^3 + 3x - 5$.

1. Programmer l'algorithme de dichotomie pour l'équation $f(x) = 0$ sur l'intervalle $[1\ ;\ 2]$.
   " $**$ " permet d'écrire en exposant.
2. Déterminer un encadrement de la solution c_1 de l'équation à l'issue de six itérations de l'algorithme.
3. Utiliser l'algorithme de dichotomie pour déterminer un encadrement de la solution c_2 de l'équation $x^4 - 2x^3 - 2x - 7 = 0$ dans l'intervalle $[2\ ;\ 3]$ à l'issue de huit itérations de l'algorithme.

Exercice 2 :

1. Modifier l'algorithme de dichotomie afin d'obtenir un encadrement de la solution inférieure ou égale à un réel p, donné à l'avance.
2. Modifier le programme réalisé dans l'exercice 1 avec les entrées a, b et p.
   Le tester avec $p = 0{,}001$.