

## Nombres premiers et tortue Logo (Denise Vella-Chemla, mars 2024)

*Avec toute ma reconnaissance et mon admiration, pour Cynthia Solomon, Seymour Papert, et Wally Feurzeig, qui ont inventé le langage Logo.*

### Le langage Logo

Après avoir relu un article au sujet de la courbe du dragon, on écrit le petit programme suivant pour la tortue Logo (module turtle en python), tout simple, qu'on peut résumer ainsi : "sur un nombre premier, tourne à droite ; sur un nombre composé, tourne à gauche ; dans les deux cas, avance".

```
import turtle
from turtle import *

def premier(atester):
    if atester in [0, 1]: return False
    if atester in [2, 3, 5, 7]: return True
    k = 2
    while True:
        if k * k > atester: return True
        else:
            if atester % k == 0: return False
            else: k = k + 1

setup()
speed(0)
setheading(90)
up()
turtle.setposition(0,0)
left(45)
down()
for n in range(2,20):
    if premier(n):
        right(90)
    else:
        left(90)
    forward(30)
down()exitonclick()
```

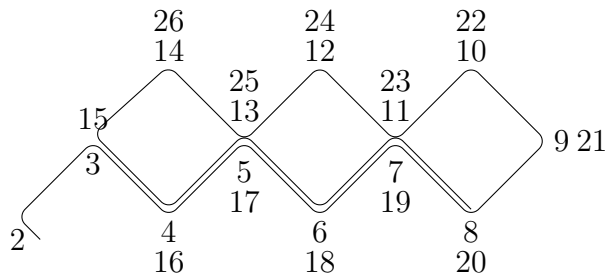
On lance le programme et l'exécution a ce résultat pour  $n$  allant de 2 à 100.



D'abord, on est très étonnée de voir la tortue se confiner ainsi dans une ligne de carrés<sup>1</sup>. Le résultat de l'exécution de ce programme jusqu'à 20 est :



Sur le graphique ci-dessous, notons les nombres déclenchant les mouvements de la tortue (jusqu'à 26) dans les angles pour s'imprégner de son cheminement.



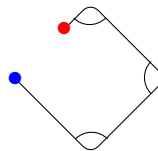
### Explication du confinement à la bande délimitée par les droites ( $H$ ) et ( $B$ )

La tortue reste confinée dans une bande délimitée supérieurement par une droite qu'on appellera ( $H$ ), inférieurement par une droite qu'on appellera ( $B$ ) et il y a au milieu de la bande une droite médiane qu'on appellera ( $M$ ). On va expliciter ci-après les mouvements de la tortue par des petits triplets de trois mouvements consécutifs, correspondant à 3 nombres successifs dont le premier est pair.

Pour expliciter les différentes sortes de trois mouvement successifs que peut faire la tortue, lorsqu'elle "gère" trois nombres successifs dont le premier est pair ( $> 2$ ), on distinguera les quatre positions initiales possibles de la tortue sur le carré sur lequel elle est couramment positionnée ; on notera ces 4 positions par les lettres  $g, h, d, b$  pour gauche, haut, droite, bas.

On se convainc que les 6 mouvements possibles de la tortue sont les suivants ; on colorie la position de départ de la tortue en bleu et sa position d'arrivée en rouge :

- 1) la tortue va "manger" un pair (tous les pairs rencontrés étant  $> 2$  ils sont composés), un nombre impair composé, un pair (composé, on ne l'écrira plus). Elle était initialement positionnée sur le sommet  $g$  du carré courant.

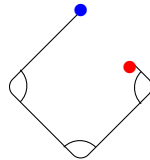


<sup>1</sup>Pour avoir longtemps enseigné le langage Scratch à des élèves d'école élémentaire, nombre d'entre eux pensent qu'un carré sans côté horizontal est un losange (ce qu'il est), mais n'est pas un carré. Cette "erreur" est due au fait que la plupart des enseignants, et j'en suis, dessinent un carré avec un côté horizontal, car il est naturellement plus aisé de le dessiner ainsi.

“Globalement” (i.e. par les 3 mouvements ci-dessus), les coordonnées de la tortue varient ainsi (si on considère qu’on est dans un plan cartésien) :

$$\begin{pmatrix} x \\ y \end{pmatrix} \longrightarrow \begin{pmatrix} x + 1 \\ y + 1 \end{pmatrix}.$$

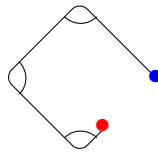
- 2) la tortue va “manger” un pair, un nombre impair composé, un pair. Elle était initialement positionnée sur le sommet  $h$  du carré courant.



*Changement de coordonnées :*

$$\begin{pmatrix} x \\ y \end{pmatrix} \longrightarrow \begin{pmatrix} x + 1 \\ y - 1 \end{pmatrix}.$$

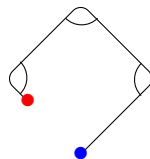
- 3) la tortue va “manger” un pair, un nombre impair composé, un pair. Elle était initialement positionnée sur le sommet  $d$  du carré courant.



*Changement de coordonnées :*

$$\begin{pmatrix} x \\ y \end{pmatrix} \longrightarrow \begin{pmatrix} x - 1 \\ y - 1 \end{pmatrix}.$$

- 4) la tortue va “manger” un pair, un nombre impair composé, un pair. Elle était initialement positionnée sur le sommet  $b$  du carré courant.



*Changement de coordonnées :*

$$\begin{pmatrix} x \\ y \end{pmatrix} \longrightarrow \begin{pmatrix} x - 1 \\ y + 1 \end{pmatrix}.$$

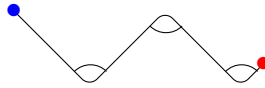
- 5) la tortue va “manger” un pair, un nombre premier, un pair. Elle était initialement positionnée sur le sommet  $d$  du carré courant.



Changement de coordonnées :

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} x - 3 \\ y + 1 \end{pmatrix}.$$

- 6) la tortue va “manger” un pair, un nombre premier, un pair. Elle était initialement positionnée sur le sommet  $g$  du carré courant.



Changement de coordonnées :

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} x + 3 \\ y - 1 \end{pmatrix}.$$

*Remarques :*

- un nombre impair de nombres traversés par la tortue la fait passer soit d’une des droites ( $H$ ) ou ( $B$ ) à la droite ( $M$ ), ou bien lui fait faire le contraire, c’est-à-dire passer de la droite ( $M$ ) à l’une des droites ( $H$ ) ou ( $B$ ). Avant de “manger” un nombre pair, la tortue est forcément sur la droite ( $M$ ), ce qui ne lui permet de se trouver qu’à deux sommets possibles d’un carré : les sommets  $g$  ou  $d$ . C’est ce qui explique les deux seuls cas 5) et 6) correspondant à la traversée d’un nombre premier, à opposer aux cas possibles correspondant à la traversée d’un nombre impair composé.
- Les cas 1) et 3) (resp. 2) et 4), ou bien 5) et 6)) sont miroirs l’un de l’autre.

### Quelle mesure choisir ?

Ce qui est vraiment rigolo à voir, c’est que de temps en temps, la tortue semble s’évader pour construire quelques carrés supplémentaires, une sorte d’extension de la suite courante, et ceci vraiment de façon erratique, parfois vers la droite et parfois vers la gauche. Elle semble simuler le comportement d’une machine de Turing, avec son ruban infini.

On compte les carrés : jusqu’à 100, la tortue en crée 8, jusqu’à 1000, elle en crée 27, jusqu’à 10000, elle en crée 65 et jusqu’à 100000, elle en crée 186 (voir le gif animé gif).

On aurait pu considérer que les pairs étant toujours composés sauf 2, on ne devrait pas leur associer de mouvement. La tortue ne bougerait que pour les nombres impairs, dans un sens pour les premiers et dans le sens opposé pour les composés et la mesure ne ferait que les compter mais le passage à la séquence de petits carrés, en effectuant le passage à deux dimensions, peut peut-être être porteur d’une information supplémentaire.

On note dans un tableau, le nombre  $n$  jusqu'auquel la mesure a été faite, l'extension maximum à gauche  $xmin$ , l'extension maximum à droite  $xmax$ , et le nombre de carrés.

n	xmin	xmax	mesure $\mu$
10	-1	6	3.5
100	-10	6	8
1000	-42	12	27
10000	-108	12	60
100000	-664	12	338
1000000	-1116	12	564
10000000	-1116	1102	1109
100000000	-3652	4922	4287
1000000000	-3652	17210	10431

Ci-dessous le programme modifié pour faire le comptage des carrés, notre mesure naturelle :

```

...
plt.axis('equal')
x = 0 ; y = 0 ; direction = 1j-1
xmin = 0 ; xmax = 0
for n in range(2,1000001):
    ancienx = x ; ancieny = y
    if direction == 1j+1:
        x = x+1
        y = y+1
    else:
        if direction == 1j-1:
            x = x-1
            y = y+1
        else:
            if direction == -1j-1:
                x = x-1
                y = y-1
            else:
                if direction == -1j+1:
                    x = x+1
                    y = y-1

    if x < xmin:
        xmin = x
    if x > xmax:
        xmax = x
    if premier(n):
        direction = direction*(-1j)
    else:
        direction = direction *(1j)
    plt.plot([ancienx, x], [ancieny, y])
    print('xmin = ',xmin, ' xmax = ', xmax)
plt.show()

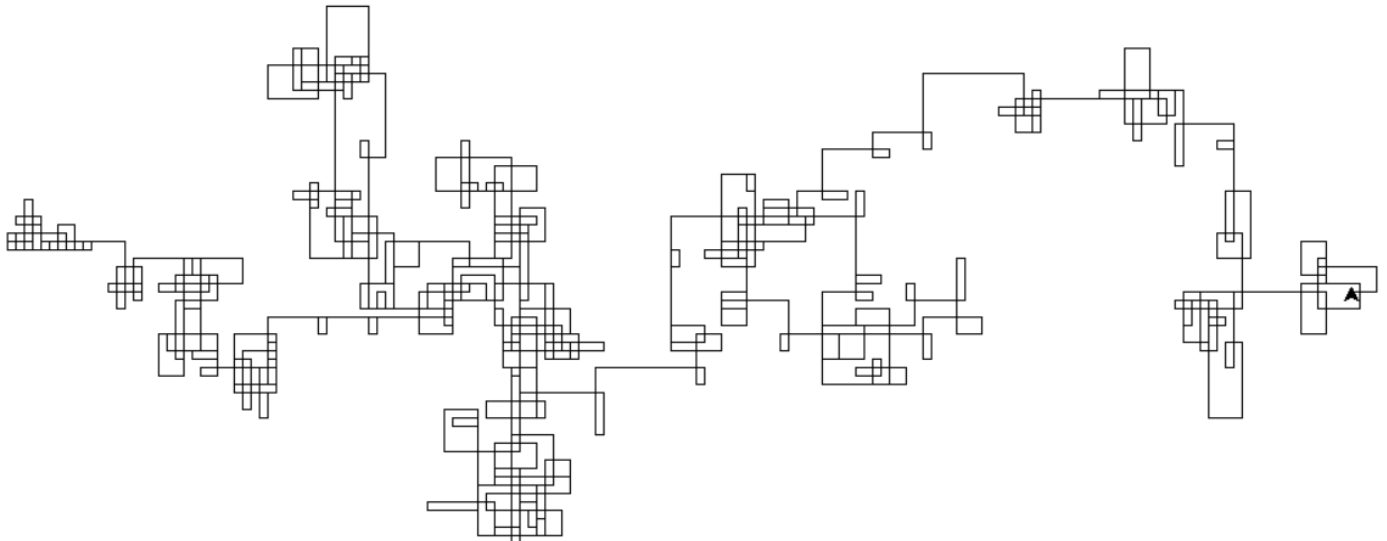
```

## Ajout d'un degré de liberté

Celui à qui je montre cette merveilleuse tortue apprivoisée et l'expérience inoubliable que provoque son observation<sup>2</sup>, gêné par ce confinement à un seul degré de liberté (qui est plus contraignant que les règles du pays Flatland !) souhaite que le programme libère un peu la tortue selon la règle "sur un nombre premier, tourne à droite ; quel que soit le nombre analysé, avance" (le début du programme est inchangé) :

```
...
setup()
speed(0)
setheading(90)
up()
turtle.setposition(-300,300)
left(45)
down()
for n in range(2,5001):
    if premier(n):
        right(90)
    forward(3)
exitonclick()
```

On obtient, pour les nombres jusqu'à 5000 la merveilleuse production artistique ci-dessous !



---

<sup>2</sup>Et cette même magie continue d'opérer, 50 ans après l'invention de Logo, sur les élèves qui découvrent le chaton Scratch, et le plaisir de le voir exécuter les instructions que notre cerveau a inventées, ça reste grisant.