

6 - Les maths en Netlogo

Héctor Satizábal

IICT- HEIG-VD

2018 - 2019

- ▶ Math: [opérateurs](#)
- ▶ Aléatoire: [random](#),
[random-seed](#), [new-seed](#)

Adapté à partir du matériel du cours créé par **Prof. Andres Perez-Uribe**

Opérateurs

Le **type** d'une variable détermine l'ensemble des opérateurs qui peuvent lui être appliqués

- ▶ Les opérateurs applicables à des variables de type **boolean** sont:
`not`, `and`, `or`, et `xor`
- ▶ Les opérateurs applicables à des variables de type **number** sont:
`+`, `-`, `*`, `/`, `^`
`>`, `<`, `>=`, `<=`, `=`, `!=` ou `not=`
`abs`, `acos`, `asin`, `atan`, `bitand`, `bitor`, `bitxor`, `cos`,
`div`, `exp`, `int`, `ln`, `log`, `max`, `min`, `mod`, `random`,
`round`, `sin`, `sqrt`
- ▶ On verra peut-être plus tard les opérateurs applicables à des variables de type **string** et **list**

La tortue et les maths

Les tortues de Netlogo peuvent réaliser des calculs complexes:

Exemples:

1. `show 2 * 80`

2. `show 1534 / 43`

3. `show 2 ^ 80`

4. Déclarer une variable x et écrire le code ci-dessous:

```
set x 1
```

```
repeat 10 [print x * 9 set x x + 1]
```

Voir le résultat dans la fenêtre du “Command center”

La tortue et les maths

5. Dessiner une ligne droite

Exemple: $b = 0.5a - 5$, pour $-15 < a < 15$

Déclarer les variables a et b:

```
globals [a b] ;dans code
```

Créer une tortue et la programmer ainsi:

```
ca ;dans observer
```

```
crt 1 ;dans observer
```

```
set a -15 ;dans turtles
```

```
repeat 30 [set b (0.5 * a) - 5 setxy a b set pcolor  
yellow set a a + 1]
```

La tortue et les maths

Dessiner une parabole:

$$b = 0.1a^2 + 10, \text{ pour } -15 < a < 15$$

Dessiner un sinus:

$$b = 10 * \sin(2 * 180 * (a + 15)/20), \text{ pour } -15 < a < 15$$

Les nombres aléatoires

aléatoire (random en anglais) = au hasard

- ▶ Produire des nombres sans suite apparente (aléatoires) est une nécessité dans de nombreux domaines de l'informatique. C'est le cas des jeux, de la modélisation, de la sécurité et de la résolution de problèmes par des méthodes heuristiques
- ▶ Par exemple, pour réussir la communication entre plusieurs ordinateurs, l'aléatoire vient à l'aide de la politesse: lorsque deux machines "parlent" en même temps, les deux se taisent et reprennent la communication après un certain temps aléatoire, diminuant ainsi la probabilité d'une nouvelle "collision". Cette technique est définie par le standard Ethernet pour les réseaux d'ordinateurs (IEEE 802.3)

Des nombres pseudo aléatoires

- ▶ On ne dispose malheureusement pas dans un ordinateur de **vraie source de hasard**. Tout est synchronisé sur des horloges très stables et les calculs sont faits avec une régularité et une fiabilité qui rendent le système prédictible.
- ▶ Pour produire une séquence de nombres sans suite apparente il y a deux approches:
 1. De “vrais” nombres aléatoires: ils sont habituellement dérivés des sources analogiques (phénomènes physiques)
 2. Des nombres pseudo aléatoires: ils sont générés à l'aide des formules mathématiques voire des algorithmes

Formule de Lehmer: $x_{n+1} = (a * x_n + b) \bmod c$

Elle génère des nombres entre 0 et $c - 1$

Des nombres pseudo aléatoires

Exemple:

$$x_{n+1} = (25 * x_n + 16) \bmod 256$$

si $x_0 = 12 \implies 60, 236, 28, 204, 252, 172, \dots$

La séquence semble ne pas avoir de suite apparente, mais les nombres sont tous des nombres pairs!

La formule est simple mais le choix des trois paramètres ne doit pas être fait à la légère.

Ceci est la formule utilisée par le système d'exploitation UNIX:

$$x_{n+1} = (1103515245 * x_n + 12345) \bmod 2^{31}$$

Nombres aléatoire dans Netlogo

Dans Netlogo, c'est l'instruction **random** qui génère des nombres pseudo aléatoires. On n'a pas à générer nous mêmes ces séquences

Exemple:

```
show random 100      ;affiche un nombre entre 0 et 99
```

Les instructions **random-seed** et **new-seed** initialisent la séquence (à l'aide de la valeur du paramètre x_0)

Exemple d'utilisation: deux tortues qui dessinent aléatoirement

Créer deux tortues

```
ca crt 2              ;dans observer
```

Faire bouger les tortues

```
pd                    ;dans turtles
```

```
repeat 20 [fd random 10 rt random 90]
```

Exemple: positionnement aléatoire

```
ca                                ;dans observer  
crt 40                            ;dans observer
```

```
setxy random 32 random 32      ;dans turtles
```

* ici, `random 32` est équivalent à $-16 + \text{random } 32$,
pourquoi?

Exercices: nombres aléatoires

1. Créer 20 tortues et les positionner aléatoirement à l'intérieur de l'environnement ($-16 \leq x \leq 16$, $0 \leq y \leq 16$) à l'aide de l'instruction `setxy`
2. Créer 20 tortues et les positionner aléatoirement à l'intérieur de l'environnement ($0 \leq x \leq 16$, $-16 \leq y \leq 0$) à l'aide de l'instruction `setxy`
3. Écrire un programme qui dessine une figure qui a un nombre `n` de côtés, `n` est un nombre aléatoire et $3 \leq n \leq 12$
4. Créer et programmer des boutons associés à des procédures pour tout effacer (`ca`), créer une nouvelle tortue (`crt 1`) et pour bouger aléatoirement en permanence, tout en changeant de couleur de manière aléatoire

Comment sauver vos projets ?

L'option “**Save**” du menu File sauvegarde:

- ▶ Les procédures qui décrivent le comportement des tortues
- ▶ Les procédures de l'environnement
- ▶ L'environnement lui même (boutons et environnement dessiné), la forme de la tortue, etc...