4 - Procédures et fonctions

Héctor Satizábal
IICT- HEIG-VD
2018 - 2019

- Procedures: to, to-report

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

Procédures et fonctions

- Il est pratiquement impossible d'écrire un programme compliqué en alignant un grand nombre d'instructions de base
- Le concept de procédure ou sous-routine dans un langage de programmation permet de grouper un bloc d'instructions et de lui donner un nom, qui pourra être utilisé ailleurs dans un même programme sans devoir récrire ledit bloc
- Une fonction est une procédure qui renvoie une valeur en sortie. Par exemple, `sin` et `cos` sont des fonctions du langage Netlogo

Exemples de procédures

- Ecrire dans l'onglet "code"

```
to angle_droit
    fd 10
    rt 90
end
```

- Maintenant on peut appeler la procédure `angle_droit` depuis la fenêtre de commandes (turtles), ou depuis une autre procédure du même programme (code), comme suit:

```
to carre
    setxy 0 0
    repeat 4 [ angle_droit ]
end
```

Exemples de procédures

- Finalement vous pouvez appeler la procédure `carre` depuis la fenêtre de commandes (turtles). Cette procédure appellera la procédure `angle_droit` 4 fois

- NOTE: si vous exécutez la procédure `angle_droit`, même si la procédure `carre` est écrite juste après, cette deuxième procédure n'est pas exécutée

Multiples procédures

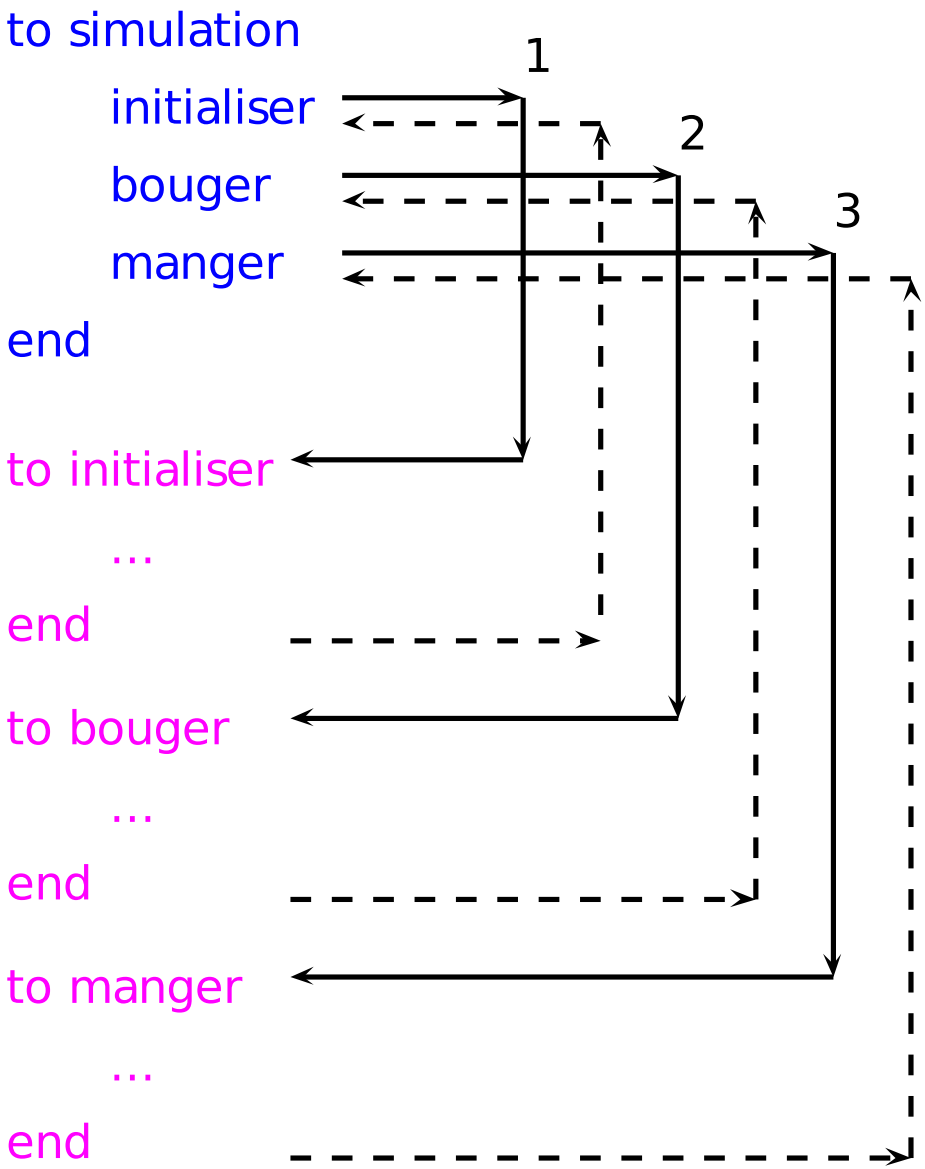
On a vu qu'une procédure peut appeler d'autres procédures, par exemple, la procédure `carre` appelait la procédure `angle_droit`

Souvent, on définit une procédure principale qui appelle d'autres procédures:

```
to simulation
    initialiser
    bouger
    manger
end
```

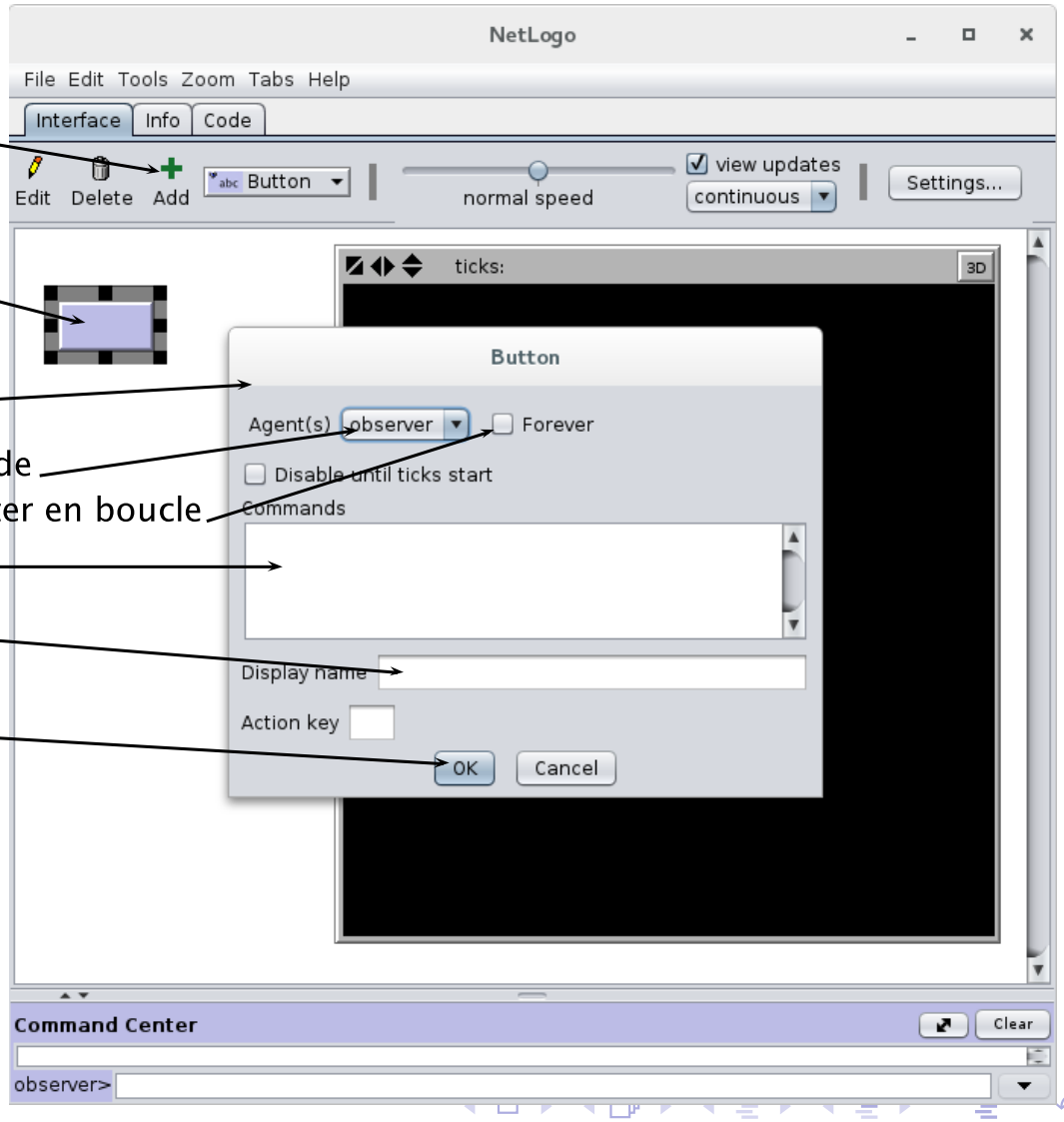
`initialiser`, `bouger` et `manger` sont d'autres procédures du même programme

Séquence d'exécution



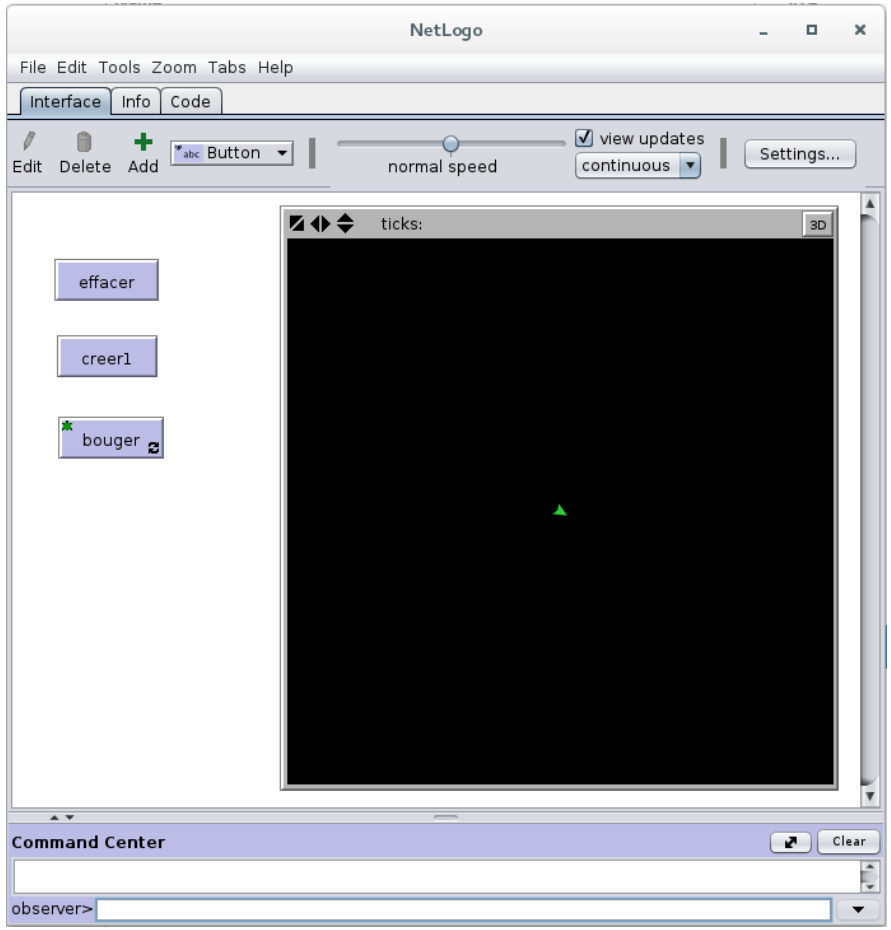
Programmation de boutons pour exécuter des procédures

1. Créer un bouton
2. Placer le bouton
3. Programmer le bouton
   a) Indiquer qui doit exécuter la commande
   b) Indiquer si la commande doit s'exécuter en boucle
   c) Indiquer la commande à exécuter
   d) Donner un nom
4. OK



Exercice

Créer et programmer des boutons associés à des procédures pour tout effacer (`ca`), créer une nouvelle tortue (`crt 1`) et bouger en dessinant une spirale



Boucles non conditionnelles

- En Netlogo, on peut programmer une boucle non conditionnelle, c'est à dire une boucle qui se répète sans cesse, tout simplement en rappelant la boucle dans la boucle elle même

```
to boucle_non_conditionnelle
   set color red
   pd
   set cote (cote + 1)
   fd cote
   rt 90
   boucle_non_conditionnelle
end
```

- Les boucles non conditionnelles sont une technique NON CONSEILLÉE en Netlogo

Paramètres de procédure

- Il est souvent utile de pouvoir écrire une procédure d'une manière générale et de pouvoir la paramétrer au moment de son exécution
- Exemple: on veut écrire une procédure qui fait bouger la tortue en dessinant un nombre de points paramétrable:

```
to dessiner [nombre_de_points]
    repeat nombre_de_points
    [   set pcolor red
        fd 2
    ]
end

dessiner 5              ;passage de paramètres
```

Paramètres de procédure

- L'information transmise via les paramètres de procédure n'est disponible que lors de l'exécution de la procédure qui les reçoit. Ces paramètres sont locaux à la procédure
- On peut passer plusieurs paramètres à une même procédure:

```
to dessiner [nombre_de_points distance couleur]
    repeat nombre_de_points
    [   set pcolor couleur
        fd distance
    ]
end

dessiner 6 3 red                ;passage de paramètres
```

Exercice 1

- Programmer une tortue pour dessiner un polygone de n'importe quel nombre de côtés et d'une taille quelconque

```
to dessiner [n l]


end
```

ou n est le nombre de côtés et l est la longueur de chaque côté du polygone

Exemple:

```
dessiner 3 10     ;doit dessiner un triangle côté 10
dessiner 6 20     ;doit dessiner un hexagone côté 20
```

Exercice 1 (suite)

- Programmer une tortue pour dessiner un polygone de n'importe quel nombre de côtés, d'une taille quelconque et d'une couleur quelconque, choisis par l'utilisateur lors de son exécution

Paramètres de fonction

- Netlogo permet la création de fonctions: procédures qui peuvent donner une valeur en sortie
- Ces fonctions sont appelées des "reporters" en Netlogo
- La syntaxe pour créer une fonction est la suivante:

```
to-report nom [paramètres_en_entrée]
    ...
    ...
    report valeur_de_sortie
end
```

Exemple

- Le code ci-dessous calcule la distance des tortues par rapport à l'origine $(0, 0)$

```
to-report calculer_distance
    report sqrt( (xcor * xcor) + (ycor * ycor) )
end
```

- Le code ci-dessous calcule la distance des tortues par rapport à n'importe quel point (ox, oy)

```
to-report calculer_distance [ox oy]
    let x_diff (xcor - ox)
    let y_diff (xcor - oy)
    report sqrt( (x_diff * x_diff) + (y_diff * y_diff) )
end
```

Exercice 2

- Programmer une fonction pour qu'une tortue dessine un polygone de n'importe quel nombre de côtés, d'une taille quelconque et d'une couleur quelconque, choisis par l'utilisateur lors de son exécution. La fonction doit calculer et retourner la surface du polygone dessiné

$$A = \frac{1}{4} \cdot n \cdot s^2 \cdot cot\left(\frac{180^\circ}{n}\right)$$

Où

A = surface
n = nombre de cotés
s = longueur de chaque coté