

Simulation de Modèles Discrets

Fabien Givors

d'après les cours d'Alberto Dennunzio

Département d'Informatique
Université de Nice-Sophia Antipolis

Définir différentes espèces de turtles

breed [nom_pluriel nom_singulier]

Définir différentes espèces de turtles

breed [nom_pluriel nom_singulier]

Automatiquement disponibles :

- 1 nom_pluriel (variable et agentset)
- 2 nom_singulier (rapporteur)
- 3 nom_pluriel-own (pour définir une variable associée à l'espèce)
- 4 create-nom_pluriel (procédure)
- 5 hatch-nom_pluriel (procédure)
- 6 sprout-nom_pluriel (procédure)
- 7 nom_pluriel-here (rapporteur)
- 8 nom_pluriel-at (rapporteur)
- 9 nom_pluriel-on (rapporteur)
- 10 is-nom_singulier ? (rapporteur)

create-nom_pluriel nombre [commandes]

création d'un certain nombre d'êtres d'une certaine espèce qui exécutent les commandes indiquées

Exemple

```
breed [moutons mouton]
breed [loups loup]
loups-own [energy]

to setup
  clear-all
  create-moutons 10 [
    setxy random-xcor random-ycor
    set color white ]
  create-loups 5 [
    setxy random-xcor random-ycor
    set color black
    set energy random 10 ]
  ask patches [
    set pcolor green ]
end
```

Dans le menu “Turtle Shapes Editor” dans “Tools”

Dans le menu “Turtle Shapes Editor” dans “Tools”

- ▶ liste des formes disponibles
- ▶ importation d'autres formes
- ▶ création de nouvelles formes

Exemple

```
breed [moutons mouton]
breed [loups loup]
loups-own [energy]

to setup
  clear-all
  create-moutons 10 [
    setxy random-xcor random-ycor
    set shape "sheep"
    set color white ]
  create-loups 5 [
    setxy random-xcor random-ycor
    set shape "wolf"
    set color black
    set energy random 10 ]
  ask patches [
    set pcolor green ]
end
```

set-default-shape nom_pluriel nom_de_forme

Spécifie la forme initiale qui sera donnée à une certaine espèce

Exemple

```
breed [moutons mouton]
breed [loups loup]
loups-own [energy]

to setup
  clear-all
  set-default-shape moutons "sheep"
  create-moutons 10 [
    setxy random-xcor random-ycor
    set color white ]
  create-loups 5 [
    setxy random-xcor random-ycor
    set shape "wolf"
    set color black
    set energy random 10 ]
  ask patches [
    set pcolor green ]
end
```

...-*here*

nom_pluriel-*here*

Rapporte l'agentset qui contient les turtles qui se trouvent sur le patch de l'agent qui appelle le rapporteur

let nom valeur

La variable locale existe seulement dans le bloc de commandes où elle est définie !

Exemple

```
to go
  ask moutons [
    move
    rep ]
  ask loups [
    move
    set energy energy - 1
    catch-moutons
    death
    rep ]
end

to catch-moutons
  let proie one-of moutons-here
  if proie != nobody [
    ask proie [ die ]
    set energy energy + 30 ]
end
```

sprout-nom_pluriel nombre [commandes]

création sur la patch courante d'un certain nombre d'êtres d'une certaine espèce qui exécutent les commandes indiquées

Exemple

```
breed [chiens chien]
breed [chats chat]
breed [vaches vache]

to setup
  clear-all
  set-default-shape chiens "dog"
  set-default-shape chats "cat"
  set-default-shape vaches "cow"
  ask patch 0 1 [ sprout-chiens 1 ]
  ask patch 0 0 [ sprout-chats 1 ]
  ask patch 0 -1 [ sprout-vaches 1 ]
end
```

hatch-nom_pluriel nombre [commandes]

création d'un certain nombre d'êtres d'une certaine espèce qui héritent les variables de son propre parent (l'agent qui appelle *hatch*) et qui exécutent les commandes indiquées