

# Simulation de Modèles Discrets

Fabien Givors

*d'après les cours d'Alberto Dennunzio*

Département d'Informatique  
Université de Nice-Sophia Antipolis

# Fichiers : ouverture et fermeture

*user-new-file*

permet à l'utilisateur de choisir le nom d'un fichier qui sera créé et retourne son nom.

# Fichiers : ouverture et fermeture

*user-new-file*

permet à l'utilisateur de choisir le nom d'un fichier qui sera créé et retourne son nom.

*file-open* string

ouvre le fichier dont le nom est passé en paramètre.

# Fichiers : ouverture et fermeture

*user-new-file*

permet à l'utilisateur de choisir le nom d'un fichier qui sera créé et retourne son nom.

*file-open* string

ouvre le fichier dont le nom est passé en paramètre.

*file-close*

ferme le fichier ouvert (avec *file-open*.)

# Fichiers : ouverture et fermeture

*user-new-file*

permet à l'utilisateur de choisir le nom d'un fichier qui sera créé et retourne son nom.

*file-open* string

ouvre le fichier dont le nom est passé en paramètre.

*file-close*

ferme le fichier ouvert (avec *file-open*.)

*file-delete* string

supprime le fichier spécifié en paramètre.

# Fichiers : ouverture et fermeture

*user-new-file*

permet à l'utilisateur de choisir le nom d'un fichier qui sera créé et retourne son nom.

*file-open* string

ouvre le fichier dont le nom est passé en paramètre.

*file-close*

ferme le fichier ouvert (avec *file-open*.)

*file-delete* string

supprime le fichier spécifié en paramètre.

*file-exists ?* string

retourne vrai si et seulement si le paramètre est le nom d'un fichier existant.

*user-directory*

permet à l'utilisateur de choisir un répertoire existant et retourne son chemin.

## *user-directory*

permet à l'utilisateur de choisir un répertoire existant et retourne son chemin.

## *set-current-directory* string

fixe comme répertoire de travail courant celui dont le chemin est spécifié en paramètre. Il sera utilisé par *file-open*, *file-delete* et *file-exists* ?.



*file-write* valeur

écrit valeur (nombre, liste, string...) dans un fichier ouvert.

*file-write* valeur

écrit valeur (nombre, liste, string...) dans un fichier ouvert.

*file-print* valeur

écrit valeur (nombre, liste, string...) suivi d'un retour chariot dans un fichier ouvert.

*file-write* valeur

écrit valeur (nombre, liste, string...) dans un fichier ouvert.

*file-print* valeur

écrit valeur (nombre, liste, string...) suivi d'un retour chariot dans un fichier ouvert.

*file-read*

lit et retourne la prochaine constante (nombre, liste, string...) d'un fichier ouvert.

# Example 1

```
to setup-write
  clear-all
  create-turtles 5 [ setxy random-xcor random-ycor ]
  set-current-directory user-directory
  file-open "myfile.txt"
  ask turtles [ file-write xcor file-write ycor ]
  file-close
end

to setup-read
  create-turtles 5
  file-open "myfile.txt"
  ask turtles with [who > 5] [ setxy file-read +1 file-read + 1 ]
  file-close
end
```

*foreach* liste [commands]

Exécute les commandes pour chaque élément de la liste.  
Utilise le symbole ? pour faire référence à l'élément courant de la liste.

```
foreach [1 2 3 4] [show ? + 10]  
foreach turtles [ ask ? [show self ] ]
```

## Example 2

```
to setup
  clear-all
  create-turtles 5 [ setxy random-xcor random-ycor ]
  let file user-new-file
  if is-string? file [
    if file-exists? file [ file-delete file ]
    file-open file
    write-to-file
  ]
end

to write-to-file
  foreach sort turtles [
    ask ? [ file-print ( word self " pxcor : " pxcor ) ]
  ]
end
```

# Film (Quicktime)

*movie-start* nom\_du\_fichier

Crée un nouvel film dans le fichier spécifié qui doit avoir l'extension .mov (fichier Quicktime).

# Film (Quicktime)

*movie-start* nom\_du\_fichier

Crée un nouvel film dans le fichier spécifié qui doit avoir l'extension .mov (fichier Quicktime).

*movie-grab-view*

*movie-grab-interface*

rajoute l'image de la vue/interface courante au film.



# Film (Quicktime)

*movie-start* nom\_du\_fichier

Crée un nouvel film dans le fichier spécifié qui doit avoir l'extension .mov (fichier Quicktime).

*movie-grab-view*

*movie-grab-interface*

rajoute l'image de la vue/interface courante au film.

*movie-close*

arrête d'enregistrer le film courant.

# Exemple

```
to setup
  clear-all
  create-turtles 5
  set-current-directory user-directory
  movie-start "out.mov"
  movie-grab-view
  repeat 30 [
    go
    movie-grab-view
  ]
  movie-close
end

to go
  ask-turtles [fd 1]
end
```