

Ce tutoriel en trois parties présente comment NetLogo vous permet d'analyser une simulation multi-agents et de créer votre premier modèle multi-agents.

## Introduction

### **Exercice 1 : Qu'est-ce que NetLogo ?**

NetLogo est un langage de programmation et un environnement de développement (IDE) pour la conception de simulations multi-agents. Initié par Uri Wilens en 1999, NetLogo propose une large bibliothèque de modèles pour les systèmes complexes issues de l'Économie, de la Biologie, de la Physique, etc. À l'aide d'une interface graphique riche (interrupteurs, curseurs, sélecteurs, etc.), NetLogo permet d'explorer des phénomènes émergents. Vous pouvez également définir vos propres modèles dans un environnement à deux dimensions à l'aide 4 types d'agents (*turtles*, *patches*, *links* et *observer*). Les tuiles (*patches*) sont le sol sur lequel les tortues (*turtles*) se déplacent. Les liens (*links*) sont les connexions entre les tortues. L'observateur est un agent qui supervise le modèle.

Développé en Scala et en Java, NetLogo est open-source et multi-plateforme, car il s'exécute sur la machine virtuelle Java.

**Q1.** Téléchargez NetLogo à l'URL suivante <http://ccl.northwestern.edu/netlogo/download.shtml> et installez le sur votre machine.

### **Exercice 2 : Exemple de modèle**

Dans cet exemple, nous allons nous intéresser à la modélisation informatique des comportements humains lors d'une fête. Avez-vous déjà remarqué que les convives circulent, se regroupent et que ces groupes changent. Pourquoi les groupes ont-ils tendance à se former en majorité d'hommes ou de femmes ?

**Q1.** Utilisez le modèle "Party" de NetLogo pour proposer une explication à ce phénomène.

**Q1.1.** Démarrer NetLogo.

**Q1.2.** Sélectionnez Models Library dans le menu File.

**Q1.3.** Ouvrez le répertoire Social Science.

**Q1.4.** Cliquez sur le modèle Party.

**Q1.5.** Appuyez sur le bouton open. Vous pouvez agrandir l'interface en sélectionnant Larger dans le menu Zoom.

**Q1.6.** Appuyez sur le bouton setup.

**Q2.** L'interface représente 150 convives divisés en 10 groupes constitués d'hommes (en bleu) et de femmes (en rose) dont les tailles sont affichées. Simulez l'évolution des groupes au cours de la fête.

**Q2.1.** Appuyez sur le bouton go. En appuyant à nouveau sur le bouton go, le modèle s'arrête manuellement.

**Q2.2.** Observez le mouvement des convives jusqu'à ce que le modèle s'arrête.

**Q2.3.** Observez les courbes pour comprendre le phénomène.

**Q2.4.** Appuyer sur le bouton setup pour réinitialiser le modèle. Utilisez le curseur de vitesse pour ralentir le modèle.

**Q3.** Vous observez que les groupes initialement mixtes disparaissent au profit de groupes unisexes. En effet, le comportement des agents a été défini par le modélisateur sur la base d'un seuil de tolérance (tolerance), i.e. qui détermine le pourcentage de personnes du sexe opposé au-dessous duquel un convive est à l'aise (happy).

Si l'individu se trouve dans un groupe dont le pourcentage de personnes du sexe opposé est supérieur à son seuil de tolérance, il devient mal à l'aise et quitte le groupe pour un autre. La simulation s'arrête quand tous les convives sont à l'aise. Notez que dans le modèle, le seuil de tolérance des convives n'est pas fixe mais modifiable à l'aide d'un curseur.

**Q3.1.** Si le bouton go est enfoncé, la simulation est en cours. Appuyez à nouveau sur le bouton go pour l'arrêter.

**Q3.2.** Réglez le curseur tolerance avec une nouvelle valeur.

**Q3.3.** Appuyez sur le bouton setup pour réinitialiser le modèle.

**Q3.4.** Appuyez sur le bouton go pour redémarrer la simulation.

**Q4.** Réinitialisez le modèle pour que tous les groupes soient initialement mixtes.

**Q4.1.** Quelle doit être le seuil de tolérance pour obtenir uniquement des groupes mixtes ? Faites une prédiction et testez-la.

**Q4.2.** Quelle doit être le seuil de tolérance pour obtenir jusqu'à 1, 2, 3, 4 ou 7 groupes unisexes ?

**Q4.3.** Est-ce que le nombre de pas de simulation varie avec le seuil de tolérance ? Pourquoi ?

## Modèles

Cette partie présente de manière plus approfondie les fonctionnalités disponibles lors de l'exploration des modèles.

### Exercice 3 : Le modèle Wolf Sheep Predation

**Q1.** Explorez en détail un modèle proies-prédateurs bien connu Biologie pour étudier la stabilité des écosystèmes.

**Q1.1.** Ouvrez la bibliothèque de modèles à partir du menu File.

**Q1.2.** Sélectionnez le modèle Wolf Sheep Predation dans le répertoire Biology et appuyez sur le bouton Open.

**Q2.** L'onglet Info explique en détail ce modèle. Lisez ces explications. Que se passe-t-il quand il n'y a plus de loups et trop de moutons ?

**Q3.** L'onglet Interface est constitué d'éléments vous permettant d'interagir avec le modèle :

- les boutons qui sont bleus permettent de configurer, de démarrer et d'arrêter le modèle;
- les curseurs et les interrupteurs qui sont verts modifient les paramètres du modèle;
- Les moniteurs et les graphiques qui sont beiges affichent les données.

À l'ouverture, la vue du modèle (i.e. la représentation graphique des agents) est vide. Vous devez configurer le modèle.

**Q3.1.** Appuyez sur le bouton setup. Que voyez-vous apparaître dans la vue ?

**Q3.2.** Appuyez sur le bouton go pour démarrer la simulation. Pendant que le modèle fonctionne, qu'arrive-t-il aux populations de loups et de moutons ?

**Q3.3.** Appuyez sur le bouton go pour arrêter le modèle.

### Exercice 4 : Contrôler le modèle

Quand vous appuyez sur un bouton, le modèle répond par une action. Un bouton est :

- soit un bouton forever (représenté avec deux flèches dans le coin inférieur droit) qui effectue une action continue. Appuyez à nouveau sur le bouton pour stopper l'action et faire réapparaître le bouton. La plupart des modèles ont un bouton forever appelé go qui permet de l'arrêter ou de le mettre en pause puis le reprendre en appuyant à nouveau sur ce bouton<sup>1</sup>;

<sup>1</sup>. Si le modèle est coincé, vous pouvez l'arrêter avec l'élément Halt du menu Tools qui peut interrompre le modèle au milieu d'une action. Réactiver ce modèle peut amener à des résultats inconsistants.

- soit un bouton once (représenté sans flèche) qui effectuent une action ponctuelle puis s'arrêtent. Lorsque l'action est terminée, le bouton réapparaît. La plupart des modèles ont un bouton once appelé setup. Certains modèles ont un bouton once appelé go once (ou step once) qui fait avancer le modèle d'un pas de temps (tick) pour suivre au plus près la progression du modèle.

**Q1.** Expérimenez avec les boutons setup et go dans le modèle de Wolf Sheep Predation. Obtenez-vous des résultats différents si vous exécutez le modèle plusieurs fois avec les mêmes paramètres ?

### **Exercice 5 : Contrôler la vitesse**

Dans l'onglet Interface, un curseur permet de contrôler la vitesse de la simulation, c'est-à-dire le laps de temps entre deux pas de simulation. Ralentir la simulation permet de mieux discerner le comportement des agents. Lorsque la simulation est accélérée, NetLogo ne met pas nécessairement à jour la vue à chaque pas.

**Q1.** Ralentissez autant que possible la simulation pour discerner les comportements individuels des loups et des moutons.

### **Exercice 6 : Réglage des paramètres**

Les **sélecteurs**, **commutateurs** et **curseurs** permettent de modifier les paramètres d'un modèle pour explorer différents scénarios ou plusieurs hypothèses.

**Q1.** Expérimenez les effets des paramètres sur le comportement du modèle.

**Q1.1.** Le nombre de pas de simulation est affiché près du curseur de contrôle de la vitesse. Exécutez 100 pas de simulation en appuyant sur setup puis sur go. Quelle est la taille de la population de moutons ?

**Q1.2.** À l'aide du sélecteur model-version, choisissez le modèle sheep-wolves-grass plutôt que sheep-wolves pour considérer l'herbe comme une ressource rare associée à un taux de repousse. Exécutez 100 pas de simulation en appuyant sur setup puis sur go. Quelle est la taille de la population de moutons ?

**Q2.** À l'aide du curseur initial-number-sheep, démarrez avec 10 moutons plutôt que 100. Quelle est la conséquence sur la population de loups ?

**Q3.** Qu'arrivera-t-il à la population de moutons s'il y a initialement plus de moutons et moins de loups ?

**Q3.1.** Changez le sélecteur model-version de sheep-wolves-grass à sheep-wolves.

**Q3.2.** Réglez le curseur initial-number-sheep sur 100.

**Q3.3.** Réglez le curseur initial-number-wolves sur 20.

**Q3.4.** Exécutez 100 pas de simulation, en appuyant sur setup et ensuite sur go. Êtes-vous surpris par le résultat ?

### **Exercice 7 : Collecte d'informations**

Au-delà de la vue qui permet d'observer la simulation, NetLogo propose également des **graphiques** et des **moniteurs**. Dans le modèle Wolf Sheep Predation, les moniteurs affichent au fur et à mesure de simulation la population de moutons et de loups, ainsi que la quantité d'herbe. Dans le modèle Wolf Sheep Predation, le graphique contient trois courbes représentant le nombre de moutons, de loups et la quantité d' $\frac{\text{herbe}}{4}$ .

**Q1.** Enregistrez les données d'une simulation pour les analyser dans une autre application.

**Q1.1.** Changez le sélecteur model-version de sheep-wolves à sheep-wolves-grass.

**Q1.2.** Exécutez 100 pas de simulation, en appuyant sur setup et ensuite sur go.

**Q1.3.** Exportez le tracé avec le bouton droit de la souris et en choisissant Export... dans le menu contextuel.

## **Exercice 8 : Contrôle de la vue**

Dans l'onglet Interface, une barre d'outils permet de contrôler la vue.

**Q1.** Expérimitez l'effet de ces outils.

**Q1.1.** Au cours de la simulation (après avoir appuyé sur setup et ensuite sur go) déplacez le curseur de vitesse vers la gauche pour observer en détail les comportements individuels.

**Q1.2.** Au cours de la simulation, déplacez le curseur de vitesse vers la droite et décochez la case view updates. Que se passe-t-il ? Stoppez la simulation et recochez cette case.

**Q2.** Appuyez sur le bouton Settings... qui permet d'accéder aux paramètres qui ne sont pas disponibles dans la barre d'outils.

**Q2.1.** Quels sont les valeurs actuelles pour min-pxcor, max-pxcor, min-pycor, max-pycor et la taille du patch ?

**Q2.2.** Appuyez sur cancel pour faire disparaître cette fenêtre sans modifier les paramètres.

**Q2.3.** Placez le pointeur de votre souris à côté de la vue. Le pointeur a la forme d'une croix.

**Q2.4.** Maintenez le bouton de la souris enfoncé et faites glisser la croix sur la vue. La vue est maintenant sélectionnée (entourée d'une bordure).

**Q2.5.** Faites glisser l'une des poignées noires carrées sur les bords et dans les coins.

**Q2.6.** Désélectionnez la vue en cliquant n'importe où sur le fond blanc de l'onglet Interface.

**Q2.7.** Appuyez sur le bouton Settings... Quels chiffres ont changé ? Quels chiffres n'ont pas changé ?

**Q3.** Un environnement NetLogo est une grille de tuiles, appelées patches, qui sont verts ou bruns dans la version

sheep-wolves-grass". Le patch (0,0) se situe au centre. Quand vous modifiez la taille d'une tuile, le nombre de tuiles ne change pas.

**Q3.1.** Dans la boîte de dialogue Setting, réglez la valeur max-pxcor à 30 et max-pycor à 10. Quelle est la forme de la vue ?

**Q3.2.** Dans la boîte de dialogue Setting, réglez la taille de la tuile à 20.

## **Commandes**

Dans cette partie, vous allez manipuler le fonctionnement interne des modèles, pour modifier leur apparence.

## **Exercice 9 : Modèle basique de trafic**

**Q1.** Le modèle basique de trafic simule le mouvement des voitures sur une autoroute. Chaque voiture suit une règle simple : elle ralentit si elle voit une voiture devant elle, sinon elle accélère sinon. Le modèle montre comment des embouteillages peuvent se former.

**Q1.1.** Ouvrez la bibliothèque de modèles à partir du menu File.

**Q1.2.** Sélectionnez Models Library dans le menu File.

**Q1.3.** Ouvrez le répertoire Social Science.

**Q1.4.** Cliquez sur le modèle Traffic basic.

**Q1.5.** Appuyez sur le bouton open.

**Q1.6.** L'onglet Info explique en détail ce modèle. Lisez ces explications. Quelles sont les variables du modèle ?

**Q2.** Ce modèle où une voiture rouge circule dans un flot de voitures bleues montre que des embouteillages peuvent se former sans cause spécifique. Testez différents jeux de paramètres et observez quelques passages pour mieux comprendre le modèle.

## **Exercice 10 : Centre de commande**

Le centre de commande, qui se trouve dans l'onglet Interface, permet de saisir des instructions pour un modèle. Ces commandes s'adressent aux agents : aux tortues (turtles), aux tuiles (patches), aux liens (link) ou à l'observateur (observer). Le centre de commande ne modifie pas le modèle de façon permanente. Il permet de manipuler directement le modèle pour répondre aux questions "What if" qui surgissent lorsque vous étudiez un modèle.

**Q1.** Appuyez sur le bouton setup.

**Q2.** Dans le centre de commande, choisissez observer et saisissez `ask patches [set pcolor yellow]`. Qu'est-il arrivé à la vue ? Pourquoi les voitures restent inchangées ? Que s'est-il passé dans le centre de commande ?

**Q3.** À quel résultat vous attendez vous après avoir envoyé à l'observer dans le centre de commande la commande

`ask turtles [set pcolor brown]`. Est-ce le cas ?

**Q4.** Dans le centre de commande, vous pouvez vous adresser à tout type d'agents.

**Q4.1.** Adressez aux tortues la commande `set color pink`.

**Q4.2.** Appuyez sur la touche de tabulation pour adresser aux tuiles la commande `set pcolor white`.

**Q4.3.** À quoi ressemble la vue maintenant ? Quelle est la différence entre ces commandes et celles passées à l'observateur précédemment ?

**Q5.** Vous avez peut-être remarqué dans la question précédent que vous avez utilisé deux variables différentes pour changer de couleur. Quelle est la différence entre `color` et `pcolor` ?

**Q5.1.** Modifiez la couleur des voitures (e.g. blue) et celle des tuiles (e.g. red)

Les couleurs sans NetLogo ont une valeur numérique<sup>2</sup>. 16 couleurs ont un nom : gray, red, ..., pink.

**Q5.2.** Appuyez sur setup. Que s'est-il passé ?

## **Exercice 11 : Monitorer et commander des agents**

**Q1.** Dans l'exercice précédent, vous avez utilisé la commande `set` pour changer les couleurs de toutes les voitures. Vous allez ici changer la couleur d'une seule voiture.

**Q1.1.** Appuyez sur le bouton setup pour faire apparaître la voiture rouge.

**Q1.2.** Faites un clic droit sur la voiture rouge.

**Q1.3.** Dans le sous-menu contextuel, sélectionnez `inspect turtle 4`.

**Q2.** Le moniteur de cette voiture apparaît. La mini-vue en haut du moniteur de l'agent reste centrée sur cet agent. Dézoomer la vue à l'aide du curseur situé sous la vue et observer cette tortue en appuyant sur le bouton Watch puis sur le bouton go dans l'onglet Interface.

**Q3.** Le moniteur de tortue affiche toutes ses variables et ses valeurs. Quel est le numéro de cette tortue ? De quelle couleur est cette tortue ? Quelle est la forme de cette tortue ?

**Q4.** Il y a deux autres façons d'ouvrir un moniteur de tortue.

**Q4.1.** Fermez le moniteur de tortue en cliquant sur la case dans le coin supérieur gauche (sous Mac) ou dans le coin supérieur droit (sous d'autres OS). Dans le menu Tools, sélectionnez Turtle Monitor. Saisissez le numéro de la tortue (e.g. 4) dans le champ who.

**Q4.2.** Fermez le moniteur de tortue et saisissez `inspect turtle 4` dans le centre de commande.

**Q5.** Il y a trois manières de changer la couleur d'une seule tortue.

**Q5.1.** En bas du turtle monitor, saisissez `set color pink` dans l'invite de commande de tortue. Que se passe-t-il dans la vue ? Est-ce que quelque chose a changé dans le turtle monitor ?

**Q5.2.** Saisissez la valeur green pour la variable couleur dans le moniteur de tortue. Que s'est-il passé ?

2. <https://ccl.northwestern.edu/netlogo/docs/programming.html#colors>

**Q5.3.** Dans le centre de commande, adressez à l'observateur la commande `ask turtle 4 [set color magenta]`. Que se passe-t-il ?

**Q6.** Contrairement aux tortues, les tuiles n'ont pas de numéro d'identifiant (who) mais ils sont situés sur un système de coordonnées.

**Q6.1.** Faites un clic droit sur la voiture rouge.

**Q6.2.** Dans le sous-menu contextuel, sélectionnez `inspect patch 0 Y` où Y est l'ordonnée de la tuile où se trouve cette voiture.

**Q6.3.** En bas du patch monitor, saisissez `set pcolor red` dans l'invite de commande de tuile. Que se passe-t-il dans la vue ? Est-ce que quelque chose a changé dans le patch monitor ?

**Q6.4.** Dans le centre de commande, adressez à l'observateur la commande `ask patch 0 Y [set pcolor yellow]`. Que se passe-t-il ?

## Procédures

Dans cette partie, vous allez découvrir l'onglet Code. On distingue 4 types d'agents (*turtles*, *patches*, *links* et *observer*). Les tuiles (*patches*) sont stationnaires et disposées dans une grille sur lesquelles les tortues (*turtles*) se déplacent. Les liens (*links*) sont les connexions entre les tortues. L'observateur (*observer*) est un agent qui supervise le modèle. Ces agents peuvent exécuter des commandes ou des procédures, i.e. des séquences de commandes.

Vous allez construire un modèle d'écosystème simple inspiré du modèle de prédatation de l'exercice n°3 en codant des procédures, en créant des moniteurs, des curseurs et des graphiques mais également en codant des procédures pour que les tortues se déplacent, mangent et se reproduisent.

## Exercice 12 : Crédit du bouton de configuration

**Q1.** Dans le menu File, créez un nouveau modèle (New).

**Q2.** Dans l'onglet Interface, créez un bouton de configuration.

**Q2.1.** Cliquez sur l'icône Add dans la barre d'outils en haut de l'onglet Interface.

**Q2.2.** Dans le menu situé à côté de Add, sélectionnez Button (s'il n'est pas déjà sélectionné).

**Q2.3.** Cliquez à l'endroit où vous voulez que le bouton apparaisse dans la zone vide de l'onglet Interface.

**Q2.4.** Une boîte de dialogue permet de modifier le bouton. Saisissez `setup` dans le champ intitulé Commands.

**Q2.5.** Appuyez sur le bouton OK pour fermer la boîte de dialogue.

**Q3.** La procédure setup est exécutée en appuyant sur le bouton setup. Une procédure est une séquence de commandes munie d'un nom. Comme vous n'avez pas encore défini cette procédure, le bouton est rouge.

**Q3.1.** Cliquez sur ce bouton. Que se passe-t-il ?

**Q4.** Vous allez créer la procédure setup.

**Q4.1.** Dans l'onglet Code, copiez le code de la procédure setup.

```
to setup
  clear-all
  create-turtles 100
  [ setxy random-xcor random-ycor ]
  reset-ticks
end
```

;; Définir une procédure intitulée setup  
;; Réinitialiser l'environnement  
;; Créer 100 tortues sur la tuile 0,0  
;; Déplacer aléatoirement chaque tortue  
;; Redémarrer l'horloge  
;; Terminer la définition de la procédure

**Q4.2.** Indenter le code n'est pas obligatoire mais cela rend le code plus facile à lire et à modifier. Les primitives `random-xcor` et `random-ycor` sont des rapporteurs (reporter) qui, contrairement à une commande, retourne une valeur : ici des coordonnées choisies de manière aléatoire. Dans l'onglet Interface, appuyez sur le bouton setup à plusieurs reprises. Qu'observez-vous dans la vue ?

## **Exercice 13 : Mise à jour à chaque tic d'horloge**

Comme vous utilisez l'horloge (avec `reset-ticks`), vous devez configurer NetLogo pour mettre à jour la vue à chaque tic plutôt que continuellement.

**Q1.** Dans la barre d'outils de l'onglet Interface, activez la mise à jour de la vue (view updates) à chaque tic d'horloge (on ticks) plutôt qu'en continue (continuous).

## **Exercice 14 : Création du bouton go**

**Q1.** De la même manière que vous avez créé le bouton setup, vous devez créer un bouton go.

**Q1.1.** Cliquez sur l'icône Add dans la barre d'outils en haut de l'onglet Interface.

**Q1.2.** Dans le menu situé à côté de Add, sélectionnez Button (s'il n'est pas déjà sélectionné).

**Q1.3.** Cliquez à l'endroit où vous voulez que le bouton apparaisse dans la zone vide de l'onglet Interface.

**Q1.4.** Une boîte de dialogue permet de modifier le bouton. Saisissez `go` dans le champ intitulé Commands.

**Q1.5.** Cochez la case forever dans la boîte de dialogue pour faire en sorte que le bouton reste enfoncé une fois qu'il a été pressé et que les commandes s'exécutent en boucle.

**Q1.6.** Cochez également la case disable until ticks start pour vous empêcher d'appuyer sur le bouton go avant la configuration.

**Q1.7.** Appuyez sur le bouton OK pour fermer la boîte de dialogue.

**Q2.** Dans l'onglet Code, ajoutez les deux procédures suivantes :

```
to go
  move-turtles
  tick
end

to move-turtles
  ask turtles [
    right random 360
    forward 1
  ]
end
```

;; Appeler la procédure définie ci-dessous  
;; Incrémenter le tic d'horloge  
;; Demander aux tortues d'exécuter les commandes  
;; Tourner aléatoirement  
;; Avancer d'un pas

**Q3.** Après avoir appuyé une fois sur setup pour créer les tortues, appuyez sur le bouton go. Observez la vue. Désactivez le bouton go. Que se passe-t-il ?

**Q4.** Ralentissez la simulation. Que se passe-t-il si une tortue sort de l'environnement ?

## **Exercice 15 : Testez les commandes**

**Q1.** Vous pouvez interagir avec le modèle au cours de la simulation.

**Q1.1.** Dans le centre de commande, sélectionnez turtles et envoyez la commande `pen-down`.

**Q2.** Vous pouvez modifier le modèle pour une autre simulation.

**Q2.1.** Modifiez la procédure move-turtles pour que les tortues tournent aléatoirement à gauche de 45°.

## **Exercice 16 : Variables de tuile**

Dans cet exercice, vous allez ajouter des tuiles et les configurer.

**Q1.** Dans l'onglet Code, modifiez les procédures de configuration de la manière suivante :

```

to setup                                ;; Définir une procédure intitulée setup
  clear-all                            ;; Réinitialiser l'environnement
  setup-patches                        ;; Initialiser les tuiles
  setup-turtles                         ;; Initialiser des tortues
  reset-ticks                           ;; Redémarrer l'horloge
end

to setup-turtles                         ;; Procédure d'initialisation des tortues
  create-turtles 100                  ;; Créer 100 tortues sur la tuile 0,0
  [ setxy random-xcor random-ycor ]   ;; Déplacer aléatoirement chaque tortue
end

to setup-patches                         ;; Procédure d'initialisation des tuiles
  ask patches                          ;; Demander aux tuiles
  [ set pcolor green ]                ;; de prendre la couleur verte
end

```

**Q2.** Retourner dans l'onglet Interface et appuyez sur le bouton setup.

### **Exercice 17 : Variables de tortue**

Dans cet exercice, vous allez faire interagir les tortues et les tuiles.

**Q1.** Pour que les tortues mangent de l'herbe, qui repoussera progressivement, se reproduisent et meurent, vous allez ajouter une nouvelle variable de tortue.

**Q1.1.** Ajoutez dans l'en-tête de l'onglet Code, la déclaration de la variable de tortue `turtles-own [energie]`.

**Q1.2.** Ré-écrivez la procédure de simulation de la manière suivante :

```

to go
  move-turtles                      ;; Se déplacer
  eat-grass                           ;; Manger l'herbe
  tick                                ;; Incrémenter le tic d'horloge
end

to eat-grass
  ask turtles [                      ;; Demander aux tortues
    if pcolor = green [              ;; s'il y a de l'herbe
      set pcolor black               ;; la supprimer
      set energie energie + 10       ;; se revigorer
    ]
  ]
end

to move-turtles
  ask turtles [                      ;; Demander aux tortues de
    left random 45                  ;; tourner aléatoirement
    forward 1                       ;; avancer d'un pas
    set energie energie - 1         ;; se fatiguer
  ]
end

```

**Q2.** Retourner dans l'onglet Interface et appuyez sur le bouton go. Que se passe-t-il ?

### **Exercice 18 : Monitoring**

**Q1.** Vous allez créer deux moniteurs dans l'onglet *Interface* grâce à la barre d'outils.

**Q2.** Créez un moniteur « mouton ».

**Q2.1.** Dans la barre d'outils de l'onglet Interface, sélectionner Monitor et appuyer sur l'icône Add. Cliquez à l'endroit où vous voulez que le moniteur apparaisse dans la zone vide de l'onglet Interface.

**Q2.2.** Dans la boîte de dialogue, saisissez **count turtles** dans le champ reporter et intitulez ce moniteur « mouton ». Appuyez sur le bouton OK pour fermer la boîte de dialogue.

**Q3.** Créez un moniteur « herbe ».

**Q3.1.** Dans la boîte de dialogue, saisissez **count patches with [pcolor = green]** dans le champ reporter et intitulez ce moniteur « herbe ». Appuyez sur le bouton OK pour fermer la boîte de dialogue.

**Q4.** Appuyez sur les boutons setup et go et observez les moniteurs.

## **Exercice 19 : Interrupteurs et étiquettes**

Afin d'observer l'évolution de l'énergie de chaque tortue, vous allez ajouter un interrupteur pour pouvoir activer/désactiver la visualisation de ces informations.

**Q1.** Dans la barre d'outils de l'onglet Interface, sélectionner Switch et appuyer sur l'icône Add. Cliquez à l'endroit où vous voulez que l'interrupteur apparaisse dans la zone vide de l'onglet Interface.

**Q2.** Dans la boîte de dialogue, saisissez **montre-energie?** dans le champ Global Variable.

**Q3.** Retournez dans l'onglet Code pour réécrire la procédure eat-grass de la manière suivante :

```
to eat-grass
  ask turtles [
    if pcolor = green [
      set pcolor black
      set energie energie + 10
    ]
    ifelse montre-energie?
      [ set label energie ]
      [ set label "" ]
  ]
end
```

**Q4.** Exécutez la simulation en utilisant les boutons setup et go en activant/désactivant l'interrupteur. Que se passe-t-il ?

## **Exercice 20 : Amélioration des comportements**

Vous allez ajouter 3 comportements pour que les moutons se reproduisent, les moutons meurent et l'herbe repousse.

**Q1.** Retournez dans l'onglet Code pour réécrire la procédure de simulation de la manière suivante en ajoutant 3 comportements :

```
to go
  move-turtles
  eat-grass
  reproduce
  check-death
  regrow-grass
  tick

to reproduce
  ask turtles [
    if energie > 50 [
      ;;
    ]
  ]
  ;;

to move-turtles
  ;;

to eat-grass
  ;;

to reproduce
  ;;

to check-death
  ;;

to regrow-grass
  ;;

to tick
  ;;
```

```

    set energie energie - 50          ;; de s'épuiser
    hatch 1 [ set energie 50 ]       ;; à se reproduire
]
]
end

to check-death                      ;; Les moutons meurent
ask turtles [                         ;; Demander aux moutons
  if energie <= 0 [ die ]           ;; qui sont épuisés de mourir
]
end

to regrow-grass                     ;; L'herbe repousser
ask patches [                         ;; demand aux tuiles
  if random 100 < 3                ;; de manière aléatoire
  [ set pcolor green ]             ;; de se régénérer
]
end

```

**Q2.** Exécutez la simulation en utilisant les boutons setup et go. Que se passe-t-il?

### **Exercice 21 : Tracé**

Vous allez créer un tracé dans l'onglet Interface et y placer quelques commandes.

Ces commandes s'exécuteront automatiquement quand les instructions `reset-ticks` et `tick` seront exécutées.

**Q1.** Dans la barre d'outils de l'onglet Interface, sélectionner Plot et appuyer sur l'icône Add. Cliquez à l'endroit où vous voulez que le tracé apparaisse dans la zone vide de l'onglet Interface.

**Q2.** Dans la boîte de dialogue, les valeurs pour les champs sont :

**Q2.1.** Total pour Name;

**Q2.2.** temps pour X axis label;

**Q2.3.** totaux pour Y axis label;

**Q2.4.** moutons pour Pen Name;

**Q2.5.** `plot count turtles` pour Pen Update Commands

**Q3.** Appuyez sur le bouton Add Pen.

**Q3.1.** La valeur pour le champs Pen Name est herbe.

**Q3.2.** La valeur pour le champs Pen Update Commands est `plot count patches with [pcolor = green]`.

**Q4.** Modifiez la couleur des tracés.

**Q5.** Appuyez sur OK dans la boîte de dialogue Plot.

**Q6.** Configurez et exécutez à nouveau la simulation. Que se passe-t-il?

### **Exercice 22 : Compteur de tics**

Pour comparer les tracés d'une simulation à une autre, il est utile d'avoir toujours la même durée d'exécution du modèle.

**Q1.** Pour fixer une limite à la durée totale d'une exécution, modifiez ainsi la procédure de simulation :

```

to go                                ;; Procédure de simulation
  if ticks >= 100 [ stop ]            ;; Arrêter après 100 tics
  move-turtles                        ;; Les moutons se déplacent
]

```

```
eat-grass          ;; Les moutons mangent l'herbe
reproduce         ;; Les moutons se reproduisent
check-death       ;; Les moutons meurent
regrow-grass      ;; L'herbe repousse
tick              ;; Le tic d'horloge est incrémenté
end
```

**Q2.** Configurez et simulez le modèle.

### **Exercice 23 : Quelques détails supplémentaires**

**Q1.** Vous allez faire varier le nombre initial de moutons.

**Q1.1.** Dans la barre d'outils de l'onglet Interface, sélectionner Slider et appuyer sur l'icône Add. Cliquez à l'endroit où vous voulez que le curseur apparaisse dans la zone vide de l'onglet Interface.

**Q1.2.** Intitulez ce curseur nombre. Modifiez les valeurs minimum et maximum.

**Q1.3.** Dans la procédure setup-turtles, remplacez `create-turtles 100` par `create-turtles nombre`.

**Q2.** Vous allez ajuster l'énergie que les moutons gagnent lorsqu'ils mangent de l'herbe et qu'ils perdent quand ils se reproduisent.

**Q2.1.** Créez un curseur appelé energie-herbe.

**Q2.2.** Créez un curseur appelé energie-reproduction.

**Q2.3.** Modifiez les procédures reproduce et eat-grass de la manière suivante :

```
to reproduce                                ;; Les moutons se reproduisent
  ask turtles [
    if energie > 50 [                         ;; Demander aux moutons
      set energie energie - energie-reproduction ;; qui sont vigoureux
      hatch 1 [ set energie energie-reproduction ] ;; de s'épuiser
    ]
  ]
end

to eat-grass
  ask turtles [                                ;; Demander aux tortues
    if pcolor = green [                        ;; s'il y a de l'herbe
      set pcolor black                         ;; la supprimer
      set energie (energie + energie-herbe) ;; se revigorier
    ]
    ifelse montre-energie?                    ;; Si l'interrupteur est activé
      [ set label energie ]                   ;; affichez le niveau d'énergie
      [ set label "" ]                       ;; sinon ne rien afficher
  ]
end
```

**Q3.** Configurez et simulez le modèle. Que se passe-t-il si on fait décroître l'énergie nécessaire à la reproduction ?

### **Exercice 24 : Conclusion**

Vous avez maintenant un modèle d'écosystème simplifié où l'herbe pousse dans des parcelles et des moutons se promènent, mangent l'herbe, se reproduisent et meurent. Vous avez créé une interface contenant des boutons, des curseurs, des interrupteurs, des moniteurs et des tracés. Vous avez également créer et modifier des variables et des procédures pour configurer le modèle, le simuler et doter les agents (tortues et tuiles) de caractéristiques et de comportements individuels. Bravo !

Le prochain TP présente le langage NetLogo en détail pour concevoir vos propres simulations multi-agents.