

Tutoriales NETLOGO

Diego Izquierdo

April 10, 2016

Abstract

Documento hecho con el fin de introducir al lector en un conjunto de tres tutoriales acerca del uso de NETLOGO

1 Introduccion

NetLogo es un entorno programable de modelado para simular fenómenos naturales y sociales. Es especialmente adecuado para modelar sistemas complejos que se desarrollan en el tiempo. Los modeladores pueden dar instrucciones a cientos o miles de "agentes" independientes todos operando en paralelo. Esto hace que sea posible explorar la relación entre el nivel micro del comportamiento de los individuos y los patrones a nivel macro que emergen de la interacción de muchos individuos.

2 Tutorial 1

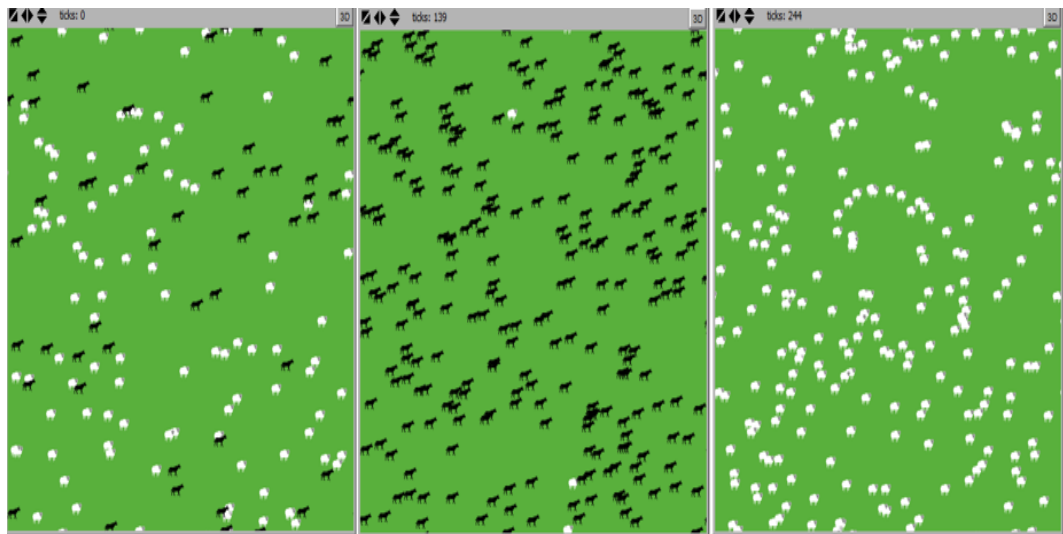
2.1 Modelo de muestra: depredación Lobo Oveja

Abriremos uno de los modelos muestra y lo exploraremos en detalle. Vamos a probar un modelo biológico: la depredación lobo oveja, un modelo poblacional predador-presa.

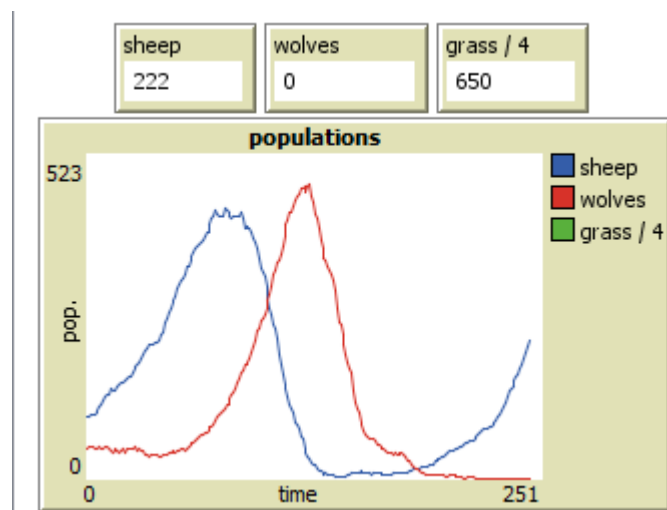
- Abra Models Library del menú File
- Elija "Wolf Sheep Predation" de la sección Biology y pulse "Open".

La interfaz estará llena un montón de botones, interruptores (switches), deslizadores (sliders) y monitores. Estos elementos de la interfaz le permiten interactuar con el modelo. Los botones son de color azul, ellos crean, inician, y detienen el modelo. Los deslizadores (sliders) y los interruptores (switches) son verdes; ellos alteran la configuración de modelo. Los monitores y parcelas son de color beige; ellos muestran los datos.

- Presione el botón "setup".
- Presione el botón "go" para iniciar la simulación. Presione el botón "go" para detener el modelo.



se puede notar que se realiza la simulación dando como resultado que en un principio hay igualdad entre el numero de ovejas y el de lobos, posteriormente los lobos se comen a las ovejas acabando con casi con toda la población de ovejas lo que causa que la población de lobos baje y finalmente se repobló la población de ovejas



la gráfica muestra la relación entre la población de lobos y ovejas. se pueden obtener resultados diferentes como por ejemplo la extinción de ambas especies.

This is an inline comment.

2.2 Ajustando la Configuración: Sliders y Switches

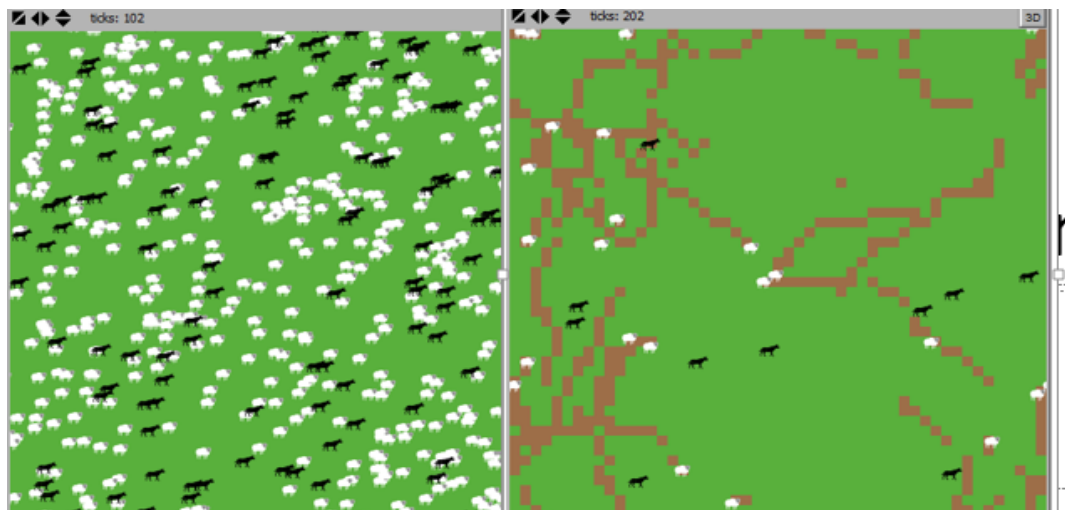
Los ajustes dentro de un modelo le dan una oportunidad de trabajar en diferentes escenarios o hipótesis. Modificando la configuración y corriendo luego el modelo para ver cómo reacciona a esos cambios puede darle una comprensión

más profunda del fenómeno que se está modelando. Los Switches (interruptores) y los Sliders (deslizadores) le dan acceso a la configuración del modelo.

Aquí están los interruptores y deslizadores del modelo de depredación lobo oveja:



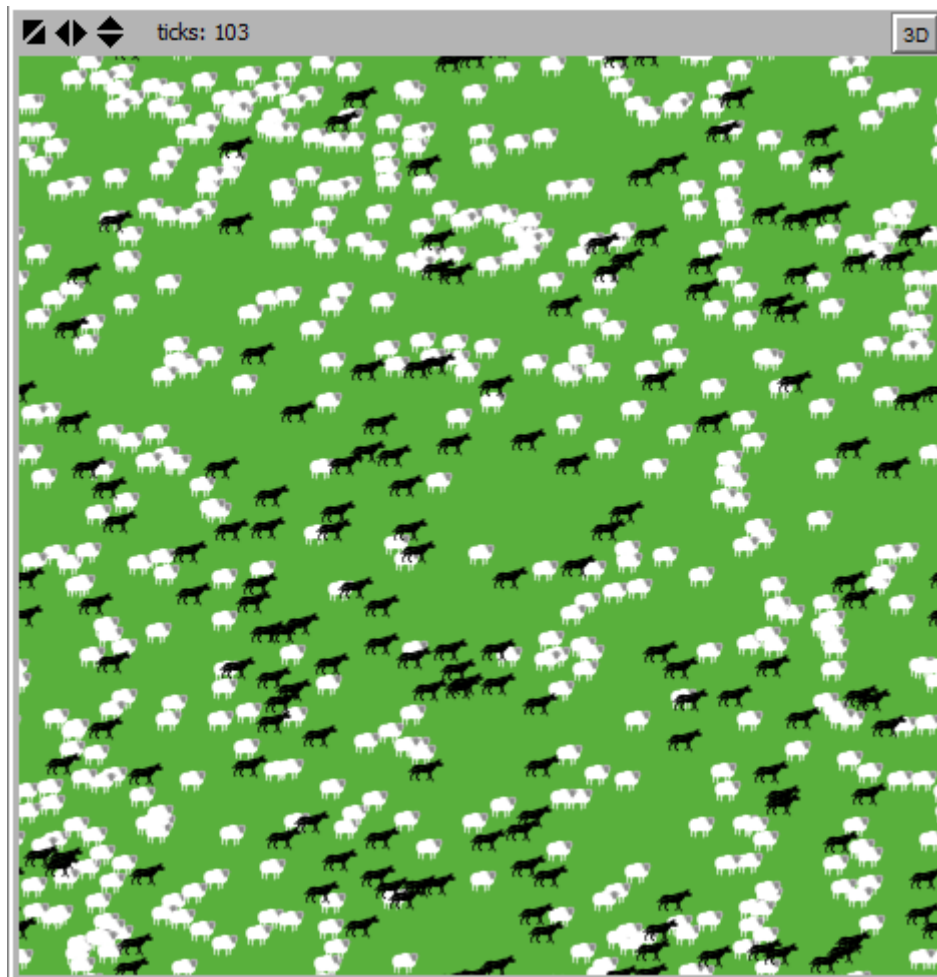
- Presione "setup" y "go" y deje que el modelo corra por aproximadamente 100 ticks de tiempo. (Nota: hay una lectura del número de ticks justo encima de la parcela)
- Detenga el modelo pulsando el botón "go".
- Encienda el switch de la hierba ("grass?").
- Presione "setup" y "go" y deje correr el modelo por una cantidad de tiempo similar al de la anterior.



al activar el switch grass por donde pasan las ovejas desaparece la hierva por lo que decrece la población de ovejas y lobos. para finalmente dejar solo ovejas

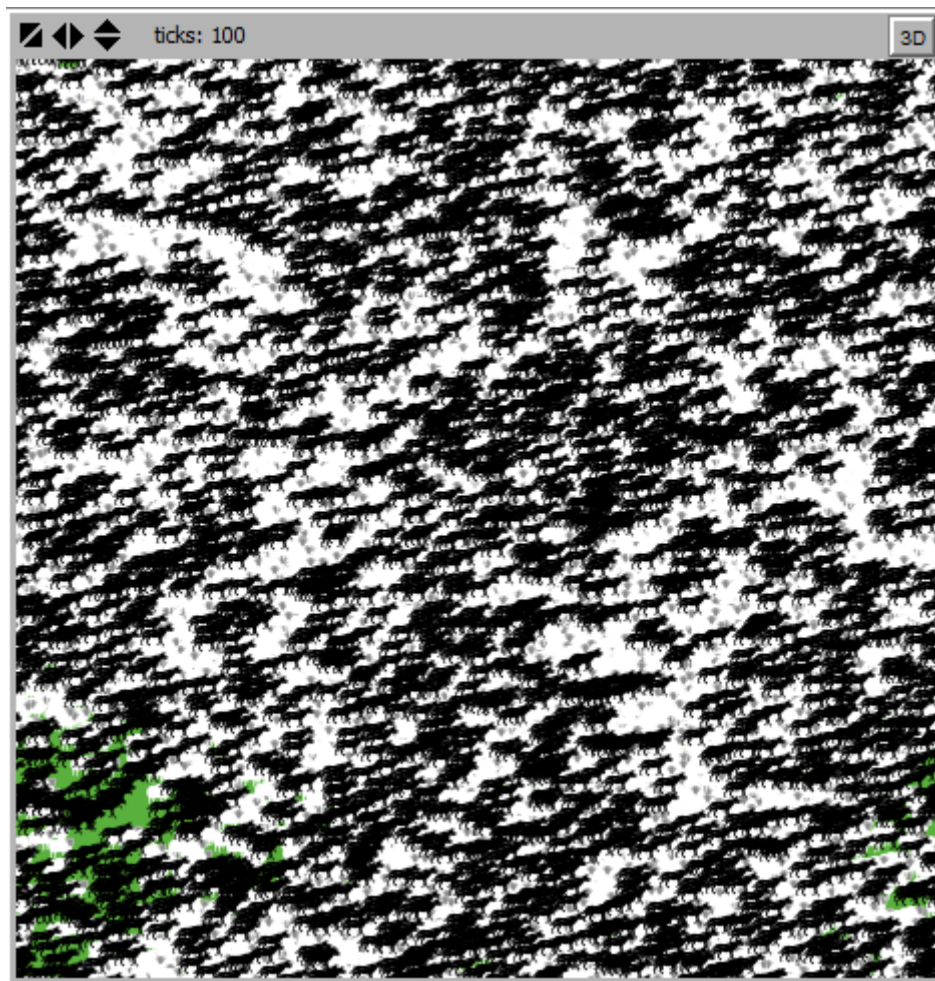
¿Qué sucedería con la población de ovejas si hay al comienzo de la simulación inician más ovejas y menos lobos?

- Apague "grass?".
- Establezca el slider del número inicial de ovejas" ("initial-number-sheep") a 100.
- Establezca el slider del número inicial de lobos ("initial-number-wolves") a 20.
- Presione "setup" y luego "go".
- Permita que el modelo corra alrededor de 100 ticks de tiempo.



al ser tan grande la población de ovejas en relaciona a la de lobos esta crece lo que genera que los lobos así mismo aumenten y se coman todas las ovejas

- Ajuste el número inicial de ovejas a 80 y el número inicial de lobos a 50. (Esto es cercano a la forma en que estaban cuando usted abrió el modelo por primera vez.)
- Fije "sheep-reproduce" en 10,0
- Presione "setup" y luego "go".
- Permita que el modelo corra alrededor de 100 ticks de tiempo.



aunque la población de ovejas no es tan superior a la de lobos en el inicio debido a que se aumento la probabilidad de reproducción de las ovejas estas se multiplicaron exponencial mente y así mismo sucedió con los lobos quedando al final la extinción de ambas especies

2.3 Recopilación de información: Parcelas y Monitores

Un propósito del modelado es reunir datos sobre una materia o tema que serían muy difíciles de obtener en condiciones de laboratorio. NetLogo tiene dos formas principales de mostrarle los datos al usuario: las gráficas y los monitores

GRAFICAS

La gráfica en el modelo depredación lobo Oveja contiene tres líneas: ovejas, lobos, y hierba / 4. (El conteo del pasto se encuentra dividido entre cuatro y esto hace que la gráfica no sea demasiado alta.) Las líneas muestran lo que le está sucediendo al modelo a través del tiempo. Para ver cuál línea es cuál, haga clic en "Pens" ("curvas") en la esquina superior derecha de la ventana de la gráfica para abrir la leyenda de curvas de la gráfica. Aparece una clave que indica a cual línea está graficando. En este caso es el conteo de la población.

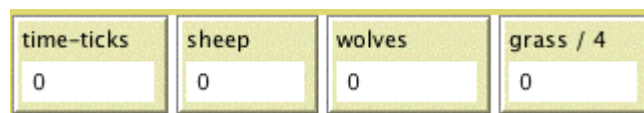
Cuando una gráfica está cerca de ser llenada, el eje horizontal aumenta de tamaño y todos los datos de antes se comprimen en un menor espacio. De esta manera, se hace más espacio para que la gráfica crezca.

Si desea guardar los datos de una gráfica para verlos o analizarlos en otro programa, usted puede utilizar el item "Export Plot" en el menú File. De esa manera guarda la información en su computador en un formato que pueda ser leído por una hoja de cálculo como Excel y por programas de bases de datos. También puede exportar una gráfica con control-clic (Mac) o haciendo clic derecho (Windows) y eligiendo "Export ..." en el menú emergente.

MONITORES

El monitor marcado con "time-ticks" nos dice cuánto tiempo ha transcurrido en el modelo. Los otros monitores nos muestran las poblaciones de ovejas y lobos, y la cantidad de hierba. (Recuerde, la cantidad de hierba está dividida entre cuatro para evitar que la gráfica sea demasiado alta.) Los números desplegados en los monitores se actualizan continuamente a medida que corre el modelo, mientras que las gráficas le muestran los datos en conjunto de todo el curso del modelo corrido.

Los monitores son otro método de visualización de la información en un modelo. Aquí están los monitores del modelo de depredación Lobo Oveja:



2.4 Control de la Vista

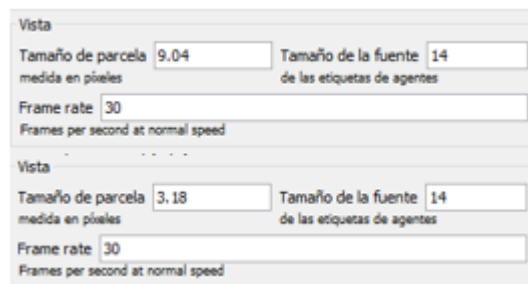
Cuando la actualización de la vista está completamente apagada el modelo sigue corriendo en segundo plano, y las parcelas y monitores se siguen actualizando; pero si quiere ver lo que está sucediendo, debe reactivar la actualización de las vistas marcando nuevamente sobre la casilla de verificación. La mayoría de los modelos corre mucho más rápido cuando las actualizaciones de las vista están apagadas.

El tamaño de la vista está determinado por cinco valores separados: Min y Max X, Min y Max Y y tamaño del parche. Echemos un vistazo a lo que sucede cuando cambiamos el tamaño de la vista en el "modelo de depredación lobo oveja".

Hay más ajustes del mundo y de la vista de lo que hay espacio para la barra de herramientas. El botón de "Settings..." le permite llegar al resto de ajustes.

- Pulse el botón "Settings ..." en la barra de herramientas.
- Pulse "cancel" para hacer que esta ventana desaparezca sin cambiar la configuración.

- Coloque el puntero del ratón al lado, pero fuera, de la vista.
- Manténgase pulsado el botón del ratón y arrastre el puntero sobre la vista.
- Arrastre una de las "asas" cuadradas negras. Las asas se encuentran en los bordes y en las esquinas de la vista.
- Deseleccione la vista haciendo clic en cualquier lugar del fondo blanco de la Interfaz.
- Pulse de nuevo el botón "Settings..." y vea los ajustes.

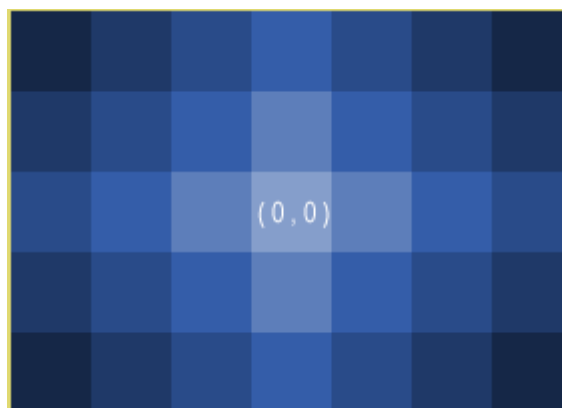


al hacer las modificaciones el tamaño de la parcela cambia

El mundo NetLogo es una cuadrícula bidimensional de "parches". Los parches son los cuadrados individuales en la cuadrícula.

En el modelo de depredación Lobo Oveja, cuando el switch "grass?" está encendido los parches individuales son fácilmente distinguibles, porque algunos de ellos son verdes, mientras que otros son de color marrón.

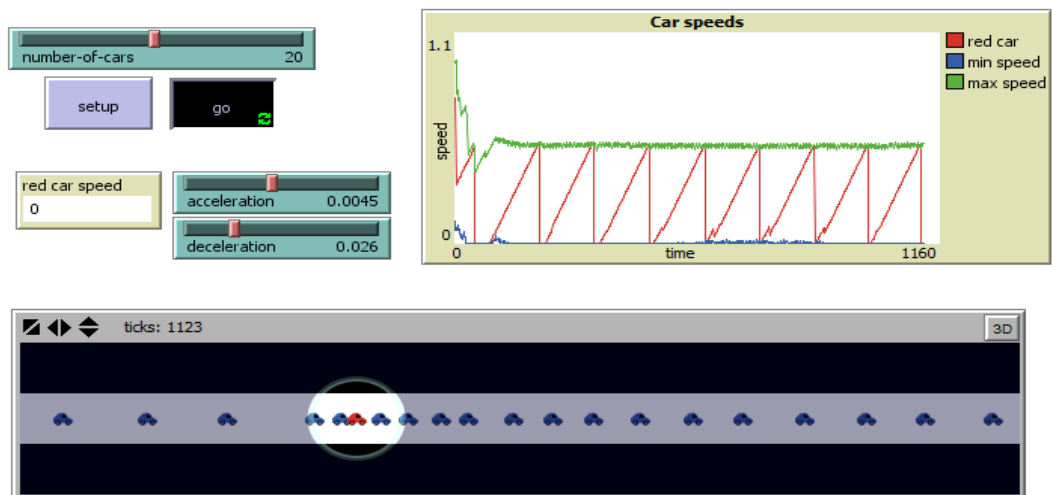
Piense en los parches como las baldosas cuadradas de una habitación que tiene piso de baldosa. De manera predeterminada, exactamente en el centro de la sala existe una baldosa marcada con (0,0); lo que significa que si la sala está dividida a la mitad en un sentido y a la mitad en el otro, y que las dos líneas divisorias se interceptan en esta baldosa. Ahora tenemos un sistema de coordenadas que nos ayudará a localizar objetos dentro de la habitación:



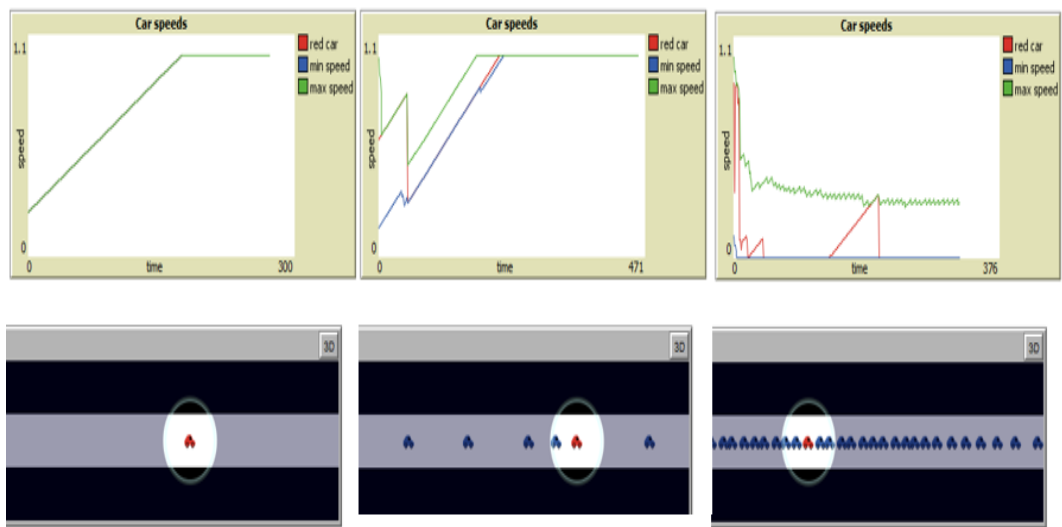
3 Tutorial 2

3.1 Comandos, Modelo de Muestra: Tráfico Básico

- Vaya a la Librería de Modelos (Models Library) en el menú de Archivo (File menu) .
- Abra tráfico básico (Traffic Basic), que se encuentra en la sección "Ciencias Sociales" ("Social Science").
- Ejecute el modelo por un par de minutos para familiarizarse con él.
- Consulte la ficha de información para cualquier pregunta que tenga acerca de este modelo. Consulte la ficha de información para cualquier pregunta que tenga acerca de este modelo.



En este modelo usted notará un automóvil rojo en un flujo de coches azules. En el flujo de automóviles todos se están desplazando en la misma dirección. De vez en cuando se "apilan" y dejan de moverse. Este modela la manera como pueden formarse atascos en el tráfico sin que exista alguna causa tal como un accidente, un puente roto, o el vuelco de un camión. No es necesaria una "causa central" para que se forme un embotellamiento de tráfico.

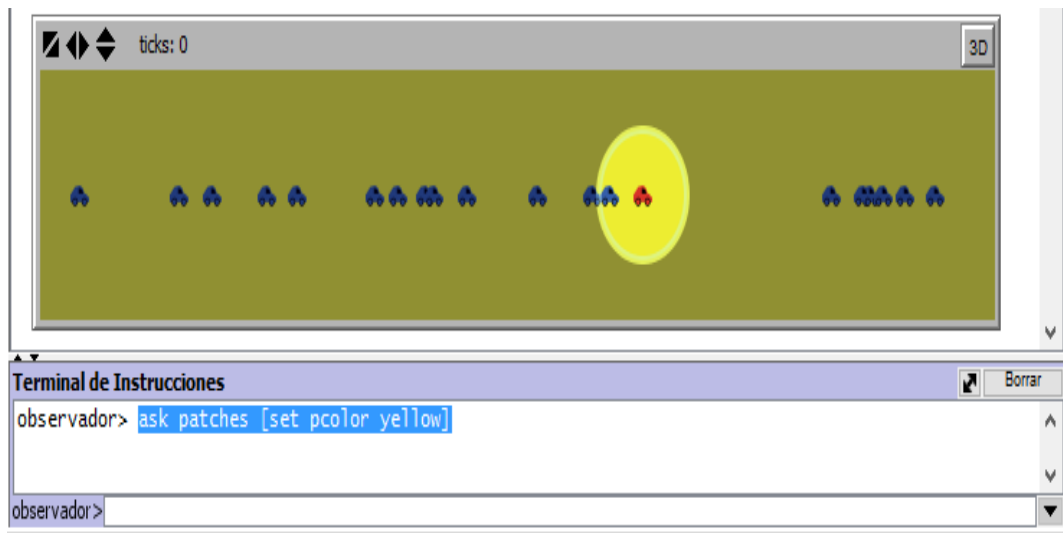


iniciando la simulación con solo un auto se puede ver que como no se detiene su velocidad aumenta hasta el punto de llegar a su velocidad máxima donde se mantiene constante. con 10 autos el carro rojo casi no se detiene lo que hace que tenga una gran velocidad la mayoría del tiempo. con 41 es muy poca la velocidad que el auto rojo puede alcanzar y se atasca muy seguido

3.2 El Centro de Comando

El Centro de Comando se encuentra en la ficha (Tab) Interface y le permite introducir comandos o instrucciones al modelo. Los comandos son instrucciones que usted puede dar a los agentes de NetLogo: tortugas, parches, enlaces y el observador.

- En el tráfico básico:
 Presione el botón "setup".
 Busque el Centro de Comando.
 Haga clic con el ratón en el cuadro blanco en la parte inferior del Centro de Comando.
 Escriba el texto que se muestra aquí: `ask patches [set pcolor yellow]`
- Pulse la tecla de retorno.

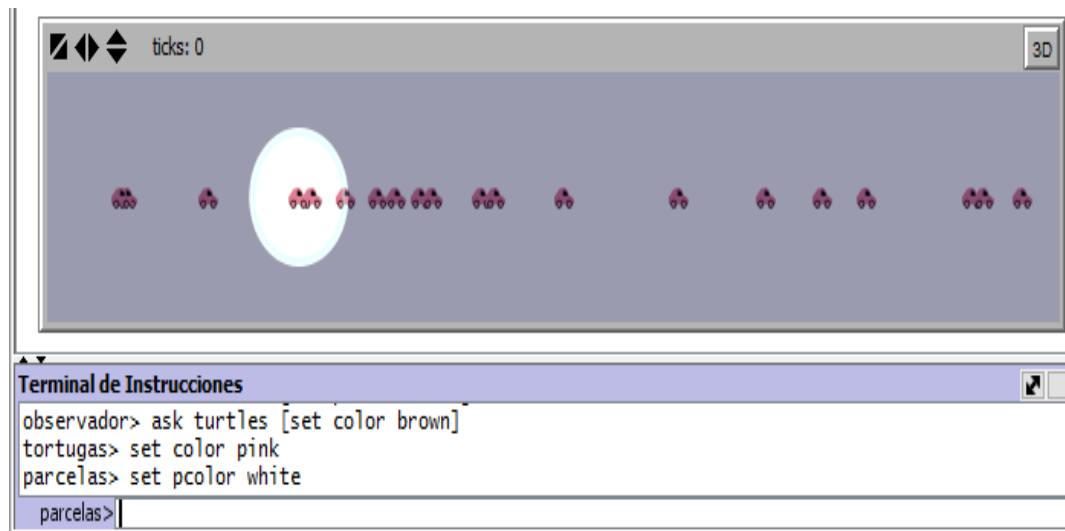


Debí haber notado que el fondo de la vista se ha vuelto completamente amarillo y que la calle ha desaparecido.

Mirando el comando que fue escrito, sólo le pedimos a los parches cambiar su color. En este modelo, los coches están representados por un tipo diferente de agente, llamados "turtles" (tortugas). Por lo tanto, los coches no recibieron estas instrucciones y por ese motivo no cambiaron.

Usted debió haber notado que el comando que acaba de escribir ahora está desplegado en el cuadro blanco en medio del Centro de Comandos.

- En el Centro de Comando, haga clic en el "observer;" en la esquina inferior izquierda
- Elija "turtles" ("tortugas") en el menú emergente.
- Escriba set color pink y pulse retorno.
- Pulse la tecla de tabulación hasta que vea "patches;" en la esquina inferior izquierda.
- Escriba set pcolor white y pulse retorno.

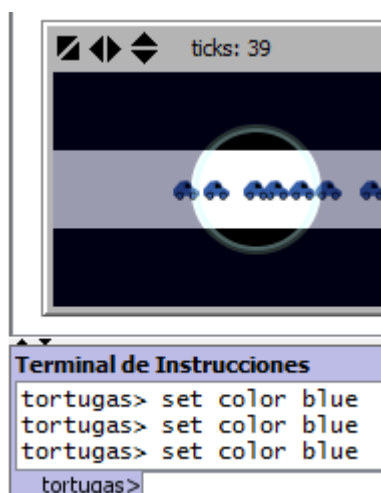


El observador supervisa el mundo y, por tanto, puede dar un comando a los parches o las tortugas utilizando ask. Al igual que en el primer ejemplo (observer; ask patches [set pcolor yellow]), el observador tiene que pedirle a los parches que fijen su pcolor en amarillo. Pero cuando un comando está dado directamente a un grupo de agentes al igual que en el segundo ejemplo (patches; set pcolor white), usted sólo tiene que dar el comando en sí.

3.3 Trabajando con colores

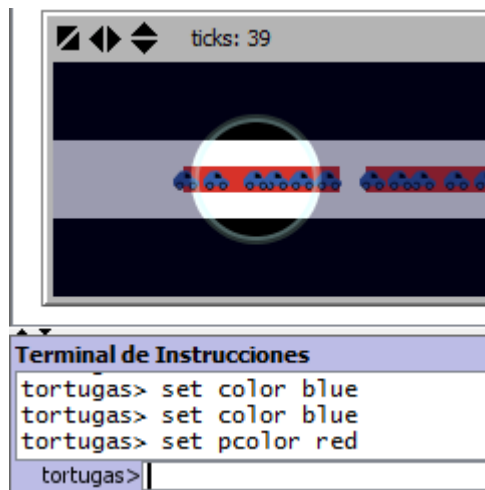
¿Cuál es la diferencia entre el color y pcolor?

- Elija "turtles" en el menú desplegable del Centro de Comando (o utilice la tecla de tabulación).
- Escriba set color blue y pulse retorno.



la tortuga que normalmente es roja se pone azul

- En su lugar escriba `set pcolor red` y pulse retorno.



Las llamamos "variables" `color` y `pcolor`. Algunos comandos y variables son específicos para las tortugas y algunos otros son específicos para los parches. Por ejemplo, la variable `color` es una variable de tortuga, mientras que `pcolor` es una variable de parche.

Continúe la práctica modificando los colores de las tortugas y de los parches utilizando el comando `set` y estas dos variables.

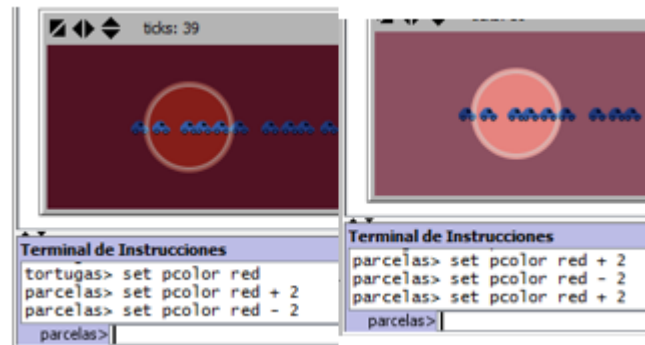
Para poder realizar más cambios en los colores de las tortugas y los parches, o tendremos que decir los automóviles y los fondos, necesitamos comprender un poco la manera cómo NetLogo trata a los colores.

En NetLogo, todos los colores tienen un valor numérico. En todos los ejercicios hemos estado utilizando el nombre del color. Esto se debe a que NetLogo reconoce 16 diferentes nombres de colores. Esto no significa que NetLogo sólo reconozca 16 colores. Entre estos colores existen muchos matices que también pueden ser utilizados. Aquí hay un gráfico que muestra todo el espacio de colores de NetLogo:

	black = 0										white = 9.9
gray = 5	0	1	2	3	4	5	6	7	8	9	9.9
red = 15	10	11	12	13	14	15	16	17	18	19	19.9
orange = 25	20	21	22	23	24	25	26	27	28	29	29.9
brown = 35	30	31	32	33	34	35	36	37	38	39	39.9
yellow = 45	40	41	42	43	44	45	46	47	48	49	49.9
green = 55	50	51	52	53	54	55	56	57	58	59	59.9
lime = 65	60	61	62	63	64	65	66	67	68	69	69.9
turquoise = 75	70	71	72	73	74	75	76	77	78	79	79.9
cyan = 85	80	81	82	83	84	85	86	87	88	89	89.9
sky = 95	90	91	92	93	94	95	96	97	98	99	99.9
blue = 105	100	101	102	103	104	105	106	107	108	109	109.9
violet = 115	110	111	112	113	114	115	116	117	118	119	119.9
magenta = 125	120	121	122	123	124	125	126	127	128	129	129.9
pink = 135	130	131	132	133	134	135	136	137	138	139	139.9

Para obtener un color que no tiene su propio nombre simplemente refiérase a él mediante un número, o añadiendo o restando un número a un nombre. Por ejemplo, escribir set color red, es la misma cosa que escribir set color 15. Usted puede conseguir una versión más clara o más oscura del mismo color usando, de la siguiente manera, un número que sea un poco mayor o un poco más pequeño.

- Elija "patches" en el menú desplegable en el Centro de Comando (o utilice la tecla de tabulación).
- Escriba set pcolor rojo - 2 (El espacio en torno a la "-" es importante.)
- Escriba set pcolor rojo + 2

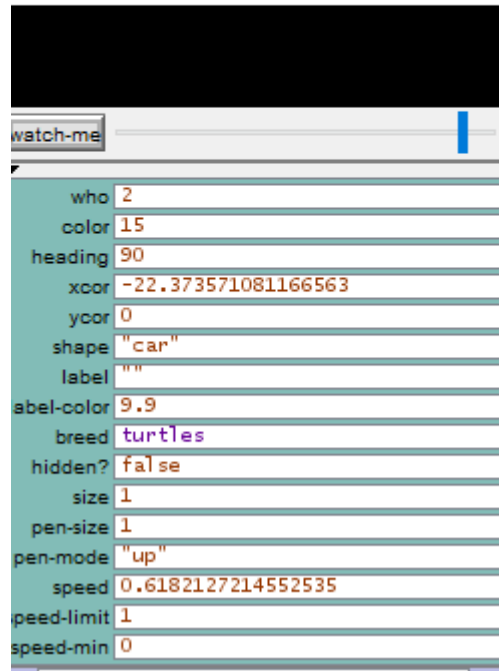


Restando de rojo, usted lo hace más oscuro. Sumando a rojo usted lo vuelve mas claro. Usted puede utilizar esta técnica en cualquiera de los colores que aparecen en la tabla.

3.4 Monitores de Agente y Comandantes de Agente

- Presione "setup" para que reaparezca el coche rojo.
- Si usted está en un equipo Macintosh, mantenga pulsada la tecla Control y haga clic en el coche rojo. En otros sistemas operativos, haga clic sobre el coche rojo con el botón derecho del ratón.
- Si hay otra tortuga cerca de la tortuga de color rojo verá más de una tortuga listada en la parte inferior del menú. Mueva el ratón encima de las tortugas seleccionadas, observe que cuando el ratón resalta un elemento del menú de la tortuga, la tortuga también es resaltada en la vista. Para la tortuga de color rojo seleccione del sub-menú la opción "inspect turtle" ("inspeccionar las tortuga").

Aparecerá un monitor de tortuga para ese coche:



Dando una mirada más cercana a este monitor de tortuga podremos ver todas las variables que pertenecen al coche rojo. Una variable es un lugar que tiene un valor que puede ser cambiado. ¿Recuerda cuando se mencionó que todos los colores están representados en el computador como números? Lo mismo es cierto en el caso de los agentes. Por ejemplo, cada tortuga tiene un número de identificación al que llamamos su "who number".

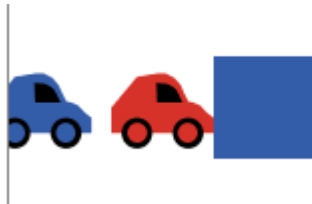
Hay otras dos maneras de abrir un monitor de tortuga además de pulsar el botón derecho del ratón (o control-clic, dependiendo de su sistema operativo). Una forma es elegir el "Turtle Monitor" en el menú de Herramientas (Tools) y a continuación escribir el who number de la tortuga que desea inspeccionar en "who" y luego pulsar sobre la tecla retorno. La otra forma es escribiendo inspect turtle 0 (o cualquier otro who number) en el Centro de Comando.

Una forma es utilizando el cuadro llamado Agent Commander (Comandante de Agente) que se encuentra en la parte inferior de un Monitor de Agente (Agent Monitor). Usted escribe aquí los comandos justamente igual a como lo haría en el Centro de Comando, pero los comandos que escriba aquí sólo aplican para esta tortuga en particular.

- En el Comandante de Agente del monitor de turtle escriba set color pink para la tortuga 0.



- Abra un monitor del parche para cualquier parche.
- En la parte inferior del monitor del parche ingrese `set pcolor blue` y pulse retorno.



cambia el color de la parcela a azul

- En el Centro de Comando, escriba `ask patch -11 -4 [set pcolor green]` y pulse retorno.

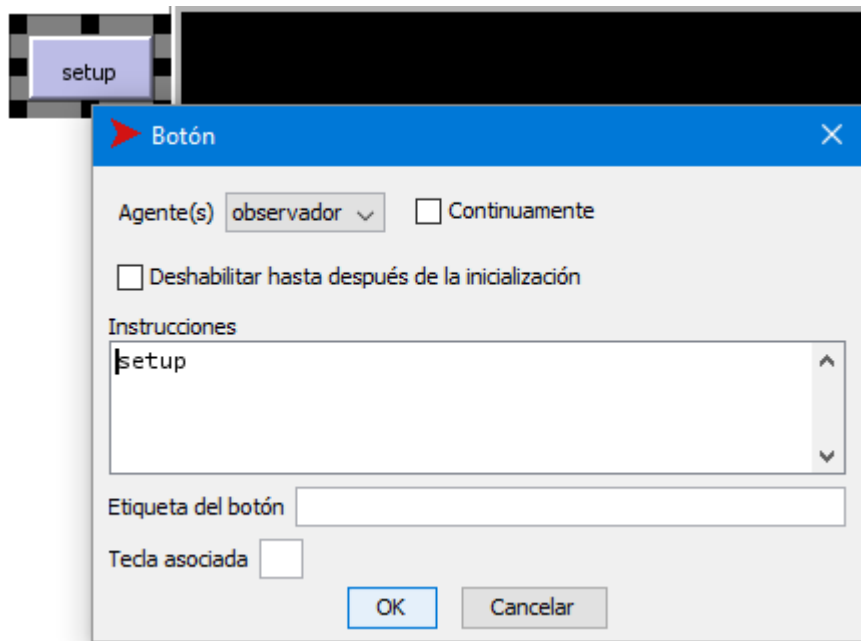


cambia el color de la parcela a verde

4 Tutorial 3: Procedimientos

4.1 Haciendo el botón setup

- Haga clic en el ícono "Button" en la parte superior de la ficha de la interfaz.
- Haga clic en donde usted desea que aparezca el botón dentro del área blanca vacía de la interfaz.
- Se abre un cuadro de diálogo para editar el botón. Escriba `setup` en la casilla marcada con "Commands".
- Pulse el botón OK cuando haya terminado; el cuadro de diálogo se cierra.



Ahora tiene un botón setup. Presionando el botón se ejecuta un procedure (procedimiento) llamado "setup". Un procedimiento es una secuencia de comandos NetLogo al que asignamos un nuevo nombre. No hemos definido aun ese procedimiento (lo haremos pronto). En el botón se pondrán las letras en rojo debido a que este botón se refiere a un procedimiento que todavía no existe. Ahora vamos a crear el procedimiento "setup"

- Cambiase a la pestaña Procedures.
- Escriba lo siguiente:

```

to setup
  clear-all
  create-turtles 100
  ask turtles[ setxy random-xcor random-ycor ]
end

```

Tenga en cuenta que las líneas tienen diferentes sangrías. La mayoría de las personas encuentra útil sangrar su código como en este, sin embargo no es obligatorio. Esto hace más fácil leer y cambiar el código. Su procedimiento inició con la palabra to y terminó con la palabra end. Cada nuevo procedimiento que cree comenzará y terminará con estas dos palabras.

- la configuración comienza definiendo un procedimiento llamado "setup".

- `clear-all` restablece el mundo al un estado inicial, un estado vacío. Todos los parches se vuelven negros y a su vez desaparece cualquier tortuga que usted pueda haber creado. Básicamente, limpia la pizarra para que corra un nuevo modelo.
- `create-turtles 100` crea 100 tortugas. Empiezan a partir del origen, es decir, del centro del parche 0,0.
- `ask turtles [...]` le dice a cada tortuga que corra, de manera independiente, los comandos dentro de los paréntesis. (Cada comando en NetLogo es corrido por un agente. `ask` también es un comando. Acá el observador está ejecutando el comando `ask` por sí solo, provocando a su vez que las tortugas ejecuten los comandos.)
- `setxy random-xcor random-ycor` es un comando que usa "reporteros". Un reportero, a diferencia de un comando, reporta un resultado. En primer lugar cada tortuga corre el `reporter random-xcor` el cual reportará un número dentro del rango permisible de las coordenadas de las tortugas a lo largo del eje de las X. Luego, cada tortuga corre el reportero `random-ycor` de igual forma para el eje Y. Por último cada tortuga corre el comando `setxy` usando a los anteriores dos números como entrada. Esto hace que la tortuga avance hasta el punto con esas coordenadas.
- `end` completa la definición del procedimiento "setup".

Cuando haya terminado de escribir cambie a la interfaz y presione el botón `setup` que hizo anteriormente. Verá las tortugas dispersas alrededor del mundo:



4.2 Haciendo el botón go

Ahora haga un botón llamado "go". Siga los mismos pasos que utilizó para hacer el botón setup, excepto:

- En Commands introduzca go en lugar de setup.
- Marque "forever" en la casilla de verificación del diálogo de edición.

La casilla de verificación "forever" ("para siempre") hace que el botón permanezca abajo una vez es pulsado, de manera que sus comandos corren y corren una y otra vez en lugar de hacerlo tan sólo una vez.

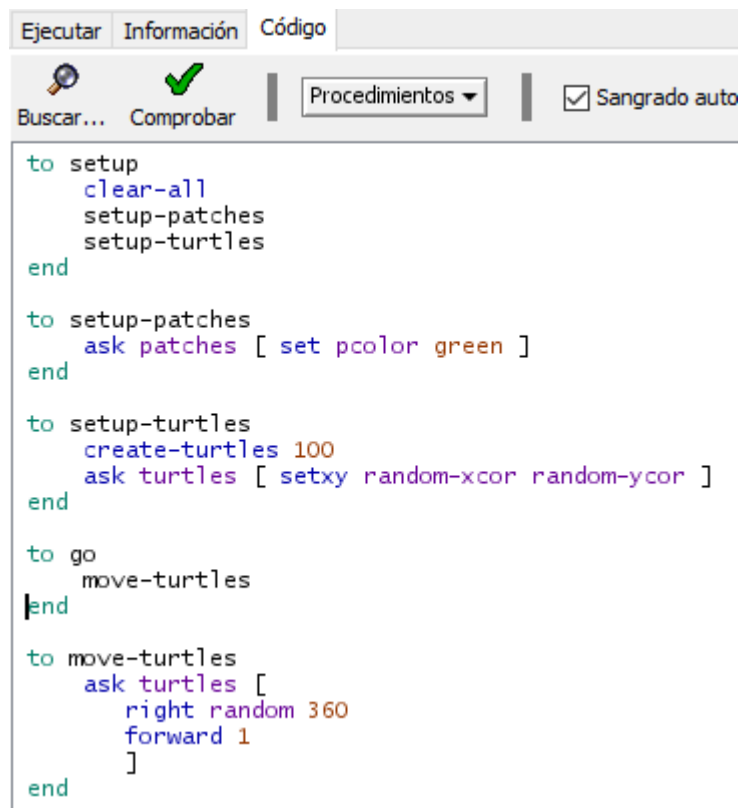
A continuación, agregue un procedimiento Go en la pestaña de procedimientos (Procedures):

Pero, ¿qué es move-turtles ? ¿Se trata de un primitivo (en otras palabras, un incorporado en NetLogo), al igual que lo es clear-all? No, es otro procedimiento que usted está a punto de añadir. Hasta ahora usted ha introducido dos procedimientos que agregó por si mismo: setup y go .

4.3 Parches y variables

Ahora tenemos a 100 tortugas moviéndose sin propósito, completamente desentendidas de cualquier otra cosa que les rodee. Vamos a hacer las cosas un poco más interesantes dándole a estas tortugas un bonito fondo en donde moverse.

- Regresemos al procedimiento `setup`. Podemos reescribirlo de la siguiente manera:
- La nueva definición de `setup` se refiere a dos nuevos procedimientos. Para definir `setup-patches` añada lo siguiente
- Añada también este procedimiento:



```
to setup
  clear-all
  setup-patches
  setup-turtles
end

to setup-patches
  ask patches [ set pcolor green ]
end

to setup-turtles
  create-turtles 100
  ask turtles [ setxy random-xcor random-ycor ]
end

to go
  move-turtles
end

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



Aparece un exuberante paisaje NetLogo completo con las tortugas y los parches verdes:

4.4 Variables de Tortuga

De modo que tenemos algunas tortugas corriendo en un paisaje, pero nada están haciendo con ello. Vamos a añadir algo de interacción entre las tortugas y los parches.

Haremos a las tortugas comer "pasto" (los parches verdes), reproducirse y morir. La hierba crecerá gradualmente después de ser comida.

Necesitamos una manera de controlar cuando una tortuga se reproduce y cuándo muere. Vamos a determinarlo haciendo el seguimiento de la cantidad de "energía" que tiene cada una de las tortugas. Para hacer esto necesitamos añadir una nueva variable en la tortuga.

Ya ha visto la construcción de variables de la tortuga tales como color. Para crear una nueva variable de la tortuga, añadimos una declaración `turtles-own` (tortugas-propia) en la parte superior de la pestaña de procedimientos, antes que todos los procedimientos. Llamela `energy` (energía)

reescriba y añada los siguientes procedimientos

```

turtles-own [energy]

to go
  move-turtles
  eat-grass
end

to eat-grass
  ask turtles [
    if pcolor = green [
      set pcolor black
      set energy (energy + 10) ]
  ]
end

to setup
  clear-all
  setup-patches
  setup-turtles
end

to setup-patches
  ask patches [ set pcolor green ]
end

to setup-turtles
  create-turtles 100
  ask turtles [ setxy random-xcor random-ycor ]
end

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

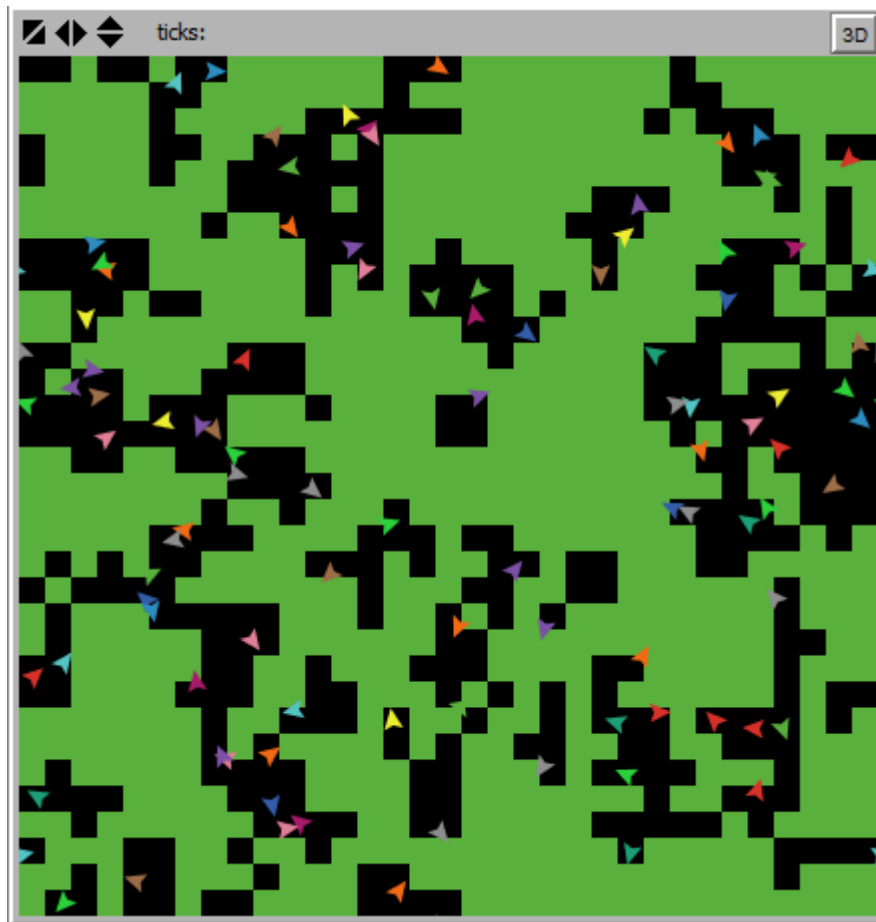
```

Estamos utilizando por primera vez el comando `if`. Observe cuidadosamente el código. Cada tortuga, cuando se ejecutan estos comandos, compara el valor del color del parche (`pcolor`) en el que está parada con el valor para verde. (Una tortuga tiene acceso directo a las variables del parche sobre el cual está parada.) Si el color del parche es verde, la comparación devuelve verdadero, y sólo entonces la tortuga ejecutará los comandos dentro de los paréntesis (de lo contrario los salta). Los comandos hacen que la tortuga cambie el color del parche a negro y que incremente su propia energía en 10. El parche se vuelve negro para significar que la hierba en ese lugar ha sido comida y que a la tortuga ha obtenido más energía por haber comido ese pasto.

A continuación vamos a hacer que el movimiento de las tortugas utilice un poco de la energía de las tortugas.

Ahora vaya a la interfaz y presione el botón `setup` y luego el botón `go`.

Verá los parches volverse negros a medida que las tortugas viajan sobre ellos.



4.5 Monitores

A continuación creará en la interfaz dos monitores con la barra de herramientas. (Usted los hace tal como a los controles deslizantes (sliders) y a los botones, usando el icono del monitor de la barra de herramientas.) Vamos a hacer ahora el primer monitor.

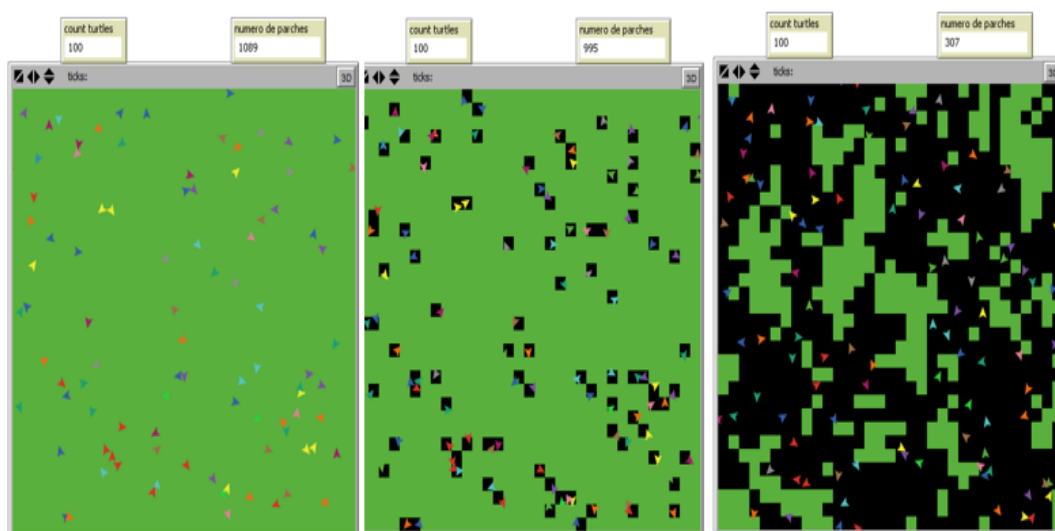
- Cree un monitor, utilizando el icono del monitor de la barra de herramientas, haga clic en un lugar abierto de la interfaz.
- En el cuadro de diálogo escriba: `count turtles` (contar las tortugas, ver imagen inferior).
- Pulse el botón OK para cerrar el cuadro de diálogo.

`turtles` es una "agentset", el conjunto de todas las tortugas. `count` nos dice la cantidad de agentes existentes en ese conjunto.

- Cree un monitor, utilizando el icono del monitor de la barra de herramientas y haga clic en un lugar abierto de la interfaz.

- En la sección Reportero del cuadro de diálogo escriba: `count patches with [pcolor = green]` (ver imagen inferior).
- En la sección Display name del cuadro de diálogo escriba: `green patches`
- Pulse el botón OK para cerrar el cuadro de diálogo.

Nuevamente estamos usando `count` para ver cuántos agentes hay en un `agentset`. `patches` es el conjunto de todos los parches, pero no solamente queremos saber cuántos parches hay en total, queremos saber cuántos de ellos son de color verde. Eso es lo que `with` hace; crea un `agentset` pequeño compuesto sólo de aquellos agentes para los cuales es verdadera la condición en el paréntesis. La condición es `pcolor = green`, de manera que sólo nos devuelve los parches verdes.



4.6 Interruptores y etiquetas

Las tortugas no sólo están volviendo los parches de color negro; ellas también están ganando y perdiendo energía. Mientras esté corriendo el modelo pruebe utilizando un monitor de la tortuga para ver como la energía de una una tortuga aumenta y baja.

Sería mucho mejor si pudiéramos ver todo el tiempo la energía de todas las tortugas. Ahora haremos exactamente eso, y añadiremos un interruptor para que podamos encender y apagar la información visual extra.

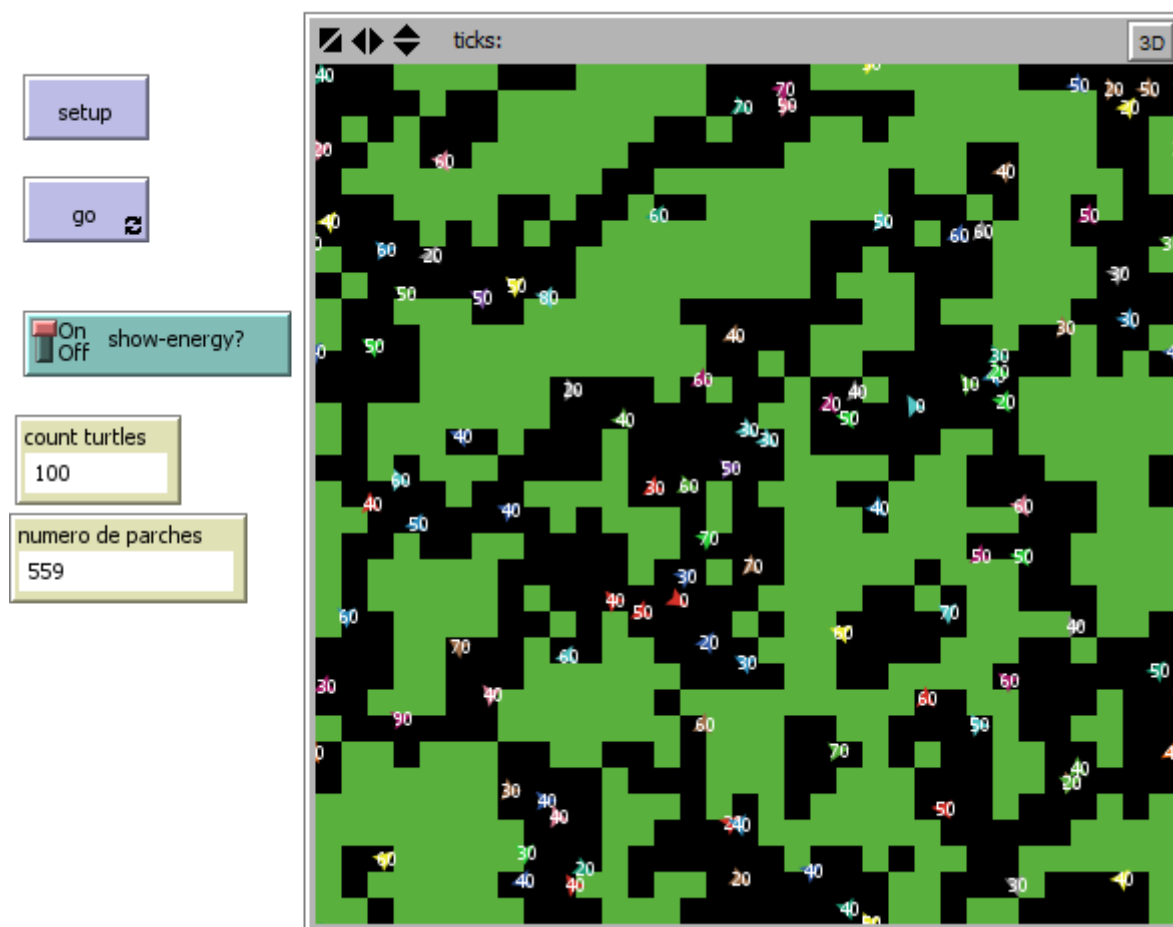
- Para crear un interruptor, haga clic en el interruptor de la barra de herramientas (en la pestaña Interfaz) y haga clic en un punto abierto en la interfaz.
- En la sección de variable Global del cuadro de diálogo escriba: `show-energy?` No olvide de incluir el signo de interrogación en el nombre. (Vea la imagen a continuación.)

Reescriba el procedimiento eat-grass de la siguiente manera:

```
to eat-grass
  ask turtles [
    if pcolor = green [
      set pcolor black
      set energy (energy + 10)
    ]
    ifelse show-energy?
      [ set label energy ]
      [ set label "" ]
  ]
end
```

El procedimiento eat-grass introduce el comando ifelse. Mire detenidamente el código. Cada tortuga, cuando se ejecuta este nuevo comando, verifica el valor show-energy? (determinado por el interruptor (switch)). Si el interruptor está activado en on, la comparación devuelve verdadero y la tortuga ejecutará los comandos del interior de la primera serie de corchetes. En este caso asigna el valor de la energía a la etiqueta de la tortuga. Si la comparación es falsa (el interruptor está apagado), entonces la tortuga ejecuta los comandos del interior de la segunda serie de corchetes. En este caso, elimina el texto de las etiquetas (estableciendo nada en la etiqueta de la tortuga).

(En NetLogo a una pieza de texto se llama una "cadena" o "string" en inglés. Una cadena es una secuencia de letras y otros caracteres escrita entre comillas dobles. Aquí tenemos dos dobles uno al lado del otro, sin nada entre ellos. Eso es una cadena vacía. Si la etiqueta de una tortuga es una cadena vacía, ningún texto se adjunta a la tortuga.)



Ahora nuestras tortugas están comiendo; vamos a hacer que también se reproduzcan y mueran. Además vamos a hacer que la hierba rebrote. Ahora mismo vamos a añadir estos tres comportamientos haciendo tres procedimientos separados. Uno para cada comportamiento.

modifique go y Añada los procedimientos para reproduce, check-death y regrow-grass, como se indica a continuación:

```

turtles-own [energy]

to go
  move-turtles
  eat-grass
  reproduce
  check-death
  regrow-grass
end

to reproduce ;; reproducirse
  ask turtles [
    if energy > 50 [
      set energy energy - 50
      hatch 1 [ set energy 50 ]
    ]
  ]
end

to check-death ;; verificar muerte
  ask turtles [
    if energy <= 0 [ die ]
  ]
end

to regrow-grass ;; rebrotar pasto
  ask patches [
    if random 100 < 3 [ set pcolor green ]
  ]
end

```

Cada uno de estos procedimientos utiliza el comando if. Cada tortuga, cuando se ejecuta reproduce, comprueba el valor de la variable energy de la tortuga. Si esta es mayor que 50, entonces la tortuga ejecuta los comandos en el interior de la primera serie de corchetes. En este caso, disminuye en 50 la energía de la tortuga, y luego 'eclosiona' una nueva tortuga con una energía de 50. El comando hatch es un primitivo de NetLogo que luce como esto: hatch number [commands]. Esta tortuga crea un nuevo número de tortugas, cada una idéntica a su padre y pide a la(s) nueva(s) tortuga(s) creada(s) que ejecuten los comandos. Puede utilizar los comandos para dar a las nuevas tortugas diferentes colores, rumbos, o lo que sea. En nuestro caso corremos un comando. Hemos establecido en 50 la energía para la recién eclosionada tortuga.

Cuando cada tortuga ejecuta check-death comprobará si su energía es menor o igual a 0. Si la expresión devuelve verdadero, entonces se le dice a la tortuga que muera (die) (die es un primitivo de NetLogo).

Cuando se ejecuta regrow-grass en cada parche este verificará si un entero aleatorio de 0 a 99 es inferior a 3. En caso afirmativo, el parche se pone de color verde. Esto sucederá para cada parche el 3 por ciento de las veces (en promedio), ya que sólo existen tres números (0, 1 y 2) de 100 posibles que son menores de 3.

Usted podrá ver ahora algunos interesantes comportamientos en su modelo. Algunas tortugas mueren, algunas nuevas tortugas son creadas (eclosionan), y parte de la hierba rebrota. Esto es exactamente lo que establecimos que hiciera.

Si sigue observando los monitores en su modelo, verá que ambos monitores, count turtles y green patches, fluctúan. ¿ Es previsible este patrón de fluctuación

? ¿Existe alguna relación entre las variables?

4.7 Graficación

Para realizar el trabajo de graficación, tendremos que crear una gráfica en la pestaña de interfaz, y establecer algunos ajustes en ella. Luego añadiremos un procedimiento más a la ficha de Procedimientos (Procedures), que actualizará la gráfica por nosotros.

Ahora agregue el nuevo procedimiento. Lo que estamos graficando será el número de tortugas y el número de parches verdes contra el tiempo. En cada paso de tiempo (una ejecución simple a través del procedimiento go) estos valores se añaden a la gráfica.

```
to do-plots
  set-current-plot "Totals"
  set-current-plot-pen "turtles"
  plot count turtles
  set-current-plot-pen "grass"
  plot count patches with [pcolor = green]
end
```

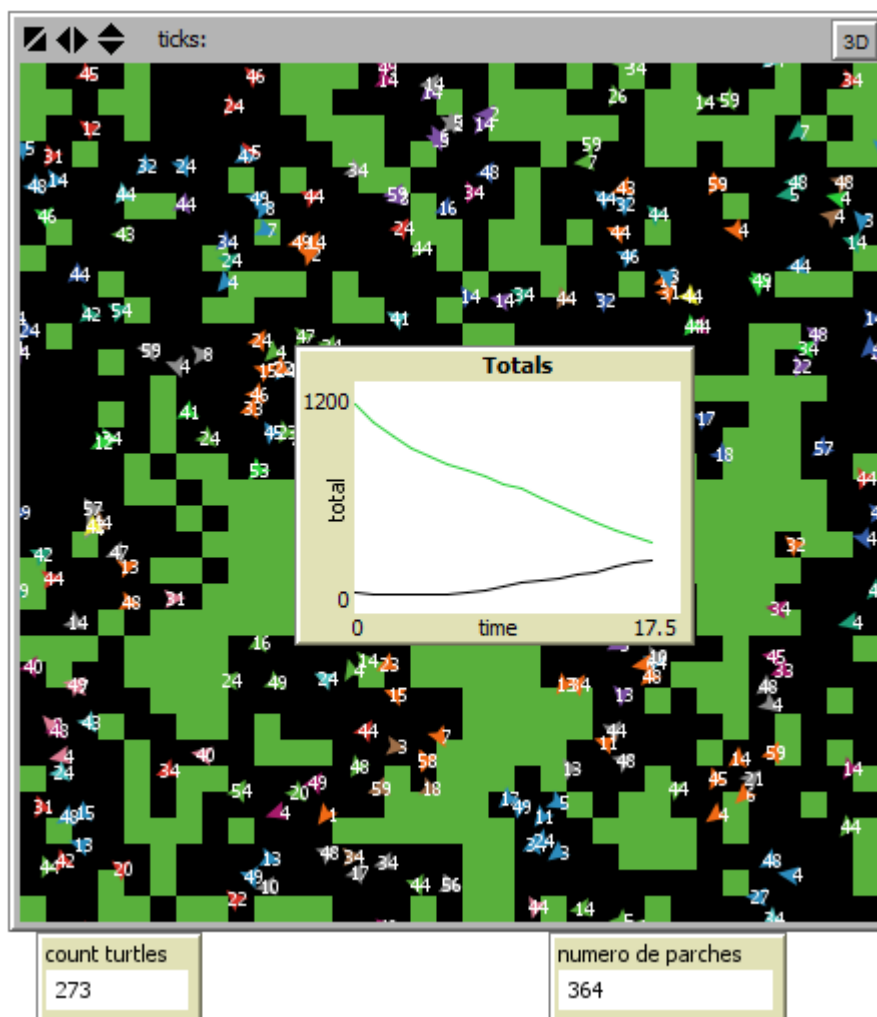
Tenga en cuenta que usamos el comando plot para añadir el siguiente punto a una gráfica. Sin embargo, antes de hacerlo, necesitamos decirle a NetLogo dos cosas. En primer lugar, tenemos que especificar qué gráfica estaremos utilizando (desde luego que nuestro modelo podría tener más de una gráfica) y en segundo lugar, tenemos que especificar qué línea queremos graficar (estaremos utilizando dos curvas en esta gráfica).

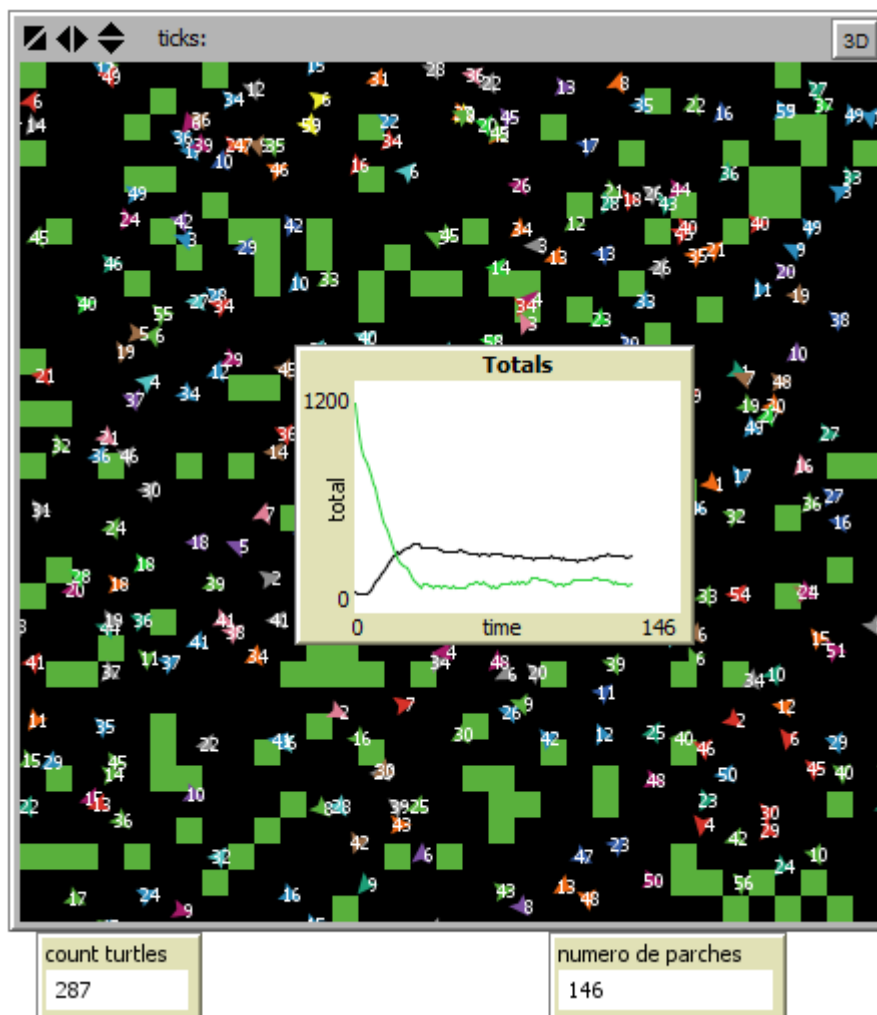
El comando plot mueve la posición actual de la curva de la gráfica hasta el punto en que tiene una coordenada X con valor 1 mayor que la coordenada previamente graficada para X y una coordenada Y igual al valor dado en el comando plot (en el primer caso, el número de tortugas, y en el segundo caso, el número de parches verdes). A medida que las curvas se mueven cada una de ellos dibuja una línea.

Con el fin de establecer los "Totales" (set-current-plots "Totals") de la -actual gráfica para trabajar, tendrá que añadir una gráfica a su modelo en la interfaz, y luego editarlo para que su nombre sea el mismo nombre utilizado en los procedimientos. Incluso un espacio extra en el nombre lo hará impreciso - debe ser exactamente el mismo en ambos sitios.

- Cree un gráfico, utilizando el icono plot de la barra de herramientas y haga clic en un lugar abierto en la interfaz.
- Establezca su nombre como "Totals" (ver imagen inferior)
- Establezca el eje de las X con la etiqueta "time"
- Establezca el eje Y con etiqueta "total"
- Con el cuadro de diálogo de Plot (Gráfica) aun abierto, pulse el botón 'Create' en el diálogo de Plot para crear una nueva curva.
- Introduzca el nombre de esta curva como "turtles" y pulse OK en el diálogo de "Enter Pen Name". (ver imagen inferior)

- Pulse nuevamente el botón 'Crear' en el cuadro de diálogo de nuevo, para crear una segunda curva nueva.
- Introduzca el nombre de esta curva como "grass" y pulse OK en el cuadro de diálogo "Enter Pen Name". (ver imagen inferior)
- Seleccione el color de la curva y cámbielo a verde.
- Seleccione OK en el cuadro de diálogo de la Plot (Gráfica).





4.8 Algunos detalles adicionales

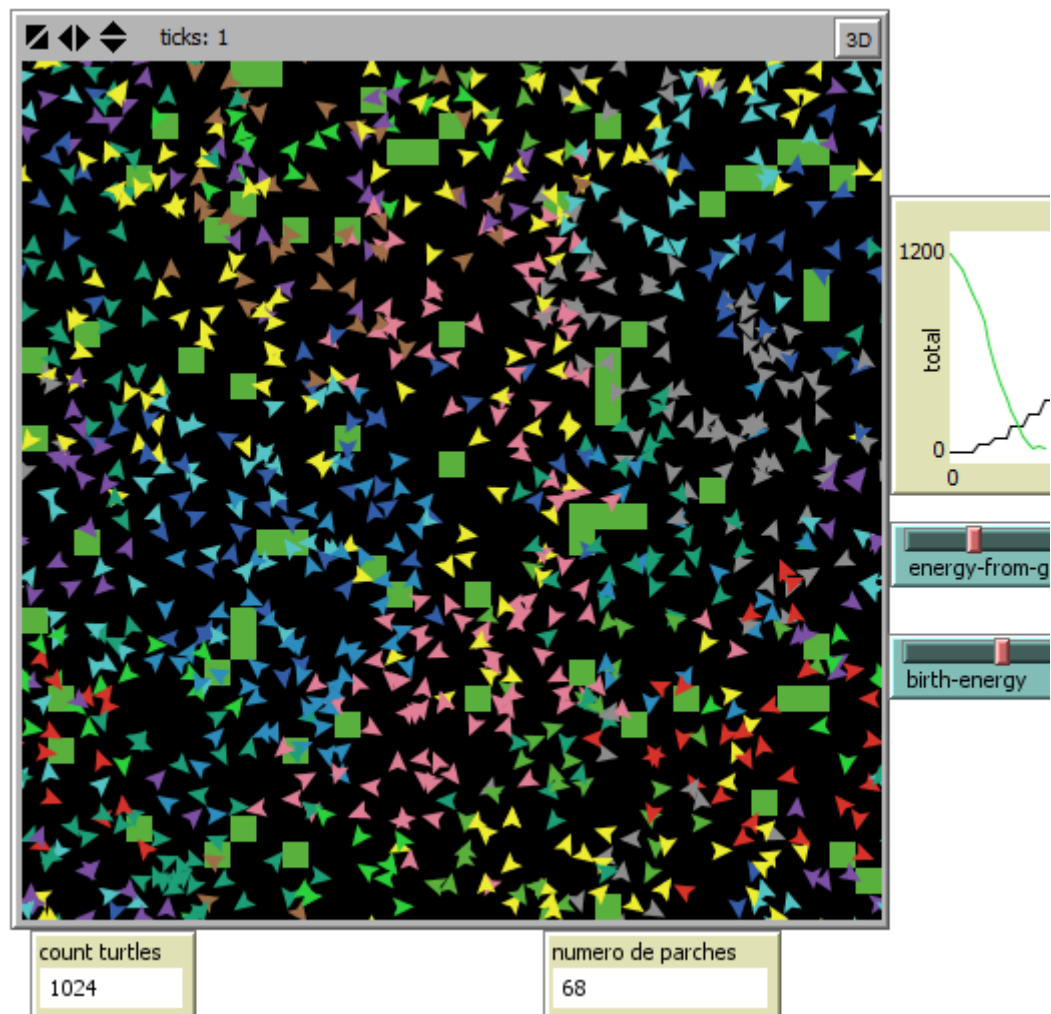
Primero, en lugar de usar siempre 100 tortugas usted puede tener un número variable de ellas.

- Cree una variable slider llamada "numero" usando el ícono monitor de la barra de herramientas y haga clic en un lugar limpio de la interfaz. Intente cambiando los valores mínimo y máximo en el slider.
- Luego, dentro de `setup-turtles`, en lugar de `create-turtles 100` usted puede escribir:

```
to setup-turtles
  create-turtles number
  ask turtles [ setxy random-xcor random-ycor ]
end
```

- Haga un slider llamado `energy-from-grass`.

- Haga otro slider llamado birth-energy.
- Luego, dentro de eat-grass, haga este cambio:
- And, inside of reproduce , make this change: Y dentro de reproduce haga este cambio



con el ultimo cambio las tortugas crecen en poblacion hasta un punto en el que su numero se estabiliza entre 600 y 700 tortugas