Universidad Autónoma de Chiapas

Inteligencia Artificial

PRÁCTICA MODELADO BASADO EN AGENTES

Integrantes:

Tomás Álvarez Gómez-A200369

Erick Francisco Palacios Trejo A190183

Nestor Horacio Zea Hernández A200727

Yahir Emmanuel Ramírez Díaz A200251

PROCEDIMIENTO “¡A LA CAZA DE CHAMPIÑONES!” (MUSHROOM HUNT)

Paso 1: Crear el Mundo

1. Abre NetLogo y selecciona “Nuevo” para crear un nuevo modelo.

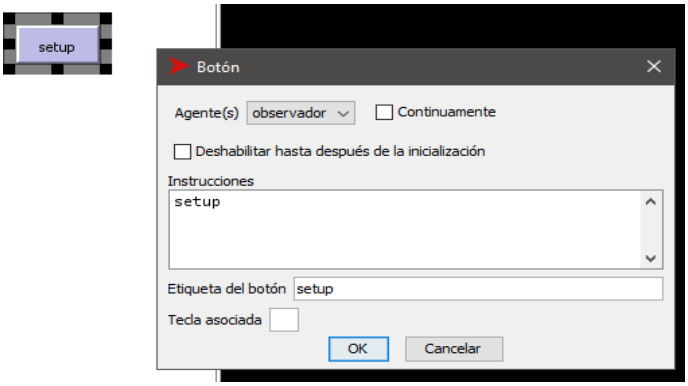
Haga clic en el botón Configuración, el botón Configuración abre el cuadro de diálogo de configuración del modelo, donde puede verificar la geometría del mundo.Usaremos la geometría predeterminada con:

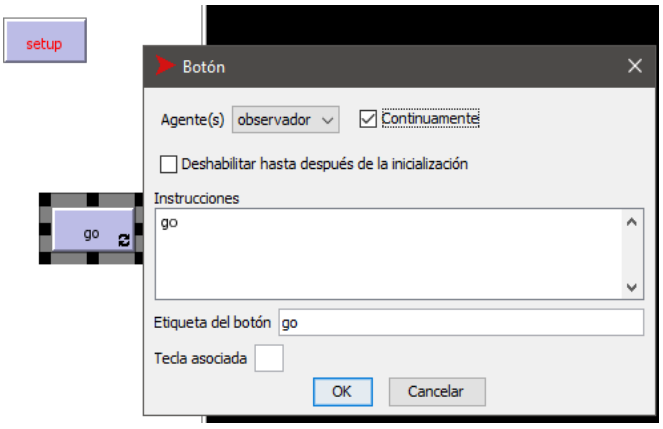
el origen (parcela 0,0) en el centro del mundo.

la coordenada máxima en x (max-pxcor) y coordenada máxima en y(max-pycor) en 16.

Paso 2: Crear los botones setup y go

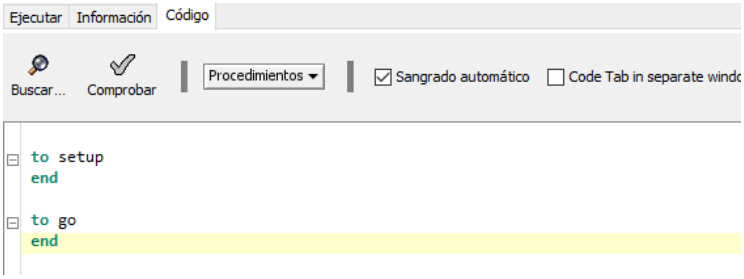
En la interfaz elegimos Botón y en input instrucciones escribimos “setup” al igual en la etiqueta del botón hacemos lo mismo con el botón “go” pero en este caso le daremos check donde dice “Continuamente” esto para que se ejecute continuamente para detenerlo tendrá que hacer clic de nuevo en el botón go.





Paso 3: Definición de setup y go

Creamos el procedimiento setup y go en la pestaña código (con dos líneas que to “nombre del procedimiento” end)



3.1 Construyendo las parcelas.

Ahora le diremos al procedimiento setup que cambie el color de las parcelas.

to setup

clear-all ;; Limpia el modelo, eliminando todos los parches y tortugas existentes.

ask n-of 4 patches [;; Selecciona aleatoriamente 4 parches.

ask n-of 20 patches in-radius 5 [;; Selecciona aleatoriamente 20 parches dentro de un radio de 5 parches alrededor del parche seleccionado anteriormente.

set pcolor red ;; Cambia el color de los 20 parches seleccionados a rojo.

]

]

create-turtles 2 [;; Crea 2 tortugas nuevas en el modelo.

pen-down ;; Pone el lápiz de la tortuga hacia abajo, para que pueda dibujar en el mundo.

set size 2 ;; Establece el tamaño de la tortuga en 2.

set color yellow ;; Establece el color de la tortuga en amarillo.

set time-since-last-found 999 ;; Inicializa una variable llamada "time-since-last-found" en 999 para cada tortuga.

]

reset-ticks ;; Restablece el contador de ticks del modelo a 0.

End

3.2 Variables globales

globals [

num-clusters ;; Declara una variable global llamada "num-clusters" que se puede usar en cualquier parte del código.

]

turtles-own [

time-since-last-found ;; Cada tortuga tiene su propia variable llamada "time-since-last-found" que puede ser diferente para cada tortuga.

]

3.3 Comportamiento de los agentes (Procedimiento go)

El procedimiento “go” define el “comportamiento” del modelo, osea los procesos que se realizarán una y otra vez.

to go

ask turtles [search] ;; Pide a todas las tortugas que ejecuten el procedimiento llamado "search".

tick ;; Aumenta el contador de ticks del modelo en 1. Los ticks son una unidad de tiempo en NetLogo.

end

Agregue la línea, ask turtles [search] al procedimiento go. Luego cree un procedimiento vacío para que pueda verificar el código :

to search end

Ahora programamos cómo buscan los agentes (buscadores) Cada vez que se ejecuta la búsqueda, queremos que los cazadores giren, un poco si no han encontrado nada recientemente (por ejemplo, en las últimos veinte parcelas recorridas), o bruscamente para seguir buscando en la misma área si encontraron un hongo recientemente. Luego de girar los buscadores deben avanzar hacia una parcela vecina. * Agregue el siguiente código al procedimiento de búsqueda, cualquier parte que no entienda búsquela en el diccionario de NetLogo:

ifelse time-since-last-found <= 20 ;;si el tiempo transcurrido desde la última vez que se encontró algo es menor o igual a 20, se ejecuta el código dentro del primer conjunto de corchetes.

[right (random 181) - 90] ;;Si el tiempo es corto la tortuga gira a la derecha un número aleatorio entre 0 y 180 grados y se resta 90 grados osea que la tortuga gira en un ángulo aleatorio entre -90 y 90 grados.

[right (random 21) - 10] ;; En este caso la tortuga gira un número aleatorio entre 0 y 20 grados, y luego se le resta 10 grados, girara entre un ángulo entre -10 y 10 grados.

forward 1 ;; Después la tortuga girar la tortuga avanza una unidad hacia adelante, se ejecutara después de la estructura "ifelse".

3.4 Variables locales de agentes

Primero, Cada cazador debe tener su propio valor único para esta variable, por lo que debe ser una variable de tipo turtle (agente), justo después de la declaración global al comienzo de su programa, agregue:

turtles-own [

time-since-last-found

]

En segundo lugar, necesitamos establecer el valor inicial de esta variable cuando creamos los buscadores Estableceremos,asumiendo que los cazadores aún no han encontrado un hongo,un valor mayor que 20.

En el procedimiento de setup, cambie la instrucción create-turtles a esto:

create-turtles 2 [

set size 2 ;; Establece el tamaño de la tortuga en 2 unidades.

set color yellow ;; Establece el color de la tortuga en amarillo.

set time-since-last-found 999 ;; Inicializa la variable 'time-since-last-found' en 999 para cada tortuga.

]

Finalmente, los cazadores deben actualizar time-since-last-found cada vez que se mueven. Si encuentran un hongo, necesitan restablecer time-since-last-found a cero (¡y recoger el hongo!); de lo contrario, necesitan agregar uno a time-since-last-found.

Agregue estas declaraciones al final del procedimiento search:

ifelse pcolor = red [

set time-since-last-found 0 ;; Si el color del parche es rojo, establece 'time-since-last-found' en 0.

set pcolor yellow ;; Cambia el color del parche a amarillo.

] [

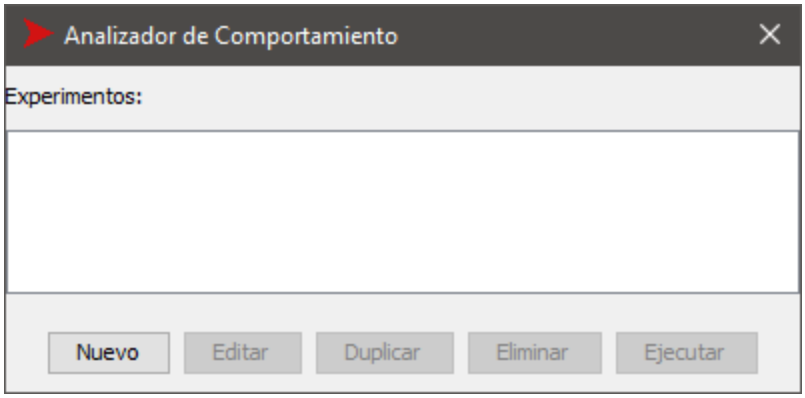
set time-since-last-found time-since-last-found + 1 ;; Si el color del parche no es rojo, incrementa 'time-since-last-found' en 1.

]

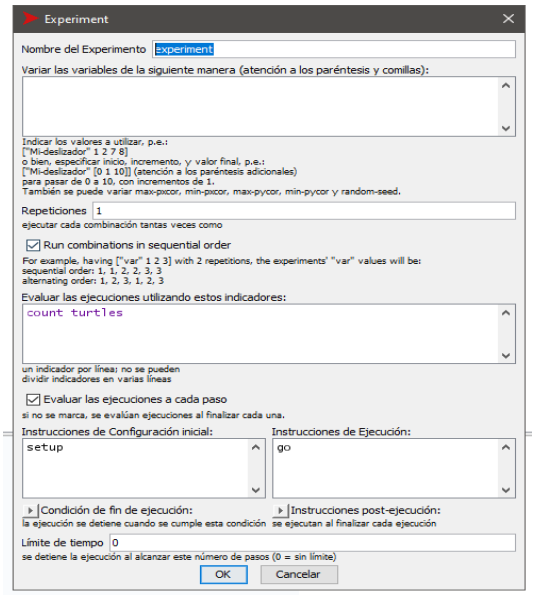
El código completo te debería quedar de la siguiente semana (Esto lo puedes ver en Resultados y conclusiones)

3.5 El analizador de comportamiento

En la pestañas de “Herramientas” vamos a elegir la opción que dice “Analizador de comportamiento” le daremos clic y se abrirá la siguiente ventana.



Le damos clic al botón “Nuevo” para crear un nuevo analizador y se nos abrirá otra ventana

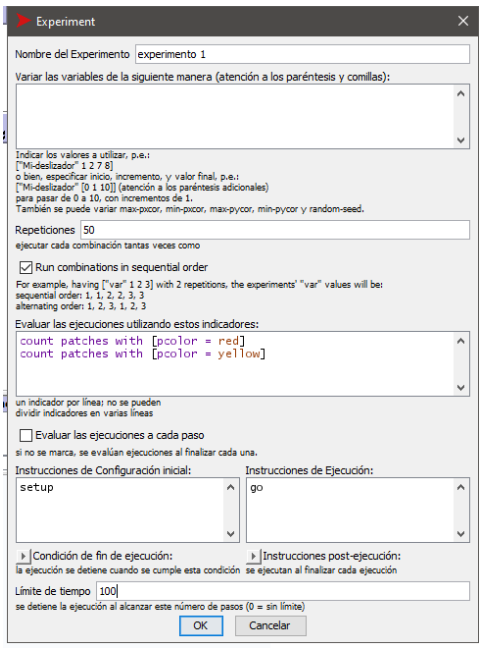


le ponemos el nombre que queramos en el número de repeticiones como lo dice el experimento le pondremos “50” Repeticiones de ejecución, en el input de “Evaluar las ejecuciones utilizando estos indicadores:” pondremos los siguientes comandos:

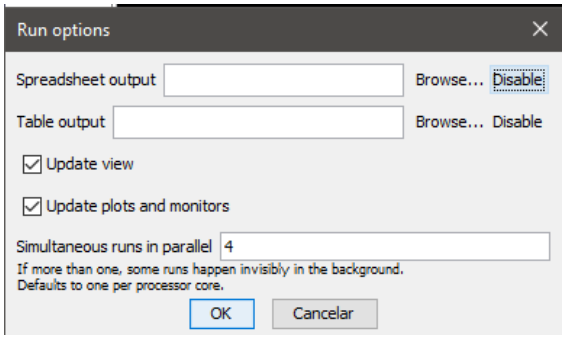
count patches with [pcolor = red] ;; Cuenta la cantidad de parches con color rojo en el modelo.

count patches with [pcolor = yellow] ;; Cuenta la cantidad de parches con color amarillo en el modelo.

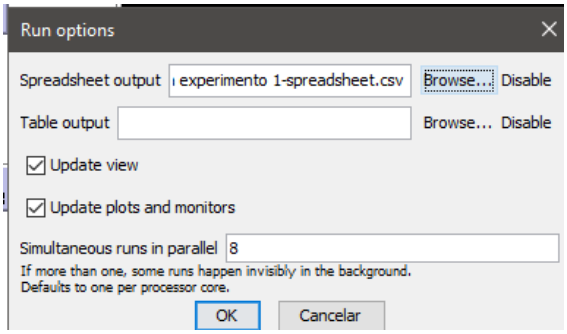
Y en el input “Límite de tiempo” para el primer experimento le pondremos 100 debería quedarnos de la siguiente manera:



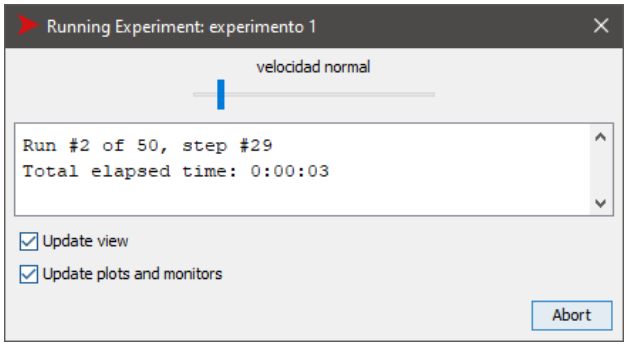
Le daremos “OK” se cerrará la ventana y a la ventana de Experimentar le damos al botón “Ejecutar” nos saldrá otra ventana:



En la ventana le daremos clic al botón “Browser” se nos abrirá la ventana para guardar archivos le pondremos el nombre que queramos y la ruta en la cual se guardará el archivo tipo Excel con la tabla de contabilidad del analizador de comportamiento y en el input “Simultaneos runs in parallel” en vez de 4 le pondremos 8 como nos dice en el experimento:



Le damos al botón “OK” y se ejecutara el experimento nos aparecerá esto:



y en la ruta donde guardamos nuestro archivo se creará y podemos ver el comportamiento que tiene nuestro programa con esos parámetros:

A	B	C	D	E	
BehaviorSpace results (NetLogo 6.3.0)					
ChampiÃ±ones a la caza.nlogo					
experimento 1					
09/29/2023 14:11:16:734 -0500					
min-pxcor	max-pxcor	min-pycor	max-pycor		
-16	16	-16	16		
[run number	1	1	2	2	
[reporter]	count patche	count patche	count patche	count patche	co
[final]	70	10	55	21	
[min]	70	0	55	0	
[max]	80	10	76	21	
[mean]	74.990099	5.00990099	62.4653465	13.5346535	6
[steps]	100	100	100	100	
[all run data]	count patche	count patche	count patche	count patche	co
	80	0	76	0	
	80	0	75	1	
	80	0	75	1	
	80	0	74	2	
	80	0	74	2	
	80	0	73	3	

Conclusiones:

Programa:

```
globals[
num-clusters
]
turtles-own [
time-since-last-found
]

to setup clear-all
ask n-of 4 patches [
ask n-of 20 patches in-radius 5 [ set pcolor red
]
]
create-turtles 2 [ pen-down
set size 2
set color yellow
set time-since-last-found 999
]
reset-ticks end

to go
ask turtles [search] tick
end

to search
ifelse time-since-last-found <= 20 [right (random 181) - 90]
[right (random 21) - 10]
forward 1

ifelse pcolor = red [
set time-since-last-found 0 set pcolor yellow
][
set time-since-last-found time-since-last-found + 1
]
end
```