

معرفی نرم افزار

نرم افزار NetLogo یک نرم افزار مدل سازی برای شبیه سازی پدیده‌های مختلف است. این نرم افزار توسط Uti Wilensky در سال ۱۹۹۹ معرفی شد و در مرکز آموزش ارتباطات و مدلسازی کامپیوتری توسعه یافت. این نرم افزار برای شبیه سازی سیستم‌های پیچیده طراحی شده است. کاربردهای آن در علوم ریاضی، فیزیک، شیمی، کامپیوتر، پزشکی، اجتماعی و اقتصادی و ... است.

یکی از مزیت‌های این نرم افزار این است که با زبان جاوا نوشته شده است. این ویژگی باعث شده که این نرم افزار بتواند بر روی هر پلتفرمی (مانند Windows, Mac, Linux) اجرا شود.

ویژگیهای NetLogo:

- سیستم عامل: قابل اجرا بر روی پلتفرمهای مختلف
- زبان: کاملا قابل برنامه‌ریزی، ساختار زبانی ساده، قابلیت مشاهده عوامل ایجاد شده در بستر ایجاد عوامل، امکان مرتبط نمودن اشیا ایجاد شده و شبکه سازی آنها، دارا بودن مجموعه قابل توجهی از اشکال از پیش طراحی شده و ...
- محیط: قابلیت مشاهده مدل بصورت دو و سه بعدی، اشکال قابل چرخش و اندازه پذیر، خط فرمان هوشمند، دارا بودن اهرم سرعت که اجازه کنترل سرعت مدل و مشاهده خروجی را با دقت بیشتر می‌دهد، سیستم رسم نمودار انعطاف پذیر و قدرتمند و ...
- عوامل: مبنی بر عوامل (Agent base) بودن به گونه‌ای که هر عامل دارای مشخصات خاص خود می‌باشد.

۱- محیط نرم افزار

باز کردن یک نمونه آماده از کتابخانه

File>Models Library>Social Science>AIDS

دکمه Setup: شرایط را بر محیط وارد می‌کند.

دکمه Go: تا زمانی که روشن باشد تعاملات را نشان می‌دهد (به صورت forever می‌باشد).

Interface: محیط شبیه سازی است.

Info: راهنمای شبیه سازی طراحی شده بوده که توسط طراح شبیه سازی تدوین می‌گردد.

Code: برنامه در اینجا نوشته می‌شود.

۲- دکمه (Button):

جهت ایجاد دکمه در قسمت سفید رنگ Interface

نام دکمه: R-Click>Button>Display name:

Forever

با انتخاب گزینه Forever، دکمه فعال می‌ماند تا زمانی که غیر فعال گردد.

۳- Info

در تب Info از طریق دکمه Edit می توان توضیحات و راهنمای برنامه را ایجاد نمود. توضیحاتی که به صورت عنوان باید درج گردند با ## (دو علامت هشتگ یا Shift+3) شروع می گردند.

۴- Code

هر رویه (Procedure) در حد فاصل to تا end نوشته می شود. پس از دستور to نام رویه درج شده که با اضافه کردن این نام در قسمت Commands دکمه، ارتباط دکمه و رویه برقرار می شود. همچنین یک رویه می تواند شامل تعدادی رویه دیگر باشد که به صورت جداگانه هر یک از آن رویه ها در حد فاصل to تا end فرآیند خود را اجرا نمایند.

۵- عامل (Agent)

- عامل های متحرک turtles: عاملی است که در سطح شبکه حرکت می کند.
- عامل های ثابت patches: ثابت یا ایستا بوده و شامل شبکه ای از سلولها ست که کنار هم چیده شدند.
- رابطه یا پیوند links: دو عامل متحرک را بهم مرتب می نماید.
- ناظر observer: بر هر آنچه که در حال اتفاق افتادن است نظارت می کند.

۶- Slider

جهت ایجاد slider (کنترل کننده وضعیت به صورت کشویی) در قسمت سفید رنگ Interface نام اسلایدر: R-Click>Slider>Global variable: حد پایینی (Minimum) و بالایی (Maximum) و همچنین تعداد پرشهای اسلایدر (Increment) مشخص می گردد. می توان ارزش پیش فرض اسلایدر (Value) و واحد آن (Unit) را نیز مشخص نمود. اسلایدر می تواند به صورت افقی و یا عمودی (Vertical) باشد. جهت اتصال به کد، نام اسلایدر را باید در جلو دستور درج نمود. مانند:

create-turtles SliderName

۷- بازرسی عامل و مشخصات آن:

در محیط شبیه سازی، می توان بر روی هر یک از عاملها راست کلیک نمود با انتخاب عنوان عامل و گزینه inspect مشخصه های عامل را مورد بازرسی قرار داد. مشخصه ها به صورت پیش فرض بدین ترتیب می باشد:

شماره عامل (who)، رنگ عامل (color)، جهت حرکت عامل (heading)، موقعیت محور Xها (xcor)، موقعیت محور Yها (ycor)، شکل عامل (shape)، بر چسب عامل (label)، رنگ برچسب (label-color)، قطر خط مسیر حرکت عامل (pen-size)، شرایط خط (pen-mode)

۸- تعریف متغیرها یا ثابت‌های عمومی (globals):

تعریف متغیر جدید در آغاز برنامه و قبل از هر تابعی صورت می‌پذیرد و متغیرهای تعریف شده به صورت سراسری و عمومی بوده و توسط تمامی عوامل در دسترس هستند و می‌تواند در هر نقطه از یک مدل استفاده شوند.

۹- تعریف یک سری عامل جدید (breed):

نرم افزار NetLogo اجازه می‌دهد که عامل‌های متحرک جدید ایجاد شوند. این عامل مانند globals باید در ابتدای برنامه و پیش از سایر توابع تعریف شود. اولین ورودی نام عامل‌های مربوطه (به صورت جمع) بوده و دومین ورودی نام یک عامل منفرد می‌باشد مانند:

```
breed [wolves wolf]
breed [sheep a-sheep]
```

۱۰- مشخصات جدید عامل (turtles-own و breed-own):

به هر یک از عوامل (ثابت، متحرک و یا حتی عامل‌های جدید تعریف شده) می‌توان مشخصات جدیدی را به خود اختصاص دهند. به عنوان مثال:

```
breed [cats cat ]
breed [dogs dog]
breed [hamsters hamster]
turtles-own [eyes legs] ; برای تمامی عوامل جدید بکار می رود
cats-own [fur kittens] ; تنه‌ای برای عامل گربه های بکار می رود
hamsters-own [fur cage]
dogs-own [hair puppies]
```

۱۱- ایجاد خط مسیر:

از طریق کد زیر می‌توان خط مسیر حرکت یک عامل را کشید. به صورت پیش فرض این مشخصه به صورت "up" می‌باشد.

```
set pen-mode "down"
```

همچنین این مشخصه می‌تواند به صورت erase باشد، که خطها را پاک می‌کند.

عرض این خط را می‌توان با دستور زیر تغییر داد

```
set pen-size 5
```

۱۲- ارتباط بین دو عامل متغیر create-link-with:

```
ask turtle 0 [ create-link-with turtle 1 ]
```

۱۳- نمودار و مانیتور:

R-Click>Monitor>Reporter:

- count turtles
- count turtles with []
- max [] of turtles

R-Click>Plot>plot pens:

- plot count turtles
- plot count turtles with [color = red]
- plot count patches with [pcolor = green]

۱۴- اضافه کردن تصویر به interface:

File > Import > Import Drawing ...

تصویر اضافه شده به صورت پس زمینه بوده و هیچ گونه ارتباطی با برنامه برقرار نمی کند.

File > Import > Import Patch color ..

در این حالت تمامی عاملها می توانند با تصویر ارتباط برقرار کنند.

تمرین ۱: سه عامل با مشخصات مختلف از پایین صفحه حرکت کرده و با سرعتهای متفاوت به بالای صفحه رسیده و متوقف گردند.

to setup

clear-all

create-turtles 3

ask turtle 0 [

set shape "wolf"

set color red

set size 2

set heading 0

setxy random-xcor -15

]

ask turtle 1 [

set shape "sheep"

set color 9.9

set size 2

set heading 0

setxy random-xcor -15

]

ask turtle 2 [

set shape "cow"

```

set color brown
set size 2
set heading 0
setxy random-xcor -15
]
end

```

```

to Go
ask turtles [
  ifelse ycor <= 14 [fd random 3]
    [ stop ]
]
end

```

تمرین ۲: تعیین ۵ مرد دارای مشخصه های «شغل» و «تناسب اندام» و «ارزش کلی» و ۳ ماشین دارای مشخصه های «قیمت» و «سرعت» و «ارزش کلی». حرکت مردها با سرعت یک خانه و حرکت ماشینها با سرعت ۳ خانه.

```

breed [men man]
breed [cars car]
men-own [job fitness]
cars-own [price speed]
turtles-own [value]
to Setup
clear-all
create-men 5
create-cars 3
ask men [
  set shape "person"
  set size 3
  setxy random-xcor random-ycor
]
ask cars [
  set shape "car"
  set size 3
  setxy random-pxcor random-pycor
]
End
to Go
ask men [fd 2 ]
ask cars [fd 3 ]
end

```

تمرین ۳: یک مرتع با وجود تعدادی گاو را شبیه سازی کنید. شرایط حاکم بدین شرح می باشد:

- ۸۵٪ از سطح مرتع پوشیده از منابع گیاهی است و به میزان ۳٪ دوباره رشد می کند
- گاوها به صورت پراکنده با انرژی بین صفر تا پنج در مرتع چرا کرده و با هر خانه حرکت ۱ واحد انرژی از دست داده و با خوردن هر واحد از منابع گیاهی ۱۰ واحد انرژی کسب می کنند. عمر طبیعی یک گاو به میزان ۴۰۰ گام حرکت می باشد.
- گاوای که ۶۰ واحد انرژی کسب کرده باید تولید مثل نموده و حداکثر ۲ گاو بدنیا آورده و ۶۰ واحد انرژی را از دست بدهد. گاو های متولد شده انرژی ای بین صفر تا ۳ دارند.

```
breed [cows cow]
```

```
cows-own [energy step generation]
```

```
to setup
```

```
  ca
```

```
  grass-set
```

```
  cow-set
```

```
  reset-ticks
```

```
end
```

```
to go
```

```
  cow-movie
```

```
  cow-eat
```

```
  cow-die
```

```
  cow-generate
```

```
  grass-growth
```

```
end
```

```
to grass-set
```

```
  ask patches [set pcolor brown]
```

```
  ask patches with [random 100 < 85]
```

```
  [ set pcolor green ]
```

```
end
```

```
to cow-set
```

```
  create-cows Cow-population ;; slider bitween 20 to 500
```

```
  ask cows [
```

```
    setxy random-pxcor random-pycor
```

```
    set shape "cow"
```

```
    set color 45
```

```
    set energy (random 5)
```

```
    set step 0
```

```
    set generation 1
  ]
end

to cow-movie
  ask cows [
    right random 360
    fd 1
    set energy energy - 1
    set step step + 1
  ]
end

to cow-eat
  ask cows[
    if pcolor = green [
      set pcolor brown
      set energy energy + 10
    ]
  ]
end

to cow-die
  ask cows [
    if energy < 0 or step = 400 [
      die
    ]
  ]
end

to grass-growth
  ask patches [
    if random 100 < 3 [
      set pcolor green
    ]
  ]
end

to cow-generate
  ask cows [
    if energy > 60 [
      set energy energy - 60
      hatch random 3 [
```

```
set energy random 4
set step 0
set color color + 1 ;;evry generation changed color
set generation generation + 1 ;;generation counter
]
]
]
end
```

تمرین ۴: یک دکمه طراحی کنید که یک عکس را به پس زمینه اضافه کند.

```
to pic
import-drawing "001.jpg"
end
```

NashrElm.com

راهنمای توابع

to ... end	شروع و پایان یک رویه
clear-all و یا ca	همه چیز را به حالت اولی بر می گرداند (تمامی اجسام و فضاها را که ساخته شده پاک می کند و فرمان را برای یک اجرای تازه آماده می کند.)
create-turtles 5	تعداد ۵ عدد عامل متحرک را در مرکز مدل می سازد
crt 5	همانند دستور قبل، تعداد ۵ عدد عامل متحرک را در مرکز مدل می سازد (تعداد ۵ برای نمونه بوده و به جای آن حتی می توان از نام slider نیز استفاده نمود)
ask turtle 5 []	صدا کردن عامل متحرک ۵ جهت تخصیص مشخصه های داخل کروشه
ask turtles []	صدا کردن تمامی عامل های متحرک جهت تخصیص مشخصه های داخل کروشه
set (let و یا)	تنظیم و تخصیص مشخصه های مختلف به عامل (مانند: color , heading , size , shape و ...)
set xcor 10	تخصیص موقعیت ۱۰ به مختصات X
set ycor 7	تخصیص موقعیت ۷ به مختصات y
setxy A B	تخصیص موقعیت A به مختصات X و موقعیت B به مختصات y
setxy random-xcor random-ycor	تخصیص موقعیت رندم به مختصات X و مختصات y
fd 1	حرکت رو به جلو به تعداد ۱ خانه
rt 90	زاویه حرکت عامل (بر اساس درجه)
tick	واحد شبیه سازی است
reset-ticks	شمارنده tick را صفر می کند
die	حذف عامل متحرک
random 86	یک عدد صحیح رندوم بین صفر تا ۸۶
hatch 1	تولید یک عامل جدید در محل و با مشخصات والد خود
create-link-with	ایجاد یک لینک بین دو عامل متغیر
; (shift + :)	نوشته های پس سیمیکولون تا پایان خط اجرا نشده و به عنوان یادداشت قلمداد می گردد.