

(IIA) Exercicios Prática NetLogo

From WikiNote

Contents

- 1 Ficha 1
 - 1.1 Resolução
 - 1.2 Layout final
 - 1.2.1 Inicio da experiência
 - 1.2.2 Fim da experiência
 - 1.2.3 Resultados da experiência
 - 1.3 Notas do Exercício
 - 1.3.1 Fonte
- 2 Ficha 2
 - 2.1 Exercício 1 a)
 - 2.1.1 Enunciado
 - 2.1.2 Resolução
 - 2.2 Exercício 1 b)
 - 2.2.1 Enunciado
 - 2.2.2 Resolução

Ficha 1

Resolução

```

?
1 turtles-own [energia] ; os agentes nascem com energia (tanto o homem como a mulher)
2 breed [homens homem] ;criação das especies
3 breed [mulheres mulher]
4
5 to setup ; procedimento do botao "setup"
6
7   ca ;limpa o ambiente
8   setup-patches ;procedimento criado por nós
9   create-homens nhomens ;cria nhomens (agentes)
10  create-mulheres nmulheres ;cria nmulheres (agentes)
11
12  ask turtles ;chama os agentes
13  [
14    setxy random-pxcor random-ycor
15    set shape "person" ;muda o shape do agente para person
16    set energia random 101 ; a energia é random entre 0 e 100 (0 e n-1)
17  ]
18
19 ;tudo o que é generico entre as especies usamos o ask turtles
20 ;tudo o que seja especifico, criamos um ask para cada especie
21
22  ask homens[
23    set color blue
24    set size 1
25  ]
26
27  ask mulheres[
28    set color red
29    set size 1.5
30  ]
31
32  reset-ticks ; da reset ao tick sempre que se inicializa um mundo
33
34end
35
36to go ;procedimento do botao "go"
37
38  ask turtles ; chama os agentes
39  [
40    comer ;procedimento para o agente começar a comer
41    forward 1 ; anda para a frente 1 unidade
42
43    ifelse breed =mulheres [ ;se o breed for mulheres
44      set energia energia - 2 ;sempre que um agente avança uma posição perde 2 de energia
45    ]
46    [
47      set energia energia - 1 ; se o breed for homem, perde 1 de energia
48    ]
49

```

```

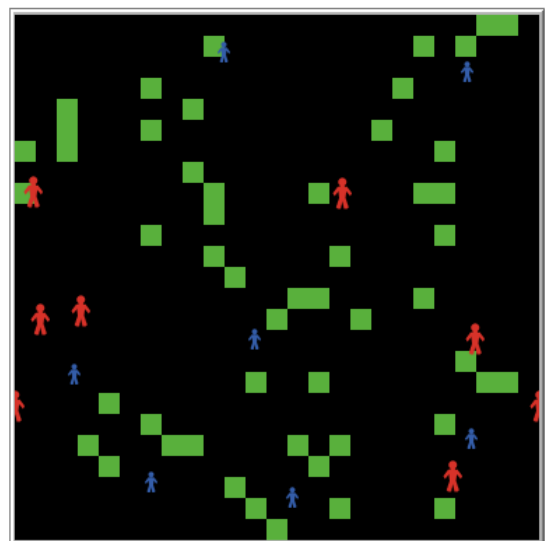
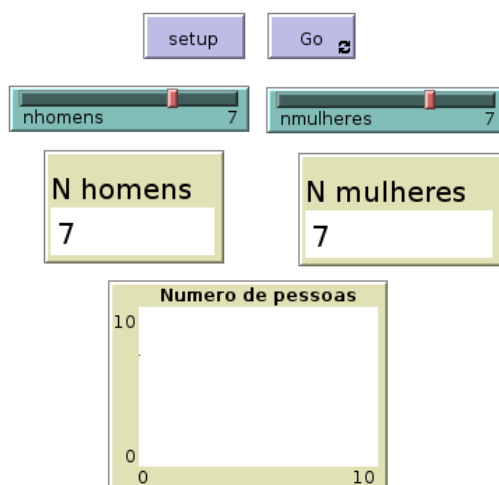
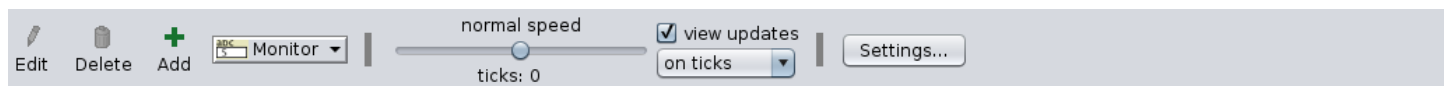
50   if energia <= 0 [           ; se a energia for menor ou igual a 0
51     ifelse breed = mulheres[ ; e se o breed for mulheres
52       set pcolor red]         ; nao precisamos de ask patch-here porque nao estamos a fazer o acesso a nenhuma patch ,quando se
53 altera o pcolor é o pcolor onde o agente esta
54   [set pcolor blue]           ; se nao, significa que a breed sao homens
55   die                           ; e mata-o
56 ]
57 ]
58
59
60 if count turtles = 0 [   ; se ja nao houverem mais agentes ou if not any turtles
61   stop                     ; o programa termina
62 ]
63 tick ; contador de iterações
64
65 end
66
67 to setup-patches
68
69   ask patches[
70     let x random 101 ; estamos atribuir uma probabilidade entre 0 e 100 a x que é uma variavel
71
72     if x < 10[
73       set pcolor green
74     ]
75   ]
76 end
77
78
79 to comer
80
81   if pcolor = green[           ; como o comer esta dentro do ask la em cima, basta vermos se a cor da patch é verde
82     set pcolor black           ; se for metemos a preto(original)
83     set energia energia + 100   ; e incrementamos a energia
84   ]
85
86   end

```

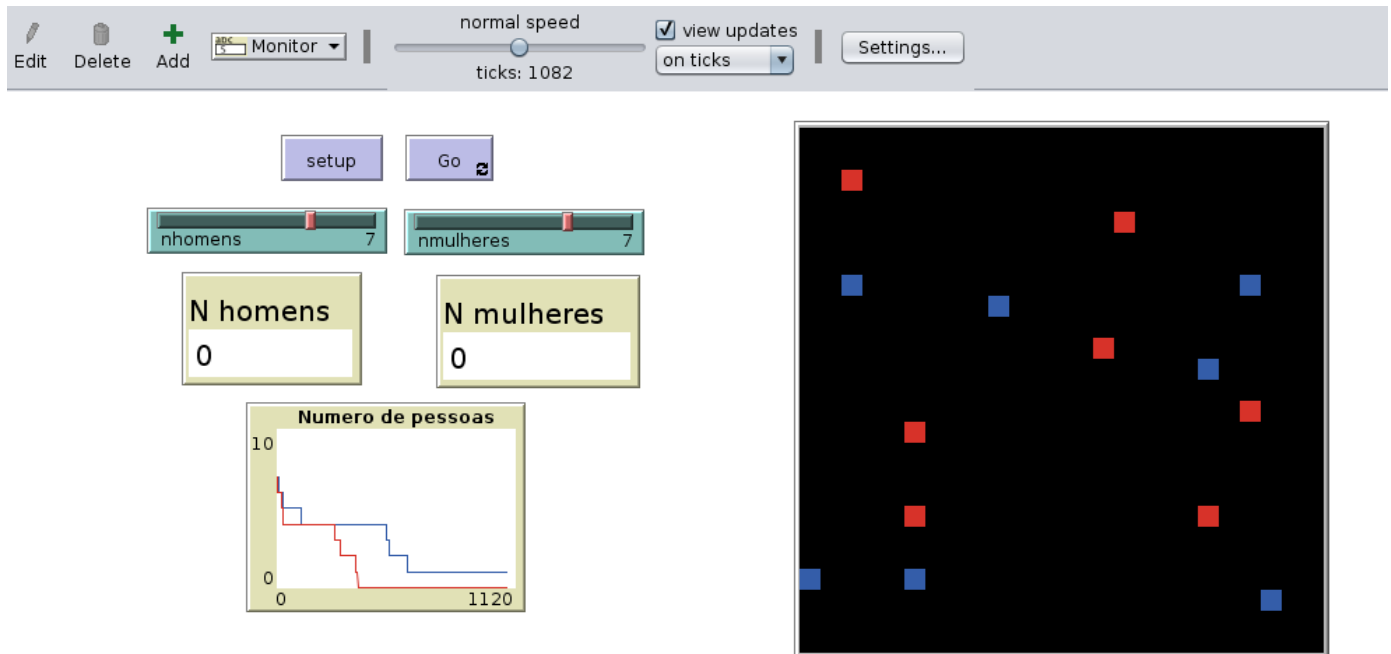
Layout final

Início da experiência

- Começaram 7 homens e 7 mulheres
- Havia 10% do mapa com comida que dava energia às pessoas
- Sempre que um homem andava 1 unidade perdia 1 de energia e a mulher perdia 2
- Caso um agente coma (patches verdes) ganha 100 de energia



Fim da experiência



Resultados da experiência

- No final houve 1082 iterações
- As mulheres foram as primeiras a morrer

Notas do Exercício

- **heading** -> indica a direção em que o agente irá ficar na primeira instância
- **patch-here** -> indica a path debaixo do agente

Se quisermos alterar de alguma forma a patch onde se encontra o agente, não precisamos de fazer "ask patch-here", podemos simplesmente mudar a cor usando o set pcolor red, isto só funciona caso seja a patch onde se encontra o agente.

Fonte

- <http://ccl.northwestern.edu/netlogo/docs/dict/patch-here.html>
- <http://ccl.northwestern.edu/netlogo/docs/dict/heading.html>
- 01-Ficha 1 - Netlogo (IIA_Introducao_Netlogo.pdf) (moodle)
- 01-Ficha 1 - Netlogo (Ficha1_Netlogo.pdf) (moodle)

Ficha 2

Exercício 1 a)

Enunciado

Abra o modelo guardado no ficheiro *IIA_Ficha2_Exercicio1.nlogo*, que está disponível no *Moodle*. Neste modelo é criado apenas um agente (identificado por uma seta), com a capacidade de poder alterar a cor das *patches* imediatamente vizinhas, quando quiser. Em cada iteração, esse agente pode deslocar-se na direção para que a seta aponta ou rodar 90° na direção dos ponteiros do relógio. O modelo já vem com todo o interface criado, i.e., com os botões para a sua inicialização (*Setup*), o deslocamento do agente (*Go*), a rotação do agente (*Turn*) e a reposição da cor inicial do ambiente (*ResetColor*). Tem, ainda, um *chooser*, que indicará quais serão as células vizinhas ao agente que deverão ser pintadas a outra cor, quando pressionado o botão *Paint*;

Resolução

?
1 to setup

```
2    clear-all
3    ; Coloca todos os patches com a cor número 48
4    ask patches
5    [
6        set pcolor 48
7    ]
8    ; Cria um agente
9    create-turtles 1
10   ; Pinta de azul esse agente (de ID 0), dá-lhe a forma de seta e coloca-o no centro do ambiente
11   ;   a apontar para a direita (Heading 0 apontaria para cima e o 90 é a rotação que aplica desde
12   ;   a posição 0, no sentido dos ponteiros do relógio)
13   ask turtle 0
14   [
15       set color blue
16       set shape "arrow"
17       setxy 0 0
18       set heading 90
19   ]
20end
21
22to go
23   ; o único agente do modelo avança uma unidade na direção em que está a apontar
24   ask turtle 0
25   [
26       forward 1
27   ]
28end
29
30to turn
31   ; o único agente do modelo roda 90 graus no sentido do ponteiro do relógio
32   ask turtle 0
33   [
34       left 90
35   ]
36end
37
38to ResetColor
39   ; Coloca todos os patches com a cor número 48
40   ask patches
41   [
42       set pcolor 48
43   ]
44end
45
46;;PROCEDIMENTO PARA COMPLETAR
47to Paint
48   if (Paint_Patches = "Forward")
49   [
50       ask turtle 0
51       [
52           ; pinta a célula vizinha logo à direita da posição do agente (por causa do 1)
53           ask patch-ahead 1
54           [
55               set pcolor 0
56           ]
57       ]
58   ]
59   if (Paint_Patches = "Actual")
60   [
61       ask turtle 0[
62           set pcolor 0
63       ]
64   ]
65
66
67   if (Paint_Patches = "Neighbor_4")
68   [
69       ask turtle 0[
70           ask neighbors4 [set pcolor 0]
71       ]
72   ]
73   if (Paint_Patches = "Neighbor_8")
74   [
75       ask turtle 0[
76           ask neighbors [set pcolor 0]
77       ]
78   ]
79   if (Paint_Patches = "Forward_3")
80   [
81       ask turtle 0[
82           ask patch-left-and-ahead 45 1 [set pcolor 0]
83           ask patch-ahead 1 [set pcolor 0]
84           ask patch-right-and-ahead 45 1[set pcolor 0]
```

```
85 ]
86 ]
87end
```

Exercicio 1 b)

Enunciado

Abra o ficheiro *IIA_Ficha2_Exercicio2.nlogo*, que está disponível no *Moodle*. Neste modelo o agente continua a ter a capacidade de alterar as células na sua vizinhança. As diferenças em relação ao exemplo anterior são:

- O agente não tem uma direção de movimentação explícita. Pode deslocar-se para cima, para baixo, para a esquerda ou para a direita, sem ser necessário efetuar movimentos de rotação, bastando para isso usar os respetivos botões;
- O agente considera sempre a mesma vizinhança de 8 células (baseada na primitiva *neighbors*), também conhecida como vizinhança de *Moore*;
- O ambiente inicial é constituído por células de duas cores.

O agente desloca-se pelo ambiente, tendo como objetivo alterar a cor de todas as células brancas para preto. Sempre que uma célula branca aparece na sua vizinhança o agente pode alterar a sua cor. Em cada iteração só pode ser alterada a cor de uma *patch*. Se, numa determinada iteração, existirem várias *patches* brancas na vizinhança, o agente deve escolher uma delas aleatoriamente para efetuar a modificação. O contador pontos que existe no modelo vai contabilizando as células que mudam de cor.

Resolução

```
?
1 ; Define a variável global scores
2 globals [scores]
3
4 to setup
5   clear-all
6   ; Coloca a variável scores a zero
7   set scores 0
8   ; 5% dos patches fica com a cor branca. os restantes a amarelo
9   ask patches
10  [
11    ifelse random 101 < 5
12    [
13      set pcolor white
14    ]
15    [
16      set pcolor 48
17    ]
18  ]
19  ; Cria um agente
20  create-turtles 1
21  ; Pinta de azul esse agente (de ID 0), dá-lhe a forma de círculo e coloca-o no centro do ambiente
22  ask turtle 0
23  [
24    set color blue
25    set shape "circle"
26    setxy 0 0
27  ]
28end
29
30to Up_
31  ; o único agente do modelo vai para o patch acima da posição em que está
32  ask turtle 0
33  [
34    set ycor ycor + 1
35  ]
36  ; o único agente do modelo verifica a sua vizinhança e se tiver elementos, pinta um deles
37  Verify
38end
39
40to Down_
41  ; o único agente do modelo vai para o patch abaixo da posição em que está
42  ask turtle 0
43  [
44    set ycor ycor - 1
45  ]
46  ; o único agente do modelo verifica a sua vizinhança e se tiver elementos, pinta um deles
47  Verify
48end
49
50to Left_
```

```
51 ; o único agente do modelo vai para o patch esquerdo à posição em que está
52 ask turtle 0
53 [
54     set xcor xcor - 1
55 ]
56 ; o único agente do modelo verifica a sua vizinhança e se tiver elementos, pinta um deles
57 Verify
58end
59
60to Right_
61 ; o único agente do modelo vai para o patch direito à posição em que está
62 ask turtle 0
63 [
64     set xcor xcor + 1
65 ]
66 ; o único agente do modelo verifica a sua vizinhança e se tiver elementos, pinta um deles
67 Verify
68end
69
70; PROCEDIMENTO A COMPLETAR
71to Verify
72  ask turtle 0[
73    if any? neighbors with [pcolor =white]
74    [
75      ask one-of neighbors with [pcolor =white] [set pcolor black]
76      set scores scores +1
77    ]
78
79  ]
80end
```

Retrieved from "[http://zebisnaga.pt/wiki/index.php?title=\(IIA\)_Exercicios_Prática_NetLogo&oldid=282](http://zebisnaga.pt/wiki/index.php?title=(IIA)_Exercicios_Prática_NetLogo&oldid=282)"