

Relatório do Trabalho Prático I

Inteligência Artificial

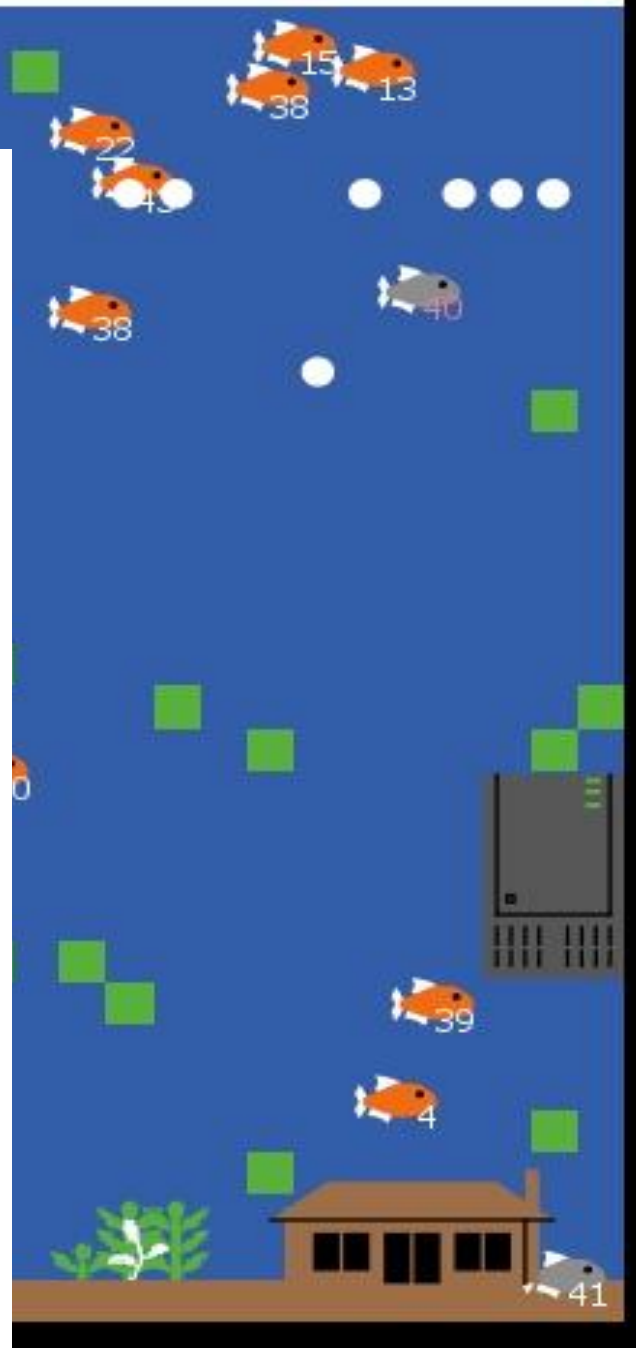
Engenharia Informática

30 de outubro de 2022

Docentes: José Paulo Barroso de Moura Oliveira
Eduardo José Solteiro Pires

Da autoria de: Daniel Oliveira al74575
Pedro Oliveira al73346

utad



Introdução

No âmbito da Unidade Curricular de Inteligência Artificial, foi realizado o trabalho onde se pretende promover a aquisição de conhecimentos e desenvolvimento de competências fundamentais relativas à modelação e simulação computacional de sistemas com agentes racionais utilizando a ferramenta NetLogo.

De modo a realizar esta tarefa, o grupo reuniu-se ao longo destas últimas 3 semanas, onde foram debatidas ideias e metodologias que são evidenciadas no projeto.

Assim, o grupo através de uma pesquisa otimizada, superou as expectativas solicitadas no protocolo, como iremos demonstrar neste relatório.

Índice

Introdução.....	2
Índice	3
1. Fase de Implementação 1	4
2. Fase de Implementação 2	6
3. Conclusão	7
4. Referências.....	8
Anexo 1	9

1. Fase de Implementação 1

Neste projeto, temos como objetivo criar e desenvolver uma simulação de um aquário que evidencie aquilo que vivenciamos num aquário real.

Para tal, através da ferramenta NetLogo, o grupo seguiu todas os parâmetros do protocolo, onde foi criado um aquário com peixes de duas diferentes espécies que tal como um peixe real, alimenta-se, reproduz-se e morre.

Assim como um aquário real, o “nosso aquário” também adquire sujidade, de modo aleatório. Para representação do nível de sujidade, fizemos um gráfico, como podemos verificar na seguinte imagem.

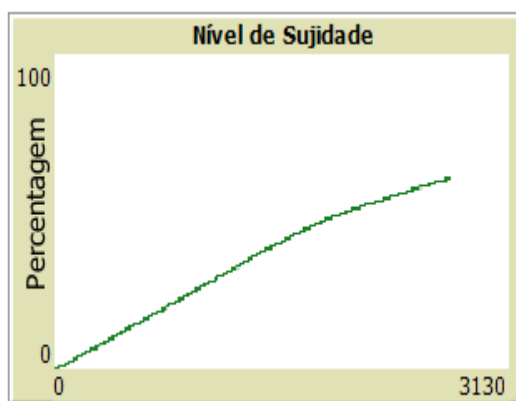


Figura 1: Gráfico do Nível de Sujidade

Para além da sujidade, o nosso modelo contém um botão **Setup** que permite reiniciar o ambiente, um botão **GO**, que permite a inicialização dos movimentos de todos os agentes, um botão **Feed**, que deposita a comida na parte superior do aquário e que por sua vez ao longo de cada tick vai se movimentando uma célula para baixo e que quando esta se encontra na mesma célula do peixe, desaparece, representando assim o peixe a alimentar-se. Implementamos também um gráfico para ver o nível da população, dois deslizadores que permite definir o número de peixes de cada espécie.

Por último, foi implementado nos peixes, uma variável que representa a sua idade e morrer conforme a espécie, como é visível na imagem abaixo.

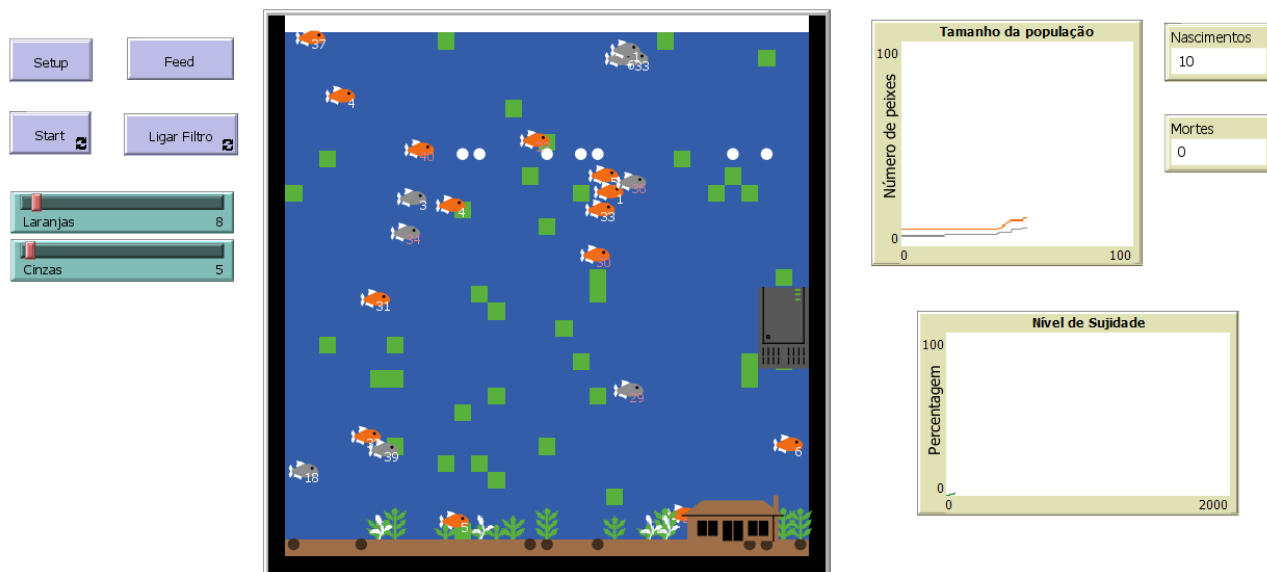


Figura 2: Interface do Projeto

Em baixo, segue-se uma parte do código do nosso trabalho que foi necessário para alcançar os objetivos da primeira fase do protocolo.

```
to setup
  clear-all

  ask patches[      ; laterais e topo (desenho do aquário)
    set pcolor blue
  ]

  if pcolor = max-pcolor[
    set pcolor white
  ]

  if pcolor = -15[
    set pcolor brown
  ]

  if pxcor = max-pxcor or
    pxcor = min-pxcor or
    pycor = min-pycor or
    pxcor = 16 or
    pxcor = -16[
    set pcolor black
  ]

  set lixo 0

  create-fishes Laranjas[
    set shape "fish"
    set color orange
    set size 1.8
    set idade 1 + random 17 ; random idade até aos 18
    set energia 60 + random 49 ; random energia do 60 até as 99
    set label idade
    random-position-dos-peixes
  ]
]
```

Figura 4: Código para o desenho do aquário e a criação dos peixes

```
ask fixes[
  set heading random-float 360

  if [pcolor] of patch-ahead 1 = black or [pcolor] of patch-ahead 1 = white[ ; para fazer os limites do aquário
    right 180
  ]

  if [pcolor] of patch-ahead 1 = 105 or [pcolor] of patch-ahead 1 = 55
  or [pcolor] of patch-ahead 1 = 62 or [pcolor] of patch-ahead 1 = 61 or [pcolor] of patch-ahead 1 = brown[ ; código que impede bug de ir
  if count comida-on patch-ahead 1 = 1 [ ; comer a comida e ficar preso na parede
    move-to min-one-of comida [ distance myself ]
  ]
]

if idade > 20 and energia > 50[
  set label-color pink
]

if idade > 20 and energia < 50[
  set label-color white
]

if count fixes-on patch-ahead 1 = 1 and energia > 50 and idade > 20[ ; tentar criar ovos que descem e só depois aparecem os peixes
  set energia(energia - 50)
  hatch 1[
    set nascimentos nascimentos + 1
    set idade 0
    set energia 30
    set label-color white
  ]
]

if any? comida-here[
  set energia(energia + 50)
]

ask comida-here [
  die
]

set energia(energia - 0.2)
set idade(idade + 0.5)
set label round idade
if idade = 99[
  set mortes mortes + 1
  die
]

forward 1
]
```

Figura 3: Alguns aspetos do código do botão Go

2. Fase de Implementação 2

Como podemos ver na *figura2*, uma das grandes inovações que concebemos, foi a criação de um simulador de um filtro, que limpa o aquário. Este tem como objetivo limpar toda a sujidade que é gerada não só pelos resíduos dos peixes e da comida, mas também pela areia e plantas, o que nos leva à seguinte novidade.

Para tornar o nosso aquário o mais realista possível, criamos um conjunto de plantas, areia e uma casa, que são as típicas decorações usadas num aquário.

Como não queríamos ficar por aqui, demos um passo à frente, e criamos uma condição que só permite aos peixes se reproduzirem a partir de uma determinada idade (20 anos). Para tal, os peixes necessitam de energia (50), que adquirem através da comida e que perdem ao se reproduzir e no seu movimento no meio ambiente.

Para os diferenciar, os peixes cujo a idade está exposta com a cor, cor-de-rosa, estão prontos para procriar.

Para além disso, o agente peixe, quando se encontra na célula adjacente à célula onde se encontra o agente comida, este desloca-se em direção à comida, acabando por se alimentar.

Se o agente comida não for comido e ficar no fundo do aquário, ainda pode ser comido embora este transite para uma nova cor, que representa a comida com resíduos.

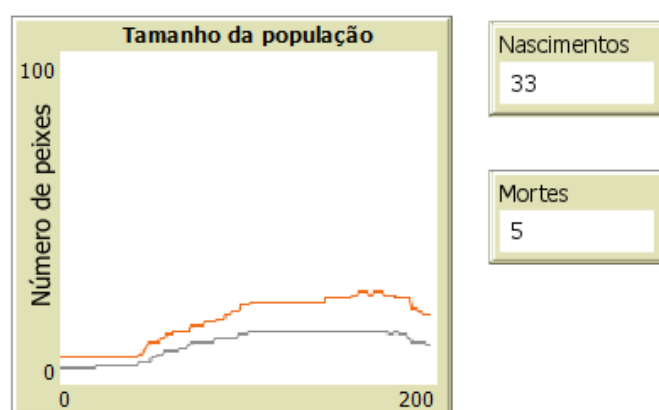


Figura 5: Gráfico em função do número de peixes e Monitores com o número de nascimentos e mortes dos peixes

3. Conclusão

Neste projeto desenvolvemos um ambiente integrado baseado em agentes com o software NetLogo, de modo a simular um aquário, concluímos que com esta ferramenta é possível fazer simulações complexas, mas que no fim acabam por ser muito satisfatórias desenvolver.

Todos os objetivos que tínhamos em mente foram alcançados, no final da realização do projeto ainda surgiram algumas ideias que esperemos poder utilizar num próximo projeto semelhante a este. Como tal todos os conhecimentos que achamos que os docentes expectavam de nós, foram adquiridos e utilizados, sendo que estamos bastante satisfeitos com o nosso desenvolvimento, assim como o nosso primeiro projeto na unidade curricular de seu nome Inteligência artificial.

4. Referências

StackOverflow

Disponível em: www.stackoverflow.com

NetLogo:

Disponível em: <http://www.netlogoweb.org>

Northwestern:

Disponível em: <http://northwestern.edu>

Anexo 1

```
breed [fishes fish]
breed [fixes maisquefixe]
breed [comida comidinha]
breed [grandes plantas]
breed [medias plants]
breed [pequenas plantass]
breed [flores plantes]
breed [casinha casa]
breed [partes filtros]
```

```
globals[nascimentos mortes sujidade]
```

```
fixes-own [idade energia]
fishes-own [idade energia]
```

```
patches-own [lixo] ; em vez de isto vamos usar a global sujidade
para facilitar a construção do gráfico
```

```
to setup
```

```
clear-all
```

```
ask patches[      ; laterais e topo (desenho do aquario)
set pcolor blue
```

```
if pycor = max-pycor[
set pcolor white
]
```

```
if pycor = -15[
set pcolor brown
]
```

```
if pxcor = max-pxcor or
pxcor = min-pxcor or
pycor = min-pycor or
pxcor = 16 or
pxcor = -16[
```

```
set pcolor black
]
```

```
set lixo 0
]
```

```
create-fishes Laranjas[
set shape "fish"
set color orange
set size 1.8
set idade 1 + random 17 ;random idade até aos 18
set energia 60 + random 49 ;random energia do 60 até as
99
set label idade
random-position-dos-peixes
]
```

```
create-fixes Cinzas[
set shape "fish"
set color gray
set size 1.8
set idade 1 + random 17
set energia 60 + random 49
set label idade
random-position-dos-peixes
]
```

```
set sujidade 0
set nascimentos 0
set mortes 0
```

```
fundo
```

```
filtro
```

```
reset-ticks
end
```

```

to go

if ticks > 5 [lixreira]

ask fishes[

  set heading random-float 360

  if [pcolor] of patch-ahead 1 = black or [pcolor] of patch-ahead
1 = white[
    right 180
  ]

  if [pcolor] of patch-ahead 1 = 105 or [pcolor] of patch-ahead 1
= 55
  or [pcolor] of patch-ahead 1 = 62 or [pcolor] of patch-ahead 1
= 61 or [pcolor] of patch-ahead 1 = brown[
    if count comida-on patch-ahead 1 = 1[
      move-to min-one-of comida [ distance myself ]
    ]
  ]

  if idade > 20 and energia > 50[
    set label-color pink
  ]

  if idade > 20 and energia < 50[
    set label-color white
  ]

  if count fishes-on patch-ahead 1 = 1 and energia > 50 and
idade > 20[
;tentar criar ovos que descem e só depois aparecam os peixes
    set energia(energia - 50)

    hatch 1[
      set nascimentos nascimentos + 1
      set idade 0
      set energia 30
      set label-color white
    ]
  ]

  if any? comida-here[
    set energia(energia + 50)
  ]
]

ask comida-here [
  die]

set energia(energia - 0.2)
set idade(idade + 0.5)
set label round idade
if idade = 99[
  set mortes mortes + 1
  die]
forward 1
]

```

```

ask fixes[

  set heading random-float 360

  if [pcolor] of patch-ahead 1 = black or [pcolor] of patch-ahead 1 =
white[ ; para fazer os limites do aquario
    right 180
  ]

  if [pcolor] of patch-ahead 1 = 105 or [pcolor] of patch-ahead 1 = 55
  or [pcolor] of patch-ahead 1 = 62 or [pcolor] of patch-ahead 1 = 61
  or [pcolor] of patch-ahead 1 = brown[ ; código que impende bug de
  ir
    if count comida-on patch-ahead 1 = 1 [
; comer a comida e ficar preso na parede
      move-to min-one-of comida [ distance myself ]
    ]
  ]

  if idade > 20 and energia > 50[
    set label-color pink
  ]

  if idade > 20 and energia < 50[
    set label-color white
  ]

  if count fixes-on patch-ahead 1 = 1 and energia > 50 and idade >
20[ ;tentar criar ovos que descem e só depois aparecam os peixes
    set energia(energia - 50)
    hatch 1[
      set nascimentos nascimentos + 1
      set idade 0
      set energia 30
      set label-color white
    ]
  ]

  if any? comida-here[
    set energia(energia + 50)
  ]
]

ask comida-here [
  die]

set energia(energia - 0.2)
set idade(idade + 0.5)
set label round idade
if idade = 99[
  set mortes mortes + 1
  die]

forward 1
]

```

```
ask comida[
```

```
  if [pcolor] of patch-ahead 1 = black[
    set color 32
    stop
  ]
  forward 0.7
]
```

```
tick
```

```
end
```

```
to alimento
```

```
  create-comida 10[
    set shape "circle"
    set size 0.7
    set color white
    set heading 180
    setxy random-pxcor max-pycor
    if pxcor = max-pxcor or
    pxcor = min-pxcor[
      setxy 0 16
    ]
  ]
end
```

; implementar comida especial para dar mais energia e por sua vez reproduzir mais vezes

```
to random-position-dos-peixes
```

```
  setxy random-pxcor random-pycor
  if pxcor = max-pxcor or
  pxcor = min-pxcor[
    setxy 0 0
  ]
  if pycor = min-pycor or
  pycor = max-pycor[
    setxy 2 2
  ]
end
```

```
to lixeira
```

```
; 1024 patches de agua
; 1024 equivale a 100
; 1 equivale a 0,097
; se cada patch 3 vezes entao arrendamentos feitos a 0.035
```

```
ask one-of patches[
  if pcolor = 105[ ;azul
    set pcolor 55
    set sujidade sujidade + 0.035 ] ;verde claro
]
```

```
ask one-of patches[
  if pcolor = 55[
    set pcolor 62
    set sujidade sujidade + 0.035 ] ;verde escuro
]
```

```
ask one-of patches[
  if pcolor = 62[
    set pcolor 61
    set sujidade sujidade + 0.035 ] ;verde muito escuro
]
```

```
end
```

```
to limpeza
```

```
ask patches[
  if pcolor = 55 or pcolor = 62 or pcolor = 61[
    set pcolor 105]
]
```

```
set sujidade 0
```

```
ask comida[
  if color = 32[
    die]
  ;só limpa a comida quando esta no fundo quando tem cor
  mais escuro
]
```

```
end
```

```
to filtro
  create-partes 1[
    set shape "computer server"
    set size 6
    set color 3 ;cinza mais escuro
    setxy 14 -2
  ]
end
```

```
to fundo
  create-grandes 6[
    set shape "plant"
    set color green
    set size 2
    setxy random-pxcor -13.5
    if pxcor = max-pxcor or
    pxcor = min-pxcor[
      setxy 0 -13.5]
  ]
end
```

```
create-medias 4[
  set shape "plant medium"
  set color green
  set size 2
  setxy random-pxcor -13.5
  if pxcor = max-pxcor or
  pxcor = min-pxcor[
    setxy 5 -13.5]
  ]
end
```

```
create-pequenas 8[
  set shape "plant small"
  set color green
  set size 2
  setxy random-pxcor -13.5
  if pxcor = max-pxcor or
  pxcor = min-pxcor[
    setxy -14 -13.5]
  ]
end
```

```
create-flores 5[
  set shape "flower budding"
  set color white
  set size 2
  setxy random-pxcor -13.5
  if pxcor = max-pxcor or
  pxcor = min-pxcor[
    setxy 9 -13.5]
  ]
end
```

```
create-casinha 1[
  set shape "house ranch"
  set color brown
  set size 6
  setxy 11 -12.5
  ]
end
```