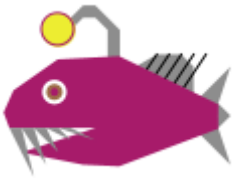
utad UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO

Licenciatura em Engenharia Informática

Inteligência Artificial

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

TRABALHO PRÁTICO 1: AGENTES



Discentes:

Eduardo Luís Ferreira Ramos	al74321
Rafael António Gonçalves Ferreira	al72951

Vila Real, 2022

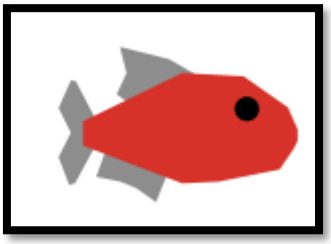
utad

Índice

Introdução

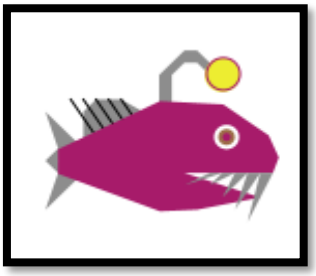
Para a realização do presente trabalho pretendeu-se promover a aquisição de conhecimentos e desenvolvimento de competências fundamentais relativas à modelação e simulação computacional de sistemas com agentes racionais utilizando a ferramenta NetLogo. Neste sentido, realizamos o projeto procurando atender a todos os elementos avaliativos obrigatórios e facultativos, procurando também utilizar a criatividade para desenvolver uma simulação funcional e divertida.

- **Agente [fishes] "Peixes-Laranja"**



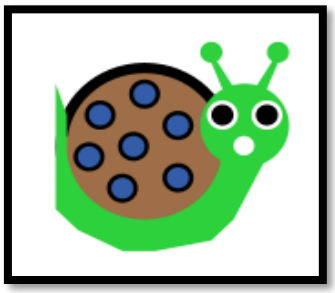
É uma das espécies de peixes desenvolvidas para a simulação. Em particular, das espécies desenvolvidas, é a única que se movimenta em cardume.

- **Agente [lamps] "Peixes-Lanterna"**



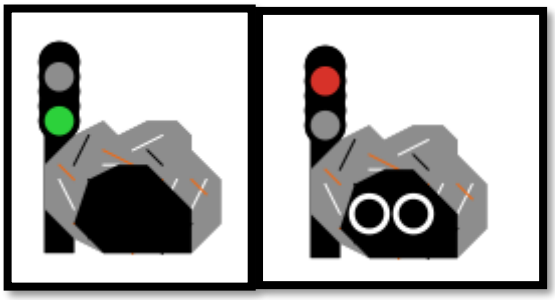
É uma das espécies de peixes desenvolvidas para a simulação.

- **Agente [snails] “Caracóis”**



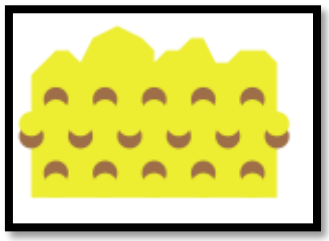
É uma espécie de caracol desenvolvidas para a simulação. Serve para limpar as “carcaças” dos peixes mortos e depositar nutrientes.

- **Agente [rocks] “Pedras”**



É uma espécie desenvolvida para a simulação. Serve para os peixes se esconderem quando passam por esse agente. Se o peixe estiver escondido a imagem passa a ser a que se encontra à direita.

- **Agente [sands] “Areia”**

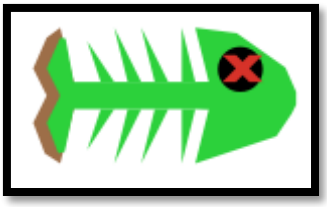


É uma espécie desenvolvida para a simulação. Não tem nenhuma funcionalidade em particular a não ser estética.

- **Agente [cleaners] “Peixe-Pleco”**

É uma das espécies de peixes desenvolvidas para a simulação. Serve para limpar a água dos resíduos deixados pelos dejetos e pela comida.

- **Agente [corpses] “Peixes-Mortos”**



É uma das espécies de peixes desenvolvidas para a simulação. Serve para sinalizar a morte de um peixe e para os caracóis terem a possibilidade de os transformar em nutrientes.

- **Agente [foods] “Comida”**

É uma espécie desenvolvida para a simulação. Serve para alimentar os peixes.

- **Agente [dungs] “Dejetos”**

É uma espécie desenvolvida para a simulação. Serve para transformar o alimento consumido em dejeto, que posteriormente polui a água.

- **Agente [nutrientes] “Nutriente”**

É uma espécie desenvolvida para a simulação. Serve para limitar o envelhecimento dos peixes quando estes os consomem.

Elementos Avaliativos

Por questões de organização de código, decidimos separar a função “to setup” num ficheiro à parte “to_setup.nls”, as várias rotinas utilizadas em outro ficheiro à parte “rotinas.nls”, e a função “to go” no ficheiro principal da simulação “traballho_final.nlogo”.

```
__includes [ "to_setup.nls" "rotinas.nls"]
```

Assim, apresentamos as principais funções (mais cruciais para o funcionamento da simulação) invocadas no ficheiro principal:

```
ask fishes ;----------------------------------------
[
  set fishes_breed 1
  verifica_heading
  escolhe_lider
  define_info_lider
  recolhe_info_lider_cardume
  muda_direcao_aleatoriamente 50000 270 -90
  envelhecer 250
  verifica_condicoes_morte size age 29 6 15
  verifica_condicoes_alimentacao last_meal 5000 -1
  if (fish_hidden = false)
  [
    verifica_limites_aquario
    forward 1
  ]
  verifica_nutrientes
  procreate
  if ((age > 2) and (color = 65))
  [
    set color red
  ]
]
```

```
ask lamps ;----------------------------------------
[
  set fishes_breed 2
  verifica_heading
  muda_direcao_aleatoriamente 50000 270 -90
  envelhecer 250
  verifica_condicoes_morte size age 29 6 15
  verifica_condicoes_alimentacao last_meal 5000 -1
  if (fish_hidden = false)
  [
    verifica_limites_aquario
    forward 1
  ]
  verifica_nutrientes
  procreate
  if ((age > 2) and (color = 65))
  [
    set color magenta
  ]
]
```

```
ask foods ;----------------------
[
  deslocamento_aleatorio_foods
  verifica_comido
  set food_age food_age + 1 ;---
  verifica_decomposicao
]
```

```
ask dungs ;-------------------
[
  set dung_age dung_age + 1 ;-
  suja_na_passagem
  desintegrar
]
```

```
ask snails ;----------------------------------------
[
  forward 0.05 ;-------------------------------------
  muda_direcao_aleatoriamente 500 1.5 one-of [-90 90]
  limites_caracois
  shape_caracois
  set can_clean 0;
  ask patch-here ;-----------------------------------
  [
    if count corpses-here = 1 ;----------------------
    [
      ask corpses-here
      [
        set xx xcor
        set yy ycor
        set ok 1
        die
      ]
    ]
  ]
]
```

```
ask cleaners
[
  forward 0.1 ;--------------------------------------
  muda_direcao_aleatoriamente 500 360 0
  limpar_lixo
  (ifelse ;------------------------------------------
      pycor = 0 ;--------------------------------
        [
          muda_direcao_aleatoriamente 1 180 90
        ]
      pycor = min-pycor + 6 ;--------------------
        [
          muda_direcao_aleatoriamente 1 180 -90
        ])
]
```

```
ask rocks
[
  shape_pedra
  hide_under_rock
  get_out_rock
]
```

```
ask seaweeds
[
  ask patches in-radius 10
  [
    set lixo 0
  ]
```

De seguida, apresentamos a lista de funções (acima mostradas) e as suas funcionalidades:

- **"to feed"**
  - Cria a comida, conforme a quantidade
- "**to pollute"**
  - Polui a água
- "**to pass_away"**
  - Cria um corpo aonde o peixe morreu
- **"to convert_corpses"**
  - Criar os nutrientes após os corpses morrerem
- **"to-report hipotenusa"**
  - Dá a dimensão da hipotenusa para encontrar o peixe-líder
- **"to procreate"**
  - Caso se encontrem dois peixes, na mesma célula, e cumpram as condições necessárias, reproduzem-se
- **"to muda_direcao_aleatoriamente"**
  - A cada X ticks, mudam de direção aleatoriamente
- **"to envelhecer"**
  - A cada X ticks, os peixes ficam mais velhos
- **"to verifica_condicoes_morte"**
  - Se os peixes tiverem menos do tamanho minimo, ou mais do que o tamanho máximo, morrem
- **"to verifica_condicoes_alimentacao"**

    - Se a ultima alimentação do peixe, tiver sido à mais de X ticks, fica mais magro
- **“to verifica_heading”**
    - Conforme a direção do peixe, muda a forma
- **“to verifica_limites_aquario”**
    - Caso o peixe esteja nos limites do aquário, muda de direção
- **“to escolhe_lider”**
    - Escolhe o peixe mais perto do ponto (0,0) para ser o peixe-lider
- **“to define_info_lider”**
    - Caso exista peixe-lider, coloca uma coroa no mesmo para o identificar
- **“to recolhe_info_lider_cardume”**
    - Os peixes que se encontrarem a X células do peixe lider, seguem-no
- **“to verifica_nutrientes”**
    - Caso um peixe coma um nutriente, fica mais novo
- **“to turn_dead”**
    - Caso o peixe esteja morto, chama a função PASS_AWAY
    - O corpo vai caindo para o fundo do aquário
- **“to deslocamento_aleatorio_foods”**
    - Faz com que a comida ande para a direita e esquerda, e não só para baixo
- **“to verifica_comido”**
    - Caso um peixe e uma comida estejam na mesma célula, o peixe come a comida e engorda
- **“to verifica_decomposicao”**
    - Se a comida não for comida em certo tempo, decompõem-se e polui o aquário
- **“to suja_na_passagem”**
    - Por onde o dejeto passa, escurece a água e fica consequentemente, mais poluída
- **“to desintegrar”**
    - Se o local onde os dejetos/comida deviam poluir já estiver poluído, a poluição vai para cima (em Y)
- **“to limites_caracois”**
    - Os caracóis só andam entre -79 e -85 em Y
- **“to shape_caracois”**
    - Muda a forma dos caracóis conforme a direção

- **“to limpar_lixo”**
  - Limpa o lixo de algumas células, chamando a função 'POLLUTE'
- **“to shape_pedra”**
  - Muda a forma da pedra
- **“to hide_under_rock”**
  - Caso não tenha nenhum peixe na pedra e esteja um peixe em cima da mesma, esconde-se
- **“to get_out_rock”**
  - Caso um peixe já esteja à mais de 9999 ticks, expulsa-o da pedra
- **“to aumenta-temperatura”**
  - Vai aumentando a temperatura das células, aleatoriamente
- **“to Trocas”**
  - Vai fazendo a média das temperaturas e atualizando-as, duas a duas, em espiral. Usa o procedimento 'TROCA-PATCHES'
- **“to switches”**
  - Se o chooser patches_mostra? mudar o valor, atualiza o que mostra nos patches, a temperatura ou qualidade da água
  - Se o switch resistencia_ligada? estiver a 'On' e a média da temperatura seja inferior ou igual a 20, liga a resistencia e coloca as células em volta na temperatura máxima, senão desliga
  - Se o switch bomba_ligada? estiver a 'On', chama o procedimento Trocas

Conclusões

A realização do presente trabalho foi produtiva para a aprendizagem em relação ao trabalho com agentes e como se torna possível realizar simulações complexas através da utilização de um software como o Netlogo.

Apresar de trabalhosa, consideramos que a realização deste trabalho foi bastante enriquecedora.

Juntamente com o presente relatório segue, os ficheiros desenvolvidos para a realização do projeto, bem como um vídeo ilustrativo do funcionamento do mesmo.