



RELATÓRIO

Trabalho Prático 2

Introdução à Inteligência Artificial

Licenciatura em Engenharia Informática



O segundo trabalho prático consiste em desenvolver um programa em C capaz de resolver o Problema do Conjunto Estável Máximo.

Trabalho realizado por:

Tomás Gomes Silva - 2020143845

Tomás da Cunha Pinto - 2020144067

Índice

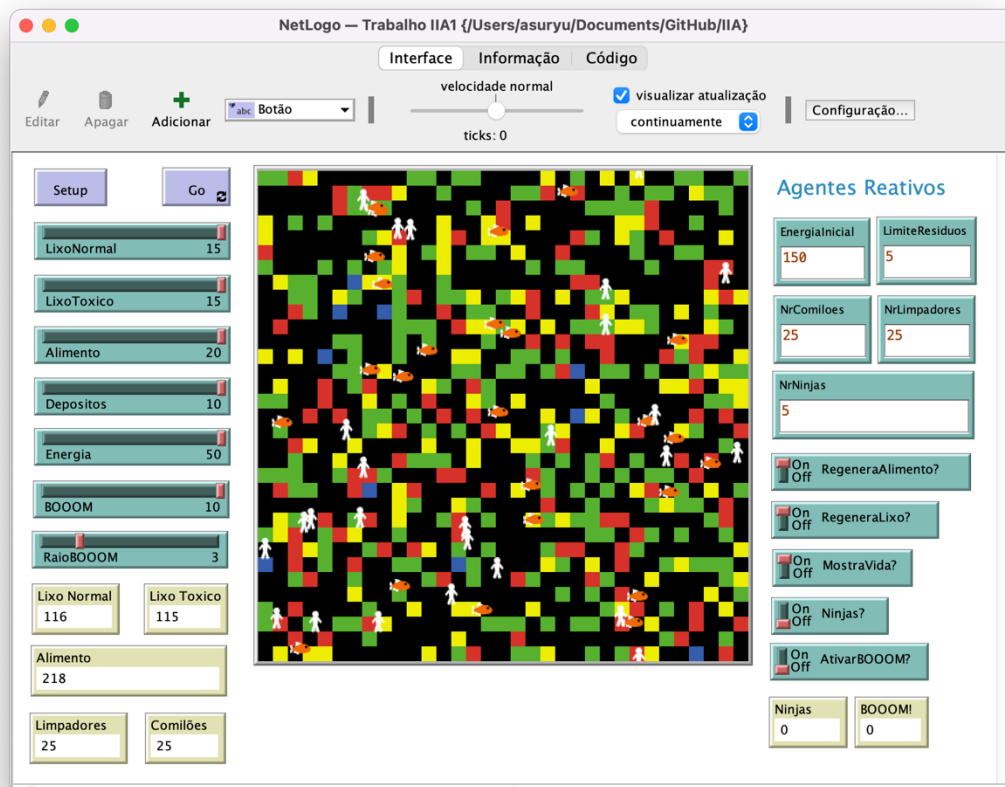
Índice	3
Introdução.....	4
Implementação	5
Modelo Base:	5
Modelo Melhorado:	6
Ninjas.....	6
Células Explosivas (BOOOM!).....	6
Análise de Resultados.....	7
Modelo Base:	7
Alterar a energia que os agentes ganham ao comer	7
Alterar a energia inicial dos agentes	7
Alterar o número de Limpadores	8
Modelo Melhorado:	8
Alterar o número de células explosivas	8
Alterar o número de ninjas existentes	9
Alterar o raio de explosão das células explosivas	9
Conclusão.....	10

Introdução

O primeiro trabalho prático de [Introdução à Inteligência Artificial](#) consiste na criação de uma simulação com agentes reativos.

Foram criados dois modelos distintos: o **modelo base**, onde implementámos todas as características que foram pedidas e o **modelo melhorado**, onde criámos duas novas funcionalidades de modo a alterar o desempenho dos agentes.

Com ambos os modelos criados realizámos também três experiências para cada modelo para podermos olhar para os dados e perceber o que é que muda consoante os parâmetros que alteramos.



Implementação

Modelo Base:

No modelo base implementámos a criação do ambiente da simulação que consistiu na criação dos agentes e preenchimento do tabuleiro com *patches* de cores variadas correspondentes aos vários elementos (comida, lixo, etc..).

```
; Cor inicial do tabuleiro
ask patches [set pcolor black]

; Preenchimento do tabuleiro com as cores
ask patches with [pcolor = black]
[
  if random 101 < LixoNormal
  [
    set pcolor yellow
  ]
  if random 101 < LixoToxico
  [
    set pcolor red
  ]
  if random 101 < Alimento
  [
    set pcolor green
  ]
]

; Criação dos depósitos de acordo com os valores
set contador 0
while [contador < Depositos]
[
  ask one-of patches with [pcolor = black]
  [
    set pcolor blue
  ]
  set contador contador + 1
]
if AtivarB000M?
[
  set contador 0
  while [contador < B000M]
  [
    ask one-of patches with [pcolor = black]
    [
      set pcolor 125
    ]
    set contador contador + 1
  ]
]

; Criação dos agentes Comilões
ask comiloes
[
  set size 1.5
  set color 25
  set shape "fish"
  set heading 0
  setxy random-xcor random-ycor
]

; Criação dos agentes Limpadores
ask limpadores
[
  set size 1.5
  set color 69.9
  set shape "person"
  set heading 0
  set limite LimiteResiduos
  set nr_residuos 0
  setxy random-xcor random-ycor
]
```

Para além disso, criámos também procedimentos para controlar o comportamento dos Comilões e dos Limpadores.

```
ask comiloes
[
  ifelse [pcolor] of patch-here = green
  [
    set pcolor black
    set energy energy + Energia
    if regeneraAlimento? [
      ask one-of patches with [pcolor = black]
      [
        set pcolor green
      ]
    ]
  ]
]
[
  ifelse [pcolor] of patch-here = red or [pcolor] of patch-here = yellow
  [
```

O procedimento **Go** tem como função avançar um tick e retirar uma unidade de vida a todos os agentes. Já o procedimento **VerificaMorte** trata de matar os agentes que tiverem uma vida menor ou igual a zero.

Modelo Melhorado:

No que toca ao modelo melhorado, o nosso grupo de trabalho decidiu criar duas novas funcionalidades:

- Ninjas
- Células Explosivas (BOOOM!)

Ninjas

Os **Ninjas** são agentes que aparecem para destabilizar o normal funcionamento da simulação. O trabalho destes agentes é encontrar alimentos e transformá-los em lixo normal (chance 80%) ou lixo tóxico (chance 20%).

Se os ninjas forem vistos por algum Comilão (ser visto significa estar ao lado de um agente) perdem energia equivalente à energia inicial a dividir por 10.

O número de Ninjas é configurável pelo utilizador com qualquer valor.

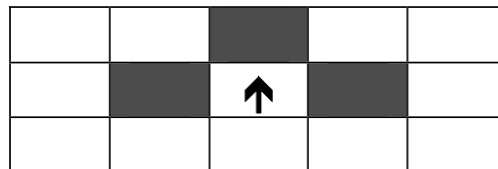


Figura 1 - Percepção dos Ninjas

Células Explosivas (BOOOM!)

As **Células Explosivas (BOOOM!)** são células roxas que quando são calcadas por qualquer agente explodem todo o lixo ao redor da célula num raio configurável. Todos os agentes menos os ninjas têm percepções programadas para este tipo de células de acordo com as suas próprias percepções.

O número de células existentes e o raio de explosão é configurável com valores entre 1 e 10.

```
if AtivarBOOOM?
[
  set contador 0
  while [contador < BOOOM]
  [
    ask one-of patches with [pcolor = black]
    [
      set pcolor 125
    ]
    set contador contador + 1
  ]
]
```

Para criar estas células definimos um contador que vai correr até ao número de células configuradas pelo utilizador na interface e usamos o **ask one-of patches** para selecionar uma célula aleatória para a pintarmos com a cor roxa.

Análise de Resultados

Modelo Base:

No modelo base efetuámos 10 repetições (com condição de paragem nos 2500 ticks) de várias experiências com parâmetros alterados para testar a capacidade de sobrevivência dos agentes.

Os testes relativos ao modelo base são:

- Alterar a energia que os agentes ganham ao comer
- Alterar a energia inicial dos agentes
- Alterar o número de Limpadores

Alterar a energia que os agentes ganham ao comer

Esta experiência consiste em analisar se ao fim de 2500 ticks sobrevivem mais agentes consoante a energia que eles ganham ao consumir um alimento. Os valores de energia que utilizámos são 10, 25 e 50.

Será que aumentar a energia que os agentes ganham ao comer um alimento vai ter um impacto grande na capacidade de sobrevivência dos agentes?

Experiência 1 - Analisar a sobrevivência dos agentes ao final de 2500 ticks alterando a energia que estes ganham																								
Número de Agentes	Energia Ganha	Média de agentes vivos no final	% de Repetições com extinção	Média de Ticks	Exp1		Exp2		Exp3		Exp4		Exp5		Exp6		Exp7		Exp8		Exp9		Exp10	
					Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks
50	10	0	100%	460,5	0	628	0	306	0	475	0	331	0	456	0	424	0	373	0	442	0	491	0	679
	25	0	100%	1259,5	0	682	0	1238	0	1692	0	1417	0	1046	0	993	0	1074	0	1524	0	993	0	1936
	50	1,8	40%	2317,3	6	2500	0	1998	2	2500	3	2500	0	2082	3	2500	2	2500	0	1886	2	2500	0	2207

Olhando para os dados que se encontram na tabela, podemos reparar que: com mais energia ganha por alimento, os agentes aguentam-se mais tempo e aumenta também a chance de ao fim dos 2500 ticks ainda estarem vivos.

Alterar a energia inicial dos agentes

Desta vez, experimentámos alterar os valores de energia com que cada um dos agentes começa. Os valores de energia inicial que utilizámos são 100, 150 e 200.

Será que aumentar a energia inicial dos agentes vai permitir que estes sobrevivam mais tempo?

Experiência 2 - Analisar a sobrevivência dos agentes ao final de 2500 ticks alterando a sua energia inicial																								
Número de Agentes	Energia Inicial	Média de agentes vivos no final	% de Repetições com extinção	Média de Ticks	Exp1		Exp2		Exp3		Exp4		Exp5		Exp6		Exp7		Exp8		Exp9		Exp10	
					Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks
50	100	2,4	30%	2269,8	4	2500	0	1709	3	2500	0	1451	3	2500	0	2038	1	2500	5	2500	2	2500	6	2500
	150	1,5	40%	2301,5	2	2500	5	2500	0	1591	0	2373	0	1712	2	2500	4	2500	0	2339	1	2500	1	2500
	200	2,5	20%	2413,1	0	2103	1	2500	6	2500	4	2500	1	2500	2	2500	7	2500	2	2500	2	2500	0	2028

Como podemos observar, alterar a energia inicial não afeta a capacidade de sobrevivência dos agentes.

Alterar o número de Limpadores

A última experiência do modelo base foca-se em alterar o número de Limpadores para averiguar se estes têm um impacto grande na sobrevivência dos agentes como um todo. Testámos isto com 10, 25 e 50 Limpadores.

Será que aumentar o número de Limpadores vai ajudar a recolher o lixo mais rápido e consequentemente aumentar a capacidade de sobrevivência dos agentes?

Experiência 3 - Analisar a sobrevivência dos agentes ao final de 2500 ticks alterando o número de limpadores																								
Número de Agentes	Nº de Limpadores	Média de agentes vivos no final	% de Repetições com extinção	Média de Ticks	Exp1		Exp2		Exp3		Exp4		Exp5		Exp6		Exp7		Exp8		Exp9		Exp10	
					Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks
25 + Limpadores	10	1,4	50%	2215,6	3	2500	2	2500	0	1049	0	2368	0	2466	0	1460	4	2500	1	2500	0	2313	4	2500
	25	1,1	50%	2281,1	1	2500	1	2500	1	2500	0	2206	0	2456	5	2500	0	1440	0	2002	3	2500	0	2207
	50	1,7	20%	2448,8	0	2401	2	2500	0	2087	2	2500	2	2500	1	2500	1	2500	5	2500	1	2500	3	2500

Apesar dos Limpadores terem uma tarefa importante de limpar o lixo do mapa, parece que isso não afeta muito a quantidade de agentes que sobrevive ao fim de 2500 ticks. Porém, com 50 Limpadores houve menos repetições com extinção total dos agentes pelo que se pode afirmar que estes amiguinhos que tratam de limpar as ruas da nossa simulação não são assim tão inúteis!

Modelo Melhorado:

No modelo melhorado quisemos testar as nossas implementações originais como as células explosivas e os ninjas. Para isso, efetuámos 10 iterações de várias experiências de modo a observar a sobrevivência dos agentes ao fim de 2500 ticks.

Os testes relativos ao modelo melhorado são:

- Alterar o número de células explosivas
- Alterar o número de ninjas existentes
- Alterar o raio de explosão das células explosivas

Alterar o número de células explosivas

Na primeira experiência do modelo melhorado começámos por alterar o número de células explosivas existentes no mapa. Na interface temos um *slider* que nos permite alterar este valor entre 1 e 10 e nesta experiência. Para esta experiência em concreto, colocámos espalhadas no mapa 1, 5 e 10 células explosivas.

Será que aumentar o número de células explosivas vai ajudar os agentes a sobreviverem mais tempo?

Experiência 1 - Analisar a sobrevivência dos agentes ao final de 2500 ticks alterando o número de células BOOOM!																									
Número de Agentes	Células BOOOM!	Média de agentes vivos no final	% de Repetições com extinção	Média de Ticks	Exp1		Exp2		Exp3		Exp4		Exp5		Exp6		Exp7		Exp8		Exp9		Exp10		
					Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos
50	1	1,9	20%	2436	2	2500	2	2500	1	2500	2	2500	2	2500	4	2500	0	2461	3	2500	0	1899	3	2500	
	5	2,4	50%	2341,9	11	2500	2	2500	0	2317	1	2500	6	2500	0	2334	0	1887	0	2344	4	2500	0	2037	
	10	4,3	10%	2499,4	5	2500	1	2500	5	2500	2	2500	10	2500	0	2494	12	2500	1	2500	6	2500	1	2500	

Como as células explosivas limpam apenas o lixo num raio predefinido isto ajuda os agentes a não morrerem para o lixo. Isto explica o facto da média de agentes vivos no final dos 2500 venha aumentar com mais células explosivas.

Alterar o número de ninjas existentes

Uma das nossas implementações para o método melhorado foi a criação de ninjas capazes de roubar a comida e trocá-la por um dos dois tipos de lixo.

Neste teste, experimentámos colocar 1, 3 e 5 ninjas ativos na simulação para observar a capacidade de sobrevivência dos restantes agentes mesmo com os ninjas a destabilizar.

Será que aumentar o número de ninjas existentes vai atrapalhar os restantes agentes de modo que estes sobrevivam menos tempo?

Experiência 2 - Analisar a sobrevivência dos agentes ao final de 2500 ticks alterando o número de ninjas existentes																								
Número de Agentes	Nº de Ninjas	Média de agentes vivos no final	% de Repetições com extinção	Média de Ticks	Exp1		Exp2		Exp3		Exp4		Exp5		Exp6		Exp7		Exp8		Exp9		Exp10	
					Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks
50 + Ninjas	1	0,3	80%	1771,7	0	1540	0	1399	0	1110	0	1125	1	2500	2	2500	0	2137	0	1934	0	1200	0	2272
	3	1	30%	748,8	0	975	1	835	2	594	1	675	0	971	1	714	0	1024	1	675	2	525	2	500
	5	1,4	50%	658,9	4	500	0	920	0	699	4	525	2	625	2	593	2	650	0	625	0	597	0	855

Estranhamente, a média de agentes vivos no final de cada iteração não ficou muito abalada com a presença de mais ninjas, mas algo que é bastante notório é a reduzida quantidade de ticks necessários para que todos os agentes morram. Ou seja, com mais ninjas os agentes aguentam, em média, menos tempo na simulação.

Alterar o raio de explosão das células explosivas

Para a nossa última experiência, quisemos testar algo relacionado com a funcionalidade do modelo melhorado em que mais gostámos de trabalhar: as células explosivas. A primeira experiência deste modelo também era relacionada com este tipo de células, mas, enquanto essa dizia respeito a número de células explosivas espalhadas no mapa, esta foca-se no raio de explosão destas mesmas células.

Experimentámos com um raio de explosão de 3, 5 e 10 unidades, respetivamente.

Será que aumentar o raio de explosão das células explosivas vai fazer diferença no que toca à sobrevivência dos agentes?

Experiência 3 - Analisar a sobrevivência dos agentes ao final de 2500 ticks alterando o raio de explosão das células BOOOM!																								
Número de Agentes	Raio de Explosão	Média de agentes vivos no final	% de Repetições com extinção	Média de Ticks	Exp1		Exp2		Exp3		Exp4		Exp5		Exp6		Exp7		Exp8		Exp9		Exp10	
					Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks	Vivos	Ticks
50	3	8,4	0%	2500	8	2500	8	2500	5	2500	3	2500	7	2500	6	2500	13	2500	9	2500	14	2500	11	2500
	5	14,1	0%	2500	10	2500	14	2500	10	2500	17	2500	17	2500	12	2500	18	2500	16	2500	16	2500	11	2500
	10	17,5	0%	2500	15	2500	19	2500	20	2500	20	2500	17	2500	17	2500	16	2500	17	2500	14	2500	20	2500

Como era de esperar, nota-se alguma diferença na capacidade de sobrevivência dos agentes visto que existe menos lixo para “atrapalhar”. Com um aumento do raio de explosão verificamos um aumento da média de agentes vivos no final de cada simulação. Algo a notar é que não houve nenhuma repetição com extinção pois limitámos o número de ticks máximo em 2500. Se a simulações tivessem corrido mais tempo poderíamos ter obtido um número mais preciso.

Conclusão

Este trabalho permitiu-nos consolidar bem a matéria das aulas de Introdução à Inteligência Artificial relativas ao **NetLogo** e dessa forma pudemos colocar-nos à prova relativamente àquilo que realmente sabíamos e também pudemos aplicar os nossos conhecimentos.

Por se tratar de uma simulação com um grande carácter visual e que de certa forma se assemelha a um jogo, creio que isso nos deu uma grande motivação para realizarmos este trabalho prático.

