



Licenciatura em Engenharia Informática *Bachelor of Science in Informatics Engineering*

Relatório de trabalho prático *Trabalho prático n- 1*

Agentes Racionais

Vasco Oliveira / 2019119011
João Rodrigues/ 2018018040

19/10/2019

Índice / Table of Contents

1. Introdução	3
2. Implementação.....	3
2.1) Descrição do modelo base	3
2.1.1) Inicialização	3
2.1.2) Movimento	3
2.1.2) Consumo	4
2.1.3) Funções de suporte	4
2.2) Comportamentos adicionais implementados	4
2.3) Ambiente Gráfico	5
3. Análise de resultados.....	5
4. Conclusão	6
Referências / <i>References</i>	7

1. Introdução

Neste projeto procura-se implementar agentes reativos num ambiente e analisar os seus comportamentos consoante a alteração das características dos mesmos.

O ambiente é bidimensional e tem várias características, as quais podem ser configuradas manualmente em cada teste. Por exemplo: a quantidade de energia ganha pelo agente na ingestão de alimentos, a quantidade de lixo e de alimento que pode ser encontrado no ambiente e a quantidade de depósitos do lixo existentes neste. Para além disso tanto os alimentos como o lixo voltam a reaparecer ao longo da simulação.

Os agentes comilões têm como função procurar alimento de maneira a sobreviver ao ambiente, perdendo energia com o seu movimento e ganhando-a de volta na ingestão de alimento, podendo também morrer caso encontrem lixo.

Os agentes limpadores têm como principal objetivo colecionar o lixo dispersado ao longo do modelo e a distribuí-lo pelos depósitos encontrados no ambiente.

No capítulo de implementação está descrito o comportamento dos diversos elementos do modelo assim como as interações entre eles, os resultados e conclusões tiradas ao decorrer do trabalho estão detalhados devidamente no capítulo de análise de resultados e de Conclusão.

2. Implementação

2.1) Descrição do modelo base

*O programa foi implementado com shapes próprias para fins gráficos, antes de utilizar é recomendado a aceder ao menu **Turtle Shapes Editor** e importar as shapes “person contruction”, “person lumberjack” e “monster”.*

2.1.1) Inicialização

No modelo base foram criados dois tipos de agentes: limpadores e comilões, estes agentes possuem uma variável chamada de **energia-agente** que corresponde á energia que vão possuir ao longo do modelo. Os limpadores possuem também **pack** que se refere á quantia de lixo que estão a carregar, incrementa 1 por lixo normal, 2 por lixo tóxico.

Foram criados quatro tipos de patches, depósitos (azul), lixo normal (amarelo), lixo tóxico (vermelho), comida (alimento). As patches possuem **lixo-depositado**, variável referente á quantidade de lixo depositada nas patches de depósito, quando estas estiverem cheias irão mudar de cor para violeta, de modo a indicar aos limpadores para não depositar mais.

2.1.2) Movimento

O algoritmo de movimento desenvolvido varia de agente para agente, os comilões têm três funções (**trilho-frente**, **trilho-direita**, **trilho-esquerda**) cuja função é identificar a cor de uma das patches e agir de acordo com esta. Se for comida avançar, lixo fugir numa outra direção assim como perder uma certa quantidade de energia. Se não encontrar nada á frente irá verificar á direita, se nada estiver á direita irá verificar á esquerda, em último caso de não encontrar nada irá mover se numa direção aleatória de modo a tentar encontrar uma das patches.

O algoritmo de movimento do limpador pode ser descrito como se perceber a frente lixo ou comida avança, caso contrário procura a direita, em último caso irá andar aleatoriamente para a frente ou para a direita.

2.1.2) Consumo

Relativamente a comida, se algum agente se encontrar em cima de uma patch verde irá ganhar a energia definida para alimento, um limpador com mais de metade do limite de transporte de lixo irá ganhar apenas metade da energia definida para o alimento.

O consumo de lixo dos limpadores irá depender se têm o **pack** cheio ou não, caso esteja irá procurar um depósito para esvaziar o pack, caso não esteja irá adicionar 1 ao pack se for lixo normal, 2 se for lixo tóxico; se o valor do adicionado exceder o limite ele irá ignorar o lixo.

2.1.3) Funções de suporte

Foram implementadas várias funções que reportam o número dos diversos elementos do modelo para facilitar a análise da parte do observador assim como facilitar a implementação de funções de geração de patches.

Quando o lixo e alimento atingem valores inferiores a percentagem escolhida, serão gerados novos patches em localizações aleatórias de modo a voltar aos valores designados.

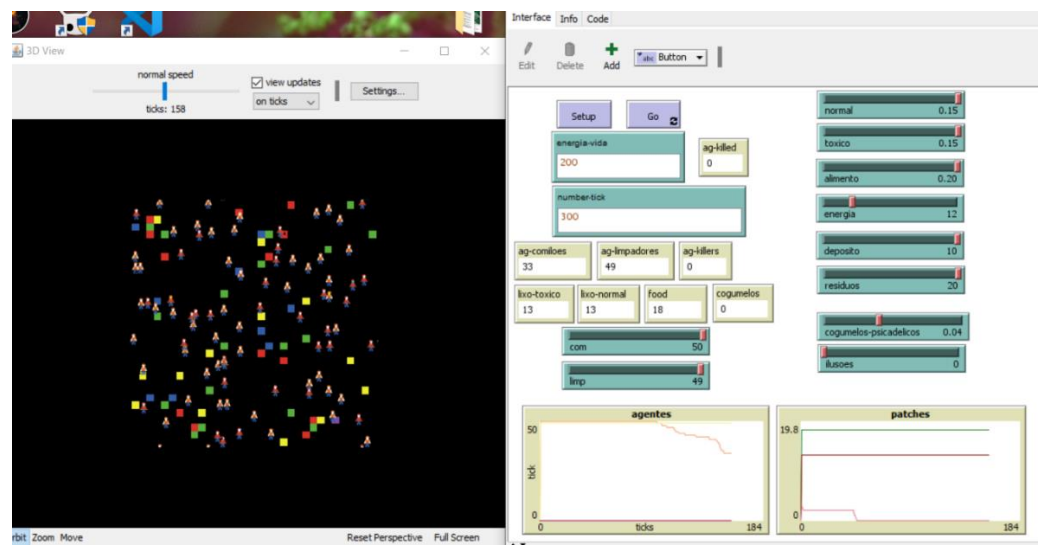
2.2) Comportamentos adicionais implementados

Como complementos adicionais o meu grupo optou por adicionar uma nova patch, esta rosa que pode ocupar de 0 a 10% das patches, chamada **cogumelos-psicadélicos**. Quando um dos agentes ingere uma patch rosa irá começar a mudar de cor aleatoriamente ao longo do modelo.

O segundo passo dos comportamentos adicionais foi adicionar um novo agente chamado **killer** (ilusões) com movimento completamente aleatório, quando este encontra um dos agentes que tenha consumido um dos patches rosa, irá matá-lo e ganhar 50 pontos de energia, este agente só se “alimenta” de agentes que estejam sobre o efeito das patches rosa.

2.3) Ambiente Gráfico

Foi desenvolvido um ambiente gráfico com monitores, plots, inputs e switches de modo a facilitar a mudança de variáveis assim como auxiliar a leitura de dados dos diversos testes aplicados ao modelo.



3. Análise de resultados

Description of the tests performed, results obtained and their interpretation.

Tabela 1: Síntese dos resultados.

Teste	Taxa de Extinção
Modelo comida	0%
Modelo base	0-5%
Modelo lixo	55-60%
Modelo max	0%
Modelo agente	100%
Modelo adicional	0-10%

No teste de modelo foram considerados 200 de energia e 300 ticks de duração, estes valores foram utilizados pois permitem uma análise clara de modelo para modelo.

O **modelo base** serve como base para comparar resultados com outros modelos de modo a conseguir distinguir as diferenças, pode ser visto como o modelo normal para quais os outros são situações mais distintas. A sobrevivência dos comilões está diretamente relacionada com a percentagem de comida presente no mapa, podemos observar também que quando comparado ao **modelo comida** que a existência de lixo irá ter um impacto muito negativo na

sobrevivência dos agentes comilões, enquanto que irá por outro lado aumentar muito a quantidade de agentes limpadores que sobrevivem.

No **modelo comida**, modelo este sem lixo e apenas comida para n agentes de cada, com energia ganha dos alimentos constante, foi concluído que sem adicionar lixo, os resultados de sobrevivência entre os agentes limpadores e comilões são muito similares, quanta mais comida mais agentes irão sobreviver ao fim de 300 ticks. Com apenas 5% das células de comida os agentes têm uma taxa de extinção de 100 % ao fim de 300 ticks, reduzindo conforme o aumento de percentagem de comida assim como número de agentes.

No **modelo lixo** foi avaliada a capacidade sobrevivência dos agentes num ambiente com o mínimo de comida possível, e foi concluído que apesar de a quantidade de agentes que sobrevivem ser extremamente baixa, vê-se claramente que quanto mais lixo o ambiente tiver, maior será a quantidade de agentes limpadores que sobrevive; relativamente a agentes comilões como é evidente quando a quantidade de lixo é muito maior que a de alimento irão ter extinção na maior parte dos testes.

O **modelo só agentes** têm configurado os valores mínimos de comida e lixo de modo a testar a sobrevivência dos agentes por si com o mínimo possível de elementos. Isto provou que ao final de 300 iterações não existe sobreviventes.

O **modelo máximo** teve como objetivo colocar as variáveis no máximo e comparar as taxas de sobrevivência entre 15 e 35 agentes para as mesmas percentagens de patches. De 15 sobraram em média 3.5 comilões e 9.7 limpadores, de 35 sobraram em média 8.1 comilões e 24.1 limpadores. Ou seja, valores praticamente proporcionais, isto deve-se á capacidade de as patches voltarem a aparecer, nunca existindo escassez de alimento.

O **modelo adicional** integra as agentes ilusões e patches rosa, este teste foi criado com a intenção de combinar a probabilidade de um agente pisar uma destas patches com a probabilidade de ser encontrado por um dos agentes ilusão. Pode ser concluído que quanto maior for a quantidade de cogumelos rosa e a quantidade de ilusões maior é a quantidade de agentes que é morto pelas ilusões.

4. Conclusão

O projeto foi realizado sem grandes problemas ou limitações, foram desenvolvidos vários agentes e patches com sucesso, com o propósito de testar o programa foram criadas várias situações de onde tiramos algumas conclusões relativas á sobrevivência dos agentes.

Enquanto que os agentes limpadores possuem a capacidade de ganhar vida não só da comida como também do transporte de lixo, os agentes comilões têm uma percentagem de sobrevivência muito menor em comparação, isto podia ser corrigido criando um método secundário para agentes comilões ganharem vida.

Quanto incluídos os agentes killer (ilusão) e cogumelos a quantidade de agentes a morrer irá aumentar consideravelmente, mesmo tendo em consideração que as ilusões só podem encontrar agentes aleatoriamente e os agentes só podem encontrar patches rosa d forma aleatória . Teria sido interessante criar um algoritmo de procura de agentes que tenham pisado patches rosa e comparar as taxas de mortalidade do algoritmo aleatório contra um estruturado.

A parte do trabalho levada em maior detalhe foram os testes do programa em vários modelos, mas para fins de precisão rendeu o esforço do grupo.

Referências

<http://ccl.northwestern.edu/netlogo/docs/>