Introdução à Inteligência Artificial

Trabalho Prático nº 1- Agentes Racionais

Docente:

Carlos Pereira

Alunos:

Paulo Henrique Figueira Pestana de Gouveia nº 2020121705 – LEI

João Filipe Silva de Almeida nº 2020144466 – LEI

Coimbra, 9 de Novembro de 2021

Índice

1. Introdução **3**

2. Implementação **4**

2.1 Modelo Base 4

2.2 Modelo Melhorado 6

3. Análise de Resultados Obtidos **7**

3.1 Resultados do Modelo Base 7

3.2 Resultados do Modelo Melhorado 9

4. Conclusão **12**

5. Referências **12**

1. Introdução

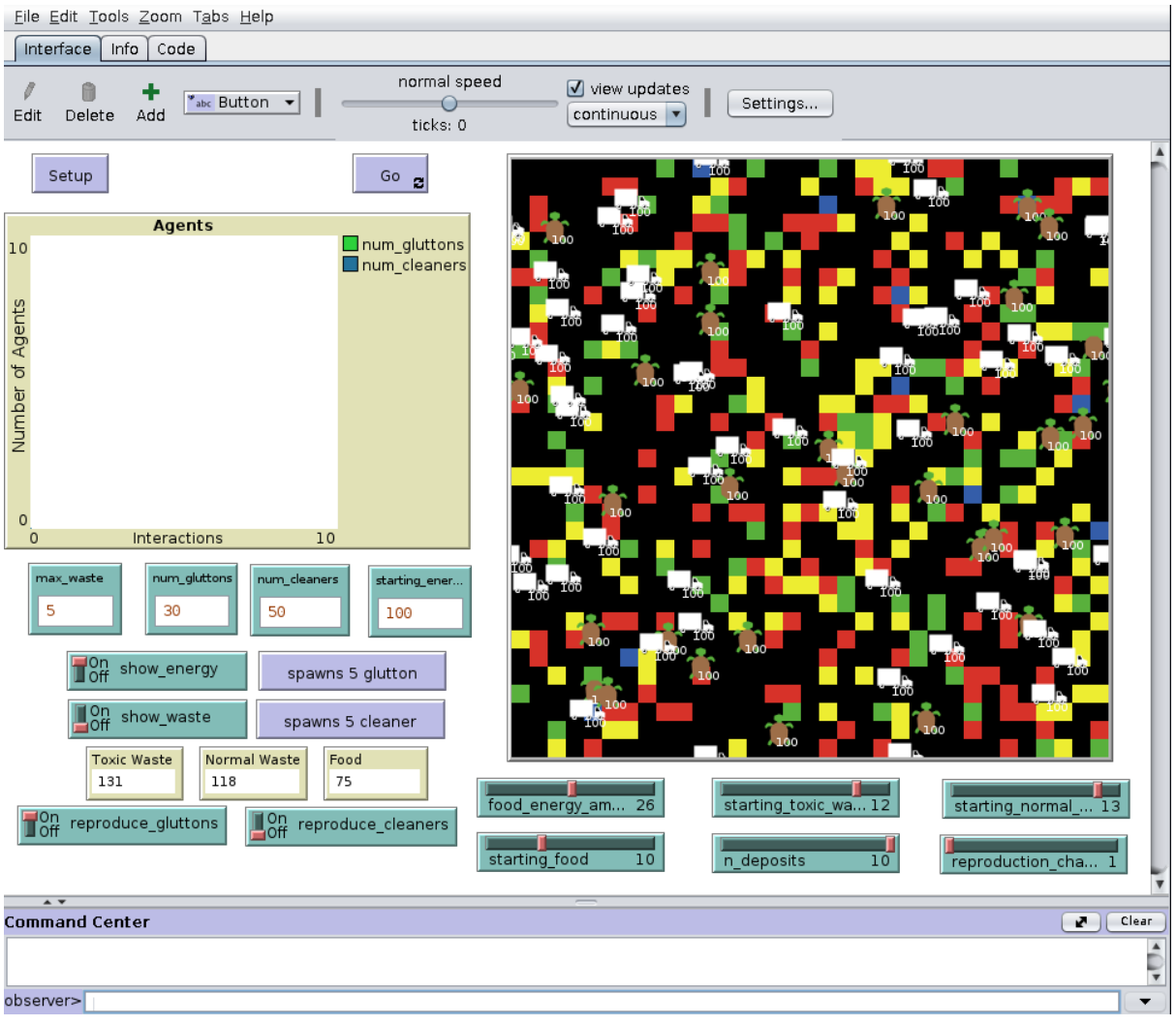
Este trabalho foi realizado no âmbito da unidade curricular de Introdução à Inteligência Artificial e consiste na criação dos “Agentes Racionais", feito em Netlogo.

Com este trabalho é possível perceber qual é a melhor ação possível que estes agentes racionais adotem diante uma situação, num certo tipo de ambiente. Foram criados dois agentes distintos, cada um com características próprias indicadas para o modelo base. Bem como quatro variáveis com qual vão interagir.

O programa termina quando todos os agentes morrem.

Foi feita uma análise dos resultados obtidos nos testes realizados neste programa.

Os objetivos foram cumpridos, pois conseguiram raciocinar, perceber, tomar decisões perante cada situação presente no ambiente.



2. Implementação

2.1 Modelo Base

O modelo base começa por indicar as configurações com restrições para o ambiente(patches) que a simulação toma lugar que podem ser configuradas pelo utilizador.

Este ambiente contém lugares com cores diferentes que serão as quatro diferentes variáveis que os agentes irão interagir.

Um patch de cor verde, significa que a mesma contém alimento, este alimento será utilizado pelas turtles para ganharem energia.

Um patch de cor azul, significa que se encontra lá um depósito, onde os limpadores podem despejar o lixo(caso se não tiver cheio) que carregam, sendo que ganham energia ao completar a tarefa .

Caso um patch tenha a cor amarela, significa que se a mesma contém lixo normal, este lixo retira uma porcentagem fixa(5%) de energia a um comilão que o detete e deverá ser recolhido pelos limpadores.

Por fim, caso um patch tenha cor vermelha, conclui-se que naquele lugar está lixo tóxico, este lixo retira uma maior porcentagem de energia(10%) que o lixo normal a um comilão e deverá também ser recolhido pelos limpadores.

Os patches de cor preta indicam que estão vazias, não tendo qualquer efeito sobre os agentes.

Os dois agentes criados no modelo base denominam-se de comilões e limpadores:

Comilões: têm o objetivo de sobreviver no ambiente, para isso, procuram os patches que contêm alimento(patches verdes), priorizando-as e evitando os patches que contêm qualquer tipo de lixo;

Conseguem percecionar o conteúdo das células à sua frente, direita e esquerda. Podem deslocar-se para a célula imediatamente à sua frente, rodar 90º para a esquerda ou rodar 90º para a direita. Em cada iteração, podem executar apenas uma destas ações. Cada ação retira uma unidade de energia;

Ganham um de energia ao consumir alimento, sendo a única maneira que este agente tem de sobreviver;

Estão representados a cor castanha e “shape” “turtle”;

Limpadores: Sobrevivem como objetivo secundário, sendo o seu objetivo primário retirar lixo do ambiente, caso considere necessário, apanha patches com alimento. Cada limpador pode carregar uma quantidade limitada de lixo, sendo que se comportam de maneira diferente conforme consigam ou não carregar mais lixo;

Eles têm associada uma variável adicional "waste", que quantifica a quantidade de lixo que cada agente carrega. Ao limparem uma patch the lixo tóxico esta variável será incrementada em 2, enquanto é incrementada em apenas 1 caso limpam lixo normal. E serão impedidos de limpar mais células de lixo caso a sua variável waste atinja o seu limite máximo configurável.

Se os limpadores estiverem cheios não conseguem carregar mais lixo, as prioridades passam a ser depositá-lo;

Se estiverem apenas a uma unidade de atingirem o seu máximo de carga, irão priorizar apanhar células de lixo normal, uma vez que as de lixo tóxico não conseguem mais recolher pois ultrapassará o seu máximo de carga.

Conseguem percecionar o conteúdo das células que se encontram imediatamente à sua frente e à direita. Os Limpadores podem deslocar-se para a célula imediatamente à sua frente, rodar 90º para a esquerda ou rodar 90º para a direita. Em cada iteração, podem executar apenas uma destas ações. Cada ação retira uma unidade de energia;

Estão representados a cor branca e "shape" "truck";

O utilizador pode inicialmente definir os seguintes aspetos da simulação:

-Percentagem de lixo tóxico (entre 0% e 15%);

-Percentagem de lixo normal (entre 0% e 15%);

-Percentagem de alimento (entre 5% e 20%);

-Número de depósitos (entre 1 e 10);

-Número de limpadores (entre 0 e 10);

-Número de comilões (entre 0 e 10);

-Valor da energia inicial (entre 20 e 50);

-Ganho de energia por encontrar alimento (entre 10 e 33);

-Quantidade máxima de lixo que um limpador consegue carregar (entre 1 e 10);

2.2 Modelo Melhorado

Face ao que nos era pedido no enunciado, vimos benéfico para o trabalho, e didático, adicionar algumas funcionalidades extra, de forma a melhorar a inteligência artificial dos agentes bem como as suas habilidades, ou a forma como interagem com o ambiente.

Algumas destas adições passaram por pequenas alterações no comportamento dos agentes, como por exemplo:

O programa continua a terminar quando todos os agentes morrem, embora seja possível pela interface, mesmo que todos os agentes tenham já morrido, spawnar mais turtles e continuar a simulação.

Devido ao alto índice de sobrevivência dos agentes limpadores adicionou-se uma regra que faz os limpadores perderem mais energia quanto mais lixo carregam.

Para contrabalançar esta desvantagem, os limpadores agora descarregam o lixo que têm acumulado se virem um depósito mesmo que não estejam cheios ainda.

Notou-se que depois de bastantes iterações os limpadores acabariam por morrer por ficar presos num ciclo infinito num poço com lixo à volta, foi por isso adicionada alguma aleatoriedade, onde existe uma pequena probabilidade de o agente executar uma ação diferente da mais adequada para impedir este evento.

Apercebemo-nos também que os comilões não tinham maneira de recuperar de certas situações de má sorte, e iam por isso apenas morrendo aos poucos. Achámos por bem portanto atribuir-lhes a propriedade de reprodução. Desta forma, quando a sua energia é maior que a energia inicial existe uma probabilidade da sua energia ser dividida em metade e ser criado um novo glutton à frente.

Desta forma podemos ter uma simulação mais dinâmica onde é possível ver os efeitos das definições que mudamos em tempo real na quantidade de comilões que existem através do gráfico dado.

E para enfatizar este dinamismo, adicionamos algumas opções à interface que nos permitem alterar a simulação em tempo real. Como a habilidade de adicionar cinco de quaisquer agentes instantaneamente, ou impedir ou não que se reproduzam, tanto os limpadores como os comilões.

3. Análise de Resultados Obtidos

3.1 Resultados do Modelo Base

Condições dos testes

De maneira a tornar os testes, tanto possíveis de realizar, como fiáveis, cada teste foi realizado no mínimo 5 vezes;

De maneira a evitar que alguns testes tenham fim, os testes serão parados nas condições em que:

Não há nenhuma turtle viva no ambiente;

Ao longo dos testes realizados vai querer ver-se a diferença que cada mudança cria para o ambiente, pelo que vão ser definidos alguns valores por defeito:

-Percentagem de lixo tóxico: 13;
-Percentagem de lixo normal: 13;
-Percentagem de alimento: 10;
-Número de depósitos: 10;
-Número de limpadores: 30;
-Número de comilões: 30;
-Valor da energia inicial: 100
-Ganho de energia por encontrar alimento: 40;
-Quantidade máxima de lixo que um limpador consegue carregar: 5;

Como já foi brevemente explicado no modelo melhorado e nas razões que nos levaram a alterar o modelo, os resultados do modelo base eram satisfatórios, mas não apresentavam dinâmica suficiente para ser fácil a experimentação de ambientes diferentes e a extração de novos dados para análise e dedução de hipóteses novas.

Estes problemas fundamentam-se na pouca versatilidade do modelo base, existia um número fixo de agentes iniciais, e nunca poderiam existir mais.

Devido a estas características pudemos fazer algumas suposições, tais como:

- Devido ao número fixo de agentes, a única coisa que poderia acontecer é eles morrerem, pelo que, depois de várias iterações, poderiam apenas acabar por se extinguir.

teste n	vida média (comilões)	vida média(limpadores)	total de ticks	limpadores vivos	comilões vivos
teste 1	1.51	30	10000	30	0
teste 2	1.40	30	10000	30	0
teste 3	1.1	30	10000	30	0
teste 4	0.71	30	10000	30	0
teste 5	1.07	30	10000	30	0

Esta suposição acabou por se provar falsa. Isto porque, uma vez que os agentes comilões acabavam quase sempre por morrer, os limpadores eram capazes de ganhar a sua energia através do lixo também, continuavam por isso a ganhar energia, chegando a níveis absurdos de energia.

- A variável de quantidade máxima de lixo que um cleaner podia levar, não iria afetar grandemente o seu índice de sobrevivência

teste n	lixo máximo	vida média (comilões)	vida média (limpadores)	total de ticks	limpadores vivos	comilões vivos
teste 1	100	0.69	30	10000	30	0
teste 2	50	0.78	30	10000	30	0
teste 3	25	0.76	30	10000	30	0
teste 4	10	0.83	30	10000	30	0
teste 5	5	1.75	30	10000	30	0

Esta suposição provou-se também falsa. Uma vez que os limpadores apenas procuravam se dirigir a depósitos quando a sua carga estivesse maximizada.

- Quanto mais comida existir, mais fácil será para os agentes subsistir.

teste n	comida inicial	vida média (comilões)	vida média (limpadores)	total de ticks	limpadores vivos	comilões vivos
teste 1	20	7.36	30	10000	30	0
teste 2	15	4.23	30	10000	30	0
teste 3	10	0.98	30	10000	30	0
teste 4	5	0.29	30	10000	30	0

Mostrou-se que para os comilões aumenta ligeiramente a sua vida média, mas mesmo assim não sendo o suficiente para garantir a sua sobrevivência, enquanto para os limpadores a mudança é indiferente.

3.2 Resultados do Modelo Melhorado

Deste modo, o processo de experimentação viu-se bastante mais interativo e entendido.

Aplicando os mesmos testes:

Condições dos testes

-Percentagem de lixo tóxico: 12;
-Percentagem de lixo normal: 13;
-Percentagem de alimento: 10;
-Número de depósitos: 10;
-Número de limpadores: 50;
-Número de comilões: 30;
-Valor da energia inicial: 100;
-Ganho de energia por encontrar alimento: 26;
-Quantidade máxima de lixo que um limpador consegue carregar: 5;

-Reprodução dos limpadores off;
-Reprodução dos comilões on;

As alterações feitas ao modelo melhorado foram feitas de modo a que o processo de experimentação fosse mais fácil e esclarecedor.

+

- A variável de quantidade máxima de lixo que um cleaner podia levar, não iria novamente afetar grandemente o seu índice de sobrevivência, uma vez que os agentes cleaners se dirigem a um depósito assim que encontram um.

teste n	lixo máximo	vida média (comilões)	vida média (limpadores)	total de ticks	limpadores vivos	comilões vivos
teste 1	100	24.89	3.65	10000	0	1
teste 2	50	30.76	1.07	10000	0	1
teste 3	25	12.07	5.79	10000	1	0
teste 4	10	19.69	4.38	10000	0	1
teste 5	5	154.88	9.23	10000	6	281

Esta suposição provou-se falsa mais uma vez. Pensamos que este resultado se deve ao facto dos cleaners priorizarem a apanha de lixo acima da apanha de alimento, portanto, com um depósito muito grande continuariam a apanhar lixo em vez de comida mesmo que a sua energia se encontrasse já bastante baixa. Achamos que isto terá também a haver com a regra adicional de quanto mais lixo carregam, mais energia perdem, que implementámos. Isto prova uma falha no comportamento do agente, poderia ser arranjada através de um teste adicional se a energia do agente estivesse abaixo de um ponto crítico, começaria a priorizar comida e depósitos em vez de lixo.

- Quanto mais comida existir, mais fácil será para os agentes subsistir.

teste n	comida inicial	vida média (comilões)	vida média (limpadores)	total de ticks	limpadores vivos	comilões vivos
teste 1	20	9206.95	49.2	1000	48	82482
teste 2	15	1244.49	38.66	1000	35	7426
teste 3	10	53.25	22.02	1000	13	71
teste 4	5	12.88	15.50	1000	0	1

Continuou a mostrar-se verdadeiro e universal para todos os agentes, uma vez que todos eles procuram e consomem ativamente comida ou patches verdes.

- Os comilões são afetados positivamente pelas limpezas dos cleaners.

teste n	número inicial de cleaners	vida média (comilões)	vida média (limpadores)	total de ticks	limpadores vivos	comilões vivos
teste 1	0	14.42	0	10000	0	1
teste 2	0	15.11	0	10000	0	1
teste 3	0	14.31	0	10000	0	1
teste 4	0	18.12	0	10000	0	1
teste 5	0	11.85	0	10000	0	1
teste 6	50	115.12	12.92	10000	10	124
teste 7	50	181.30	8.58	10000	3	131
teste 8	50	206.82	12.39	10000	6	256
teste 9	50	75.97	8.90	10000	6	84
teste 10	50	229.34	14.61	10000	13	269

Suposição verdadeira, quando na simulação ativamos a reprodução dos cleaners e eles aumentavam bastante em números, víamos também um aumento na quantidade de comilões, no entanto, este aumento era temporário e rapidamente voltava a estabilizar. Isto porque com mais cleaners na simulação, mais comida vão consumir, limitando a reprodução dos gluttons até certo ponto.

4. Conclusão

No final das análises dos comportamentos dos agentes, pode-se inferir que muitas vezes o comportamento esperado não se alinha com o comportamento observado, e que é benéfico analisar os dados em concreto para se poder tirar conclusões acertadas relativamente aos vários ambientes com diferentes parâmetros.

O trabalho foi realizado, maioritariamente em NetLogo, utilizando-se ferramentas como Visual Studio Code, Git e Github desktop, Google Docs, Discord, para colaboração.

As maiores dificuldades que enfrentamos foi maioritariamente a obtenção e interpretação dos dados para o relatório, devido à natureza repetitiva do processo de obtenção de dados.

Outra dificuldade foi a organização do código, devido às múltiplas condições criadas de forma a englobar todas as situações que os agentes poderiam encontrar, o código tornou-se bastante pesado a termos de *if*s causando secções com grandes aglomerados de chavetas. Ultrapassamos esta dificuldade graças à funcionalidade da auto-indentação do NetLogo.

5. Referências

[1] Net Logo Dicionário, https://ccl.northwestern.edu/netlogo/docs/dictionary.html

[2] Net Logo Documentação, https://ccl.northwestern.edu/netlogo/docs/

[3] Net Logo Behaviour Space, https://ccl.northwestern.edu/netlogo/docs/behaviorspace.html/