

Programação Orientada a Objectos 2017/2018

Trabalho prático

Este enunciado é constituído por duas partes:

Parte A – Regras geraispágina 1

Parte B – Descrição do temapágina 2

Parte A - Regras gerais do trabalho

O trabalho é constituído por um programa em C++, que deve implementar o tema proposto usando correctamente os princípios de orientação a objectos e os mecanismos e classes biblioteca dados nas aulas. A interface com o utilizador é feita segundo o conceito de consola não-gráfica, e caso se justifique, será disponibilizada uma biblioteca para auxiliar nesta tarefa. O tema proposto deixa diversos aspectos em aberto. Devem ser resolvidos de acordo com a lógica e bom senso, e segundo os bons princípios de estruturação orientada a objectos.

O trabalho pode ser feito individualmente ou em grupos de 2, e neste último caso os alunos podem ser de laboratórios diferentes.

O trabalho é entregue em duas metas.

Meta 1 – Prazo: 17 de Dezembro de 2017

Objectivos e funcionalidade a entregar:

- Conceitos de mundo com ninhos/comunidades; formigas que têm apenas movimentos aleatórios; ainda não se consideram migalhas
- Configuração do mundo e início da simulação
 - Leitura de comandos, interpretação e processamento
 - Comandos de configuração, incluindo o comando *executa* e *inicio*
- Simulação
 - Comandos:
 - *ninho* – comando para criar um ninho
 - *criaf* – comando para criar formigas, considerando apenas a formiga exploradora com movimentos aleatórios (ainda não come porque não são incluídas migalhas)
 - *tempo* – comando para avançar uma iteração
 - *listamundo* – listar elementos do mundo
 - *listaninho* – listar toda a informação relacionada com o ninho
 - *listaposicao* - listar toda a informação relacionada com a posição
 - Visualização da grelha
 - Visualização do movimento das formigas
- Relatório – serão dadas indicações acerca do conteúdo do relatório.

Defesa: obrigatória.

Meta 2 – Prazo: 10 de Janeiro de 2018

Objectivos e funcionalidade a entregar:

- Programa com todas as funcionalidades.
- Relatório – serão dadas indicações acerca do conteúdo do relatório.

Defesa: obrigatória.

Em ambas as metas a entrega será feita via *moodle*, e as indicações serão dadas na devida altura.

O trabalho é sujeito a defesa sem a qual não haverá nota. A defesa afecta a nota final e alunos do mesmo grupo podem vir a ter notas diferentes.

As questões omissas que sejam relevantes serão resolvidas e divulgadas pelos docentes no moodle.

Parte B - Descrição do tema

1 INTRODUÇÃO E IDEIA GERAL

Pretende-se um programa em C++ para simulação de populações de formigas. O espaço onde as formigas vivem (o seu mundo) é plano e pode ser entendido como uma espécie de grelha bidimensional fechada (“fechada” = quando as formigas atingem um limite, não transitam para o extremo oposto). Os vários elementos do programa (formigas e outras coisas a descrever mais adiante) existirão nessa grelha em posições com valores inteiros (não se está metade numa posição e metade na posição ao lado).

É importante ter em atenção que neste momento estão a descrever-se os conceitos e não a implementação. O facto de se mencionar aqui “grelha” não significa que uma grelha (matriz) seja ou deixe de ser a forma adequada para a representação destas ideias.

Existirão outras coisas no ambiente para além das formigas, por exemplo, migalhas que servem de alimento. Para simplicidade de escrita e entendimento, neste enunciado vai-se chamar a tudo o que pode existir no ambiente de “elementos”. Ao ambiente vai chamar-se simplesmente “mundo”.

Os elementos que existem no mundo têm um valor de energia. Os elementos só existem enquanto têm energia acima de um determinado limiar. Quando a energia de um destes elementos atinge um valor muito baixo, o mundo remove-o (a formiga morreu, a migalha secou, etc.). Pode-se imaginar que a energia é uma espécie de “quantidade de vida”.

Ao nascerem, as formigas pertencem a uma comunidade, possuem um conjunto de propriedades e adoptam um comportamento específico. Existem vários comportamentos possíveis, definidos mais à frente. O comportamento específico de cada formiga determina como interage com o mundo e com os restantes elementos nele existentes (por exemplo, como se desloca e alimenta, entre outras coisas).

A simulação decorre em sucessivas unidades de tempo discretas, às quais se vai chamar iterações. Em cada iteração, todos os elementos que estão colocados no mundo executam uma determinada acção de acordo com a sua natureza e as propriedades que possuem: as formigas podem deslocar-se ou alimentar-se, as comunidades podem gerar formigas, os alimentos degradam-se, etc..

O programa a elaborar irá permitir criar mundos com formigas e outros elementos, e depois ver o que acontece durante a simulação. O utilizador poderá definir o estado inicial, simular o avanço das iterações e criar ou modificar elementos e propriedades da simulação. Será sempre apresentado no ecrã o conteúdo do mundo e estado da simulação. A interface com o utilizador é em “modo texto” e o utilizador dá indicações ao programa através da lógica de comandos escritos (os quais se descrevem mais adiante).

2 O MUNDO

O mundo das formigas é uma espécie de grelha bidimensional quadrada e plana. A dimensão é descrita em número de linhas (igual ao número de colunas) e só é conhecida em *runtime*. A dimensão mínima é 10x10, mas o limite superior da dimensão é apenas aquele que for imposto pela memória do computador.

Eventualmente poderá não ser possível visualizar toda a grelha no ecrã. Neste caso será apresentada uma zona, cujo centro é o *foco de atenção*, descrito por linha e coluna. O utilizador poderá definir e mudar o foco de atenção sempre que o desejar.

3 ELEMENTOS NO/DO MUNDO

Para além das formigas, o mundo contém outros elementos: ninhos, comunidades, migalhas.

Ninhos e comunidades

As formigas nascem e existem no contexto de comunidades. Cada comunidade possui o seu próprio ninho. Durante a simulação pode haver várias comunidades a competir pelos recursos no mundo. As formigas sabem a que ninho pertencem e consideram como inimigas todas as formigas que não pertencem à sua comunidade.

O ponto fulcral de cada comunidade é o seu ninho que se encontra numa determinada posição. Cada ninho é criado com uma quantidade inicial de energia (valor configurável). Os ninhos são identificados por um número que é automaticamente atribuído pelo programa, sendo o primeiro o ninho 1, e cada novo ninho tem o número seguinte ao do ninho anterior.

Um ninho serve essencialmente para criar novas formigas e também como refúgio de algumas das formigas existentes na sua comunidade. Algumas formigas possuem características que lhes permitem voltar ao seu ninho para depositar energia em excesso ou obter energia em falta.

Uma formiga que esteja no ninho e que tenha um valor energético inferior a metade daquele com que foi criada, irá obter a energia que lhe falta (até chegar a 50% do seu valor inicial) a partir da energia acumulada no ninho (ou seja, recarrega parcialmente a sua bateria). A recarga é feita da seguinte forma: por cada iteração em que a formiga esteja no ninho recebe deste N unidades de energia. O valor de N é configurável e é o mesmo para todas as formigas e ninhos.

Uma formiga que esteja no ninho e que tenha mais energia do que aquela com que foi criada transfere a energia em excesso para o ninho. A transferência é feita N unidades por iteração em que a formiga esteja no ninho (o mesmo N do caso anterior).

As formigas saem do ninho nas seguintes situações:

- Quando têm entre 50% e 100% da sua energia inicial (ou seja já não têm energia para dar e também já não conseguem carregar energia a partir do ninho).
- Quando têm menos de 50% da sua energia inicial mas o ninho já só tem 1 unidade de energia (se tirar mais o ninho morre, por isso não retiram e vão-se embora – pode ser que encontrem umas migalhas para comer).

O ninho também fabrica novas formigas. Sempre que a energia armazenada no ninho atinja C% a mais do que a sua energia inicial e não existam formigas a retirar energia nesse instante, então uma nova formiga é criada, na posição do ninho, sendo a sua energia inicial retirada da energia do ninho. O valor de C é definido no início da simulação. As propriedades da formiga são definidas de forma aleatória (ver mais adiante o que são as propriedades das formigas).

As formigas que são criadas através de comandos não retiram energia ao seu ninho. Só durante a simulação, é que as novas formigas que forem sendo criadas pelos ninhos seguem a regra de transferência de energia explicada no parágrafo anterior.

Migalhas

Uma migalha é um alimento que tem energia (valor inteiro) e ocupa uma posição no mundo. Em cada posição pode existir, no máximo, uma migalha. No início da simulação, uma determinada percentagem das posições do mundo são preenchidas com migalhas (a percentagem é configurável). Em cada iteração, surgem entre 0 a N novas migalhas no mundo, sendo N um valor configurado e o valor concreto em cada iteração obtido de forma aleatória (novo valor em cada nova iteração).

As migalhas são uma das fontes de alimento das formigas (existem outras formas de obter energia, descritas mais à frente). As migalhas são criadas com um valor inicial de energia, configurável. As formigas podem comer migalhas, de acordo com um conjunto de regras descritas mais à frente. Quando isto acontece, a formiga retira uma parte da energia da migalha, transferindo-a para si. O valor concreto a retirar depende das características da formiga. Em cada iteração, as migalhas perdem 1 unidade de energia. Se uma migalha tiver menos de 10% da sua energia inicial, considera-se que está completamente degradada e desaparece.

4 AS FORMIGAS

As formigas pertencem a uma comunidade e são criadas dentro do respectivo ninho. As formigas são identificadas por um número gerado automaticamente: cada nova formiga tem o número seguinte à da formiga anterior. A primeira formiga tem o número de identificação 1. O ninho pode conter várias formigas na sua posição. Nas posições do mundo fora dos ninhos pode existir, no máximo, uma formiga. Uma formiga pode estar na mesma posição em que está uma migalha. Cada formiga tem um determinado conjunto de características iniciais e um comportamento, ambos definidos no momento da sua criação.

Energia das formigas

As formigas são criadas com um nível de energia inicial (o valor depende das suas características). Em cada iteração, o nível de energia de cada uma das formigas existentes varia de acordo com os seguintes princípios:

- Se não estiver no ninho, consome (isto é, perde) sempre 1 unidade de energia. Pode consumir mais, dependendo das acções que executar;
- Se estiver no ninho, transfere para ele ou obtém dele algumas unidades de energia (de acordo com o que já foi dito na secção acerca do ninho)
- Pode obter energia das migalhas ou de outras formigas;
- Se o nível de energia for negativo ou nulo, a formiga morre e desaparece.

Visão das formigas

As formigas têm um raio de visão que define a zona do mundo à sua volta que conseguem ver. Pode considerar-se que, estando na posição (x, y), podem pesquisar o conteúdo de células em posições (x1, y1), tal que:

$$\max(\text{abs}(x - x1), \text{abs}(y - y1)) \leq \text{raio de visão}$$

O comportamento das formigas pode depender do conteúdo das posições na sua vizinhança. Assim, o raio de visão define a distância a que as formigas conseguem reagir a algo que se passa perto delas.

Movimento das formigas

As formigas têm um raio de movimento que define a zona do mundo para onde se podem mover. Para qualquer formiga, o raio de movimento é sempre menor ou igual do que o raio de visão. Pode considerar-se que, estando na posição (x, y), podem deslocar-se na iteração actual para uma célula numa posição (x1, y1), tal que:

$$\max(\text{abs}(x - x1), \text{abs}(y - y1)) \leq \text{raio de movimento}$$

Em cada iteração, o movimento efectivo corresponde ao número de células percorrido pela formiga (número de posições percorridas na horizontal mais o número de posições percorridas na vertical). Assumindo que a formiga se deslocou do ponto (x, y) para o ponto (x1, y1), o movimento efectivo é obtido da seguinte forma:

$$\text{Movimento Efectivo} = \text{abs}(x - x1) + \text{abs}(y - y1)$$

Comportamento das formigas

Existe apenas um tipo base de formiga, mas a sua forma de actuar varia bastante consoante um conjunto de regras de comportamento que possuem. Quando as formigas estão no ninho comportam-se como foi referido na secção acerca do ninho. As regras de comportamento só se aplicam às formigas, quando estão fora do ninho. Cada regra determina uma acção que a formiga executa se uma determinada condição se verificar. Existem várias regras, cada uma delas com uma actuação diferente. Formigas diferentes terão conjuntos diferentes de regras e portanto, comportamentos diferentes. Cada regra é composta por duas partes:

- **Condição** - Determina se a regra é ou não aplicada.
- **Acção** – Especifica o que deve ser feito, caso a regra seja aplicada (caso a condição seja verdadeira).

Exemplo:

A *RegraFoge* exemplifica a condição e acção para uma formiga fugir de outra que a persegue:

RegraFoge:

Condição – Está uma formiga de outra comunidade no seu raio de visão.

Acção – Deslocar no sentido oposto ao da formiga inimiga.

A *RegraSMS* exemplifica a condição e acção que permitem a uma formiga enviar um SMS.

RegraSMS:

Condição – Nenhuma ... (isto é, a condição é sempre verdadeira...)

Acção – Enviar um SMS

Note-se que no exemplo anterior, a acção é sempre executada, dado que a condição é sempre verdadeira (ou seja, não existe).

Processamento das Regras:

Cada formiga pode ter várias regras, ordenadas por nível decrescente de prioridade. Em cada iteração, a formiga percorre a sua lista ordenada de regras e aplica a primeira que conseguir (ou seja, a primeira que tiver a sua condição verdadeira). Cada formiga só aplica uma regra em cada iteração. Se nenhuma das suas regras for verdadeira nessa iteração, a formiga não executa nenhuma acção.

Formigas diferentes podem atribuir importância diferente às mesmas regras, dependendo do ordenamento parcial que adoptarem. Por exemplo:

- A formiga A tem as 2 regras exemplificadas anteriormente, estando a *RegraFoge* na primeira posição e a *RegraSMS* na segunda. Se nessa iteração estiver uma formiga inimiga no seu raio de visão, a formiga A executa a acção de se deslocar no sentido oposto. Caso não esteja nenhuma formiga inimiga ao alcance da vista, então a formiga envia um SMS.
- A formiga B tem as mesmas duas regras, mas com prioridade trocada. Neste caso a formiga prefere enviar SMS a fugir de inimigos (e uma vez que a condição de *RegraSMS* é sempre verdadeira, a formiga só envia SMS e não faz mais nada.

Regras a implementar

No trabalho devem ser implementadas as seguintes regras

- **RegraFoge** – Caso exista uma formiga de outra comunidade no seu raio de visão, desloca-se no sentido oposto.
- **RegraPersegue** – Caso existam formigas de outras comunidades no seu raio de visão, movimenta-se na direcção daquela que tiver mais energia.
- **RegraAssalta** – Caso existam formigas de outras comunidades em posições abrangidas pelo raio de movimento, assalta a que tiver mais energia. O assalto corresponde a transferir metade da energia da formiga assaltada para si.
- **RegraProtege** – Caso exista uma formiga de outra comunidade e uma formiga da mesma comunidade no seu raio de visão, move-se para uma posição intermédia entre estas duas para dificultar o caminho à inimiga.

- **RegraProcuraMigalha** – Caso existam migalhas no seu raio de visão, movimenta-se na direcção da que tiver mais energia.
- **RegraComeMigalha** - Caso existam migalhas na mesma posição ou adjacente, escolhe a que tiver mais energia e retira-lhe uma parte (a percentagem de energia retirada varia conforme o tipo de formiga).
- **RegraVaiParaNinho** - Caso a formiga tenha um valor de energia superior à sua energia inicial ou inferior a metade da sua energia inicial, tenha o ninho no seu raio de visão e esteja fora do ninho há mais de 10 iterações, desloca-se na sua direcção. Se o ninho já estiver no seu raio de movimento, entra no ninho.
- **RegraPasseia** – Movimenta-se numa direcção aleatória dentro do seu raio de movimento, de forma incondicional.

Formigas especializadas

Existem vários tipos especializados de formigas. Tal como na natureza existem formigas guerreiras, operárias, etc., também neste simulador existem alguns tipos de configurações predefinidos. Estas especializações vão reflectir-se a dois níveis:

- Formigas com características diferentes têm regras de actuação diferentes e/ou prioridades distintas para estas regras.
- Valores operacionais de algumas características, tais como raio de visão, energia inicial, etc.

As especializações já existentes são:

Formiga Cuidadora

- Raio de visão: 5 unidades
- Raio de Movimento: 3 unidades
- Energia Inicial: 100
- Variação de energia:
 - Obtém energia de migalhas
 - Se estiver fora do ninho, em cada iteração perde 1+movimento efectivo unidades de energia
 - Se voltar ao ninho, em cada iteração, pode depositar ou recarregar energia

Comportamento – Descrição

- Perante o perigo de ser assaltada, foge.
- Se estiver ao pé de uma migalha, retira-lhe metade da energia.
- Procura activamente migalhas para ingerir e obter energia.
- Quando a formiga cuidadora tiver um valor de energia maior do que a energia inicial ou menor do que metade da energia inicial, tiver o ninho no seu raio de visão e estiver fora do ninho há mais de 10 iterações, dirige-se para o ninho.

Comportamento – Na prática traduz-se simplesmente nas regras:

- RegraFoge, RegraComeMigalha, RegraProcuraMigalha, RegraVaiParaNinho, RegraPasseia.

Formiga Vigilante

- Raio de visão: 7 unidades
- Raio de Movimento: 5 unidades
- Energia Inicial: 150
- Variação de energia:
 - Obtém energia de migalhas
 - Em cada iteração perde 1+movimento efectivo unidades de energia

Comportamento – Descrição

- Perante o perigo de uma formiga da sua colónia ser assaltada, tenta dificultar a deslocação do inimigo. Se, no seu raio de visão, detectar uma formiga inimiga e uma formiga da sua população, move-se em direcção a uma posição entre a formiga da sua família e a inimiga.
- Se estiver ao pé de uma migalha, retira-lhe 75% da sua energia.
- Procura activamente migalhas para ingerir e obter energia.

Comportamento – Na prática traduz-se simplesmente nas regras:

- RegraProtege, RegraComeMigalha, RegraProcuraMigalha, RegraPasseia.

Formiga Assaltante

- Raio de visão: 8 unidades
- Raio de Movimento: 4 unidades
- Energia Inicial: 80
- Variação de energia:
 - Obtém energia de migalhas e de outras formigas
 - Em cada iteração perde 1+2×movimento efectivo unidades de energia

Comportamento – Descrição

- Havendo uma formiga de outra comunidade numa posição adjacente, assalta-a.
- Se existirem formigas de outras comunidades no seu raio de visão, dirige-se à que tiver mais energia.
- Se estiver ao pé de uma migalha, retira-lhe 25% da sua energia.
- Procura activamente migalhas para ingerir e obter energia.

Comportamento – Na prática traduz-se simplesmente nas regras:

- RegraAssalta, RegraPersegue, RegraComeMigalha, RegraProcuraMigalha, RegraPasseia.

Formiga Exploradora

- Raio de visão: 10 unidades
- Raio de Movimento: 8 unidades
- Energia Inicial: 200
- Variação de energia:
 - Obtém energia de migalhas
 - Em cada iteração perde 1+movimento efectivo unidades de energia

Comportamento – Descrição

- Esta formiga passeia alegremente pelo mundo. Só quer explorar. Não se preocupa com inimigos, nem ninhos nem nada. Além de passear só tem a actividade de comer e isso apenas por que a natureza a isso a obriga: não procura comer, mas come migalhas (retirando-lhes toda a sua energia) se por acaso tropeçar nelas.

Comportamento – Na prática traduz-se simplesmente nas regras:

- RegraComeMigalha, RegraPasseia.

Formiga Surpresa

Esta formiga é surpresa na medida em que é totalmente definida pelos alunos e fará aquilo que estes entenderem. Se grupos diferentes definirem formigas surpresas iguais, então a surpresa será verdadeiramente surpreendente, inclusive, na defesa.

5 COMANDOS DO UTILIZADOR

O utilizador controla manualmente alguns aspectos da simulação. Pode intervir no mundo, criando comunidades, formigas e migalhas, e dando ordens para gerir o evoluir da simulação. Não dá ordens directas às formigas. Estas têm livre arbítrio (pensam elas) e fazem aquilo que as suas regras determinam.

Numa primeira fase, a fase de configuração, é necessário definir alguns parâmetros para que a simulação possa decorrer.

Fase de configuração

Os comandos necessários na fase de configuração são:

- **defmundo <limite>** - Define a dimensão do mundo como uma grelha bidimensional de *limite x limite* posições, sendo obrigatoriamente *limite* ≥ 10 .
- **defen <energia>** - Define o valor inicial da energia dos ninhos.
- **defpc <valor>** - Define o valor (percentagem) da energia inicial a partir do qual o ninho faz uma formiga. Exemplo: 50 -> o ninho faz uma formiga nova a se tiver 50% de energia a mais do que o inicial.
- **defvt <valor>** - Define o número de unidades de energia transferidas entre ninho e formiga por iteração. O valor inicialmente estabelecido é 1 e pode ser modificado com este comando.
- **defmi <valor de 0 a 100>** - Define a percentagem inicial posições vazias que têm (inicialmente) migalhas
- **defme <energia>** - Define a energia inicial das novas migalhas.
- **defnm <max>** - Define o número máximo de migalhas a serem criadas a cada instante. O número de migalhas novas é sorteado entre 0 e max.

- **executa <nomeFicheiro>** - Lê comandos do ficheiro de texto designado por *nomeFicheiro*, um por linha, e executa-os. Este comando pode ser utilizado para configurar o estado inicial da simulação, mas também para executar um conjunto gravado de comandos quando a simulação já está a decorrer. Cada linha do ficheiro tem um comando com a mesma estrutura dos comandos que se digitam através do teclado, e portanto sujeita à mesma interpretação e processamento.
- **inicio** – Se todos os parâmetros que é necessário configurar estiverem definidos, dá por encerrada a configuração e passa à simulação. Em caso contrário, mostra uma mensagem indicando que há parâmetros em falta, e mantém-se a fase de configuração.

Fase de simulação

Depois de estarem definidas as configurações necessárias e a simulação ter sido iniciada, o utilizador poderá conduzir a simulação através dos comandos que a seguir se apresentam. O utilizador poderá também utilizar os comandos de configuração, se pretender atualizar algum parâmetro, com exceção do comando *defmundo* e *inicio*.

Os comandos apresentados em seguida são ignorados na fase de configuração.

- **ninho <linha> <coluna>** - Coloca o ninho na posição (linha, coluna). Este comando cria implicitamente uma nova comunidade. Os ninhos são identificados por um número. O número é atribuído automaticamente pelo programa, sendo o primeiro o ninho 1, e os restantes 2,3, etc.
- **criaf <F> <T> <N>** - Cria e acrescenta F formigas do tipo T em posições aleatórias vazias. As formigas pertencem implicitamente ao ninho N (identificado por um valor inteiro). T indica o tipo das formigas: C – Cuidadora, V – Vigilante, A – Assaltante, E – Exploradora e S – Surpresa.
- **cria1 <T><N><linha><coluna>** - Cria e acrescenta uma formiga do tipo T, do ninho N, na posição (linha, coluna). T indica o tipo das formigas: C – Cuidadora, V – Vigilante, A – Assaltante, E – Exploradora e S – Surpresa.
- **migalha <linha> <coluna>** - Cria uma migalha na posição (linha, coluna).
- **foca <linha, coluna>** - Nos casos em que não é possível a visualização total da grelha do mundo, pode representar-se apenas uma parte. Este comando define o centro da zona rectangular a ver no ecrã. A dimensão da zona é aquilo que fizer sentido relativamente ao ecrã em questão e à qualidade de visualização.
- **tempo** – Executa a passagem de uma iteração (unidade de tempo).
- **tempo <n>** – Executa a passagem de n iterações (unidades de tempo).
- **energ ninho <N><E>** - Acrescenta um valor de energia E ao ninho N.
- **energ formiga <linha> <coluna> <E>** - Acrescenta um valor de energia E à formiga que estiver na posição (linha, coluna).
- **mata <linha> <coluna>** - Mata a formiga que estiver na posição (linha, coluna).

- **inseticida <N>** - Elimina o ninho N e a sua comunidade.
- **listamundo** – Apresenta uma lista de todos os ninhos e de todas as migalhas com os seus atributos. Apenas é considerado o número de formigas da comunidade de cada ninho e não a sua listagem.
- **listaninho <N>** - Apresenta uma lista de toda a informação sobre o ninho, incluindo a lista das formigas da comunidade que lhe corresponde. Relativamente a cada formiga, deve ser apresentado o tipo, identificação, posição e energia.
- **listaposicao <linha> <coluna>** - Apresenta os elementos que estão localizados na posição (linha, coluna) e seus atributos.
- **guarda <nome>** - Salvaguarda o mundo fazendo uma cópia integral dele em memória. O nome é aquilo que o utilizador quiser e serve para identificar posteriormente esta cópia. O nome “default” não pode ser usado neste comando.
- **muda <nome>** - Muda para uma cópia do mundo previamente guardada. Pode ser usado para voltar para a “versão original” que tem sempre automaticamente o nome “default”.
- **apaga <nome>** - Apaga uma cópia previamente guardada. Se se apagar o mundo “default”, a simulação é terminada. O programa não termina, fica de novo na fase de configuração.
- **sair** - Termina a execução do programa.

6 INTERACÇÃO COM O UTILIZADOR

Em cada passagem de tempo e após cada comando, o programa deve mostrar o quadrado que representa a parte visível do mundo com todos os seus elementos identificados.

Sugere-se também o uso de cores para identificar os elementos da mesma comunidade. Cada elemento poderá ter a cor correspondente à sua comunidade e a letra correspondente ao seu tipo: C – Cuidadora, V – Vigilante, A – Assaltante, E – Exploradora, S – Surpresa e N – ninho.

Sugere-se que, na visualização de cada formiga, a sua identificação apareça com letras maiúsculas se a formiga não tiver fome e com letras minúsculas se tiver fome (energia menor que 50% do valor inicial).

A utilização deste programa fica facilitada se for possível saber os números das linhas e colunas da grelha. Faça por apresentar uma espécie de “régua” nos lados da grelha com esta informação.

7 FINAL DA SIMULAÇÃO

A simulação termina quando o utilizador assim o entender. Não há um cenário de vitória de uma comunidade sobre outra, apesar de poder acontecer uma determinada comunidade ficar sem formiga nenhuma. Por outras palavras, a “detecção de situação de fim de jogo” é algo com que não precisa de se preocupar.