

Trabalho prático

Este enunciado é constituído por duas partes:

Parte A – Regras página 1

Parte B – Descrição do tema.... página 2

Parte A - Regras 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 de biblioteca dados nas aulas. A interface com o utilizador é feita segundo o conceito de consola não-gráfica, e será disponibilizada uma biblioteca para auxiliar nesta tarefa. O tema proposto deixa diversos aspectos em aberto que são de relevância menor para o trabalho e devem ser resolvidos de acordo com a lógica e bom senso.

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

O trabalho é entregue em duas metas.

- **Meta 1 – Prazo: 15 de Dezembro de 2016**

Objetivos e funcionalidade a entregar:

- Configuração da Colónia (e início da simulação)
- Seres apenas com características de movimento básicas para poderem ser vistos.
- Castelo que permite criar novos seres.
- Leitura e interpretação de comandos via teclado e via ficheiro (configuração)
- Avançar um instante.

Defesa: revisão simples a combinar laboratório a laboratório, a ter lugar dias 15 e 16 de Dezembro.

- **Meta 2 – Prazo 15 de Janeiro de 2017**

Objetivos: o programa completo, com relatório.

Entregas com atrasos:

- Até ao dia 22/Jan -> Sem penalização
- Posteriores ao dia 22: 10% por cada dia para além do dia 22.

Defesa: obrigatória, marcada grupo a grupo via moodle.

As entregas serão feitas via moodle, e as indicações serão dadas na devida altura.

O trabalho é sujeito a defesa obrigatória, cuja prestação afeta a nota, podendo inclusivamente levar à perda total da nota. A defesa (e nota resultante) é individual, mas os alunos do grupo fazem a defesa ao mesmo tempo.

As regras adicionais relativas ao trabalho prático encontram-se descritas na ficha da disciplina, cuja consulta é obrigatória antes da colocação de questões quanto a estas regras.

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

Parte B - Descrição do tema: Castle War

1. INTRODUÇÃO

O trabalho prático de POO consiste num simulador de batalhas entre seres vivos imaginários. O simulador tem alguns elementos de estratégia definida pelo utilizador que lhe emprestam algumas características de jogo e o utilizador é referido aqui como jogador e o programa como jogo, mas trata-se mais de um simulador de que um jogo.

O programa pretendido tem o aspecto “modo texto” (sem imagens/gráficos), e a interacção com o utilizador é relativamente simples, cobrindo os aspectos essenciais de controlo da simulação. O controlo do programa é feito através de comandos lidos do teclado ou de ficheiros de texto.

Uma parte muito importante do trabalho reside na forma como o programa é feito, nomeadamente, se os conceitos de orientação a objectos em C++ são correctamente aplicados na obtenção da funcionalidade pretendida.

2. DETALHES DO TEMA - SIMULADOR

Considere uma planície onde existem seres. A planície está organizada como uma grelha de posições, e os seres passam de uma posição para outra, nunca se encontrando a meio entre duas posições. Para efeitos de ordem prática, considere que a planície é rectangular e tem a dimensão configurável, por exemplo 100x100 posições. Os seres são essencialmente todos iguais mas possuem diversas características que os dotam de comportamentos específicos: dois seres com características diferentes irão ter comportamentos diferentes. Na planície podem também existir edifícios. Os edifícios podem ser feitos por ordem do jogador e têm diversos efeitos sobre o jogo, como se verá adiante. Os seres não constroem os edifícios mas podem interagir com eles, como por exemplo: atacar e reparar edifícios. Poderão existir inicialmente alguns edifícios na planície.

1.1 Elementos principais e lógica do simulador

Os seres estão organizados em colónias (nações). Cada colónia compreende um conjunto de seres e um conjunto de edifícios. O jogador controla uma colónia de seres e existirão uma ou mais colónias de seres controlados pelo computador. Os seres poderão atacar os seres e edifícios de outras colónias. O objectivo do jogo é: sobreviver, exterminando as outras colónias (nota: mal ou bem, o computador tentará atingir os mesmos objectivos).

Os seres terão um comportamento ao longo do tempo que é ditado maioritariamente pelas características que possuem: o utilizador não irá dar ordens a seres individuais (seria impraticável). Os edifícios também podem desempenhar acções, que são igualmente autónomas consoante a sua natureza. O jogador controla a sua colónia através de comandos. Estes comandos afectam a colónia como um todo (ver mais adiante) e tem essencialmente a ver com acções tais como atacar / fugir / fazer edifício / vender edifício / fazer seres. O computador também pode dar ordens. O uso de uma inteligência artificial foge ao âmbito da disciplina, por isso aceitam-se ordens sorteadas segundo uma heurística simplificada qualquer (mais adiante são dadas algumas sugestões).

O tempo é medido pela passagem de instantes e não tem relação directa com o relógio do computador. A cada novo instante cada ser pode desempenhar uma acção/comportamento, cada edifício pode efectuar um comportamento/acção, e o jogador/computador pode tomar decisões e dar ordens (atacar, fugir, fazer edifício, etc.). O jogador terá a possibilidade de fazer avançar a simulação 1 instante, ou N instantes de uma vez. Nota: se o jogador fizer avançar N instantes de uma só vez, o computador continuará a ter a possibilidade de emitir as suas ordens em cada um dos instantes).

1.2 Estratégia do jogo/simulação

O objectivo é muito simples: sobreviver, eliminando todas as outras colónias. Para eliminar as outras colónias será necessário eliminar os seus seres e os seus edifícios. Cada colónia tem apenas edifícios e seres.

Cada colónia tem uma economia simples baseada em moedas. Fazer seres e fazer edifícios custa moedas, e a escolha de quais os seres e quais os edifícios que se vão “comprar” é um aspecto importante da estratégia. As moedas podem ser geradas durante a simulação e os seres e os edifícios podem ser feitos (comprados) a qualquer instante desde que a colónia tenha moedas suficientes.

Os edifícios são necessários para fazer seres, fazer moedas, ou para defender. Existe um conjunto limitado de tipos de edifícios e o jogador terá que escolher com cuidado quais os edifícios que vai fazer.

Os seres são necessários para combater. Os seres são basicamente todos do mesmo tipo mas terão capacidades e custos diferenciados consoante características que lhe são atribuídas. Existe um conjunto de características disponíveis que podem ser combinadas de muitas maneiras possíveis, resultando em muitos “tipos” de seres possíveis. No entanto, o jogador terá que definir à priori um conjunto reduzido de perfis de características – ou seja, combinações (“menus”) de características - e durante a simulação só poderá criar seres com esses perfis que definiu antes. A definição desses perfis (conjuntos predefinidos de características) é um dos aspectos mais importantes da estratégia.

Cada característica tem um preço. O preço de um ser será o somatório do preço das características dele. Além disso, cada característica vai exigir algo ao ser (por exemplo, pontos de força), limitando-se assim o número de características que se pode ter em simultâneo no mesmo perfil/ser. Assim é importante escolher bem quais as características para figurar nos perfis. Dois exemplos de perfis seriam: um perfil “ligeiro”, com características que dão velocidade ao ser mas pouca capacidade de ataque, e um perfil “combatente” com características de muito poder de ataque mas lento. Eventualmente, não se conseguirá ter no mesmo perfil características de velocidade e de poder de ataque, ou porque excede a capacidade do ser, ou então porque ficaria muito caro fazer um ser assim. Poderão ser criados dois perfis diferentes e o jogador mais tarde escolhe fazer seres de um perfil ou de outro (desde que os tenha definido previamente).

1.3 Interacção com o utilizador

Visualização

Não é possível ver toda a planície em simultâneo no ecrã. Assim o utilizador define o foco – a zona da planície que deseja ver, e só lhe é apresentada essa porção no ecrã. O jogador pode mudar o foco para outro lado a qualquer altura. O foco é definido pelas coordenadas linha, coluna do canto superior esquerdo da parte rectangular (o foco) do mapa que deseja ver. Essa área não deve ser demasiado pequena e deverá ter a dimensão 20 linhas e 40 colunas pelo menos.

Opcionalmente (para 5% a mais na cotação), pode ser implementado um efeito *zoom-out*: de forma a conter uma zona maior da planície no ecrã, cada 2x2 posições no mapa acumulam numa única posição no ecrã. Pode haver vários níveis de *zoom-out* (2x2 -> 1, 4x4 ->1 etc.). Esta questão é opcional e valorizada como extra.

Não deverá haver mais do que um ser na mesma posição nem mais do que um edifício na mesma posição. Pode haver seres e um edifício na mesma posição apenas no caso em que os seres da colónia estão no seu castelo (e visualmente é o castelo que é apresentado).

O ser é representado por uma letra que indica o perfil do ser, tendo uma cor que representa a colónia a que pertence. Os edifícios seguem uma lógica semelhante.

Deve ser possível obter, para uma colónia indicada, uma lista com os seres e outra lista com os edifícios em que são apresentados todos os detalhes dos seres/edifício, incluindo a posição em que se encontram.

Deve ser possível obter a lista de todos os perfis do jogador, sendo indicado em cada um as características que os compõem, bem como os valores cumulativos da defesa, ataque e velocidade.

Controlo da simulação

O jogador comandará o desenrolar do simulador fazendo avançar a simulação 1 instante ou vários de cada vez. Se avançar vários de uma só vez, ainda assim o computador terá a oportunidade de controlar as suas colónias a cada novo instante. As ordens do jogador são dadas sob a forma de comandos escritos (ver mais adiante).

3. SERES, CARACTERÍSTICAS E EDIFÍCIOS

3.1 Seres

Os seres assemelham-se a formigas no sentido de serem todos parecidos e desempenharem acções de uma forma automática e incansável. Existem alguns aspectos gerais dos seres, que são:

- Saúde: inicialmente 10. Quando chega a zero, o ser morre, desaparecendo sem deixar rasto.
- Força: inicialmente 10. Traduz-se na capacidade de ter mais ou menos características.
- Velocidade: o movimento e a velocidade dependem das características que o ser tem (ver mais adiante acerca de características). Sempre que se fala de movimento, considera-se que pode ser sempre feito na horizontal, vertical ou diagonal.
- Ataque/capacidade destrutiva: depende das características que têm.
- Defesa/capacidade defensiva: depende das características que têm.

O comportamento do ser é baseado nas características que tem. As características são as que estão associadas aos perfis, definidos na fase de configuração. As características têm as seguintes propriedades:

- Cada característica tem um custo em moedas. A construção do ser consome uma determinada quantidade de moedas que é o somatório do custo de cada característica do seu perfil.
- Cada característica “consome” uma determinada quantidade de força ao ser, o que significa que só se consegue ter um conjunto limitado de características em cada ser/perfil.
- Têm um determinado efeito sobre o ser que a tem. Há dois tipos de efeitos:
 - Efeitos que modificam a natureza do ser: mais saúde, velocidade, etc.
 - Efeitos de comportamento: Fazem o ser desempenhar uma determinada acção a cada instante.

Cada característica pode ter 0 ou 1 efeitos de cada tipo. O efeito de todas as características é cumulativo: todas as modificações fazem efeito, e todos os comportamentos são actuados.

- As características podem ser repetidas: os efeitos acumulam.

As características são:

Nome	ID	Custo monetário	Custo força	Efeito
Bandeira	1	1	0	Faz com que o ser pertença ao jogador dessa bandeira. Na prática, faz com que o ser seja de uma certa colónia e reconheça os seres que são dessa mesma colónia ou de outra colónia. Se um ser não tiver esta característica será um pária: nunca atacará ninguém e será atacado por todos.
Superior	2	1	1	Acrescenta 1 ponto de saúde (aumenta a saúde máxima) ao ser.
Pele dura	3	2	2	Acrescenta 1 ponto de defesa ao ser
Armadura	4	2	3	Acrescenta 2 pontos de defesa ao ser
Faca	5	1	1	Acrescenta 1 ponto de ataque ao ser
Espada	6	2	2	Acrescenta 2 pontos de ataque ao ser. Nas duas primeiras vezes que são usadas dão 3 pontos de ataque. Para voltar a dar 3 pontos em vez de 2 o ser tem que regressar ao castelo.
Agressão	7	1	1	Faz com que o ser ataque um ser de uma bandeira diferente da sua se estiver adjacente (todas as características podem ser repetidas: se tiver duas "Agressão" ataca duas vezes.
Ecológico	8	1	1	Parecido com "agressão" mas o alvo do ataque são edifícios inimigos. Se vir um edifício inimigo na posição ao lado ataca-o.
HeatSeeker	9	1	1	Procura inimigos (seres de outras colónias) e vai até junto deles (mas não os ataca. Quem faz isso é outra característica). Na prática procura o ser inimigo mais próximo. Se não houver inimigos volta ao seu castelo. Avança na direcção do ser 1 posição de cada vez.
BuildSeeker	10	1	1	Parecido com HeatSeeker mas procura edifícios em vez de seres.
Walker	11	1	1	Anda ao calhas quando fora do castelo. Pode ser que encontre algum ser ou edifício inimigo (e se encontrar, talvez os ataque se tiver as características necessárias).
Remédio	12	2	1	Faz com que o ser recupere 2 pontos de saúde quando a sua saúde atinge 3 ou menos. Só pode ser usado uma vez em cada saída do castelo (relativo ao ser que a tem).
SecondChance	13	3	0	Dá uma segunda vida ao ser: quando este morre, volta a aparecer no castelo, mas sem esta característica.
Aluno	14	?	?	Faz o que os alunos do grupo decidirem (se for igual a outro grupo vai mesmo dar nas vistas)

As características são opcionais – nenhuma é obrigatória. As características podem ser repetidas e o seu efeito acumula. Isso pode fazer sentido ou não. Alguns exemplos:

- Pode-se repetir a Bandeira. Mas isso não tem interesse nenhum.
- Pode não haver bandeira nenhuma. Nesse caso o ser não reconhece nem amigos nem inimigos e será atacado por todos. Pouco interessante.
- Pode-se ter duas espadas. Ataca-se com mais força mas o ser fica muito caro ou terá outra característica a menos por causa do máximo de pontos de força.
- Pode-se ter vários HeatSeeker. O ser dá vários passos na direcção do ser inimigo mais próximo em vez de um, sendo mais rápido. Bom para fazer "caçadores", à custa de não ter outra característica qualquer.
- Pode-se ter HeatSeeker e BuildSeeker: dá-se um passo na direcção do ser inimigo mais próximo e também outro passo em (outra) direcção do edifício inimigo mais próximo. É capaz de não funcionar muito bem.
- Pode-se ter Agressão e Ecológico: ataca tanto seres inimigos como edifícios, desde que estejam numa posição adjacente. Se não tiver nenhuma destas duas características o ser é muito pacífico: "leva e não dá".

Acerca de movimento e ataque:

- As características que fazem mover deslocam o ser uma posição na horizontal, vertical ou diagonal. Se houver várias características destas, cada uma faz o seu efeito de mover e o ser desloca-se várias posições no mesmo instante.
- A planície tem fronteiras fechadas: quando um ser atinge o limite não transita para o extremo oposto. Simplesmente não pode avançar mais nessa direcção.
- O ser só ataca (conforme as características que tem) os inimigos que estejam em posições adjacentes (horizontal, vertical ou diagonal).

Os seres são construídos por comando do jogador desde que o castelo da sua colónia não tenha sido destruído.

3.2 Edifícios

Os edifícios são úteis essencialmente para atacar inimigos, para poder fazer seres, ou para fazer moedas. Os edifícios têm as seguintes propriedades:

- São feitos por comando do utilizador e a sua construção é instantânea. A construção tem um custo em moedas.
- São sujeitos a destruição pelos inimigos mas podem ser reparados desde que não tenham sido totalmente destruídos.
- A reparação de um edifício é feita por comando do jogador, é instantânea e custa moedas na proporção do grau de destruição face ao custo inicial do edifício.
- Alguns edifícios podem ser “melhorados” (upgrades). Cada novo melhoramento custa dinheiro e melhora o funcionamento do edifício.
- Os edifícios podem ser vendidos, recuperando-se metade das moedas correspondentes ao preço + melhoramentos que tenham sido feitos nele.

Existem os seguintes edifícios.

Nome	Custo monetário	Saúde	Defesa	Efeito
Castelo	0	50	10	Fabrica seres quando pedido por comandos do utilizador. Não pode ser construído nem vendido por comando, mas pode ser reparado.
Torre	30	20	10	Ataca todos os inimigos à distância 2 com força de ataque 3. Cada novo nível custa 10 e traz mais 2 de defesa e 1 de ataque.
Quinta	20	20	10	Fabrica 2 moedas por cada instante. Cada novo nível custa 10 e traz mais 1 de defesa e 1 moeda adicional por instante.

3.3 Ataques

Os ataques ocorrem automaticamente sempre que os seres com as características certas encontram inimigos em posições adjacentes, ou sempre que uma torre “vir” um inimigo a duas posições ou menos de distância.

A cada instante todos os seres e todos os edifícios têm oportunidade de atacar (conforme as suas características), não sendo importante quem ataca primeiro porque os seres e edifícios só são considerados mortos no final de todos agirem.

O ataque é simplesmente um efeito possível de entre as várias coisas que as características podem fazer, e segue a seguinte lógica:

Em cada instante:

- 1º - Os edifícios actuam/atacam
- 2º - Os seres actuam/atacam
- 3º - Identificam-se os seres mortos / edifícios destruídos e eliminam-se.

Ou seja, um ser que seja destruído por outro ainda terá oportunidade de ripostar porque só será considerado morto no final do instante.

O ataque é muito simples, válido tanto para seres como para edifícios:

- Se a força de ataque fa do atacante for maior que a defesa fd do atacado, o atacado perde tantos pontos de saúde quanto $fa-fd$.
- Caso contrário o atacado perde um ponto de saúde.

4. DESENROLAR DO SIMULADOR/JOGO

A simulação decorre em duas fases.

- Na primeira fase é feita toda a configuração: os perfis de seres, o número de oponentes que vai existir, a quantidade de moedas inicialmente disponível, os edifícios iniciais, etc. – os aspectos configuráveis são apresentados ao longo de todo o enunciado.
- Na segunda fase tem lugar a simulação propriamente dita. A simulação tem início quando o jogador dá por terminada a configuração e manda iniciar. Durante a simulação o jogador e o computador controlam as suas colónias. A simulação pára quando é detectado o fim de todas as colónias menos uma que é a vencedora.

4.1 Fase de configuração:

Existe sempre 1 jogador. Haverá um oponente (ou mais) controlado pelo computador. O número de oponentes é configurável. Antes da simulação começar, o jogador define 5 perfis de características à sua escolha. Durante a simulação apenas poderá usar esses tipos de seres, portanto, a definição dos tipos de seres com que vai jogar é estratégica. O computador copia 3 desses tipos predefinidos e sorteia outros 2 (escolhendo de um conjunto já definido no código, ou então sorteando dinamicamente (“ao calhas” ou algo melhor que isso – esforce-se um pouco que vai ver que consegue).

Os comandos de configuração são os indicados abaixo. As palavras em *itálico* são para substituir por uma letra, valor numérico ou sequência de caracteres conforme o caso:

dim <i>linhas</i> <i>colunas</i>	Define a dimensão da planície.
moedas <i>numero</i>	Define a quantidade de moedas iniciais das colónias.
oponentes <i>numero</i>	Define quantas colónias controladas pelo computador vai haver. Deve haver pelo menos uma. A colónia do jogador é identificada pela letra ‘a’ e as colónias do computador são ‘b’, ‘c’, etc.
castelo <i>colónia lin col</i>	Define a posição do castelo da colónia indicada. As colónias têm sempre um castelo inicialmente em posições aleatórias. Este comando permite mudar o castelo para outro local.

mkperfil letra	Define um perfil identificado por uma letra especificada pelo jogador. O perfil está inicialmente vazio (sem características).
addperfil letra característica	Adiciona a característica indicada ao perfil identificado pela letra. O perfil não pode ultrapassar o limite de força. As características são identificadas pelo ID numérico dado atrás.
subperfil letra característica	Remove a característica indicada ao perfil especificado.
rmperfil letra	Abandona (remove) o perfil especificado pela letra
load ficheiro	Carrega comandos de configuração lendo-os do ficheiro de texto indicado. O ficheiro está organizado em linhas, um comando por cada linha, e o seu processamento é igual tal como se tivessem sido lidos do teclado (ler de um teclado ou ler de um ficheiro vai dar ao mesmo e é indiferente para o seu processamento). Vão ser dados exemplos de leitura de ficheiros.
inicio	Dá por encerrada a configuração e passa à simulação

Os comandos de simulação (mais adiante) que surjam durante a fase de configuração serão simplesmente ignorados com um breve aviso ao jogador.

4.2 Fase de simulação

Durante a simulação o jogador pode mandar fazer seres, pagando o seu custo, os quais aparecem inicialmente no edifício “castelo”. O jogador pode mandar fazer edifícios. O jogador também pode vender (reciclar) edifícios seus para obter dinheiro (o castelo não pode ser vendido).

Depois da simulação começar, o jogador vai dar ordens que correspondem por alto às seguintes acções:

- Comprar edifícios, colocando-os em determinadas posições, desde que fique a não mais de 10 de distância do seu castelo. Cada edifício tem um preço.
- Efectuar *upgrade* em edifícios, melhorando as suas características. Cada *upgrade* tem um preço.
- Vender (reciclar edifícios) para recuperar algum dinheiro para emergências. Quando vendidos, os edifícios valem metade do dinheiro que já foi gasto nele. Os edifícios têm uma identificação qualquer que permita ao jogador especificar o edifício alvo das suas ordens.
- Atacar: faz com que os seus seres saiam do castelo, se desloquem e ajam consoante as suas características.
- Recuar: faz com que os seus seres voltem todos para o seu castelo, à velocidade fixa de 1 posição (horizontal, vertical ou diagonal) por instante. Esta ordem sobrepõe-se ao comportamento das características dos seres.
- Os seres estão inicialmente no castelo. Quando estão no castelo recuperam um ponto de saúde por instante, até ao máximo de cada ser.

Os comandos de simulação são os indicados abaixo. As palavras em itálico são para substituir por uma letra, valor numérico ou sequência de caracteres conforme o caso:

foco linha coluna	Define o foco: área rectangular da planície a ver no ecrã. A altura e largura são fixas e adaptadas à dimensão do ecrã (mínimo: 20x40).
zoomout n	Define o efeito de <i>zoom-out</i> como factor de “des-ampliação” <i>n</i> (1: 1 posição da planície → 1 posição no ecrã, 2: 2x2 posições na planície → 1 posição no ecrã, etc.). Este comando é opcional e valorizado como extra.

setmoedas <i>colonia num</i>	adiciona ou retira <i>num</i> moedas à colónia <i>colonia</i> . Serve essencialmente para testar (<i>debug</i>) e não é suposto ser usado durante um jogo “a sério”).
build <i>edif lin col</i>	Constrói um edifício <i>edif</i> (nome do edifício) nas coordenadas <i>lin,col</i> respeitando o custo dele e desde que esteja a não mais de 10 posições de distância do castelo da colónia. O programa atribui um número <i>EID</i> ao edifício para o jogador poder identificar esse edifício em comandos posteriores.
list <i>colonia</i>	Apresenta uma lista com tudo sobre a colónia indicada. Isto inclui os edifícios, as suas posições saúde, ataque, etc., os seres: os seus perfis, posições, saúde etc.
listp <i>perfil</i>	Apresenta os detalhes do perfil especificado: a lista das características do perfil, incluindo os valores cumulativos da defesa, ataque e velocidade.
listallp	Semelhante ao anterior, mas para todos os perfis do jogador e do computador.
mkbuild <i>edif lin col colonia</i>	Adiciona um edifício <i>edif</i> (nome do edifício) pertencendo à colónia indicada nas coordenadas <i>lin,col</i> à borla e sem qualquer restrição de posição. Serve essencialmente para testar (<i>debug</i>) e não é suposto ser usado num jogo “a sério”).
repair <i>EID</i>	Repara o edifício indicado por <i>EID</i> (da colónia do jogador).
upgrade <i>EID</i>	Melhora o edifício <i>EID</i> respeitando o custo dessa operação.
sell <i>EID</i>	Vende o edifício <i>EID</i> da colónia do jogador.
ser <i>num perf</i>	Faz num seres com o perfil <i>perf</i> respeitando o custo deles.
next	Avança um instante na simulação.
nextn <i>num</i>	Avança num instantes de uma só vez. Em cada instante desenrolam-se as acções dos seres e edifícios, do jogador e do computador.
ataka	Faz os seres da colónia do jogador saírem do castelo, actuar as suas características.
recolhe	Faz os seres recolherem ao castelo. As características deixam de actuar: os seres deslocam-se à velocidade 1 posição de cada vez em direcção ao castelo. Uma vez no castelo, os seres recuperam um ponto de saúde por cada instante.
fim	Termina a simulação, mesmo que vá a meio (numa situação normal a simulação termina quando só resta uma colónia).
save <i>nome</i>	Faz uma cópia da simulação para memória (uma espécie de <i>savegame</i> sem envolver ficheiros) – é uma cópia de toda a informação. A cópia fica associada a um nome para poder ser resposto mais tarde. Pode haver várias cópias.
restore <i>nome</i>	Repõe em acção a cópia/ <i>savegame</i> com o nome indicado.
erase <i>nome</i>	Elimina a cópia/ <i>savegame</i> indicada, libertando a memória por ela ocupada.
load <i>ficheiro</i>	Carrega comandos de simulação lendo-os do ficheiro de texto indicado. O ficheiro está organizado em linhas, um comando por cada linha. Esta funcionalidade é a mesma que o comando <i>load</i> da fase de configuração.

Durante a simulação serão ignorados os comandos de configuração que surjam (com um breve aviso ao jogador).

5. COLÓNIAS CONTROLADAS PELO COMPUTADOR

O computador irá controlar uma ou mais colónias. O computador controlará as suas colónias de forma análoga ao jogador: através de ordens. A forma como essas ordens são concretizadas pode ser (sugestão):

- Através do envio de cadeias de caracteres para as mesmas funções que já tratam das ordens do jogador. Neste caso a origem dos “comandos” seria o computador e não o teclado. O computador teria um dicionário de ordens predefinidas que vai emitindo consoante um algoritmo (aleatório?) qualquer.
- Construção de funções à parte para lidar com as ordens e acções do computador. A ideia anterior talvez poupe código.

O que é importante a reter é que o computador apenas emite ordens tal como o jogador. O comportamento dos seus seres e edifícios é autónomo e funciona tal como os seres do jogador. O computador irá copiar três dos perfis definidos pelo jogador e sortear (ou definir de forma inteligente segundo uma ideia qualquer) os restantes 2. Após isso irá emitir ordens. Constrói, ataca, recolhe, etc. Isto pode ser baseado num algoritmo inteligente, que ultrapassa os objectivos da disciplina, ou pode ser baseado em geração de número aleatórios (exemplo: sortear um número entre 1 e 10. Se sair 1 não faz nada, se sair 2 ataca, etc.). Usar exclusivamente aleatoriedade faz com que o computador não jogue de forma interessante. Sugere-se um pequeno esforço para tomar algumas decisões semi-inteligentes: se estiver a ser atacado então ataco também, se os meus seres estiverem quase a morrer mando recolher, se tiver muito dinheiro mando fazer mais seres, etc. Não custa muito e fica mais interessante.

Sugestão: foque-se primeiro na situação em que o computador controla apenas uma colónia. Quando tiver isso a funcionar acrescente algo para passar a controlar de mais do que uma. O importante é controlar pelo menos uma.