

Sistemas Inteligentes I - Trabalho de grupo

Introdução

O presente trabalho envolve a gestão de recursos de uma base militar, e a proteção desta através da construção e coordenação de um “exército”, num ambiente 2D simulado.

Este exército defensor, cujo desenvolvimento da sua estratégia é o principal objectivo deste trabalho, irá combater um exército invasor, existente no ambiente, sendo que ambos serão constituídos por soldados.

O jogo está organizado em turnos, sendo que os exércitos defensor e invasor executam as suas ações em cada turno, primeiro o defensor e depois o invasor. O jogo terá MAX_T (=2000) turnos no máximo.

Pretende-se que o exército defensor tome as suas decisões (escolha de ações) de forma a impedir que soldados do exército invasor alcancem a sua base. Quanto mais turnos o exército defensor conseguir aguentar, sem que a sua base seja atacada, melhor será a sua pontuação.

O exército defensor pode executar um conjunto de ações em cada turno. Algumas destas ações custam recursos, recursos esses que são fornecidos ao exército defensor no fim de cada turno. Apesar do objectivo principal do exército defensor ser a defesa da sua base, este exército pode também atacar o território do invasor, fazendo chegar soldados seus à coluna mais à direita na matriz. O jogo termina quando um dos soldados invasores consegue movimentar-se para a posição da base, ou quando se atinge o último turno.

NOTA: Existem diferentes níveis de dificuldade que irão afetar os recursos recebidos e o custo das decisões. Neste momento, apenas o primeiro nível de dificuldade (nível 0) está completamente desenvolvido. Dentro de cada nível de dificuldade, o custo inicial da base e a respetiva produção podem variar.

Características do ambiente

O ambiente tem a forma de uma matriz de HEIGHT (11) linhas por WIDTH (3?) colunas.

Em cada turno do invasor, são criados novos soldados invasores nas células do território do invasor que são apenas as da última coluna da matriz (lado direito). O número de novos soldados por célula é dado por:

```
spawnsoldiers = 2 + int( (max(turn - retard, turn/3)**2)/65)
```

Em que *turn* é o número do turno actual e *retard* é a penalização aplicada ao inimigo.

O desconto/penalização *retard* é agravado sempre que o exército defensor ataca o território do inimigo. O território do inimigo considera-se atacado quando soldados defensores alcançam a coluna mais à direita na matriz. O agravamento do desconto é dado pela seguinte expressão:

```
retard += cell[1]/5.0
```

Em que *cell* é uma célula do território inimigo e *cell[1]* é o número de soldados defensores que alcançaram essa célula.

O comportamento dos soldados invasores é essencialmente reactivo e sem memória.

Não existe limite no número de soldados defensores por célula. Estes podem movimentar-se parcialmente ou totalmente, de uma célula para outra adjacente.

Tipos de soldados

A unidade elementar do exército é o soldado. Vários soldados podem estar presentes na mesma célula. Os soldados podem mover-se e entrar em confrontos. Este confronto pode acontecer a curto alcance (soldados *melee*) ou a longo alcance (soldados *ranged*). Mais em detalhe, os tipos de soldados são os seguintes:

- **Soldados *ranged*:** Estes soldados atacam à distância (3 Manhattan). Para atacarem/eliminarem inimigos, basta estarem estáticos no turno em questão. Atacam em todas as direções, sendo que um bom posicionamento destes soldados é uma das chaves principais para o sucesso.
- **Soldados *melee*:** Estes soldados apenas entram em confrontos directos na mesma célula. Têm também uma habilidade especial estática: quando se encontram menos de 20 *melee* numa célula, eles ficam “invisíveis” para os inimigos, embora não fiquem invulneráveis. A existência de soldados invisíveis numa célula é visualizada através da representação de um arbusto. Deve-se jogar com esta funcionalidade ao seu máximo potencial, tanto para distrair brevemente os inimigos como para atacar a base adversária como ataque ninja. Para estes ataques é preciso desviar-se dos soldados inimigos e ter cuidado para não criar formações visíveis no processo.

O exército invasor apenas tem soldados *melee*.

Ações do exército defensor

As ações mencionadas podem ser de três tipos:

- `upgradeBase()` - Evolui o nível da base, i.e. aumenta o nível da base, melhorando a quantidade de recursos recebidos. O custo desta decisão irá variar consoante o nível atual da base (ver adiante “Recursos”).
- `recruitSoldiers(SoldType, NumSold[, Pos])` - Recrutar soldados, i.e. adicionar novos soldados ao exército defensor numa posição junto à base. O 3º argumento, se fornecido, tem que ser uma posição adjacente à base. Se não for fornecido o 3º argumento, os novos soldados são colocados à direita desta (deslocamento $x=+1$). O custo desta decisão irá variar consoante o tipo de soldado adquirido. O custo dos soldados *ranged* (150) é o triplo do custo dos soldados *melee* (50).
- `moveSoldiers(Pos1, Pos2, NumSold)` - Mover soldados, i.e. reposicionar soldados do exército. Ao mover um soldado para a posição de um soldado inimigo, esta ação resulta num ataque. Um soldado não poderá ser movido para a posição de um soldado aliado de outro tipo. Os soldados apenas se podem mover para células adjacentes à posição de origem.

Não existe uma ação específica para atacar os soldados ou o território do invasor. Os ataques resultam dos movimentos e posicionamentos dos soldados (ver adiante “Regras sobre confrontos entre soldados”).

O exército defensor poderá executar 0 até `MAX_ACTIONS` (500) ações por turno. Algumas destas ações custam recursos, recursos esses que são fornecidos ao exército defensor no fim de cada turno (ver adiante “Recursos”).

Para atacar o território do invasor, o exército defensor faz chegar soldados seus à coluna mais à direita na matriz. Estes ataques sacrificam os próprios soldados, mas o custo deles é reembolsado em dobro. Além disso, por cada soldado sacrificado no território inimigo, é aplicada uma penalização cumulativa na produção de soldados inimigos (ver acima “Características do ambiente”).

Do lado do servidor do jogo, os três tipos de ações são executados pela ordem apresentada acima. Dentro de cada tipo, as várias ações são executadas sequencialmente pelo servidor por uma ordem arbitrária. Conceptualmente, as ações de um mesmo tipo são executadas em paralelo. A consistência do estado é verificada após a execução de todas as ações.

A execução das ações do exército defensor produz efeito antes da execução das ações do inimigo no mesmo turno.

Recursos

Os recursos são fornecidos ao exército defensor no fim de cada turno. A quantidade recebida irá depender do nível atual da base.

R_0 -- Custo da base no nível inicial.

Custo de evolução: incrementa 40% a cada nível da base.

$$R_n = 1.4 R_{n-1} = 1.4^n R_0 \quad (n > 0)$$

P_0 -- Valor dos recursos fornecidos em cada turno no nível inicial da base.

Custo de evolução: incrementa 20% a cada nível.

$$P_n = 1.2 P_{n-1} = 1.2^n P_0 \quad (n > 0)$$

Recursos adicionais podem ser obtidos através de ataques à base adversária. Neste caso, o exército defensor recebe um reembolso em recursos correspondente ao dobro do custo dos soldados que foram sacrificados.

Regras sobre confrontos entre soldados

Soldados *melee* (invasores) vs. soldados *melee* (defensores): Em confrontos de soldados de curto alcance, cada unidade mata uma e morre também.

Soldados *melee* (invasores) vs. soldados *ranged* (defensores): Em confrontos de soldados *melee* vs. *ranged* em curto alcance os soldados *ranged* morrem todos independentemente do número de unidades de cada parte. Em confrontos à distância, os soldados *ranged* não morrem e disparam num alcance de exatamente 3 Manhattan sendo que cada soldado *ranged* mata 1 soldado inimigo por célula, mas só têm efeito de ataque as unidades estáticas, isto é, unidades às quais não foram aplicadas ações no turno em questão, e no máximo apenas 500 por célula.

Gestão dos turnos no servidor

1. Tomada de ações, validando a sintaxe e terminologia de ações.
2. Validar todas as compras, tanto de soldados como do edifício; restrições apenas um de edifício por turno e apenas um recrutamento de soldados, se a localização do recrutamento dos soldados estiver ocupado por soldados de outro tipo ou inimigos, esta ação será ignorada.
3. Processamento dos movimentos de soldados aliados que colidem com uma célula ocupada pelo inimigo.
4. Processamento e validação dos movimentos de soldados aliados, todas as células resultantes devem ter tipos diferentes de soldados.

5. Desenhar o estado atual do ambiente.
6. Ataque direto dos soldados inimigas quando possível.
7. Movimentação dos restantes soldados sem sobreposição de tipos de soldados (os alunos podem ser encorajados a tirar ideias de como evitar colisões).
8. Spawn de soldados inimigos no canto contrário ao edifício principal.
9. Atribuição de recursos de acordo com o nível do edifício.
10. Desenhar o estado atual do ambiente.
11. Fim do turno.

Código base

O código base do trabalho encontra-se no seguinte repositório, a partir do qual cada grupo deve gerar o seu próprio repositório:

<https://classroom.github.com/a/jzJJ5lDm>

O repositório de cada grupo deve ser criado por um dos membros do grupo, que depois fará a associação dos restantes membros.

Para executar o jogo, pode-se correr os seguintes comandos:

```
python3 pipe.py python3 server.py -dif 0 -- python3 client.py  
python3 pipe.py python3 server.py -dif 1 -- python3 myarmy.py
```

Entregas e prazos

Estão previstas duas entregas:

1ª entrega: 2022/01/07 (peso: 1/3)

2ª entrega: 2022/02/01 (peso: 2/3)

A primeira entrega consiste apenas do código Python desenvolvido. O módulo principal deve chamar-se “myarmy.py” (deve ser copiado do client.py), e poderá importar outros módulos. Na entrega final, além do código, deverá ser entregue um relatório e uma apresentação PowerPoint, que será também apresentada oralmente.

NOTA: O programa desenvolvido vai ser testado com os módulos fornecidos, e não com versões eventualmente modificadas pelos alunos.

Formação de grupos

O trabalho deverá ser realizado por grupos de 2 ou 3 alunos. Em casos bem justificados, poderão ser aceites trabalhos realizados de forma individual.

Já estão formados os seguintes grupos:

- Miguel Silva + Carlos Lopes
- Pedro Carvalho + Tiago Pinho + Beatriz Costa
- Pedro Tavares + Pedro Amaral + Luís Silva
- João Santos (trabalhador-estudante)