

Gestão de recursos e proteção de uma base militar

Beatriz Costa
Universidade de Aveiro
Aveiro, Portugal
Número Mecanográfico: 109657

Pedro Carvalho
Universidade de Aveiro
Aveiro, Portugal
Número Mecanográfico: 84670

Tiago Pinho
Universidade de Aveiro
Aveiro, Portugal
Número Mecanográfico: 92938

Abstract—O presente documento inclui a exposição de todo o trabalho desenvolvido no âmbito da unidade curricular de Sistemas Inteligentes I, do Mestrado de Robótica e Sistemas Inteligentes da Universidade de Aveiro.

I. INTRODUÇÃO

Neste projeto pretende-se fazer a gestão de recursos de uma base militar, através de um ambiente simulado. Esta gestão inclui como objetivos principais reunir os recursos necessários para a construção e coordenação de um exército que consiga proteger a sua base militar dos seus inimigos.

II. ARQUITETURA GERAL

Tal como mencionado anteriormente, neste trabalho pretende-se fazer a gestão de recursos de uma base militar, através de um ambiente simulado. Como tal, um dos principais objetivos consiste no desenvolvimento de uma estratégia que permita defender a base do exército defensor, isto é, impedir que soldados do exército invasor alcancem essa mesma base.

Tendo em conta que o jogo tem no máximo 2000 turnos ou que termina quando é aplicado um *retard* de 20000, a estratégia engendrada neste projeto passa por infligir o máximo de penalização, *retard*, possível ao exército inimigo.

Para aplicar penalização ao exército inimigo temos de atacar o território inimigo, sendo que, considera-se que o mesmo foi atacado sempre que soldados defensores alcancem a coluna mais à direita da matriz. A equação 1 determina o *retard*.

$$retard+ = cell[1]/5.0 \quad (1)$$

, em que $cell[1]$ é o número de soldados defensores que alcançaram a coluna mais à direita.

As vantagens de aplicar esta penalização consiste no facto de o número de *spawn soldiers* diminuir, até um certo limite, e que o custo gasto nas tropas "sacrificadas" - que atacaram o inimigo - seja reembolsado no dobro.

Considerando estas vantagens, a estratégia conceptualizada assenta no objetivo de aplicar o máximo de *retard* possível, de forma a oprimir as tropas inimigas, e assim dominar o jogo, acabando por ganhá-lo.

A estratégia traçada foi dividida em três fases, sendo estas:

- 1) *Early Game*
- 2) *Mid Game*
- 3) *Late Game*

A. Early Game

No início do jogo focámo-nos em utilizar os soldados iniciais para começar o nosso ataque. Para atacar o inimigo estabelecemos que as tropas *ranged* atacam na linha mais acima da matriz, atraindo assim as tropas inimigas, e por baixo atacam os soldados *melees* invisíveis. Para este ataque ter os melhores resultados possíveis, as tropas *melees* têm de avançar com um atraso face às tropas *ranged*.

Enquanto utilizamos as tropas iniciais para atacar, vamos investindo todos os recursos para dar *upgrade* ao nível da base, permitindo assim produzir mais recursos nos próximos turnos. Quando o nível da base chega a sete transita-se para a fase de *Mid Game*.

B. Mid Game

Ao ter a base a nível sete, tal como mencionado anteriormente, entramos na fase de *Mid Game*. Nesta fase focámo-nos em gastar inicialmente os recursos na compra de tropas *melees*, tendo assim sempre tropas invisíveis a atacar o inimigo e, conseqüentemente, a aumentar a penalização. Após comprar os soldados *melees*, compramos *ranged*, tendo em conta o número de inimigos que vão surgindo durante cada turno. Estes soldados irão também atacar o inimigo.

Após este dispêndio de recursos, vamos verificar se estamos a dominar o jogo, isto é, se estamos a oprimir o suficiente o inimigo, o que se traduz num alto *retard*, e caso estejamos e tenhamos recursos, evoluímos a base.

Por fim, o dinheiro que sobrar, investimos em mais soldados *ranged*, ficando assim, com uma maior defesa/ataque e com mais segurança a defender a base.

Assim que o número de tropas *ranged* enviados supere o número ideal, 500, pois este é o máximo de tropas que podem atacar, passamos para a fase de *Late Game*.

C. Late Game

Nesta última fase, o nosso objetivo é fazer *All-In*, ou seja, sendo que os soldados *ranged* perderam um pouco a qualidade que os destacavam perante as outras tropas, investimos maioritariamente (daí o termo *All-In*) em tropas *melees*, não deixando de comprar *ranged*, mas não na mesma razão que na fase anterior.

Esta mudança de atitude permite-nos continuar a dominar o inimigo e a aplicar penalização, fazendo assim com que eventualmente tenha-se subjugado tanto o inimigo que o *retard* chega ao máximo e nós ganhamos o jogo.

III. PRINCIPAIS ALGORITMOS E PARÂMETROS

A. Movimentação e Compra

Nesta secção, apresentam-se os algoritmos de movimentação e de compra respetivos a cada tipo de tropas. Expõe-se de igual forma a metodologia por detrás do processo de *upgrade* da base militar.

1) *Ranged*: Esta tropa é o nosso principal meio de dano. Tem a vantagem de poder, ao contrário das tropas *melees*, atacar sem auto-sacrifício. Esta tropa possui uma área de ataque específica, sendo que, o dano é igual em qualquer ponto da área. Estas características fazem com que estes soldados sejam extremamente valiosos. Os *ranged* não conseguem ficar invisíveis e são mais caros que os *melee*. Assim, a estratégia de ataque é definida de forma a tirar o melhor proveito das suas características.

Primeiramente enuncia-se o algoritmo de movimentação. Este encontra-se descrito pela função *ActionRange()*. De um ponto de vista mais abrangente, esta tropa irá se manter sempre no topo da *board* com o principal objetivo de atrair as tropas inimigas dando espaço para os *melee* poderem avançar, reduzindo ao máximo a perda de unidades.

Estas irão se mover em, geralmente, grupos de três, com o propósito de maximizar a área de ataque de três células de distância (*manhattan*). Todos os membros, exceto o último, movem metade das suas unidades, enquanto o último membro move sempre no seu total. Caso haja inimigos dentro da área de ataque, é garantido que ficará estática a quantidade de *ranged* necessária para matar o grupo de inimigos com maior número, frisando que os *ranged* quando movidos, não atacam.

Em relação à decisão de compra destas tropas, a mesma é feita após a compra de *melees*, logo é segunda prioridade a nível económico. Para a quantidade de *ranged* comprados utilizamos o parâmetro de controlo *enemies_next5rounds*, que será melhor explicado na secção **Parâmetros de Controlo**, em III-C. Este estima o número de inimigos que irão estar nas seguintes rondas, e neste caso, compramos um terço dos inimigos que vão surgir após cinco rondas. Como referido na secção **Arquitetura Geral**, em II, existe uma segunda instância de compra de tropas. Nesta instância, utiliza-se a maioria dos recursos restantes para a compra de *ranged*.

2) *Melee*: O propósito desta tropa é constantemente incrementar o *retard*, pois é mais barata e tem a capacidade de ficar invisível caso o número de unidades seja menor ou igual a 20. Tendo os *ranged* responsáveis por eliminar inimigos bem como chamar à atenção dos mesmos, os *melee* são a tropa mais eficiente para uma estratégia cujo objetivo é ganhar o jogo por maximização de *retard*.

A movimentação dos *melee* é extremamente simples. Esta é feita na função *ActionMelee()*. Estes apenas percorrerão as células mais abaixo da matriz. Desta maneira aproveitam ao máximo o facto dos inimigos estarem constantemente a subir para atacar os *ranged*. O fator de invisibilidade é controlado na compra, e como tal, a quantidade de unidades movimentada pode ser sempre o seu total.

A compra segue um algoritmo que tem em conta o nível da base e o turno atual, preservando sempre o número máximo de unidades a 20 para manter a invisibilidade. Este algoritmo rege-se pela seguinte expressão:

$$\min(\max(\text{floor}(\text{building level}/0.8), \text{turno}/100 + 8), 20) \quad (2)$$

No caso de estarmos em fase *Late Game*, como a quantidade de *ranged* ultrapassa os 500 e estamos a dominar completamente o jogo, podemos então utilizar os recursos restantes para comprar *melees*, acabando por comprar uma quantidade de soldados superior à quantidade de tropas inimigas, perdendo a invisibilidade, no entanto, conseguindo eliminar qualquer tropa que apareça no caminho. Desta maneira, fazemos o referido *All-in* em que a quantidade de tropas *melee* que irá chegar ao fim, poderá atingir até valores como 2000, o que aumenta muito o valor de *retard*.

3) *Base*: A evolução da base militar dependerá da fase do jogo. Enumeramos seguidamente as diferentes situações em que são feitos *upgrades* à mesma:

- 1) *Early Game*: Nesta fase o foco é exclusivamente o *upgrade* da Base, logo será feita sempre que possível;
- 2) *Mid Game*: O *upgrade* será feito apenas na situação de dominância em relação ao inimigo, e nessas rondas que tenha recursos para evoluir a base, este processo será tomado até chegar ao nível 17;
- 3) *Late Game*: Neste momento a base apenas é evoluída quando os recursos disponíveis superam três vezes o custo de evolução. Para tal valor ser atingido significa que tem-se grande abundância de recursos e que um *upgrade* não terá impacto direto e negativo na estratégia de *All-in*.

B. Parâmetros de transição de fases

Como existem três fases, existirão duas situações de transição. Estas são:

1) *Early to Mid Game*: A passagem de *Early to Mid Game* é caracterizada apenas pelo nível da base. Como em *Early Game* o foco é atingir o nível sete, quando este for atingido, será passado à fase de *Mid Game*.

2) *Mid to Late Game*: Esta transição é mais demorada que a anterior, pois depende da quantidade de *ranged* a serem comprados. Caso este número atinja os 500, começamos a ter indícios de *Late Game*, onde se aplica a estratégia de *All-in*. A razão desta transição ser demorada é devido à dependência dos recursos disponíveis em cada turno, o que nem sempre permite fazer esta operação.

C. Parâmetros de controlo

Como parâmetros de controlo criaram-se variáveis que especificam as ações a realizar. Estas variáveis têm o objetivo de calcular:

- *retard*;
- *retard* máximo;
- *retard* necessário;

- número de soldados inimigos que irão surgir nas próximas cinco rondas.

1) Retard: O cálculo do parâmetro *retard* realiza-se antes da ação de movimentação.

Para realizar este cálculo é verificado no momento da ação se os soldados que se irão mover têm tropas inimigas à frente ou não. Caso não tenham, eles avançam, caso tenham, eles ficam parados.

Caso as tropas que estão na penúltima célula avancem, dá-se então *update* do *retard* através da equação 3.

$$retard = n^{\circ} \text{ de tropas que avançam} / 5 + retard \quad (3)$$

Ao realizar o cálculo segundo este procedimento, o seu resultado é impreciso perante o esperado, no entanto, essa imprecisão é desprezável perante a utilidade desta variável.

2) Retard máximo: O parâmetro *retard* máximo é calculado segundo a equação 4, abaixo demonstrada.

$$retard \text{ máximo} = (2/3) * turno \quad (4)$$

Este parâmetro permite-nos ter informação, a cada turno, de quantos soldados são necessários para aplicar a penalização máxima, nesse turno.

3) Retard necessário: Ao subtrair ao parâmetro *retard* máximo o valor do parâmetro *retard* conseguimos estimar se estamos a aplicar o máximo de penalização possível ou não.

4) Número de soldados inimigos que irão surgir nas próximas cinco rondas: Tal como o nome indica, este parâmetro permite-nos calcular o n° de inimigos que iremos enfrentar (irão "nascer") nos cinco turnos a seguir ao atual.

Estes parâmetros acima mencionados permitem-nos verificar se estamos a dominar, através da expressão 5, e permitem-nos também calcular o número de soldados *ranged* necessários a comprar a fim de matar os inimigos que irão surgir, sendo que, este cálculo é feito através do número de soldados inimigos que irão dar *spawn* no turno atual mais quatro, multiplicado por um terço.

$$retard \text{ necessário} < -(2/3) * (turno + 2) \quad (5)$$

IV. AVALIAÇÃO DE DESEMPENHO

A. Análise de resultados

Um projeto padece de possíveis alterações consoante o avanço temporal e algoritmos desenvolvidos. Assim, é necessário gerir o mesmo de acordo com as novas alterações e garantir que este se mantém escalável a qualquer momento.

Desta forma, é importante avaliar o comportamento do programa face o objetivo estabelecido no âmbito deste projeto.

Segue-se então uma exposição e reflexão dos resultados obtidos para ambas as dificuldades presentes no jogo.

Através de uma leitura detalhada do comportamento do jogo e das tabelas que se seguem em I, em II e em III, podemos

verificar que existe uma maior eficiência dos algoritmos desenvolvidos para a dificuldade um. Para esta dificuldade, o algoritmo ganha sempre o jogo.

Em relação à dificuldade zero do jogo, verificamos que por aleatoriedade do próprio exército inimigo, existem casos em que o algoritmo excede as expectativas e ganha o jogo e outros momentos em que perde após poucas rondas. Apresenta-se na tabela I uma situação que o algoritmo desenvolvido ganha o jogo e atinge o máximo *retard* e na tabela II uma situação em que o algoritmo não é eficiente.

TABLE I

TABELA DE RESULTADOS PARA A DIFICULDADE ZERO (PRIMEIRA SITUAÇÃO)

Rondas	Nível da Base	Score	Retard
404	18	2798	20000

TABLE II

TABELA DE RESULTADOS PARA A DIFICULDADE ZERO (SEGUNDA SITUAÇÃO)

Rondas	Nível da Base	Score	Retard
308	12	308	505.8

TABLE III

TABELA DE RESULTADOS PARA A DIFICULDADE UM

Rondas	Nível da Base	Score	Retard
534	18	2733	20000

V. CONCLUSÃO

Todo este projeto contribuiu para um incremento do conhecimento dos elementos do grupo relativamente a inúmeros algoritmos explorados na cadeira de Sistemas Inteligentes, contribuindo igualmente para realçar a importância de uma organização e gestão de um trabalho em equipa.

Considera-se assim, após reflexão presente neste documento, que os objetivos inicialmente estabelecidos foram concluídos com sucesso.