

Trabalho Prático 1

Elton M. Cardoso
Guilherme Baumgratz Figueiroa

Abril 2019

Capítulo 1

Sobre o Trabalho

O objetivo desse trabalho é servir como aplicação prática dos conceitos apresentados em sala de aula por meio da implementação de um jogo no estilo de RPG. Os detalhes sobre o funcionamento do jogo serão apresentados na seção 1.1, requisitos e regras de implementação estão descritos na seção 2, e finalmente as forma de avaliação e de entrega estão descritas na seção 3.



O enunciado do trabalho cobre os aspectos mais importantes do jogo, no entanto ele não é detalhado ao ponto de determinar quais classes devem ser implementadas ou como elas devem ser implementadas. Os detalhes de implementação e a solução dos problemas que surgem durante a etapa da de modelagem são parte do trabalho e responsabilidade do aluno.

Esse trabalho deve:

- Ser implementado em linguagem Java.
- Ser desenvolvido de forma individual.
- Estar bem documentado e identificado.



Atenção

Alunos avançados estão proibidos de empregar recursos tais como herança, interfaces, classes abstratas ou arquivos para resolver os problemas encontrados nesse trabalho. O uso de interfaces gráficas não é necessário e nem será apreciado nesse momento. No entanto uma segunda versão desse trabalho fará uso desses recursos, sendo portanto prudente modelar seu código de modo que ela possa ser reaproveitado.

1.1 O Jogo



O jogo consiste de um mapa, personagens, e arenas de batalha e é jogado em turnos. Os personagens podem ser jogadores, NPCs ou monstros. Os jogadores são móveis (podem se mover pelo mapa) e controlados por um usuário. Os NPCs também são móveis mas controlados pelo próprio jogo (ou por uma inteligência artificial se você preferir !) e são opcionais nesse trabalho. Os monstros por sua vez são entidades fixas (imóveis no mapa) em certas células do mapa e também controladas pelo jogo.

Ao final de cada turno o status e posição de cada personagem é atualizado com base nas ações que o personagem realizou e nas ações realizadas sobre aquele personagem. A seguir descreve-se em mais detalhes cada um dos elementos do jogo.

1.2 O Mapa



O mapa do jogo é formado um conjunto de células, cada uma das quais pode conter até quatro células vizinhas. A relação de vizinhança não é, necessariamente, reflexiva. Aos Jogadores e NPCs é sempre permitido se mover para qualquer uma das células vizinhas à célula a qual o personagem está posicionado. Na Figura 1.1, a seguir, a relação de vizinhança é representada pelas setas e o personagem na célula 2 pode se mover apenas para as células 6 ou 1.

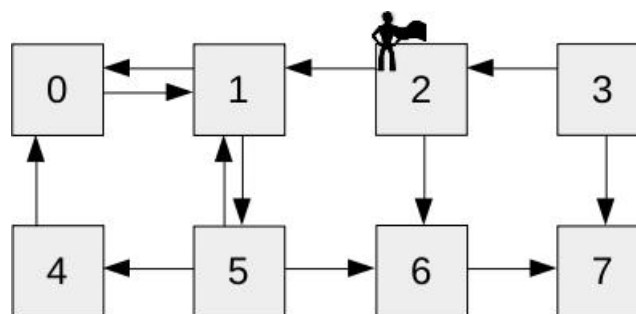


Figura 1.1: Exemplo de um possível mapa

Cada célula do mapa pode conter um número arbitrário de monstros que permanecem fixos na célula,

e um valor arbitrário de probabilidade de batalha (β) entre 0 e 100. Ao entrar em uma célula um personagem pode ou não iniciar uma batalha, dependendo apenas do acaso e de β .

Quando uma batalha é iniciada em uma célula todos os personagens naquela célula, naquele momento, participarão da batalha. Na figura 1.2 vemos que se o personagem na célula 5 se mover para a célula 6, ele terá 80% de chance de enfrentar todos os 3 monstros daquela célula. Note ainda que a célula 7 contém um monstro e tem 100% de chance de iniciar uma batalha.

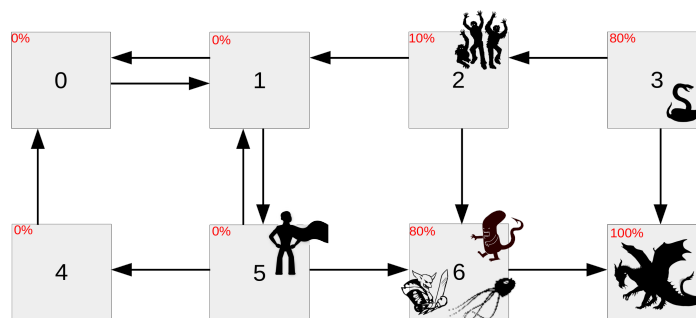


Figura 1.2: Mapa, monstros e probabilidades de batalha

Uma batalha também pode ser iniciada se um jogador (usuário ou NPC) entrar em célula já ocupada por **outro jogador**. Independentemente da forma como aconteça, um vez iniciada uma batalha a célula no qual ela ocorre é trancada, impedindo o acesso de qualquer outro jogador aquela célula até que a batalha termine.

✓ Dica

Para determinar se um combate deve ou não ocorrer, primeiro verifique se há um personagem na célula, se não houver faça com que, após seu personagem entrar na célula, um valor aleatório entre 0 e 100 seja sorteado e compare-o com o valor da célula.

Existem dois tipos de células especiais, uma célula de armadilha tem uma probabilidade de lançar uma maldição a um personagem (um personagem amaldiçoado não pode usar nenhuma arma até que a maldição seja removida). Um segundo tipo de célula restaura qualquer personagem a seu estado pleno.

Finalmente ressalta-se que o mapa não é pré-definido, isto é, não há um mapa definitivo para o jogo e, além disso, deve ser possível ao usuário construir seus próprios mapas, possibilitando assim a flexibilidade da construção de histórias diferentes e com diferentes níveis de dificuldade. Não há uma especificação de como o usuário deve especificar com o mapa e, portanto, forçar o usuário a criar uma classe mapa é aceitável nesse estágio do trabalho.

1.3 Personagens



Para fins de uniformidade monstros, jogadores e NPCs serão todos considerados personagens do jogo. Quando for necessário distinguir as características de um personagem específico empregar-se-á o nome da classes específica do referido personagem. Para todos os efeitos quando nos referirmos a jogadores, estaremos nos referindo tanto a jogadores quanto a NPCs.

Todos os personagens possuem:

- ID: Um valor inteiro que **identifica de forma única** cada personagem no jogo.
- HP (*health points*): A quantidade de "vida" ou "saúde" de um dado personagem. Cada ataque que o personagem sofre reduz o valor de HP. Se o HP de qualquer personagem for zerado ele será considerado morto.
- MP (*magic points*): A quantidade de "magia" ou disponível a um dado personagem. Ao utilizar qualquer habilidade mágica, esse valor é reduzido e se zerado o respectivo personagem não poderá usar nenhuma habilidade mágica.
- Exp (*Pontos de experiência*): Esse valor influencia a destreza, força e esquiva do personagem.
- Atk (Força do ataque): Influência a quantidade de dano causado por um ataque.
- Esq (Esquiva): Influência o quão bem o personagem se desvia ou anula ataques direcionados a ele. O valor significa a probabilidade do personagem se esquivar do ataque (tanto físico quanto mágico)
- RFogo (Resistência a fogo): Resistência à ataques mágicos a base de fogo. Quanto maior o valor, menor causado por fogo.
- RGelo (Resistência a gelo): Resistência à ataques mágicos a base de gelo. Quanto maior o valor Menor o danos causado por gelo.
- RFísico (Resistência a golpes físicos): Resistência a golpes físicos. Quanto maior o valor menor o dano causado por golpes físicos.
- Status: Informa a condição do seu personagem (Morto, Amaldiçoado ou Normal). Um personagem morto não direito a realizar ações em qualquer turno. Um personagem amaldiçoado não pode realizar ações de combate até que seu status volte ao normal.

Para fins de identificação jogadores controlados pelo usuário e NPCs também deve conter um nome (String), além do seu ID.

A experiência de um personagem (Exp) é um valor inteiro que influencia diretamente os atributos de destreza, esquiva e força de ataque. Cada tipo de personagem pode ser influenciado de forma diferente pela experiência adquirida e naturalmente, com o tempo o incremento da experiência não mais afeta as demais características do personagem. Essas comportamento é capturado pela fórmula:

Definição 1.3.1: Evolução do personagem

$$a(x) = c(1 - e^{-\frac{x}{t}})$$

Onde:

- a : É um dos atributos do personagem.
- c : Limite máximo do referido atributo para o personagem.
- e : É a constante Neperiana.
- x : experiência de personagem.
- t (Constante de evolução): Quanto menor o valor de t , mais rápido o valor da fórmula converge para seu limite.

Para fins de clareza, as constantes serão subscritas pelo nome do atributo a que se referem. Exemplo c_{HP} é a constante c do atributo HP.

Ajustando os valores de c e t pode-se criar diferentes curvas de evolução para diferentes tipos de habilidades e assim pode-se ter por, exemplo, um guerreiro pode ter mais conseguir ficar forte mais rápido que um ladrão, mas um ladrão poderá ter uma esquiva melhor em menos tempo.

Os valores de HP e MP são definidos em termos de $a(x) + i$, em que i é um valor base de HP/MP que depende do personagem.

O dano gerado por um ataque depende exclusivamente do atributo de ataque (Atk) do personagem, do valor de MP ou HP do personagem e do dano d em si é valor aleatório limitado obedecendo as seguintes restrições:

Definição 1.3.2: Definição do dano gerado por um personagem

$$d(x) = \begin{cases} [\frac{a}{100}x, \max(0, \frac{a}{100}x - |px|)] * fc, & \text{se o ataque for físico} \\ [\frac{a}{100}x, \max(0, \frac{a}{100}x - |px|)] * fm, & \text{se o ataque for mágico} \end{cases}$$

Onde:

- a : Valor do atributo de ataque do personagem.
- x : Dano causado pelo método empregado (ataque físico, mágica etc..) dado pela Tabela 1.2
- p : Dado por: $0, 17(\frac{100-a}{100})$
- $fc = \frac{hp+0,1c_{HP}}{\min(500, c_{HP})}$
- $fm = \frac{mp+0,1c_{MP}}{\min(500, c_{MP})}$

Note que quanto maior o valor de ataque, menor a variabilidade do dano infligido pelo personagem. Por fim todos os personagens sempre começam com 10 pontos de experiência, exceto pelos monstros que podem ter iniciar o jogo com valores arbitrários de experiência em favor de tornar o jogo interessante.

A Figura 1.3 apresenta a evolução dos atributos de um guerreiro, por exemplo, de acordo com a quantidade de experiência.

O dano infligido a um personagem depende do dano aplicado ao personagem e da defesa do personagem em relação ao tipo de dano.

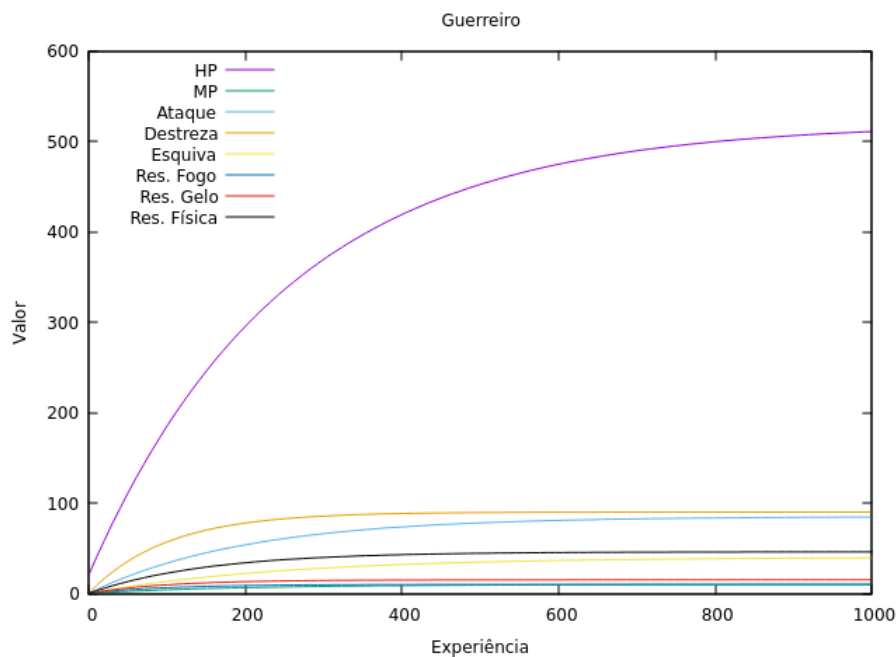


Figura 1.3: Evolução das características de um personagem em função da experiência

| Tipo | HP | | | MP | | | Atk | | Esq. | | R.Fogo | | R.Gelo | | R.Física | |
|-----------|-----|-------|-----|-----|------|-----|-----|-----|------|-----|--------|-----|--------|-----|----------|-----|
| | i | c | t | i | c | t | c | t | c | t | c | t | c | t | c | t |
| Guerreiro | 20 | 500 | 250 | 0 | 10 | 200 | 85 | 200 | 40 | 250 | 10 | 100 | 15 | 100 | 46 | 150 |
| Mago | 15 | 300 | 200 | 90 | 700 | 200 | 40 | 200 | 20 | 200 | 50 | 110 | 58 | 110 | 10 | 180 |
| Ninja | 21 | 350 | 150 | 5 | 20 | 110 | 73 | 170 | 65 | 200 | 18 | 100 | 19 | 105 | 22 | 120 |
| Monstros | | | | | | | | | | | | | | | | |
| Goblin | 10 | 900 | 350 | 0 | 10 | 100 | 105 | 400 | 10 | 100 | 5 | 100 | 5 | 100 | 5 | 100 |
| E.Fogo | 90 | 3200 | 300 | 100 | 3000 | 300 | 60 | 400 | 22 | 110 | 500 | 100 | 10 | 250 | 30 | 170 |
| Dragão | 200 | 10000 | 250 | 500 | 8000 | 250 | 700 | 400 | 10 | 110 | 500 | 300 | 500 | 300 | 800 | 290 |

Tabela 1.1: Relação de Personagens do jogo

Definição 1.3.3: Definição do dano infligido a um personagem

$$di(x) = x * \frac{100}{100 + r}$$

Onde:

- x : Valor do dano gerado pelo atacante.
- r : Valor da resistência do personagem ao tipo de ataque empregado.

A Tabela 1.1, a seguir, apresenta os valores dos atributos de alguns personagens que já devem estar contidos no jogo.

1.4 Arena de Batalha



Quando uma batalha é iniciada em uma célula esta se torna uma arena. Nenhum personagem pode entrar ou sair da arena até que a batalha termine. Qualquer batalha iniciada em um célula envolverá todos os personagens daquela célula.

Para todos os efeitos todos os monstros em uma célula jogam como se estivessem em um mesmo time, ou seja os monstros não atacam uns aos outros. Todos os demais jogadores são considerados inimigos entre si. A batalha ocorre em rodadas, cada uma das quais é dividida em turnos. A cada turno um personagem vivo, jogador ou monstro, em direito a uma ação. A batalha termina somente quando restar um único jogador/NPC ou se restar somente monstros na arena. Ao jogador vencedor será computado 25% de experiência dos inimigos por ele derrotado.

Durante a batalha todos os demais jogadores não envolvidos na batalha têm direito aos seus respectivos turnos, podendo inclusive se mover pelo mapa.

1.5 Ações dos Personagens



Quando não estão em batalha os personagens podem apenas se mover pelo mapa e suas ações são limitadas a escolher a célula para onde irão se mover.

Quando estão em batalha, todos os personagens podem usar as seguintes ações, com seus respectivos valores de dano:

| Ação | Tipo | Dano | MP Requerido |
|-----------------|---------------|------|--------------|
| Fogo | Mágico (Fogo) | 200 | 6 |
| Gelo | Mágico (Gelo) | 350 | 10 |
| Ataque simples | Físico | 210 | 0 |
| Impulso Cruzado | Físico | 362 | 0 |

Tabela 1.2: Valore dos danos do ataques

1.6 O Objetivo do Jogo



O objetivo do jogo é deixado a cargo do aluno. Alguns exemplos de objetivos possíveis são:

- Derrotar todos os monstros do mapa.
- Chegar a uma célula específica do mapa.
- Derrotar um monstro específico.
- Não permitir que um personagem específico seja morto.

Tenha em mente, no entanto, que a segunda versão do TP demandará do jogo, a capacidade de determinar um objetivo arbitrário.

Capítulo 2

Sobre a Implementação

O jogo deve necessariamente ser implementado em Java, usando classes e os conceitos visto em sala até o momento. Estruturas de dados comuns e classes utilitárias da linguagem Java podem ser usadas, desde que não se faça necessário o uso de herança. Os objetos devem estar bem definidos e bem encapsulados. Por exemplo: O mapa deve servir apenas para determinar a localização do personagem e se o mesmo iniciou ou não uma batalha.

A reusabilidade é um critério crucial assim como a flexibilidade da implementação. Tente fazer com que modificar o código seja fácil. Considere que você está criando um jogo para um usuário que também é programador da linguagem Java. Dessa forma é possível permitir que novos mapas e NPCs assim como novos objetivos possam ser adicionados ao jogo, ainda que faça necessário modificar o código fonte do jogo.

O código deve estar documentado em Java DOC, e todas as classes devem conter identificação do aluno. Além disso é altamente recomendado que o aluno observe a indentação e organização do código.

2.1 Como Resolver o Trabalho



Comece entendendo o jogo. Jogue uma partida com um colega e veja como o jogo funciona. Em seguida tente identificar as classes e as relações entre elas.

As seguintes questões tem o objetivo de ajudá-lo a antever alguns potenciais problemas a serem resolvidos:

1. Se considerar criar uma classe para mapa, com certeza um mapa é um conjunto de células. Mas alguma células podem virar arenas. Como você pretende modelar isso ?

2. É fácil modelar monstros, NPCs e jogadores como classes separadas, mas considere o seguinte problema: Como armazenar vários personagens em uma mesma estrutura de dados ? Esse é um requisito fundamental para se criar uma classe arena reusável.
3. Quão trabalho seria adicionar um novo personagem ao jogo ?
4. Em uma arena existem vários jogadores que devem executar ações. Em qual ordem eles devem executar essas ações? Como modelar a interação entre dois personagens ? Essa interação poderia ser uma classe em si ?

✓ Dica

Você pode usar enumeração para permitir que uma única classe possa **representar mais de um objeto**. Por exemplo um personagem pode ser um jogador ou um monstro então você criar uma única classe **Personagens** com um campo de tipo **enum** que funciona como uma marca para o tipo do personagem. ^a

^aMais adiante será mostrado que isso é uma solução horrível, dado os recursos que a linguagem oferece !

Capítulo 3

Entrega e Critérios de Avaliação

Todo o **código fonte** produzido deve ser entregue em um único arquivo comprimido (.zip) no moodle até as 23h e 50m do dia 09/06/2019. Os trabalhos serão avaliados por meio de entrevista com a apresentação do código. É de responsabilidade do aluno agendar a entrevista entre as datas de 10/06/2019 e 17/06/2019.

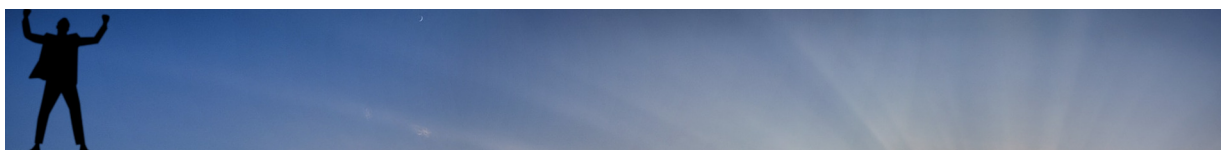
⚠ Atenção

Esteja ciente que os códigos fontes serão verificados automaticamente por plágio e cópia. Caso constatado cópia ambos os trabalhos serão zerados. Atrasos serão penalizados por dia pela fórmula $p(x) = 2^{x-1}$ onde $p(x)$ é o número de pontos subtraídos do valor total do trabalho.

| Critério | Descrição | Valor | Modo |
|--------------------|--|-------|----------|
| Assiduidade | Código bem documentado, indentado, emprego de nomes apropriados para classes | 3 | Binário |
| Organização | Distribuição das funcionalidades entre as classes, tamanho das classes, Organização dos arquivos de código | 3 | Binário |
| Modelagem | Classes identificadas, relações entre as classes, Solução do problema de interação entre personagens e desse com o mapa. | 10 | Contínuo |
| Flexibilidade | Quão facilmente a implementação pode ser modificada. | 5 | Contínuo |
| Compilação | Se o trabalho compila | 3 | Binário |
| Testes de execução | Testes do funcionamento do trabalho | 6 | Binário |

Tabela 3.1: Distribuição dos pontos

3.1 Pontos Extras



As seguintes extensões receberão pontuação extra:

- Criar novos monstros (2pt).
- Criar novos tipos de jogadores (2pt)
- Criar NPCs (3pt).
- Tornar o objetivo do jogo facilmente customizável (3pt).
- Criar novos tipos de células especiais (2pt).