

# **Projeto Monster Hunter Tactics**

**Especificação de Requisitos de Software**

**Versão 3.0**

**Mauro José Vieira Junior (15200618)**

**Florianópolis, 1 de dezembro de 2017**

## Sumário

Lista de Ilustrações .....	4
Versões .....	5
1. Introdução.....	6
1.1. Finalidade.....	6
1.2. Limites do Projeto.....	6
1.3. Materiais de Referência .....	6
1.4. Organização do Documento.....	7
2. Visão Geral .....	8
2.1. Objetivos Gerais.....	8
2.2. Usuários .....	8
2.2.1. Jogador (Ator).....	8
2.3. Restrições .....	8
2.4. Regras do Jogo .....	8
2.5. Arquitetura do Programa .....	9
2.6. Premissas de Desenvolvimento .....	10
3. Requisitos Específicos.....	10
3.1. Requisitos Funcionais .....	10
3.1.1. Implementar estratégia .....	10
3.1.2. Iniciar Partida.....	10
3.1.3. Procedimento de Lance .....	10
3.1.4. Comunicação de resultado de Lance .....	11
3.2. Requisitos Não Funcionais.....	11
3.2.1. Especificação do projeto em UML 2.0 .....	11
3.2.2. Interface Gráfica em 2D.....	11
3.3. Requisitos Descartados .....	11

3.3.1.	Persistência de Dados (Local) .....	11
3.3.2.	Iniciar ou carregar o jogo em um save file específico .....	11
3.3.3.	Iniciar uma missão .....	12
3.3.4.	Requisitos de sucesso ou falha da missão .....	12
3.3.5.	Interagir com o ambiente .....	12
3.3.6.	Menus .....	12
3.3.7.	Controle de Navegação e Interação (Menus) .....	12
3.3.8.	Controle de Navegação e Interação (Mapa) .....	12
3.3.9.	Mapa de Tiles em 2D .....	12
3.3.10.	Gestão de Inventário .....	12
3.3.11.	Inteligência Artificial de Inimigo Comum .....	13
3.3.12.	Inteligência Artificial de Inimigo Especial (Caça) .....	13
Anexos	.....	14
i.	Esboço da interface .....	14
a.	Tela inicial .....	14
b.	As partidas .....	14
c.	Turno .....	15
d.	Movimentação .....	15
e.	Ataque .....	16
f.	Dano .....	17
g.	Modo Multiplayer e inimigo principal .....	18

## **Lista de Ilustrações**

Figura i.1 – Tela inicial. ....	14
Figura i.2 – Estado inicial de uma batalha.....	15
Figura i.3 – As opções de ação em um turno.....	15
Figura i.4 – Opção Mover selecionada.....	16
Figura i.5 – Opção Agir selecionada. ....	17
Figura i.6 – Um ataque na iminência de ser realizado. ....	17
Figura i.7 – Exemplo de uma batalha multiplayer.....	18

### Versões

<b>Versão</b>	<b>Autor (es)</b>	<b>Data</b>	<b>Ação</b>
<b>1.0</b>	Filipe Oliveira de Borba Mauro José Vieira Junior	<b>31/3/2017</b>	Estabelecimento dos requisitos iniciais
<b>2.0</b>	Filipe Oliveira de Borba Mauro José Vieira Junior	<b>12/4/2017</b>	<ul style="list-style-type: none"> <li>• Atualizada 1.4 Materiais de Referência</li> <li>• Atualizada 2.3 Restrições</li> <li>• Atualizada 3.1 Requisitos Funcionais</li> <li>• Atualizada 2.1 Objetivos Gerais</li> <li>• Removida 2.2 Traços do Projeto</li> <li>• Adicionada Lista de Ilustrações</li> <li>• Adicionada 2.4 Regras do Jogo</li> <li>• Adicionada 2.5 Arquitetura do Programa</li> <li>• Adicionada seção 2.6 Premissas de Desenvolvimento</li> </ul>
<b>2.0.1</b>	Filipe Oliveira de Borba	<b>16/4/2017</b>	<ul style="list-style-type: none"> <li>• Atualizada 2.5 Arquitetura do Programa</li> <li>• Atualizada 2.6 Premissas de Desenvolvimento</li> <li>• Atualizada 3.1 Requisitos Funcionais</li> </ul>
<b>2.0.2</b>	Filipe Oliveira de Borba Mauro José Vieira Junior	<b>19/4/2017</b>	<ul style="list-style-type: none"> <li>• Atualizada 1.4 Materiais de Referência</li> <li>• Atualizada 3.3 Requisitos Descartados</li> <li>• Adicionado Anexos</li> </ul>
<b>3.0</b>	Mauro José Vieira Junior	<b>1/09/2017</b>	<ul style="list-style-type: none"> <li>• Atualizados Anexos</li> </ul>

## 1. Introdução

### 1.1. Finalidade

Este projeto tem como fim a criação de um jogo digital distribuído, cujo tema principal escolhido é a caça de monstros, com capacidade de ser jogado em rede por até 4 pessoas, via o uso do NetGamesNRT.

### 1.2. Limites do Projeto

- Não se tem como objetivo fazer deste projeto um projeto comercial;
- Não haverá mais que 2 jogadores por Missão;
- Não será utilizada música ou efeitos sonoros de autoria própria;
- Não haverá mecânicas de jogo extremamente polidas, apenas o suficiente para funcionar;

### 1.3. Materiais de Referência

Serão observadas as características, mecânica e jogabilidade do jogo Monster Hunter Freedom Unite (MHFU) para serem adaptadas em RPG Tactics e 2D, para isso utiliza-se como referência o jogo Final Fantasy Tactics A2 (FFTA2), como guia artístico serão utilizados os livros Monster Hunter Illustrations Vol. 1 e Monster Hunter Illustrations 2 para a criação de sprites, animação e escolha da paleta de cores e linhas gerais da interface visual com o usuário.

Os anexos i e ii mostram quais características dos jogos se tomou como base para a criação dos sistemas propostos e quais detalhes foram percebidos, mas descartados.

CAPCOM. **Monster Hunter Freedom Unite**. CAPCOM, 2009. Disponível em <<https://drive.google.com/open?id=0B8m2AyX9U8LCZIFwdzVhbXJHaW8>>.

É necessário um emulador para poder acessar o jogo via computador (recomenda-se o emulador PPSSPP, disponível em <<http://www.ppsspp.org/downloads.html>>) ou um console Playstation Portable (PSP) desbloqueado.

CAPCOM. **Monster Hunter Illustrations Vol. 1**. Udon Entertainment, 2011. Disponível em <<https://drive.google.com/open?id=0B8m2AyX9U8LCQIhIOW84c2Y1enc>>.

CAPCOM. **Monster Hunter Illustrations 2**. Udon Entertainment, 2014. Disponível em <<https://drive.google.com/open?id=0B8m2AyX9U8LCWk5nU2tIVUJSZIk>>. CAPCOM. **Final Fantasy Tactics A2: Grimoire of the Rift**. Square Enix, 2007. Disponível em <<https://drive.google.com/open?id=0B8m2AyX9U8LCZIFwdzVhbXJHaW8>>.

É necessário um emulador para poder acessar o jogo via computador (recomenda-se o emulador no\$gba, disponível em < [http://www.nogba.com/no\\$gba-download.htm](http://www.nogba.com/no$gba-download.htm)>) ou um console Nintendo DS (NDS) desbloqueado.

#### 1.4. Organização do Documento

Este documento está organizado em 3 seções principais, respectivamente, a Visão Geral, Requisitos Específicos e Funcionalidades do Sistema.

Com a Visão Geral se mostra as linhas principais e os possíveis rumos que este software pode seguir conforme as considerações realizadas.

Nos Requisitos Específicos são mostrados os requisitos funcionais e não funcionais do software, assim como os que foram descartados conforme as novas versões surgiram.

## 2. Visão Geral

### 2.1. Objetivos Gerais

O objetivo principal é a criação de um jogo do gênero RPG Tactics, guiado por um sistema de requisições de caça (missões), capaz de suportar 2 jogadores conectados em rede via o framework NetGamesNRT.

### 2.2. Usuários

#### 2.2.1. Jogador (Ator)

- Definição: usuário que controla avatares no jogo e interage via eles.
- Função: servir de interface para um usuário que queira jogar.

### 2.3. Restrições

- 2 jogadores conectados em rede

### 2.4. Regras do Jogo

Considerando apenas o contexto da jogabilidade das missões propõe-se possui as seguintes regras:

- Há um mapa (tabuleiro) com uma quantia de tiles (quadrados que representam posição) definida em sua criação. Tal quantia pode variar conforme os mapas, porém se estipulou um tabuleiro de 8x8 para fins de teste.
- Os tiles possuem atributos como tipo de terreno, coordenadas, altura e ocupante, futuramente haverá sprite para refinar a interface gráfica.
- Há 3 comandos principais para uma unidade na missão: mover, agir e esperar. Não há uma ordem específica para cada comando, mas a unidade pode ser mexer apenas uma vez, agir apenas uma vez e caso realize as duas ações em um mesmo turno é forçada a esperar, porém nada a impede de realizar “agir->esperar” ou “mover->esperar” em seu turno. Movimentações a princípio podem ser desfeitas, ações realizadas, jamais.
- É mandatório um objetivo na missão que possa ser claramente realizado, como por exemplo, mas não limitado a: “mate todos os inimigos”, “mate o líder dos inimigos”, “obtenha 5 itens”. Tais objetivos servem como condição de vitória para o jogador na determinada missão. Em missões cooperativas



em rede, contanto que um dos jogadores participantes alcance o objetivo da missão, todos terão sucesso na missão.

- Como condições de derrota para o jogador se tem: a falha no cumprimento do objetivo da missão, se render, ou a morte de todas as suas unidades. Caso alguma condição de derrota seja cumprida, o jogador recebe uma tela de game-over, exceto em missões cooperativas em rede. Nas missões cooperativas em rede se sobrepõe a condição “a morte de todas as unidades do jogador” por “a morte de todas as unidades de todos os jogadores”, o ato de “se render” não é permitido neste modo.
- De maneira extremamente básica, uma unidade possui atributos como HP, ataque, defesa, movimentação e alcance. HP define a quantia de pontos de vida da unidade, ataque quanto dano ela causa, defesa quanto dano ela reduz, movimentação quantos tiles ela pode se locomover durante um turno e alcance quantos tiles de distância ela pode atacar.
- Unidades morrem quando seu HP chega em 0 ou por outras condições impostas por efeitos de habilidades.
- Unidades podem possuir condições negativas (envenenamento, paralisia, etc.), equipamentos e habilidades, porém a implementação destes não é prioritária.
- Unidades podem possuir resistências ou fraquezas quanto a elementos e condições negativas.
- Para cada missão e missões cooperativas em rede há um número máximo de unidades permitidas por jogador, o mínimo é sempre 1. Por convenção adota-se 4 unidades como limite.
- Para cada missão e missões cooperativas em rede há uma área inicial de tiles onde podem ser colocados as unidades para a batalha. Por convenção esta área é de 2x4.

## 2.5. Arquitetura do Programa

A arquitetura do software deve ser um programa distribuído cliente-servidor, utilizando o paradigma de orientação a objetos.

## 2.6. Premissas de Desenvolvimento

- O software deve ser implementado em Java devido a pré-requisitos impostos tanto pelo NetGamesNRT, quanto por seus autores.
- O software deve ser capaz de ser executado em qualquer computador que possua ou possa executar o Java Runtime Environment 1.8.0\_121 ou posteriores.
- O software deve obrigatoriamente ser executado de forma distribuída, utilizando para isso o NetGamesNRT.
- Sua interface gráfica deve ser feita inteiramente em 2D.

## 3. Requisitos Específicos

### 3.1. Requisitos Funcionais

#### 3.1.1. Implementar estratégia

O programa deve permitir disputa de partidas entre os jogadores e o programa, sem deixar que personagens controlados pelo programa realizem movimentos inválidos. Os movimentos realizados pelos personagens controlados pelo programa serão definidos por uma inteligência artificial.

#### 3.1.2. Iniciar Partida

O programa deve apresentar ao usuário, no menu principal, a opção de iniciar uma nova partida. Nesta operação é definido o número de jogadores.

#### 3.1.3. Procedimento de Lance

O programa deve suportar procedimentos de lance ao jogador detentor do turno através de um menu de ações exibido na interface do programa com as seguintes opções:

- Mover: neste tipo de lance o programa deve exibir interface as possíveis posições para as quais o jogador pode se locomover, de acordo com as regras do jogo. Então o jogador pode escolher uma das posições ou cancelar a ação;
- Atacar: neste tipo de lance o programa deve exibir na interface os possíveis personagens os quais podem ser atacados a partir da posição atual do jogador detentor do turno, de acordo com as regras. Então jogador pode escolher um dos personagens para atacar ou cancelar a ação;

- Esperar: neste tipo de lance o jogador apenas encerra o seu turno sem realizar ação alguma - neste caso, é pedida a confirmação do usuário para encerrar o turno.
- Abandonar partida: neste lance o jogador abandona a partida, impossibilitando a continuidade da partida.

Observados os possíveis procedimentos, na realização destes as seguintes restrições devem ser consideradas:

- A cada lance de ataque que cause a redução de pontos de vida de um personagem, o programa deve verificar se estes foram reduzidos a zero e, em caso positivo, deve-se verificar se a condição de vitória/derrota foi estabelecida de acordo com as regras.

#### 3.1.4. Comunicação de resultado de Lance

A cada lance realizado o programa deve informar o resultado deste na interface, caso o lance altere o estado de algum elemento externamente perceptível como pontos de vida, posição, etc.

### 3.2. Requisitos Não Funcionais

#### 3.2.1. Especificação do projeto em UML 2.0

O projeto tem a necessidade de ser descrito na linguagem UML 2.0.

#### 3.2.2. Interface Gráfica em 2D

O projeto tem a necessidade de haver uma interface gráfica para o usuário, deve ser baseada em *Java-swing*.

### 3.3. Requisitos Descartados

#### 3.3.1. Persistência de Dados (Local)

Permite a criação de até 3 *load/save files* em um diretório local para o progresso ser continuado quando for conveniente ao usuário.

#### 3.3.2. Iniciar ou carregar o jogo em um save file específico

O programa deve permitir o usuário iniciar um novo jogo ou carregar um previamente existente em seu menu inicial

### 3.3.3. Iniciar uma missão

O programa deve permitir o usuário escolher uma missão, de uma lista de missões, para ser realizada, contanto os pré-requisitos estejam atendidos. Nesta lista deve ser citado os detalhes da missão, seus requisitos de sucesso ou de falha nela e suas recompensas.

### 3.3.4. Requisitos de sucesso ou falha da missão

O programa deve saber do estado atual da missão escolhida pelo usuário e ser capaz de saber seu atual estado (em andamento, bem-sucedida, malsucedida) e retornar a resposta em acordo a esta informação.

### 3.3.5. Interagir com o ambiente

O programa deve permitir a interação do avatar do usuário, durante uma missão, com o meio da missão, incluindo atacar inimigos e coletar itens.

### 3.3.6. Menus

O programa deve ter menus que mostram as escolhas possíveis para o usuário realizar no momento. Isto é aplicado de maneira genérica, logo haverão menus em praticamente tudo que envolva escolhas para o jogador.

### 3.3.7. Controle de Navegação e Interação (Menus)

O programa deve ter um controlador de fluxo das seleções dos itens dos menus do jogo para garantir a coerência das escolhas do jogador com as disponíveis.

### 3.3.8. Controle de Navegação e Interação (Mapa)

O programa deve ter um controlador da movimentação do avatar do usuário no mapa conforme sua entrada de comandos. As interações são tratadas como ataques e vasculhas, logo são tratadas por comandos diferentes das movimentações.

### 3.3.9. Mapa de Tiles em 2D

O programa deve ter mapas de formas geométricas (quadrados) que delimitam a movimentação do controle de navegação.

### 3.3.10. Gestão de Inventário

O programa deve permitir o descarte ou venda dos itens do inventário conforme a vontade do jogador, mas o uso de itens consumíveis é limitado apenas a missões.

#### 3.3.11. Inteligência Artificial de Inimigo Comum

Regula o comportamento de inimigos delimitados como comuns ou de baixa prioridade. A característica principal desta inteligência artificial é sua extrema simplicidade e uso generoso de repetições simples.

#### 3.3.12. Inteligência Artificial de Inimigo Especial (Caça)

Regula o comportamento de inimigos especiais definidos como as Caças de uma missão. O diferencial desta inteligência artificial é sua imprevisibilidade aos olhos do jogador e adaptação conforme a vida da Caça.

## Anexos

### i. Esboço da interface

#### a. Tela inicial

Na tela inicial do jogo aparecerão as modalidades disponíveis de jogo - O jogo terá a possibilidade de jogar uma única pessoa contra o programa ou então de duas a quatro pessoas, em modo cooperativo, contra o programa. A figura abaixo ilustra a tela de início do jogo. Caso o jogador escolha a opção Singleplayer o jogo inicia imediatamente. Caso a escolha seja Multiplayer, o jogador deverá esperar até que todos os jogadores conectem para iniciar.



Figura i.1 – Tela inicial.

#### b. As partidas

As partidas do jogo ocorrem em um terreno, uma espécie de tabuleiro, medindo 7x7 unidades de medidas. Cada unidade quadrada de medida representa uma posição possível para os personagens. A figura a seguir ilustra o terreno e os personagens ocupando posições deste. O objetivo do jogo é derrotar todos os inimigos, abaixando seus pontos de vida a zero para isso. Os jogadores se revezam em turnos, onde ações podem ser tomadas, como se locomover, atacar um personagem próximo, esperar, etc. Apenas uma dessas ações pode ser executada a cada turno.

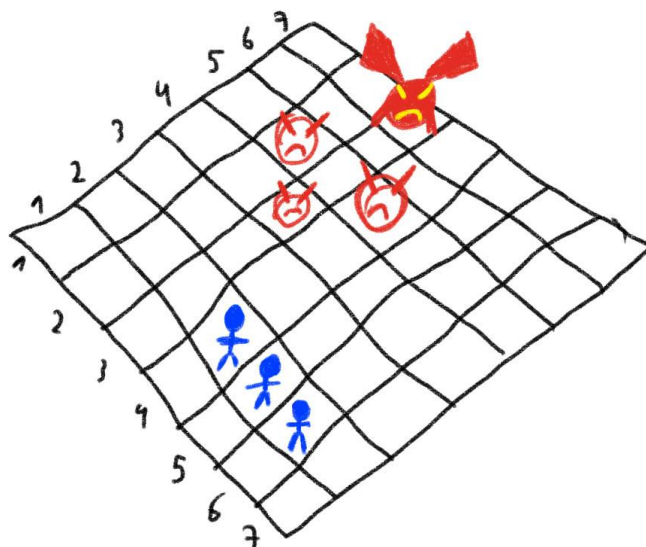


Figura i.2 – Estado inicial de uma batalha.

#### c. Turno

No início do turno de um personagem, é mostrado no canto inferior da tela sua imagem, informações como o seu status atual. Já na parte superior é exibido um menu com opções de ações a serem realizadas durante o turno. Também é dado um destaque na posição do campo de batalha ocupada pelo personagem. A figura a seguir ilustra a fase inicial de um turno destacando os elementos descritos acima.

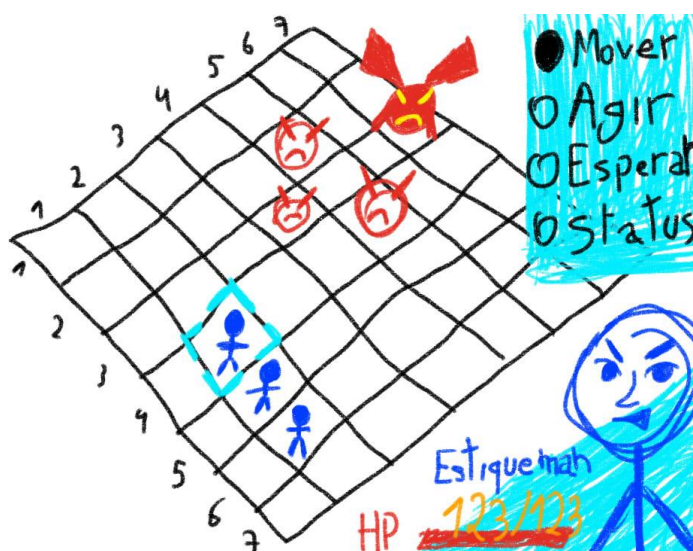


Figura i.3 – As opções de ação em um turno.

#### d. Movimentação

Caso o personagem escolha a opção de realizar um movimento durante o turno, este deve fazer escolhendo a opção adequada na interface. Após isso as opções de destino do movimento são apresentadas. A figura abaixo mostra em azul quais as

opções disponíveis para o personagem se locomover, observado sua habilidade de deslocamento e obstáculos. Note que em amarelo estão indicadas as posições para as quais o personagem normalmente alcançaria, mas, no entanto, não é possível por já estarem ocupadas ou estarem fora do terreno.

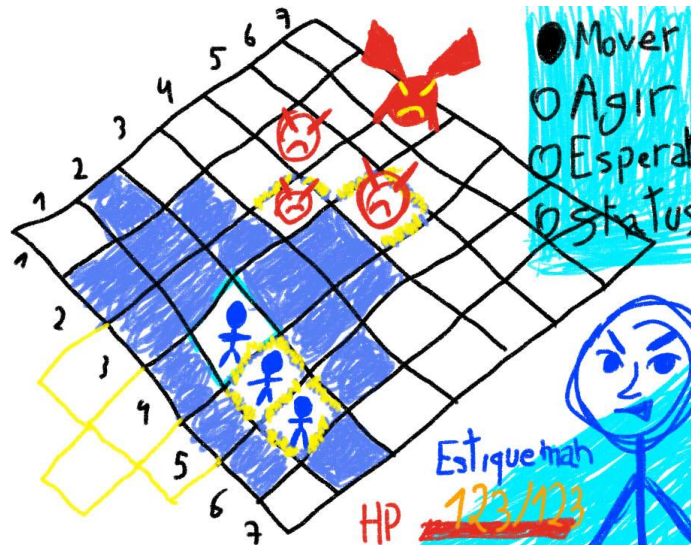


Figura i.4 – Opção Mover selecionada.

#### e. Ataque

Para realizar um ataque, o jogador deve selecionar a opção adequada na interface. O jogo então exibe, em vermelho, as possíveis posições para as quais o personagem pode desferir um ataque, observado sua habilidade de alcance. Selecionada a posição, o personagem tenta desferir um ataque. A figura abaixo ilustra este procedimento. Note que é possível atacar companheiros de equipe, ou seja, há a opção de fogo amigo durante o jogo.



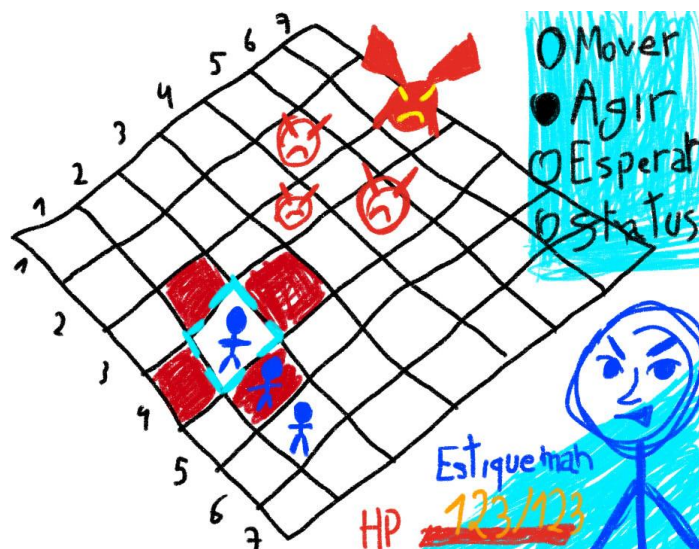


Figura i.5 – Opção Agir selecionada.

f. Dano

Caso o personagem desferir um ataque a outro personagem próximo, o dano resultante do ataque é calculado e exibido na interface, como mostra a próxima figura. O personagem que recebeu o ataque é mostrado no canto oposto do jogador do turno, assim como os pontos deduzidos dos seus pontos de vida. Caso o personagem atinja zero pontos de vida com o ataque recebido, ele é retirado do jogo e não mais poderá realizar ações. Caso este seja o último personagem restante de uma equipe, esta perde a partida e a equipe a qual pertence o jogador do turno é declarada a vencedora.

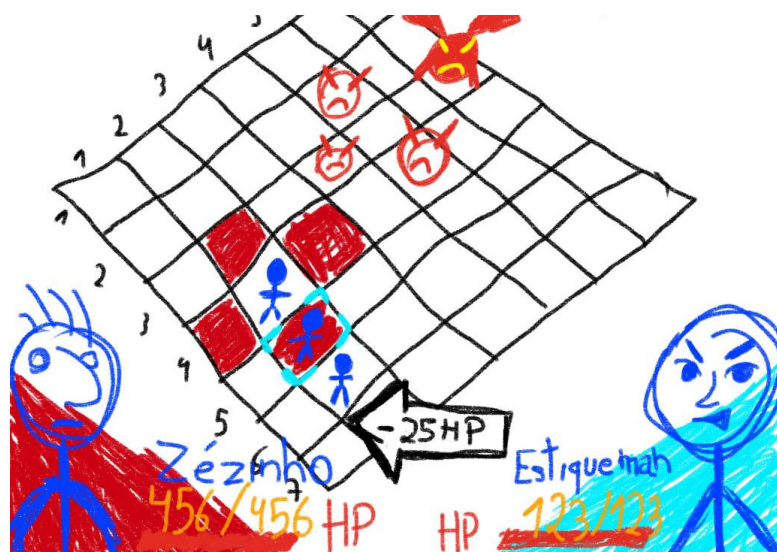


Figura i.6 – Um ataque na iminência de ser realizado.

g. Modo Multiplayer e inimigo principal

A próxima figura ilustra um cenário de jogo em modo Multiplayer com dois jogadores. Também nesta figura é destacado a presença de um inimigo especial, com habilidades maiores, oferecendo um desafio aos jogadores, os quais precisarão de cooperação para derrotar este inimigo mais poderoso.

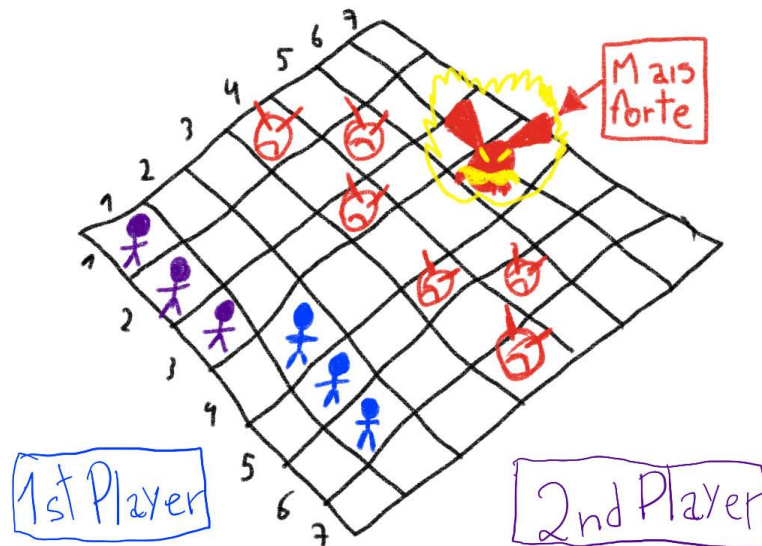


Figura i.7 – Exemplo de uma batalha multiplayer.