

# **Projeto: Monstros dos Dados Masmorra**

## **Especificação de Software**

Gustavo Olegário  
João Victor Fagundes

October 6, 2016

Versão 1.0  
04/09/2016

Versão	Autores	Data	Ação
1.0	Gustavo Olegário João Victor Fagundes	21/09/2016	Estabelecimento de requisitos
1.1	Gustavo Olegário João Victor Fagundes	06/10/2016	Revisão de requisitos

## Conteúdo

1. Introdução
2. Visão Geral
3. Requisito de software
4. Esboço Interface Gráfica
5. Referências

## 1 Introdução

**Objetivo:** desenvolvimento de um software distribuído que fornece uma disputa entre dois jogadores em um tabuleiro de 8 posições por 8 posições onde os usuários travarão um Duelo baseado no jogo Monstros dos Dados Masmorra da série Yu-Gi-Oh.

**Nomenclatura usada neste documento:**

- Ataque: se refere ao atributo do monstro que representa seu poder de força. Sempre que um monstro atacar o outro, será comparado o valor de ataque de ambos. Aquele que tiver o maior valor, ganha.
- Dado de invocação: se refere ao dado que será rolado pelo jogador. Cada face do dado, terá uma estrela de invocação enumeradas de 1 a 6.
- Estrela de invocação: Item necessário para poder realizar a invocação de um monstro em campo.
- Habilidade: Determinados monstros terão habilidades particulares que trarão vantagens ao jogador que possuir um monstro com habilidade. Cada habilidade é única e exclusiva de cada monstro.
- Jogador: entidade que corresponde ao usuário do programa.
- Monstro: Entidade a ser invocada através do uso das estrelas de invocação.

- **Pontos de vida:** Medidor de vida de cada jogador. Ao ter seus pontos de vida zerados, o jogador automaticamente perde.
- **Tabuleiro:** entidade onde os monstros batalham, caminham e também são invocados. Também responsável por processar jogadas e realizar a conexão entre os dois jogadores.

**Regras do Jogo:** Ao iniciar o jogo, assim que for decidido, o jogador irá poder dar início ao seu turno. Após encerrar sua jogada, a vez será passada para o outro jogador. Em seu respectivo turno, os jogadores poderão executar as seguintes ações:

- **Atacar:** o jogador poderá atacar quando tiver um monstro suficientemente próximo de um monstro do oponente ou estiver próximo da área dos pontos de vida do adversário.
- **Desistir:** o jogador adversário será automaticamente considerado vencedor.
- **Fazer invocação:** Se o jogador atender a todas as requisições, ele poderá fazer uma invocação no seu lado do campo e trazer um novo monstro para dentro do tabuleiro.
- **Mover monstro:** movimentar um monstro que já esteja em campo pelos quadrantes.
- **Passar a vez:** preferencialmente, o último ato do usuário durante a sua jogada. Após realizar todas as ações que ele desejava, sua jogada será encerrada e o turno do adversário será iniciado.
- **Rolar o Dado:** rolar os três dados que estão disponíveis.
- **Usar habilidade:** permitido somente para os monstros que possuem habilidades.
- **Verificar habilidade:** verificar se um monstro selecionado possui habilidade e o que ela faz.

O jogo funciona da seguinte forma: Cada jogador fica em lados opostos do tabuleiro. As duas casas do meio do tabuleiro, são onde ficam os 4 pontos de vida de cada jogador, duas pra cada quadrante. O jogador que tiver os 4 pontos de vida zerados, perde. Para atacar os pontos de vida do adversário é necessário que o usuário tenha um monstro suficientemente próximo de uma dessas duas casas para atacar os pontos de vida. O jogo é melhor explicado mais adiante. O jogador irá rolar 3 dados de invocação. Após o resultado, será avaliado o cenário. Para poder invocar um monstro de determinado nível, suponha 3 estrelas de invocação, é necessário que o jogador tenha tirado no mínimo 2 faces com 3 estrelas de invocação. Caso ele não tire 2 ou mais faces iguais, as estrelas atuais serão guardadas, para fazer futuras jogadas, uma vez que qualquer tipo de jogada necessita de um número mínimo de estrelas. É importante salientar que

estrelas de diferentes níveis de invocação não serão misturadas. Caso o usuário tenha atendido os requisitos para poder fazer uma invocação, uma lista com os possíveis monstros para invocar surgirá e ele poderá escolher um deles e poderá colocar o monstro em campo no seu lado do campo, ou seja na primeira linha e em qualquer uma das colunas. Outro ponto importante, só pode haver um monstro por quadrante.

Para o monstro poder se mover, é necessário usar um determinado número de estrelas, isso dependerá do monstro que está sendo movimentado. Quanto mais quadrantes ele andar, mais estrelas serão necessárias.

Para poder atacar, é necessário que o monstro esteja a uma distância mínima de uma casa de seu alvo, independentemente se o alvo for os pontos de vida do adversário ou um monstro simples. Para atacar, também é necessário um determinado número de estrelas de certo nível, isso depende do monstro que está atacando. Em caso de um monstro querer atacar o quadrante dos pontos de vida do adversário e lá residir um monstro inimigo, ele necessariamente vai atacar o monstro. Um monstro só pode atacar uma vez por turno.

Para os monstros com habilidades, o jogador poderá tirar vantagem delas. Por exemplo, um certo monstro pode ter a habilidade de andar duas casas pelo preço de um ao invés de duas. Cada habilidade é exclusiva de cada monstro.

## 2 Visão Geral

**Arquitetura do Programa:** programa orientado a objetos, distribuído.

### **Premissas de desenvolvimento:**

- O programa deve apresentar uma interface gráfica bidimensional, onde haverá uma tela com o tabuleiro, os dados para o jogador rolar, quantas estrelas acumuladas ele possui e as possíveis jogadas que ele pode realizar. A interface também apresentará outros aspectos importantes como usuário da vez e quantos pontos de vida cada usuário possui.
- O programa fará uso da linguagem de programação Java em sua versão SE8, com o paradigma de Orientação a Objetos, do software para cliente-servidor NetGames.

## 3 Requisitos do Software

### 3.1 Requisitos Funcionais

1. **Conectar:** como será um software distribuído, deverá apresentar uma interface gráfica que permita o usuário conectar-se ao servidor NetGames que fará a gestão dos usuários conectados. Para se conectar, o usuário primeiro deve inserir seu nome.

2. **Desconectar:** deve permitir ao usuário que possa se desconectar de um servidor via interface gráfica.
3. **Iniciar partida:** uma vez conectado, o programa deve possibilitar a inicialização de uma nova partida. O programa deve permitir partida contendo dois jogadores.
4. **Dar a vez:** quando um jogador encerra sua jogada, o outro jogador deve poder realizar sua jogada.
5. **Invocar monstro:** se o jogador tiver o número de estrelas mínimas necessárias, ele poderá invocar um monstro do mesmo nível de estrelas tiradas.
6. **Atacar:** quando o usuário tiver o número de estrelas de invocação suficiente, ele poderá atacar com um dos seus monstros em campo. O usuário pode tanto atacar monstros quanto os pontos de vida do adversário, desde que respeite a distância mínima.
7. **Avaliar vencedor:** verifica quais dos jogadores é o ganhador, comparando seus pontos de vida.
8. **Mover monstro:** move o monstro através dos quadrantes.
9. **Rolar dados:** joga os três dados.
10. **Mostrar habilidades:** mostra uma lista com todos os monstros do usuário que estão em campo e mostra a habilidade de cada um.
11. **Mostrar pontos de vida:** um painel que mostra constantemente os pontos de vida de ambos os jogadores.
12. **Registrar jogador:** permite o usuário digitar seu nome e registrar um novo jogador.
13. **Usar habilidade:** permite selecionar um dos monstros que tenham habilidade e usar ela. Assim como o ataque, a habilidade do monstro só pode ser usada uma vez por turno.
14. **Desistir:** o jogador adversário será automaticamente considerado vencedor.
15. **Verificar habilidade:** verificar se um monstro selecionado possui habilidade e o que ela faz.

### 3.2 Requisitos não funcionais

1. **Especificação de projeto:** além do código Java, deve ser produzida especificação de projeto baseada em diagrams UML versão 2.

2. **Identificação de jogadores:** os jogadores devem ser identificados por um nome inserido ao iniciar uma nova partida.
3. **Tecnologia GUI:** Graphics User Interface baseada na biblioteca nativa do Java swing.

## 4 Esboço da Interface Gráfica

A interface gráfica (GUI) do jogo será composta de componentes encontrados no próprio pacote swing do Java e de alguns sprites da série Yu-Gi-Oh retirados da internet.

