## Projeto ColiseumRPG

**Especificação de Requisitos de Software**

Versão 0.1  
02/03/2018

|  |  |  |  |
| --- | --- | --- | --- |
| **Versão** | **Autor** | **Data** | **Ação** |
| 1.0 | Matheus Renaud Pacheco de Campos | 02/03/2018 | Levantamento de Requisitos |
| Luis Henrique Sima Seifert |

Conteúdo

Sumário

[Projeto ColiseumRPG 1](#_Toc510602896)

[1. Introdução 1](#_Toc510602897)

[2. Visão Geral 2](#_Toc510602898)

[3. Requisitos de Software 2](#_Toc510602899)

[3.1 Requisitos Funcionais: 2](#_Toc510602900)

[3.2 Requisitos não Funcionais: 3](#_Toc510602901)

[4. Esboço da Interface gráfica 4](#_Toc510602902)

# Introdução

**Objetivo:** Desenvolvimento de um software para jogo RPG de combate, com suporte a multijogadores.

**O Jogo:** Para o inicio do jogo cada jogador deverá escolher dois personagens entre as classes já predefinidas, cada um desses personagens tem especialidades diferentes já estabelecidas inicialmente (Guerreiro, mago e Caçador).  
 Em jogo os personagens serão posicionados em lados opostos da arena que será uma matriz de 20X15, com os personagens de um mesmo time lado a lado. O jogo se passará com turnos sendo que cada personagem terá:

1. Movimentações iguais ou menores que o atributo de velocidade do personagem em quadrados.
2. uma ação seja ela de uso de especialidade ou de ataque.
3. uma ação menor, apenas para uso de especialidades mais simples.

Em seu turno pode-se realizar todas, duas, uma ou nenhuma dessas ações, em qualquer ordem, a escolha do jogador. Realizar movimentação consome um pouco do atributo de movimentação naquele turno, atacar consome a ação, especialidades tem custo variado, sempre custando a ação menor, porem podendo consumir também mana, a ação inteira, ou algum item que o personagem possua.

Os turnos do jogo se darão de forma alternada entre os personagens e jogadores, exemplo: Começa jogando o jogador 1 com seu personagem 1, depois o jogador 2 com seu personagem 1, volta para o jogador 1 dessa vez com o personagem 2 e termina o ciclo com o jogador 2 personagem 2.

O objetivo final do jogo é matar todos os personagens do seu adversário, quando essa condição for alcançada o jogo encerra e o jogador com personagens vivos ganha.

# Visão Geral

**Arquitetura do Programa:** Programa orientado a objetos em Java, distribuído, no formato cliente-servidor.

**Premissas de Desenvolvimento:**

* O programa deve ser implementado em Java, devendo executar em qualquer plataforma de software que possua a Java Virtual Machine (JVM).
* Possuir Interface Gráfica feita com Java swing.
* Possuir suporte para aplicações distribuídas produzida com NetGamesNRT, não possuirá tempo real.

# Requisitos de Software

## Requisitos Funcionais:

1. **Conectar:** O programa deve apresentar a opção “Conectar” para se conectar a um servidor de jogo NetGamesNRT.
2. **Desconectar:** O programa deve apresentar a opção de menu “Sair” para encerrar a conexão com o servidor de jogo NetGamesNRT.
3. **Iniciar Partida:** O jogo deve apresentar a opção de iniciar partida, que enviara solicitação de inicio a todos os jogadores.
4. **Receber Solicitação de Início:** Leva o jogador para a tela de seleção de personagens, também inicia o tabuleiro, estabelece a ordem entre os jogadores.
5. **Selecionar Personagens:** Permite a cada jogador selecionar dois personagens para jogar, quando confirmado cria os personagens e os preenche do tabuleiro e abre a tela de jogo. Os personagens selecionados por um mesmo jogador não podem ser o mesmo.
6. **Mover:** Em seu turno, cada personagem pode se mover. Para se mover basta apertar o botão mover e escolher um local dentro da distância possível do seu personagem, isso gastará a distância possível de alcançar naquele turno.
7. **Atacar:** Cada personagem pode atacar uma vez em seu turno. Basta pressionar o botão atacar e selecionar um personagem inimigo dentro do alcance de seu ataque básico, isso gastará sua ação daquele turno e causará dano igual ao atributo de dano básico do personagem.
8. **Especialidade:** Cada personagem tem suas habilidades predefinidas, que podem variar entre auto modificação e “Local-Alvo”. Para utilizar uma de auto modificação basta pressionar o botão especialidade para abrir as opções e escolher qual especialidade usar, caso seja “Local-Alvo” depois disso é necessário escolher onde deseja utilizar a habilidade. Todas as especialidades gastam a ação menor do turno atual do personagem, algumas possuem custo extra.
9. **Desistir:** Cada jogador pode, no turno de qualquer um de seus personagens, desistir do jogo. Isso encerrará seu turno e a partida, dando a vitória ao outro jogador.
10. **Encerrar Turno:** Quando não quiser mais realizar ações em sua jogada, o jogador deve pressionar o botão “Encerrar Turno”. Isso enviará a jogada ao outro jogador, passando a rodada para o turno do próximo jogador.
11. **Finalizar Partida:** Quando todos os personagens de um jogador morrer, o jogo deve encerrar.
12. **Enviar Jogada:** Quando um jogador terminar seus movimentos e apertar o botão encerrar turno, ou quando o mesmo desistir do jogo, deve-se enviar a jogada e passar a vez ao outro jogador.
13. **Tratar lance:** Quando receber a jogada do jogador adversário o jogo deve realizar sequencialmente as jogadas do outro jogador, de forma que o atual possa entender o que esta acontecendo.

## Requisitos não Funcionais:

1. **Especificação do Projeto:** Possuir modelagem UML produzida na ferramenta Visual Paradigm.
2. **Tecnologia Interface gráfica:** A interface gráfica deve ser feita usando Java Swing.
3. **Compartilhamento de Interface gráfica:** A interface gráfica deve ser a mesma para todos os usuários.

# Esboço da Interface gráfica

