**Projeto: Lane Attack**

**Especificação de Requisitos de Software**

ERS-LaneAttack-2018.001

Versão 1.0.0.0

26 de agosto de 2018

|  |  |  |  |
| --- | --- | --- | --- |
| Versão | Autor | Data | Ação |
| 1.0.0.0 | Vinícius Cerqueira Nascimento | 26/08/2018 | Elaboração da Especificação de Requisitos |

**Conteúdo:**

1. Introdução
2. Visão Geral
3. Requisitos da Aplicação
4. Requisitos Não-Funcionais
5. Esboço da Interface Gráfica
6. **Introdução**
   1. **Objetivos Gerais**

Desenvolvimento de jogo baseado em computação distribuída que possibilite que dois usuários joguem entre si o jogo Lane Attack.

* 1. **Definições e Abreviaturas**

Classe: Grupo de personagens com uma característica específica em comum.

Atributo: Característica singular do personagem. Variação numérica.

* 1. **Referências**
     1. – Não há referências externas. Jogo de própria autoria.
  2. **Regras do Jogo**

**R1.** O jogo é realizado entre dois e somente dois jogadores, em uma interface gráfica (apresentada no item 4).

**R2.** O jogo é definido ao longo de três rodadas.

**R3.** Antes do início de cada rodada, os jogadores escolhem qual será a classe de seus personagens.

**R4.** Obrigatoriamente quatro personagens da classe escolhida podem ser utilizados por rodada.

**R5.** Os jogadores escolhem os personagens disponíveis da classe escolhida, que são carregados com valores de atributos por padrão. São disponibilizadas mais 15 unidades de atributos, que podem ser distribuídas entre os personagens escolhidos.

**R6.** Os atributos dos personagens podem ser alterados para mais ou para menos, respeitando o limite estipulado.

**R7**. Vence a rodada o jogador que tiver mais personagens vencedores.

**R8.** Ao final de três rodadas, o jogador que venceu mais rodadas vence o jogo.

**R9.** Os jogadores não podem ver os personagens do jogador oposto, ao menos os atributos de seus personagens. A visualização será permitida apenas no momento da ocorrência da partida.

1. **Visão Geral**

**2.1 Arquitetura da aplicação**

Programa orientado a objetos, em um sistema distribuído do tipo cliente – servidor.

**2.2 Premissas de desenvolvimento**

* O programa deve obrigatoriamente executar distribuído, através do suporte à aplicações distribuídas NETGAMES NRT.
* O programa deve ser implementado exclusivamente na linguagem Java e executar em qualquer plataforma que disponha da máquina virtual Java instalada.
* A interface do programa deve ser bidimensional.

1. **Requisitos Funcionais**

**RF1 – Conectar-se ao servidor**

A aplicação deve dispor da opção “Conectar”, no menu superior esquerdo. Quando selecionada, é verificado se a conexão já está estabelecida, em caso contrário, a conexão é feita. Em caso de problemas na conexão, é informada mensagem de aviso ao usuário com o motivo.

**RF2 – Desconectar-se do servidor**

A aplicação deve dispor da opção “Desconectar”, no menu superior esquerdo. Quando selecionada, é verificada se a conexão está em andamento e se há partida corrente. Em caso positivo, é solicitada uma mensagem de confirmação da finalização da partida. Em caso negativo, é solicitada a desconexão ao servidor do NETGAMES e informada a mensagem de desconexão ao usuário. Em caso de falha na operação, é informada mensagem de aviso ao usuário com o motivo da falha.

**RF3 - Iniciar Partida**

A aplicação deve dispor da opção “Iniciar Partida”. Ao ser selecionada, é validado se o outro jogador já está conectado. A partida só será iniciada quando ambos os jogadores possuírem conexão estabelecida com o servidor. Ao término da validação, o jogador logado ‘primeiro’ recebe a titulação de ‘Jogador 1’, e o jogador logado em seguida a titulação de ‘Jogador 2’. O sistema carregará a tela de configuração da partida, juntamente com a grid abaixo disponível para seleção de classe, personagens e alteração de atributos.

**RF4 – Enviar Composição**

A tela de configuração da partida deve dispor do botão “Enviar composição”. O botão será habilitado apenas quando o jogador já houver selecionado a classe (caso seja a primeira rodada) e os personagens da rodada atual. Quando disponível e selecionado, o sistema valida se o jogador oponente já finalizou a composição de seus jogadores para a rodada. Em caso negativo, o sistema informa ao jogador que o oponente ainda não finalizou a configuração da composição, e bloqueia as demais funcionalidades da tela, exibindo uma subtela com a descrição ‘Aguarde’, até que o jogador oponente esteja disponível.

Em caso positivo, o sistema carrega a tela de confrontos para ambos os jogadores.

**RF5 – Confrontos entre personagens**

Após o envio dos dados de personagens por ambos os jogadores através do botão ‘Composição’, a tela de Confrontos será aberta. O sistema deve exibir, no lado inferior da tela, os personagens escolhidos pelo jogador atual e sua ‘Base’. De sua base, devem haver quatro linhas, que apontam diretamente para a base do oponente, na parte superior da tela. Os objetos dos personagens se movem até o centro da tela, onde se confrontam. Os personagens com maior nível de atributo vencem o oponente, e caminham até a base adversária. O ‘mover’ dos personagens se dará em sequência.

**RF5 – Resultados da Rodada ou Partida**

Ao final dos confrontos, o sistema calcula qual jogador teve mais personagens vencedores. É exibida a mensagem “Você venceu esta rodada!” ao usuário vencedor, e a mensagem “Você perdeu esta rodada” ao perdedor. Se a rodada atual é a terceira rodada, é calculado o vencedor do jogo, e exibida a mensagem “Você ganhou a partida” ao usuário vencedor, e a mensagem “Você perdeu! Peça a revanche” ao usuário perdedor.

**RF6 – Configuração da Partida Entre Rodadas**

Ao final de cada rodada, o sistema deve calcular os dados dos confrontos e disponibilizar para os jogadores no lado direito da tela de configuração da partida. No centro e no lado esquerdo da tela, o jogador pode selecionar os personagens que irá utilizar para a próxima rodada. O sistema não deve permitir a seleção de personagens já utilizados em rodadas anteriores.

1. **Requisitos Não-Funcionais**

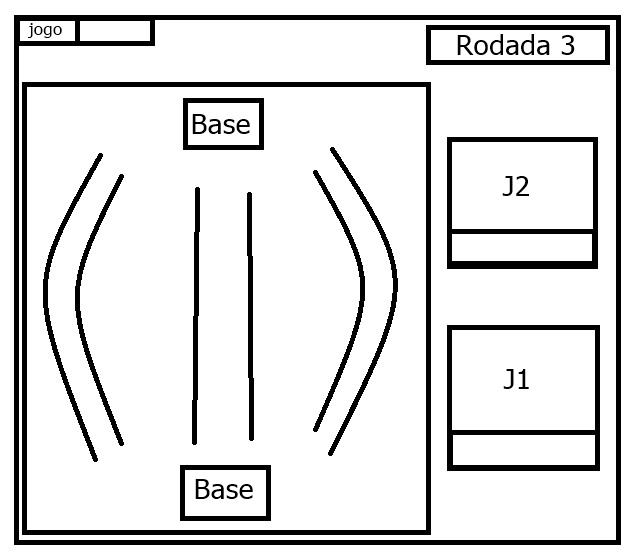
RNF 1 – Codificação em linguagem Java e modelagens de especificação baseadas em UML2.

RNF 2 – A interface gráfica será única, partilhada entre os usuários e baseada em Java Swing.

RNF 3 – Os personagens terão característica gráfica padrão, desenvolvida conforme a concepção do software.

1. **Interface Gráfica**

IG 1 – Esboço Tela Combate.



IG 2 – Esboço Tela Atributos.

