**Especificação de Requisitos**

**JOGO: Palavra Naval**

**Especificação de requisitos de software:**

|  |  |  |
| --- | --- | --- |
| Versão | Autores | Ação |
| 1.3 | Diogenes Vanzela  Jacyara Bosse  Renato Manoel de Souza | Especificação de Requisitos |
| 1.4 | Diogenes Vanzela  Jacyara Bosse  Renato Manoel de Souza | Novos requisitos funcionais. Alteração nas especificações já existentes. |
|  |  |  |

**Conteúdo:**

1. Introdução
2. Visão Geral
3. Regras do Jogo
   1. Objetivo do Jogo
4. Requisitos do Software
5. Referencias

**1 - Introdução**

Palavra Naval tem como objetivo a disputa entre dois jogadores, visando a obtenção da maior quantidade de pontos dentro de uma mesma partida.

**2 - Visão Geral**

**Arquitetura do programa:** Programa Orientado a Objetos desenvolvido, aplicação distribuída utilizando a ferramenta NetGamesNRT[1].

**Premissas de Desenvolvimento:**

* O programa deve apresentar uma interface gráfica.
* O desenvolvimento deve ser feito na linguagem de programação Java.
* Uso da ferramenta NetGamesNRT, para suporte de aplicação distribuída.
* O programa deve apresentar interação entre participantes.

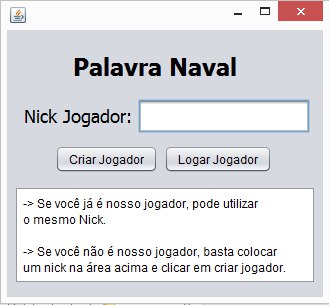
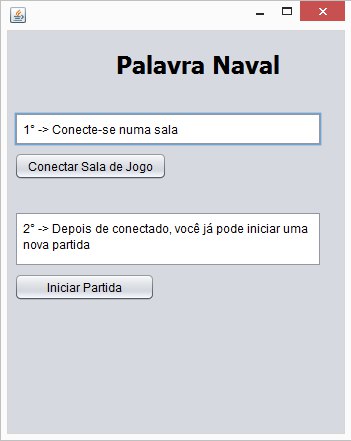
**3 – Regras do Jogo**

* Não é permitido uma jogada sobre uma casa selecionada anteriormente.
* Cada letra possui uma pontuação com base na opinião dos desenvolvedores;
* A formação de uma palavra correta acarreta em ganho de pontos extra, além da pontuação individual de cada letra;
* O jogo informará a quantidade de palavras presente no tabuleiro.
* O jogador a começar a partida é escolhido aleatoriamente.
* O tabuleiro possui casas aleatórias com bombas.
* Jogador que acerta uma bomba, perde pontos e a vez de jogar.
* O tabuleiro possui letras avulsas, para aumentar o nível de dificuldade e induzir o jogador ao erro.
* O jogo permite ao usuário “chutar” uma palavra, caso acertar, o jogador recebe uma quantidade de pontos extra. Caso errar a palavra o jogador perde a vez e também uma quantia de pontos.
* O jogador pode acabar uma partida com pontos negativos.
* Ao final da partida, ganha o jogador que obteve a maior quantidade de pontos.
* O tabuleiro permite somente palavras no sentido horizontal e vertical. Não permitindo no sentido inverso e nem em diagonal.
* O jogo apresenta a opção “desistir”. Caso o jogador opte por desistir, seus pontos serão movidos para o oponente e a partida acaba.

**3.1 – Objetivo do Jogo**

* As partidas têm como objetivo a interação dos dois jogadores com o intuito de uma disputa pela maior quantidade de pontuação de letras e palavras acertadas.
* O tabuleiro do jogo é uma matriz quadrada, onde os elementos (letras e bombas) ficam por trás dos quadrados. Cada jogada o participante escolherá um quadrado para ver o que há atrás.
* Cada quadrado que o jogador escolher terá uma letra pré-definida pelo sistema ao começar a partida. Porém, os quadrados poderão ser bombas, o que terá como consequência perder a jogada e passar para o outro participante.
* Cada letra terá uma pontuação, porém ao chutar uma palavra que se forma com tais letras, e tendo o respectivo acerto da mesma, terá o ganho de pontuação extra. Logo, o jogador que acertar mais palavras durante o jogo terá mais chances de ser o vencedor.
* Ao final de todas as palavras completadas, temos o campeão pelo maior número de pontos.

**3.2 – Esboço de Interface**

** **

****

**4 – Requisitos de Software**

**Requisitos Funcionais:**

**Requisito funcional 1 – Iniciar:** o programa deve apresentar a opção de menu “iniciar” para o início de uma nova partida. O jogador, ao clicar em iniciar, é levado a uma sala, onde aguarda até a entrada de seu oponente. Uma vez que seu oponente é encontrado dá-se início à partida.

**Requisito funcional 2 – Desistir:** o programa deve apresentar a opção de menu “desistir” para finalizar a partida atual, considerando as seguintes restrições:

• Caso não haja jogadores previamente definidos, a operação é equivalente a iniciar;

• Caso a partida estiver em andamento, o programa deve solicitar autorização para sua interrupção – neste caso, apenas com a autorização do usuário a operação pode prosseguir. Alertando o jogador sobre a perca dos pontos obtidos.

**Requisito funcional 3 – Estabelecimento de quem inicia:** após ter dois jogadores definidos, o programa usa um padrão aleatório para definir qual dos jogadores começa jogando;

**Requisito funcional 4 – Procedimento da jogada:** o programa deve suportar o procedimento de jogada dos usuários por meio de click de botão esquerdo de mouse sobre a posição em que deve ser mostrado a letra/bomba, considerando as seguintes restrições:

• O programa deve permitir apenas lances válidos, de acordo com as regras do jogo;

• A cada jogada válida, o programa deve verificar se o jogador que jogou ganhou a partida;

* A cada jogada válida, o programa deve contabilizar os pontos para o jogador.

• A cada jogada válida, caso não haja vencedor, o programa deve verificar se a partida encerrou empatada, com a mesma pontuação para ambos os jogadores.

**Requisito funcional 5 – Comunicação do estado da partida:** a cada tentativa de procedimento de jogada (bem-sucedida ou não) o novo estado da partida deve ser exibido na interface, considerando as seguintes restrições:

• A matriz de posições deve ser atualizada com a presença de uma nova letra/bomba, no caso de procedimento de jogada bem-sucedida;

• Em caso de jogada irregular, deve ser comunicado o tipo de irregularidade: tentativa de colocar letra/bomba em posição ocupada.

**Requisito funcional 6 – Chutar palavra:** é permitido ao usuário, na vez dele, “chutar” uma palavra. Trata-se de uma tentativa de adivinhar as letras ainda escondidas. Tal movimento se acertado, caracteriza em mais ponto para o usuário, caso de errado, o usuário perde pontos e a sua vez de jogo também.

**Requisito funcional 7 – Carregar jogador:** Ao jogador, antes do início da partida é solicitado um nick (apelido), o qual deve ser único. Este nick será armazenado numa base de dados e carregado quando, na interface de preenchimento de dados, for solicitado. Isso permite ao jogador, participar de várias partidas e acumular pontos de uma partida para outra.

**Requisitos Não Funcionais**

**Requisito não funcional 1 – Especificação de projeto:** além do código Java, deve ser produzida especificação de projeto baseada em UML, segunda versão;

**Requisito não funcional 2 – Interface gráfica para usuário:** o programa deverá ter interface gráfica distribuída, com cada usuário tendo a sua interface.

**Requisito não funcional 3 – Circular palavra formada:** a cada palavra formada o programa deve destacar de uma cor diferente a borda da respectiva palavra.

**Requisito não funcional 4 – Tecnologia de interface gráfica para usuário:** A interface gráfica deve ser baseada em Java-Swing.

**5 – Referencias**

[1] – Leonardo de Souza Brasil, Suporte ao desenvolvimento de jogos multijogador, sem exigência de tempo real. Disponível em: https://www.inf.ufsc.br/~netgames/NetGamesNRT/Documentos/NetGamesNRT\_TCC.pdf