**Projeto Shutdown**

1. **Especificação de Requisitos de Software**

Versão 1.0

08/09/2014

|  |  |  |  |
| --- | --- | --- | --- |
| **Versão** | **Autor(es)** | **Data** | **Ação** |
| 1.0 | Eduardo Getassi da Rosa | 08/09/2014 | Estabelecimento dos requisitos |

1. **Conteúdo:**
2. Introdução;
3. Visão Geral;
4. Requisitos de software.
5. **Introdução**

**Objetivo:** desenvolvimento de programa que suporte a disputa de partidas do jogo Shutdown na modalidade usuário contra usuário.

**Referências:**

<Punhalada Game Design Document – Lucas Bergmann da Silva>

Obs.: Esboço de interface no documento referenciado.

1. **Visão Geral**

**Arquitetura do programa:** programa orientado a objetos

**Premissas de desenvolvimento**:

* O programa deve apresentar uma interface gráfica bidimensional;
* O programa deve ser implementado em Java, devendo executar em qualquer plataforma que disponha de máquina virtual Java instalada;
* O programa deve ser distribuído utilizando o NetGamesNRT como *framework* de comunicações.

1. **Requisitos de Software**
   1. **Requisitos Funcionais**

**Requisito funcional 1 – Iniciar partida:** o sistema deve apresentar na interface inicial a opção “Iniciar partida” para permitir o início de uma nova partida, operação em que são identificados os jogadores e qual o jogador que inicia.

**Requisito funcional 2 – Abandonar partida:** o programa deve apresentar na interface de jogo a opção “Abandonar partida”, operação em que a partida é encerrada e o jogador que permaneceu é notificado como vitorioso.

**Requisito funcional 3 – Estabelecimento de turnos:** o programa deve implementar uma estrutura de turnos para a execução de cada partida. Em um determinado turno é possível que apenas um jogador esteja habilitado para realizar um procedimento de lance.

**Requisito funcional 4 – Procedimento de lance:** o programa recebe o procedimento de lance dos usuários por meio das setas de movimento do teclado, atualizando o tabuleiro observado as seguintes restrições:

* O jogador da vez ainda deve dispor de movimentos não utilizados do total de movimentos recebidos aleatoriamente (mínimo um, máximo seis) no início da rodada, para poder mover seu robô.
* A posição de destino é válida: não está ocupada por outro robô, ou a entrada na posição se dá pela face que contém a chave de desligamento do robô.
* A nova posição é no sentido de frente ou lados.

**Requisito funcional 5 – Procedimento de shutdown:** o programa deve suportar o procedimento de shutdown, ocorrido quando um robô A, entra no espaço ocupado por outro robô B a partir da face onde se encontra o mecanismo de desligamento do robô B, considerando as restrições a seguir:

* A disputa pelo espaço alocado é entre robôs de jogadores diferentes.
* Ao fim de uma jogada válida de desligamento, o programa deve verificar se há um ganhador para a partida.

**Requisito funcional 6 – Comunicar estado da partida:** a cada tentativa de procedimento de lance no tabuleiro ou de procedimento de shutdown, o novo estado da partida deve ser exibido na interface, dadas as seguintes observações:

* O tabuleiro deve ser atualizado com o número de movimentos restantes para o usuário da vez.
* Qualquer lance irregular deve ser comunicado ao usuário o motivo da irregularidade.
* Os robôs já desligados devem ser informados.
  1. **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 e *release notes*;

**Requisito não funcional 2 – Tecnologia de interface gráfica:** a interface gráfica deve ser baseada em Java-Swing;

**Requisito não funcional 3 – Interface gráfica do usuário:** o programa deverá apresentar uma interface única para ambos os jogadores nos casos de partidas entre usuários.