



UNIVERSIDADE DE SANTA CATARINA- UFSC
CENTRO TECNOLÓGICO
LUCAS VARELLA
MARCOS LAYDNER

**REQUISITOS DE SOFTWARE
PROJETO OLIMPIADAS DO FAUSTOP**

Florianópolis
02 de Outubro de 2017

Projeto Olimpíadas do Faustop

Especificação de Requisitos de Software:

Versão 0.3

28/11/2018

Versão	Autores	Data	Ação
0.1	Lucas Varella e Marcos Laydner	20/08/2018	Especificação de Requisitos inicial
0.11	Lucas Varella e Marcos Laydner	10/09/2018	Ajuste da Especificação de requisitos baseado nas orientações do professor para a entrega 2
0.2	Lucas Varella e Marcos Laydner	02/10/2018	Ajuste da Especificação de requisitos, com uma melhor descrição do jogo e seu <i>gameplay</i>
0.3	Lucas Varella e Marcos Laydner	28/11/2018	Ajustes Finais

ÍNDICE

1 INTRODUÇÃO.....	03
2 VISÃO GERAL.....	03
3 REQUISITOS DE SOFTWARE.....	03
4 ESBOÇO DA INTERFACE.....	05

1 Introdução

Objetivo: Desenvolvimento de um programa que suporte partidas do jogo “Olimpíadas do Faustop” onde competidores devem se mover pelo mapa, acumulando pontos e atrapalhando seu adversário.

2 Visão geral

Arquitetura do programa: programa orientado a objetos, com arquitetura cliente/servidor, utilizando o servidor NetGames para possibilitar a execução distribuída da aplicação.

Premissas de desenvolvimento:

- O programa deve apresentar uma interface gráfica bidimensional;
- O programa deve ser implementado em Java, e executar em qualquer plataforma que disponha da máquina virtual Java;
- A aplicação deverá suportar rede, através da arquitetura cliente/servidor, fazendo uso do NetGames, permitindo uma aplicação distribuída.

O jogo

Dado o início do jogo, dois jogadores são colocados em posições aleatórias de um mapa, formado por uma matriz de campos(tiles), que podem ser de diversos tipos, definidos aleatoriamente pelo programa no momento que são gerados. Cada tipo possui propriedades diferentes. As possibilidades são:

- **Obstáculo** - um *tile* para onde o jogador não pode se mover;
- **Blank** - um *tile* comum, não apresenta propriedades especiais, mas é válido para se mover;
- **Pergunta** - ao entrar nesse *tile* o jogador pode responder uma pergunta, escolhendo a resposta certa entre 4 alternativas. Se o jogador acertar, ele ganha certa quantidade de pontos, se ele errar, ele perde essa mesma quantidade. Além disso, o jogador pode passar a pergunta para seu oponente responder, caso não queira arriscar;
- **Surpresa** - ao entrar nesse *tile*, o jogador recebe uma das seguintes bonificações, escolhidas aleatoriamente pelo sistema:
 - **Bônus:** o jogador recebe uma certa quantidade de pontos;
 - **Armamento:** o jogador escolhe onde colocar uma armadilha, transformando o *tile* escolhido em um de Armadilha;
- **Armadilha** - ao entrar nesse *tile*, o jogador fica incapacitado, impedido de fazer movimentos, por um turno.

Os jogadores se movem pelos *tiles*, clicando nos *tiles* desejados. Os jogadores só conseguem identificar os *tiles* que já passaram. Os jogadores competem por pontos, e aquele que atingir a marca de X pontos primeiro, ou o que tiver mais pontos quando todos *tiles* forem explorados, ganha a partida.

3 Requisitos de Software

3.1 Requisitos Funcionais

Requisito Funcional 1 - Implementar o ciclo do jogo: o programa deve permitir que um jogador realize uma partida contra outro jogador, sendo que o programa deve anunciar um vencedor ao final da partida;

Requisito Funcional 2 – Iniciar: o programa deve apresentar a opção de menu “iniciar”, após se conectar à outro jogador, para o início de uma nova partida, operação que instancia o mapa e os jogadores e dá início a partida;

Requisito Funcional 3 – Reiniciar: o programa deve apresentar a opção de menu “reiniciar” para o início de uma nova partida, considerando as seguintes restrições:

- A operação mantém os dois jogadores já definidos;
- Não deve existir partidas em andamento para que se reinicie. Caso uma partida esteja acontecendo, esta opção não é disponível;

Requisito Funcional 4 – Jogada: o programa deve suportar o procedimento de comando dos usuários, para mover-se, responder perguntas e lançar armadilhas;

Requisito Funcional 5 - Conectar: o programa deve apresentar a opção de menu “conectar” para se conectar ao servidor NetGames, para que se possa dar início a uma partida;

Requisito Funcional 6 - Sair: o programa deve apresentar a opção “sair” no menu principal e quando a partida acabar, o que desconecta o usuário do server NetGames e encerra o programa;

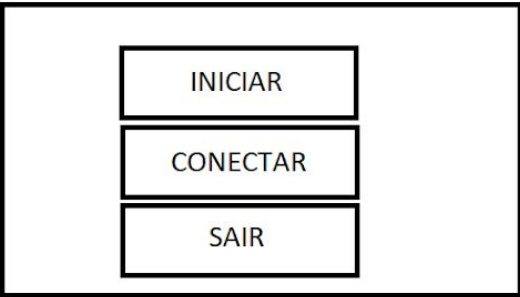
Requisito Funcional 7 - Desconectar : o programa deve apresentar a opção “Desconectar” quando a partida acabar, o que retorna o jogador ao menu principal e o desconecta do NetGames.

3.1 Requisitos não Funcionais

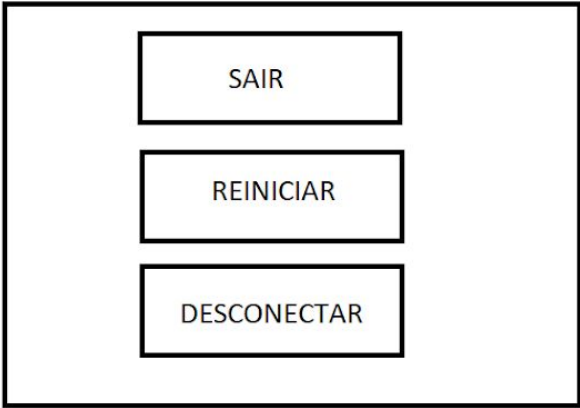
Requisito não Funcional 1 – Interface gráfica: o programa deve ter interface gráfica para cada um dos usuários;

Requisito não Funcional 2 – Especificação de projeto: além do código Java, deve ser produzida especificação de projeto baseada em UML 2;

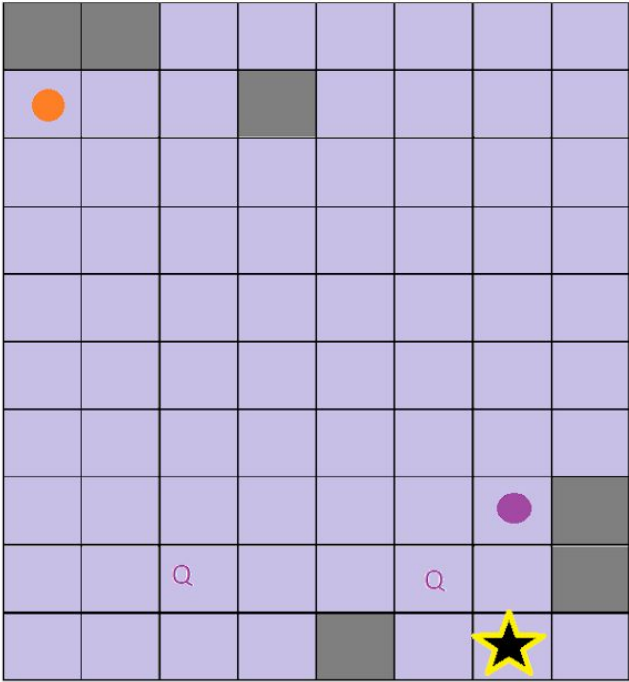
Requisito não Funcional 3 - Execução Distribuída : O jogo deve executar distribuído, com o suporte para aplicações distribuídas, NetGamesNRT, produzido por Leonardo Brasi.



Menu Principal



Menu Ingame



Esboço do jogo