

Daniel Yoshizawa e Jacqueline Cardozo

## **Especificações de Requisitos - Pack The Bag**

Florianópolis, SC, Brasil

14 de Setembro de 2016

# Sumário

	<b>Introdução</b>	<b>2</b>
<b>1</b>	<b>VISÃO GERAL</b>	<b>3</b>
1.1	Arquitetura do Programa	3
1.2	Descrição do Jogo	3
1.3	Detalhes de Implementação	3
1.4	Fluxo do Jogo	4
1.4.1	Posição Inicial	4
1.4.2	Jogadas	4
1.4.3	Objetivo do Jogo	4
1.5	Regras do Jogo	4
1.6	Premissas de desenvolvimento	4
<b>2</b>	<b>REQUISITOS DE SOFTWARE</b>	<b>6</b>
2.1	Requisitos Funcionais	6
2.2	Requisitos Não Funcionais	6
<b>3</b>	<b>ESBOÇO DA INTERFACE GRÁFICA</b>	<b>7</b>

# Introdução

O objetivo do projeto é desenvolver um jogo virtual em rede utilizando a linguagem Java, com o intuito de preencher malas com blocos de formas e cores variados de maneira que o máximo desta seja preenchido.

# 1 Visão Geral

## 1.1 Arquitetura do Programa

- Programa orientado a objetos, escrito em Java, utilizando a biblioteca Slick2D;
- Aplicação distribuída cliente-servidor utilizando o NetGamesNRT;

## 1.2 Descrição do Jogo

Dois jogadores disputam por peças para preencher suas malas, com o objetivo de usar o máximo de peças possíveis para deixar o mínimo de espaços em branco na sua mala. As peças serão agrupadas por cores que possuirão diferentes pontuações. Ao final, quando não houver mais posições disponíveis para peças em ambas as malas, os jogadores serão pontuados. Quem tiver mais pontos ganha.

A complexidade se dá em escolher as peças certas das categorias certas e reduzir a quantidade de peças que o adversário pode utilizar para ganhar mais pontos. Serão utilizadas peças do padrão Tetris e a mala em forma de grade. Também existirão categorias por cores de peças. As peças dentro das categorias com maior pontuação vão aparecer com menor frequência.

A grade terá tamanho variado, quadrados e retangulares. Cada peça pode assumir quatro posições pré-determinadas, o jogador não terá a opção de virar as peças. Estas peças podem assumir qualquer cor das categorias disponíveis. Não serão escolhas randômica, e sim um algoritmo de seleção para controlar a dificuldade.

A cada turno, um grupo novo de 3 peças serão apresentadas e o jogador da vez escolhe uma peça conforme sua estratégia e a arrasta para a posição desejada na grade. Logo após, passando o turno para o adversário o qual repete o processo até que não exista mais posições disponíveis nas malas de ambos os jogadores. Caso um jogador ocupe todas as posições disponíveis antes do outro, o jogador que ainda possuir espaços poderá continuar jogando.

## 1.3 Detalhes de Implementação

O jogo será desenvolvido em cima do framework Slick2D para aproveitar os padrões de projetos comuns a jogos, como o game loop, pools, commands entre outros, assim como a integração com engines de áudio e renderização gráfica.

## 1.4 Fluxo do Jogo

### 1.4.1 Posição Inicial

O jogo iniciará quando os dois jogadores estiverem conectados, o algoritmo escolherá um dos jogadores aleatoriamente para começar a jogada.

### 1.4.2 Jogadas

O jogador da vez escolherá uma peça dentro do catálogo de peças disponíveis e a posicionará na sua grade. O jogador poderá fazer uso de estratégia ao escolher a peça de uma categoria a fim de obter mais pontos ou dificultar o jogo do adversário. Após posicionar sua peça, o jogador passa a vez para o adversário.

### 1.4.3 Objetivo do Jogo

O jogo chega ao seu final quando as grades de ambos jogadores não possuir mais espaços para receber peças.

## 1.5 Regras do Jogo

- Cada jogador seleciona e adiciona em sua grade apenas uma peça por turno;
- A peça deve caber e não sobrepor outras peças na grade;
- O jogador tem o máximo de um minuto para realizar a sua jogada em um turno;
- O jogador tem autorização para adicionar peças apenas na sua grade;
- O jogador poderá continuar suas jogadas caso ainda tenha espaço disponível para peças em sua grade e haja peças disponíveis, mesmo que o adversário não tenha mais espaço em sua grade;

## 1.6 Premissas de desenvolvimento

O software deverá :

- Ser implementado em linguagem de programação Java, utilizando paradigma de orientação a objetos, devendo executar em qualquer plataforma que disponha de máquina virtual Java;
- Apresentar uma interface gráfica bidimensional (2d);

- Ser um jogo virtual em rede, na modalidade jogador contra jogador, utilizando o framework NetGamesNRT;

## 2 Requisitos de Software

### 2.1 Requisitos Funcionais

- Conectar - O programa deve apresentar uma opção conectar, para estabelecer conexão com o servidor. O jogador deverá definir o nome que o identificará durante a partida nesta etapa;
- Desconectar - O jogo deve apresentar a opção desconectar, permitindo ao jogador desconectar do servidor, encerrando assim uma possível partida em andamento;
- Iniciar Partida - O programa deve apresentar uma opção iniciar para que a partida tenha início. É necessário que os dois jogadores estejam conectados ao servidor para iniciar uma nova partida;
- Selecionar peça - As peças devem ser geradas no início do turno, podendo ser selecionadas e arrastadas até a posição desejada, está ficará na posição caso não colida com nenhuma peça já colocada na grade ou com os limites da grade;
- Sair da partida - O programa deve apresentar uma opção sair ao terminar a partida, que fará o programa voltar a tela inicial;

### 2.2 Requisitos Não Funcionais

- Especificações de projeto - Software escrito na linguagem de programação Java, e especificação feita em UML segunda versão;
- Interface gráfica para o usuário - O programa deverá ter interface gráfica única, compartilhada pelos usuários;
- Peças - Serão geradas peças no padrão Tetris, podendo assumir diferentes cores;
- Tecnologia de interface gráfica para usuário - A interface gráfica será desenvolvida a partir da biblioteca Slick2d;

### 3 Esboço da Interface Gráfica

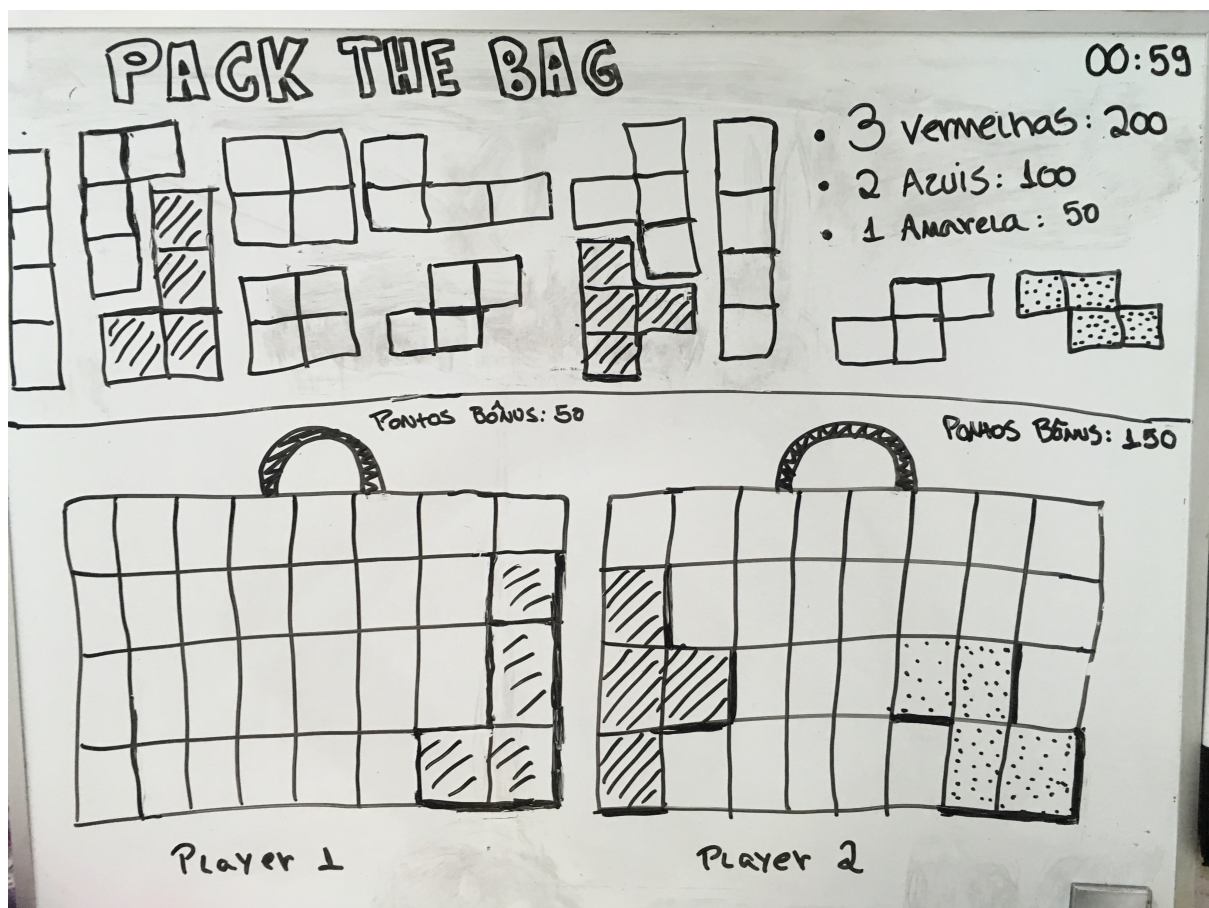


Figura 1 – Esboço da Interface Gráfica