

Projeto Trivial Pursuit

Especificação de Requisitos de Software

Versão 1.0
26/08/2019

Histórico de Versões			
Versão	Autor(es)	Data	Ação
1.0	Alexandre Müller Júnior Cainã Rinaldi Esteche Fernanda Narloch Rizzo Hahn	26/08/2019	Estabelecimento dos requisitos

Alunos: Alexandre Müller Júnior
Cainã Rinaldi Esteche
Fernanda Narloch Rizzo Hahn

Conteúdo

1. Introdução	3
1.1 Objetivos	3
1.2 Referências	3
1.3 Adaptação das regras do jogo	3
1.4 Definição do projeto	3
2. Visão Geral	4
2.1 Arquitetura da aplicação	4
2.2 Premissas de desenvolvimento	4
3. Requisitos da aplicação	4
3.1 Requisitos funcionais	4
3.2 Requisitos não funcionais	4
4. Protótipo da interface gráfica do programa	4

1. Introdução

1.1 Objetivos

O projeto aqui especificado tem como objetivo o desenvolvimento de um programa que suporta partidas em rede de um jogo baseado no jogo de tabuleiro Trivial Pursuit para dois jogadores.

1.2 Referências

As regras do jogo original, sem as devidas adaptações para o projeto, podem ser encontradas no endereço: <https://pt.m.wikihow.com/Jogar-Trivial-Pursuit>.

1.3 Adaptação das regras do jogo

Para simplificar o desenvolvimento, algumas regras do jogo bem como o design do tabuleiro foram alterados. A começar pelo número de jogadores, que foi reduzido a, no máximo, dois jogadores ou dois grupos. Além disso, as perguntas seguirão o padrão de respostas através de alternativas pré-definidas, ao invés de texto digitado pelo usuário. Por fim, o tabuleiro circular não possuirá a parte central e nem as casas de “Jogue novamente”, sendo, também, reduzido em número de casas. Pelo fato de não possuir a porção central, será adicionada ao tabuleiro circular uma casa para início dos peões no jogo.

1.4 Definição do projeto

O jogo possui um tabuleiro com 18 casas. Dessas, sendo uma casa de início e 6 casas de perguntas, o restante das casas não possuem efeito algum (casas comuns). As casas de perguntas são separadas por 2 casas comuns.

Cada casa de pergunta possui uma cor correspondente a uma área de conhecimento, essas encontram-se listadas abaixo:

- **Azul:** Geografia
- **Amarelo:** História
- **Roxo:** Arte e Literatura
- **Laranja:** Desporto e Lazer
- **Verde:** Ciências e Natureza
- **Rosa:** Entretenimento

No início do jogo, será sorteado o jogador ou grupo que deverá começar a jogar. Após, em seus turnos, os jogadores devem se deslocar pelo tabuleiro de forma a tentar posicionar seu peão em uma casa de pergunta. Para isso, os jogadores rolam o dado

para designar o número de casas de deslocamento e podem, então, escolher o sentido do movimento (horário ou anti-horário).

Ao cair com seu peão em uma casa de pergunta e acertá-la, o jogador ganha uma peça da cor da área de conhecimento em questão. O objetivo do jogo é que seja coletada uma peça de cada cor pelo jogador. Quando um jogador conquista o objetivo, o jogo é encerrado.

2. Visão Geral

2.1 Arquitetura da aplicação

Aplicação orientada a objetos e um sistema distribuído com base no modelo cliente-servidor.

2.2 Premissas de desenvolvimento

- A linguagem de programação utilizada para o desenvolvimento do programa deverá ser Java;
- O programa deverá possuir uma interface gráfica bidimensional;
- A modelagem da aplicação deverá ser desenvolvida em UML 2;
- O programa deverá ser executado distribuído, com o suporte para aplicações distribuídas do *NetGamesNRT*.

3. Requisitos da aplicação

3.1 Requisitos funcionais

- 1) Conectar: o programa deve oferecer uma opção no menu que permita ao usuário se conectar ao servidor do *NetGames* e começar uma partida com outro jogador em um computador diferente.
- 2) Iniciar nova partida: caso conectado ao servidor, o menu deve ter uma opção para começar uma partida. O programa, então, envia uma solicitação ao servidor e aguarda uma resposta de que está tudo certo para dar início à partida.
- 3) Receber solicitação de início: após receber a informação do servidor, o programa sorteia o jogador que irá iniciar o jogo. Também posiciona dois peões na casa de início, um para cada jogador. E, passa a mostrar a contagem de peças coloridas dos jogadores.

Com a partida em andamento, o programa deve possibilitar as seguintes funcionalidades aos jogadores:

- 5) Lançar dado: em sua rodada, o jogador poderá rolar o dado de 6 lados para definir aleatoriamente o número de casas do deslocamento.
- 6) Escolher sentido do movimento: após lançar o dado, o jogador poderá escolher se seu peão irá se deslocar no sentido horário ou anti-horário no tabuleiro.
- 7) Responder pergunta: ao cair numa casa de pergunta, o jogador poderá escolher entre 5 opções de resposta para a pergunta do tema da casa em questão.
- 8) Receber jogada: o programa deverá atualizar o tabuleiro e a quantidade de peças de cada jogador à medida em que as jogadas são feitas. Também deverá anunciar o vencedor, caso um dos jogadores conquiste o objetivo.
- 9) Desconectar: deverá existir uma opção no menu que permita que o usuário se desconecte do servidor *NetGames*.

3.2 Requisitos não funcionais

- 1) Especificações do projeto: as especificações do projeto serão baseadas em UML 2.
- 2) Linguagem de programação: a linguagem utilizada para desenvolvimento do projeto será Java.
- 3) Interface gráfica: a interface bidimensional do projeto será desenvolvida utilizando a biblioteca *Java Swing*.
- 4) IDE de desenvolvimento:
- 5) Framework de computação distribuída: o framework a ser utilizado para a comunicação servidor-usuário será o *NetGamesNRT*.

4. Protótipo da interface gráfica do programa



