Alunos: Maurício de Vasconcelos Barros, Matheus Lucca do Carmo, Lucas Cunha de Oliveira

**Projeto para Análise e Projeto de Sistemas – Jogo “21: Dado Duplo”**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versão** | **Autores** | **Data** | **Ação** |
| 0.1 | Maurício Barros, Lucas Cunha, Matheus Carmo | 28/08/2018 | Estabelecimento dos requesitos |
| 0.2 | Maurício Barros, Lucas Cunha, Matheus Carmo | 15/10/2018 | Adicionado requisito funcional : Controle de turnos |
| 0.3 | Maurício Barros, Lucas Cunha, Matheus Carmo | 20/11/18 | Eliminado caso de uso : Troca de mensagens |

**Conteúdo:**

1. Introdução;

2. Visão geral;

3. Requisitos de software.

4. Interface gráfica (esboço).

**1 Introdução**

**Objetivo:** Desenvolver um programa (jogo) permitindo a disputa multiplayer, de 2 a 6 jogadores do jogo “21: Dado Duplo”.

**2 Visão Geral**

**Arquitetura do programa:** Programa baseado no paradigma orientado a objetos, construido com auxílio do NetGames Framework de forma a estabelecer uma aplicação distribuida cliente-servidor.

**Regras do jogo:** “21: Dado Duplo” é um jogo multijogadores, para ser jogado entre 2 a 6 pessoas, sendo necessário 2 dados para fazer as jogadas, onde o jogador tenta identificar se outro jogador está mentindo quando declara o número que tirou nos dados. O objetivo é identificar os mentirosos e não ser pego mentindo.

Para jogar é necessário dois dados e um copo não-transparente, decidido a ordem de início, o jogo irá rodar no sentido anti-horário. Cada jogador irá jogar os dois dados (somente ele poderá ver o conteúdo dos dados) e declarar para a mesa o resultado dos dados (entre os dois dados, o número maior sempre pertencerá à casa da dezena, por exemplo os números 2 e 3, sempre será 32 e nunca 23); tendo declarado para a mesa o número de seus dados (podendo ser verdade ou não), o próximo jogador tem duas opções: Aceitar o número dito ou duvidar da veracidade do número dito. Tendo duvidado do número dito, o jogador que declarou os números, mostrará para todos o conteúdo dos dados, se ele estiver falando a verdade, o jogador que duvidou perde 2 pontos; se ele estiver mentindo e for pego na mentira, perde-se 1 ponto (por mentir). Tendo ele aceito os números do jogador anterior (acreditado), deve-se por sua vez jogar os dados novamente, porém o número a ser declarado deve sempre ser igual ou maior que o número anterior; segue a ordem dos números do jogo do menor para o maior: 31, 32, 41, 42, 43, 51, 52, 53, 54, 61, 62, 63, 64, 65, 11, 22, 33, 44, 55, 66, 21.

Aquele que alcançar 21 pontos (negativos) está elimindao, o último a ficar no jogo é o vencedor.

**Premissas de desenvolvimento:**

O software deve apresentar alguma interface gráfica.

O software deve ser implementado seguindo o paradigma orientado a objetos.

O software deve ser implementado em linguagen Java.

O software deve ser cliente do servidor NetGames.

**3 Requisitos de Software**

**Requisitos Funcionais:**

Requisito funcional 1: **Conectar**: O programa deve possuir a opção de menu “Conectar ao NetGames” aonde o Jogador faz um requerimento para se conectar ao server.

Requisito funcional 2: **Iniciar:** O programa deve possuir a opção de menu “Iniciar Partida” aonde o Jogador faz um requerimento para iniciar partida.

Requisito funcional 3: **Sair:** O programa deve possuir a opção de menu “Sair” onde o Jogador faz o requerimento para sair do jogo.

Requisito funcional 4: **Associar Nome:** O programa, ao início da partida, deve permitir ao jogador escrever seu nome que aparecerá para os outros jogadores.

Requisito funcional 5: **Procedimento da Jogada:** O programa deve permitir ao jogador atual as opções possíveis; ou jogar os dados, ou duvidar/aceitar a jogada do jogador anterior. O programa deve certificar-se de permitir apenas jogadas legais, e ao final de cada jogada deve verificar por vencedores ou perdedores.

Requisito funcional 6: **Comunicação do estado da partida:** A cada lance, o número atual do jogo deve ser disponobilizado para todos os jogadores.

Requisito funcional 7: **Persistência:** A cada final de partida, o programa deve salvar o nome do Jogador campeão.

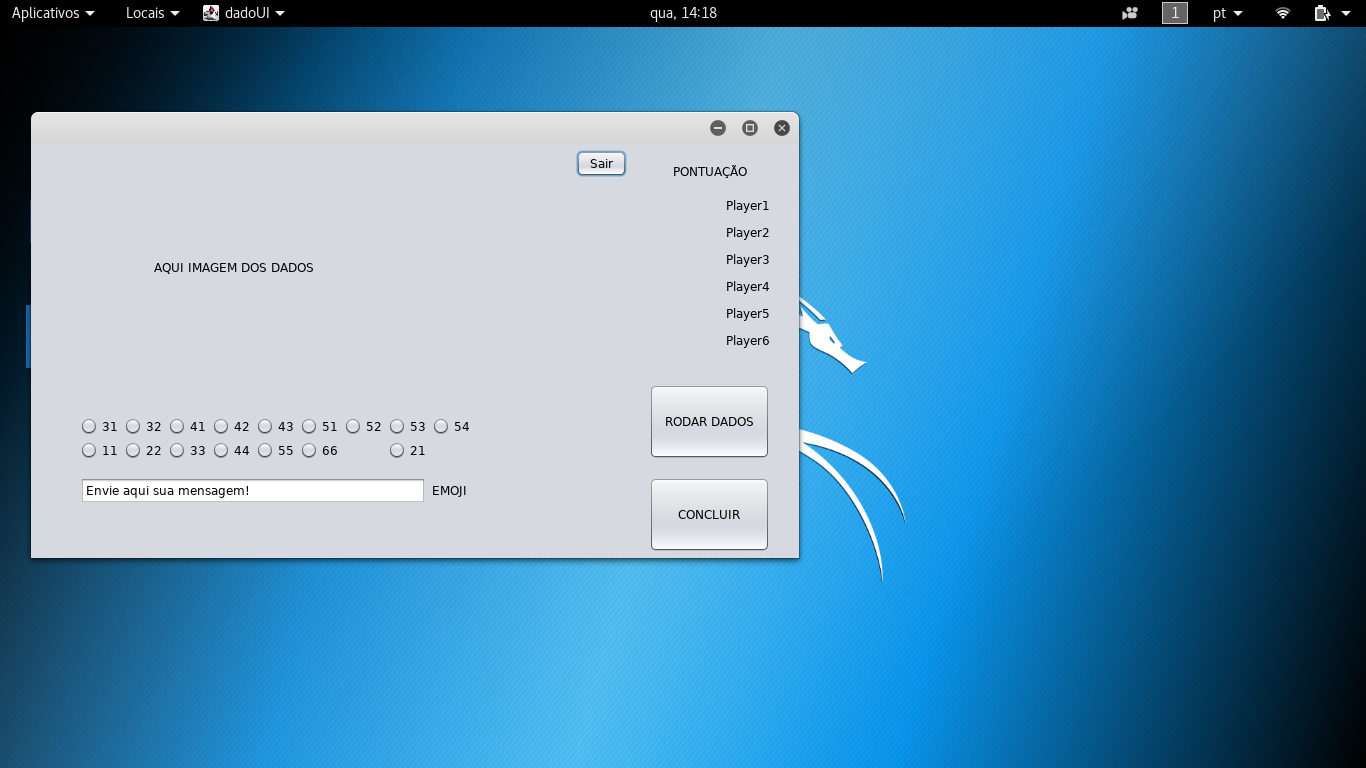
Requisito funcional 8: **Controle de turnos:** O programa deve controlar a ordem de jogadas, assim como a o jogador atual.

**Requisitos não-funcionais:**

Requisito não-funcional 1: O software deve ser modelado em UML no Visual Paradigm.

Requisito não-funcional 2: O software deve possuir uma interface gráfica em duas dimensões.

**4 Interface gráfica**

****