

Projeto Jogo da Bugalha

Especificação de Requisito de Software

Es_PJ_jogoDaBugalha_2022

Versão 1.1

17/10/2022

Versão	Autor(es)	Data	Ação
1.0	Emily Sayuri Kiba, Marco Antônio Machado de Arruda, Mikaella Cristina Bernardo Vieira	19/09/2022	Estabelecimento dos requisitos
1.1	Emily Sayuri Kiba, Marco Antônio Machado de Arruda, Mikaella Cristina Bernardo Vieira	24/10/2022	Incluindo imagem da GUI no requisito não funcional
1.2	Emily Sayuri Kiba, Marco Antônio Machado de Arruda, Mikaella Cristina Bernardo Vieira	14/11/2022	Fusão de recursos funcionais
1.3	Emily Sayuri Kiba, Marco Antônio Machado de Arruda, Mikaella Cristina Bernardo Vieira	04/12/2022	Mudanças na interface gráfica

Conteúdo:

1. Introdução
2. Visão Geral
3. Requisitos de Software

1. Introdução

1.1. Objetivos

O objetivo do projeto é desenvolver um minigame encontrado no jogo Cult of the Lamb chamado de Jogo da Bugalha, com suporte a Usuário x Usuário de forma distribuída.

1.2. Referências (Regras do jogo)

Para regras escritas ver apêndice

[1] <https://www.youtube.com/watch?v=rLx_N2VVLX0&t=59s> (vídeo exemplificando o funcionamento do jogo, acessado em 11/09/2022)

2. Visão Geral

2.1. Arquitetura do Programa

Programa orientado a objeto com aplicação distribuída.

2.2. Premissas de Desenvolvimento

O programa deve:

- Ser implementado em python
- Usar o DOG na execução distribuída
- Usar a plataforma Visual Paradigm no desenvolvimento da documentação UML

3. Requisitos de Software

3.1. Requisitos Funcionais

- **[RF1] Iniciar Programa**

Ao iniciar o jogo o usuário será recebido por uma tela inicial do programa que pedirá pelo seu nome, logo após isso a conexão com o servidor tentará ser estabelecida, caso seja bem sucedida o jogador será levado para a interface do jogo em si, com o tabuleiro já disposto onde ele utilizará o tabuleiro da parte inferior enquanto seu adversário será representado na parte superior.

- **[RF2] Iniciar Partida**

Será apresentado a opção de “Iniciar partida” no menu do jogo, assim que o oponente também estiver pronto, começará o jogo. Esse evento será ativado pelo jogador com um clique no botão após a conexão ser efetuada, enviando o pedido de início via rede do DOG.

- **[RF3] Receber Determinação de Início**

Relacionado com o [RF2], um jogador pode receber uma notificação de início de partida caso esteja conectado via DOG, essa opção é ativada por outro jogador na rede ao pedir o início da partida em sua interface. Neste caso, o DOG poderá permitir ou não o jogador a fazer o primeiro movimento na partida.

- **[RF4] Receber Jogada**

O jogo local irá receber jogadas do seu adversário a cada rodada do oponente, enviadas pelo DOG Server. Temos duas possibilidades, a que deve ocorrer primeiramente na vez do oponente, dada pelo envio da face tirada no dado para que a interface seja atualizada e a segunda que é a posição em que o dado foi jogado, a jogada em si que deve ser um lance legal conforme as regras e conter as informações necessárias ao programa, nela, o jogador deve escolher a posição de um dado no seu lado do tabuleiro e que os efeitos sejam verificados e calculados, sendo eles: apagar os dados do adversário caso estejam na mesma coluna (mas cada um na sua metade do tabuleiro), ativar o bônus de multiplicação caso mais de um dado com o mesmo número na mesma coluna do seu campo apareçam simultaneamente, deve também verificar a finalização da partida, dada pelo campo cheio após uma jogada, nesse caso de encerramento o jogador deve ser notificado com quem foi o vencedor, caso contrário o jogador local deve estar habilitado para realizar uma jogada.

- **[RF7] Receber Notificação de Abandono**

O servidor DOG pode enviar uma notificação de abandono caso ocorra desconexão do jogador remoto, uma notificação de desistência deve ser recebida, jogo considerado finalizado e nesse caso a conexão do local deve ser interrompida para que o usuário não fique travado a espera de uma resposta na rede.

- **[RF8] Selecionar Posição**

Durante o turno do jogador, primeiramente um dado será “rodado” (ver RF9), então conforme as regras, o jogador precisa selecionar uma posição em uma das colunas do tabuleiro, na qual o dado com a face sorteada será inserido, respeitando as regras do jogo, isto é, garantindo a escolha de uma posição válida, em seu próprio lado do tabuleiro e em uma posição anteriormente vazia, após a jogada em uma posição disponível, o jogo deve realizar os efeitos do posicionamento do dado, como, multiplicar a soma de dados com um mesmo lado na mesma coluna e também verificar se há algum dado adversário com igual numeração na mesma coluna mas no campo oposto, caso tenham N dados, todos os N dados devem ser retirados do tabuleiro. Ao final da jogada, ela precisa ser enviada ao oponente e o tabuleiro precisa ser verificado, caso o lado esteja completo, então o jogo é encerrado, as pontuações são comparadas e o vencedor é escolhido, caso contrário a rodada é passada ao jogador remoto.

- **[RF9] Reiniciar Tabuleiro**

Reseta o tabuleiro para que uma nova partida seja iniciada, para que os jogadores não precisem se desconectar e reconectar a cada nova partida, os jogadores podem aceitar uma revanche após um deles ser declarado vencedor. A oportunidade deve ser aceita por ambos jogadores para acontecer e enviada via DOG Server.

- **[RF10] Lançar Dado**

O lançar dado acontece quando o jogador clicar em um botão, o qual vai girar um dado, aleatorizando uma face para o jogador, onde ele posteriormente vai encaixar o dado com essa face gerada no tabuleiro, após atualização da interface o dado deve ser enviado ao adversário via DOG

- **[RF11] Atualizar tabuleiro - caso de uso auxiliar**

Atualizar tabuleiro é um novo caso de uso criado com a interseção de casos de uso selecionar posição e receber jogada. Alteraria as informações no tabuleiros para ambos lados de jogo. Primeiro verifica se é sua vez, se for realizar verificações de dados iguais no oponente e remove o dado. Depois redesenha tabuleiro e coloca dado onde jogador selecionou no interface gráfica. Caso não seja sua vez, verifica igualmente se possui dados iguais na mesma coluna do jogador remota e se houver remove dado do seu lado. Após isso atualiza a interface gráfica igualmente.

- **[RF12] Verificar vencedor - caso de uso auxiliar**

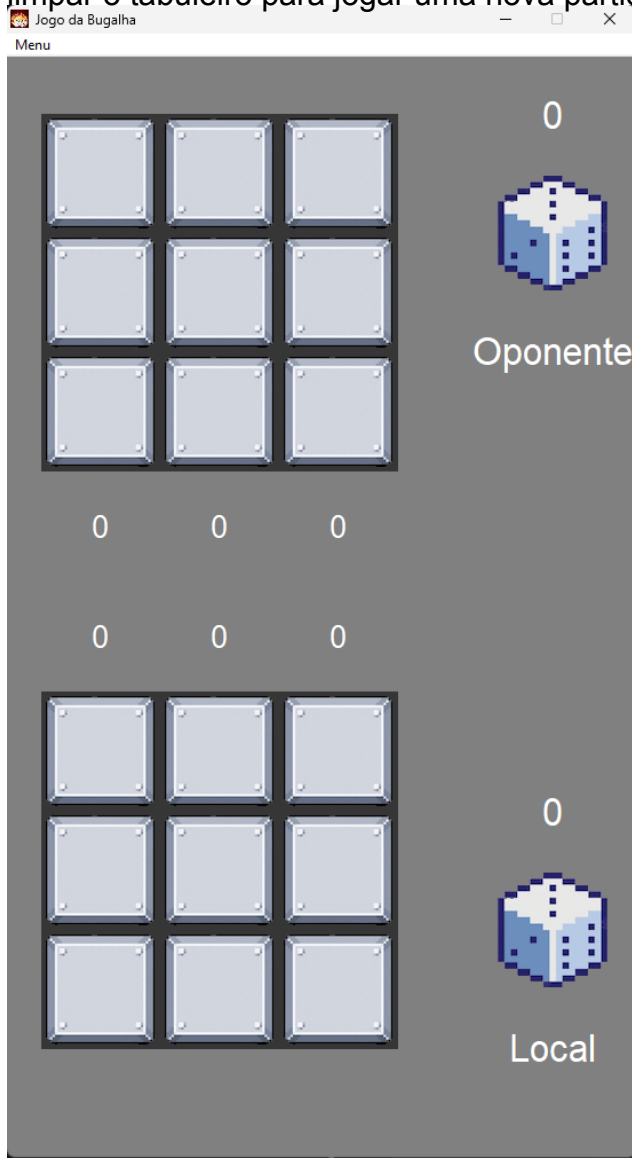
Deve verificar se após uma jogada o tabuleiro se encontra cheio ou não, em caso afirmativo ele deve comparar as pontuações, armazenar o nome de quem ganhou e enviar uma notificação de fim da partida (ou de empate)

3.2. Requisitos Não funcionais

Requisito não funcional 1 - Linguagem de programação: O programa deve ser implementado utilizando a linguagem de programação Python, em sua versão 3.

Requisito não funcional 2 - Especificação do Projeto: Além do código em Python deve ser produzido uma especificação de projeto baseado no UML2.

Requisito não funcional 3 - Interface Gráfica com o Usuário: O programa deve possuir uma interface gráfica com o usuário baseada em Tkinter, nela na parte superior é apresentado o campo do adversário com as 9 posições e os dados já inseridos. Da mesma forma, na parte inferior temos o tabuleiro do jogador local, ao lado de ambos tabuleiros temos o dado da jogada atual e entre os tabuleiros serão apresentadas as somas das colunas de ambos jogadores, à direita temos o nome dos jogadores e sua pontuação total. Será também apresentado um menu capaz de limpar o tabuleiro para jogar uma nova partida.



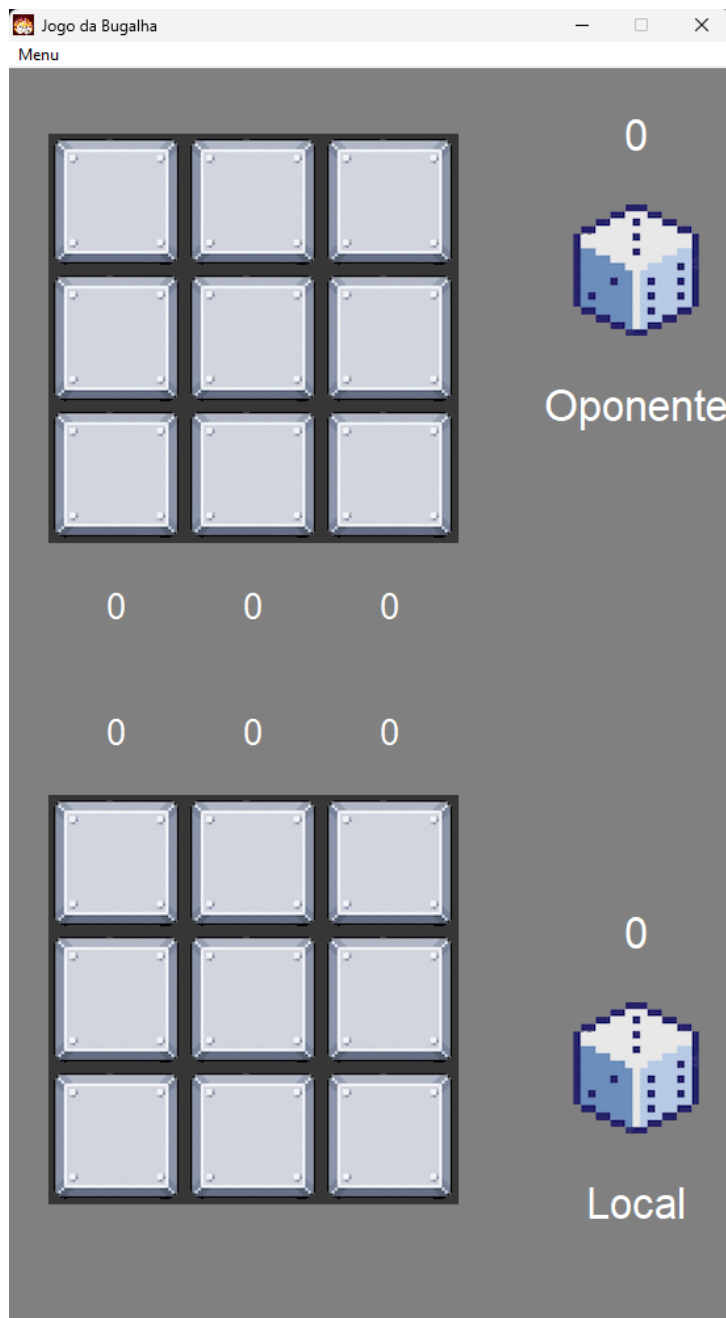
Requisito não funcional 4 - Implementação distribuída: O jogo deve apresentar suporte para multiplayer utilizando DOG Framework.

Apêndice: Regras Jogo da Bugalha

O Jogo da Bugalha é disputado entre 2 jogadores, cada um com seu lado do tabuleiro sendo esse representado em um formato matricial 3x3, o objetivo principal do jogo é obter uma pontuação maior que a de seu oponente ao final da partida.

Elementos do Jogo

O jogo é representado da seguinte forma:



Na parte superior temos o campo adversário onde os dados são postos nas jogadas do usuário remoto, já na parte inferior é representado o jogador local. Cada jogada representa a inserção de um dado cuja face é sorteada aleatoriamente para os jogadores em sua rodada. Ao lado de cada jogador temos também o dado que foi sorteado para sua vez junto de seu nome e a pontuação total. Entre as colunas aparece também a pontuação de cada uma delas num dado instante.

Posicionamento do Dado

Durante a rodada de cada jogador um dado é “rodado” e disposto ao lado de seu tabuleiro qual a face sorteada, então, cada jogador precisa seguir regras simples de posicionamento, sendo elas a escolha de uma posição do tabuleiro, e a escolha de uma posição não ocupada por outro dado. Após ocorrer a escolha os efeitos do posicionamento devem ser analisados, sendo eles:

Caso haja mais um dado em uma mesma coluna do lado do seu campo com a mesma face, a pontuação neste caso é dada pela soma dos dois dados multiplicada pela quantia de dados existentes na coluna com o mesmo lado, exemplo: em uma mesma coluna temos 3, 3, 6 então a pontuação daquela coluna é $(3+3)*2 + 6 = 18$.

Caso ocorra de seu adversário manter em uma coluna de seu lado do tabuleiro um dado e na rodada do jogador local a mesma face for sorteada, caso ainda tenha espaço na coluna do jogador, a jogada realizada pode ser feita na mesma coluna do seu oponente (mas em seu lado do campo), se assim feita o dado adversário deve ser apagado, independentemente de quantos dados existirem (nota, somente aqueles com mesma face voltada para cima).

Encerramento da Partida

A partida se dá por encerrada após a jogada de algum dos jogadores resultar em seu lado do tabuleiro sem espaço para mais dados, caso isso aconteça o jogo é dado por encerrado e as pontuações totais são calculadas, resultando em vencedor aquele com mais pontos obtidos.