

Projeto Conjunto

Especificação de Requisitos de Software

Versão	Autores	Data	Ação
1.0	Julia Macedo de Castro Tiago Faustino de Siqueira William Kraus	15/04/2024	Estabelecimento dos requisitos
2.0	Julia Macedo de Castro Tiago Faustino de Siqueira William Kraus	13/05/2024	Novo requisito funcional “11 - Selecionar destino” e reescrita outros requisitos funcionais.
3.0	Julia Macedo de Castro Tiago Faustino de Siqueira William Kraus	01/07/2024	Mudança no requisito funcional “1 - Inicializar”.

Conteúdo

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

Apêndice: Regras do jogo Conjunto

1. Introdução

1.1 Objetivo

Desenvolvimento de um programa distribuído que suporte a disputa de partidas de Conjunto na modalidade usuário contra usuário.

1.2 Definições, abreviaturas

Regras do jogo: ver apêndice

1.3 Referências

Apresentação das regras do jogo (video do canal Vem Ka Jogar):

<https://www.youtube.com/watch?v=pS9mUC2sQEc>

2. Visão Geral

2.1 Arquitetura do programa

Cliente-servidor distribuído.

2.2 Premissas de desenvolvimento

O programa deve ser implementado em Python;

- O programa deve usar DOG como suporte para execução distribuída;
- Além do código, deve ser produzida especificação de projeto baseada em UML, segunda versão.

3. Requisitos de Software

3.1 Requisitos Funcionais

Requisito funcional 1 – Inicializar:

Ao ser executado, o programa deve alocar os recursos necessários para carregar a interface, escolher um nome para o usuário e se conectar ao DOG. Caso a conexão falhe, mostra na interface a mensagem de erro pedindo pro jogador tentar novamente. Caso contrário, o jogador estará conectado ao DOG e o programa exibirá o nome do jogador. Neste ponto, o jogador está pronto para iniciar uma partida ou entrar numa partida iniciada remotamente.

Requisito funcional 2 – Iniciar partida:

O programa deve apresentar a opção de menu “iniciar jogo” para o início de uma nova partida. O procedimento de início de partida consiste em enviar uma solicitação de início a Dog Server, que retornará o resultado, que será a identificação e a ordem dos jogadores, em caso de êxito, ou a razão da impossibilidade de início de partida, caso contrário. A interface do programa deve ser atualizada com as informações recebidas e caso o jogador local seja quem inicia a partida, a interface deve estar habilitada para seu procedimento de lance. Esta funcionalidade só deve estar habilitada se o programa estiver em seu estado inicial, isto é, sem partida em andamento e com o tabuleiro em seu estado inicial;

Requisito funcional 3 – Receber pedido de início:

O programa deve poder receber uma notificação de início de partida, originada em Dog Server, em função de solicitação de início de partida por parte de outro jogador conectado ao servidor. O procedimento a partir do recebimento da notificação de início é o mesmo descrito no ‘Requisito funcional 2 – Iniciar partida, isto é, a interface do programa deve ser atualizada com as informações recebidas e caso o jogador local seja quem inicia a partida, a interface deve estar habilitada para seu procedimento de lance.

Requisito funcional 4 – Selecionar argola:

O programa deve permitir que o jogador habilitado selecione uma argola dentro da sua área de peças. Primeiro, ele verifica a argola atualmente selecionada. Caso nenhuma esteja selecionada, ele marca o tipo da argola que o jogador acabou de clicar. Se já houver uma argola selecionada e seu tipo for diferente do recém-selecionado, a seleção é alterada para o novo tipo. Em seguida, ele verifica a quantidade de argolas disponíveis do tipo escolhido e, se houver uma ou mais disponíveis, a argola é marcada como selecionada. Se não houver argolas desse tipo (zero disponíveis), a seleção não é feita e o programa simplesmente sai.

Requisito funcional 5 – Posicionar argola:

O programa deve permitir que novas argolas possam ser posicionadas no tabuleiro. Esta ação pode acontecer com iniciativa local ou remota. O programa

deve verificar se a posição destino aceita a argola escolhida, seguindo as regras do jogo descritas [aqui](#). Se a colocação for válida, o procedimento é encerrado.

Requisito funcional 6 – Selecionar célula:

O programa deve permitir que o jogador habilitado selecione uma célula no tabuleiro. Ele verifica a célula atualmente selecionada e, caso haja uma, ela é desmarcada. A célula recém-selecionada pelo jogador é então marcada. Se a nova célula for diferente da anterior, o programa destaca todas as células para as quais é possível mover o conjunto associado. Caso contrário, simplesmente desmarca a seleção.

Requisito funcional 7 – Mover conjuntos entre células:

Este procedimento denota a capacidade do programa de mover conjunto de argolas entre células do tabuleiro. A ação pode ser disparada pelo jogador local ou remoto. As regras para um movimento ser válido estão detalhadas [aqui](#).

Se o movimento for válido, o programa posiciona o conjunto na célula de destino. Por fim, a interface do usuário é atualizada caso algum movimento tenha sido feito. Nos casos em que a célula inicial está vazia (sem conjunto a ser movido) ou a célula de destino for inválida, o procedimento não possui efeito.

Requisito funcional 8 – Receber movimento:

O programa deve ser capaz de receber uma jogada do adversário, enviada pelo Dog Server, quando for a vez do adversário do jogador local. A jogada recebida deve ser um lance válido e conter as informações especificadas nos requisitos funcionais [5](#) e [10](#). O programa deve então adicionar um novo anel ao tabuleiro ou mover um conjunto, conforme o tipo de movimento recebido. Depois disso, ele atualiza o estado do jogo para habilitar o próximo jogador e, em seguida, atualiza a interface do usuário.

Requisito funcional 9 – Receber pedido de abandono de partida:

O programa deve poder receber uma notificação de abandono de partida por parte do adversário remoto, enviada por Dog Server. Neste caso, a partida deve ser considerada encerrada e o abandono notificado na interface.

Requisito funcional 10 – Restaurar estado inicial:

O programa deve ser capaz de "restaurar estado inicial" para redefinir o jogo. Ao selecionar essa opção, o programa limpa o conteúdo do tabuleiro, restaura a quantidade de anéis de cada jogador e define o estado do jogo como "Partida não iniciada". Em seguida, atualiza a interface do usuário para refletir as alterações.

Requisito funcional 11 - Selecionar destino:

O programa espera que um jogador clique numa célula no tabuleiro para selecionar o destino do posicionamento de uma nova argola ou o movimento de uma célula do tabuleiro e, para isso, uma argola ou célula precisa já estar selecionada. Se este for o caso, o movimento é realizado, como descrito pelos requisitos funcionais 5 e 7. Após isso, deve-se avaliar o encerramento de partida. No caso de encerramento de partida, deve ser notificado o nome do jogador vencedor, enviando o movimento de encerramento ao adversário. Caso a partida ainda prossiga, alterna-se o jogador habilitado (trocando de turno) e marca como "próximo" o movimento a ser enviado. Por fim, caso algum movimento tenha sido feito, a interface do usuário deverá estar atualizada e o movimento realizado deverá ser enviado pro servidor.

3.2 Requisitos Não Funcionais

Requisito não funcional 1 -

- Tecnologia de interface gráfica para usuário:
A interface gráfica deve ser baseada em TKinter;

Requisito não funcional 2 -

- Suporte para a especificação de projeto:
A especificação de projeto deve ser produzida com a ferramenta *Visual Paradigm*;

Requisito não funcional 3 -

- Interface do programa:
A interface do programa será produzida conforme ilustra a figura 1.

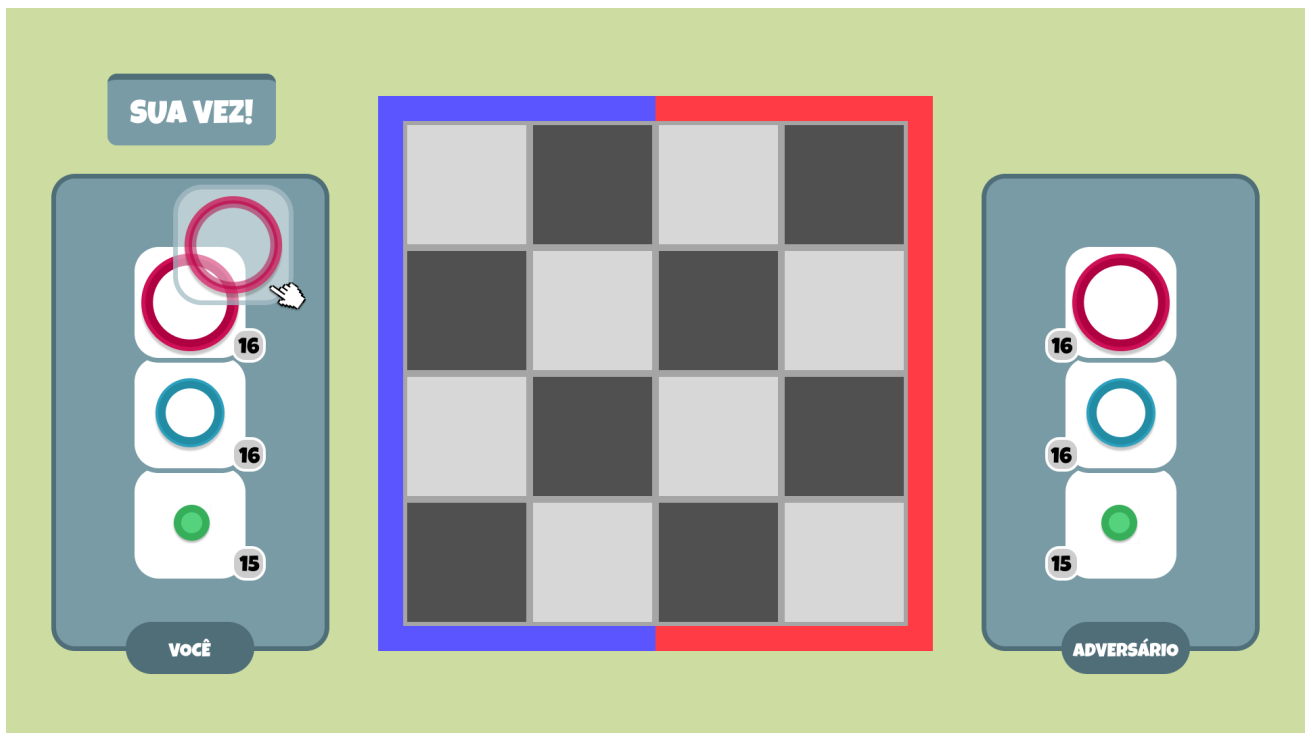


Figura 1: Interface gráfica do jogo

Apêndice: Regras do jogo Conjuntos

Elementos do Jogo

O jogo Conjunto é jogado por 2 jogadores em um tabuleiro 4x4. Cada jogador possui argolas de 3 tamanhos diferentes. As principais informações visuais do jogo estão descritas na figura 2.

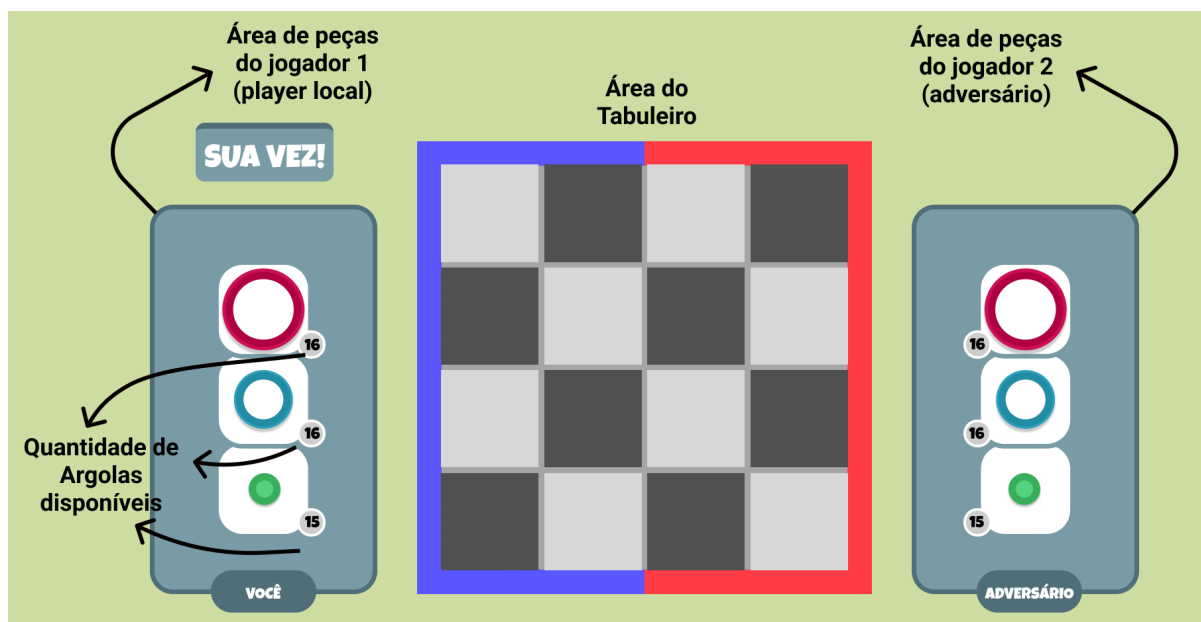


Figura 2: Elementos do jogo Conjunto

Lance dos Jogadores

Os jogadores alternam suas jogadas a cada rodada. Durante sua vez, cada jogador pode optar por inserir uma argola numa célula vazia ou em uma que já contenha argolas de menor tamanho. Alternativamente, o jogador pode escolher mover um conjunto de argolas já no tabuleiro para outra célula. A seguir, detalhamos as regras específicas para cada uma dessas ações:

Posicionar uma peça no tabuleiro: esse movimento envolve retirar uma peça da área reservada ao jogador e inseri-la em uma posição específica no tabuleiro. Este posicionamento só pode ocorrer em casas desocupadas ou onde haja uma peça de menor tamanho, situação na qual ambas as peças (a menor e a inserida) ocuparão a mesma casa. Esse movimento só é permitido enquanto houver peças disponíveis na área de peças do jogador, e deve ser realizado somente com as peças que lá se encontram.

Mover peça no tabuleiro: o jogador move inteiramente um conjunto de argolas de uma célula a outra horizontalmente, verticalmente ou diagonalmente, desde que a célula de destino esteja vazia e o conjunto de argolas ao ser movido, não “pule por cima” de argolas em outras posições.

Encerramento da Partida

O jogo encerra quando algum dos jogadores formar uma linha de 4 conjuntos iguais (com os mesmos tipos de argolas) na horizontal, vertical ou diagonal do tabuleiro.