Projeto Ultimate Dots and Boxes

Especificação de Requisitos de Software

Versão 1.0

Versão	Autor(es)	Data	Ação
1.0	Thayssa Yasmin Alcantara de Godoi Gabriel Gomes da Costa Gonçalves Diego Martins Meditsch	11/09/2024	Estabelecimento dos requisitos

Tabela de Conteúdos

1. Introdução

1.1. Objetivo

Desenvolvimento de um programa distribuído para efetuar partidas do jogo Ultimate Dots and Boxes em formato usuário contra usuário.

1.2. Definições, abreviaturas

Regras do jogo: ver apêndice

1.3. Referências:

Apresentação das regras de parte do jogo (*dots and boxes*) (video do canal divertudo):

https://www.youtube.com/watch?v=KK_12KOj5iA

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 – Iniciar programa: ao ser executado, o programa deve apresentar na interface o tabuleiro do jogo em seu estado inicial (todas os noves jogos dots and boxes vazios, isto é, sem nenhuma marcação) e solicitar o nome do jogador. Após isso, deve solicitar conexão com DOG Server (utilizando os recursos de DOG). O resultado da tentativa de conexão deve ser informado ao usuário. Apenas em caso de conexão bem sucedida as demais funcionalidades estarão habilitadas. No caso de conexão mal sucedida, a única alternativa deve ser encerrar o programa;

Requisito funcional 2 – Iniciar jogo: 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 (clicar no tabuleiro). 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 – Restaurar estado inicial: o programa deve apresentar a opção de menu “restaurar estado inicial” para levar o programa ao seu seu estado inicial, isto é, sem partida em andamento e com o tabuleiro em seu estado inicial. Esta funcionalidade só deve estar habilitada se o programa estiver com uma partida finalizada;

Requisito funcional 4 – Clicar no tabuleiro: O programa deve permitir a um jogador habilitado ligar dois pontinhos no tabuleiro, segurando o botão do mouse de um até o outro, gerando um risco. O risco só será colocado definitivamente quando o jogador em questão selecionar uma posição de risco vazia do tabuleiro correto (para determinar o tabuleiro correto ver apêndice). Se não respeitar as condições, será considerada uma ação inválida e o jogador terá que tentar novamente. Após o jogador conseguir colocar um risquinho, será verificado se o jogador completou uma caixinha, caso este no qual jogará de novo, e posteriormente se há vencedor em algumas das esferas do jogo (para saber como se determina o vencedor de ambas as esferas do jogo, ver apêndice) se o jogo grande se encerra naquela jogada, o jogo encerra. Esta funcionalidade só estará habilitada quando estiver em andamento um jogo e for a vez do jogador.

Requisito funcional 5 – Receber determinação 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 jogo', 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 6 – Receber jogada: o programa deve poder receber uma jogada do adversário, enviada por Dog Server, quando for a vez do adversário do jogador local. A jogada recebida deve ser um lance regular e conter as informações especificadas para o envio de jogada no 'Requisito funcional 4 – Clicar no tabuleiro'. O programa deve atualizar o(s) risco(s) que foi(ram) marcado(s). Após isso, deve-se avaliar o encerramento de partida. No caso de encerramento de partida, deve ser notificado o nome do jogador vencedor; no caso de não encerramento, deve ser habilitado o jogador local, para que possa proceder a seu lance;

Requisito funcional 7 – Receber notificação de abandono: 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. Além disso, o jogador que abandonou será considerado derrotado e a vitória será do jogador que recebeu a notificação de abandono.

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 o esboço da imagem abaixo.

Apêndice: Regras Ultimate Dots and Boxes

A ideia do jogo é misturar dois jogos bastante comuns e simples, de maneira que apareça nessa junção uma complexidade interessante tanto para os jogadores quanto para lógica computacional necessária para implementá-la. Os dois jogos comuns são jogo da velha - que dispensa apresentações - e *dots and boxes*, exemplificado abaixo.

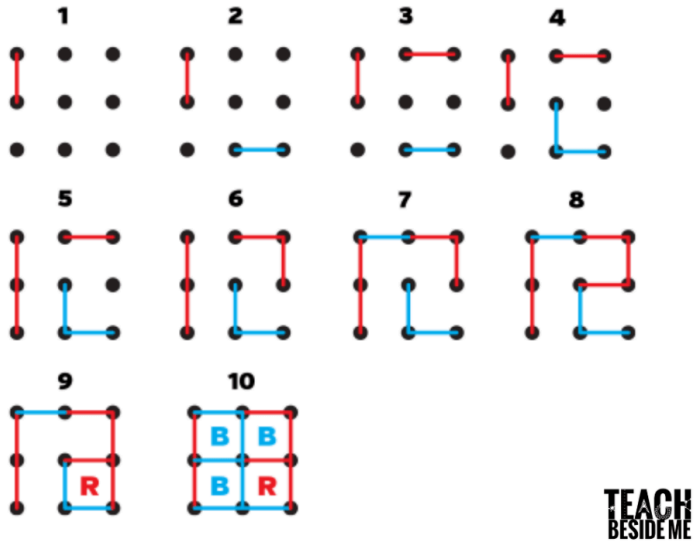

Figura 1 - *Dots and Boxes*

No *dots and boxes*, o jogador deve conectar os pontos entre si (quando não estão ainda conectados), de forma a fechar uma caixinha. Se um jogador consegue fechar uma caixinha em sua jogada, ele pode jogar novamente. O jogador que tiver a maioria das caixinhas fechadas é o vencedor.

No jogo da velha, por outro lado, o objetivo é alinhar 3 posições de uma matriz por 3x3, de maneira que um mesmo jogador tenha suas marcações em uma linha de qualquer orientação, seja ela vertical, horizontal ou diagonal. Os jogadores tem jogadas alternadas, onde posicionam um de seus símbolos em uma posição vazia da matriz, e o jogo se encerra quando há o alinhamento supracitado, ou quando não há mais espaços vazios - gerando empate.

O resultado da mistura desses dois jogos é um tabuleiro que tem uma grade 3x3 de partidas de *dots and boxes*, onde cada partida tem uma grade de 4x4 pontos, como na imagem abaixo.

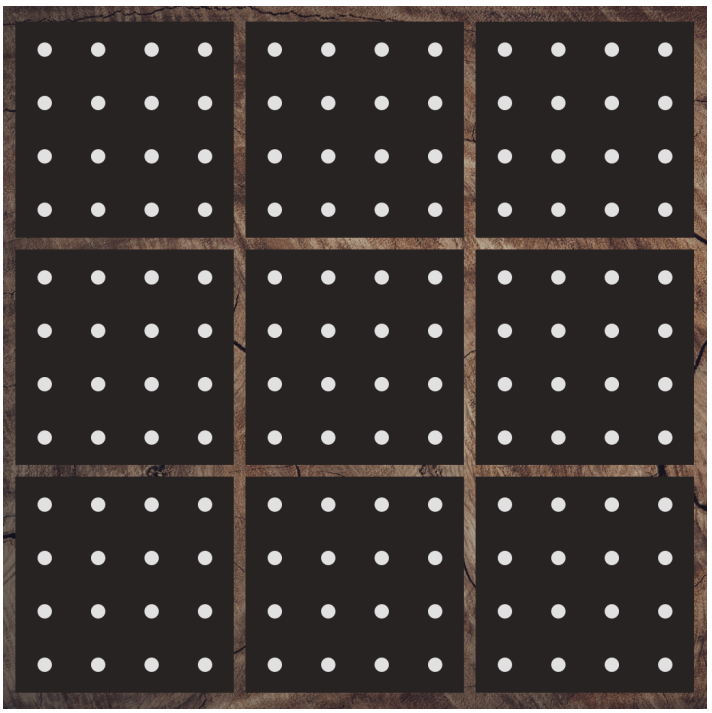

Figura 2 - Mockup de UI do nosso jogo

Para fins de notação, vamos chamar cada matriz pequena de *dots and boxes* de micro jogo 1, micro jogo 2, …, microjogo 9, na ordem que estão apresentados, enquanto que o macro jogo será o tabuleiro completo de *Ultimate Dots and Boxes*.

A partir daí os jogadores começam a intercalar suas jogadas, jogando apenas no quadrado central da matriz grande (microjogo 5), como se fosse uma partida normal de *dots and boxes*. Cada jogador uma vez, mas pode jogar de novo - no mesmo micro jogo - se completar um quadrado. A partir do momento que o primeiro jogador completa um quadrado - depois de sua jogada extra - o outro jogador deve fazer a sua jogada no quadrado que é equivalente ao completado no dentro do micro jogo, mas na escala maior.

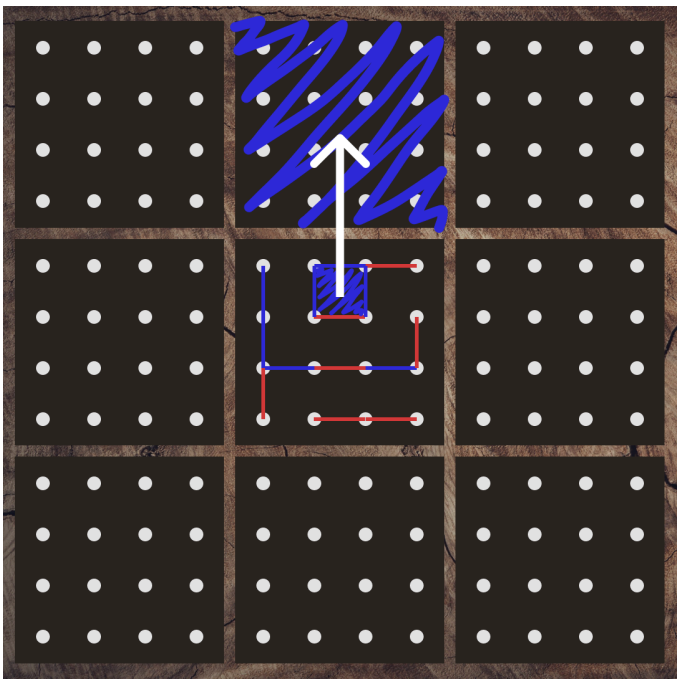

Figura 3 - Quando o quadrado superior central **de um micro jogo** é preenchido, o jogador seguinte deve **jogar no micro jogo** que ocupa a mesma posição da matriz grande, no caso da imagem, o micro jogo 2.

A lógica do jogo se mantém essa ao longo das jogadas, variando em qual micro jogo os jogadores se encontram, até que um dos jogadores consiga preencher 5 quadrados de um mesmo microjogo. Nesse momento, o jogador que faz isso ganha aquele micro jogo, que fica portanto marcado como ganho por esse jogador.

Para ganhar a totalidade do jogo, o jogador deve ganhar 3 microjogos em linha (vertical, horizontal ou diagonal), **ou** ganhar quaisquer 5 microjogos.

Abaixo, segue uma possível sequência de jogadas em uma partida, onde o jogador azul ganha o jogo após vencer 3 microjogos em linha.

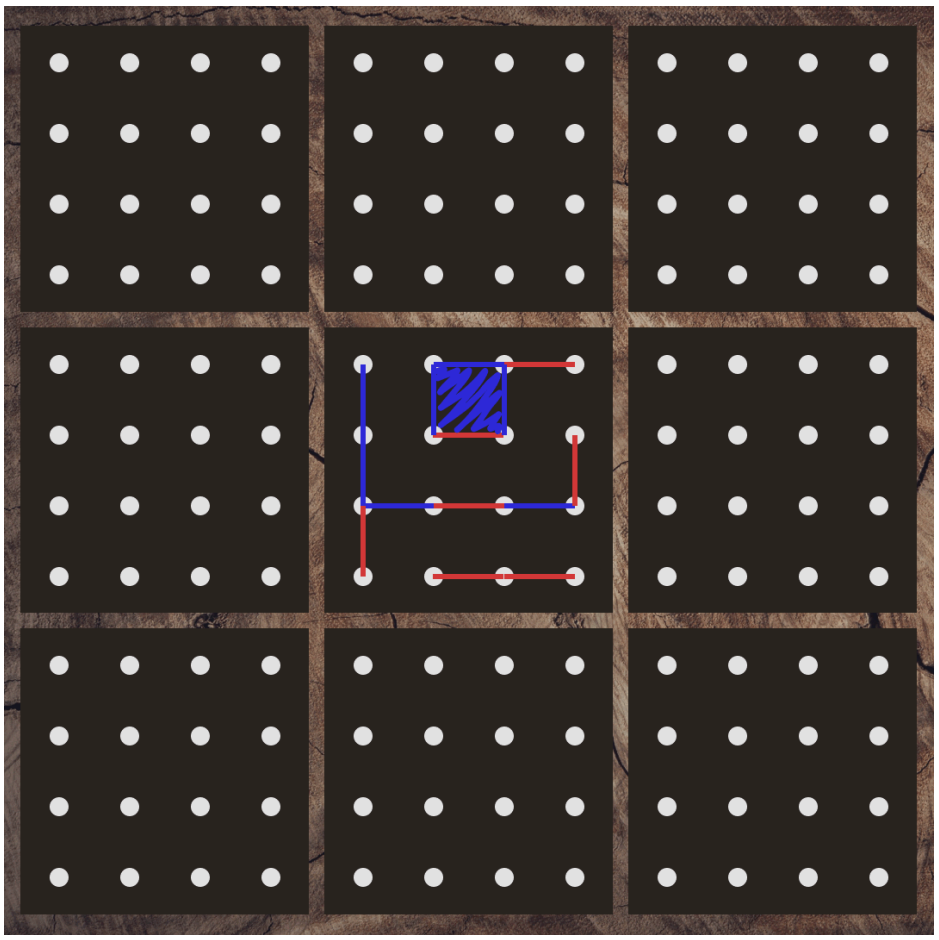

Figura 4 - Partida simulada de Ultimate Dots and Boxes

No demais, vale ressaltar o caso em que um dos jogadores completa um quadrado em um micro jogo que direciona a jogada para um microjogo já completo, que seria impossível seguindo a lógica já descrita. Neste caso, o jogador seguinte pode escolher qualquer um dos microjogos incompletos para fazer sua jogada, seguindo a partir daí a lógica normal do jogo.