

Projeto Duelo Binário

Especificação de Requisitos de Software

Versão 1.0
09/04/2024

Versão	Autor(es)	Data	Ação
1.0	Henrique Afonso Alves de Lima Rita Louro Barbosa Thiago Kenzo Takahashi	17/08/2024	Estabelecimento dos requisitos
2.0	Henrique Afonso Alves de Lima Rita Louro Barbosa Thiago Kenzo Takahashi	30/05/2024	Atualização das regras de pontuação

Conteúdo:

- 1. Introdução
- 2. Visão geral
- 3. Requisitos de software
- Apêndice: Regras Duelo Binário

1. Introdução

1.1 Objetivo

Desenvolvimento de um programa distribuído que suporte a disputa de partidas do jogo autoral Duelo Binário na modalidade usuário contra usuário.

1.2 Definições, abreviaturas

Regras do jogo: ver apêndice

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 (vazio) e os demais campos de informação da interface vazios. Em seguida, deve 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 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. Em caso positivo para início de partida, a interface do programa deve ser atualizada com as informações recebidas, incluindo o nome dos 2 jogadores no placar. O jogador que executa essa funcionalidade será o jogador inicial, então, 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. A interface também deverá indicar, em um campo específico, a cor do jogador com a vez (o que iniciará a partida);

Requisito funcional 3 – Restaurar estado inicial:

O programa deve apresentar a opção de menu “Novo jogo” para levar o programa ao seu estado inicial, isto é, sem partida em andamento, com o tabuleiro em seu estado inicial (vazio) e os demais campos de informação da interface vazios. Esta funcionalidade só deve estar habilitada se o programa estiver com uma partida finalizada;

Requisito funcional 4 – Inserir dígito:

O programa deve permitir a um jogador habilitado inserir um dígito ‘0’ ou um dígito ‘1’ em alguma posição de casa branca vazia no tabuleiro. A escolha do dígito e da casa se dará pela interação com o mouse. Se o jogador clicar com o mouse no clique direito em cima de uma casa branca vazia, a casa receberá o dígito ‘1’ e se clicar com o clique esquerdo, a casa receberá o dígito ‘0’. A casa preenchida deve estar visualmente destacada, com o dígito recém preenchido escrito em azul na casa. Se a ação for executada após o jogador já ter anteriormente, na mesma jogada, inserido um dígito em outra casa, o dígito escrito em azul na casa anterior deve ser removido do tabuleiro e o novo dígito deve ser inserido no local selecionado pelo jogador. Se a ação for realizada sobre uma casa de cor diferente, ou sobre uma casa já preenchida em outras jogadas, nada deve mudar no estado do jogo e da interface, pois estas casas não estarão habilitadas para mudança. Se a ação for realizada sobre uma casa já preenchida nesta mesma jogada (casa com dígito em azul), o dígito correspondente ao clique deve substituir o dígito escrito anteriormente.

Requisito funcional 5 – Enviar jogada:

O programa deve apresentar a opção de se botão “Enviar jogada” que determina o fim da jogada do jogador habilitado. O programa deve avaliar se algum dígito já foi inserido em alguma casa pelo jogador da vez (Ver requisito funcional 4). Caso a inserção (jogada) ainda não tenha sido feita, a tentativa de “Enviar jogada” não deve produzir nenhum efeito no jogo (botão desabilitado) e um aviso de “Insira um dígito no tabuleiro” deve ser mostrado ao jogador. Caso o requisito de dígito inserido tenha sido cumprido, o programa deverá verificar, para cada casa preta atrelada à casa branca em que o dígito foi inserido, se a jogada levou ao preenchimento final das 4 casas brancas atreladas àquela casa preta. Em caso positivo, o número decimal correspondente ao binário formado por esses dígitos é automaticamente inserido na casa preta (vide regras do jogo) e a casa preta adquire a cor que representa o jogador que realizou a jogada. Além disso, a devida pontuação será somada aos pontos deste jogador (vide regras do jogo) e o placar da interface gráfica é atualizado. Caso a jogada não gere a conclusão do preenchimento de nenhum número, não há mudança adicional no estado do tabuleiro e nem do placar. Em seguida, o programa deve avaliar o encerramento de partida (vide regras do jogo). No caso de encerramento de partida, deve ser notificado o nome do jogador vencedor (Vide regras do jogo) e a pontuação final de cada jogador; Independentemente do encerramento de partida, as ações

do jogador local sobre o tabuleiro deverão ser desabilitadas. Por fim, o programa deve enviar a jogada ao adversário (utilizando os recursos de DOG). A jogada enviada deve conter o dígito e a posição do tabuleiro em que o dígito foi inserido.. No caso de não encerramento, o programa fica no aguardo de jogada do adversário (ver Requisito funcional 7) ou de notificação de abandono (ver Requisito funcional 8);

Requisito funcional 6 – 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. O jogador que executou a solicitação de início de partida será o jogador inicial. Logo, o jogador local tem sua interface desabilitada, para aguardo de recebimento de jogada.

Requisito funcional 7 – 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 conter as informações especificadas para o envio de jogada no ‘Requisito funcional 5 – Enviar jogada’. O programa deve atualizar o tabuleiro com o dígito inserido pelo adversário. Caso a jogada do adversário tenha gerado a inserção de algum decimal em uma casa preta, tal atualização deve ser feita no tabuleiro local, de acordo com a posição, o número e a cor de modificação recebidos, bem como a nova pontuação do jogador adversário deve ser atualizada no placar. Após isso, deve-se avaliar o encerramento de partida. No caso de encerramento de partida (vide regras do jogo), 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 8 – 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.

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 das imagens abaixo.

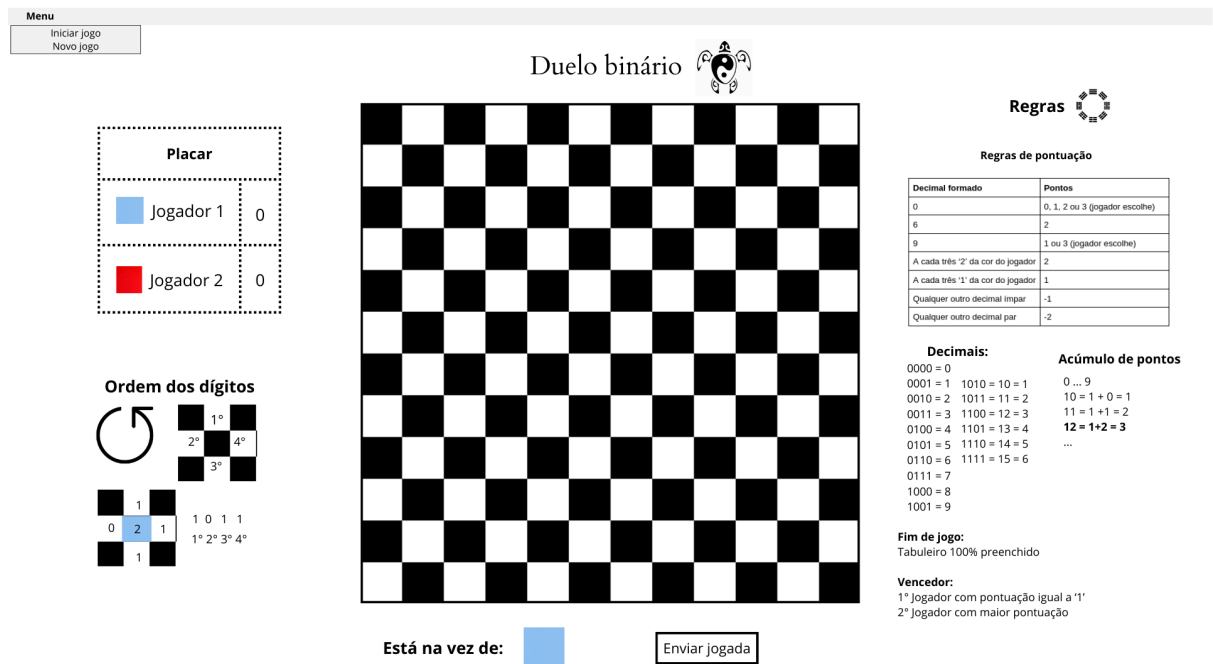


Imagem 1: Interface em estado inicial, antes da partida iniciar

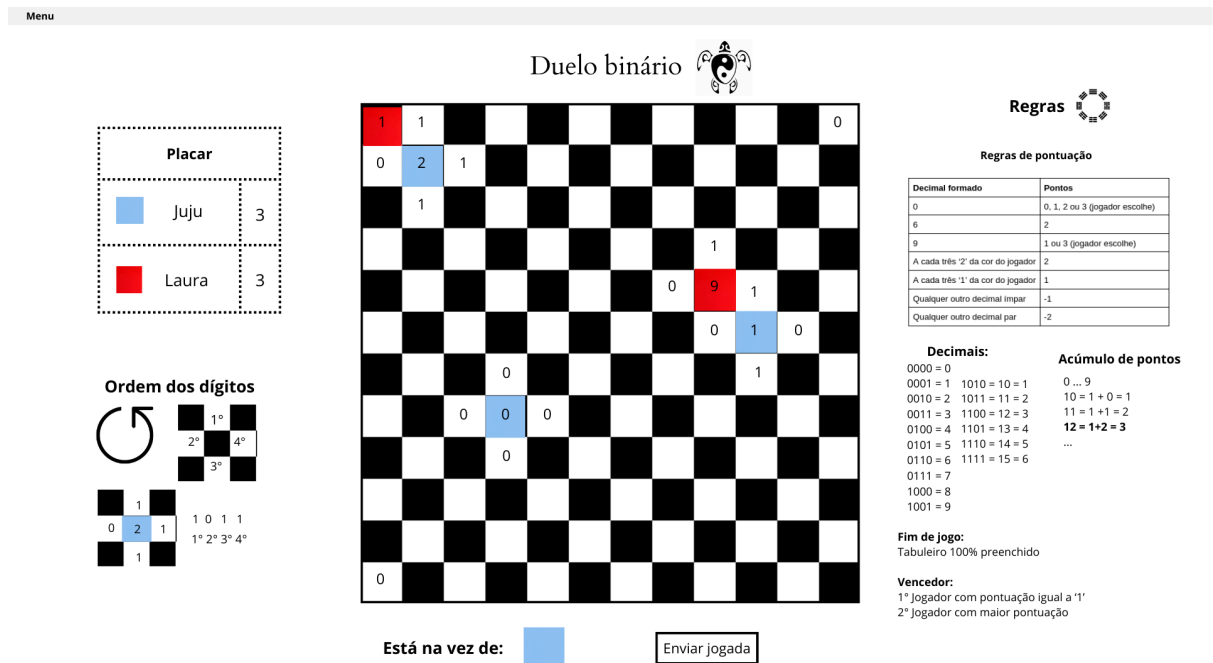


Imagem 2: Exemplo de interface durante uma partida.

Apêndice: Regras do jogo Duelo Binário:

O jogo é um desafio lógico entre 2 jogadores envolvendo a disposição e criação de números binários em um tabuleiro xadrez de tamanho 12 x 12. O objetivo dos jogadores é preencher o tabuleiro de forma a finalizar a partida com pontuação igual a '1' ou acumular o máximo de pontos possíveis antes do fim do jogo (vide regra de acúmulo de pontos).

Elementos principais do jogo:

- **Placar:** Exibe o nome e a pontuação atual dos jogadores


Placar		
	Jogador 1	0
	Jogador 2	0

Imagem 3: Placar

- **Área de tabuleiro:** tabuleiro no qual serão feitas as jogadas (elemento central do jogo)

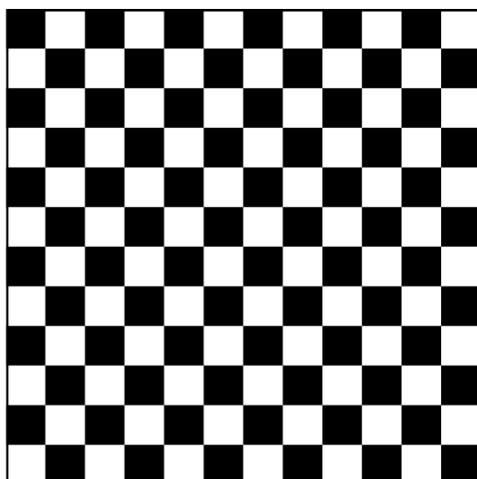


Imagem 4: Tabuleiro

- **Indicador de jogador ativo:** Indica o jogador que deverá executar a jogada naquele momento

Está na vez de:



Imagem 5: Campo indicativo de jogador da vez

- **Botão de enviar jogada:** Confirma a escolha de jogada do jogador

Enviar jogada

Imagem 6: Botão

Significado das casas do tabuleiro:

As 4 casas brancas que rodeiam uma casa preta correspondem a 4 posições de um número binário. A casa acima corresponde à posição do bit mais significativo. A casa à esquerda corresponde ao 2º bit mais significativo. A casa abaixo corresponde ao 3º bit mais significativo. A casa à direita corresponde ao 4º bit, o bit menos significativo do número binário (Imagem 6).

No caso de casas localizadas nas extremidades do tabuleiro, seu complemento são as casas das extremidades opostas (ver imagem 7).

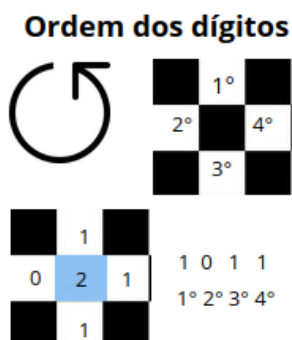


Imagem 6: Visualização da relação posição da casa x posição do dígito no número binário

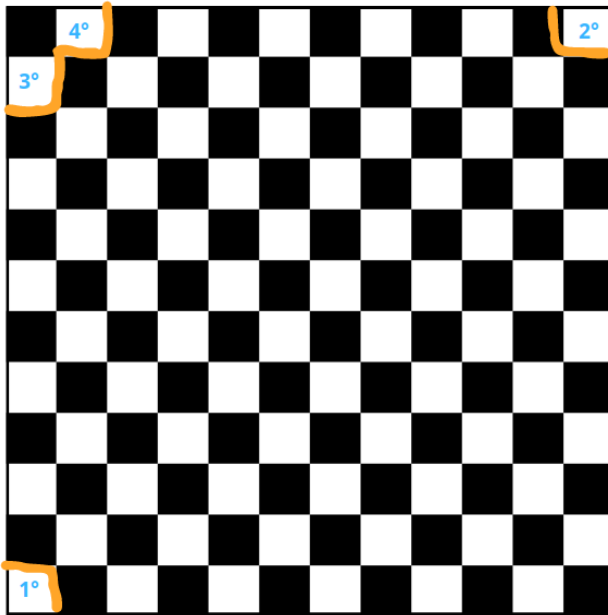


Imagem 7: Visualização da relação entre casas nas extremidades.

Estado inicial do tabuleiro:

- O tabuleiro começa com todas as casas vazias.

Dinâmica das jogadas:

Inserir dígitos: Em cada jogada, o jogador da vez deve escolher entre inserir o dígito '0' ou o dígito '1' em qualquer casa branca que esteja vazia.

A escolha será feita com o uso do mouse: O botão direito do mouse insere o dígito '1' e o botão esquerdo do mouse insere o dígito '0'.

Após a escolha, deverá submeter a jogada por meio do botão de "Enviar jogada".

Consequências da inserção do dígito: Se a jogada levar o preenchimento final das 4 casas brancas ao redor de uma casa preta, a casa preta adquire a cor que representa o jogador que realizou a jogada e o número decimal correspondente ao binário formado por esses dígitos é automaticamente inserido nesta casa (vide regra de formação do decimal). Além disso, a devida pontuação será somada aos pontos deste jogador (vide regras de pontuação).

Jogadas que não gerem a conclusão do preenchimento de nenhum número não serão pontuadas.

Exemplo de tabuleiro parcialmente preenchido e pontuação correspondente:

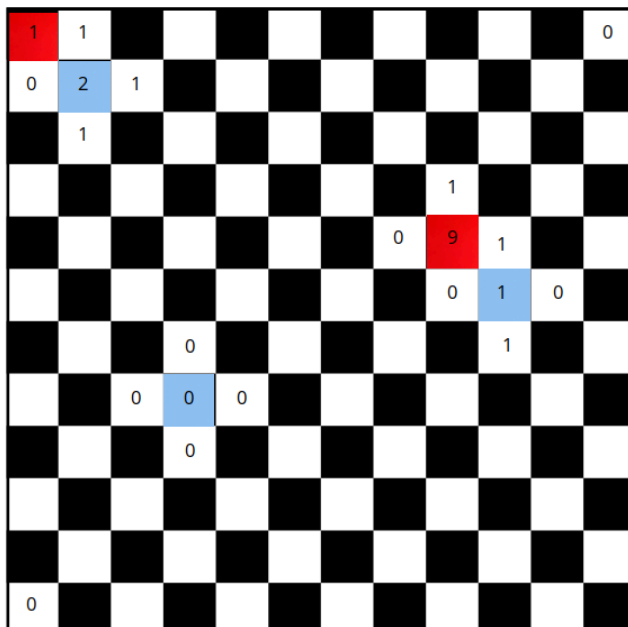


Imagem 8: Exemplo de preenchimento do tabuleiro



Placar		
	Juju	3
	Laura	3

Imagem 9: Exemplo de placar correspondente ao estado atual (Imagem 8) do jogo.

Encerramento da partida:

O jogo acaba quando todas as casas brancas forem preenchidas com dígitos binários.

Vence o jogador que:

- 1º Finalizar jogo com **pontuação igual a 1**.
- 2º Finalizar com **maior pontuação**

Regras de pontuação:

Decimal formado	Pontos
0	Jogador tem pontuação zerada
6	2 pontos
9	3 pontos
A cada três '2' da cor do jogador	2 pontos
A cada três '1' da cor do jogador	1 ponto
Qualquer outro decimal ímpar	Penalidade: -1 pontos
Qualquer outro decimal par	Penalidade: -2 pontos

Detalhamentos: Jogadas que gerem os decimais '2' e '1' só serão pontuadas a cada 3 casas com esse decimal preenchidas com a cor do jogador.

Em caso de penalidade, os pontos são decrescidos **até a marca de '0' pontos**, sendo esta a **pontuação mínima** que um jogador pode atingir.

Regra de formação do decimal:

Os decimais a serem postos no tabuleiro e utilizados para determinação da pontuação recebida pelo jogador variam entre 0 e 9. Para isso, eles são formados seguindo a seguinte lógica:

Número binário	Conversão para decimal	Lógica de soma dos dígitos	Decimal inserido no tabuleiro e utilizado para atribuição da pontuação
0000	0		0
0001	1		1
0010	2		2
0011	3		3
0100	4		4
0101	5		5
0110	6		6
0111	7		7

1000	8		8
1001	9		9
1010	10	$1 + 0 = 1$	1
1011	11	$1 + 1 = 2$	2
1100	12	$1 + 2 = 3$	3
1101	13	$1 + 3 = 4$	4
1110	14	$1 + 4 = 5$	5
1111	15	$1 + 5 = 6$	6

Regra de acúmulo de pontos:

Os pontos acumulados de um jogador podem variar entre 0 e 9, correspondendo sempre ao somatório dos dígitos da pontuação cada vez que ela ultrapassa 9 pontos. Sendo assim, a pontuação máxima que um jogador pode receber é 9.

Exemplo 1:

Um jogador que possui 8 pontos acumulados faz uma jogada e recebe 2 pontos;

$8 + 2 = 10$. Somando os dígitos do resultado, temos: $1 + 0 = 1$.

Logo, a pontuação total do jogador passa a ser 1.

Exemplo 2:

Um jogador que possui 9 pontos acumulados faz uma jogada e recebe 3 pontos.

$9 + 3 = 12$. Somando os dígitos do resultado, temos: $1 + 2 = 3$.

Logo, a pontuação total do jogador passa a ser 3.