

Projeto Hobbes – Um pega-pega entre dois reis

Especificações de Requisitos de Software

REQ_PR_hobbes_2021nov19

Versão 1.0

19/11/2021

Versão 1.0	Autor(es)	Data	Ação
1.0	Eduardo, José Daniel e Raquel	19/11/2021	Estabelecimento dos requisitos
2.0	Eduardo, José Daniel E Raquel	08/12/2021	Reformulação do documento
3.0	Eduardo, José Daniel E Raquel	27/01/2022	Inserindo imagens descritivas das regras do jogo
4.0	Eduardo, José Daniel E Raquel	31/01/2022	Melhor estruturando regras do jogo

Sumário

1	INTRODUCAO	3
1.1	OBJETIVO DO DESENVOLVIMENTO	3
1.2	REGRAS DO JOGO	3
1.3	REFERÊNCIAS.....	20
2	VISÃO GERAL DO SISTEMA	20
2.1	ARQUITETURA DE SOFTWARE.....	20
2.2	PREMISSAS DO DESENVOLVIMENTO	20
3	REQUISITOS DA APLICAÇÃO.....	20
3.1	REGRAS DE NEGÓCIO.....	20
3.2	REQUISITOS FUNCIONAIS.....	20
3.3	REQUISITOS NÃO FUNCIONAIS	21

1 INTRODUCAO

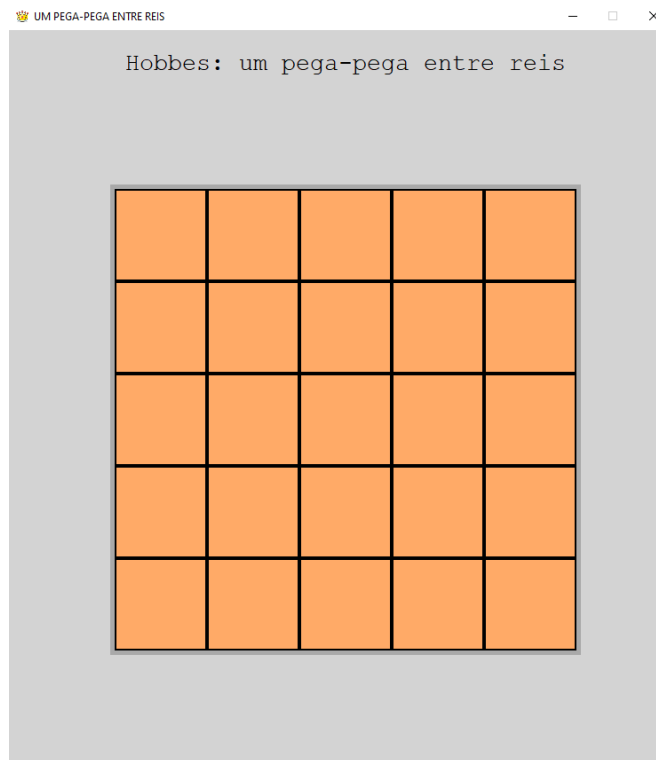
1.1 OBJETIVO DO DESENVOLVIMENTO

Desenvolver um programa que suporte a disputa de partidas do jogo Hobbes – um pega-pega entre dois reis na modalidade usuário contra usuário em um mesmo computador.

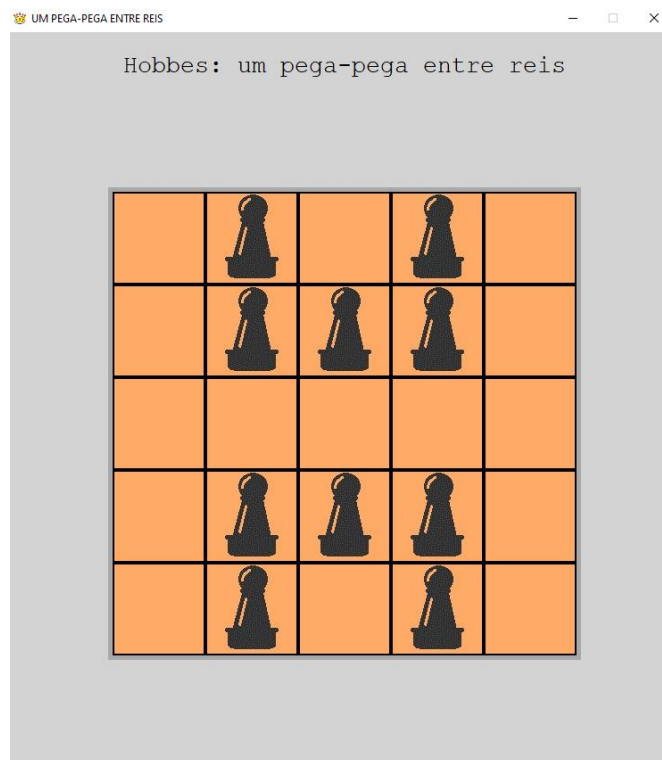
1.2 REGRAS DO JOGO

Premissas do jogo:

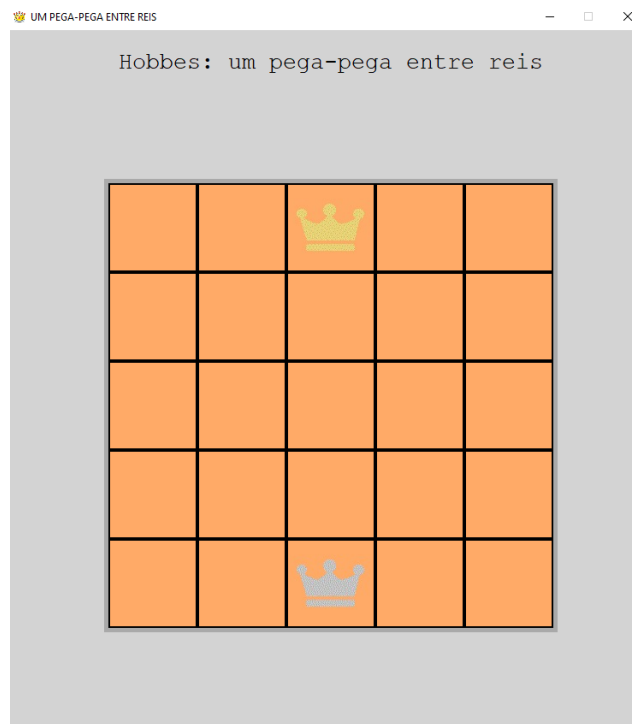
- Há 25 casas (5x5);



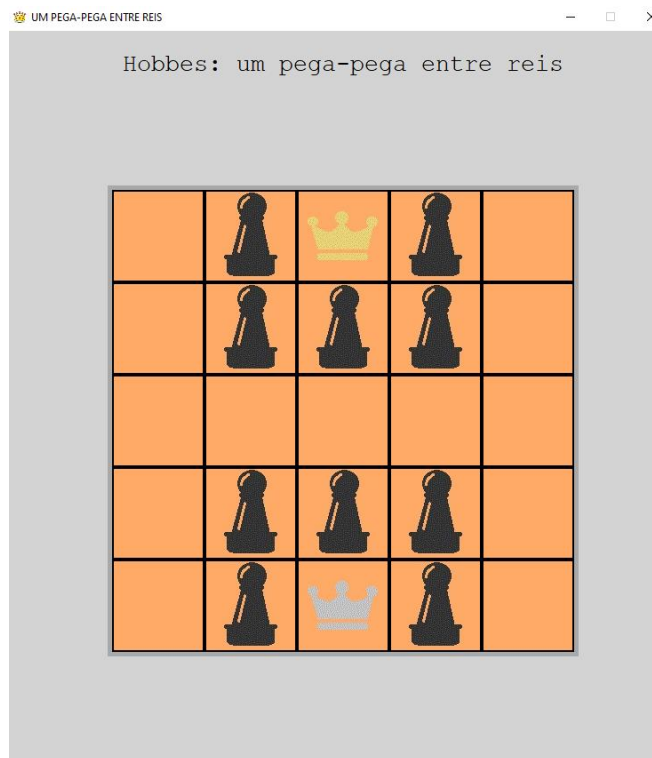
- Há 10 peças neutras compartilhadas;



- Há 2 reis;
 - Primeiramente cada rei fica na peça central dos extremos;



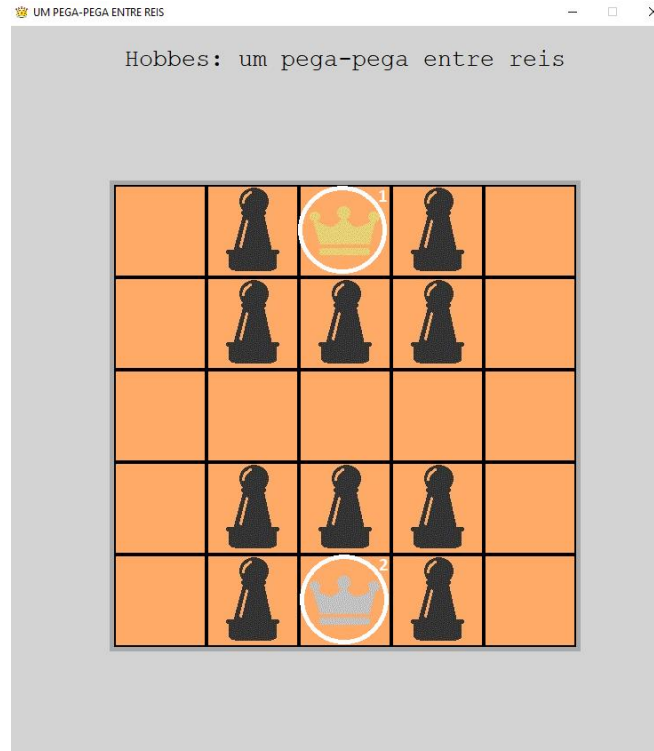
- Para cada rei, há 5 peças neutras em sua volta;



- São 2 movimentos (1 movimento opcional e 1 obrigatório);
- Movimento opcional: mover-se por casas livres na horizontal ou vertical;
- Movimento obrigatório: empurrar ou puxar uma peça neutra adjacente 1 ou 2 casas;
- Objetivo: capturar o rei adversário (ficando adjacente a ele) ou bloqueá-lo.

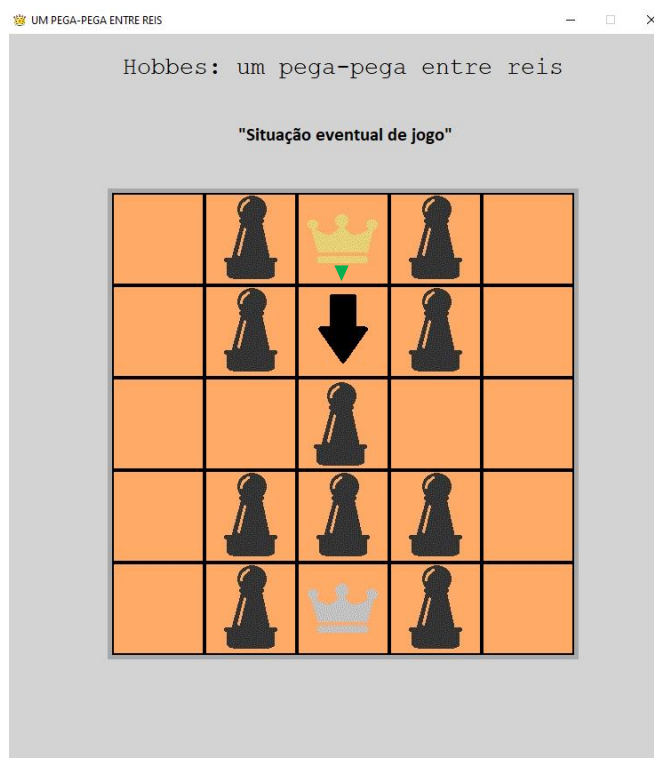
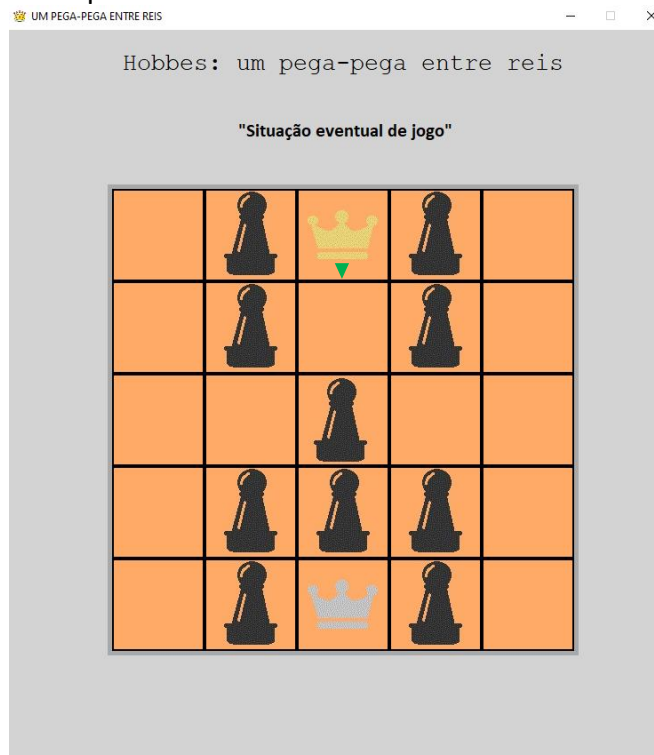
Como jogar:

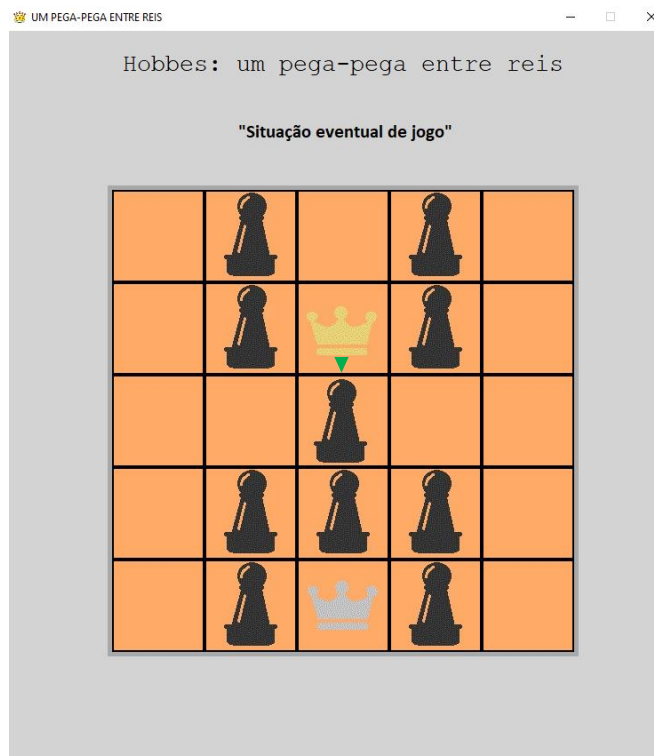
1. O jogo será realizado com dois jogadores distintos;



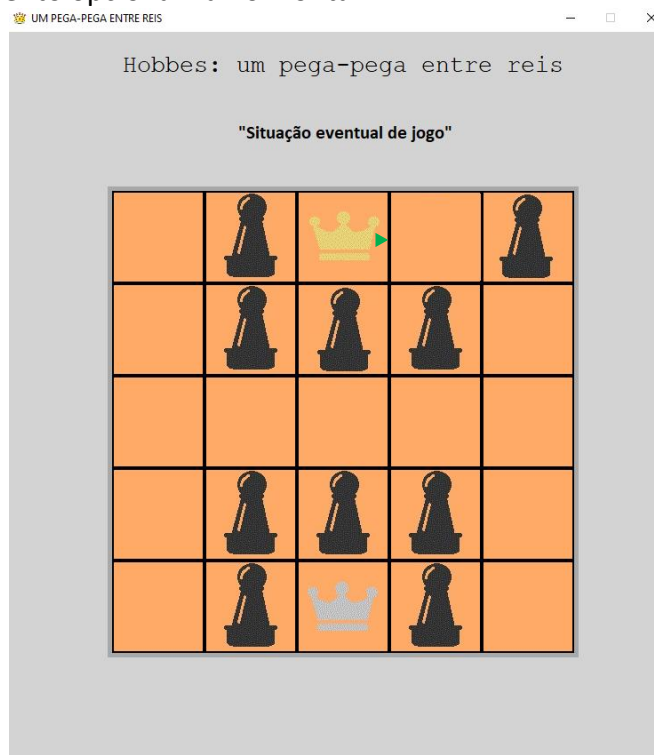
2. Quando for a vez do jogador, primeiramente ele poderá escolher se fará uma jogada opcional (mover-se por casas livres na horizontal ou vertical), e, após isso, terá que fazer uma jogada obrigatória (empurrar ou puxar peões);
 - a. Na primeira jogada de cada jogador só é possível realizar a jogada obrigatória, pois não há regisões livres para ele se mover.
3. Na sua jogada opcional, o rei pode se mover para qualquer lugar livre do tabuleiro, conquanto que não seja na diagonal, e nem pule peões;

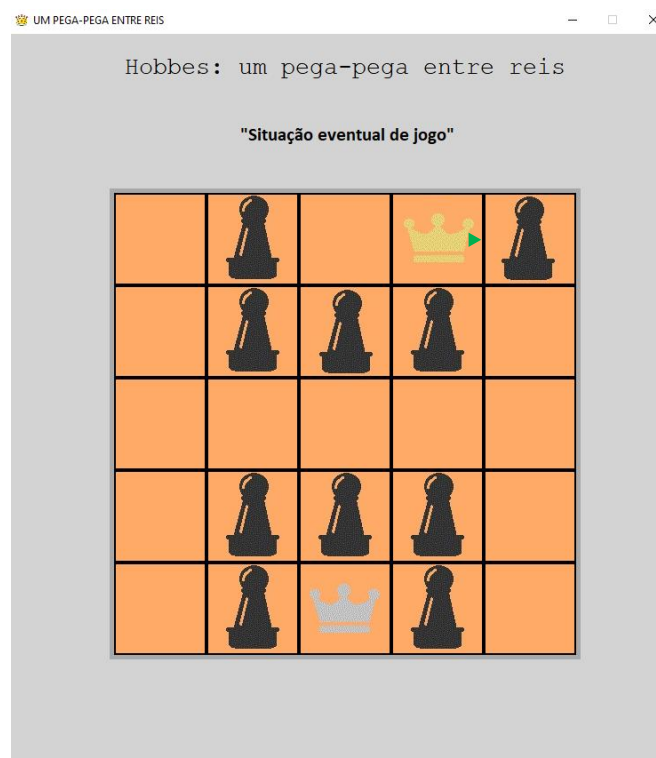
1. Movimento opcional na vertical.





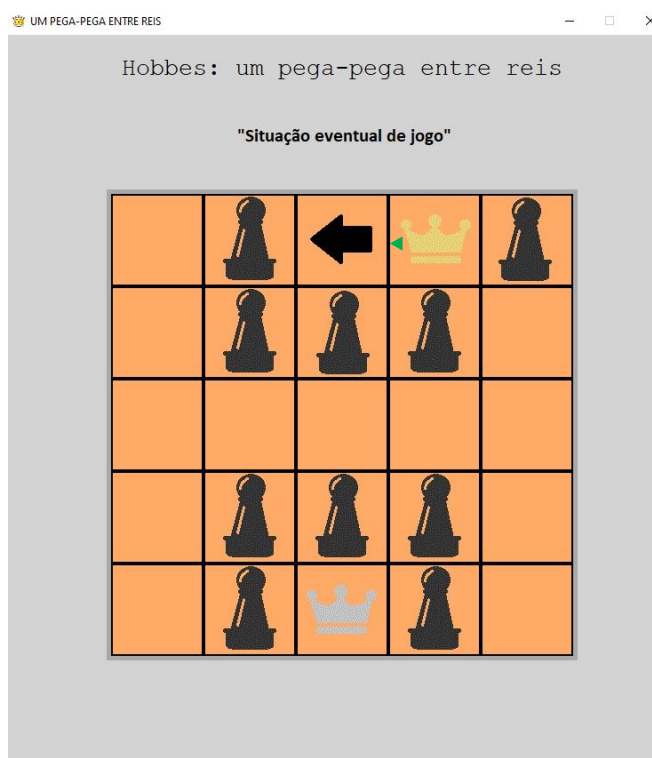
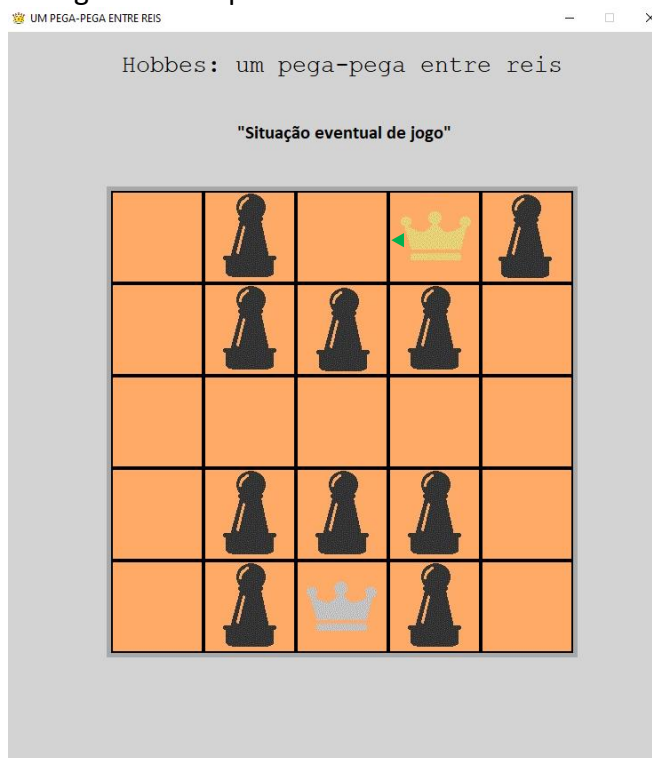
2. Movimento opcional na horizontal.

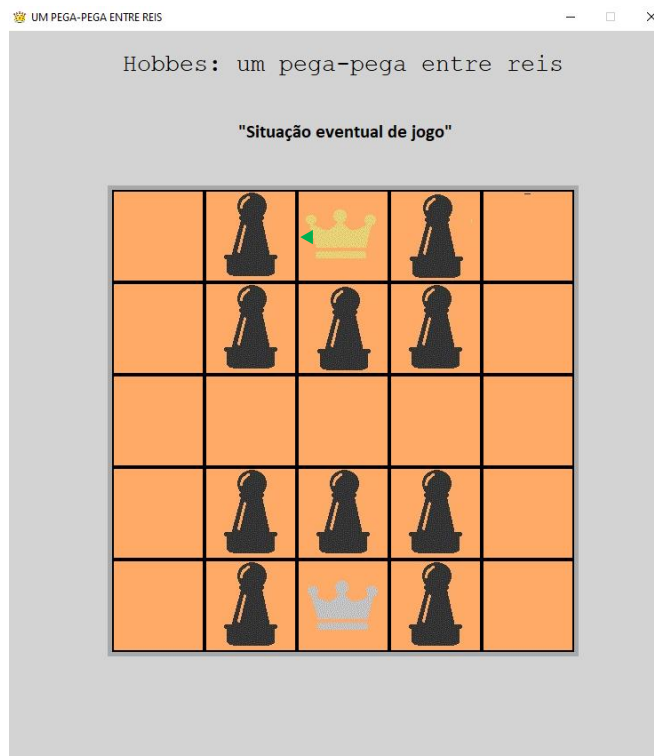




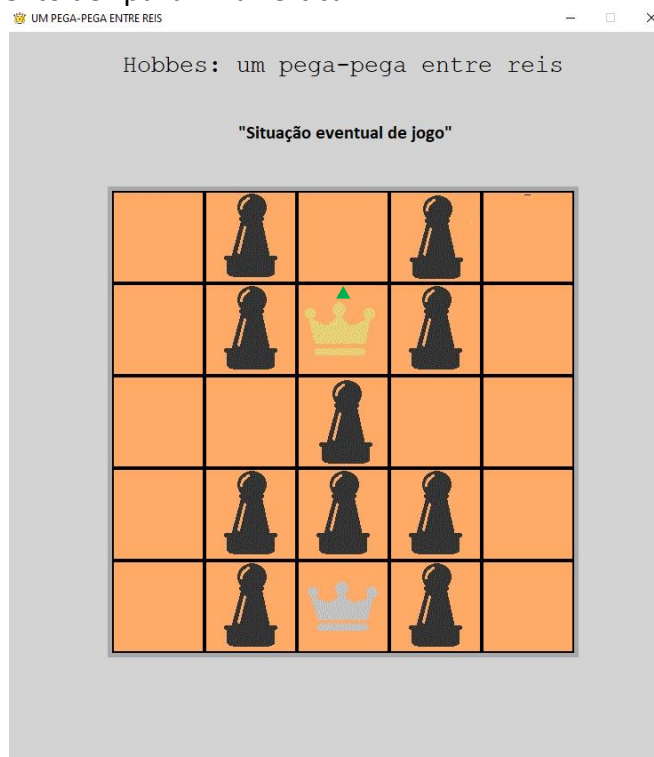
- Em sua jogada obrigatória, o jogador poderá puxar ou empurrar apenas um peão por jogada. E só poderá puxar ou empurrar determinado peão caso estiver adjacente a ele (contudo, não em sua diagonal);

1. Movimento obrigatório de “puxar” na horizontal.



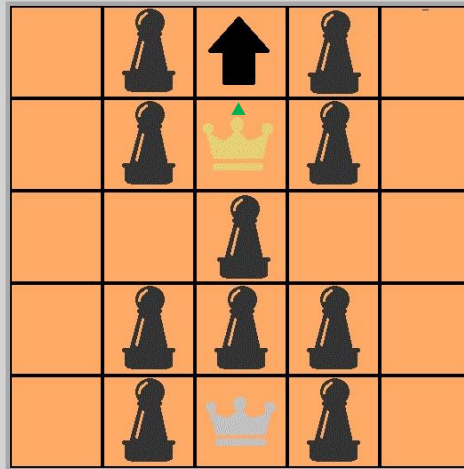


2. Movimento de “puxar” na vertical.



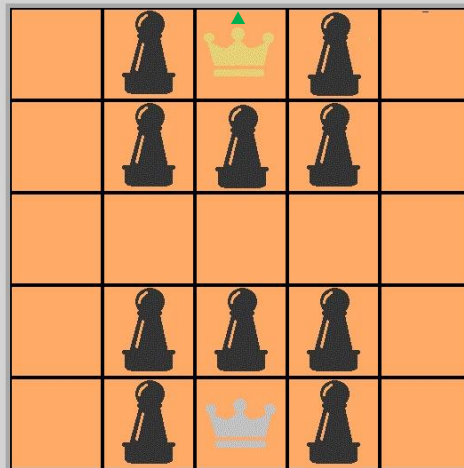
Hobbes: um pega-pega entre reis

"Situação eventual de jogo"

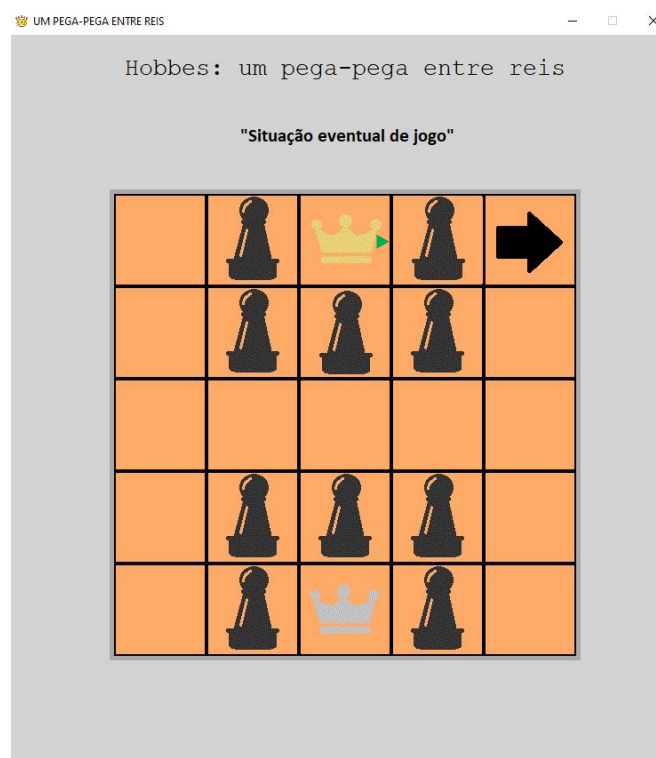
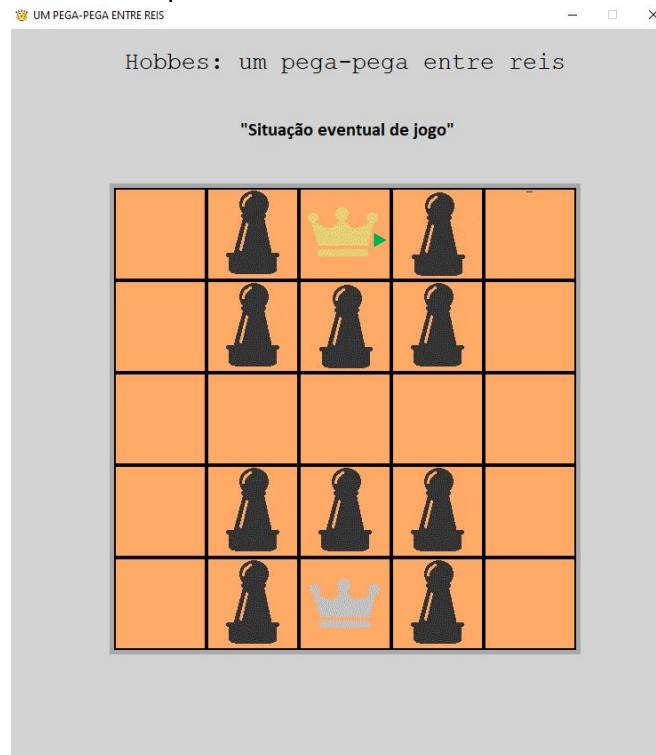


Hobbes: um pega-pega entre reis

"Situação eventual de jogo"

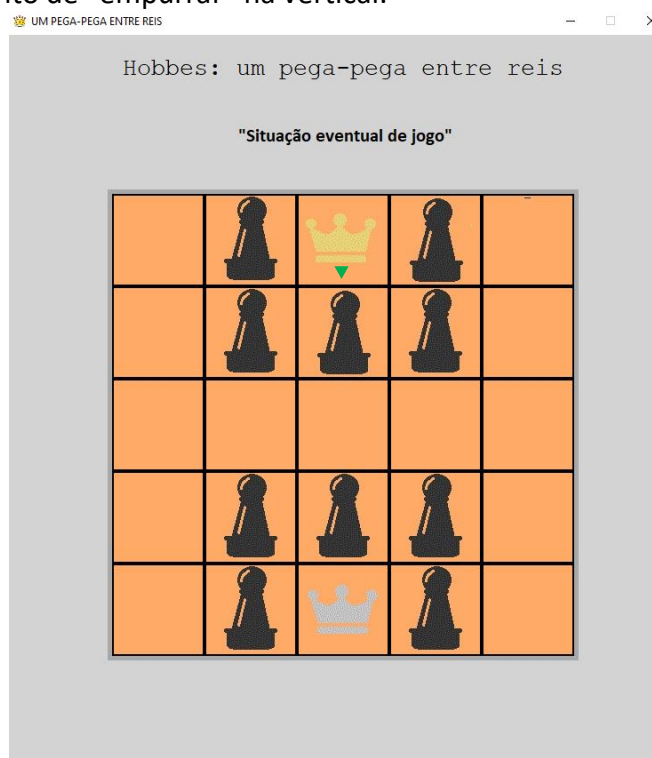


3. Movimento de “empurrar” na horizontal.



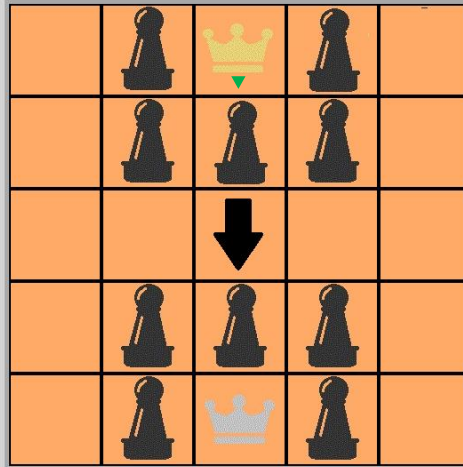


4. Movimento de “empurrar” na vertical.



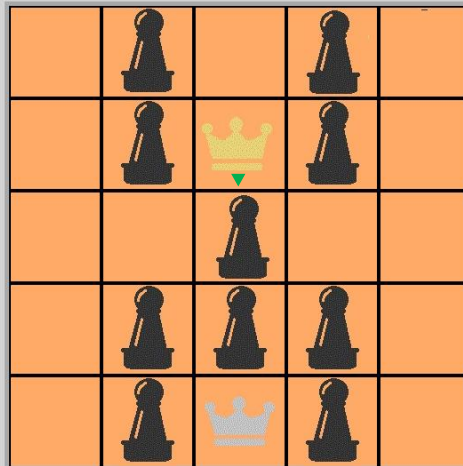
Hobbes: um pega-pega entre reis

"Situação eventual de jogo"

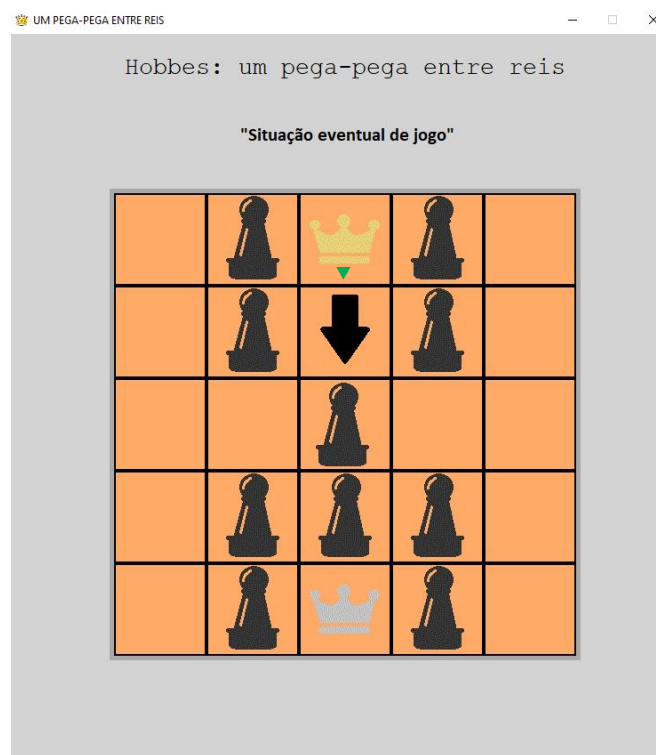
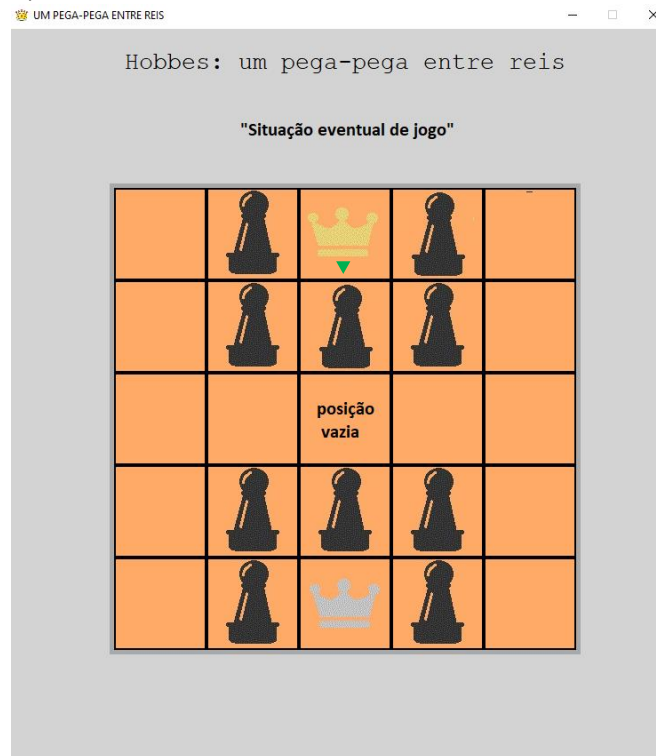


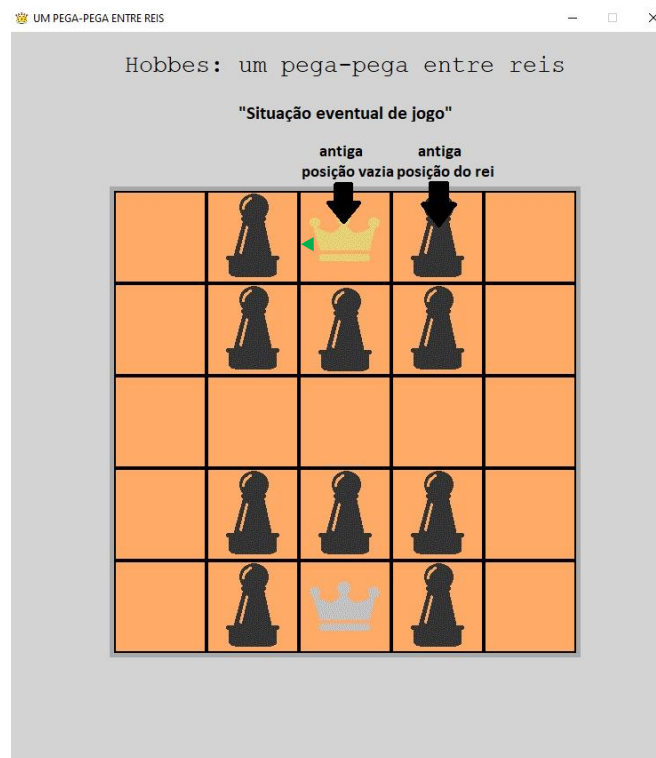
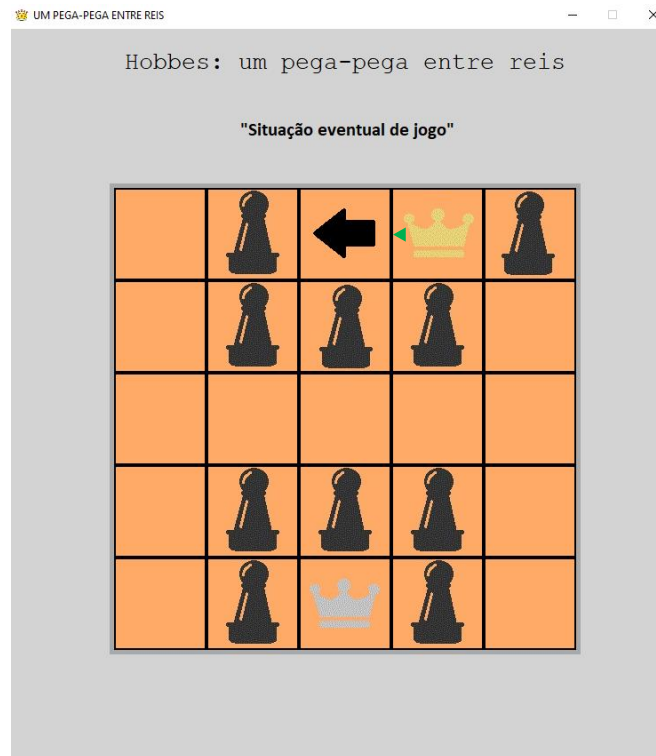
Hobbes: um pega-pega entre reis

"Situação eventual de jogo"



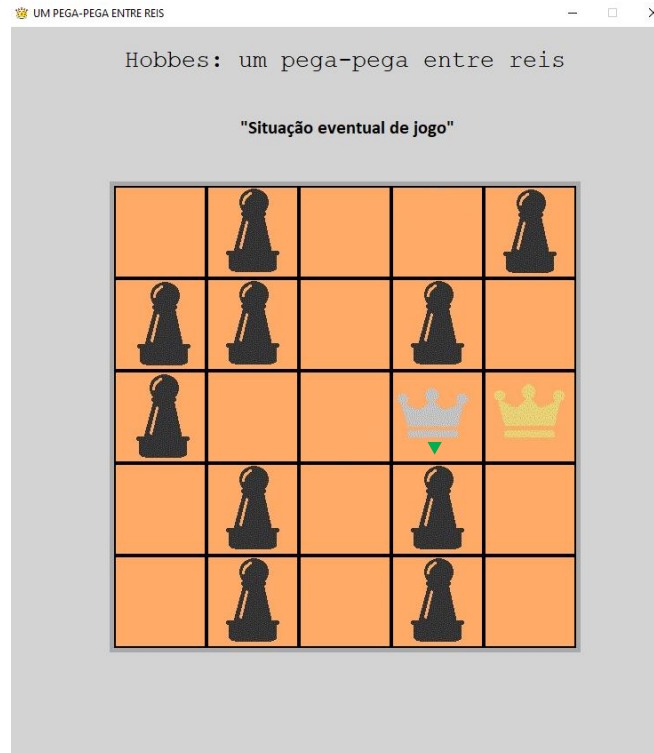
5. Quando o rei empurra um peão, o rei fica na posição do peão, e o peão na posição vazia. E esse movimento só poderá acontecer se a posição seguinte estiver vazia;



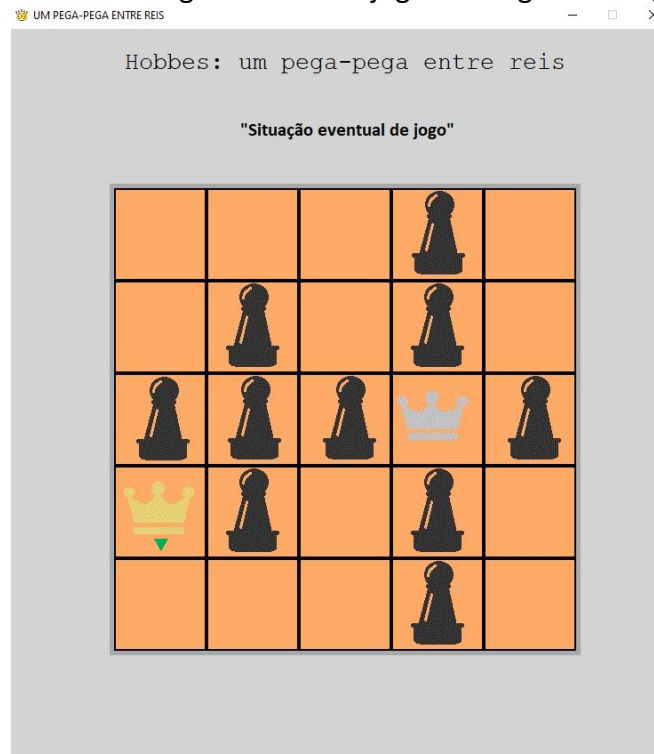


- O jogo acaba quando o rei de um jogador fica adjacente (mas não na diagonal) ao rei do jogador inimigo, ou quando o rei do jogador inimigo não consegue fazer nenhuma jogada obrigatória;

1. Jogador 2 ficou adjacente ao jogador 1. Jogador 2 venceu.



2. Jogador 2 não consegue fazer uma jogada obrigatória. Jogador 1 venceu.



8. Um jogador pode render-se a qualquer momento, concedendo a vitória ao jogador inimigo.

1.3 REFERÊNCIAS

[1] <https://www.youtube.com/watch?v=n8cbLsxb77w&ab_channel=VemKaJogar>
(regras do jogo).

2 VISÃO GERAL DO SISTEMA

2.1 ARQUITETURA DE SOFTWARE

Programa orientado a objetos, *stand-alone* com interface gráfica para interação com o usuário.

2.2 PREMISSAS DO DESENVOLVIMENTO

- O programa deve apresentar uma interface gráfica bidimensional;
- O programa deve ser implementado em Python, devendo executar em qualquer plataforma que disponha da linguagem interpretada Python;
- O programa deve dar suporte à interação local entre os usuários;
- A biblioteca de suporte ao desenvolvimento de interfaces em Python, Tkinter, deverá ser usada;
- Os diagramas devem ser produzidos com o software Visual Paradigm.

3 REQUISITOS DA APLICAÇÃO

3.1 REGRAS DE NEGÓCIO

RG01 – Tabuleiro: Terão 25 posições no tabuleiro, organizadas como uma matriz 5x5;

RG02 – Peças: Terão 10 peões no total, 5 dispostos em torno de cada rei, da cor prata;

RG03 – Cores das Peças: Cada rei terá uma cor diferente do outro, dourado e prata;

RG04 – Direção do Jogador: Cada rei terá um ícone em forma de seta para indicar sua direção do momento (seta para cima, baixo, direita ou esquerda).

3.2 REQUISITOS FUNCIONAIS

RF01 – Menu Jogo: O programa deve apresentar, inicialmente, um menu com o botão “iniciar”, para o jogador clicar e iniciar uma nova partida, e o botão “sair” para o jogador fechar o jogo.

RF02 – Desistir: Após o jogo ser iniciado, enquanto a partida estiver em andamento, haverá um botão na tela chamada “desistir”, que, quando clicado, o jogador da vez perderá automaticamente;

RF03 – Mover Rei: Após o jogo ser iniciado, enquanto a partida estiver em andamento, o jogador da vez poderá mover seu rei com as teclas “w” (para cima), “a” (para a esquerda), “s” (para baixo), “d” (para a direita) para as posições adjacentes à peça que estejam livres, até encontrar a posição que deseja ficar, sem poder pular peças que já estejam no tabuleiro;

RF04 – Puxar ou Empurrar Peão: Durante a partida, o jogador da vez deverá, com sua peça rei, puxar (com a tecla “p” do teclado) ou empurrar (com a tecla “e” do teclado) algum peão que esteja adjacente (mas não na diagonal) ao seu rei após ter movido ou não o rei. Se escolher puxar algum peão, a posição atrás do rei, que está na mesma direção do peão, deverá estar vazia, e, assim, o rei ficará nesta posição, e o peão na posição antiga do rei. Se escolher empurrar, a posição seguinte da do peão deverá estar vazia, e, assim, o peão ficará nesta posição e o rei na antiga posição do peão.

RF03 – Menu Partida Encerrada: Após a partida acabar (algum jogador vencer ou algum jogador tiver desistido), haverá um novo menu com um botão “reiniciar”, que, quando selecionada, iniciará uma nova partida, e o botão “sair”, que fechará o jogo imediatamente quando selecionada.

3.3 REQUISITOS NÃO FUNCIONAIS

RNF01 – Especificação de projeto: As especificações de projeto devem ser produzidas por meio de diagramas baseados em UML;

RNF02 – Linguagem de programação: O jogo deverá ser produzido na linguagem de programação *python*;

RNF02 – Interface gráfica: O programa terá uma interface gráfica única para os dois jogadores, e deve ser produzida com o auxílio da biblioteca de desenvolvimento *Tkinter*.