

Projeto Splendor

Especificação de Requisitos de Software

Versão 2.0

03/06/2025

Versão	Autor(es)	Data	Ação
1.0	Gabriel Rodrigo da Gama, Leonardo de Sousa Marques, Thayse Estevo Teixeira	09/04/2025	Estabelecimento dos requisitos
2.0	Gabriel Rodrigo da Gama, Leonardo de Sousa Marques, Thayse Estevo Teixeira	03/06/2025	Novo requisito funcional (15)

Conteúdo

1. Introdução
1. Visão geral
2. Requisitos de software
3. Apêndice: Regras Splendor

1. Introdução

1.1 Objetivo

Desenvolvimento de um programa distribuído que suporte a disputa de partidas do jogo Splendor no modo usuário contra usuário.

1.2 Definições, abreviaturas

Regras do jogo: ver apêndice

2. Visão Geral

2.1 Arquitetura do programa

A arquitetura do programa é baseada em 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 menu principal do jogo (ver requisito não funcional 3) 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 menu principal deve apresentar o botão “jogar” 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 para o tabuleiro com as informações recebidas, os baralhos embaralhados e as cartas e pedras distribuídas e posicionadas no estado inicial. Caso o jogador local seja quem inicia a partida, a interface deve estar habilitada para seu procedimento de lance. Jogadas habilitadas e desabilitadas devem ser claramente distinguíveis. Esta funcionalidade só deve estar habilitada se o programa estiver em seu estado inicial, isto é, sem partida em andamento e no menu principal;

Requisito funcional 3 – Navegar nos menus: No menu principal, além da opção “jogar”, tem as opções de ver “créditos” e “regras do jogo” – também presentes no menu do jogo em andamento (ver requisito não funcional 3). Ao clicar em créditos deve aparecer na tela o nome da empresa que criou o jogo e dos desenvolvedores da aplicação. Já para as regras, devem aparecer as instruções para jogar. Em ambos os casos, ao clicar no botão “voltar” no canto superior esquerdo da tela, o programa retorna à tela anterior (menu principal ou a tela do jogo, dependendo do caso), o menu do jogo também tem a opção “sair” pra abandonar partida (ver requisito funcional 4).

Requisito funcional 4 – Abandonar partida: o programa deve apresentar a opção de menu “sair” (ver requisito não funcional 3) para levar o programa ao seu seu estado inicial, isto é, para o menu principal, encerrando a partida anterior. Esta funcionalidade – e as próximas – só deve estar habilitada se o programa estiver com uma partida em andamento;

Requisito funcional 5 – Selecionar jogada: o programa deve permitir a um jogador habilitado selecionar uma das opções de jogada presente em sua área de jogadas. A jogada selecionada deve ser visualmente destacada das demais. O programa deve avaliar se o jogador consegue comprar alguma carta ou realizar um roubo (ver requisitos funcionais 7 e 9) se ele não consegue, a opção não satisfeita deve ficar desabilitada. Se a seleção for executada após o jogador já ter selecionado ou estiver efetuando uma jogada (ver os Requisitos funcionais de 6 a 10), a nova seleção se torna a jogada escolhida. O programa deve identificar se a jogada será a última antes do final da partida – se for, todas, apenas a opção “comprar cartas” será habilitada. O programa deve identificar se a jogada de compra de carta também estiver desabilitada e encerrar a partida;

Requisito funcional 6 – Comprar pedras: o programa deve permitir a um jogador habilitado que clique em “comprar pedras” (ver requisito não funcional 3) que selecione 3 pedras diferentes ou duas iguais, clicando nelas no tabuleiro. Se o jogador selecionar as pedras necessárias, o botão “finalizar jogada” deve ser habilitado e se o jogador clicar nele, o programa deve atualizar o número de pedras do tabuleiro e do jogador que efetuou a compra e enviar a jogada (ver Requisito funcional 12) contendo essa atualização. O programa deve garantir que as pedras compradas não podem ser ouro e nem pedras que acabaram (quando as 7 já foram compradas, ver apêndice), se isso acontecer, deve ser notificada “jogada inválida” e retomar a jogada onde parou. Se o jogador clicar em uma das jogadas a ser selecionada, esta deve seguir sendo a nova jogada a ser efetuada, do começo (ver requisito funcional 5);

Requisito funcional 7 – Comprar carta: O programa deve permitir a um jogador habilitado que clicou em “comprar cartas” (ver requisito não funcional 3) que selecione 1 carta das 12 viradas na mesa ao clicar nela no tabuleiro. Se o jogador clicar em uma das jogadas a ser selecionada, esta segue sendo a nova jogada a ser efetuada, do começo (ver requisito funcional 5). O programa deve identificar se o jogador tentar comprar uma carta que não tem pedras o suficiente para comprar, notificar “jogada inválida” e continuar esperando a seleção de uma carta. Se o jogador possui as pedras necessárias para a compra, o botão “finalizar jogada” deve ser habilitado, e se ele o jogador clicar nele a pontuação deve ser atualizada e outra carta de mesmo nível deve ser posta no lugar vago no tabuleiro se a carta comprada não for uma reservada anteriormente. O programa deve então identificar se a nova carta for de roubo, o jogador atual deve ficar com ela e outra carta deve ser virada, fazendo o mesmo teste novamente para garantir que qualquer carta de roubo virada fique com o jogador atual. O programa deve então enviar a jogada (ver Requisito funcional 12) – com a atualização das cartas (do tabuleiro e do jogador) e da pontuação – e avaliar o encerramento de partida. No caso de encerramento de partida, deve ser notificado o nome do jogador vencedor;

Requisito funcional 8 – Reservar carta: O programa deve permitir a um jogador habilitado que clicou em “reservar carta” (ver requisito não funcional 3) que selecione 1 carta das 12 viradas na mesa ao clicar em cima dela no tabuleiro. Se o jogador clicar em uma das jogadas a ser selecionada, esta segue sendo a nova jogada a ser efetuada, do começo (ver requisito funcional 5). Com a carta selecionada, o botão “finalizar jogada” deve ser habilitado e, se o jogador clicar nele, outra carta de mesmo nível deve ser posta no lugar da carta retirada do tabuleiro e o jogador deve ganhar 1 ouro, se ainda houver. Se a nova carta for de roubo, o jogador atual deve ficar com ela e outra carta deve ser virada, fazendo o

mesmo teste novamente para garantir que qualquer carta de roubo virada fique com o jogador atual. O programa deve enviar a jogada (ver Requisito funcional 12) contendo a atualização das cartas do tabuleiro e do jogador que efetuou a reserva;

Requisito funcional 9 – Roubar pedras: O programa deve permitir a um jogador habilitado que clicou em “roubar pedras” (ver requisito não funcional 3) que selecione 1 carta das suas cartas de roubo ao clicar nela. Se o jogador clicar em uma das jogadas a ser selecionada, esta segue sendo a nova jogada a ser efetuada, do começo (ver requisito funcional 5). O programa deve identificar se o adversário não tiver nenhuma das pedras indicadas na carta, notificar “jogada inválida” e voltar a esperar a escolha de outra carta. Se o adversário tem pelo menos 1 das pedras indicadas na carta selecionada, o botão “finalizar jogada” deve ser habilitado e, se o jogador clicar nele, as pedras indicadas na carta que estiverem na mão do adversário devem ser transferidas para o jogador atual – e a carta é descartada. O programa deve então enviar a jogada (ver Requisito funcional 12) contendo a atualização das pedras dos dois jogadores e da carta usada;

Requisito funcional 10 – Oferecer troca: O programa deve permitir a um jogador habilitado que clicou em “trocar pedras” (ver requisito não funcional 3) que selecione quaisquer pedras que possua para trocar com quaisquer outras do adversário. O botão “finalizar jogada” deve então ser habilitado e, se o jogador clicar nele, o programa deve enviar a jogada (ver Requisito funcional 12) contendo a proposta de troca. Se o jogador clicar em uma das jogadas a ser selecionada, esta segue sendo a nova jogada a ser efetuada, do começo (ver requisito funcional 5). Com a troca escolhida e a jogada finalizada,;

Requisito funcional 11 – 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 para o tabuleiro 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 12 – Enviar jogada: o programa deve enviar a jogada realizada (ver requisitos não funcionais de 6 a 10) ao adversário utilizando os recursos de DOG. Deve ser desabilitado o jogador local e o programa fica no aguardo de jogada do adversário (ver Requisito funcional 13) ou de notificação de abandono (ver Requisito funcional 14);

Requisito funcional 13 – 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 nos Requisitos funcionais de 6 a 10. O programa deve atualizar o tabuleiro e as peças dos jogadores de acordo com o que foi enviado. Se está sendo recebida uma proposta de troca, um pop-up com as opções de aceitar ou não deve aparecer para o jogador habilitado escolher – quando a escolha é feita, a jogada contendo a resposta é enviada imediatamente (ver Requisito funcional 12). Se foi recebida uma resposta negativa a uma proposta de troca, o tabuleiro deve atualizar para que outra jogada possa ser solicitada. Caso uma troca tenha sido aceita, a jogada é enviada contendo a atualização das pedras dos dois jogadores

e da carta usada. Se a jogada recebida for uma compra de carta, deve-se avaliar o encerramento de partida. No caso de encerramento, deve ser notificado o nome do jogador vencedor e qualquer clique na tela deve levar de volta ao menu principal. Depois de tratados os recebimentos e em caso de partida não encerrada, deve ser habilitado o jogador local, para que possa proceder a seu lance;

Requisito funcional 14 – 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, qualquer clique na tela depois disso deve levar ao menu principal.

Requisito funcional 15 – Desfazer Jogada: Após selecionar uma jogada (comprar carta, comprar pedra, trocar pedra ou reservar carta), o jogador deve ser possibilitado de clicar em “Desfazer Jogada” (ver requisito não funcional 3), a fim de retornar o estado inicial do tabuleiro do início do turno, da mesma rodada.

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.



[VOLTAR](#)

Regras do Jogo

Quem inicia a partida é sorteado de forma aleatória. A cada turno, cada jogador pode escolher apenas uma das seguintes ações:

- Comprar três pedras diferentes
- Comprar duas pedras iguais
- Reservar 1 das cartas da mesa e, se ainda houver ouro disponível, comprar um
- Comprar uma das cartas dispostas na mesa se tiver pedras o suficiente e retirar outra do baralho para pôr no lugar.

Se for uma carta de roubo, o jogador fica com ela pra si e retira outra carta

- Fazer uma oferta de troca de pedras ao outro jogador. Se ele recusar, escolher outra ação (pode ser outra oferta)
- Usar uma carta de roubo, se tiver, para roubar as pedras descritas na carta do outro jogador.

Se o jogador roubado não tiver todas as pedras, apenas as que ele tiver serão roubadas.
Se ele não tiver nenhuma, o roubo não pode ser feito.

Encerramento da Partida:

A partida é encerrada quando um dos jogadores atingir 15 ou mais pontos.
O adversário daquele que alcançou os 15 pontos ainda tem direito a mais uma jogada final de compra de carta na tentativa de ultrapassá-lo. Ganhá aquele que tiver mais pontos.

[VOLTAR](#)

Créditos

Splendor

Criado por Marc André

Distribuído por Space Cowboys

© 2014 Space Cowboys. Todos os direitos reservados.

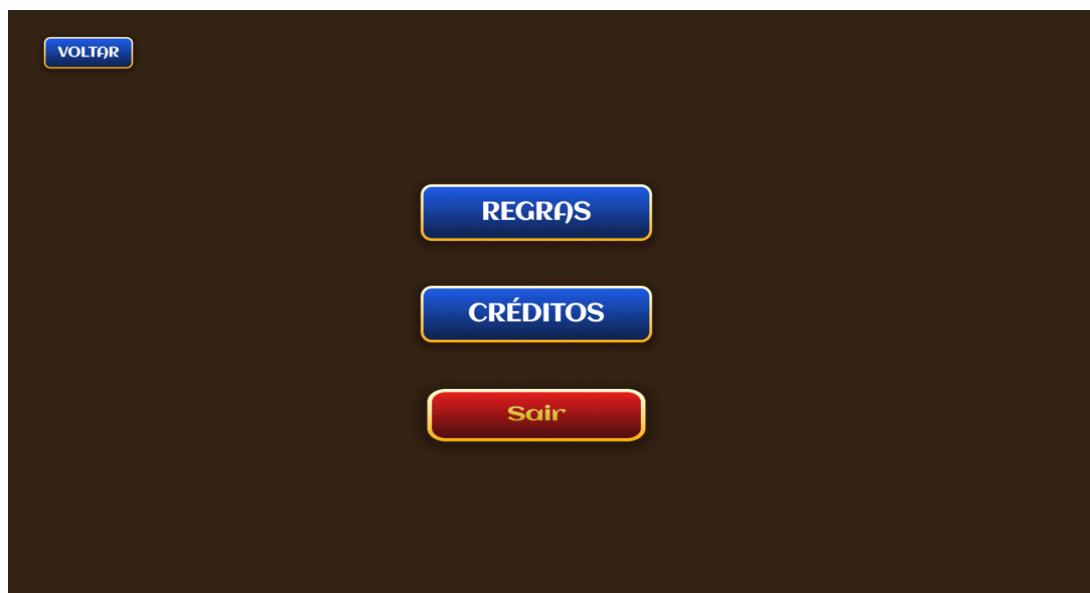
Esta versão digital é uma inspiração acadêmica criada por:

Gabriel Rodrigo da Gama

Leonardo de Sousa Marques

Thayse Estevo Teixeira

Jogo criado para a disciplina INE5417 - Engenharia de Software I
Universidade Federal de Santa Catarina (UFSC)



4. Apêndice: Regras Splendor

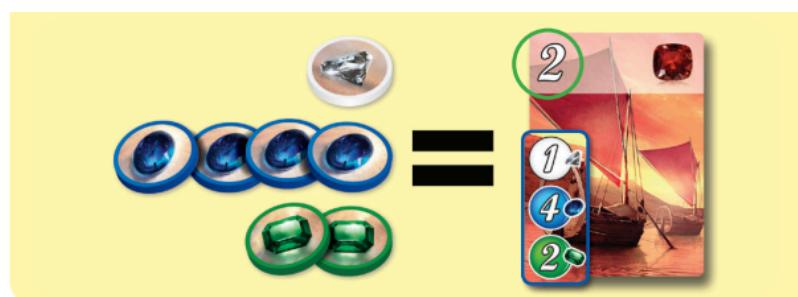
O jogo Splendor é disputado entre 2 jogadores. É um jogo onde você compra cartas a partir das pedras adquiridas durante a partida. Cada carta vale uma certa quantidade de pontos e o objetivo dos jogadores é alcançar 15 pontos ou mais.

Elementos do jogo

- 40 pedras: 7 de cada para os tipos diamante, ônix, rubi, esmeralda e safira e 5 do tipo ouro.

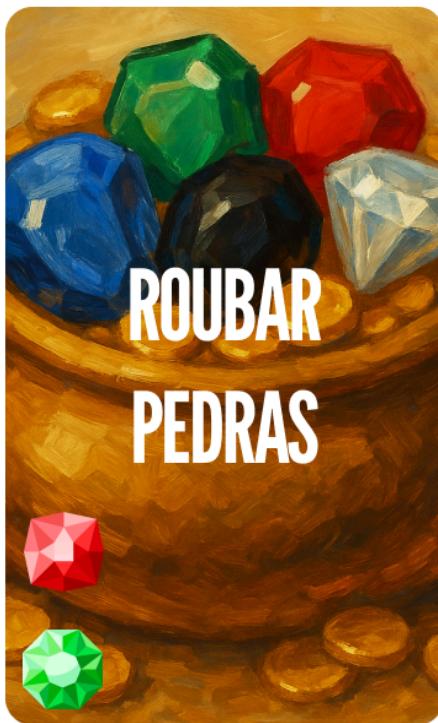


- Ouro são pedras coringa e podem substituir uma pedra de qualquer tipo.
- 49 cartas: 20 cartas de nível 1, 15 cartas de nível 2, 14 cartas de nível 3
 - No canto superior esquerdo fica quantos pontos a carta vale, de 0 a 5. Quanto maior o nível da carta, mais pontos ela vale;
 - No canto superior direito fica a pedra adquirida ao comprar a carta. Nem todas as cartas tem uma pedra de bônus;
 - No canto inferior esquerdo ficam as pedras necessárias para comprar a carta. Quanto maior o nível da carta, mais cara ela é;



Essa carta, por exemplo, custa 1 diamante, 4 safiras e 2 esmeraldas. Ela vale 2 pontos de prestígio e, ao comprá-la, o jogador ganha um rubi.

- Dentre as cartas de nível 3, existem 4 cartas de roubo. Elas não dão pontos nem pedras e não podem ser compradas. As pedras indicadas nelas são as que o jogador que a possui pode roubar do adversário. Se o adversário não tiver todas as pedras, o roubo ainda pode acontecer e apenas as pedras que ele tiver serão roubadas. Se o adversário não tiver nenhuma das pedras, o roubo não pode ser feito.



Essa carta, por exemplo, dá o direito de roubar um rubi e uma esmeralda. Mas se o adversário só tiver um rubi, a carta pode ser usada e apenas o rubi será roubado.

Preparação

- As cartas são separadas por nível (1, 2 e 3), embaralhadas e os montes são postos à mesa na ordem crescente.
- De cada monte são viradas as primeiras 4 cartas, que são deixadas à mostra na mesa.
 - As cartas de roubo não podem começar como uma carta virada, elas são separadas e reembaralhadas junto às de nível 3 após as cartas serem viradas.
- As pedras são separadas por tipo e postas na mesa.

Lances dos Jogadores

Quem inicia a partida é sorteado de forma aleatória. A cada turno, cada jogador pode escolher apenas uma das seguintes ações para fazer:

- Comprar três pedras diferentes;
- Comprar duas pedras iguais;

- Reservar 1 das cartas da mesa e, se ainda houver ouro disponível, comprar um;
- Comprar uma das cartas dispostas na mesa ou uma reservada anteriormente se tiver pedras o suficiente e retirar outra do baralho para pôr no lugar. Se for uma carta de roubo, o jogador fica com ela pra si e retira outra carta;
- Fazer uma oferta de troca de pedras ao outro jogador, se ele recusar, escolher outra ação (pode ser uma outra oferta);
- Usar uma carta de roubo, se tiver, para roubar as pedras descritas na carta do outro jogador, se ele tiver pelo menos uma delas. Se o jogador roubado não tiver todas as pedras, apenas as que ele tiver serão roubadas. Se ele não tiver nenhuma, o roubo não pode ser feito.

Encerramento da Partida

A partida é encerrada quando um dos jogadores atingir 15 ou mais pontos. O adversário daquele que alcançou os 15 pontos ainda tem direito a mais uma jogada final de compra de carta na tentativa de ultrapassá-lo. Ganha aquele que tiver mais pontos.