

Projeto Gobblet Gobblers

Especificação de Requisitos de Software

Versão 1.0

17/08/2022

Versão	Autor(es)	Data	Ação
1.0	Ricardo Pereira e Silva	17/08/2022	Estabelecimento dos requisitos

Conteúdo:

1. Introdução
 2. Visão geral
 3. Requisitos de software
- Apêndice: Regras Gobblet Gobblers

1.Introdução

1.1 Objetivo

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

1.2 Definições, abreviaturas

Regras do jogo: ver apêndice

1.3 Referências:

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

<https://www.youtube.com/watch?v=Xltsno2ubgc>

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 as peças dos jogadores em suas respectivas áreas de peças e tabuleiro vazio e a mensagem 'Gobblet Gobblers') 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. 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 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 – Selecionar peça: O programa deve permitir a um jogador habilitado selecionar uma peça presente em sua área de peças. A peça selecionada deve ser visualmente destacada da interface do programa. Se a ação for executada após o jogador já ter uma peça selecionada ou ter selecionado origem (neste caso, ver Requisito funcional 5), deve ser notificado lance irregular e o programa deve novamente aguardar a primeira ação do jogador habilitado (selecionar peça ou selecionar origem);

Requisito funcional 5 – Selecionar origem: O programa deve permitir a um jogador habilitado selecionar uma peça sua presente em uma posição do tabuleiro. A peça selecionada deve ser visualmente destacada da interface do programa. Se a ação for executada após o jogador já ter selecionado origem ou ter uma peça selecionada em sua área de peças (neste caso, ver Requisito funcional 4), deve ser notificado lance irregular e o programa deve novamente aguardar a primeira ação do jogador habilitado (selecionar peça ou selecionar origem);

Requisito funcional 6 – Selecionar destino: O programa deve permitir a um jogador habilitado selecionar uma posição do tabuleiro, onde será colocada a peça previamente selecionada (de sua área de peças ou de uma posição do tabuleiro). Esta funcionalidade só deve estar habilitada se o programa estiver com peça selecionada (ver Requisito funcional 4) ou com posição origem selecionada (ver Requisito funcional 5). O programa deve avaliar se a posição destino pode receber a peça selecionada, de acordo com as regras do jogo (ver apêndice). Em caso positivo, deve efetuar a jogada - isto é, remover a peça de seu posicionamento inicial e colocá-la na posição destino - e, em caso negativo, notificar o lance irregular e o programa deve novamente aguardar a primeira ação do jogador habilitado (selecionar peça ou selecionar origem). No caso de êxito na colocação de peça em seu destino, o programa deve enviar a jogada ao adversário (utilizando os recursos de DOG) e avaliar o encerramento de partida. A jogada enviada deve conter a posição de origem do tabuleiro (no caso de seleção de origem) ou o tamanho da peça selecionada (no caso de seleção de peça) e a posição destino. No caso de encerramento de partida, deve ser notificado o nome do jogador vencedor; no caso de não encerramento, deve ser desabilitado o jogador local e

o programa fica no aguardo de jogada do adversário (ver Requisito funcional 8) ou de notificação de abandono (ver Requisito funcional 9);

Requisito funcional 7 – 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 8 – 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 6 – Selecionar destino’. O programa deve remover a peça da origem definida e colocá-la no destino. Após isso, deve 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 9 – 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 da imagem abaixo.



Apêndice: Regras Gobblet Globbers

O jogo Gobblet Globbers é disputado entre 2 jogadores em um tabuleiro de 9 posições, organizado como uma matriz 3x3 e o objetivo dos jogadores, similar ao que ocorre no Jogo da Velha, é alinhar 3 de suas peças horizontal, vertical ou diagonalmente.

Elementos do jogo

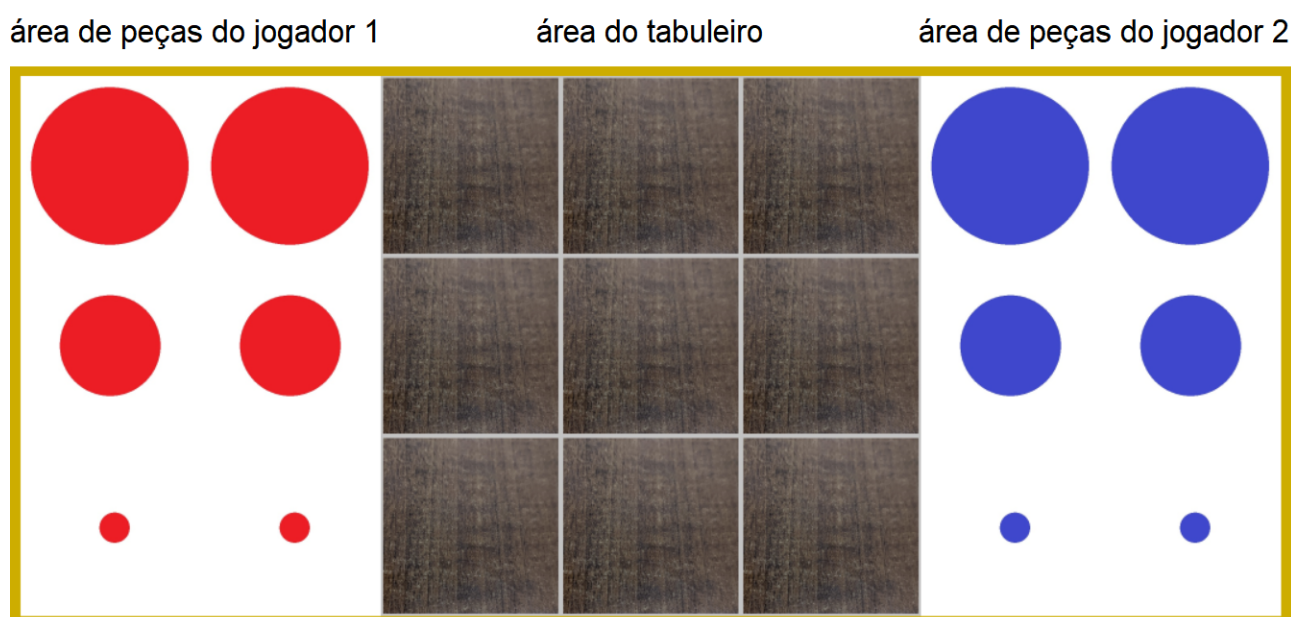


Figura 1 - elementos do jogo Gobblet Globbers

Além do tabuleiro 3x3, representado na área do tabuleiro na figura 1, o jogo conta com peças de 3 tamanhos, grande, médio e pequeno. Cada jogador inicia a partida com 6 peças, de cor diferente das peças do adversário, sendo 2 de cada tamanho, conforme representado nas áreas de peças da figura 1. No tabuleiro, uma peça maior pode sobrepor uma peça menor, do próprio jogador ou do jogador adversário.

Lances dos jogadores

Os jogadores procedem a seus lances de forma alternada (adotando algum critério para definir o jogador que inicia a partida) e, na sua vez, o jogador pode efetuar um dos seguintes lances: colocar peça no tabuleiro e mover peça do tabuleiro.

Colocar peça no tabuleiro: consiste em remover uma de suas peças presente na sua área de peças e colocá-la em uma posição do tabuleiro. Só é possível colocar a peça selecionada em uma posição vazia do tabuleiro ou em uma posição ocupada por uma peça menor que aquela peça que está sendo colocada, sendo que neste caso tanto a peça menor como a maior permanecem na posição. Esse tipo de lance só é possível enquanto o jogador possuir pelo menos uma peça na sua área de peças e só pode ser efetuado com as peças lá disponíveis.

Mover peça do tabuleiro: consiste em selecionar uma de suas peças presente uma posição do tabuleiro e movê-la para uma posição adjacente horizontal, vertical ou diagonalmente. Assim como na colocação de peça, só será possível efetuar o movimento se a posição destino estiver vazia ou ocupada por uma peça menor que aquela peça que está sendo colocada.

Encerramento da partida

A partida encerra apenas quando um dos jogadores for vencedor. Caso um jogador não possua peça na sua área de peças, apenas pode efetuar movimento de peça do tabuleiro. Para avaliação da condição de vitória (alinhamento de 3 peças), considera-se a maior peça presente em cada posição. Se uma posição estiver ocupada por 3 peças simultaneamente, por exemplo, será considerada apenas a cor da peça maior, que sobrepõe as demais. O alinhamento pode ser obtido por meio dos lances 'colocar peça no tabuleiro' ou 'mover peça do tabuleiro'. Caso, ao mover uma peça, um jogador obtenha a condição de vitória, mas se o movimento descobrir uma peça do adversário e, com isso, produzir a condição de vitória também para o adversário, a vitória será do adversário, como penalização pelo lance imprudente (um jogador deveria saber que peças estão sendo sobreposta por outras peças).