Programação Modular - Segunda entrega

Grupo 8
EDUARDO SARDENBERG TAVARES
SOL CASTILHO ARAUJO DE MORAES SEDA
VICTOR NIELSEN RIBEIRETE

Regras:

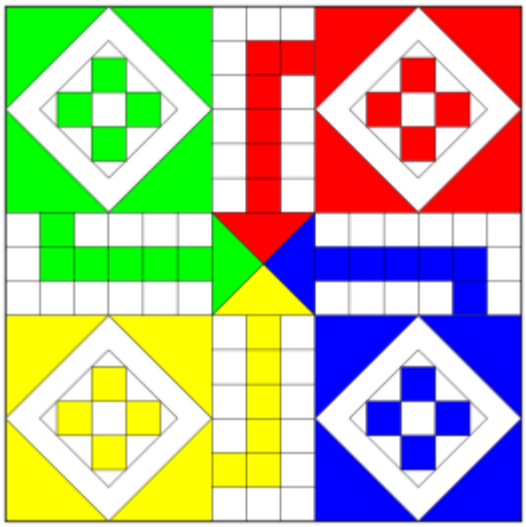
Cada jogador por sua vez lança um dado e faz avançar um dos seus peões em jogo o número de casas indicado. O seis permite colocar em jogo um peão que esteja na casa inicial ou fazer avançar um peão seis casas, e ainda um novo lançamento de dados. O número um também permite que o jogador tire o peão, mas é só o seis que permite o jogador a lançar o dado novamente.

Quando o jogador entra com um peão na parte final, poderá completar o percurso somente se tirar o número de casas exato da casa final. Caso tire um número maior, o jogador entra e retrocede o número das casas que sobraram.

Não é permitido mais do que um peão em cada casa. Caso um peão venha a ocupar uma casa ocupada por um peão de outro jogador, o peão original regressará à casa inicial, é o chamado "comer" (principalmente no Brasil). É proibido "comer" o adversário que está na casa de saída.

Quando dois peões de uma mesma cor se encontram em uma mesma casa, forma-se uma torre, impedindo outro peão de ocupar esta casa. Só poderá comer a torre com outra torre. Dois peões somente poderão caminhar como torre (ou seja, ambos juntos) caso haja uma torre no meio do caminho para ser "comida" uma vez que somente uma torre poderá comer outra, mandando os dois peões para casa inicial. Não havendo outra torre, e lançando o dado, o jogador deverá desfazer a torre, caminhando somente com um dos peões.

Tabuleiro:



Requisitos:

- Máximo de 4 jogadores;
- Só pode usar digitação para escrever o nome de cada jogador;
- As pontuações devem ser salvas;
- Os jogadores se revezam para jogar durante o turno;
- Movimentar cada peça ou jogar um dado deve ser feito através de arrastar ou clicar;
- Cada jogador joga um dado para movimentar uma peça;
- Um jogador deve chegar até o final do percurso para que o jogo acabe;
- Para uma peça entrar no percurso o número do dado do jogador do turno deve ser 1 ou 6;
- Caso um jogador tire 6 ele poderá jogar novamente;
- Caso uma peça do jogador do turno for cair em um espaço contendo uma peça de outro jogador, o peão desse segundo volta para o início;
- Existem “safe zones” no percurso onde a peça dos jogadores estão salvas de serem retornadas para o início;
- Quando dois peões do mesmo jogador estão em um mesmo espaço eles formam uma torre e só podem voltar para o início se outra torre de outro jogador para no mesmo espaço;
- Após o fim do jogo uma tela mostrando o vencedor irá aparecer e o resultado será salvo.

Itens de Backlog Gerais da aplicação:

- Menu inicial:
    - Escolha de quantidade de jogadores (2-4);
    - Nome dos jogadores;
        - Digitação pelo teclado
        - Valor da aposta de cada um
    - Tela de regras;
    - Histórico de partidas;
    - Botão de fechar;
- Tabuleiro:
    - Mostrar botão de ativar/desativar música
    - Mostrar os jogadores em seus respectivos cantos, junto de suas apostas
    - Botão para rolar o dado
        - Conecta com função random que retorna o valor na tela
    - No início rolar um dado decide quem começa;
    - Peão se mover pelo mapa baseado no número do dado;
    - Clicar no peão que o jogador quer mover
        - Peão andará número de casas fornecido pelo dado

    - Se o jogador tirar 1 ou 6, ele pode libertar um peão na sua base (caso tenha), ou mover o número indicado por um peão/torre no tabuleiro
    - Caso saia 6 no dado o jogador também poderá jogar novamente;
    - Se tiver só um peão, ou se dois peões estiverem no mesmo local eles se movem sozinhos;
    - Se tiver mais de um peão de um jogador em espaços diferentes o jogador escolhe qual mover;
    - Se um peão caiu em um local onde já tinha um de outro jogador este do outro volta ao início;
    - Se dois peões do mesmo jogador ocuparem o mesmo espaço e tiver uma torre inimiga no número exato do dado eles se movem juntos;
    - Se dois peões andando juntos caírem em um espaço ocupado por dois peões de outro jogador estes do outro voltam para o início;
    - Se o jogador da vez possuir uma torre e houver outra torre no tabuleiro, podem haver 3 casos:
        - Se o número do dado for menor que a distância da torre do jogador à torre adversária, a torre do jogador se desfaz e ele anda com um dos seus peões
        - Se o número do dado for igual à distância da torre adversária, a torre do jogador “come” a torre adversária, fazendo os 2 peões adversários retornarem a base
        - Se o número do dado for maior que a distância da torre adversária, o jogador não poderá mover nenhum dos peões de sua torre
    - Se um peão estiver na “safe zone” e o de um jogador tentar comer o que tentou voltar para o início;
    - Caso um peão esteja no caminho final se o jogador tirar um número maior que que a quantidade de espaços até o fim ele volta o excedente;
    - O jogo termina quando todos os peões de todos os jogadores chegam no fim do percurso;

- Tela de Fim de Jogo
    - Mostrar jogador vencedor
    - Botão de sair
    - Botão de jogar novamente

- Banco de dados
    - Registrar o peão escolhido (caso jogador tenha escolhido o peão)
    - Número do dado
    - Registar a ordem de chegada no final do circuito.

Itens de Backlog atendidos na primeira entrega:

- Menu inicial:
    - Escolha de quantidade de jogadores (2-4);
    - Nome e cor dos jogadores;
        - Digitação pelo teclado + cor pelo Mouse
    - Tela de regras;
    - Botão de fechar;**:**
- Obter os jogadores da interface no módulo main.py (Feito por Victor 10/10 até 20/10 05:00)
- Implementar mensagem/interface com as regras no módulo regras.py (Feito por Victor 18/10 durante 6h)
- JogadorPeca.py (Feito por Victor e Eduardo 19/10 20h até 20/10 04:30)
- Testes completos da aplicação testes.py (Feito por Sol 01/10 13:00 até 19/10 00:00)

Diagrama Estrutural

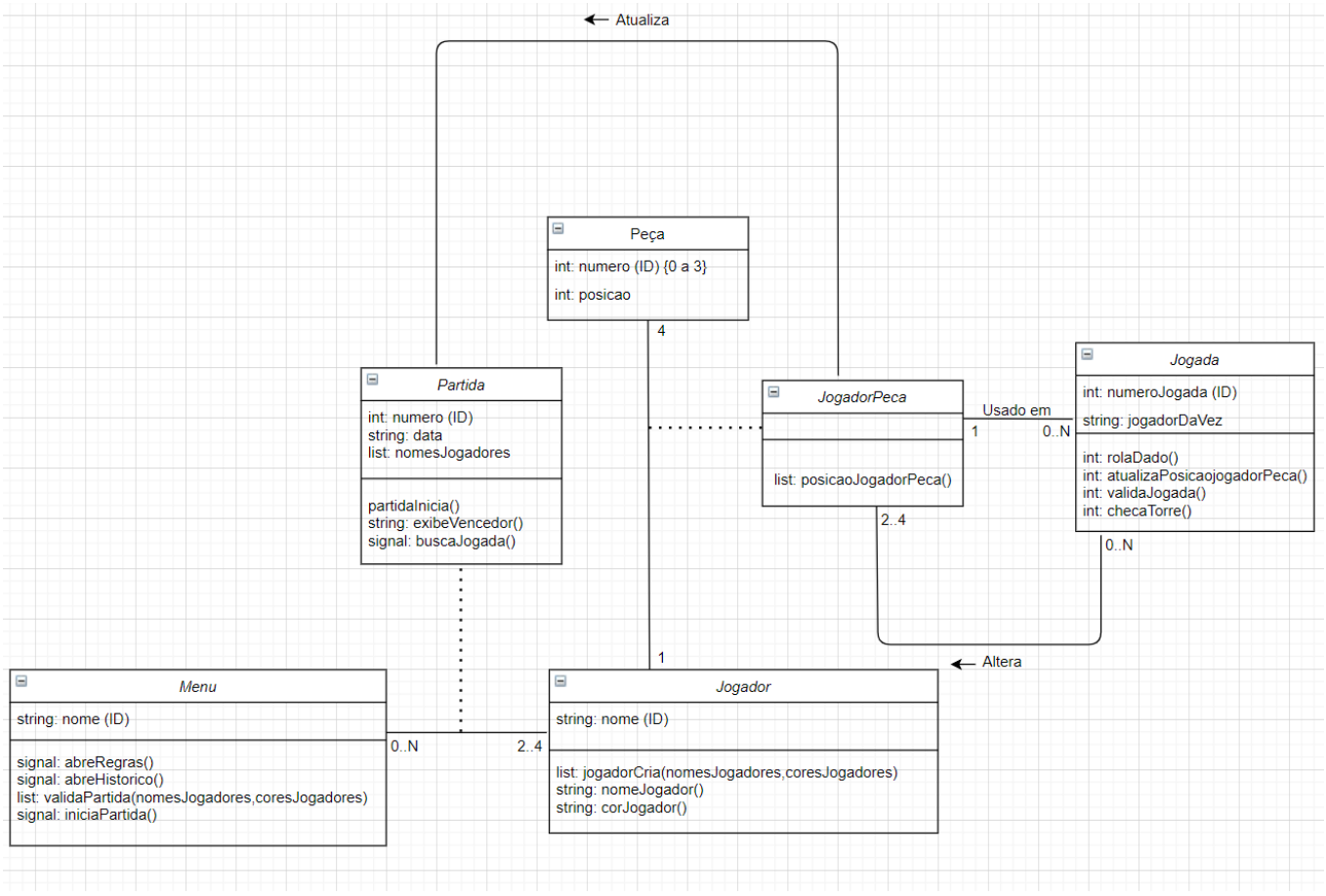
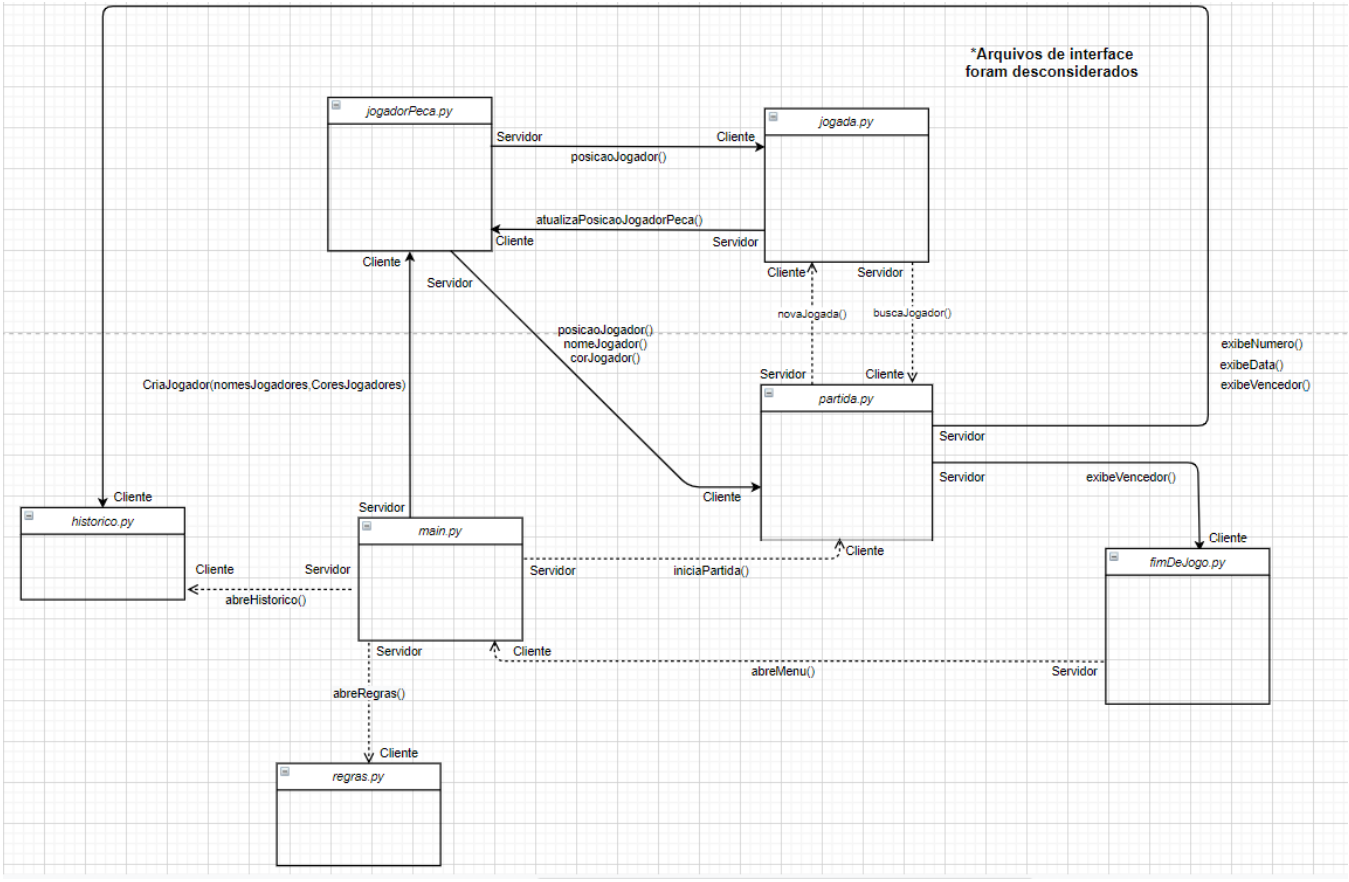


Diagrama de Arquitetura Modular



Planejamento das entregas restantes:

- T3:
  - Implementar todas as verificações das regras no módulo jogada.py
  - Implementar a interface e comunicação de sinais do módulo partida.py
  - Implementar a tela de fim de jogo no módulo fimDeJogo.py
  - Conectar o módulo fimDeJogo.py com o módulo main.py
  - Implementar os testes dos modulos anteriores no Testes.py

- T4:
  - Implementação do banco de dados
  - Implementar o histórico de partida no módulo historico.py + o teste_historico.py