

Descrição das Classes:

Bank: Essa classe representa o banco, e é responsável por fazer quaisquer transações bancárias, entre jogadores ou entre um jogador e o banco. Como, inicialmente, o banco é dono de todas as propriedades, quando um jogador for comprar uma, ele irá comprar do banco. Por fim, o banco também é responsável por pagar o salário dos jogadores.

Initializer: Responsável pela maioria das inicializações, como o mapa, jogadores, casas, etc...

Cartas: Para implementar as cartas, nos baseamos em um arquivo CSV, onde os valores separados por vírgula, representam o efeito da carta e alguns sinais para os métodos agirem sobre eles. Essa classe contém métodos para abrir o CSV, ler as informações dele, sortear uma carta, e possui métodos para executar a função das cartas.

Dice: Uma classe bem simples, que apenas joga os dados e devolve seus resultados, o dado só pode ser possuído por um player por vez.

Draw: Essa classe contém o loop principal e herda Application, que é uma classe do JavaFX, que usamos para desenharmos a interface, além do loop principal do jogo, essa classe conta com funções para desenhar os dados, o mapa, o menu, e qualquer interface necessária.

ImageManager: Essa classe é responsável por carregar dos arquivos as imagens que serão usadas no jogo.

Log: Cuida das informações das hipotecas e dívidas de cada jogador, para isso, conta com métodos para atualizar esses dados.

Monopoly: Esse é o mapa, ele pode acessar o banco, um vetor com todos os jogadores, o dado e todas as propriedades do mapa, e ele contém métodos para devolver a quantidade de casas, cartas, jogadores, ou a posição de algum jogador específico.

Player: Contém todas as informações básicas de um jogador, seu ID, sua carteira, sua posição, um portfólio com suas propriedades, métodos para devolver sua posição, se mover, devolver seu portfólio e comprar propriedades com o banco.

Portifolio: O portfólio contém a lista de propriedades das quais uma determinada entidade é dona, e possui métodos para buscar, remover ou adicionar propriedades na lista.

propInfo: De maneira similar a Cartas, essa classe acessa um arquivo CSV com as informações necessárias para inicializar as propriedades.

Squares: Informações básicas de uma propriedade, como sua posição e seu tipo.

Property: Essa classe herda Squares, e ela gerencia propriedades, que são um tipo de casa. Contém métodos para pagar aluguel, melhorar e devolver suas informações gerais

como: seu valor, o custo do aluguel, seu estado de melhoria, a que monopólio ela pertence, etc.

Register: É um arquivo do banco que registra quem é dono de qual propriedade, e registra os diferentes monopólios, com isso, possui métodos para definir os valores mencionados e checar se um player possui um monopólio.

Save: Responsável por salvar o estado do jogo, para isso, ela cria um arquivo CSV, com as informações de cada jogador na rodada atual, como o id, a posição atual, todas as propriedades que eles possuem e seus estados de melhoria, além de suas dívidas e hipotecas. Também possui um método para fazer o inverso, tirar essas informações do CSV e inseri-las no jogador.

Special: Essa classe herda Squares, pois gerencia casas especiais, que são um tipo de casa, ela cuida das casas de prisão, notícia, feriado, início e imposto.

Stocks: Herda Squares, pois também é um tipo de casa, e contém seu valor e métodos para atualizar seu valor baseado em quantos stocks um player tem.

Wallet: É a carteira de um jogador, além de conter dinheiro, também contém métodos para conferir se o jogador está com saldo negativo, ou seja, falido.

,