

Projeto Monopoly

Padrões e Frameworks – UEFS, turma 2008.2

Geral

Este projeto é uma adaptação/simplificação do projeto “Monopoly” solicitado pelo professor Jacques Sauv , em um dos seus cursos de gradua  o da UFCG. Inclusive, parte significativa do texto foi copiada literalmente.

O Projeto

Voc s devem projetar e implementar uma vers o em software do jogo de tabuleiro “Monopoly”. Monopoly   uma vers o brasileira do jogo “Banco Imobili rio”. O Monopoly original, americano, tem diferen as importantes na disposi  o do tabuleiro e nas regras.   a vers o americana que ser  feita (nada nos impede de mudar as regras para fazer a vers o brasileira depois). Voc s podem baixar o jogo no site www.gamehouse.com, ou usem qualquer outro site, para se familiarizarem. Cuidado para n o perderem tempo demais nessa etapa, pois o jogo pode viciar rapidamente – lembrem-se de que voc s t m um projeto para entregar.

Toda a an lise e especifica  o   dada pronta para voc s, atrav s desse documento (leiam com bastante aten  o) e, principalmente, atrav s dos artefatos de an lise execut veis representados pelos testes de aceita  o autom ticos que ser o fornecidos. Um dos objetivos desse projeto   que voc s consigam construir um jogo de Monopoly que passe 100% dos testes de aceita  o. Se conseguirem, significa que voc s atingiram o objetivo mais importante no desenvolvimento de software de qualidade: *atender aos requisitos do cliente*. Os testes de aceita  o acessam apenas a l gica do neg cio, n o a interface com o usu rio.

Em um software como um jogo, interfaces bonitas e f ceis de usar s o uma caracter stica muito importante. Entretanto, para facilitar a vida de voc s,   suficiente criar um Monopoly com uma simples interface textual (comandos de console). Se voc s tiverem tempo de fazer uma GUI, ter o um b nus na nota *se* o programa rodar 100% dos testes de aceita  o. Caso isso n o aconte a, n o ganham b nus, porque foram se preocupar com GUI antes de fazer o dever de casa (n o lembraram do objetivo principal); minha dica   portanto: *esque am* de interface GUI bonitinha. H  outros objetivos t mb m que somam para a nota final de voc s: vejam se  o que discute avalia  o.

Voc s ter o que projetar e implementar o Monopoly de uma forma incremental. Haver  tr s milestones, e em cada uma novas funcionalidades ser o adicionadas ao sistema. As funcionalidades est o descritas dentro de User Stories. Uma User Story consiste de duas partes: 1) uma descri  o informal do que deseja de funcionalidade e das intera  es que o sistema deve suportar; 2) um conjunto formal de testes de aceita  o que, se executarem direitinho, comprovam que a funcionalidade foi implementada como desejada (ver Se  o Testes de Aceita  o). Em cada milestone haver  um conjunto de user stories que dever  ser implementado. *Antes de entregar um milestone*, sempre leia as recomenda  es e a listagem do que deve ser entregue.

O Jogo

O Monopoly que voc s v o fazer   assim: um certo n mero (2 a 8) de jogadores (humanos) est o na frente de um computador (em uma outra disciplina a gente faz um Monopoly em rede ...), e se alternam fazendo jogadas atrav s de comandos de console (comandos para jogar os dados, comprar propriedades, pagar para sair da pris o, construir casas, etc.), at  que s  reste um jogador ativo (que n o entrou em fal ncia), o qual ser  considerado vencedor. Veja nas descri  es das user stories os detalhes da an lise (h  t mb m um gloss rio dos termos usados no jogo):

User Stories

As User Stories podem ser listadas abaixo e podem sofrer alterações no decorrer do projeto.

User story 1 – Criar um novo jogo

Permite que um jogo de Monopoly seja iniciado. Primeiramente, o programa deve pedir ao usuário que entre com o número de jogadores, que deve ser entre 2 e 8. Então, para cada jogador, o programa pede um nome e uma cor para o peão, que deve ser escolhida entre as seguintes: {preto, branco, vermelho, verde, azul, amarelo, laranja, rosa}. Quando esse processo de coleta de informações acaba, o programa deve mostrar uma mensagem indicando que o jogo está começando, e pedir que o primeiro jogador forneça um comando.

Exemplo do console:

```
Entre com o número de jogadores [2-8]: 2
Entre com o nome do jogador no. 1 : ana
Escolha a cor do peão do jogador no. 1 entre as opções seguintes:
[preto][branco][vermelho][verde][azul][amarelo][laranja][rosa]
: azul
Entre com o nome do jogador no. 2 : joao
Escolha a cor do peão do jogador no. 2 entre as opções seguintes:
[preto][branco][vermelho][verde][amarelo][laranja][rosa]
: vermelho
O jogo Monopoly foi iniciado.
A jogada de ana começou.
Comandos disponíveis: [jogar][sair]
Entre com um comando:
```

User story 2 – Jogada do Monopoly

Permita que os jogadores se alternem realizando jogadas, implementando as regras e requisitos listados abaixo. A ordem dos jogadores é a mesma fornecida na criação do jogo. A jogada começa com uma mensagem indicando quem é o jogador atual, e quais são os comandos disponíveis. Inicialmente, os únicos comandos disponíveis para os jogadores são “jogar” (os dados) ou “sair”. Em jogadas posteriores, outros comandos poderão ficar disponíveis. O comando “sair” é usado para sair do jogo. O programa deve mostrar uma mensagem e pedir uma confirmação “sim/não”, para garantir que o usuário deseja realmente terminar o jogo. O comando “jogar” faz os dados serem lançados e avança o peão no tabuleiro de acordo com a soma dos números obtidos no lançamento dos dados (uma mensagem deve informar os números obtidos e onde o peão caiu). Dependendo do lugar onde o peão caia, várias coisas podem acontecer, de acordo com as regras e requisitos listadas ao final dessa User Story. Implemente todas essas regras e requisitos. Ao usar o comando “jogar”, o jogador está ciente de que quer terminar sua jogada. Se o jogador cair em uma propriedade ou ferrovia que possua um título disponível para venda, o jogador compra o título automaticamente, se tiver dinheiro suficiente. Após as consequências da jogada terem sido tratadas (pagamentos, compras, etc.) e informadas no console, a jogada do jogador termina e começa a jogada do próximo jogador, da mesma forma: uma mensagem deve aparecer indicando quem é o jogador da vez, quais os comandos disponíveis, e o programa pede que seja entrado um comando.

Exemplo do console:

A jogada de ana começou.

Comandos disponíveis: [jogar][sair]

Entre com um comando: `jogar`

O jogador ana tirou 6,4 e o peão avançou para 10 – Jail, Just Visiting.

Nada aconteceu.

A jogada de joao começou.

Comandos disponíveis: [jogar][sair]

Entre com um comando: `sair`

Você tem certeza de que quer sair (Sim/Não)? `S`

Jogo encerrado.

Regras e requisitos que devem ser implementados nessa User Story (vai haver testes de aceitação para tudo):

- O tabuleiro é composto de 40 lugares, incluindo o ponto de partida (“Go”). A Mediterranean Avenue é o lugar 1, e o ponto de partida é o lugar 40 (isso para efeito da consulta à tabela 1 – implemente como quiser).
- O tabuleiro deve seguir a disposição do Monopoly clássico (com nomes das ruas de Atlantic City, veja a figura 1 abaixo); os preços e aluguéis estão na tabela 1; cuidado! Implemente os nomes corretos, e os valores corretos da tabela 1 (os testes são baseados neles).
- Implemente as ações do ponto de partida (“Go”), propriedades, ferrovias, e impostos, apenas; outros tipos de lugar devem ser tratados como Parada Livre (Free Parking) – nada acontece.
- Toda propriedade pertence a um grupo com uma cor associada. As quatro ferrovias (Railroads) formam um grupo à parte. Os dois serviços públicos (Utilities – Water Works e Electric Company) também formam um grupo à parte.
- Todo jogador recebe \$1500 quando o jogo começa.
- Qualquer número de jogadores pode estar em um mesmo lugar ao mesmo tempo.
- Se um jogador cai em uma propriedade que outro jogador possui, ele deve pagar ao dono o valor do aluguel (use a tabela 1).
- Se um jogador cai em uma ferrovia que outro jogador possui, ele deve pagar ao dono o valor da “corrida”, que depende do número de ferrovias que o dono possui: \$25 se o dono só possui 1, \$50 se o dono possui duas, \$75 se o dono possui 3, ou \$100 se o dono possui todas as 4.
- Se o jogador cair no imposto de renda (income tax), ele deve pagar \$200 ao banco.
- Se o jogador cair no imposto de riqueza (luxury tax), ele deve pagar \$75 ao banco.
- Se o jogador cair no ou passar pelo ponto de partida (“Go”), ele recebe \$200.
- Se o jogador cair na Parada Livre (Free Parking), ou qualquer dos lugares não implementados, nada acontece.
- Se o jogador entrar em falência (dinheiro < 0), ele é automaticamente excluído do jogo, e todos os seus pertences retornam ao banco e ficam disponíveis para serem comprados por outros jogadores. Débitos que excedam o que o jogador falido pôde pagar são perdidos (por exemplo, um jogador tem \$50 e sai do jogo por dever \$75 de aluguel – ele entra em falência, mas o credor só recebe \$50).

- O jogo termina quando só restar um único jogador.

Go →	Mediterranean Avenue	Community Chest	Baltic Avenue	Income Tax	Reading Railroad	Oriental Avenue	Chance	Vermont Avenue	Connecticut Avenue	Jail
Boardwalk										St. Charles Place
Luxury Tax										Electric Company
Park Place										States Avenue
Chance										Virginia Avenue
Short Line Railroad										Pennsylvania Railroad
Pennsylvania Avenue										St. James Place
Community Chest										Community Chest
North Carolina Avenue										Tennessee Avenue
Pacific Avenue										New York Avenue
Go To Jail	Marvin Gardens	Water Works	Ventnor Avenue	Atlantic Avenue	B&O Railroad	Illinois Avenue	Indiana Avenue	Chance	Kentucky Avenue	Free Parking

MONOPOLY

Fig 1 – Tabuleiro Monopoly

Posição	Nome	Preço	Aluguel sem casa	1 casa	2 casas	3 casas	4 casas	Hotel	Hipoteca	Preço da casa
1	Mediterranean Avenue	60	2	10	30	90	160	250	30	50
2	Community Chest 1									
3	Baltic Avenue	60	4	20	60	180	320	450	30	50
4	Income Tax									
5	Reading Railroad	200							100	
6	Oriental Avenue	100	6	30	90	270	400	550	50	50
7	Chance 1									
8	Vermont Avenue	100	6	30	90	270	400	550	50	50
9	Connecticut Avenue	120	8	40	100	300	450	600	60	50
10	Jail – Just Visiting									
11	St. Charles Place	140	10	50	150	450	625	750	70	100
12	Electric Company	150							75	

13	States Avenue	140	10	50	150	450	625	750	70	100
14	Virginia Avenue	160	12	60	180	500	700	900	80	100
15	Pennsylvania Railroad	200							100	
16	St. James Place	180	14	70	200	550	750	950	90	100
17	Community Chest 2									
18	Tennessee Avenue	180	14	70	200	550	750	950	90	100
19	New York Avenue	200	16	80	220	600	800	1000	100	100
20	Free Parking									
21	Kentucky Avenue	220	18	90	250	700	875	1050	110	150
22	Chance 2									
23	Indiana Avenue	220	18	90	250	700	875	1050	110	150
24	Illinois Avenue	240	20	100	300	750	925	1100	120	150
25	B & O Railroad	200							100	
26	Atlantic Avenue	260	22	110	330	800	975	1150	130	150
27	Ventnor Avenue	260	22	110	330	800	975	1150	130	150
28	Water Works	150							75	
29	Marvin Gardens	280	24	120	360	850	1025	1200	140	150
30	Go to Jail									
31	Pacific Avenue	300	26	130	390	900	1100	1275	150	200
32	North Carolina Avenue	260	22	110	330	800	975	1150	130	150
33	Community Chest 3									
34	Pennsylvania Avenue	320	28	150	450	1000	1200	1400	160	200
35	Short Line Railroad	200							100	
36	Chance 3									
37	Park Place	350	35	175	500	1100	1300	1500	175	200
38	Luxury Tax									
39	Boardwalk	400	50	200	600	1400	1700	2000	200	200
40	Go									

Tabela 1: Valores Oficiais do Monopoly Clássico

User story 3 – Status

Na sua vez, antes de usar o comando “jogar”, o jogador tem a opção de usar o comando “status”. Através dele, o jogador obtém informação textual sobre seu estado no jogo. A maneira como o texto será arrumado e formatado pode ser feita como você quiser, mas ele deve conter o seguinte: o nome do jogador, a cor do peão, o nome da posição atual no tabuleiro, quanto dinheiro o jogador possui, uma lista dos títulos que o jogador possui (seus pertences) com as informações relativas a cada título (cor do grupo se for propriedade, aluguel das propriedades, valor de corridas das ferrovias, etc.). Após mostrar a informação, o programa pede para o jogador entrar com um comando, da maneira usual (através da user story 2).

Exemplo do console:

A jogada de ana começou.

Comandos disponíveis: [jogar][status][sair]

Entre com um comando: **status**

O status de ana - azul é o seguinte:

Situado na posição 39 - Boardwalk

Possui \$304

Títulos:

[Baltic Avenue] - propriedade roxa, aluguel 4

[B & O Railroad] - ferrovia, corrida 25

[Pennsylvania Avenue] - propriedade verde, aluguel 28

Comandos disponíveis: [jogar][status][sair]

Entre com um comando:

User story 4 – Compra de Propriedades e Ferrovias

Quando o peão cai em uma propriedade ou ferrovia sem dono, o jogador tem a opção de comprar o título da propriedade ou ferrovia pelo preço indicado na tabela 1. Se o jogador tiver o dinheiro necessário para a compra, o programa deve mostrar uma mensagem indicando o nome do item (propriedade ou ferrovia), o tipo (um texto, que pode ser a cor do grupo, se for uma propriedade, ou “ferrovia” para as ferrovias), o preço e o dinheiro que o jogador possui, e perguntar se o jogador quer comprar ou não. Se o jogador não quiser comprar o item, ou não tiver dinheiro suficiente, o item fica disponível para ser comprado depois por qualquer jogador (inclusive o que não o comprou) em outra jogada.

Exemplo do console:

A jogada de ana começou.

Comandos disponíveis: [jogar][status][sair]

Entre com um comando: **jogar**

O jogador ana tirou 6,5 e o peão avançou para 11 - St. Charles Place.

A título da propriedade St. Charles Place está disponível por \$140.

Ana, você possui \$304.

Você deseja comprar St. Charles Place (Sim/Não)? **S**

User story 5 – Sorte/Revés e Cofres Comunitários

Você deve implementar os lugares do tipo Sorte/Revés (Chance) e os Cofres Comunitários (Community Chests). Sempre que um jogador cair em Sorte/Revés ou Cofres Comunitários, uma carta aleatória é selecionada e a ação indicada na carta ocorre. A lista de cartas e as ações associadas estão nas tabelas 2 e 3. Uma mensagem deve mostrar a carta sorteada e o jogador deve pressionar qualquer tecla para continuar. Veja mais detalhes nas regras abaixo. Se o jogador receber uma carta de saída livre da prisão, ela deve ser informada quando o comando “status” for usado.

A jogada de ana começou.

Comandos disponíveis: [jogar][status][sair]

Entre com um comando: **jogar**

O jogador ana tirou 4,3 e o peão avançou para 7 – Chance 1.
O banco lhe pagou dividendos de \$50.

Número da carta	Nome da carta	Descrição	Observações
1	Avance para o Ponto de Partida (Go)	Receba \$200	Isso não é um bônus, ou seja, os \$200 referem-se ao valor que o jogador recebe normalmente ao passar pelo Ponto de Partida.
2	Erro do banco em seu favor	Receba \$200	
3	Taxa do médico	Pague \$50	
4	Abertura da Grande Ópera	Receba \$50 de cada jogador pelas entradas	
5	Da liquidação fora de estoque	Receba \$45	
6	Saia livre da prisão, sem pagar	Esta carta pode ser mantida até o uso ou venda.	Veja a user story sobre a prisão
7	Vá para a prisão	Vá direto para a prisão – Não passe pelo ponto de partida – Não receba \$200	
8	Restituição do Imposto de Renda	Receba \$20	
9	Aniversário do seguro de vida	Receba \$100	
10	Pague o hospital	Pague \$100	
11	Pague taxa da escola	Pague \$150	
12	Receba por seus serviços	Receba \$25	
13	Aniversário dos fundos de Natal	Receba \$100	
14	Você tirou o segundo lugar no concurso de beleza	Receba \$10	
15	Você herdou	Receba \$100	
16	Você deve pagar os reparos da rua	Pague \$40 por cada casa, \$115 por hotel	Isso se aplica apenas às casas e hotéis que o jogador possui; ele não precisa pagar pelas casas e hotéis dos outros jogadores.

Tabela 2: Cartas dos Cofres Comunitários

Número da carta	Nome da carta	Descrição da carta	Observações
1	Avance para o Ponto de Partida	Receba \$200	
2	Avance para	Illinois Avenue	O jogador não recebe \$200 se passar pelo Ponto de Partida.
3	Avance para St. Charles Place	Se passar pelo ponto de partida, receba \$200	
4	Avance o peão para o serviço público mais próximo	Se não tiver dono você pode comprá-lo do banco. Se tiver, lance os dados e pague ao dono 10 vezes o resultado do lançamento.	O jogador não recebe \$200 se passar pelo Ponto de Partida.
5	Avance o peão para a ferrovia mais próxima	Pague ao dono duas vezes o valor que ele deveria receber normalmente. Se a ferrovia não tiver dono, você pode comprá-la do banco.	O jogador não recebe \$200 se passar pelo Ponto de Partida.

6	O banco lhe pagou dividendos de	\$50	
7	Volte 3 espaços		
8	Vá diretamente para a Prisão	Não passe pelo ponto de partida, não receba \$200.	
9	Faça a manutenção de todas as suas propriedades	Para cada casa pague \$25, para cada hotel \$10.	Somente para casas e hotéis que o jogador construiu.
10	Pague imposto de pobreza de	\$15	
11	Esta carta pode ser mantida até seu uso ou venda	Get Out Of Jail Free	Veja user stories sobre Prisão.
12	Dê uma volta na Reading	Se você passar pelo Ponto de Partida receba \$200	
13	Dê uma caminhada no Board Walk	Avance o peão para Board Walk	
14	Você foi eleito presidente do tabuleiro	Pague \$50 para cada jogador	
15	Seus empréstimos e investimento fizeram aniversário	Receba \$150	

Tabela 3: Cartas dos Sorte/Revés

Regras para cofres comunitários e sorte/revés.

- Há duas pilhas de cartas: uma para os cofres comunitários e outra para os sorte/revés. Os três lugares de sorte/revés acessam a mesma pilha, assim como os três cofres comunitários acessam a mesma pilha.
- Se o jogador cair em um Cofre Comunitário ou Sorte/Revés, ele recebe uma carta aleatória indicando o evento que deve ocorrer (perda ou ganho de dinheiro, mover-se para outro lugar, etc.) – veja as tabelas 2 e 3 acima. As duas pilhas de cartas são embaralhadas quando o jogo começa. A carta que é tirada deve voltar para o fim da pilha, isto é, não deve ser tirada novamente até que todas as outras tenham sido tiradas. Quando todas as cartas saírem, a pilha não é embaralhada de novo – as cartas voltam na mesma ordem que vieram desde o começo.
- Se o jogador cair no Sorte/Revés da posição 2 ou no Cofre Comunitário da posição 7 em uma jogada em que o peão passou pelo ponto de partida, e o jogador receber uma carta “Avance para o Ponto de Partida” (Cofre Comunitário número 1 e Sorte/Revés número 1), ele recebe \$200 por ter passado pelo ponto de partida, mais \$200 por ter retornado para lá.

User story 6 – Prisão

Adicionar a funcionalidade da prisão. O jogador pode ir para a prisão se cair em Vá Para a Prisão (Go To Jail), receber uma carta no Sorte/Revés ou Cofre Comunitário mandando ir para a prisão, ou tirar números iguais nos dados por três vezes seguidas. Quando o jogador está na prisão, sua primeira jogada depois de ser preso começa com as seguintes opções de comandos, além de “status”, “jogar” e “sair”: “pagar”, e “carta” (se disponível). Se o jogador escolher “pagar”, ele garante sua saída da prisão – ao digitar “jogar” após o pagamento, a jogada ocorre normalmente como se o jogador estivesse em um lugar qualquer (mas não pode jogar novamente se tirar números iguais nos dados). A mesma garantia é feita quando o jogador escolhe usar uma carta – comando “carta” (se tiver uma, que pode ser conseguida no Sorte/Revés ou nos Cofres Comunitários). Se o jogador escolher “jogar” sem haver pago ou usado uma carta, a jogada só se realiza normalmente se o jogador conseguir tirar números iguais nos dados. Senão, uma mensagem aparece indicando o que aconteceu, e a jogada termina, sendo a vez passada para o próximo jogador. Imediatamente após a terceira jogada sem conseguir sair da prisão (após um terceiro comando “jogar” sem sucesso), o jogador é obrigado a pagar \$50. Uma mensagem deve aparecer indicando isso, e então a jogada se

completa normalmente, com o peão avançando a quantidade de lugares indicada pela soma dos dados do último lançamento.

Exemplos do console:

A jogada de ana começou.

ana está na prisão.

Comandos disponíveis: [pagar][carta][jogar][status][sair]

Entre com um comando: **pagar**

Comandos disponíveis: [jogar][status][sair]

Entre com um comando: **jogar**

O jogador ana tirou 3,2 e o peão avançou para 15 - Pennsylvania Railroad.

Outro exemplo:

A jogada de ana começou.

ana está na prisão.

Comandos disponíveis: [pagar][carta][jogar][status][sair]

Entre com um comando: **jogar**

O jogador ana tirou 3,2 e não conseguiu sair da prisão.

A jogada de joao começou

Comandos disponíveis: [jogar][status][sair]

Entre com um comando:

Regras para a prisão:

- Sempre que o jogador tirar números iguais nos dados, ele ganha uma jogada adicional, após a jogada atual.
- Se o jogador tirar números iguais três vezes seguidas, a jogada acaba imediatamente (antes de o peão avançar), o jogador vai para a prisão sem receber dinheiro, e a jogada termina.
- Se o jogador cair na Prisão, nada acontece (apenas visitando).
- Se o jogador cair em Vá Para a Prisão (“Go To Jail”) ou pegar uma carta “Vá diretamente para a Prisão” no Sorte/Revés, vai para a prisão sem receber dinheiro, e perde sua jogada, mesmo que tenha tirado números iguais nos dados (não joga de novo).
- Um jogador na prisão continua recebendo aluguel.
- Para sair da prisão, o jogador deve tirar números iguais nos dados. Se conseguir, o peão anda o número correspondente de espaços, mas o jogador não ganha uma jogada extra.
- O jogador também pode sair da prisão pagando \$50 ao banco ou usando uma carta de saída da prisão, que pode ser adquirida em Sorte/Revés ou em Cofres Comunitários. Ele então lança os dados e anda o número de espaços correspondente à soma do valor dos dados, mas não joga de novo se conseguir números iguais.
- Se a terceira tentativa de sair da prisão for frustrada, o jogador é obrigado a pagar \$50 e o peão avança o número correspondente ao último lançamento dos dados.

User story 7 – Serviços Públicos

Se o jogador cair em um serviço público (Water Works e Electric Company) sem dono, ele tem a opção de comprá-lo pelo preço indicado na tabela 1 – mostrar o nome do serviço público, o preço, e perguntar se o jogador deseja comprá-lo. Se o jogador pedir o status do jogo, o texto deve incluir informações sobre os serviços públicos que possui (nome, tipo; serviço público). Quando um jogador cai em um serviço público com dono, ele deve pagar ao dono o resultado do lançamento dos dados multiplicado por um fator que pode ser 4 ou 10, caso o dono possua apenas um ou os dois serviços públicos, respectivamente. Informar o que aconteceu antes de terminar a jogada.

User story 8 – Construção de casas

Em sua jogada, antes de lançar os dados, um jogador pode usar o comando “construir”, que é usado para construir casas e/ou hotéis em suas propriedades. Esse comando só fica disponível se o jogador detiver monopólio sobre um grupo de propriedade (possui todas da mesma cor), se houver casas e/ou hotéis disponíveis no banco, se o jogador tiver dinheiro suficiente para construir (veja tabela 1) e se ela ainda puder construir em suas propriedades (se já tiver hotéis em todas as propriedades de um monopólio, ele não pode mais construir nesse grupo). Use as regras para construção abaixo. Se o jogador pedir seu status no jogo, o texto deve conter informações sobre casas: para cada propriedade, exibir o número de casas (ou hotel) construídos, se houver. Quando o jogador entra com o comando “construir”, uma lista numerada de todas as propriedades onde se pode construir, com o dinheiro correspondente necessário para construir a próxima casa ou hotel (o número de casas já construídas deve ser informado) é mostrado, juntamente com o dinheiro que o jogador possui. O jogador deve então entrar com o número correspondente à propriedade na qual quer construir, ou 0 para cancelar a operação. Quando ele constrói, a lista é atualizada, mostrada, e o programa pede outro número. O processo continua até que o jogador entre com 0, ou até que não haja mais propriedades onde se possa construir. Se o jogador cancelar (entrando 0), ele pode voltar a digitar “construir” se ainda puder, desde que seja antes de “jogar”.

A jogada de ana começou.

Comandos disponíveis: [construir][vender][jogar][status][sair]

Entre com um comando: **construir**

ana possui \$530

Escolha onde quer construir:

1 - Park Place tem 0 casa(s) construídas, casa custa \$200

2 - Boardwalk tem 1 casa(s) construídas, casa custa \$200

Entre com o número da propriedade (0 para sair): **1**

ana possui \$330

Escolha onde quer construir:

1 - Park Place tem 1 casa(s) construídas, casa custa \$200

2 - Boardwalk tem 1 casa(s) construídas, casa custa \$200

Entre com o número da propriedade (0 para sair): **0**

Comandos disponíveis: [vender][jogar][status][sair]

Entre com um comando:

Regras para a construção de casas:

- Há um total de 32 casas e 12 hotéis disponíveis no banco para venda.
- Em sua jogada, antes de lançar os dados, um jogador pode construir qualquer número de casas (ou hotéis) em qualquer de suas propriedades, desde que:

1. ele detenha monopólio do grupo que inclui a propriedade onde quer construir;
 2. ele tenha dinheiro suficiente para comprar as casas e hotéis que deseja (use a tabela 1);
 3. haja casas ou hotéis suficientes no banco para venda;
 4. ele ainda possa construir em suas propriedades (não se pode construir mais em um grupo com um hotel em cada propriedade);
- Para construir um hotel em uma propriedade, 4 casas devem ter sido previamente construídas no local.
 - Quando um hotel é construído em uma propriedade, as 4 casas são retornadas ao banco (não se recebe dinheiro por isso) e ficam disponíveis para serem compradas pelos jogadores (inclusive o que construiu o hotel).
 - Apenas um hotel pode ser construído em cada propriedade.
 - As casas devem ser construídas de modo igualitário dentro de um grupo: primeiro, uma casa é construída em cada propriedade, então a segunda casa de cada propriedade pode ser construída, e assim por diante (ou seja, dentro de um grupo, não pode haver uma propriedade com 3 casas, e outra com 1 casa apenas); o mesmo serve para hotéis – um hotel só pode ser construído se todas as propriedades do grupo já tiverem a quarta casa construída.
 - A regra anterior também vale para a venda de casas na user story 9 – não deve haver uma diferença de duas casas entre quaisquer duas propriedades dentro de um grupo. Primeiramente, os hotéis devem ser vendidos, então as quartas casas de cada propriedade, depois as terceiras, e assim por diante.
 - Se um jogador cair em uma propriedade cujo dono detém monopólio sobre o grupo que a contém, e nenhuma casa estiver construída no local, ele deve pagar ao dono o dobro do aluguel indicado na tabela 1.
 - Cada propriedade tem um aluguel associado à quantidade de casas (ou hotel) construídos; se um jogador cair em uma propriedade com casas construídas (ou hotel), deve pagar o aluguel correspondente (use a tabela 1).

Use story 9 – Venda de casas

Se o jogador tiver qualquer casa ou hotel construído, ele deve, durante sua jogada, poder vendê-los de volta para o banco. O comando “vender” fica disponível nesta condição. Uma lista numerada igual à usada para a construção de casas aparece, e o jogador escolhe a casa ou hotel que quer vender digitando o número correspondente, ou 0 para cancelar. O processo se repete até que o jogador digite 0, ou até que não haja casas adicionais a serem vendidas. O jogador recebe a metade do valor de compra pela casa ou hotel vendida (veja tabela 1). O comando deve continuar disponível se o jogador não tiver vendido todas as suas casas.

A jogada de ana começou.

Comandos disponíveis: [vender][jogar][status][sair]

Entre com um comando: **vender**

ana possui \$330

Escolha onde quer vender:

1 – Park Place tem 1 casa(s) construídas, você recebe \$100

2 – Boardwalk tem 1 casa(s) construídas, você recebe \$100

Entre com o número da propriedade (0 para sair): **1**

ana possui \$430

Escolha onde quer vender:

1 - Park Place tem 0 casa(s) construídas, você recebe \$100

2 - Boardwalk tem 1 casa(s) construídas, você recebe \$100

Entre com o número da propriedade (0 para sair): 0

Comandos disponíveis: [vender][jogar][status][sair]

Entre com um comando:

User story 10 – Hipoteca

Permita que títulos de propriedades, ferrovias e serviços públicos possam ser hipotecados. Todo jogador que detiver um título tem disponível em sua jogada um novo comando chamado “hipotecar”. Quando o jogador entrar esse comando, uma lista numerada similar à da compra de casas aparece, mostrando os títulos que podem ser hipotecados (veja regras e requisitos abaixo). O jogador deve entrar com o número do título que quer hipotecar, ou 0 para cancelar. O valor das hipotecas de cada título pode ser encontrado na tabela 1. O processo continua até que 0 seja digitado, ou não houver mais títulos que possam ser hipotecados.

As informações sobre títulos de propriedades, ferrovias e serviços públicos hipotecados também devem estar disponíveis se o jogador pedir o status do jogo (apenas dizer para cada um de seus títulos se está hipotecado ou não).

A jogada de ana começou.

Comandos disponíveis: [hipotecar][construir][jogar][status][sair]

Entre com um comando: **hipotecar**

Escolha o título que quer hipotecar:

1 - Park Place pode ser hipotecado por \$175

2 - Boardwalk pode ser hipotecado por \$200

Entre com o número do título (0 para sair): 1

Escolha o título que quer hipotecar:

1 - Boardwalk pode ser hipotecado por \$200

Entre com o número do título (0 para sair): 0

Regras para hipotecas:

- Jogadores só podem hipotecar propriedades que não tenham habitações construídas (se quiserem hipotecar, primeiro terão que vendê-las ao banco).
- Jogadores não podem construir casas em propriedades hipotecadas (se quiserem, primeiro devem desfazer a hipoteca), nem em qualquer propriedade cujo grupo contenha propriedades hipotecadas. Quanto todas as propriedades hipotecadas dentro de um grupo estiverem desfeitas, o jogador pode voltar a construir habitações.
- Propriedades, ferrovias e serviços públicos hipotecados não geram receita para seus donos. Entretanto, a regra do aluguel dobrado em um monopólio sem casas continua válida para as outras propriedades do grupo que não estejam hipotecadas.

User story 11 – Quitação de hipoteca

Permita que a hipoteca de títulos possa ser quitada. Um comando chamado “desipotocar” deve estar

disponível para todo jogador que tiver títulos hipotecados. Quando usado, este comando abre uma lista numerada de títulos hipotecados (cuja hipoteca o dono tem dinheiro para quitar), juntamente com o dinheiro que o jogador possui. O jogador quita as hipotecas digitando o número correspondente. O valor é o da hipoteca + 10% de juros. O processo é repetido até que 0 seja digitado, ou até que não haja mais títulos hipotecados.

Exemplo do console:

A jogada de ana começou.

Comandos disponíveis: [hipotecar][desipotectar][construir][jogar]
[status][sair]

Entre com um comando: `desipotectar`

ana possui \$430

Escolha o título que quer desipotectar:

1 - Park Place pode ser desipotecado por \$192

Entre com o número do título (0 para sair): `1`

Comandos disponíveis: [hipotecar][construir][jogar][status][sair]

Entre com um comando:

User story 13 – Evitar falência

Sempre que o dinheiro de um jogador ficar abaixo de 0, um evento especial ocorre que interrompe o jogo e muda a atenção para o jogador ameaçado de falência. Uma jogada especial é dada para este jogador para que ele possa vender habitações e hipotecar títulos para tentar continuar no jogo. Os comandos disponíveis para ele são “sair” (sai do jogo), “hipotecar” (se ainda tiver títulos para hipotecar), “vender” (se tiver habitações construídas), e “status”. Ele deve usar os comandos hipotecar e vender para tentar pagar suas dívidas. Se “hipotecar” e “vender” se tornarem indisponíveis e o dinheiro do jogador ainda estiver abaixo de zero, ele é excluído do jogo, e uma mensagem aparece dizendo isso. Ao conseguir fazer o dinheiro voltar a ficar acima de zero, um comando chamado “evitar” se torna disponível, e faz com que o jogo recomece do ponto em que parou.

Funcionalidades extras não-essenciais (mas que rendem pontos, se você implementá-las, além de fazer o essencial)

Permitir regras customizáveis – Isso poderia ser feito através de um arquivo de configuração. Através dele, o usuário poderia personalizar o jogo, modificando as regras do Monopoly. É aqui que a gente vê se o seu programa está bem projetado (em um Monopoly bem projetado, é fácil implementar essa funcionalidade). Algumas regras customizáveis poderiam ser:

- Regra da construção igualitária – sim ou não
- Coletar o dobro do dinheiro ao cair exatamente no Ponto de Partida – sim ou não
- Regra da Parada Livre – sim ou não (quando o jogador cai na Parada Livre, ele ganha uma bolada fixa).
- Casas por hotel - 4 or 5 (número de casas necessárias antes de construir um hotel em uma propriedade)
- Casas totais - 12, 32, 60 or 88
- Hotéis totais - 4, 12, 16 or 22
- Valor da bolada da Parada Livre - 0, 250, 500, 750 (dinheiro que o jogador recebe ao cair em Parada Livre)

- Dinheiro inicial - 500, 100, 1500, 2000
- Salário - 0, 100, 200, 400 (dinheiro que o jogador recebe ao passar pelo ponto de partida)
- Valor do Imposto de Renda (\$) - 0, 100, 200, 400 (valor que o jogador paga ao cair no Imposto de Renda)
- Valor do Imposto Especial (\$) - 0, 75, 150, 300 (valor que o jogador paga ao cair no Imposto Especial)
- Número máximo de rodadas na Prisão - 1, 2, 3, 4
- Valor da saída da Prisão - 0, 50, 100, 200
- Taxa de juros da hipoteca - 0, 5, 10, 20
- Títulos distribuídos no início do jogo - 0, 2, 4, todos (títulos podem ser distribuídos no início do jogo)

Permitir salvar/abrir jogos – comandos adicionais (que poderiam ser usados a qualquer momento) chamados “salvar” e “abrir”. Através deles, o estado de um jogo pode ser salvo em um arquivo, e depois aberto.

Implementar jogadores artificiais – na criação do jogo, cada jogador seria marcado como humano ou artificial. Um jogador artificial é comandado pelo computador, joga automaticamente, e pode ter diferentes níveis de inteligência.

Implementar jogabilidade em rede/online – o Monopoly pode ser jogado em rede, cada jogador em seu computador. Um jogador teria que esperar enquanto os outros fazem suas jogadas. Funcionalidades interessantes poderiam ser trocas entre jogadores a qualquer momento e um leilão baseado em tempo, em que os jogadores podem dar lances a qualquer momento.

A Entrega do Projeto

O projeto será entregue de forma incremental. Serão 3 milestones. Para cada milestone você deve entregar:

- Código-fonte documentado
- Javadoc do código-fonte
- Relatório impresso e em arquivo eletrônico, de até 10 páginas, contendo
 - Um percentual de participação de cada integrante da dupla no desenvolvimento do trabalho
 - A descrição do seu projeto
 - Modelo e explicação do design (projeto)
 - Padrões utilizados, padrões que poderiam ser utilizados
 - Pontos fortes e pontos fracos do seu projeto.

Regras para entrega (devem ser rigorosamente seguidas):

- Entregue um único arquivo zip (ou rar). O nome do arquivo zip deve ser "joao-maria.zip" (ou.rar). “joao-maria” são os nomes dos integrantes da dupla.
- Ao extrair o arquivo zip (ou rar) tudo deve cair no lugar certo para ajudar minha tarefa de testar e verificar seu trabalho. Isso significa que não vou tentar entender porque as coisas não funcionam.
- Posso extrair o arquivo em qualquer diretório da minha máquina.
- Use apenas nomes de arquivos *relativos*, pois nomes absolutos que poderão existir na sua

máquina poderão não existir na minha máquina.

- Use ant com os seguintes alvos: limpa, compila, doc, testa. Só garanto que o PATH de minha máquina permite executar ant, e os comandos java (javac, java, javadoc, ...).
- O arquivo build.xml deve estar na raiz do seu zip(rar).
- O comando "ant testa" deve chamar testa.bat. Esse comando executa todos os testes de aceitação. Para poder testar a persistência corretamente, os testes devem ser chamados individualmente. Um template de testa.bat será entregue juntamente com este documento. Se preferir, pode fazer no Linux e entregar um testa.sh equivalente, mas eu testarei no Windows, então entregue um testa.bat.
- Para testar seu programa, usarei os seguintes comandos: ant limpa compila doc testa.
- Posso criar novos testes de aceitação e usá-los normalmente.
- Para ter certeza que tudo funcione adequadamente, teste os passos acima numa máquina diferente daquela onde você desenvolveu o software.
- Testarei com JDK 1.5 ou superior.
- *Não topo* editar build.xml ou arquivos .bats para acertar o classpath ou qualquer outra coisa. Posso editar o .bat apenas para inserir novos testes de aceitação.

Milestone 1

Implemente as User Stories 1, 2, 3 e 4.

A data limite para entrega é dia 26/11/2009.

Milestone 2

Implemente as User Stories 5, 6, 7, 8, 9.

A data limite para entrega é dia 23/12/2009.

Milestone 3

Você vai receber o material produzido no milestone 2 do projeto de *outro grupo* e deverá implementar as user stories 10, 11 e 12, obedecendo às regras correspondentes. Mude o mínimo necessário no programa que você recebeu para implementar a User Story. Você deve avaliar os seguintes pontos (incluir sua avaliação no relatório):

- Comente as fraquezas do design que você recebeu.
- Comente as virtudes do design que você recebeu.
- Comente as dificuldades que você teve para implementar as user stories 10, 11, 12.
- Descrever o refatoramento que você efetuou.
- Dê uma nota de 0 a 10 para cada uma das áreas abaixo:
 - Qualidade da documentação
 - Qualidade do design
 - Qualidade do código

A data limite para entrega é dia 31/01/2010.

Avaliação

Pesos para avaliação de cada milestone:

- Compilação (0% - se não compilar, não aceito o projeto) . Mesmo que corrigido depois, haverá penalização de 20% do valor final da nota.
- Execução dos testes de aceitação (50%). Serão considerados os testes de aceitação relacionados às User Stories solicitadas.
- Qualidade da documentação (20%)
- Qualidade do design (20%)
- Qualidade do relatório (10%)

Passos para a correção. Vou corrigir o projeto da seguinte forma:

- Extrairei o arquivo .zip (rar) e executarei “ant limpa”. Vou verificar se os objetos foram apagados. Depois executarei “ant compila”. O programa deve compilar. Se não compilar a correção pára e a dupla será avisada.
- Testes de aceitação: executarei “ant testa”. Observarei se todos os testes de aceitação executam. Para a user story 1, os testes estão em us1.txt e assim por diante. Vou retirar pontos proporcionais à quantidade de testes não executados. Por exemplo, se são passados 18 testes, significa que a execução de 18 testes corretamente, implica em 100% da nota correspondente a este item. Se somente 9 dos 18 testes executam corretamente, então você terá 50% da nota referente a este item. Ou seja, será usada uma regra de três simples.
- Javadoc. Vou observar se há descrição dos pacotes, classes, métodos, parâmetros, valor de retorno e exceção. Claro que não vou olhar todo o javadoc. Vou dar uma olhada geral.
- Relatório. Vou ler o relatório e ver se atende à solicitação feita na Seção “A Entrega do Projeto”.
- Design. Vou observar as principais classes modeladas a partir de seu relatório e observar no código suas relações, acoplamentos, coesão dos principais métodos.
- Darei uma nota para o projeto e multiplicarei por 2. Depois usarei o percentual de esforço de cada participante, descrito no relatório, para calcular a nota individual. Por exemplo, se eu considerar que o projeto como todo tem uma nota final 7 e os dois integrantes tiveram 100% de participação, cada um ficará com 7. Se um dos participantes teve 30% e o outro 70%, a nota será 4,2 e 9,8, respectivamente. A nota máxima de um integrante é 10.

Testes de Aceitação

Você está recebendo juntamente como este documento, os testes de aceitação que você deve rodar para saber se o seu jogo está atendendo direitinho os requisitos. Você usará o EasyAccept para rodar os testes (acesse o site <http://easyaccept.sourceforge.net/>). Para poder rodar os testes, vocês precisarão criar uma Façade que acessa a business logic do seu jogo segundo a linguagem de script apresentada na próxima seção.

Linguagem de Script

Para acessar a business logic de seu programa e poder realizar os testes de aceitação, você precisará implementar uma linguagem de script que consiste nos comandos abaixo. Esses são os comandos usados nos testes de aceitação (us1.txt, us2.txt, etc.). Cada um deles deve corresponder a um

método na sua façade.

Obs.: por enquanto apenas estão listados os comandos usados nos testes das user stories 1,2,3 e 4. O restante pode ser extraído dos testes de aceitação.

```
<void> createGame numPlayers=<int> playerNames=<String>  
tokenColor=<String>
```

Obs.: o vetor de nomes e o de cores será passado através de uma String, com o seguinte formato: começa com abre chaves "{", seguida dos itens, que são separado por vírgulas, sem aspas; a String deve terminar com fecha chaves "}". Ex.: createGame numPlayers=2 playerNames="{ana,joao}" tokenColors="{red,blue}"

```
<int> getNumberOfPlayers  
<String> getPlayerToken playerName=<String>  
<int> getPlayerMoney playerName=<String>  
<int> getPlayerPosition playerName=<String>  
<String> getPlayerDeeds playerName=<String>
```

Obs.: o vetor de títulos deve ser retornado através de uma String, da seguinte forma: começa com abre chaves "{", seguida dos itens, que são separado por vírgulas, sem aspas; a String deve terminar com fecha chaves "}". Ex.: expect "{Marvin Gardens,Boardwalk}" getPlayerDeeds playerName="ana"

```
<String> getCurrentPlayer playerName=<String>  
<String> getPlaceName placeID=<int>  
<String> getPlaceGroup placeID=<int>  
<String> getPlacePrice placeID=<int>  
<String> getPlaceOwner placeID=<int>  
<String> getPropertyRent placeID=<int>  
<void> rollDice firstDieResult=<int> secondDieResult=<int>  
<String> getCurrentPlayer  
<String> getCommands
```

Obs.: o vetor de comandos deve ser retornado através de uma String, da seguinte forma: começa com abre chaves "{", seguida dos itens, que são separado por vírgulas, sem aspas; a String deve terminar com fecha chaves "}". Ex.: expect "{roll,status,quit}" getCommands

```
<void> setAutomaticBuying auto=<int>  
<boolean> gameIsOver  
<void> buy  
<void> quitGame
```

Glossário

Propriedade – lugar do tabuleiro que rende dinheiro para os donos sob a forma de aluguéis, cada vez que os outros jogadores caem nele. Nelas, casas ou hotéis podem ser construídas para aumentar o valor do aluguel. Formam grupos de 3, com cores associadas.

Ferrovia – lugar do tabuleiro que rende dinheiro para o dono sob a forma de corridas, cada vez que os outros jogadores caem nele. O valor da corrida aumenta de acordo com a quantidade de ferrovias que o dono possui. Há quatro ferrovias no total.

Serviço Público – lugar do tabuleiro que rende dinheiro para o dono sob a forma de serviços, cada vez que os outros jogadores caem nele. O valor devido depende do número obtido no lançamento dos dados e da quantidade de serviços que o dono possui. Há dois serviços públicos: electric company e water works.

Cofre Comunitário – lugar do tabuleiro que gera para o jogador que cai nele uma carta que contém um evento aleatório, na maioria das vezes envolvendo ganho ou perda de dinheiro.

Sorte/Revés – lugar do tabuleiro que gera para o jogador que cai nele uma carta que contém um evento aleatório, na maioria das vezes envolvendo avanços ou recuos no tabuleiro.

Imposto de Renda – lugar do tabuleiro em que o jogador deve pagar o valor de \$200 ao banco.

Imposto de Riqueza – lugar do tabuleiro em que o jogador deve pagar o valor de \$75 ao banco.

Título – é um documento que representa uma propriedade, ferrovia ou serviço público – lugares compráveis do tabuleiro. Os jogadores que detêm títulos de um lugar são os donos do lugar.

Habitação – casas ou hotéis, que podem ser construídas em propriedades.

Peão – a peça representativa de um jogador que se move pelo tabuleiro, ocupando os lugares.

Lugar – cada um dos 40 espaços do tabuleiro onde os peões se movimentam.

Jogada – é a “vez” de um jogador, a chance que ele tem de realizar ações através de comandos e de lançar os dados para avançar no tabuleiro, ocupando um novo lugar.

Rodada – um conjunto de jogadas, formado por 1 jogadas de cada jogador.