



CIU: 74.800

2018

SUMÁRIO

1. INTRODUÇÃO	3
1.1 FINALIDADE	3
2. DESCRIÇÃO GERAL	3
2.1 DIVISÃO DAS REGRAS DE NEGÓCIOS	3
2.2 DESCRIÇÃO DAS CLASSES DAS REGRAS DE NEGÓCIOS	3
2.2.1 PACKAGE - CONTROL.....	3
2.2.1.1 CLASS – CONTROLADORDEACOESDOJOGADOR	4
2.2.1.2 CLASS – CONTROLADORDEATAQUE	4
2.2.1.3 CLASS – CONTROLADORDEJOGO	4
2.2.1.4 CLASS – CONTROLADORDEMILITARESAPOIOS	4
2.2.1.5 CLASS – CONTROLADORDEOBJETIVO	4
2.2.1.6 CLASS – CONTROLADORESCOLHERJOGADORES	4
2.2.1.7 CLASS – INICIALIZADORDEJOGO.....	5
2.2.1.8 CLASS – INTERFACEINTERFACEGRÁFICA	6
2.2.1.9 CLASS – INTERFACEMOSTRAESTATÍSTICAS	6
2.2.2 PACKAGE – MODEL.MASTER.....	6
2.2.2.1 CLASS – JOGADOR.....	7
2.2.2.2 CLASS – OBJETIVO	7
2.2.2.3 CLASS – SORTEIO	7
2.2.3 PACKAGE – MODEL.MILITARES.....	7
2.2.3.1 CLASS – MILITAR	7
2.2.3.2 CLASS – CABO.....	7
2.2.3.3 CLASS – SARGENTO.....	7
2.2.3.4 CLASS – SOLDADO	8
2.2.3.5 CLASS – TENENTE	8
2.2.4 PACKAGE – MODEL.MUNDO.....	8
2.2.4.1 CLASS – MUNDO	8
2.2.4.2 CLASS – CONTINENTE	8
2.2.4.3 CLASS – TERRITÓRIO.....	8
2.2.5 PACKAGE – VIEW.....	8
2.2.5.1 CLASS – ACOESDOJOGADOR.....	9
2.2.5.2 CLASS – ESCOLHERJOGADOR	9
2.2.5.3 CLASS – ESCOLHERQUANTIDADEJOGADOR	9
2.2.5.4 CLASS – PLAYERINFO	9
2.2.5.5 CLASS – RESULTADOATAQUE	10
2.2.5.6 CLASS – SELECIONARMILITAR	10
2.2.5.7 CLASS – SELECIONARTERRITORIO	10
3. TESTES	10
4. CONCLUSÃO	10
5. BIBLIOGRAFIA.....	11

1.Introdução

1.1 Finalidade

Este documento tem como finalidade, descrever e criar um cenário de como o jogo War funciona, suas características e também seus objetivos. O jogo em si é disputado através de um mapa (também podemos chamar de tabuleiro) do mundo, porém é dividido em 6 regiões (Europa, Ásia, África, América do Norte, América do Sul e Oceania). Cada jogador recebe um objetivo e quem completar primeiro o seu será o vencedor. É disputado em rodadas, nas quais os participantes colocam exércitos e atacam outros oponentes. Uma partida pode durar várias horas, com disputas regidas pela estratégia dos jogadores e pela sorte lançada pelos dados.

2.Descrição Geral

Acima, na introdução foi realizado um esboço geral de como funciona o jogo. A partir desta etapa, estaremos descrevendo a implementação em si do jogo e como o mesmo foi distribuído durante o desenvolvimento deste software que simula o jogo e também as características do mesmo em relação a este jogo.

2.1 Divisão das regras de negócios

Abaixo, preferimos dividir em uma tabela de como foi realizada a divisão das regras de negócios (ou podemos também falar como que ficou os pacotes – packages) do jogo. Portanto, abaixo, será descrito cada pacote com suas classes e características.

Nome regra/pacote	Descrição
control	Todas as classes e respectivos métodos de cada uma que realizam o controle do jogo estão dentro deste pacote. Aqui as classes controlam o jogo desde a inicialização das classes, bem como da inicialização do próprio jogo, controle todo fluxo do jogo, bem como ataques, pontuações, distribuição de jogadores, territórios, ganhadores, etc.
model.master	Já aqui neste pacote, o mesmo é responsável por gerenciar as classes de jogadores, os objetivos dos jogadores e também um classe sorteio.
model.militares	Há apenas a implementação das classes de cada militar, são apenas classes abstratas para a formalização de todos os militares que há no jogo.
model.mundo	Aqui neste pacote, organiza as classes que implementam a parte geográfica do jogo, como Continente, os seus respectivos territórios e o mundo, que no caso do jogo é o próprio tabuleiro.
view	Neste pacote, ele é responsável por armazenar, organizar e gerenciar todas as classes e métodos que estejam relacionados a parte visual do jogo, ou seja, a interface gráfica.

2.2 Descrição das classes das regras de negócios

2.2.1 Package - control

O pacote control, podemos dizer que ele é o principal pacote do sistema que implementa o jogo War, pois neste pacote é aonde tudo que há implementado, aqui se torna real, ou

seja, neste pacote é instanciado os objetos, inicializado o jogo, como a distribuição dos jogadores, distribuição dos continentes, dos territórios e seus respectivos vizinhos. Vejamos então as principais classes deste pacote:

2.2.1.1 Class – ControladorDeAcoesDoJogador

Esta classe controla todas as ações do jogador no jogo quando chegou a sua vez de jogar, há três opções que são: **Atacar, Remanejar ou Passar a vez**. O jogador por vez é obrigado a escolher uma destas opções, porém somente poderá escolher uma opção por vez em que está apto para jogar.

2.2.1.2 Class – ControladorDeAtaque

O controlador de ataque serve pra processar o ataque, os parâmetros de entrada são o território atacante. O território atacado, lista militares atacante e lista militares atacado.

2.2.1.3 Class – ControladorDeJogo

Aqui nesta classe, podemos dizer que é a classe principal do jogo, aonde ela controla todo o fluxo do jogo e possui todas as informações necessárias para fazer o jogo fluir. É ela quem controla os ataques, os jogadores, as pontuações de cada ataque, realiza as jogadas, verifica quem foi o vencedor do duelo atual, inicializa o jogo e conclui o jogo, ou seja, todo o jogo passa por esta classe.

2.2.1.4 Class – ControladorDeMilitaresEApoios

Já esta classe determina e realiza a gerência como está a situação dos continentes após o término de cada rodada, também realizar o posicionamento dos militares que estão sem territórios, determina se o jogador conquistou ou não determinado território. Outro detalhe que ele controla é a questão dos apoios que estão disponíveis com uma certa quantidade para cada território.

2.2.1.5 Class – ControladorDeObjetivo

Os objetivos foram distribuídos por um ID, ou seja, essa é a identificação de cada um deles no jogo, então esta classe tem o controlador de objetivo que verifica se o jogador passado por parâmetro ganhou o jogo ou não de acordo com o seu objetivo que foi definido aleatoriamente no início do jogo.

2.2.1.6 Class – ControladorEscolherJogadores

Já esta classe, é a inicialização do jogo em relação ao momento de definição de jogadores, aonde não será possível haver apenas 1 jogador e/ou mais que 4 players. Também ela controla as cores, pois o jogo distingue os jogadores por cores, então por exemplo, se o jogador de nome “Fischer” escolheu a core preta, o jogador “Max” não poderá escolher esta mesma cor, pois senão como o sistema/jogo irá diferir os jogadores, bem como seus militares, territórios ocupados e respectivas pontuações? Por isto mesmo, para não haver ambiguidade no jogo, esta classe existe e realiza todo este controle. As cores que estão disponíveis são: preto, branco, amarelo e azul.

2.2.1.7 Class – InicializadorDeJogo

Nesta classe todas as outras classes que compõe o jogo são instanciadas/inicializadas, então arquivo dentro são várias determinação para que o jogo comece e a classe de controle do jogo possa controlá-lo diante destes objetos. Primeiramente ela define aleatoriamente os jogadores e após realiza a divisão de territórios, com as respectivas quantidades de territórios pelos 6 continentes e também de acordo com a quantidade de jogadores, também realiza um sorteio para direcionar os territórios para os jogadores.

Após isso, ela inicializa também o tabuleiro e os respectivos continentes do jogo, os continentes são: América do Sul, América do Norte, Europa, África, Ásia e Oceania e no momento desta inicialização dos continentes é distribuído também os militares de apoio que estão disponíveis para cada um.

Ao inicializar os 6 continentes, então a classe também irá inicializar e determinar todos os territórios para cada um desses 6 continentes, vejamos a distribuição dos 24 territórios por continentes:

- América do Sul: Brasil, Argentina e Colômbia.
- América do Norte: Estados Unidos, Canadá, México e Groenlândia.
- África: Egito, Angola, Líbia e África do Sul.
- Europa: Alemanha, Inglaterra, França, Holanda e Portugal.
- Ásia: Rússia, China, Índia, Japão e Coreia do Sul.
- Oceania: Austrália, Nova Zelândia e Indonésia.

Esta classe também determina de quem os territórios são vizinhos e quem são os seus respectivos vizinhos, pois é uma regra do jogo, aonde militares de um território só conseguem atacar territórios que são vizinhos. Portanto, para ficar mais claro a verificação de quem é vizinho de quem, segue abaixo um mapa realizando esta demonstração. Cada aresta significa que aquele território (bandeira de seu respectivo território) é vizinho do qual está interligando.



Ainda nesta classe, ela também realiza o povoamento dos militares e suas respectivas patentes em cada um dos territórios que foram inicializados acima de acordo também com a determinação passada antes de implementação do jogo.

Por fim, esta classe inicializa os objetivos do jogo para após serem sorteados em uma outra classe para cada jogador que tiver o jogo. Abaixo segue uma tabela com cada objetivo que esta classe inicializa:

Id Objetivo	Descrição Objetivo
1	Conquistar na totalidade a América do Sul e a Europa
2	Conquistar a Oceania e mais um continente a sua escolha
3	Conquistar 24 territórios
4	Conquistar a América do norte e a África (continente)
5	Conquistar a Ásia e outro continente da sua escolha
6	Conquistar a Oceania, América do Sul e outro continente da sua escolha
7	Destruir o exército preto, se for você o exército preto ou se ele não estiver no jogo, seu objetivo passa a ser conquistar 24 territórios
8	Destruir o exército branco, se for você o exército branco ou se ele não estiver no jogo, seu objetivo passa a ser conquistar 24 territórios
9	Destruir o exército azul, se for você o exército azul ou se ele não estiver no jogo, seu objetivo passa a ser conquistar 24 territórios
10	Destruir o exército amarelo, se for você o exército amarelo ou se ele não estiver no jogo, seu objetivo passa a ser conquistar 24 territórios

2.2.1.8 Class – InterfaceInterfaceGráfica

Aqui nesta classe é apenas uma chamada das classes do pacote view, aonde ocorre a implementação das classes que estão relacionadas a interface gráfica do jogo War.

2.2.1.9 Class – InterfaceMostraEstatísticas

Esta classe tem como objetivo de mostrar informações após o término do jogo, e essas informações são:

- Qual player foi o vencedor do jogo
- O total de rodadas jogadas
- Os territórios conquistados e que o jogador terminou o jogo
- Número de militares pertencentes aos jogadores após o término do jogo
- Também informará além do jogador vencedor o seu respectivo objetivo

2.2.2 Package – model.master

Já este pacote, já fica apenas com a responsabilidade de alocar as classes de implementações de algumas entidades importantes do jogo, como o jogador, objetivo e a classe responsável pelo sorteio. Vejamos detalhes destas classes:

2.2.2.1 Class – Jogador

Esta classe determina informações do jogador de acordo com as informações que o jogador vai preenchendo via teclado, como nome, cor, definição do objetivo (porém é definida aleatória), os territórios que ele tomará posse e seus respectivos militares.

2.2.2.2 Class – Objetivo

Esta classe realiza a criação/definição dos objetivos do jogos bem como o seu ID, que é o seu identificador. Como já foi citado acima, na classe de inicialização do jogo, é instanciado os objetivos.

2.2.2.3 Class – Sorteio

A classe sorteio é responsável pelos métodos do jogo que realizam os número randômicos, no caso do dado aonde é aplicado para os ataques, os valores são sorteados de acordo com um dado mesmo, aonde o menor número é 1 e o maior número é 6, então os ataques dos militares são sorteados números entre 1 e 6 através dos métodos implementados nesta classe.

Além disso, há um método também que realiza o sorteio dos objetivos para cada jogador durante a inicialização do jogo através do ID que tem um total de 10 objetivos, já listados acima.

2.2.3 Package – model.militares

Este pacote, armazena as classes de todos os tipos de militares do jogo. Segue abaixo uma tabela dos tipos de militares e depois um esboço de cada classe deste pacote:

Tipo militar	Poderio
Soldado	-
Cabo	Aumenta 1 ponto no dado
Sargento	Aumenta 2 pontos no dado
Tenente	Aumenta 3 pontos no dado

2.2.3.1 Class – Militar

É um classe abstrata, aonde há apenas dois atributos para a mesma, porém tem o objetivo de tornar uma base para as demais classes de militares.

2.2.3.2 Class – Cabo

Classe de implementação do militar tipo Cabo, aonde a diferença das outras classes é o número de pontos que é acrescentado em cada ataque quando o militar é um Cabo.

2.2.3.3 Class – Sargento

Classe de implementação do militar tipo Sargento, aonde a diferença das outras classes é o número de pontos que é acrescentado em cada ataque quando o militar é um Sargento.

2.2.3.4 Class – Soldado

Classe de implementação do militar tipo Soldado, aonde a diferença das outras classes é o número de pontos que é acrescentado em cada ataque quando o militar é um Soldado.

2.2.3.5 Class – Tenente

Classe de implementação do militar tipo Tenente, aonde a diferença das outras classes é o número de pontos que é acrescentado em cada ataque quando o militar é um Tenente.

2.2.4 Package – model.mundo

Este pacote, armazena as classes das partes geográficas deste jogo, como o tabuleiro, que é chamada de mundo, a classe Continente e também os territórios. Segue abaixo um esboço de cada classe deste pacote:

2.2.4.1 Class – Mundo

Classe de implementação do tabuleiro do jogo e ainda cria a lista de Continentes e também dos territórios existentes no jogo.

2.2.4.2 Class – Continente

Já esta classe foi implementada com o objetivo de receber informações e “setar” informações dos 6 continentes existentes no jogo, como o nome, a quantidade de territórios que fazem parte dele e os militares de apoio.

2.2.4.3 Class – Território

Por fim, esta classe foi implementada com o objetivo de receber informações e “setar” informações dos 24 territórios existentes no jogo, como o nome, qual continente pertence, o jogador proprietário inicial do território, também adiciona militares no início do jogo e remove os militares de acordo com andamento/fluxo do jogo e por fim, recebe a lista de vizinhos que é inicializada na classe que tem o objetivo de inicializar o jogo, já descrita o funcionamento acima.

2.2.5 Package – view

Por fim, vamos comentar sobre o nosso último pacote da regra de negócios, que é chamado de package view. Este pacote, tem o objetivo de agrupar as classes que fazem a interface gráfica do jogo acontecer. Então todas as classes que estão agrupadas neste pacote, há apenas implementações relativas a interface gráficas, aonde a interface gráfica foi toda desenvolvida baseado nos objetos e respectivos métodos que dispõe o Java.Swing.

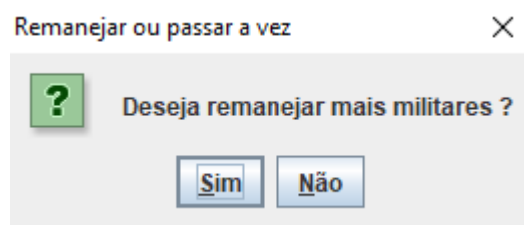
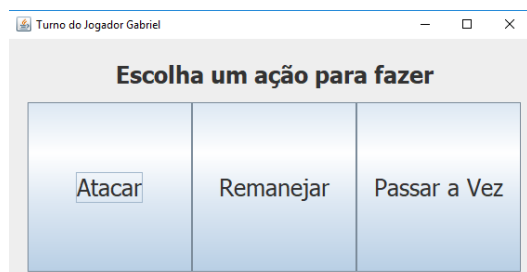
A intenção da interface gráfica (de todas as classes componentes) é capturar o nome de um dado da classe e outro detalhe que é igualitário para todas as classes, é que as mesmas há um FLAG e com valor *false* enquanto a tela não seja fechada, porém é alterado para *true* quando ocorre algum evento específico.

Abaixo, apenas iremos listar as classes pertencentes a este pacote e mostrar uma imagem de como ficou a interface de cada classe, porém o funcionamento delas não serão descritos, pelo fato de serem apenas implementações de interface gráfica e não há métodos

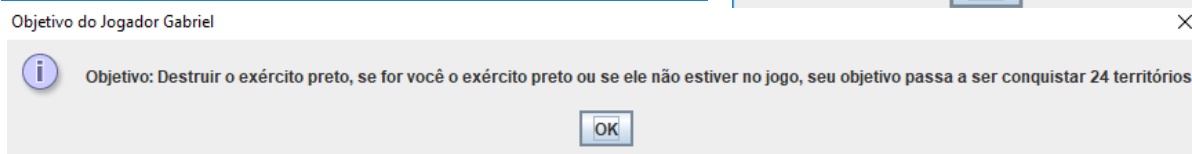
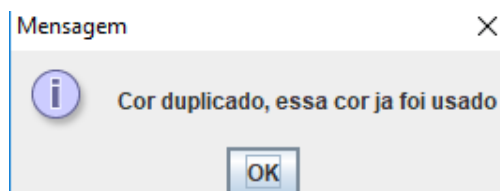
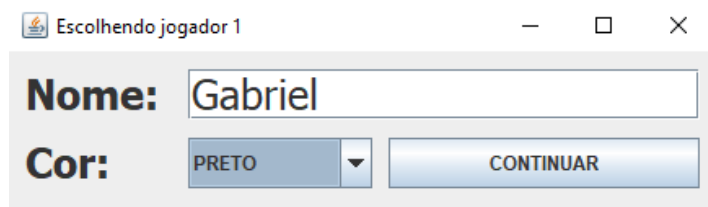
ou objetos que interfiram no funcionamento do jogo, e vamos além disso, o nome de cada classe terá a ideia clara de que cada uma faz e está ligada a sua interface.

Este pacote é apenas um bônus para aprimorarmos a parte visual do jogo, vejamos a listagem das classes pertencentes a este pacote:

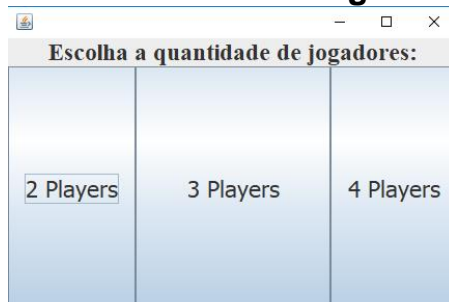
2.2.5.1 Class – AcoesDoJogador



2.2.5.2 Class – EscolherJogador



2.2.5.3 Class – EscolherQuantidadeJogador



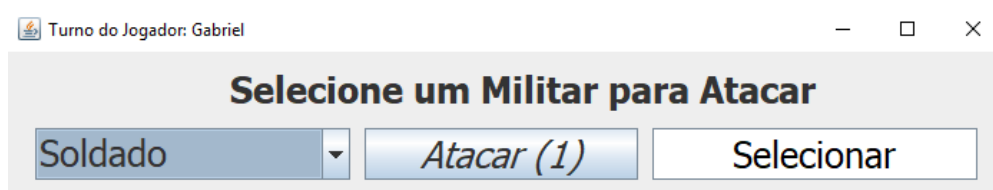
2.2.5.4 Class – PlayerInfo



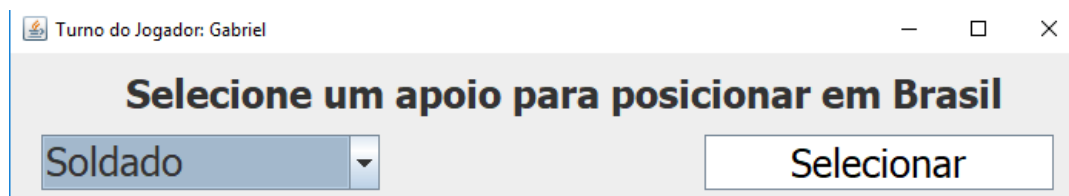
2.2.5.5 Class – ResultadoAtaque



2.2.5.6 Class – SelecionarMilitar



2.2.5.7 Class – SelecionarTerritorio



3. Testes

Os testes de funcionamento deste jogo foram realizados juntamente com a implementação das classes e seus respectivos métodos, então o mesmo foi sendo realizado gradativamente a medida em que o código iria sendo desenvolvido e avançando, caso o teste não foi satisfatório em alguma etapa, o desenvolvimento era pausado até que aquela etapa tornasse satisfatório no resultados dos respectivos testes.

Foi o único método de teste utilizado durante o desenvolvimento deste jogo, não foi utilizado nenhum outro recurso, como por exemplo, o JUnit do Java.

4. Conclusão

Chegamos à conclusão deste projeto, aonde a primeira conclusão a ser listada é que o objetivo de desenvolvimento de todos os pontos levantados foram cumpridos e executados com sucesso. No final de toda implementação, análise, conversas, discussões, contratempos, melhorias e etc, o jogo está funcionando diante dos parâmetros e requisitos que foram listados para o pleno e correto funcionamento do jogo, agora é possível jogar o War via a nossa interface desenvolvida através da linguagem Java.

Um ponto importante a ser ressaltado é que na etapa de definição dos vizinhos dos territórios, foram encontradas durante o desenvolvimento vários questionamentos e quais seriam os melhores modos de desenvolvimento, para que o jogo pudesse fluir da melhor maneira em relação a determinação dos vizinhos de cada território. Foram longos períodos de conversas entre a dupla executante do projeto e até mesmo houve um tempo significativo que também

teve a inserção do professor Fischer, a fim de encontrarmos a melhor solução para este assunto. No final destas conversas, encontramos uma forma aonde a implementação seria mais fácil, pois os vizinhos de cada territórios foram apontados na classe de inicialização do jogo, já comentada e explicada acima.

Também oute detalhe a ser mencionado é que no início do projeto houve um erro de lógica e a mesma foi implementada erroneamente e isto custou caro depois, pois estávamos implementando os territórios como “pais” dos continentes, então isto chegou a um momento insustentável no projeto, após a ajuda do professor Fischer, foi possível identificarmos isto e então realizar todo o retrabalho, aonde foi necessário literalmente excluirmos esta relação (excluir as linhas de códigos que faziam esta relação) e então refazer novamente porém da forma correta, aonde os continentes ficaram “pais” dos territórios.

No mais, também houve um grande avanço na parte de conhecimento na linguagem e desenvolvimento de OO da dupla, pois através da prática e desenvolvimento deste projeto, houve ganhos positivos em relação ao conhecimento e aprimoramento de nossas técnicas.

5. Bibliografia

Jônatas, Fischer. “TP1_ War o jogo de estratégia” Universidade de Itaúna (2018) - PDF.

“WindowBuilder - is a powerful and easy to use bi-directional Java GUI designer”. Disponível em: <https://www.eclipse.org/windowbuilder/> Acesso em: 20 abr. 2018.

“Java – Uma visão geral do pacote Swing”. Disponível em: <https://imasters.com.br/artigo/410/java/java-uma-visao-geral-do-pacote-swing?trace=1519021197&source=single> Acesso em: 19 abr. 2018.