

Universidade Estadual de Maringá
Centro de Tecnologia - Departamento de Informática
Graduação em Bacharelado em Ciência da Computação e
Informática

Trabalho: War

Ricardo Henrique Brunetto
RA 94182
Gustavo Luiz Furuata Ferreira
RA: 95675

Prof. Valéria Delisandra Feltrim
Prof. Juliana Keiko Yamaguchi

Maringá, 6 de fevereiro de 2017

Compilação

O projeto foi criado utilizando a IDE NetBeans, logo deverá ser executado na mesma para que a compilação e o funcionamento ocorram sem problema. O projeto apresenta uma documentação JavaDoc para um melhor entendimento de seu funcionamento.

Conceitos de Orientação a Objetos

Encapsulamento: Para a implementação desse conceito, todos os atributos das classes foram definidos com **private**. Desse modo foi garantido o controle de acesso aos dados das classes, maior flexibilidade e facilidade de modificação. Para a alteração e obtenção de dados, é necessário a utilização de métodos get e set.

Exemplo: a classe mapa possui os seguintes atributos:

```
private String nome;  
private Cor cor;  
private Continente continente;  
private int posX, posY;  
private ArrayList<Aereo> exercitosAereos;  
private ArrayList<Terrestre> exercitosTerrestres;
```

Herança: A implementação desse conceito incluiu a utilização de uma classe que estende de outra. Ou seja, uma classe filha que herda os atributos e métodos da classe mãe:

Exemplo: Classe Action e Ataque Terrestre.

Action é a classe mãe onde estão os atributos e métodos gerais. É uma classe abstrata.

AtaqueTerrestre é a classe filha que herda as características que Action, assim, tudo que Action possui, AtaqueTerrestre também possui, com a ressalva de que a classe filha poderá sobrescrever os métodos da classe mãe. O que nesse caso será obrigatório, pois o método executar de Action é do tipo abstrato.

Polimorfismo: Diz respeito a um método com uma determinada função e que pode assumir diferentes formas para cada instância.

Exemplo: As classes AtaqueAereo e AtaqueTerrestre, herdam e sobrescrevem o método combater. Desse modo um mesmo método funcionará de diferentes formas de acordo com sua instância. O método combater tem um funcionamento para a classe AtaqueAereo e outro funcionamento para AtaqueTerrestre.

Pacotes

Para uma maior organização e entendimento do funcionamento do jogo, o projeto foi dividido em cinco pacotes, onde cada pacote contém classes com comportamentos ou funções semelhantes. Os cinco pacotes são Ações, Controladores, ControleJogo, Dados e GUI.

O pacote **Ações** contém as classes que manipulam ações de ataque e remanejo de exércitos. As classes contidas neste pacote são Action, AtaqueAereo, AtaqueTerrestre e Remanejo.

O pacote **Controladores** possui as classes que formam o esqueleto do jogo, as quais são Mapa e Principal.

O pacote **ControleJogo** possui as classes que realizam o controle de rodadas e turnos. As classes desse pacote são Rodada, Turno e TurnoMapa.

O pacote **Dados** possui classes que representam todas as informações necessárias para o funcionamento do jogo. As classes contidas no pacote são Aereo, Continente, Cor, Exercito, Jogador, Terrestre e Territorio.

O pacote **GUI** possui os elementos que representam toda a interface gráfica do jogo.

Classes

Principal: Na classe Principal é onde ocorre todo o início do jogo. É onde se encontra o método main. Ao ser executada, a classe cria o mapa de territórios, cria os jogadores, realiza as chamadas de métodos para sorteio de territórios e cores, distribui os exércitos iniciais para cada jogador, invoca a interface gráfica e dá início à partida.

Mapa: É a classe que contém a matriz de territórios. Cada posição da matriz é um espaço para um território pré-definido. Os territórios são sempre alocados nas mesmas posições da matriz, pois o mapa nunca muda. Para a alocação de territórios na matriz, é criado um vetor do tipo String, onde cada posição do vetor representa um nome de território. As posições onde não há nenhum território são representadas por uma *static final string* VAZIO. A execução entra então em um laço de repetição que verificará todas as posições da matriz e atribuirá o nome e o continente ao qual o território pertence, de acordo com as posições verificadas. Os métodos de Mapa são:

getFronteirasValidas: Faz a verificação de fronteiras válidas para a realização de ataques e remanejo. Desse modo, após o jogador selecionar o tipo de ação desejada, será exibido na interface gráfica, quais territórios fronteiriços permitem a realização da ação selecionada.

getOrigensValidas: Verifica de quais territórios é permitido realizar um certo tipo de ataque. Desse modo, quando o jogador selecionar algum tipo de ataque, a interface gráfica exibirá de quais territórios é possível realizar essa ação. O método não permite que o jogador selecione um território inválido para a realização da ação ou que não seja do jogador.

distribuirTerritorios: Esse método faz a distribuição inicial de territórios para o jogador e para o computador. A distribuição de territórios é feita de forma aleatória.

getAuxilioValido: Esse método faz a verificação dos continentes adjacentes ao território selecionado, e mostra para o jogador quais são os territórios de onde é possível solicitar um auxílio aéreo. Para que um território de continente adjacente seja disponibilizado para o auxílio, é necessário que o mesmo possua pelo menos um exército aéreo.

getTerritorio: Exibe o território de acordo com as coordenadas fornecidas.

mostrarDados: Exibe as informações de cada território da matriz.

Territorio: É a classe que cria e manipula territórios. Cada território criado recebe um nome, cor, continente, posições na matriz e dois ArrayList que representam, respectivamente, os exércitos aéreos e terrestre alocados no território. Os principais métodos da classe são:

mostrarDados: mostra os dados do território selecionado.

getTotalExercitosNoTerritorio: retorna a soma do total de exércitos aéreos e terrestre alocados no território.

getoTotalExercitosAereos: retorna o valor do total de exércitos aéreos alocados no território.

getTotalExercitosTerrestres: retorna o valor total de exércitos terrestres alocados no território.

setNome: É atribuído um nome ao território. A escolha do nome é feita na classe Mapa.

getNome: Retorna a string que representa o nome do território.

setContinente: É atribuído um continente ao território. A escolha do continente é feita na classe mapa.

getContinente: Retorna o enum que representa o continente.

setCor: É atribuída uma cor ao território. A cor do território será a mesma cor do jogador que o possui.

getCor: Retorna o enum que representa a cor.

getPosX: Retorna o número da linha da matriz onde se encontra o território.

setPosX: É atribuído um valor para representar a linha da matriz onde o território se encontra. A posição da linha é passada através da classe Mapa.

getPosY: Retorna o número da coluna da matriz onde se encontra o território.

setPosY: É atribuído um valor para representar a coluna da matriz onde o território se encontra. A posição da coluna é passada através da classe Mapa.

getExercitosAereos: Retorna o ArrayList de exércitos aéreos alocados no território.

getExercitosTerrestres: Retorna o ArrayList de exércitos terrestres alocados no território.

adicionarExercito: Adiciona um exército ao território. O exército adicionado pode ser aéreo ou terrestre, pois a assinatura do método contém um exército do tipo Exercito, logo qualquer classe que herde de Exercito, poderá ser passada como parâmetro.

removeExercito: Remove um exército do território. O exército removido pode ser aéreo ou terrestre, pois a assinatura do método contém um exército do tipo Exercito, logo qualquer classe que herde de Exercito, poderá ser passada como parâmetro.

Jogador: É a classe onde são definidos os jogadores. Nesta classe os jogadores são diferenciados entre humano e máquina. Os atributos que um jogador deve ter são um ArrayList com os territórios possuídos pelo jogador, ArrayList com os exércitos terrestres disponíveis, ArrayList com os exércitos aéreos disponíveis, cor do jogador, tipo do jogador (onde é definido se este humano ou máquina) e um Map que faz a relação entre continente e os territórios do jogador, permitindo que seja possível saber o número de territórios possuídos em cada continente. Os métodos dessa classe são:

adicionarTerritorio: Realiza a adição de um território ao ArrayList *territoriosPossuidos*. Ao ser adicionado um território, a cor do mesmo é alterada para

a cor do jogador, e também é feita a relação entre o território adicionado e o continente do mesmo, através do Map.

removerTerritorio: Realiza a remoção de um território do ArrayList *territoriosPossuidos*. Ao ser removido um território, o mesmo também é desassociado do Map que faz a relação entre continente e território.

mostrarDados: Exibe as informações de territórios e exércitos pertencentes ao jogador.

getTotalExercitosNosTerritorios: É verificado para cada território possuído pelo jogador, o total de exércitos aéreos e terrestre. O método então retorna um vetor de 2 posições, sendo a posição 0 o total de exércitos terrestres e a posição 1 o total de exércitos aéreos. Para o retorno do total de exércitos é feita a soma de todos os exércitos aéreos e terrestres de cada território.

getTerritoriosPossuidos: Retorna o ArrayList de territórios possuídos pelo jogador.

getTotalExercitosDisponiveis: Retorna um inteiro que possui o total de exércitos disponíveis pelo jogador. O total de exércitos disponíveis é a soma de exércitos terrestres disponíveis e aéreos disponíveis.

getTotalAereoDisponiveis: Retorna um inteiro contendo o total de exércitos aéreos disponíveis.

getTotalTerrestreDisponiveis: Retorna um inteiro contendo o total de exércitos aéreos disponíveis.

getTotalTerritoriosPossuidos: Retorna um inteiro contendo o total de territórios possuídos pelo jogador.

getCor: Retorna o Enum que representa a cor do jogador.

setCor: Define a cor do jogador. A cor é aleatória e o método que realiza a chamada de escolha de cor está na classe Principal.

getTipo: Retorna o tipo do jogador, que pode ser HUMANO ou COMPUTADOR.

setTipo: Define o tipo do jogador: HUMANO ou COMPUTADOR.

getQtdTerritoriosCadaContinente: Retorna o mapa que faz a relação entre Território e Continente.

adicionarTerrestreAoTerritorio: Adiciona um exército terrestre disponível ao território selecionado.

adicionarAereoAoTerritorio: Adiciona um exército aéreo disponível ao território selecionado.

adicionarDisponiveisAereos: Adiciona um exército aéreo ao ArrayList *aereosDisponiveis*, aumentando assim o total de aéreos disponíveis.

adicionarDisponiveisTerrestres: Adiciona um exército terrestre ao ArrayList *terrestresDisponiveis*, aumentando assim o total de terrestres disponíveis.

Exercito: É a classe que define os exércitos de um modo geral. É uma classe abstrata logo não pode ser instanciada. A classe possui um único método abstrato, que deverá ser sobrescrito pelas classes que a herdarem. Esse método é um exemplo de polimorfismo, pois assumirá diferentes formas de acordo com a classe que implementá-lo.

Terrestre: É a classe que representa os exércitos terrestres. A classe herda de Exercito, sendo este um exemplo de herança, logo precisará sobrescrever o método atacar, pois este é um método abstrato.

Aéreo: É a classe que representa os exércitos aéreos. A classe herda de Exercito, sendo este um exemplo de herança, logo precisará sobrescrever o método atacar, pois este é um método abstrato.

Continente: Representa os continentes aos quais os territórios podem pertencer. É do tipo Enum. A cada Enum de continente há associado uma String que representa o nome do continente e um inteiro que representa o total de territórios que cada continente pode possuir. Os métodos desse Enum são:

getNome: Retorna a String representando o nome associado ao continente.

getQuantidadeTerritorios: Retorna o inteiro que representa o total de territórios permitidos em cada continente.

getContinentesAdjacentes: Esse método recebe um Enum do tipo Continente e verifica quais são os continentes adjacentes a ele. O método retorna um ArrayList de Continentes contendo todos os continentes adjacentes.

Cor: Representa as cores que podem ser associadas ao jogador. É do tipo Enum sendo que a cada Enum de Cor há associado três inteiros: *red*, *green* e *blue*. Desse modo o tom das cores é obtido através de RGB. Os tons que podem ser gerados são Vermelho, Azul, Amarelo, Verde, Rosa, Roxo e Neutro. A cor neutra representa territórios vazios, que na interface gráfica é representado como mar. Os métodos desse Enum são:

getRed: retorna o inteiro que representa o código RGB do tom vermelho utilizado na cor selecionada.

getBlue: retorna o inteiro que representa o código RGB do tom azul utilizado na cor selecionada.

getGreen: retorna o inteiro que representa o código RGB do tom verde utilizado na cor selecionada.

getRandomColor: Esse método recebe como parâmetro um ArrayList de jogadores e para cada jogador, é atribuída uma cor aleatória entre as cores disponíveis. A cor Neutra nunca é escolhida, pois será a cor que representa os territórios vazios. O método também garante que não haverá jogadores com a mesma cor.

getColor: Retorna um objeto do tipo Cor gerado pelos inteiros que representam o RGB da cor.

getRGB: Retorna uma String do RGB da cor.

Action: É uma classe abstrata que representa os tipos de ação que o jogador pode realizar. Por ser abstrata, a classe não poderá ser instanciada. Os atributos que um Action deve ter são: Território *origem* e *destino*, que representam respectivamente, o território que está lançando a ação e o território que receberá a ação; Map *exercitosAtivos*, que fará a associação entre o território e os exércitos disponíveis em cada um (um Map é necessário, pois nos Ataques Aéreos, um território pode solicitar apoio de um território que esteja em um continente adjacente, logo, para fazer a distinção entre os exércitos de cada território, um Map se faz necessário). Os métodos que a classe possui são:

getOrigem: retorna um objeto do tipo Territorio, sendo este o território origem, ou seja, o território que executará a ação.

setOrigem: recebe como parâmetro um território origem. Esse será o território que executará a ação.

addExercitoAtivo: Adiciona um exército que será utilizado para a realização da ação. O método recebe como parâmetro o território em que o exército está alocado e o próprio exército. É feita então a associação entre território e exército no Map exercitosAtivos. Como o método é utilizado tanto para ataques aéreos quanto terrestres, o exército presente na assinatura do método será da superclasse Exército, desse modo será possível receber exércitos de qualquer classe que herde de Exército.

removeExercitoAtivo: Realiza a remoção dos exércitos ativos que seriam utilizados para a realização da ação. Recebe como parâmetro o território onde se encontra o exército. Ao ser removido um exército, o mesmo é desassociado do território contido no Map.

getExercitosAtivos: retorna um List que contém todos os exércitos ativos que serão utilizados para a realização da ação.

getDestino: retorna um objeto do tipo Territorio, sendo este o território destino, ou seja, o território que receberá a ação.

setDestino: recebe como parâmetro um território destino. Este será o território que receberá a ação.

executar: Método abstrato que representa a execução da ação. Por ser abstrato, toda classe que herdar de Action, deverá sobrescrever o método. Esse é um caso de polimorfismo, onde um método poderá assumir diferentes formas de acordo com a classe que o sobrescrever.

AtaqueTerrestre: É a classe que representa a ação de ataque terrestre. A classe herda de Action, sendo assim, também herda todos os métodos e atributos da mesma. A classe possui o atributo int quantidadeDefesa, que representa o total de exércitos que o destino usará para se defender. Os métodos da classe são:

executar: Método herdado de Action e por ser abstrato, deverá ser sobrescrito. No ataque terrestre, o método sorteará um valor entre 1 e 6 para cada exército utilizado no ataque através do método *combater()*. O mesmo ocorrerá para os exércitos de defesa. Após o sorteio dos valores, os mesmos são ordenados em ordem decrescente e comparados. Caso o valor do ataque seja maior que o de defesa, a defesa perde um exército, caso seja menor ou igual, o ataque perde um exército, o processo se repete para os outros valores até que todos os valores sejam verificados ou até que o ataque ou defesa fiquem sem valores para comparação.

setQuantidadeDefesa: Recebe como parâmetro um inteiro que representa a quantidade de exército que a defesa usará.

AtaqueAereo: É a classe que representa a ação de ataque aéreo. A classe herda de Action, sendo assim, também herda todos os métodos e atributos da mesma. A classe possui o atributo int quantidadeDefesa, que representa o total de exércitos que o destino usará para se defender. Os métodos da classe são:

executar: Método herdado de Action e por ser abstrato, deverá ser sobrescrito.

setQuantidadeDefesa: Recebe como parâmetro um inteiro que representa a quantidade de exército que a defesa usará.

Remanejo: É a classe que representa a ação de remanejo de exércitos. A classe herda de Action, sendo assim, também herda todos os métodos e atributos da mesma. A classe possui o atributo ArrayList *exercitosParaTransferir*, que representa os exércitos que serão transferidos para um território do jogador. A classe possui os seguintes métodos:

addExercito: Faz a adição de exércitos no ArrayList *exercitosParaTransferir*. Recebe como parâmetro um exército da classe Exercito, pois o remanejo pode acontecer para os tipos terrestre e aéreo, logo, qualquer classe que herde de Exercito poderá ser passada como parâmetro.

executar: Método herdado de Action e por ser abstrato, deverá ser sobrescrito. Adiciona ao território destino os exércitos contidos no ArrayList *exercitosParaTransferir*.

Rodada: A classe representa as rodas que acontecerão no jogo, cada rodada é iniciada após todos os jogadores terem concluído seus turnos. A classe possui os atributos ArrayList de jogadores, que representa os jogadores que estão participando do jogo; Mapa que representa a matriz de mapas do jogo; interface gráfica, para que as informações da rodada sejam passadas para o jogador. O método presente na classe é:

iniciar: Para cada jogador inserido no ArrayList de jogadores, é criado um objeto do tipo Turno e é iniciado o turno do jogador. O início do turno ocorre quando é feita a chamada do método iniciar, presente na classe Turno. A rodada é finalizada após todos os jogadores terem encerrado seus turnos.

Turno: Classe que representa a vez de jogada do jogador. Possui os atributos JogadorAtual, que representa o jogador que está executando a jogada e a interface gráfica, para transmitir as informações do turno ao jogador. Possui o método:

iniciar: Inicia a jogada do jogador. Para que o jogador possa realizar sua ação, o método verifica se o jogador é humano. Caso seja, a interface gráfica inicia a rodada e permite que o jogador realize sua jogada.

TurnoMaquina: Representa a vez de jogada do computador. Todas as jogadas de ataque e defesa são realizadas automaticamente.

PrincipalGUI: Classe que representa a interface gráfica do jogo. Para a interface gráfica foi utilizada a API Java Swing. Os métodos utilizados na classe PrincipalGUI são:

mapaTerritorioMouseClicked: Método que verifica os cliques do mouse em um célula da tabela (território).

btnAtaqueTerrestreMouseClicked: Método de evento de clique. Representa a ação de Ataque Terrestre. O evento é acionado quando o botão de Ataque Terrestre é clicado.

btnAtaqueAereoMouseClicked: Método de evento de clique. Representa a ação de Ataque aéreo. O evento é acionado quando o botão de Ataque Aéreo é clicado.

btnRemanejoMouseClicked: Método de evento de clique. Representa a ação de Remanejo de exército. O evento é acionado quando o botão de Remanejo é clicado.

btnAddAereoActionPerformed: Método de evento de clique. Representa a ação de adicionar um exército aéreo ao território. O evento é acionado quando o botão Aéreo é clicado. A ação fica disponível no início de cada turno, quando o jogador deve alocar seus exércitos disponíveis.

btnAddTerrestreMouseClicked: Método de evento de clique. Representa a ação de adicionar um exército terrestre ao território. O evento é acionado quando o botão Terrestre é clicado. A ação fica disponível no início de cada turno, quando o jogador deve alocar seus exércitos disponíveis.

btnFinishedActionMouseClicked: Método de evento de clique. Representa a quantidade de exércitos que o jogador utilizará no ataque ou remanejo. O jogador deverá clicar na caixa de seleção de exércitos, para aumentar ou diminuir a quantidade que deseja. O número de exércitos que podem ser selecionados é de no máximo 3, além disso, o jogador só poderá utilizar exércitos de acordo com a quantidade disponível para a realização da ação. Caso o jogador tenha selecionado ataque Aéreo, haverá ainda o botão Solicitar apoio aéreo, que servirá para obter apoio aéreo de territórios adjacentes.

btnApoioAereoMouseClicked: Método de evento de clique. Representa a ação de solicitar apoio aéreo. O evento é acionado quando o botão Solicitar apoio aéreo é clicado. Ao clicar nesse botão, a interface gráfica destaca todos os territórios de onde é possível solicitar apoio aéreo.

btnCancelMouseClicked: Método de evento de clique. Representa a ação de cancelar. O evento é acionado quando o botão Cancelar é clicado.

start: Método que inicializa a tabela da interface gráfica. Ao ser executado, o método preenche a tabela com os territórios da matriz.

atualizarDados: Método que atualiza os dados que serão exibidos na interface gráfica, como a quantidade de exércitos espalhados pelo mapa e o painel da relação dos continentes.

atualizarMapa: Método que faz a atualização do mapa da interface gráfica.

showDetalhes: Método que exibe as informações de cor, total de exércitos geral, total de terrestres, total de aéreos e nome do território. O método é acionado quando um território é clicado.

ativarTurno: Método que ativa o turno. O método recebe como parâmetro um boolean que servirá para ativar e desativar o turno.

selecionarOrigem: Método que seleciona o território que realizará a ação. A seleção de origem serve para Ataque Terrestre, Ataque Aéreo e Remanejo de exército. A ação é iniciada após selecionar o tipo de ação desejada. A interface destaca então os territórios de origem disponíveis. Para realizar o evento de seleção de origem, o jogador deve dar um duplo-clique em um dos territórios destacados.

instanciarAcaoGUI: Método que instancia o tipo de ação selecionada pelo jogador.

selecionarDestino: Método que seleciona o território que receberá a ação. A ação é iniciada após ser realizada a seleção do território origem. A interface destaca então os territórios de destino disponíveis. Para realizar o evento de seleção de destino, o jogador deve dar um duplo-clique em um dos territórios destacados.

vezDaMaquina: Método que realiza o turno de defesa da máquina, selecionando automaticamente a quantidade de exércitos para a defesa.

iniciarRodada: Método que inicia uma rodada. Ao ser iniciada uma rodada, o jogador deverá posicionar seus exércitos disponíveis em seus territórios.

verificarFimDistribuicaoExercitos: Realiza a verificação de distribuição de exércitos do jogador. Ao fim da distribuição, as opções de ação ficam disponíveis.

Classes Internas de PrincipalGUI:

BoardTableCellRenderer: Realiza a renderização padrão do mapa. A classe herda de DefaultTableCellRenderer.

SelecionarTerritorioCellRenderer: Realiza a renderização de seleção de territórios no mapa. A classe herda de DefaultTableCellRenderer.

Manual:

Início:

Ao ser inicializado, o jogo gera a matriz de territórios, distribui os territórios entre os jogadores, define o número de exércitos iniciais disponíveis para alocação, sorteia a cor de cada jogador e aloca um exército a cada território dos jogadores. É instanciada então a interface gráfica, onde são inseridas todas as informações do jogo. Após renderizar toda a parte visual, a janela do jogo é aberta. O jogador então verá a interface com todas as informações já preenchidas. Os territórios do jogador estarão destacados em relação aos territórios do computador. Para começar o jogo, o jogador deverá clicar em uma célula de seu território. Na parte inferior da janela de jogo, surgirão os botões Terrestre e Aéreo. O jogador deverá então escolher uma opção e alocar seus exércitos disponíveis em seus territórios. A alocação de exércitos é explicada a seguir.

Alocando de exércitos:

Para realizar a alocação de exércitos em seus territórios, o jogador deverá clicar em uma célula de seus territórios, os quais estarão com a cor e escrita destacados em relação aos outros territórios. Após realizar o clique, surgirão na parte inferior os botões Terrestre e Aéreo. O jogador deverá clicar na opção de exército que deseja alocar.

Alocando Terrestres: Para alocar exércitos terrestres, o jogador deverá clicar no botão Terrestre. Ao clicar no botão, um exército do tipo Terrestre é alocado no território selecionado. Em baixo do botão Terrestre, há o total de exércitos desse tipo que ainda estão disponíveis para alocação. Cada vez que um exército é alocado, o total diminui. Quando o total chegar a 0, o jogador não poderá mais alocar exércitos do tipo Terrestre. Caso o jogador deseje alocar exército Terrestre em outro de seus territórios, deverá clicar no território desejado e realizar o mesmo procedimento.

Alocando Aéreos: Para alocar exércitos aéreos, o jogador deverá clicar no botão Aéreo. Ao clicar no botão, um exército do tipo Aéreo é alocado no território selecionado. Em baixo do botão Aéreo, há o total de exércitos desse tipo que ainda estão disponíveis para alocação. Cada vez que um exército é alocado, o total diminui. Quando o total chegar a 0, o jogador não poderá mais alocar exércitos do tipo Aéreo. Caso o jogador deseje alocar exército Aéreo em outro de seus territórios, deverá clicar no território desejado e realizar o mesmo procedimento.

Após realizar a alocação de todos os seus exércitos, o mapa da interface atualizará e colocará em destaque todos os territórios do mapa, indicando assim que o jogador deverá realizar uma ação.

Realizando ações:

Após o término de alocação de exército, o jogador deverá realizar uma ação, que pode ser Ataque Terrestre, Ataque Aéreo ou Remanejo de exército. Os botões para realização dessas ações estarão disponíveis logo abaixo do mapa. O jogador deverá então clicar na ação desejada e seguir os procedimentos para realizá-la.

Ataque Terrestre: Para realizar um ataque terrestre, o jogador deverá clicar no botão Ataque Terrestre, a qual instancia uma nova ação Action do tipo AtaqueTerrestre. Após realizar o clique, o mapa atualizará e deixará em destaque apenas os territórios de origens válidas para um ataque terrestre, sendo que as origens válidas são verificadas pelo método 'getOrigensValidas' da classe Mapa. O jogador deverá então dar um duplo clique no território de origem do ataque. Ao realizar o duplo clique, uma nova *label* surgirá, contendo o nome do território selecionado e um campo para indicar a quantidade de exércitos que deseja utilizar no ataque. O jogador não pode utilizar mais de 3 exércitos para o ataque. Não se pode deslocar todo o contingente do exército para o ataque: deve permanecer pelo menos um exército no território para sua ocupação. Portanto, para efetuar um ataque terrestre é necessário que haja pelo menos dois exércitos. Após selecionar o total de exércitos para o ataque, o jogador deverá clicar no botão Ok!. Após realizar o clique, o mapa atualizará e destacará os territórios no qual é possível realizar um ataque, o qual é feito pelo método 'getFronteirasValidas' da classe Mapa. O jogador deverá dar um duplo clique em um dos territórios destacados. Após realizar o duplo clique, a *label* exibirá o nome do território escolhido e os exércitos que o computador utilizará para se defender. Para iniciar o ataque, o jogador deverá clicar no botão Iniciar Ataque!. Após clicar no botão, será invocado método 'combate' da classe AtaqueTerrestre a qual simulará um dado e sorteará um número aleatório entre 1 e 6 para cada exército utilizado no ataque. Após isso, os valores obtidos pelo jogador e pelo computador são comparados. A comparação acontece

da seguinte forma: Verifica-se o maior valor do ataque e o maior valor da defesa. Caso o valor do ataque seja maior que o da defesa, o computador perde um exército; caso o valor da defesa seja igual ou maior que o de ataque, o jogador perde um exército. O processo se repete para o resto dos valores. O ataque tem fim após todos os valores serem comparados ou até que um valor fique sobrando. Exemplo:

Jogador utiliza 3 exércitos para o ataque.

Valores sorteados para o jogador: 2, 6, 3

Computador utiliza 2 exércitos para a defesa

Valores sorteados para o computador: 5, 4

$6 > 5$ - Computador perde um exército.

$4 > 3$ - Jogador perde um exército.

O computador não possui valor para comparar com 2, logo o ataque é finalizado.

Ataque Aéreo: Para realizar um ataque aéreo, o jogador deverá clicar no botão Ataque Aéreo, a qual instancia uma nova ação Action do tipo AtaqueAereo. Após realizar o clique, o mapa atualizará e deixará em destaque apenas os territórios origens de onde é possível realizar um ataque aéreo, sendo que as origens válidas são verificadas pelo método 'getOrigensValidas' da classe Mapa. O jogador deverá então dar um duplo clique no território de origem do ataque. Ao realizar o duplo clique, uma nova *label* surgirá, contendo o nome do território selecionado e um campo para indicar a quantidade de exércitos que deseja utilizar no ataque. O jogador não pode utilizar mais de 3 exércitos para o ataque. O jogador deverá então selecionar a quantidade de exército que deseja utilizar no ataque. É possível ainda utilizar exércitos aéreos de territórios em continentes adjacentes. Para que isso ocorra, o jogador deverá clicar no botão Solicitar apoio aéreo, localizado ao lado do botão Ok!. Ao clicar no botão Solicitar apoio aéreo, o mapa destacará os territórios aos quais é possível solicitar apoio aéreo. O jogador deverá então dar um duplo clique no território e selecionar a quantidade de exércitos que deseja solicitar (compondo um ataque de no máximo três exércitos aéreos). Após indicar o total de exércitos que utilizará no ataque, o jogador deverá clicar no botão Ok!. Para efetuar um ataque aéreo, o território adversário deve ter no mínimo 1 exército aéreo e 4 terrestres. Assim o método 'getFronteirasValidas' da classe Mapa é invocado, destacando no mapa quais são os territórios destino onde é possível realizar um ataque aéreo. O jogador deverá dar um duplo clique em um dos territórios destacados. Após realizar o duplo clique, a *label* exibirá o nome do território escolhido e o exército que o computador utilizará para se defender. Para iniciar o ataque o jogador deverá clicar no botão Iniciar Ataque!. Após clicar no botão, o programa invocará o método 'combater' da classe AtaqueAereo e simulará um dado e sorteará um número aleatório entre 1 e 3 para cada exército utilizado no ataque. Após isso, os valores obtidos pelo jogador e pelo computador são analisados. A defesa iniciará o ataque verificando os números obtidos no sorteio. Cada valor obtido indica a quantidade de exércitos aéreos de ataque que serão abatidos (Caso seja 0, nenhum será abatido). A defesa poderá utilizar apenas 1 exército aéreo para se defender. Após a defesa terminar o ataque, o jogador realiza o ataque de acordo com os valores sorteados:

quando o retorno é igual 0, nada acontece (nenhum exército oponente é abatido);

quando o retorno é igual a 1, o território atacado perde 1 exército aéreo e 1 terrestre;

quando o retorno é igual a 2, o território atacado perde 1 exército aéreo e 2 terrestres;

quando o retorno é igual a 3, o território atacado perde 1 exército aéreo e 3 terrestres;

Exemplo:

Jogador ataca o México com 1 exército da Califórnia, 1 de Nova York e 1 da Colômbia.

México possui um total de 9 exércitos terrestres e 6 aéreos.

Valores sorteados para defesa: 1

Jogador utilizou 3 exércitos aéreos. Jogador perde 1 exército de um território aleatório que foi abatido pelo México. Suponha que o exército abatido seja de Nova York. Assim, o jogador ficou com 2 exércitos aéreos.

Jogador realiza ataque

Valores sorteados para ataque: 2, 0

México perde 1 exército aéreo e 2 terrestres, ficando com um total de 7 exércitos terrestres e 5 exércitos aéreos.

Jogador opta por continuar ataque.

México realiza defesa.

Valores sorteados para defesa: 2

Colômbia e Califórnia perdem 1 exército aéreo cada, deixando o jogador sem ataque e finalizando o turno.

Caso o jogador continue com exércitos aéreos para atacar, só será permitido prosseguir com o combate aéreo se houver 1 exército aéreo e 4 exércitos terrestres, no mínimo no território adversário.

Remanejamento de exército: Para realizar um remanejamento de exército, o jogador deverá clicar no botão Mover Exército, a qual instancia uma nova ação Action do tipo Remanejo. Após realizar o clique, o mapa atualizará e deixará em destaque apenas os territórios origens de onde é possível realizar um remanejamento, sendo que as origens válidas são verificadas pelo método 'getOrigensValidas' da classe Mapa. Ao realizar um duplo clique no território origem, uma nova *label* surgirá, contendo o nome do território selecionado e um campo para indicar a quantidade de exércitos terrestres e aéreos que deseja remanejar. O território de origem deverá permanecer com no mínimo 1 exército terrestre. Após selecionar a quantidade de exércitos para remanejamento, o jogador clica no botão Ok!. O programa destaca então os territórios onde é possível realizar um remanejamento através do método 'getFronteirasValidas' da classe Mapa. O jogador dá então um duplo clique no território desejado. A label exibe então o nome do território selecionado e o total de exércitos aéreos e terrestres já alocados no mesmo. Para concluir o remanejamento o jogador clica no botão Mover exércitos. Ao realizar um remanejamento, o jogador passa a vez.

Cancelar ação: Caso o jogador queira cancelar uma ação antes de concluí-la, deverá clicar no botão Cancelar, que estará disponível logo abaixo no Mapa, na posição esquerda

Conquistando Território:

Para conquistar um território, o jogador deverá realizar a ação de ataque terrestre. Após concluir o ataque terrestre, será verificado se o território atacado ficou com 0 exércitos terrestres. Em caso afirmativo, o território passa a pertencer ao jogador. O jogador deverá então mover ao território conquistado, pelo menos 1 exército terrestre que foi utilizado para o ataque. Caso o território conquistado possua exércitos aéreos, estes passam a pertencer ao jogador.

Visualizando dados:

Ao iniciar a partida, o lado esquerdo da janela do jogo exibirá as informações úteis ao jogador. As informações exibidas seguindo a ordem de cima para baixo são, respectivamente: nome do território, nome do continente ao qual o território pertence, cor do território selecionado, total de exércitos que o território possui, total de exércitos aéreos e total de exércitos terrestres alocados no território, cor do jogador, total de exércitos terrestres que o jogador possui espalhados pelo mapa, total de exércitos aéreos que o jogador possui espalhados pelo mapa e total de territórios possuídos por continente. Caso o jogador deseje ver informações de qualquer território, deverá dar um clique simples no território desejado.

As células do mapa também mostram as informações do território. As informações presentes nas células são nome do território, continente e total de exércitos terrestre e aéreos alocados no território, onde o valor do lado esquerdo representa a quantidade de exércitos terrestres e o da direita representa o total de exércitos aéreos. A cor de fundo da célula representa a cor do jogador que possui o território.

Ganhando a partida:

A partida será ganha pelo jogador que conseguir conquistar 2 continentes primeiro.