

Projeto Brasil War 2018

EXA 836 - Padrões e Frameworks – UEFS, turma 2018.2

Geral

Vocês devem projetar e implementar uma suite do clássico jogo War. War é um jogo de tabuleiro lançado no Brasil pela Grow em 1972 (<https://pt.wikipedia.org/wiki/War>). Sua implementação deve criar uma suite de jogos para as versões War e War 2, no Brasil.

O projeto foi pensado para ser realizado em dupla. Uma pessoa pode fazer sozinha, se assim decidir desde o início. Uma vez adotada a dupla, ela deve seguir até o final do projeto. Caso não haja colaboração igualitária, o professor deve ser avisado para dividir a nota dada de forma proporcional.

O projeto será avaliado em dois milestones. O peso das notas, com relação à nota final do curso, será o seguinte:

Milestone 1 (60%)

Milestone 2 (40%)

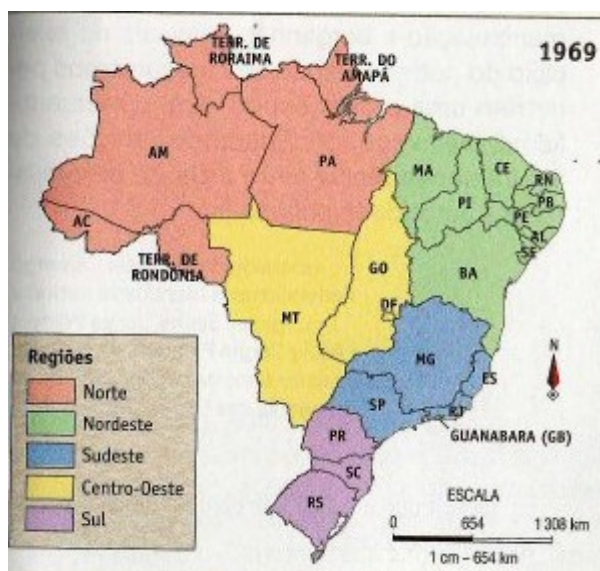
A especificação apresentada abaixo pode estar incompleta e não é formada apenas por este documento. O professor é o cliente e pode solicitar alterações na especificação apresentada. É sua responsabilidade, ao ter alguma dúvida, procurar o professor. Se omitir quanto a regras óbvias a serem implementadas será considerado passível de punição.

O Projeto

Parte 1

Para o milestone 1 você deve implementar a primeira versão do jogo War (War I). Neste site você encontra as regras do jogo: <http://www.bananaquantica.com.br/regras-do-jogo-war/>. Neste outro site, você pode jogar o War I para se familiarizar com o jogo, caso não o conheça: <http://www.growgames.com.br/Jogos/War> (cuidado para não gastar tempo demais jogando, em detrimento do desenvolvimento do projeto).

O mapa da primeira versão do Brasil War 2018 segue abaixo.



Existem diferenças entre a versão original do jogo e a versão Brasil War 2018:

- i) no Brasil War 2018, você usará o mapa do Brasil de 1969 (note que a divisão é anterior à década de 80, onde ainda não existiam os estados de Tocantins e Mato Grosso do Sul);
- ii) DF deve ser desconsiderado;
- iii) não há cartas de objetivos. O objetivo é conquistar o país inteiro e destruir todos os seus inimigos;
- iv) Os continentes da versão original do War serão substituídos por cada região do Brasil (norte, nordeste, sul, sudeste, centro-oeste);
- v) para a distribuição dos exércitos, cada região conquistada vale apenas 1 exército; as cartas-estados valem também 1 exército quando for realizada a troca;
- vi) O número de exércitos em cada troca de cartas-estados aumentará em 1: ou seja, a primeira troca vale 1 exército, a segunda troca vale 2 exércitos, e assim por diante.

Casos de uso

1 – Iniciar Jogo – indica número de jogadores; associar uma cor a cada jogador; distribuir cartas-estados.

2 – Distribuição – cada jogador distribui seus exércitos em seus estados.

3 – Combates – Cada jogador realiza suas ações de ataque e redistribuição dos exércitos.

4 – Carta-estado – Ao final de cada rodada, o jogador recebe uma carta-estado se conquistou ao menos 1 território.

5 – Ver mapa – Os jogadores visualizam a distribuição de exércitos no mapa.

6 – Parabenizar – O programa deve perceber que um jogador venceu o jogo e deve parabenizá-lo.

7 – Encerrar – O jogo pode ser encerrado a qualquer momento.

Vista (interface com o usuário)

Você deve implementar uma interface caracter, que pode ser a mais simples que você achar válida. Veja um exemplo no arquivo “exemploVistaUsuarioCaracter.txt”, onde todos distribuem seus exércitos na fase inicial do jogo e depois o jogador 1 realiza um ataque.

Parte 2

Você entregará seu programa a outra dupla e receberá o programa de uma outra dupla. Isso tem um objetivo: fazer você perceber como é importante escrever programas flexíveis e fáceis de entender. Haverá um sorteio para a redistribuição dos programas.

Você receberá o programa da outra dupla e implementará a nova versão baseada nas regras do War II. As regras do War II podem ser encontradas em <http://pxygames.blogspot.com/2008/12/regras-do-war2.html>.

O mapa da versão do Brasil War II 2018 segue abaixo.



Existem diferenças entre a versão original do jogo e a versão Brasil War II 2018:

- i) no Brasil War II 2018, você usará o mapa do Brasil de 1988 (o que muda da versão inicial do jogo é que no novo mapa há os estados de Tocantins e Mato Grosso do Sul);
- ii) DF deve ser desconsiderado;
- iii) não há cartas de objetivos. O objetivo é conquistar o país inteiro e destruir todos os seus inimigos;
- iv) Os continentes da versão original do War serão substituídos por cada região do Brasil (norte, nordeste, sul, sudeste, centro-oeste);
- v) para a distribuição dos exércitos, cada região conquistada vale apenas 1 exército; as cartas-estados valem também 1 exército quando for realizada a troca;
- vi) O número de exércitos em cada troca de cartas-estados aumentará em 1: ou seja, a primeira troca vale 1 exército, a segunda troca vale 2 exércitos, e assim por diante;
- vii) Agora existem aviões e centros estratégicos. Você deve implementar estes novos elementos;
- viii) Os centros estratégicos serão os estados de SC, SP, MT, AP e PI.

Casos de uso

1 – Escolher jogo – O usuário pode escolher se vai jogar o War I ou War II de sua suite.

2 – Todos os casos de uso da versão Brasil War (I) 2018.

Vista (interface com o usuário)

Você deve implementar uma interface não caracter. Não precisa ser altamente elaborada. Uma vista simples é suficiente.

Milestone 1

Você deve implementar todos os Casos de Uso da Parte 1 do Projeto. **A data/hora limite é a aula**

do dia 29/11/2018. Atenção para este prazo. Se você não entregar no dia, seu projeto não será recebido por nenhuma outra equipe. Isso implicará em uma perda de 40% da nota do Milestone 2.

Vocês devem entregar até a aula do dia 29/11/2018 um relatório em formato .pdf . Vocês também devem preparar uma apresentação para explicar seu projeto, que será discutido junto com a turma. Algumas informações importantes a serem apresentadas no relatório e na apresentação:

- Um explicação geral do seu projeto
- Projeto arquitetural
- Modelo conceitual em UML (Diagrama de Classes)
 - Uma descrição de cada entidade
 - Uma descrição das responsabilidades assumidas pela entidade (1 frase curta para cada responsabilidade)
- Modelo de baixo nível em UML (Diagrama de Classes) com as principais classes/métodos/atributos previstos para o projeto
- Diagramas de sequência, caso considere relevante
- Para cada modelo apresentado, uma explicação sobre o mesmo. Você deve conduzir minha compreensão sobre o modelo

A Entrega do Projeto no Milestone 1

Você deve entregar:

- Código-fonte documentado (testes de unidades inclusos)
- Javadoc do código-fonte
- Relatório impresso e em arquivo eletrônico contendo
 - Um percentual de participação de cada integrante da dupla no desenvolvimento do trabalho
 - A descrição do seu projeto
 - Modelo e explicação do design (Modelo Conceitual)
 - 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.
- O arquivo compactado deve trazer uma pasta para o relatório, uma para o programa e uma para o Javadoc. Além disso deve conter um arquivo readme.txt indicando se o projeto é Eclipse ou Netbeans (só aceito projeto Eclipse ou Netbeans).

- *Não topo* tentar entender porque alguma coisa não está funcionando. No dia da entrega, instalarei e você terá a oportunidade de resolver algum problema. Passado esse dia, começarei a descontar pontos, caso eu não consiga executar os testes ou o programa.

Milestone 2

Você vai receber o material produzido no milestone 1 do projeto de *outro grupo* e deverá implementar os casos de uso da Parte 2 do Projeto. Mude o que achar necessário no programa que você recebeu para implementar. **A data limite para entrega é a aula do dia 18/02/2018.**

A Entrega do Projeto no Milestone 2

Você deve entregar:

- Código-fonte documentado (testes de unidades inclusos)
- Javadoc do código-fonte
- Você deve realizar uma apresentação e avaliar pontos do programa modificado contendo:

Itens que devem constar (obrigatoriamente) nas apresentações:

- Um percentual de participação de cada integrante da dupla no desenvolvimento do trabalho
- As fraquezas do design que você recebeu.
- As virtudes do design que você recebeu.
- As dificuldades que você teve para implementar os casos de uso.
- O refatoramento que você efetuou.
- O refatoramento que você poderia efetuar para tornar o código melhor.
- Dê uma nota de 0 a 10 para cada uma das áreas abaixo, tanto para o resultado final, quanto para o projeto que você recebeu:
 - Qualidade da documentação
 - Qualidade do design
 - Qualidade do código
- Você deve colocar o programa para executar, demonstrando os pontos em que houve mudanças relevantes (e o grau de dificuldade dessas mudanças).

Além disso você deve seguir todas as “Regras para entrega (devem ser rigorosamente seguidas)” solicitadas no Milestone 1.

Avaliação

A dupla será penalizada em 1,0 ponto por dia de atraso. Não serão consideradas desculpas do tipo: “Minha internet caiu na hora de enviar”, “Faltou energia”, “Meu computador pifou”, “Perdi tudo e não fiz backup”, “Perdi tudo e meu backup estava no interior”, “O trabalho ficou com meu parceiro”, “Achei que meu parceiro tinha enviado”, “Meu parceiro não fez a parte dele”, etc.

ATENÇÃO!!!! No caso de atraso na entrega do Milestone 1, seu trabalho não será analisado

por outra dupla e isso implicará em uma perda de 40% da nota no Milestone 2.

Durante os Milestones serão realizadas sessões de acompanhamento do trabalho (ver cronograma de aulas). Estas sessões serão usadas para esclarecer dúvidas e para o professor avaliar a evolução do trabalho. A percepção do professor quanto à não evolução do trabalho durante estes encontros será levada em consideração na nota final do projeto

Milestone 1 e 2

O que vou considerar:

- Compilação (Nota 0 se não compilar. Não aceito o projeto) . Mesmo que corrigido depois, haverá penalização de 20% do valor final da nota.
- Funcionamento. Vou jogar um pouco para observar o jogo funcionando. Se não funcionar, paro de avaliar.
- Qualidade dos testes (testes de unidade usando JUNIT)
- Qualidade da documentação (comentários e JavaDOC) .
 - 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. Se não gerar o Javadoc, desconsiderarei o quesito documentação.
- Qualidade do design
- Qualidade do relatório ou da apresentação
- Darei uma nota proporcional à participação de cada integrante da dupla. Levarei em consideração o caso em que um dos integrantes da dupla fez o trabalho sozinho, minimizando o rigor na avaliação.

Observações Finais

Os testes de unidade (JUNIT) são parte do seu código e da documentação do seu programa. Dêem importância porque eles representam parte da nota.