



UNIVERSIDADE FEDERAL DE PELOTAS
CENTRO DE DESENVOLVIMENTO TECNOLÓGICO
CURSOS DE CIÊNCIA E ENGENHARIA DE COMPUTAÇÃO
Disciplina de Programação Orientada a Objetos - 2022/2
Prof. Felipe de Souza Marques (felipem@inf.ufpel.edu.br)
Prof. Ulisses Brisolara Correa (ub.correa@inf.ufpel.edu.br)
Prof. Rafael Burlamaqui Amara (rafael.amaral@inf.ufpel.edu.br)

Trabalho Final de POO

O objetivo deste trabalho é exercitar os conceitos de Programação Orientada a Objetos, considerando abstração, encapsulamento, herança e polimorfismo. O trabalho consiste em implementar o jogo “Combate” utilizando a linguagem Java. Seu programa precisa ter uma interface gráfica. Todos os passos para a construção do programa deverão ser descritos em um relatório.

OBSERVAÇÃO: Este documento poderá ser alterado em sala de aula ou no site da disciplina sem aviso prévio. Portanto, compareçam às aulas e visualizem esse documento no site sempre que possível.

Da Descrição do Jogo



O Combate é a versão brasileira do Stratego, originário de “La Haye” nos Países Baixos em 1940. Entretanto, o Stratego provém do jogo “L’Attaque”, inventado por Hermance Edan em Paris no ano 1908. É um jogo praticado por duas pessoas em que o objetivo é capturar a bandeira adversária (existem variações onde a bandeira pode ser alterada por um prisioneiro).

O tabuleiro é um quadriculado com dez casas na horizontal por dez casas na vertical. Existem dois lagos que

ocupam 4 casas cada um. O espaço para cada jogador organizar o seu exército de 40 peças é de 10x4 casas.

Existem dois tipos de peças: As móveis e as imóveis. Os exércitos são diferenciados pelas cores azul e vermelho. Inicialmente as peças tinham suas insígnias em baixo relevo e pintadas em prata. Posteriormente, o formato das peças permaneceu, mas foram mudadas as identificações para um adesivo com o desenho dos respectivos soldados. Hoje em dia os exércitos do jogo são preto e verde. As imóveis são uma bandeira (prisioneiro) e seis bombas (minas terrestres).



A bandeira (prisioneiro) deve ficar escondida, pois quando é capturada pelo adversário, o jogo acaba e o que teve sua bandeira (prisioneiro) capturada perde. O jogador também pode perder a partida quando não tiver mais opção de movimento.

As bombas são peças imóveis que derrotam qualquer outra, com exceção do cabo-armeiro. As peças móveis são utilizadas para invadir o território adversário, derrotar suas peças e capturar a bandeira, sendo esta, o objetivo do jogo. Cada peça possui um nível e quanto maior, mais habilidades de combate estão associadas. Cada jogador possuirá as seguintes peças:

- Um espião (agente secreto), Nível 1, que é a única peça que pode derrotar o marechal.
- Oito soldados, Nível 2, que podem mover mais de uma casa por vez, a corrida. Um soldado não pode correr e atacar na mesma vez.
- Cinco cabos-armeiros (cabo), Nível 3, que podem desarmar as bombas.
- Quatro sargentos, Nível 4.
- Quatro tenentes (subtenente), Nível 5.
- Quatro capitães (tenente), Nível 6.
- Três maiores (capitão), Nível 7.
- Dois coronéis (major), Nível 8.
- Um general (coronel), Nível 9.
- Um marechal (general), Nível 10, que pode ser morto pelo espião, ou seja, entre o espião e o marechal (general), quem atacar primeiro, vence a batalha.

Da Descrição das Atividades/Requisitos desse Trabalho

1. CARACTERÍSTICAS DO JOGO QUE SERÁ DESENVOLVIDO

Nesse trabalho, será utilizado o conjunto de regras do jogo STRATEGY no tabuleiro 5x5, disponível no Google Play¹, ilustrado pela Figura 1.

Assim, utilize os conceitos de orientação objeto (como composição, encapsulamento, herança e polimorfismo) para modelar este jogo. Estes conceitos não podem ser oriundos apenas da utilização de pacotes para a implementação da interface gráfica. Você deve modelar **SUAS PRÓPRIAS CLASSES**, identificando seus atributos e métodos.

¹ <https://play.google.com/store/apps/details?id=com.famlinkup.stratego.free&hl=pt>



Figura 1 - Tela do Jogo Strategy

2. ATIVIDADE 1

Criar uma janela de boas vindas para o jogador e, logo em seguida, perguntar se as peças serão distribuídas aleatoriamente ou serão definidas pelo usuário.

- Definir Posição
 - Mostrar uma grade/tabela de botões: o usuário deve selecionar a peça a ser posicionada e marcar a posição desejada nesta tabela através do clique do mouse (ao clicar no botão, este deve ser desabilitado e a peça marcada nesta posição). Lembrando que as regiões da tabela devem ser respeitadas para cada jogador.
 - Ofereça uma opção “DEBUG”, onde o usuário poderá ver a posição de todas as peças do jogador e do computador. Esta opção exibirá a tabela do jogo revelando a posição de cada peça. Essa função é importante para testes e correção do trabalho.
- Posição Aleatória



- O programa deverá gerar a tabela com todas as peças em posições aleatórias somente na área reservada de cada jogador. Lembre-se de mesmo assim proteger a bandeira.

Observação: Deve existir um lago em uma posição aleatória, que não seja uma posição reservada para os jogadores. Distribua: 1 Bandeira, 2 Bombas, 1 Espião (Nível 1), 3 Soldados (Nível 2), 2 Cabo Armeiros (Nível 3) e 1 Marechal (Nível 10).

3. ATIVIDADE 2

Após posicionar todas as peças, o jogador deve clicar em um botão para prosseguir (por exemplo, o nome do botão poderia ser “Jogar”). Ao clicar neste botão, uma nova janela deve ser criada com a grade/tabela de botões do jogo. As peças do time do usuário estarão visíveis e identificadas. Já as peças do computador estarão apenas visíveis, mas não estarão identificadas. Esta tela também deve conter as seguintes opções:

- Dica:
O jogador informa uma posição e o programa indica se existe alguma bomba na coluna informada (no máximo 2 dicas por jogo).
- Sair:
Encerrar o jogo e mostrar as opções “Reiniciar Jogo” e “Novo Jogo”.
Uma jogada é definida como segue:
- O usuário deve escolher uma peça (clicar em um botão da tabela que possua uma peça do seu time), logo em seguida, deve indicar o local de destino (clicar em um botão da tabela para qual a peça possa se mover). As peças móveis poderão deslocar-se apenas uma posição na horizontal e vertical, com exceção do soldado que poderá avançar até o limite do tabuleiro ou até a primeira peça do adversário.
- Se na posição de destino existir uma peça, as regras descritas na primeira página definirão qual peça permanecerá na posição, caso as peças possuam o mesmo nível, ambas serão retiradas do tabuleiro (da mesma forma que acontece no jogo STRATEGY).

Após cada jogada, o computador deverá realizar uma jogada de forma aleatória (mas respeitando as regras do jogo).

4. ATIVIDADE 3

No final do jogo, criar uma janela informando se o jogador ganhou ou perdeu o jogo e mostrar as opções “Reiniciar Jogo” e “Novo Jogo”.



Obs: O jogo acaba quando alguma bandeira for capturada ou quando algum jogador não tiver mais opção de movimento.

5. ATIVIDADES COMPLEMENTARES

- Lembre-se que você deve implementar as jogadas do computador!
- O computador posiciona as peças de maneira aleatória.
- Após cada jogada do usuário, o computador também faz sua jogada e o programa deve mostrar a quantidade de cada peça atualizada.

Dos Grupos

O **trabalho** poderá ser feito em grupos de até **TRÊS** pessoas.

Da Entrega do Trabalho

O **trabalho** deve ser **entregue via e-aula**:

- Até o dia **08/05/2023 às 09:00**, com a possibilidade de obter **valor integral da nota**.
- Até às 23:59 do dia 10/05/2023 (considerando **1 ponto a menos por dia de atraso**. Por exemplo: Trabalhos entregues no dia 08/05/2023 às 10:00 terão nota máxima igual a 8).
- **Após às 23:59 do dia 10/05/2023 os trabalhos não serão mais avaliados em hipótese alguma.**

Um representante do grupo deve submeter os seguintes arquivos para completar a entrega do trabalho:

- Relatório do projeto, incluindo:
 - Capa, identificando o título do trabalho e os integrantes do grupo.
 - Introdução.
 - Breve descrição das funcionalidades implementadas, e um resumo das dificuldades encontradas durante o trabalho.
 - Passos para compilar e rodar o programa gerado.
 - Exemplos de utilização. (Algumas telas durante o jogo)
 - Diagrama de classes, com nome das classes, atributos e métodos.
 - Breve explicação em onde e como foram aplicados os conceitos de orientação objeto.
 - Conclusão.



UNIVERSIDADE FEDERAL DE PELOTAS
CENTRO DE DESENVOLVIMENTO TECNOLÓGICO
CURSOS DE CIÊNCIA E ENGENHARIA DE COMPUTAÇÃO
Disciplina de Programação Orientada a Objetos - 2022/2
Prof. Felipe de Souza Marques (felipem@inf.ufpel.edu.br)
Prof. Ulisses Brisolara Correa (ub.correa@inf.ufpel.edu.br)
Prof. Rafael Burlamaqui Amara (rafael.amaral@inf.ufpel.edu.br)

- Sugestões de trabalhos futuros.
- Arquivos fonte: para simplificar, pode-se compactar o diretório do projeto em um arquivo ZIP, que deverá ser entregue via e-aula.

Da Avaliação

Como resultados desse trabalho, deverão ser entregues a implementação (código fonte compilável, representando 70% da nota) e um relatório (representando 30% da nota).

Importante: Os trabalhos deverão ser apresentados para os respectivos professores durante os horários das aulas da última semana do semestre (08 a 12 de maio). **Trabalhos que não forem apresentados ou que não compilarem, não serão avaliados!**

Critérios de avaliação da implementação:

- (2 pontos) Habilidade em estruturar programas pela decomposição da tarefa em subtarefas, utilizando subprogramação para implementá-las.
- (2 pontos) Emprego dos conceitos de POO: abstração, encapsulamento, herança e polimorfismo.
- (3 pontos) Atendimento aos requisitos mínimos da especificação do problema.

Código de Honra

O trabalho deve ser implementado na sua totalidade, sem uso de código de outros (colegas ou não). O trabalho enviado deve representar um esforço honesto em resolver o problema - isto é, não é algo "pela metade", que não implementa funcionalidades essenciais. Violações a esta conduta serão penalizadas e o violador não só terá nota nula neste trabalho, ficando portanto sem parte da nota da disciplina. Os plágios estão sujeitos a sanções administrativas pelo colegiado do curso.