

# MAC0218 - Técnicas de Programação II

## EP Galaxy Raiders - Parte 1

Alfredo Goldman, Cainã Setti, Dayanne Gomes e Renato Cordeiro

Instituto de Matemática e Estatística – USP

23 de abril de 2023

### 1 Introdução

Galaxy Raiders é um jogo inspirado por clássicos da Era de Ouro dos Arcades: Space Invaders (1978), Star Raiders (1979), Asteroids (1979) e Galaga (1981). Esses jogos se popularizaram graças aos fliperamas (máquinas especializadas para jogos eletrônicos) e aos primeiros videogames (dispositivos caseiros para jogos eletrônicos), introduzidos pela empresa americana Atari.

No ambiente espacial onde o Galaxy Raiders está inserido, o jogador precisa controlar a nave para desviar de objetos que possam colidir na nave e causar dano na sua estrutura e também controlar os canhões de laser para lançar projéteis para destruir os asteroides.

O jogo é composto por duas partes. Um componente para visualização do jogo, chamado de front-end ou interface Web, onde o usuário consegue interagir com o jogo; e um componente de lógica do jogo, chamado de back-end ou API, onde o motor do jogo processa os comandos do usuário, atualiza o estado do jogo, e o envia para visualizar. O repositório do projeto pode ser encontrado em <https://github.com/galaxy-raiders>. Nesta parte do EP, a API será explorada.

A API foi estruturada utilizando o estilo arquitetural hexagonal. Este estilo arquitetural propõe desacoplar a lógica de negócio com dependências externas utilizando estruturas conhecidas como portas e adaptadores. A estrutura de pastas do código está dividida em três pacotes:

- Pacote core, com a lógica do jogo. Ele é subdividido em dois pacotes para as funcionalidades de física (core.physics) e do jogo (core.game).
- Pacote ports, com as interfaces para dependências externas, fornecendo um ponto de testabilidade para a lógica do jogo. Ele é subdividido num pacote com portas para a interface do usuário (ports.ui).
- Pacote adapters, com a implementação das dependências externas. Ele é subdividido em dois pacotes que implementam uma interface de texto (adapters.tui) e uma interface web (adapters.web) para interagir com o jogo.

## 2 Tarefa

Esta parte do EP aborda TDD (Test-Driven Development). Utiliza-se TDD para implementar a física básica do jogo. Os testes estão disponíveis em `/app/src/test/kotlin/galaxyraiders/core/physics/`. Você deverá implementar as classes `Point` e `Vector` para rodar o código do jogo, baseando-se nos testes criados. Para isso, precisa-se modificar o código que se encontra em:

- `/app/src/kotlin/galaxyraiders/core/physics/Vector2D.kt`, e
- `/app/src/kotlin/galaxyraiders/core/physics/Point2D.kt`.

Para o desenvolvimento do EP, deve-se atentar à utilização de Docker. A documentação inclui um README com instruções para execução. O front-end encontra-se no repositório <https://github.com/galaxy-raiders/galaxy-raiders-web>. Não é necessário alterá-lo.

## 3 Entrega

O EP pode ser feito em até duas pessoas. A entrega deve feita por uma url do repositório de git. Cada pessoa deve fazer um fork do repositório <https://github.com/galaxy-raiders/galaxy-raiders-api> e colocar a url do repositório no eDisciplinas. Caso seja feito em dupla, cada um dos integrantes deve fazer sua entrega no sistema, indicando qual é sua dupla.

No repositório é necessário criar uma branch `p1`, que deve conter a alteração do código do pacote `physics` do jogo.

## 4 Avaliação

O EP será avaliado a partir da implementação do código de cada parte da tarefa. A primeira parte (`p1`), de implementação da física do jogo com TDD, considera os pontos:

- Código compila sem erros e warnings: 2.0
- Boas práticas de programação e clareza do código: 2.0
- Código executa sem erros e produz os resultados desejados: 6.0