

# Projeto de COO - 1º semestre/2024

## Aplicação de princípios OO a um código procedural

### Descrição

Neste projeto, a ser feito em grupo, vocês devem reescrever o código de um jogo, disponibilizado no arquivo **Projeto\_COO.zip**, de modo a aplicar diversos conceitos estudados na disciplina *Computação Orientada a Objetos*. O jogo, trata-se de um *shoot 'em up* ([http://en.wikipedia.org/wiki/Shoot\\_'em\\_up](http://en.wikipedia.org/wiki/Shoot_'em_up)) vertical bastante simples, sem acabamento (não possui tela de título, placar, vidas, fases, chefes, *power-ups*, etc) e que roda de forma indefinida (até que o jogador feche a janela do jogo).

Embora funcione, seu código **não** foi elaborado seguindo bons princípios de orientação a objetos. Apesar de escrito em Java, o código foi elaborado seguindo um estilo de programação procedural e, mesmo considerando este estilo, ainda assim não muito bem feito uma vez que há muito código redundante. Existem portanto inúmeras oportunidades de melhoria do código. Os dois principais aspectos que devem ser trabalhados no desenvolvimento deste projeto são:

- A aplicação de **princípios de orientação a objetos**, através da criação de uma boa estrutura de interfaces e classes bem encapsuladas, definição de uma hierarquia de classes/interfaces adequada, e uso de recursos como herança/composição.
- O uso das **coleções Java** ao invés de *arrays* para manter/gerenciar as entidades do jogo que aparecem em multiplicidade (inimigos, projéteis, etc).

O código do jogo é composto por dois arquivos fonte: **Main.java** e **GameLib.java**. No primeiro arquivo está implementada toda a lógica do jogo, enquanto o segundo implementa uma mini biblioteca com recursos úteis no desenvolvimento de jogos: inicialização da interface gráfica, desenho de figuras geométricas simples e verificação de entrada através do teclado.

O foco da reescrita de código deve ser a classe **Main**. Pode-se assumir que a classe **GameLib** é uma caixa-preta à qual não se tem acesso ao código-fonte (como se realmente fosse uma biblioteca feita por terceiros) e portanto ela não precisa ser modificada neste trabalho, apenas utilizada.

Além da reescrita do código, também deverão ser implementadas algumas funcionalidades extras no jogo. Se a reescrita do código for bem feita, a implementação destas funcionalidades extras deve acontecer de forma relativamente simples. As funcionalidades extras a serem implementadas são as seguintes:

- Criação de um **novo tipo de inimigo**. Este novo tipo deve apresentar comportamento de ataque e movimentação diferente dos inimigos já existentes no jogo. Deve apresentar visual próprio também.
- Criação de um **power-up**. Um *power-up* é um item que aparece na tela de jogo e quando coletado melhora algum aspecto do comportamento da nave controlada pelo jogador. O efeito produzido pelo *power-up* fica a cargo do grupo.
- **Pontos de vida para o jogador**: deve ser implementado um esquema de pontos de vida para o

jogador. O jogador deve possuir uma certa quantidade de pontos de vida (definida via linha de comando) e ao ser atingido por um projétil (ou inimigo) a quantidade de pontos de vida deve ser reduzida em 1 unidade. A nave do jogador só deve explodir quando a quantidade de pontos de vida chegar a zero. Deve-se implementar ainda a exibição de uma barra de vida, ilustrando graficamente quantos pontos de vida o jogador ainda possui, e também algum tipo de *feedback* gráfico que indique quando o jogador é atingido por um projétil ou inimigo (sugestão: implementar algo parecido com o efeito flash que pode ser visto aqui: <https://youtu.be/j56eUNx4sZk?t=4m26s>).

Também faz parte do trabalho a elaboração de um relatório que deve documentar:

- Críticas ao código original do jogo.
- Descrição e justificativa para a nova estrutura de classes/interfaces adotada.
- Descrição de como as coleções Java foram utilizadas para substituir o uso de *arrays*.
- Descrição de como as novas funcionalidades foram implementadas e como o código orientado a objetos ajudou neste sentido.

## Algumas observações sobre o uso das coleções Java

Na versão original do código, é feito um uso extensivo de *arrays* para gerenciar entidades do jogo que ocorrem em multiplicidade (inimigos, projéteis, etc). Devido ao fato de *arrays* serem estruturas de armazenamento estáticas, todos os *arrays* são alocados com tamanhos fixos e suas posições são reutilizadas sempre que uma entidade associada a um determinado índice torna-se inativa (quanto sai da tela, no caso dos inimigos e projéteis, ou quando é abatida pelo jogador, no caso dos inimigos).

Fazendo-se bom uso das coleções Java, para armazenar e gerenciar esses conjuntos de entidades, a reutilização de posições deixa de ser necessária pois todas as coleções implementadas pelo Java são dinâmicas. Contudo é importante não esquecer de remover as entidades que se tornam inativas da coleção que as armazena a fim de evitar vazamentos de memória durante a execução do jogo.

## Entrega

Este projeto pode ser feito (e recomenda-se que seja feito) em grupos de até 4 pessoas. Deve ser entregue:

- Código fonte reescrito.
- Relatório (em formato **PDF**).

A entrega deverá ser feita pelo eDisciplinas, na atividade aberta para este fim, até o final do dia 23/06/2024. Entregue um único arquivo **ZIP** contendo tanto o código reescrito quanto o relatório.

Boa diversão!