

Projeto de COO - 1º semestre/2015

Refatoração do código de um jogo

Descrição

Este projeto consiste em realizar a refatoração do código de um jogo, disponibilizado no arquivo ProjetoCOO.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) 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 ser escrito em Java, o código foi elaborado seguindo um estilo de programação estruturada e mesmo assim não muito bem feito, com muita repetição de código, havendo numerosas oportunidades de melhoria. Há três principais aspectos do código que devem ser melhorados na refatoração do código:

- Aplicação de princípios de orientação a objetos, através da definição de uma boa estrutura de classes, interfaces e hierarquia de classes/interfaces.
- Uso da API de coleções do Java ao invés de *arrays* para manter/gerenciar o conjunto de informações relativas às entidades do jogo (inimigos, projéteis, etc).
- Aplicação de padrões de projeto, com o objetivo de tornar a extensão/manutenção do código mais fácil e flexível. **Devem ser aplicados pelo menos 2 padrões de projeto** (observação: o uso de alguma biblioteca/API do Java que implementa um determinado padrão não conta como aplicação de um padrão).

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 e verificação de entrada através do teclado.

O foco da refatoração do código deve ser a classe **Main**. Você pode assumir que a classe **GameLib** é uma caixa-preta à qual você não tem acesso ao fonte (como se realmente fosse uma biblioteca feita por terceiros) e portanto ela não precisa ser trabalhada na refatoração, apenas utilizada. Contudo, se por algum motivo você desejar modificá-la, para melhorá-la ou aplicar algum padrão de projeto, fique à vontade também.

Além da refatoração do código, você também deve implementar:

- Um **novo tipo de inimigo** que deve apresentar comportamentos de movimento e de disparo de projéteis que sejam diferentes dos comportamentos apresentados pelos dois tipos de inimigos já presentes no jogo.
- **Power-ups**: itens que aparecem na tela de jogo e quando coletados pela nave do jogador modificam o comportamento de disparo da mesma. Devem ser implementados dois tipos de *power-ups* diferentes.

Além da refatoração do código, faz parte do trabalho a elaboração de um relatório no qual devem ser documentadas:

- Suas críticas ao código original do jogo.
- Descrição e justificativa da nova estrutura de classes/interfaces adotada.
- Descrição de como a API de coleções foi utilizada.
- Descrição dos padrões de projetos adotados e justificativa para a aplicação dos mesmos.

Observações sobre o uso da API de coleções

Na versão original do código, é feito uso extensivo de *arrays* para gerenciar conjuntos de informações relacionados às diversas entidades do jogo (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 relacionada a 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 da API de coleções do Java, para armazenar e gerenciar os 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 se evitar vazamentos de memória durante a execução do jogo.

Entrega

Este projeto pode ser feito em grupos de até 4 pessoas. Deve ser entregue:

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

A entrega deverá ser feita pelo TIDIA-Ae, até o dia 01/07/2015. Entregue um único arquivo **ZIP** contendo tanto o código refatorado quanto o relatório.

Boa diversão!