Projeto de COO - 1ºsemestre/2016

Refatorando o 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 escrito em Java, o código foi elaborado seguindo um estilo de programação estruturada e, mesmo considerando este estilo, não muito bem feito, com muito código redundante. Existem portanto inúmeras oportunidades de melhoria do código. Há três principais aspectos que devem ser trabalhados durante o desenvolvimento deste projeto:

- 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**. 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 trabalhada na refatoração, apenas utilizada. Contudo, se houver vontade ou alguma razão especial para modificá-la, melhorando-a ou aplicando algum padrão de projeto, ela também pode ser modificada.

Além da refatoração do código, também deverão ser implementadas algumas funcionalidades extras no jogo. **Tais funcionalidades serão anunciadas em breve, em um adendo a este enunciado**.

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 a API de coleções foi utilizada em substituição ao uso de *arrays*.
- Descrição dos padrões de projetos adotados e justificativa para a aplicação dos mesmos.
- Descrição de como as novas funcionalidades foram implementadas e como a refatoração do código ajudou neste sentido.

Algumas 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 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 30/06/2016. Entregue um único arquivo **ZIP** contendo tanto o código refatorado quanto o relatório.

Boa diversão!