

Relatório de Erro

Marco Antonio Cabral dos Santos Junior¹, Osvaldo Gomes Rodrigues¹

¹ Faculdade de Informática – PUCRS - Av. Ipiranga, 6681 - Prédio 32
Caixa Postal 1429 90.619-900 - Porto Alegre - RS Brasil

{marco.antonio.002, osvaldo.rodrigues}@acad.pucrs.br

1. Atividade Desenvolvida

O trabalho deve implementar um jogo ao estilo "Space Invaders" a partir do framework fornecido pelo professor. Este framework constitui-se de uma série de classes e interfaces quais deverão ser derivados os personagens e as regras de funcionamento do jogo.

O jogo consiste de uma série de personagens animados que descem em direção a parte de baixo da tela através de movimentos próprios. O objetivo do jogador é controlar um canhão que se movimenta para a direita ou esquerda na parte de baixo da tela e que é capaz de atirar para eliminar os invasores. Na medida em que os invasores são eliminados o jogador acumula pontos. O jogo só acaba se algum invasor atinge a parte de baixo da tela ou se o canhão é destruído.

2. Regras Disponibilizadas

- Devem existir diferentes tipos de personagens que são representados por uma hierarquia de herança. Cada tipo de personagem tem uma aparência e um comportamento diferente. Por exemplo, alguns apenas se movimentam lentamente da esquerda para a direita e da direita para a esquerda, e sempre que chegam numa extremidade eles descem. Alguns podem ter um movimento semelhante, mas também podem atirar. Outros também podem ser mais difíceis de eliminar, necessitando receber dois ou mais tiros para serem eliminados. Por fim pelo menos um dos tipos deve prever comportamento de grupo.
- Devem ser explorados os padrões de projeto “singleton”, “factory” e “composite”.
- Os personagens devem, obrigatoriamente, serem derivados das classes fornecidas, explorando adequadamente os conceitos de programação orientada a objetos.
- A interface com o usuário deve ser aprimorada usando JavaFX.
- Os métodos que definem o comportamento dos personagens devem ser anotados com JML. Deve ser implementado um pequeno “driver” de teste capaz de demonstrar que violações às especificações são detectáveis pelo JML.
- O jogo deve ser capaz de manter a relação das 10 melhores pontuações em arquivo.

3. Erro Penultima Regra

Na realização dessa etapa foi constatado que, o JML não suporta o JavaFx, impossibilitando a compilação individual das classes Java, fato confirmado pelo professor da disciplina de Programação Orientada a Objetos BERNARDO COPSTEIN, o qual acabou impossibilitando a verificação dos métodos que definem o comportamento dos personagens que devem ser anotados com JML. Como demonstra na figura a seguir (ver figura 1):

```

rm: *.class: No such file or directory
error: An internal JML error occurred, possibly recoverable. Please report the bug with as much information as you can.
Reason: Internal exception: class java.lang.NullPointerException
java.lang.NullPointerException
    at org.jmlspecs.openjml.esc.JmlAssertionAdder.addImplicitConversion(JmlAssertionAdder.java:7576)
    at org.jmlspecs.openjml.esc.JmlAssertionAdder.convertArgs(JmlAssertionAdder.java:5679)
    at org.jmlspecs.openjml.esc.JmlAssertionAdder.applyHelper(JmlAssertionAdder.java:5889)
    at org.jmlspecs.openjml.esc.JmlAssertionAdder.visitApply(JmlAssertionAdder.java:5683)
    at com.sun.tools.javac.tree.JCTree$JMethodInvocation.accept(JCTree.java:1465)
    at com.sun.tools.javac.tree.TreeScanner.scan(TreeScanner.java:49)
    at org.jmlspecs.openjml.esc.JmlAssertionAdder.convertExpr(JmlAssertionAdder.java:1317)
    at org.jmlspecs.openjml.esc.JmlAssertionAdder.visitExec(JmlAssertionAdder.java:4646)
    at com.sun.tools.javac.tree.JCTree$JExpressionStatement.accept(JCTree.java:1296)
    at com.sun.tools.javac.tree.TreeScanner.scan(TreeScanner.java:49)
    at org.jmlspecs.openjml.esc.JmlAssertionAdder.convert(JmlAssertionAdder.java:1235)
    at org.jmlspecs.openjml.esc.JmlAssertionAdder.convertMethodBodyNoInit(JmlAssertionAdder.java:1024)
    at org.jmlspecs.openjml.esc.JmlAssertionAdder.visitJmlMethodDecl(JmlAssertionAdder.java:10745)
    at org.jmlspecs.openjml.JmlTree$JmlMethodDecl.accept(JmlTree.java:1147)
    at com.sun.tools.javac.tree.TreeScanner.scan(TreeScanner.java:49)
    at org.jmlspecs.openjml.esc.JmlAssertionAdder.visitJmlClassDecl(JmlAssertionAdder.java:9653)
    at org.jmlspecs.openjml.JmlTree$JmlClassDecl.accept(JmlTree.java:1065)
    at com.sun.tools.javac.tree.TreeScanner.scan(TreeScanner.java:49)
    at org.jmlspecs.openjml.esc.JmlAssertionAdder.convert(JmlAssertionAdder.java:1235)
    at org.jmlspecs.openjml.JmlCompiler.rac(JmlCompiler.java:512)
    at org.jmlspecs.openjml.JmlCompiler.desugar(JmlCompiler.java:389)
    at com.sun.tools.javac.main.JavaCompiler.compile2(JavaCompiler.java:898)
    at org.jmlspecs.openjml.JmlCompiler.compile2(JmlCompiler.java:591)
    at com.sun.tools.javac.main.JavaCompiler.compile(JavaCompiler.java:867)
    at com.sun.tools.javac.main.Main.compile(Main.java:549)
    at com.sun.tools.javac.main.Main.compile(Main.java:410)
    at org.jmlspecs.openjml.Main.compile(Main.java:556)
    at com.sun.tools.javac.main.Main.compile(Main.java:399)
    at com.sun.tools.javac.main.Main.compile(Main.java:398)
    at org.jmlspecs.openjml.Main.execute(Main.java:392)
    at org.jmlspecs.openjml.Main.execute(Main.java:350)
    at org.jmlspecs.openjml.Main.execute(Main.java:337)
    at org.jmlspecs.openjml.Main.main(Main.java:309)
Note: ./Game.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
1 error
MacBook-Pro-de-Osvaldo:Atual Osvaldo$

```

Figura 1. Erro encontrado no JML

Agravante que nos prejudicou, na implementação e desenvolvimento dessa regra no trabalho, pois não podemos testar a validade das pré e pós condições, elaboradas por nós com o JML.