Instituto Politécnico de Beja
Escola Superior de Tecnologia e Gestão
Licenciatura em Engenharia Informática & Licenciatura em Informática de Gestão
Programação 1

Trabalho Prático de Avaliação 2
Falling figures

Beja, 7 de Novembro de 2008
(revisto em 9 de Novembro com sugestões para funcionalidades extra, no final da Secção 4, e em 10 de Novembro para alteração da numeração dos requisitos)

Antes de realizar este trabalho prático deve completar o Guia Prático 4.

Deve verificar cuidadosamente se cumpre integralmente cada um dos requisitos. O cumprimento parcial de um qualquer requisito implica a sua avaliação como não cumprido. Assim, é mais importante cumprir totalmente alguns do que parcialmente todos. Na verdade, cumprir parcialmente todos equivale a não cumprir nenhum.

1 Introdução

O objectivo geral deste trabalho é o de construir um jogo para um só jogador cujo objectivo é o de acertar em objectos que se deslocam de cima para baixo no ecrã.

IMPORTANTE: se não utilizar a linguagem Java™ ou se não utilizar a biblioteca ACM apresentada nas aulas o trabalho será avaliado com zero valores.

2 Requisitos Essenciais

Req. 1 - Objectos cadentes Após um clique no rato, surge um objecto numa posição **aleatória** no topo da janela. Este objecto inicia imediatamente um movimento de cima para baixo. Quando o objecto desaparece da janela, surge um novo objecto no topo da janela que efectua o mesmo tipo de movimento. Tal é feito para a quantidade total de objectos indicada pelo jogador no início no jogo. O jogo termina quanto logo que o último objecto é atingido ou atinge a parte de baixo da janela e desaparece.

Req. 2 - Aleatoriadade Os objectos do requisito anterior podem ser quadrados ou círculos de dimensões indicadas pelo jogador no início do jogo.

Req. 3 - Detecção de acertos O jogador tenta acertar o objecto em movimento. Quando atinge o objecto, a pontuação é aumentada um ponto e o objecto desaparece do ecrã.

Req. 4 - Pontuação No canto superior esquerdo da janela, está sempre presente uma label que indica a pontuação actual. Esta pontuação é actualizada logo que o objecto é atingido.

3 Requisitos Não Essenciais

Req. 5 - Margem para pontuação e área de jogo Os objectos deslocam-se apenas dentro de um zona da janela que exclui uma margem esquerda de 50 pixéis. Essa margem é utilizada para colocar a label com a pontuação. A área de jogo deve ter uma cor diferente da margem.

Req. 6 - Repetição Adicione o código seguinte, ou semelhante, que permite perguntar ao jogador se ele pretende jogar mais um jogo. Se o jogador responder "S" então poderá jogar mais um jogo. Se responder algo de diferente, o programa termina. Para tal adicione o seguinte código ao seu programa:

```
String oneMore = ""; // utilizada na condição while
do // início do ciclo
{
   ...
   oneMore = dialog.readLine("Mais um jogo?");
} while (oneMore.equals("S"));  // repete enquanto esta
                                // condição for verdadeira
```

Note a comparação utilizando o método `equals` da classe String: perguntamos ao objecto `String`, cujo nome (referência ou endereço) está na variável referência `oneMore`, se ele é igual ao objecto `String` "S", ou seja se tem o mesmo valor.

Req. 7 - Pontuação máxima Com o ciclo anterior, é agora possível jogar vários jogos em cada execução do programa. Adicione então a seguinte funcionalidade: no final de cada jogo é mostrada a pontuação máxima nessa execução do programa. Para tal vai necessitar de mais um atributo onde irá guardar essa pontuação:

```
...
private int maxPoints;

public FallingFigures()
{
   ...
   this.maxPoints = 0; // inicialização a zero
}
```

No final de cada jogo, terá de verificar se a pontuação obtida é maior do que a pontuação máxima até à data e, se for, actualizar a pontuação máxima. Para tal deve utilizar uma instrução **`if`**. Além da label, na margem de pontuação é mostrada a pontuação máxima até à data.

Req. 8 - Decomposição em métodos O seu programa não deve conter qualquer método com mais de 30 linhas e devidamente formatadas. Para tal, cada operação com identidade própria deve ser um método. Por exemplo, muitas vezes, os ciclos realizam uma operação identificável que pode passar a estar num método próprio. O mesmo se passa com o código para o utilizador indicar valores. Cada método deve ter um nome "bom", ou seja, que permita perceber que operação é que o método realiza.

Req. 9 - Regras de estilo e elegância do código O código entregue deve respeitar as regras de estilo, nomeadamente **todas** as seguintes:

Comentários Antes da classe e antes de cada método deve escrever um comentário java doc(/** */) que explique o que o método faz, os respectivos parâmetros e valor devolvido (se existentes). Os comentários podem estar em português mas tente colocá-los em inglês.

Identificadores em inglês Os nomes de todas as variáveis, métodos e classes devem estar em inglês.

Nomes das variáveis, constantes e classe Utilização de letras minúsculas/-maiúsculas e informação transmitida pelos nomes;

Alinhamento das chavetas Cada uma por baixo da correspondente.

Os espaçamentos Antes e depois dos operadores e das vírgulas.

Indentação coerente e para cada bloco.

Utilização do `this` Utilização da referência **`this`** antes do nome das operações que se aplicam ao objecto da janela de desenho (por exemplo, **`this`**`.add(oneRectangle)`.

Req. 10 - Auto-avaliação No ficheiro .zip entregue deve conter um ficheiro de texto com o nome "auto-aval.txt"que indica quais os requisitos que estão **totalmente** cumpridos (os únicos que contam como cumpridos) e a classificação resultante.

4 Regras para a Realização e Entrega

Este trabalho será avaliado, de acordo com o previsto no Guia da Unidade Curricular, em função da quantidade de requisitos essenciais e não essenciais que estejam **totalmente** cumpridos. **Para ser considerado como entregue cada trabalho tem de cumprir todos os seguintes requisitos:**

Quantidade de autores Os trabalhos a entregar têm de ser realizados por grupos de 2 alunos ou por um único aluno.

O que é entregue O trabalho entregue tem de ser um projecto BlueJ pronto a funcionar, sob a forma de um único ficheiro zip contendo toda a directoria do projecto. Antes de entregar, verifique que sabe pôr a funcionar o código no ficheiros zip entregue. Para tal parta desse ficheiro, descompacte-o e tente executá-lo no BlueJ. Tal será pedido na apresentação individual do trabalho. Se não conseguir pôr a funcionar o conteúdo do ficheiro zip entregue (no moodle e por e-mail) a classificação no trabalho será de zero valores.

Nome do projecto O nome do projecto em BlueJ (abaixo referido como `nome DoProjecto`) tem de respeitar um dos seguintes formatos, conforme o trabalho tenha sido realizado por um só aluno ou por dois alunos. Note que `Primeiro1` e `Ultimo1` representam o primeiro e o último nome de um autor, e `Primeiro2` e `Ultimo2` representam o primeiro e o último nome do outro autor. `Numero1` e `Numero2` representam os números de aluno de cada autor.

Para um autor:

```
Numero1_Primeiro1Ultimo1_TP2_P1_2008-2009
Por exemplo: 1232_AnaGomes_TP2_P1_2008-2009
```

Para dois autores:

```
Numero1_Primeiro1Ultimo1_Numero2_Primeiro2Ultimo2_TP2_P1_2008-2009
Por exemplo: 1232_AnaGomes_3454_JoaoSilva_TP2_P1_2008-2009
```

O trabalho é entregue compactando a directoria do projecto BlueJ num ficheiro .zip de forma a que este fique com o nome `nomeDoProjecto.zip`.

Entrega A entrega tem de ser feita num ficheiro no formato zip com o nome `nomeDo Projecto.zip` e por duas vias:

1. Na página da disciplina. Os trabalhos realizados por grupos de dois alunos devem ser entregues têm de ser entregues via moodle por **um** dos elementos do grupo.
2. Por e-mail, respeitando as seguintes regras: A entrega por e-mail é feita para `trabalhos.p1ARROBAgmail.com`. O e-mail a enviar deve conter em attach um único ficheiro no formato zip. O subject do e-mail deve ser o `nomeDoProjecto`. Pode entregar mais do que uma vez, desde que dentro do prazo. A última entrega dentro do prazo é a única que conta.

Data limite de entrega O trabalho tem de ser entregue no moodle e por e-mail até às **14:00 de 17 de Novembro de 2008**. Os trabalhos que não tenham sido entregues até essa data e hora, nem no moodle nem por e-mail, serão considerados como não entregues e avaliados com zero valores.

Finalmente, note que necessita de realizar mais do que o pedido para obter mais de 16 valores. Uma hipótese é ~~o desenho de uma ou mais ilusões ópticas realmente complicadas; outra será adicionar opções adicionais de configuração das ilusões pedidas.~~ a adição de níveis de dificuldade (objectos cada vez mais rápidos e/ou mais pequenos) de forma a que o jogo vá ficando mais difícil. A criatividade também pode justificar uma melhor classificação pelo que extras mais originais e sofisticados serão uma boa aposta. Note que estas estas adições só contam para a classificação do trabalho se forem consideradas suficientemente significativas e se **todos** os requisitos estiverem totalmente cumpridos.

5 Nota importante

Todas as contribuições para o trabalho que não sejam da exclusiva responsabilidade dos autores têm de ser identificadas (com comentários no código e referências no relatório) e a sua utilização bem justificada e expressamente autorizada pelo professor responsável. Justificações insatisfatórias, ausência de autorização, ou ausência de referências para trabalhos ou colaborações externas utilizadas serão consideradas fraude sempre que os trabalhos sejam considerados demasiado semelhantes para terem sido criados de forma independente. **Tal terá como consequência a reprovação na unidade curricular de todos os alunos envolvidos**. Excepcionalmente, poderão ser penalizados apenas os alunos que se declarem como únicos culpados. Assim, nenhum aluno deve dar cópia do seu código (ainda que em fase inicial) a outro. Por essa mesma razão não é boa ideia partilhar código com os colegas, quer directamente quer através do fórum. Naturalmente, cada aluno pode trocar impressões e esclarecer dúvidas com todos os colegas, mas deve escrever o seu código apenas com a colaboração do seu colega de grupo. Deve também saber escrever todo o código sozinho. Lembre-se que será avaliado num teste prático final em frente a um computador. A classificação neste trabalho prático fica dependente de uma apresentação individual do mesmo, tal como previsto no guia da unidade curricular.

Finalmente, leia com MUITA atenção todo o enunciado. A falha de parte do exigido num requisito implica o nãi cumprimento desse requisito e consequente penalização.

Bom trabalho!

João Paulo Barros