

UFF – Universidade Federal Fluminense
TIC – Instituto de Computação
TCC – Departamento de Ciência da Computação

TCC 00.309 | Programação de Computadores II | Turma A-1 | 2015.1
Professor Leandro Augusto Frata Fernandes

Trabalho Final
Mosaico de Imagens

Divulgação: 02/05/2015 | Última atualização: 02/05/2015 | Entrega: Vide cronograma

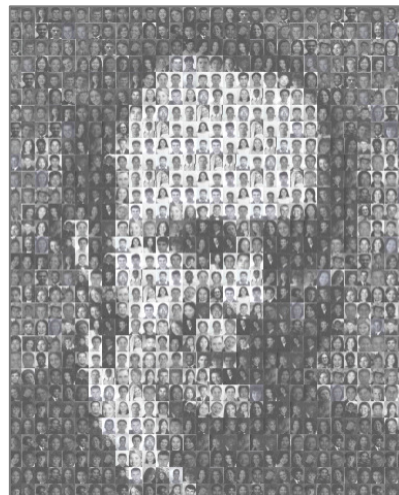
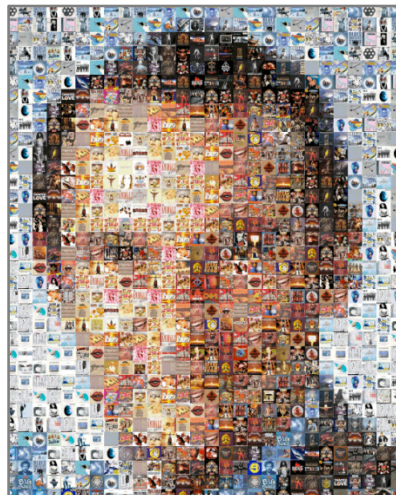
Instruções

- 1) Este trabalho é individual. Será atribuída nota ZERO a trabalhos entregues com partes do código-fonte identificadas como plágio.
- 2) A entrega do trabalho consiste no envio do código fonte Java organizado em um projeto NetBeans autocontido, i.e., que não dependa de ou inclua pacotes externo às APIs que compõe o Java SDK.
- 3) Serão considerados entregues trabalhos enviados para laffernandes@ic.uff.br ou laffernandes@gmail.com até as 23:59 da data de entrega definida acima. A única exceção a esta regra é o caso de ambas as contas de e-mail citadas estarem fora do ar. Será atribuída nota ZERO a trabalhos não entregues.
- 4) Leia atentamente o enunciado antes de proceder com as implementações.

Enunciado

O aluno deverá criar um pacote que conterá classes responsáveis pela aplicação de um **foto-mosaico**. Este pacote será utilizado por um aplicativo Java escrito pelo aluno e que, uma vez pronto, será invocado pelo usuário a partir do *prompt* de comando do DOS ou Linux.

Um **mosaico** é uma imagem criada pela composição de um grande número de objetos (historicamente pedaços de vidro ou pedra). Em um **foto-mosaico**, a imagem final é composta por um conjunto de imagens pequenas auxiliares chamadas ladrilhos. Quando vistos de perto, os ladrilhos devem ser identificáveis e, de longe, devem compor a imagem final. Dois exemplos de foto-mosaico podem ser vistos abaixo, um colorido e um em tons de cinza:



Neste trabalho, consideraremos apenas foto-mosaicos em tons de cinza. Cada pixel da imagem base informada como entrada deve ser substituído na imagem final por uma imagem pequena da coleção de ladrilhos. Logo, a resolução da imagem final será maior que a da imagem de entrada. O critério de escolha do ladrilho que vai compor o mosaico é baseado na distância da intensidade do pixel C_p e na intensidade média da imagem pequena C_m :

Se $|C_p - C_m| \leq \epsilon$ e a imagem da coleção de ladrilhos não foi usada mais de 10 vezes, então a imagem é escolhida e a troca é realizada. ϵ é um limiar informado pelo usuário como argumento na invocação do programa.

Com o intuito de facilitar a carga, manipulação e salvamento dos arquivos de imagem, está autorizado o uso das classes: `javax.imageio.ImageIO`, `java.awt.image.BufferedImage` e `java.io.File`. Segue um exemplo simples de como essas classes podem ser utilizadas para: (1) abrir um arquivo de imagem; (2) trocar a cor dos três primeiros pixels da primeira linha para, respectivamente, azul, verde e vermelho; e (3) salvar a imagem em um arquivo diferente do primeiro.

```
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import javax.imageio.ImageIO;

public class ExemploSimples {

    public static void main(String[] args) throws IOException {
        BufferedImage img = ImageIO.read(new File("imagem_original.png"));

        int blue  = 0x000000FF; // Assumindo image.getType() ==
        int green = 0x0000FF00; // BufferedImage.TYPE_3BYTE_BGR
        int red   = 0x00FF0000;

        img.setRGB(0, 0, blue );
        img.setRGB(1, 0, green);
        img.setRGB(2, 0, red  );

        ImageIO.write(img, "png", new File("imagem_modificada.png"));
    }
}
```

Para o caso de imagens em tons de cinza, os três canais recebem o mesmo valor.

As recomendações específicas para este trabalho são:

- Com exceção da classe principal da aplicação, todas as outras classes deverão estar contidas no pacote `fotomosaico`. Crie quantas classes julgar necessário.
- O pacote `fotomosaico` deve conter classes responsáveis pela leitura e manipulação da base de imagens pequenas, bem como leitura e manipulação da imagem que servirá como base para a composição do mosaico.
- A classe principal deve ser chamar `FotoMosaicoApp` e estar no pacote `app`.
- O programa Java deverá esperar quatro argumentos de entrada informados em sua invocação, nessa ordem: (1) o caminho para o arquivo de imagem que será convertido em mosaico; (2) o caminho para a pasta contendo as imagens utilizadas como ladrilhos; (3) o valor de ϵ ; e (4) o caminho para o arquivo de imagem gerado. Por exemplo:

```
java -jar FotoMosaicoApp "C:\entrada.png" "C:\Ladrilhos" 50 "C:\saida.png"
```

Obs. 1: As operações aplicadas sobre o mosaico deverão estar implementadas dentro no pacote `fotomosaico`. Ou seja, não implemente nenhuma operação de processamento ou composição de processamentos dentro no método `main` de sua classe principal.

Obs. 2: O uso adequado de conceitos e princípios de orientação a objetos será avaliado.

Obs. 3: Lance e trate exceções de forma adequada tanto nos métodos implementados dentro das classes do pacote `fotomosaico` (e.g., métodos que esperam uma instância de objeto como argumento, mas que recebem uma referência nula) quanto no método `main` (e.g., argumentos inválidos informados pelo usuário). Além disso, lembre-se que o usuário final do programa é uma pessoa e esta deve ser instruída pelo programa caso algum erro pertinente a ela aconteça (e.g., caminho de arquivo inexistente informado).

Obs. 4: Não é permitida a solicitação de dados ao usuário via entrada padrão ou janelas durante a execução do programa.

Obs. 5: Em <http://www.ic.uff.br/~laffernandes/teaching/2015.1/tcc-00.309/imagens.zip> são disponibilizadas imagens de entrada e um banco de imagens quadradas (50×50 pixels) em tons de cinza. É permitido e recomendado que o aluno utilizar suas próprias imagens, tanto de entrada quanto de ladrilhos.

Bom Trabalho!