CIS2168 006 Fall 2015 Assign1

Hint Summary

Here is the summary of the hints given in the assignment description and in class.

1) Setup your project for using SimpleGUI tool: adding the jar file to your project library
2) Place the image file directly under your java project folder.
3) Create a class that contains two methods, main() and another method, say named computeAvgColor.
    a. import the classes in package simplegui
    b. import the class java.awt.Color
    c. In main():
        - Read the image
            - DrwImage im = new DrwImage("myFirstPet.jpg"))
        - Create a SimpleGUI object that has the same width and height as the image. And display the image.
            - im.getWidth(), im.getHeight()
        - Step through the image and use a nested loop to process each square (box) of g x g pixels (hit each g-th row and colomn).
            a) Test if you get the correct location of all squares (box):
                - For each g x g square (box), draw a black dot at the top-left corner of the square (box).
                    - sg.drawDot(x,y,1)

            b) Pixelize the square(box) with after a) is done
                - For each g x g square (box), draw a box of the same size: g x g and filled with the average color of all the pixels in the square.
                    - sg.drawFilledBox(x,y,width,height,color,transparency, null)

```
public void drawFilledBox(double x,
        double y,
        double width,
        double height,
        java.awt.Color c,
        double transparency, java.lang.String name)
```

- You can first use a fixed color e.g. Color.blue to draw all filled boxes. Then use the actual average color returned by computeAvgColor(…) to fill each box.
- You need to pass to computeAvgColor(…) the location and size of each square, and the reference to the DrwImage object.
- You need to extract the red, green, blue components in the returned RGB object and use them to create a Color object before you can call drawFilledBox with the actual average color.

- **Use a variable g with initial value 16. Do NOT use a constant g.**
- **Make sure that you consider the image widths and heights that are NOT multiples of g**

d. In computeAvgColor(…)

- Compute the average color of the given g x g square (box), returns the result as a RGB object
- The average color is the (average red component, average green component, average blue component) considering all pixels in the given square.
- Use a nested loop to cycle through each pixel in the given square (box).
- To access the color of each pixel in the given square, use something like below
    - RGB rgb = im.getPixelRGB(x, y);
- Depending what you passed to this method about the size of the square (box), you may need to consider the image widths and heights that are NOT multiples of grid size g. So you may have to handle the boxes on the image boundaries in a special way.