AP/ITEC 2610A Summer 2014 Assignment 3

Due at 23:59 Sun 3 Aug 2014

The assignment should be submitted by email to itec2610@gmail.com:

- All source code files should have a header to include your name, student ID, and a brief description of your code in the form of comments.
- Put all source code files (.java, not .class) in a zip file named xxx.zip, where xxx stands for your last name+first name initial. For example, Tom Cruise, the file name should be CruiseT.zip.
- The zip file should include all source files for both tasks including the driver class file or html file for Applet and needed image files. Please in the Subject line of email include the following information: ITEC2610A Assignment3, your name, and your student ID. Make sure your code compiles and runs under JDK of TEL 2116/2027/2032.

Write an applet which displays the dynamics of three circles of different colors moving with slightly different speeds around within the applet frame. Two different sounds are produced when the circles hit the walls or each other. You can find sound resource files in the A2SupportingFiles.zip, please use those files (or at least those file names).

Note: A good way to start this assignment is to use the file Anim2b.java. As in that example you may use the class Applet instead of JApplet.

At the beginning, three circles shoul appear in the applet frame in random positions, and start moving in random directions. Colisions will be handled as follows:

- When a circle reaches the edge of the applet frame, it changes direction, so that is can continue to move within the region. The new direction should be a "reflection" of the prior direction. *Hint: movement of the circle actually means re-drawing the circle with a center point moved. How much the (x,y) coordinates of the center point change by, determine the direction and angle of the perceived motion.*
- Use the same collision algorithm for all 3 circles. *Hint: the dimensions of the applet are given by: appl.size().width and appl.size().height.*
- When two circles collide they have to change direction (any way you want). *Hint: the distance between points (x1, y1) and (x2, y2) is given by the square root of* $(x1-x2)^2 + (y1-y2)^2$

In addition, the user can change the direction of a single circle (say the red circle). In the lower part of the applet build a button labeled "Turn Red Circle". When clicked it changes the direction of movement of the red circle by 45% clockwise.