Assignment 4 – Interactive graphics programs

This program modifies and builds upon the RandomCircles program from the previous assignment. If you did not fully complete the RandomCircles assignment, then it is recommended that you work on that first, at least until you develop it enough to be able to implement the following program.

Modify your RandomCircles program to make a game in which the player attempts to click on all the circles as fast as possible. The game works as follows:

- First, a dialog box appears asking the player how many circles should be used in the game.
- After a number is chosen, that many random circles appear in the window, following the same rules as in the previous assignment. That is, their radii should be random between 5 and 50, their color should be random, and their position should be random, but chosen so that the circle fits entirely in the window.
- The player must click all the circles. Whenever the player clicks a circle, that circle disappears.
- Once no more circles are left on the screen, the player is given a message telling them that they have completed the game. This message can be a dialog window, or it can be a GLabel somewhere in the window.

After you have the basic game working, the next step is to add more features to make it more fun. From here, what you do with the game is up to you. Here are some ideas for additions, but you should feel free to come up with and implement your own ideas:

- At the end of the game, tell the user how long it took them to click all the circles. Recall that the method `System.currentTimeMillis()` is useful for finding how long something took.
- At the end of the game, show a dialog asking if the user would like to play again. If they say yes, start the game again from the beginning.
- In addition to displaying circles on the screen to be clicked, also add a number of squares which should *not* be clicked. If the player clicks on a square, you could penalize their time or make them lose the game immediately. If you make this particular addition, two hints for you:
  - You may find the keyword `instanceof` useful. If `o` is a GObject and you are not sure if it is a GRect or a GOval, you can write `if (o instanceof GRect)`, where the condition holds only if `o` is actually a GRect and not a GOval.
  - Be careful not to draw squares which completely cover some of the circles, as this could make the game unwinnable. One way to avoid this is to draw the squares before drawing the circles, so that the circles will appear on top of the squares.
- Instead of using random colors, use specific colors that mean something. Maybe you can keep score, and give more points for clicking several circles of the same color in a row.
- **Challenge** – Make the circles move in some fashion, so that they are harder to click.

You will need to think carefully about what exactly needs to be stored in instance variables. How will you detect when all circles have been clicked?