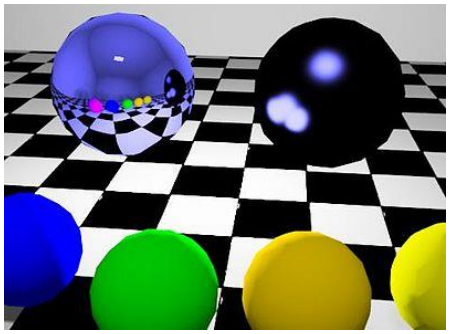

Department of Computer Science & CINSAM

NKU Summer Programming Workshop 2015

Project 5: Graphics

Part 1: Play!

Using the GraphicsSkeleton program as a starting point, see what you can draw using setColor, drawRect, fillRect, drawOval, fillOval and drawLine messages. Make something up and use a variety of colors. Next, draw a "star field" by making the entire background black with fillRect and then use a for loop to iterate 20 times and generate random x, y locations for each star, random values for the size of the stars and random colors and use fillOval to draw them.

Part 2: Adding a Graphics Screen

Use paint or the Java Graphics instructions and draw a splash (intro) screen or ending screen for your fight game. Copy and paste your entire fight game into the GraphicsSkeleton program. Place your fight game at the bottom of main if you want to show a splash screen and then play the game, or place your entire fight game at the top of main so that your ending screen appears at the end.

Part 3: Convert your Hare/Tortoise game into a graphical one.

Use both the FirstAnimation program and your Hare & Tortoise program from Wednesday to turn it into a graphical game as follows:

- Your main method will be the same, but add to it the code needing to generate the JFrame, JPanel and insert the JPanel into the JFrame. Move the variables to denote the location of the hare and tortoise to outside of main (in FirstAnimation, see where x and y are decalred). If you don't have a Scanner in your Hare/Tortoise program, add it. At the end of each turn, add the following two instructions:

  ```
  panel.repaint();
  in.nextLine();
  ```

  Remove all System.out.println statements. Make sure the while loop uses the hare/tortoise condition instead of while(true).
- Set the JFrame size to 400x300 or larger.
- In paintComponent, draw the race course. Use a green background (for grass), draw the hare and tortoise as different colored ovals, or use images (gif, jpg or png files from the Internet). The hare and tortoise are placed at the location given by their variable values. This is their y-coordinate. For their x-coordinate, use 100 for one and 200 for the other. You could draw the hare for instance using:

  ```
  g.fillOval(hare*2+100,150,5,5);
  ```
- Add to your paintComponent method instructions to also draw the other game objects (streams, carrot patch, black holes).
- Add to your paintComponent a nested if-else which tests if hare >= 100 or tortoise >= 100 and if so, tests to see if hare > tortoise and if so, use a drawString to output Hare wins and if tortoise > hare then output Tortoise wins. Only output a message if one of hare or tortoise >= 100.