

**Practical 9: Programming with Graphics2D and Transformations. (Chapter 6). 15, 17 January 2007**

**Can you please take out and open your Chapter 6 notes!**

1. Make sure that you have copied my ch06 into x:\gp1\progs\ and (2.) make sure you are in folder ch06.
  - (a) Compile HouseViewer1.java;
  - (b) Execute HouseViewer1;
  - (c) Make the following changes, one at a time, to HouseViewer1.java:
    - (i) Move its bottom left corner to (x= 50, y= 30); (what needs to be changed in the call to createHouse?
    - (ii) In the program for (i) change width to 50 and wallHt to 150;
    - (iii) In the program for (i) change width to 150 and wallHt to 50.
  
3.
  - (a) Compile HouseViewer2.java;
  - (b) Execute HouseViewer2;
  - (c) Make the following changes, one at a time, to HouseViewer2.java:

In addition to the shiftedHouse at (150, 50), I want to see:

    - (i) shiftedHouse1 at (120, 20), colour `g2.setPaint(Color.red);`
    - (ii) shiftedHouse2 at (10, 160), colour `g2.setPaint(Color.green);`
    - (iii) shiftedHouse3 at (120, 150), colour `g2.setPaint(Color.blue);`

To do (i), (ii), (iii), copy and modify the following code:

```
AffineTransform at1 = new AffineTransform(); at1.translate(150, 50);
Shape shiftedHouse = at1.createTransformedShape(house);
g2.setPaint(Color.red); g2.draw(shiftedHouse);
```

That is, after you have done (iii) there will be three *new* versions of the code above; you will have to include `at1= new AffineTransform();` each time to *clear* the transform; normally, `AffineTransforms` are cumulative.
  
4.
  - (a) Compile HouseRotViewer.java;
  - (b) Execute HouseRotViewer;
  - (c) Make the following changes, one at a time, to HouseRotViewer.java:

In addition to the rotated house at the origin (0, 0), I want to see: to see (keep the colour black, but change it if you like):

    - (i) rotatedHouse1 rotated by 60°.
    - (ii) rotatedHouse2 rotated by 90°;
    - (iii) rotatedHouse3 rotated by 120°;
    - (iv) rotatedHouse4 rotated by 150°;
    - (v) rotatedHouse5 rotated by 180°;
    - (vi) rotatedHouse6 rotated by 270°;
    - (vii) rotatedHouse7 rotated by 315°;
    - (viii) rotatedHouse8 rotated by 358°; paint this in red;
    - (ix) rotatedHouse9 rotated by 360°; paint this in green.

To do above, **copy and modify** the following code: (X = 1, 2, 3 ... 9)

```
at1 = new AffineTransform();
at1.rotate(Math.toRadians(30.0));
Shape rotatedHouseX = at1.createTransformedShape(house);
...
g2.setStroke(stroke);
g2.draw(rotatedHouseX);
```

That is, after you have done (ix) there will be nine *new* versions of the code above; you will have to include `at1= new AffineTransform()`; each time to *clear* the transform; normally, `AffineTransforms` are cumulative.

5. (a) Compile `HouseScaleViewer.java`;
- (b) Execute `HouseScaleViewer`;
- (c) Make the following changes, one at a time, to `HouseScaleViewer.java`:

In addition to the `shiftedHouse` at (150, 50), I want to see:

- (i) `scaledHouse11` scaled by `at1.scale(0.25, 0.5)`; `g2.setPaint(Color.red)`;
- (ii) `scaledHouse12` scaled by `at1.scale(1.0, 3.0)`; `g2.setPaint(Color.green)`;
- (iii) `scaledHouse12` scaled by `at1.scale(1.0, -1.0)`; `g2.setPaint(Color.blue)`;

To do (i), (ii), (iii), copy and modify the following code:

```
at1 = new AffineTransform();
at1.scale(0.5, 2.0);
Shape scaledHouse2 = at1.createTransformedShape(shiftedHouse);
...
g2.setStroke(stroke);    g2.draw(scaledHouse2);
```

That is, after you have done (iii) there will be three *new* versions of the code above; you will have to include `at1= new AffineTransform()`; each time to *clear* the transform; normally, `AffineTransforms` are cumulative.

- (d) What rotation is `at1.scale(1.0, -1.0)`; equivalent to? make sure you can answer this. Hint, hint, May examination.