

## Java - Images

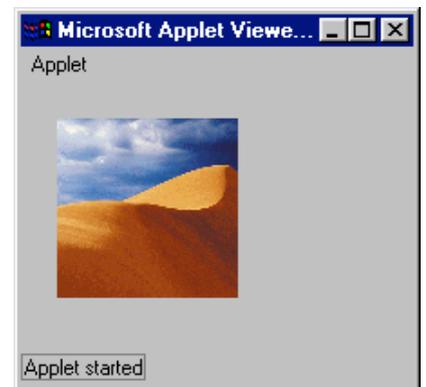
C&G criteria: 5.2.2

Java cannot cope with all image formats, notably it will not handle bmp files. It will support jpeg and gif files and these can be loaded on the form or applet that Java is running on. Java needs to know where the picture is and its file name, these details are assigned to an Image object with `getImage()`. Java finds the path of an image through a URL. Rather than using the URL of an image the image can be placed in the same directory as the Java program itself and `getCodeBase()` will sort out the URL.

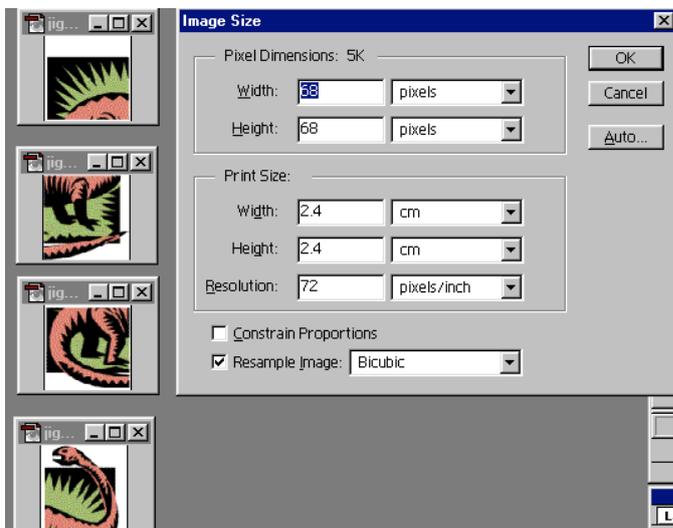
Create a suitable gif in Photoshop and adapt this code to display it. The gif will have to be saved to the same folder as the Java program.

```
import java.awt.*;
import java.applet.Applet;

public class ShowPict extends Applet
{
    public Image pict; //create an Image object
    public void init()
    {
        //use getCodeBase() to direct Java to the current path
        //getImage() takes the URL and assigns it to the Image object
        pict = getImage(getCodeBase(),"Dune.gif");
    }
    public void paint(Graphics g)
    {
        //display the image in a box 20, 20, 100, 100
        boolean b = g.drawImage(pict,20,20,100,100,this);
    }
}
```

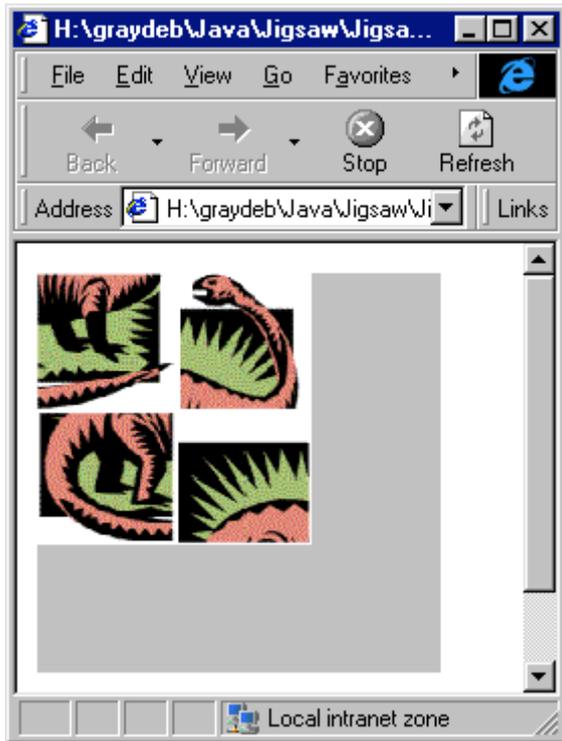


### The jigsaw

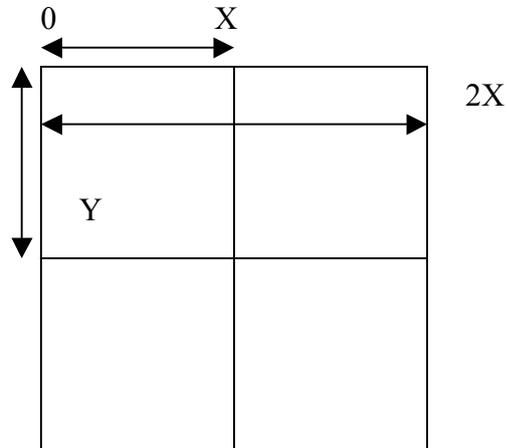


This program requires more work in Photoshop to set up the gifs than to write the code. The example uses a single image that has been cut into 4 images of equal size, 68 pixels square. The exact size of the images is not important but they must all be the same size.

When the program runs the images are mixed up. They can be put into the correct big picture using the mouse.



The images are part of an array. The elements 0 to 3 are placed on the Applet with drawImage() based on knowing how wide each image is.



This code outline handles the 4 images. These have been named jig0 to jig4 to enable a loop to load the images in turn. The images are 68 pixels square.

```
import java.applet.*;
import java.awt.*;
import java.awt.image.*;
import java.awt.event.*;
//requires a set of gifs of roughly the same size
//this version uses Java 2
public class Jigsaw2 extends Applet implements MouseListener
{
    //array of images
    Image[] jigsaw = new Image[4];
    int pick,drop; //position in array
    Image temp;
    public void init()
    {
        this.addMouseListener(this);
        //load gifs into array
        for(int count=0;count<4;count++)
        {
            //these are now in order jigsaw[0] is jig0.gif etc.
            jigsaw[count]=getImage(getCodeBase(),"jig"+ (count + 1) + ".gif");
        }
        //swap the array
        for(int count=0;count<4;count++)
        {
            int rand;
            rand=(int)(java.lang.Math.random()*4);
            temp=jigsaw[rand];
```

```

        jigsaw[rand]=jigsaw[count];
        jigsaw[count]=temp;
    }
}

public void mouseClicked(MouseEvent e)
{
    } //can use instead of MousePressed
public void mouseReleased(MouseEvent e)
{
    //convert x y to image array no' x= 0 to x = 68 becomes 0 with (int)
    drop=(int)(e.getX()/68)+(int)(e.getY()/68)*2;
    //swap this with pick
    temp=jigsaw[drop];
    jigsaw[drop]=jigsaw[pick];
    jigsaw[pick]=temp;
    repaint();
}
public void mousePressed(MouseEvent e)
{
    //sizes are based on gif sizes - *2 due to 2x2 jigsaw
    pick=(int)(e.getX()/68)+(int)(e.getY()/68)*2;
}
public void mouseExited(MouseEvent e)
{
}
public void mouseEntered(MouseEvent e)
{
}
public void paint(Graphics g)
{
    int xnow[] = new int[4];
    int ynow[] = new int[4];
    int count;
    //set up pieces in grid - based on gif size
    xnow[0]=0;ynow[0]=0;
    xnow[1]=68;ynow[1]=0;
    xnow[2]=0;ynow[2]=68;
    xnow[3]=68;ynow[3]=68;
    for(count=0;count<4;count++)
    {
        //this tiles images
        g.drawImage(jigsaw[count],xnow[count],ynow[count],this);
    }
}
}

```

## Modifications

Change the code to handle a 3 x 3 image of your own creation. The Image array will be of size 8 (for 9 elements). The code will be a lot easier if the images are all square. Photoshop can be used to cut up a larger image to 9 pieces. The code will still run if the images do not exactly fit together.