

Myro

Class MyroImage

java.lang.Object

Myro.MyroImage

All Implemented Interfaces:

java.lang.Iterable<Myro.MyroPixel>

```
public abstract class MyroImage extends java.lang.Object implements  
java.lang.Iterable<Myro.MyroPixel>
```

This class represents an image in the Myro/Java environment. Methods permit simple manipulation of an image, as well as methods to display and hide an image. This is an abstract class and therefore cannot be instantiated; classes MyroGrayImage and MyroColorImage are derived from MyroImage and can be instantiated.

Version:

1.0 - September 2010

Author:

Douglas Harms

Method Summary

Myro.MyroBlobSpec	defineBlob (int xlow, int ylow, int width, int height) Calculate a blob based on a rectangular area of the image.
abstract java.awt.Color	getColor (int x, int y) Returns the RGB color of pixel (x,y).
abstract int	getGray (int x, int y) Returns the grayscale value of pixel (x,y).
int	getHeight () Returns the height of the image.
java.awt.image.BufferedImage	getImage () Returns the BufferedImage defined in this MyroImage.
static Myro.MyroFrame	getMyroFrame (java.lang.String frameName) Returns the MyroFrame that has a specified name.
Myro.MyroPixel	getPixel (int x, int y) Returns a MyroPixel instance that can be used to examine and modify a pixel in this image.
int	getType () Returns the type (i.e., Color or Grayscale) of this image.
Myro.MyroBlobSpec	getUserDefinedBlob () Allows the user to select a rectangular area of the image used to define a blob.
int	getWidth () Returns the width of the image.
void	hide () Causes this image to be invisible (i.e., the window disappears from the screen).

java.util.Iterator<Myro.MyroPixel>		<u>iterator</u> () Returns an iterator that can be used to iterate over the pixels in this image.
abstract	boolean	<u>loadImage</u> (java.lang.String filename) Load an image from a file.
	void	<u>repaint</u> () Changes made to the image (e.g., by calling <code>set</code>) are reflected in the displayed window.
	void	<u>resize</u> (int newWidth, int newHeight) resizes the image to be width x height
	void	<u>save</u> (java.lang.String filename) Save the image to a file.
abstract	void	<u>setColor</u> (int x, int y, java.awt.Color color) Sets the color of pixel (x,y) to an RGB color.
abstract	void	<u>setGray</u> (int x, int y, int grayLevel) Sets the color of pixel (x,y) to a grayscale color.
	void	<u>show</u> () Causes this image to be visible in a window.
	void	<u>show</u> (int x, int y, java.lang.String title) Causes this image to be visible in a window located at a specified location on the screen.
	void	<u>show</u> (java.lang.String title) Causes this image to be visible in a window.

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Method Detail

defineBlob

```
public Myro.MyroBlobSpec defineBlob(int xlow,
                                     int ylow,
                                     int width,
                                     int height)
```

Calculate a blob based on a rectangular area of the image. The specified rectangle must be non-empty and be completely within the image. The returned blob contains the average color of the rectangular area of the image and can be passed to the scribbler's `configureBlob` method.

Returns:

A blob that can be passed to a scribbler `configureBlob` method.

Precondition:

(`xlow,ylow`) is the coordinate of the upperleft corner or a rectangle within the image (i.e., ≥ 0), `width` and `height` are both > 0 , and the lowerright corner of the rectangle is within the image.

getColor

```
public abstract java.awt.Color getColor(int x,
```

int y)

Returns the RGB color of pixel (x,y).

Parameters:

x - x coordinate of the pixel

y - y coordinate of the pixel

Returns:

The color of pixel (x,y)

Precondition:

(x,y) is the coordinate of a pixel in the image.

getGray

```
public abstract int getGray(int x,  
                             int y)
```

Returns the grayscale value of pixel (x,y).

Parameters:

x - x coordinate of the pixel

y - y coordinate of the pixel

Returns:

The grayscale value of pixel (x,y), in the range 0..255

Precondition:

(x,y) is the coordinate of a pixel in the image.

getHeight

```
public int getHeight()
```

Returns the height of the image.

getImage

```
public java.awt.image.BufferedImage getImage()
```

Returns the BufferedImage defined in this MyroImage.

getMyroFrame

```
public static Myro.MyroFrame getMyroFrame(java.lang.String frameName)
```

Returns the MyroFrame that has a specified name.

Parameters:

frameName - Name of a MyroFrame

Returns:

the MyroFrame with the associated name, or null if no frame has that name

getPixel

```
public Myro.MyroPixel getPixel(int x,  
                                 int y)
```

Returns a MyroPixel instance that can be used to examine and modify a pixel in this image.

Parameters:

x - x coordinate of the pixel

y - y coordinate of the pixel

Returns:

The MyroPixel that references pixel (x,y)

Precondition:

(x,y) is the coordinate of a pixel in the image

getType

```
public int getType()
```

Returns the type (i.e., Color or Grayscale) of this image.

Returns:

Value Scribbler.IMAGE_COLOR or Scribbler.IMAGE_GRAY.

getUserDefinedBlob

```
public Myro.MyroBlobSpec getUserDefinedBlob()
```

Allows the user to select a rectangular area of the image used to define a blob. The image is first made visible, then a message at the bottom of the window instructs the user to drag an area to define a blob. The blob returned contains the average color of the selected area and can be passed to the scribbler configureBlob method.

Returns:

null if the user selected a zero-size rectangle; otherwise a blob that can be passed to a scribbler configureBlob method.

getWidth

```
public int getWidth()
```

Returns the width of the image.

hide

```
public void hide()
```

Causes this image to be invisible (i.e., the window disappears from the screen). If this image was already invisible, no changes are made to any window.

iterator

```
public java.util.Iterator<Myro.MyroPixel> iterator()
```

Returns an iterator that can be used to iterate over the pixels in this image.

Specified by:

iterator in interface java.lang.Iterable<Myro.MyroPixel>

loadImage

```
public abstract boolean loadImage(java.lang.String filename)
```

Load an image from a file.

Parameters:

filename - Name of a file containing an image.

Returns:

true returned if the image was successfully loaded, false returned if an error occurred

repaint

```
public void repaint()
```

Changes made to the image (e.g., by calling `set`) are reflected in the displayed window. If the window is invisible no changes are made to any window.

resize

```
public void resize(int newWidth,  
                  int newHeight)
```

resizes the image to be width x height

save

```
public void save(java.lang.String filename)
```

Save the image to a file. The only supported formats are jpg and png.

Parameters:

filename - Name of the file to save the image to.

Precondition:

filename must have an extension of .jpg or .png

setColor

```
public abstract void setColor(int x,  
                               int y,  
                               java.awt.Color color)
```

Sets the color of pixel (x,y) to an RGB color. Changes will not appear on the screen until either [show](#) or [repaint](#) is called.

Parameters:

x - x coordinate of the pixel

y - y coordinate of the pixel

color - An RGB Color that pixel (x,y) is set to.

Precondition:

(x,y) is the coordinate of a pixel in the image, color is not null.

setGray

```
public abstract void setGray(int x,  
                              int y,  
                              int grayLevel)
```

Sets the color of pixel (x,y) to a grayscale color.

Parameters:

x - x coordinate of the pixel

y - y coordinate of the pixel

grayLevel - A grayscale value between 0 and 255 that pixel (*x,y*) is set to.

Precondition:

(*x,y*) is the coordinate of a pixel in the image, *grayLevel* is between 0 (inclusive) and 255 (inclusive).

Changes will not appear on the screen until either [show](#) or [repaint](#) is called.

show

```
public void show()
```

Causes this image to be visible in a window. If the image had not been displayed previously a window will be created at the upperleft corner of the screen. The title of the window will be "Myro".

show

```
public void show(int x,  
                 int y,  
                 java.lang.String title)
```

Causes this image to be visible in a window located at a specified location on the screen. The parameters specify the location of the upperleft corner of the window and the title of the window. If the window already exists, the location parameters are ignored.

Parameters:

x - x coordinate of the upperleft corner of the window

y - y coordinate of the upperleft corner of the window

title - Title of the window

show

```
public void show(java.lang.String title)
```

Causes this image to be visible in a window. If the image had not been displayed previously a window will be created at the upperleft corner of the screen. The title of the window is passed as a parameter.

Parameters:

title - Title of the window