Rotating images in Java ME

This code snippet demonstrates how to rotate images in Java ME applications.

Overview

To rotate an image to a specified angle, a new image must be created by calling `Image.createImage(Image sourceImage, int x, int y, int width, int height, int transform)` method with source image passed as the `sourceImage` parameter and angle value passed as the `transform` parameter.

Some possible values of `transform` parameter are:

1. Sprite.TRANS_NONE - causes the specified image region to be copied unchanged
2. Sprite.TRANS_ROT90 - causes the specified image region to be rotated clockwise by 90 degrees.
3. Sprite.TRANS_ROT180 - causes the specified image region to be rotated clockwise by 180 degrees.
4. Sprite.TRANS_ROT270 - causes the specified image region to be rotated clockwise by 270 degrees.

Source file: ImageRotating.java

```
import javax.microedition.midlet.MIDlet;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Image;
import javax.microedition.lcdui.game.Sprite;

import java.io.IOException;


public class ImageRotating extends MIDlet implements CommandListener {

    private Display display;
    // Form where image is placed
    private Form form;

    // Command for rotating source image.
    private Command cmdRotate0;
    private Command cmdRotate90;
    private Command cmdRotate180;
    private Command cmdRotate270;
    // Command for exiting from midlet.
    private Command cmdExit;

    private Image sourceImage;
    private Image destImage;

    /**
     * Constructor
     */
    public ImageRotating() {
        initializeComponents();

        form.append(sourceImage);
```


(C) Copyright Nokia 2014. All rights reserved.

```
    }

    /**
     * Initializes components of midlet
     */
    private void initializeComponents() {
        // Get display
        display = Display.getDisplay(this);

        // Load image
        try {
            sourceImage = Image.createImage("/image.jpg");
        } catch (IOException e) {
            exitMIDlet();
        }

        // Create form
        form = new Form("Image rotating");

        cmdRotate0 = new Command("Rotate 0", Command.SCREEN, 0);
        form.addCommand(cmdRotate0);
        cmdRotate90 = new Command("Rotate 90", Command.SCREEN, 0);
        form.addCommand(cmdRotate90);
        cmdRotate180 = new Command("Rotate 180", Command.SCREEN, 0);
        form.addCommand(cmdRotate180);
        cmdRotate270 = new Command("Rotate 270", Command.SCREEN, 0);
        form.addCommand(cmdRotate270);
        cmdExit = new Command("Exit", Command.EXIT, 0);
        form.addCommand(cmdExit);

        form.setCommandListener(this);
    }

    /**
     * Rotates image and display it in the form.
     * @param rotateType - angle of rotation.
     */
    private void rotateImage(int transform) {
        destImage = Image.createImage(sourceImage, 0, 0,
                sourceImage.getWidth(), sourceImage.getHeight(), transform);
        form.deleteAll();
        form.append(destImage);
    }

    /**
     * From MIDlet.
     * Signals the MIDlet that it has entered the Active state.
     */
    public void startApp() {
        display.setCurrent(form);
    }

    /**
     * From MIDlet.
     * Signals the MIDlet to enter the Paused state.
     */
    public void pauseApp() {
```


(C) Copyright Nokia 2014. All rights reserved.

```
        // No implementation required
    }

    /**
     * From MIDlet.
     * Signals the MIDlet to terminate and enter the Destroyed state.
     */
    public void destroyApp(boolean unconditional) {
        // No implementation required
    }

    /**
     * Performs exit from midlet.
     */
    private void exitMIDlet() {
        notifyDestroyed();
    }

    /**
     * From CommandListener.
     * Indicates that a command event has occurred on Displayable d.
     * @param cmd - a Command object identifying the command.
     * @param d - the Displayable on which this event has occurred.
     */
    public void commandAction(Command cmd, Displayable d) {
        if(cmd == cmdRotate0) {
            rotateImage(Sprite.TRANS_NONE);
        } else if(cmd == cmdRotate90) {
            rotateImage(Sprite.TRANS_ROT90);
        } else if(cmd == cmdRotate180) {
            rotateImage(Sprite.TRANS_ROT180);
        } else if(cmd == cmdRotate270) {
            rotateImage(Sprite.TRANS_ROT270);
        } else if(cmd == cmdExit) {
            exitMIDlet();
        }
    }
}
```

Postconditions

After loading the snippet image, now rotated, is shown on display.

By selecting the menu commands "Rotate 0" - "Rotate 270" the user can rotate the source image to a specific angle. After this, the rotated image will be shown on display.

Supplementary material

Executables and source files can be found here: RotatingImages.zip

(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.