Image at different positions

package hello;

import java.io.IOException;

import javax.microedition.midlet.\*;

import javax.microedition.lcdui.\*;

public class HelloMIDlet extends MIDlet{

private Command exitCommand; // The exit command

public Display display; // The display for this MIDlet

public Image image;

public HelloMIDlet() {

display = Display.getDisplay(this);

exitCommand = new Command("Exit", Command.EXIT, 0);

}

public void startApp() {

Canvas obj = new DrawImage();

display = Display.getDisplay(this);

display.setCurrent(obj);

}

public void pauseApp() {

}

public void destroyApp(boolean unconditional) {

}

public void commandAction(Command c, Displayable s) {

if (c == exitCommand) {

destroyApp(false);

notifyDestroyed();

}

}

public class DrawImage extends Canvas{

int width = getWidth();

int height = getHeight();

protected void paint(Graphics g) {

//System.out.println("111111");

try{

image = Image.createImage("images/kola.png");

}

catch (IOException ex) {

throw new RuntimeException ("Unable to load Image: "+ex);

}

g.setGrayScale (255);

g.fillRect (0, 0, getWidth (), getHeight ());

g.drawImage (image, 0, 0, Graphics.TOP | Graphics.LEFT);

g.drawImage (image, getWidth () / 2, getHeight () / 2,

Graphics.HCENTER | Graphics.VCENTER);

g.drawImage (image, getWidth (), getHeight (),

Graphics.BOTTOM | Graphics.RIGHT);

}

}

}

OUTPUT:

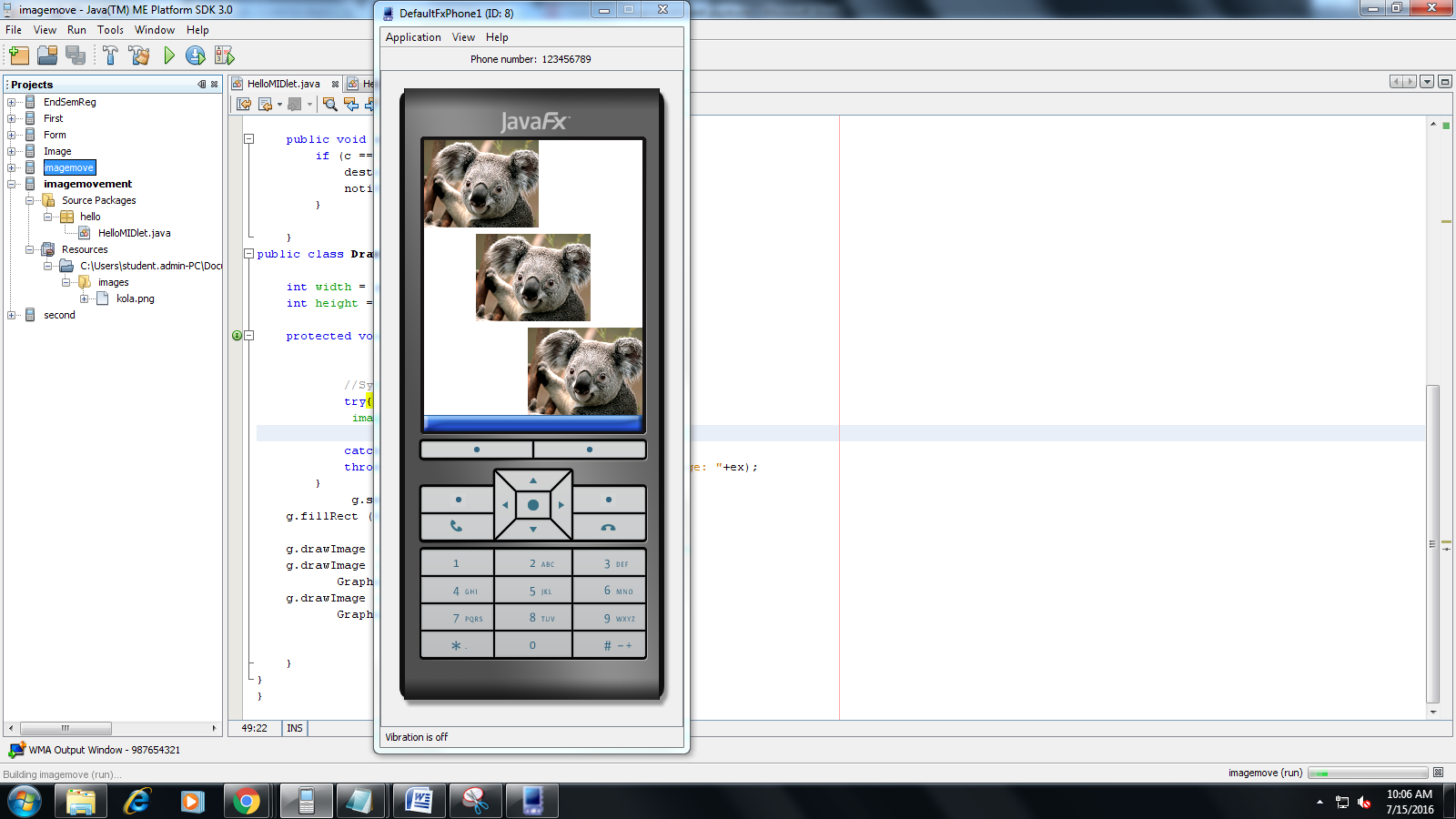


IMAGE moving on keypress:

package hello;

import javax.microedition.midlet.\*;

import javax.microedition.lcdui.\*;

public class HelloMIDlet extends MIDlet implements CommandListener {

private Command exitCommand; // The exit command

public Display display; // The display for this MIDlet

public Image image;

public Displayable nd;

public HelloMIDlet() {

try {

image = Image.createImage("images/kola.png");

} catch (Exception e) {

}

display = Display.getDisplay(this);

nd=new navigate(image);

exitCommand = new Command("Exit", Command.EXIT, 0);

nd.addCommand(exitCommand);

nd.setCommandListener(this);

display.setCurrent(nd);

}

public void startApp() {

}

public void pauseApp() {

}

public void destroyApp(boolean unconditional) {

}

public void commandAction(Command c, Displayable s) {

if (c == exitCommand) {

destroyApp(false);

notifyDestroyed();

}

}

}

class navigate extends Canvas {

private Image image;

private int newX = 0;

private int newY = 0;

private int stepX = 0;

private int stepY = 0;

public navigate(Image image) {

this.image = image;

newX = 0;

newY = 0;

stepX = getWidth() / 4;

stepY = getHeight() / 4;

}

public void steppingXY(int x, int y) {

newX += x;

newY += y;

}

public void paint(Graphics g) {

int width = this.getWidth();

int height = this.getHeight();

g.setGrayScale(255);

g.fillRect(0, 0, width - 1, height - 1);

g.setGrayScale(0);

g.drawRect(0, 0, width - 1, height - 1);

g.translate(newX, newY);

g.drawImage(image, 0, 0, g.TOP | g.LEFT);

}

protected void keyPressed(int keyCode) {

int gameaction = getGameAction(keyCode);

switch (gameaction) {

case UP:

steppingXY(0, stepY);

break;

case DOWN:

steppingXY(0, -stepY);

break;

case LEFT:

steppingXY(stepX, 0);

break;

case RIGHT:

steppingXY(-stepX, 0);

break;

}

repaint();

}

}

OUTPUT:

