Parallax Scrolling in Java ME

This article illustrates how to use parallex scrolling in Java ME.

Introduction

Parallax Scrolling is something that I use in my game. I decided to release the code because I thought it could be useful to somebody.

This article also demonstrates how to:

- pause/unpause an app (RIGHT SOFT KEY / incoming phone call)
- get high frames por second (fps)
- get randomNumbers / randomBoolean
- use custom fonts (bitmap fonts)
- use LayerManager/TiledLayer.

Go to my projects and grab JavaMEParallaxScrolling project (with all resources and full source code). You can easily import it in NetBeans 6.9

Code Snippet

```
import java.io.IOException;
import javax.microedition.lcdui.Display;
import javax.microedition.midlet.MIDlet;
import javax.microedition.midlet.MIDletStateChangeException;

public class Main extends MIDlet {

  private ParallaxScrollingCanvas canvas;
  private Display display;

  public Main() { }

  public void startApp() {
    if (canvas==null) canvas = new ParallaxScrollingCanvas();
    display = Display.getDisplay(this);

    try {
      canvas.init();
      canvas.start();
    }
    catch (IOException e) { }
    display.setCurrent(canvas);
  }

  public void pauseApp() { }

  public void destroyApp(boolean destroy) throws MIDletStateChangeException {
    display = null;
    if (canvas!=null) canvas.stop();
    System.gc();
    notifyDestroyed();
  }
}
```

```
/*
 * This class demonstrate how to use LayerManager/TiledLayer.
 *  by George Roberto Peres
 *  reflexus@ig.com.br
 */

import java.io.IOException;
import javax.microedition.lcdui.Image;
import javax.microedition.lcdui.game.LayerManager;
import javax.microedition.lcdui.game.TiledLayer;

/**
 *
 * @author George Roberto Peres
 */
public final class Design {

  // Tiled Layers
  private Image back, front, solo;
  protected TiledLayer layerBack, layerFront, layerSolo;

  // font
  private Image imgFont;   // font image font.png
  protected int charLarg, charAlt;

  protected Design() {
    try {
      layerBack = defineCenario(layerBack, getBack(), 4);
      layerFront = defineCenario(layerFront, getFront(), 4);
      layerSolo = defineCenario(layerSolo, getSolo(), 2);
    }
    catch (IOException ex) { }
  }

  // the order is important. The last added represents the most deep layer
  protected void updateLayerManager(LayerManager lm) throws IOException {
    lm.append(layerSolo);
    lm.append(layerFront);
    lm.append(layerBack);
  }

  protected TiledLayer defineCenario(TiledLayer layer, Image image, int rows) throws IOException {
```


(C) Copyright Nokia 2013. All rights reserved.

```
    short c=1, l=0;
    layer = new TiledLayer(32, rows, image, 16, 16);
    for (byte lin=0; lin<rows; lin++) {
      for (byte col=0; col<32; col++) {
        layer.setCell(col, lin, c);
        c++;
        if ((c-1)%(l+16)==0) c = (byte)(l+1);
      }
      l+=16;
      c+=16;
    }
    return layer;
  }

  private Image getBack() throws IOException {
    if (back == null) {
      back = Image.createImage("/back.png");
    }
    return back;
  }

  private Image getFront() throws IOException {
    if (front == null) {
      front = Image.createImage("/front.png");
    }
    return front;
  }

  private Image getSolo() throws IOException {
    if (solo == null) {
      solo = Image.createImage("/solo.png");
    }
    return solo;
  }

  protected Image getFont() {
    if (imgFont == null)
      try {
        imgFont = Image.createImage("/fonts.png");
        charLarg = imgFont.getWidth() / 96;   //9
        charAlt = imgFont.getHeight();   //12
      }
      catch (IOException e) { }
    return imgFont;
  }
}
```

```
/*
 * This class demonstrate how to get randomNumber/randBoolean.
 * brought to you by George Roberto Peres
 *                    reflexus@ig.com.br
 */

import java.util.Random;
import javax.microedition.lcdui.Graphics;

public final class Tools {

  private static final Random RAND = new Random();

  protected static final int GRAPHICS_TOP_LEFT = Graphics.LEFT | Graphics.TOP;

  // Screen size  Game Canvas
  protected static final int LARG = 240;// screen width (x)
  protected static final int ALT = 320; // screen height (y)
  protected static final int Y = 200;   // the height of the main canvas game
  protected static final boolean TELA_CHEIA = true; // fullscreen

  // each manufacture have its own softkey code
  // -6 and -7 are the correct values for the Nokia Devices
  protected static final int LEFT_SOFTKEY_CODE = -6;
  protected static final int RIGHT_SOFTKEY_CODE = -7;

  /*****************************************************************************
   * Random number generator                                                   *
   * @param max the generated number is greater or equals to zero or less than max *
   * @return  r, a pseudo-random number                                        *
   *****************************************************************************/
  protected static int randomNumber(int max) {
    return Math.abs(RAND.nextInt() % max);   // abs value from -(max-1) to max-1
  }

  /*************************************************************************
   * boolean random number generator                                       *
   * @return boolean                                                       *
   *************************************************************************/
  protected static boolean randBoolean() {
    return (randomNumber(2)==1);
  }
}
```

```
/*
 * This is a simple ParallaxScrolling example.
 * It also demonstrate how to: pause/unpause an app (RIGHT SOFT KEY / phone call);
 * use custom fonts; get high frames por second (fps): smaller painted screen area,
 * no threads (only canvas thread, of course), no timers, no synchronized.
 * The code presented here is easy to understand (clean code).
 * brought to you by George Roberto Peres
 *                    reflexus@ig.com.br
 */

import java.io.IOException;
import javax.microedition.lcdui.Graphics;
import javax.microedition.lcdui.game.GameCanvas;
import javax.microedition.lcdui.game.LayerManager;

/**
 * @author George Roberto Peres
 * @version 1.0
 */

public class ParallaxScrollingCanvas extends GameCanvas implements Runnable {

  private final Graphics g = getGraphics();
  private LayerManager lm;
  private Thread t;
```


(C) Copyright Nokia 2013. All rights reserved.

```
    private Design design;
    private static final byte[] PAUSE = { 80, 65, 85, 83, 69 };
    private static final byte[] FPS = { 70, 80, 83, 58 };
    private boolean pausa;
    private byte cont;   // timing
    private int a;   // general helper
    private boolean isRunning;
    private boolean showFPS = true;
    // frames per second shown on the screen
    private int fps;
    private int cyclesThisSecond;
    private long lastFPSTime;

    // overwrite the javax.microedition.lcdui.Canvas method
    public void keyPressed(int keyCode) {
      if (keyCode==Tools.RIGHT_SOFTKEY_CODE) {
        if (!pausa) pausa(); else resume();
      }
      else
      if (keyCode==Tools.LEFT_SOFTKEY_CODE) {
        if (pausa) resume();
      }
      else
      if (keyCode == this.KEY_NUM0) {
        showFPS = !showFPS;
      }
    }

    /** Creates a new instance of theCanvas */
    public ParallaxScrollingCanvas() {
      super(true);
      setFullScreenMode(Tools.TELA_CHEIA);
      design = new Design();
      g.setColor(0,0,0);
      g.fillRect(0, 0, getWidth(), getHeight());
    }

    protected void init() throws IOException {
      // layer manager
      lm = new LayerManager();
      design.updateLayerManager(lm);
      design.layerBack.setPosition(0, 56);
      design.layerFront.setPosition(0, 80);
      design.layerSolo.setPosition(0, 140);
    }

    public void start() {
      isRunning = true;
      t = new Thread(this);
      t.start();
    }

    public void stop() {
      isRunning = false;
      t = null;
    }

    private void tarefas() {
      // background scene. slower move
      if (cont%5==0) {
        design.layerBack.move(-1, 0);
        // pack to repeat the same graphics again
        if (Math.abs(design.layerBack.getX())>255) design.layerBack.setPosition(0, 56);
      }

      // foreground scene. fastest move
      if (cont%3==0) {
        design.layerFront.move(-1, 0);
        if (Math.abs(design.layerFront.getX())>255) design.layerFront.setPosition(0, 80);
      }

      // moves the layerSolo
      design.layerSolo.move(-1, 0);
      if (Math.abs(design.layerSolo.getX())>255) design.layerSolo.setPosition(0, 140);
    }

    private void renderiza() {
      // background
      g.setColor(0, 0, 0);
      g.fillRect(0, 0, Tools.LARG, Tools.Y);
      lm.paint(g, 0, 0);   // draws. layer manager
      if (showFPS) {
        drawString(FPS, 0, 0);
        drawScoreInt(fps, 2, design.charLarg*4, 0);
      }
      drawScoreInt(Math.abs(design.layerSolo.getX()), 3, 120, 0);
      flushGraphics(0, 0, Tools.LARG, Tools.Y);   // unload graphics
    }

    // draws a character based in the ASCII code table
    // helpful to add customized special characters
    private void drawCharASCII(int ASCII, int x, int y) {
      // the space character and 'not printable' or with ASCII code  > 255 are drawn
      if (ASCII < 33  ||  ASCII > 255) return;
      g.clipRect(x, y, design.charLarg, design.charAlt); // change the clip area to character size
      // draw the character inside the rectangle. (cIndex - 32) * charLarg = the character position
      g.drawImage(design.getFont(), x - ((ASCII - 32) * design.charLarg), y, Tools.GRAPHICS_TOP_LEFT);
      g.setClip(0, 0, Tools.LARG, Tools.ALT);  // reset the clip to back to fullscreen (normal clip size)
    }

    private void drawString(byte []s, int x, int y) {
      int cx = x; // start position

      // loop among all the string character
      for (a = 0; a < s.length; a++) {
        drawCharASCII(s[a], cx, y);
        cx += design.charLarg; // go to the next position to draw
      }
    }


    // draw the score int at x,y (ASCII de 48~57).  ex: 000777 (6 digits)
    private void drawScoreInt(int score, int digitos, int x, int y) {
      if (digitos<1 || digitos>8) return;
      int d = 1;
      for (a=1; a<digitos; a++)  d *= 10;
      a = x;   // start position
      for (int i=0; i<digitos; i++) {
        drawCharASCII(score/d + 48, a, y);  // +48 to match the ASCII table
        a += design.charLarg;
```


(C) Copyright Nokia 2013. All rights reserved.

```
      score -= d*(score/d);
      d /= 10;
    }
  }

  public void hideNotify() {
    if (!pausa) pausa();
  }

  public void showNotify() { }

  protected void pausa() {
    pausa = true;
    stop();
    drawString(PAUSE, Tools.LARG/2 - (design.charLarg*5)/2, Tools.Y/2 - design.charAlt/2);
    flushGraphics(Tools.LARG/2 - (design.charLarg*5)/2, Tools.Y/2 - design.charAlt/2,
            design.charLarg*5, design.charAlt);
  }

  protected void resume() {
    start();
    pausa = false;
  }

  public void run() {
    while(isRunning) {
      // calculate the FPS
      if (System.currentTimeMillis() - lastFPSTime > 1000) {
        lastFPSTime = System.currentTimeMillis();
        fps = cyclesThisSecond;
        cyclesThisSecond = 0;
      } else
      cyclesThisSecond++;
      // timing base 60. why? it divides by 1,2,3,4,5,6,10,12,15,20,30
      cont++;
      if (cont==60) cont=0;
      tarefas();
      renderiza();
      // sleep a little
      try { Thread.sleep(0); }
      catch (InterruptedException e) { }
    }
  }
}
```


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.