How to use custom fonts in JavaME

This article demonstrates how to use custom font (bitmap fonts) in Java ME.

Introduction

This is my first post and I would start with something that everybody wants to know. Most people (including me) get stuck here and that's why i decided to write this article.

Bitmap fonts solves portability issues in your app or game as different phones sometimes have different font sizes and can really screw up app/game design if you are relying only on the native drawString() method (which is also slow). Using bitmap fonts ensures that the text will be rendered correctly on different devices.

You can use the fonts.png from one of my projects: Queops, LunarPatrol/LunarPatrol v0.09 & X-Rally.

```
/*
 * Fonte customizada utilizando-se de arquivo de imagem fonts.png (monospace).
 * Esta versao esta customizada especialmente para J2ME, pois nao utiliza Strings.
 *
 * Custom fonts using image.png (monospace). This is a custom version (fast),
 * cause it does not use any String native method (it means no object creation!).
 * You can also move those 3 methods directly into your Canvas class
 * (of course move the vars too).
 *
 * usage ex:
 * private Fonte fonte;  // holds the fonte
 * fonte = new Fonte();  // new instance
 * fonte.drawString(g, fonte.HISCORE, 0, 0);  // draw
 *
 *
 */

import javax.microedition.lcdui.Graphics;
import javax.microedition.lcdui.Image;

/**
 * @author George Roberto Peres
 *         reflexus@ig.com.br
 */

public class Fonte {

  // armazena codigo ASCII dos caracteres. cada variavel e' como se fosse uma string
  // placar
  // The only inconvenient is that you must have an ASCII table at hands to create
  // these arrays. maybe there is a better solution. if you know some, please notify me.
  // ...but after all you have no String creation (good and fast!).
  protected static final byte[] PAUSE = { 80, 32, 65, 32, 85, 32, 83, 32, 69 };
  protected static final byte[] HISCORE = { 72, 73, 45, 83, 67, 79, 82, 69 };
  //
  private static final int LARG = 240;   // largura da tela. screen width
  private static final int ALT = 320;   // altura da tela.  screen height
  protected static int charAlt;   // altura de cada caractere.  charHeight
  protected static int charLarg;   // largura de cada caractere.  charWidth
  private Image imgFont;   // imagem da fonte.png
  private static final int GRAPHICS_TOP_LEFT = Graphics.LEFT | Graphics.TOP;
```


(C) Copyright Nokia 2014. All rights reserved.

```
  // Cria uma nova instancia de Fonte
  // Constructor. Load image file, sets width and height
  protected Fonte() {
    // carrega o arquivo da imagem da fonte
    try {
      if (imgFont==null) imgFont = Image.createImage("/fonts.png");
      charLarg = imgFont.getWidth() / 96;   //=9.       96 = total chars contained in
image.png
      charAlt = imgFont.getHeight();   //=12
    }
    catch (Exception e) { }
  }



  // desenha caracter de acordo com o codigo da tabela ASCII
  // util para acrescentarmos caracteres especiais customizados
  // draw char according to ASCII table. very useful, for example,
  // to draw some special custom char (127,128)
  protected void drawCharASCII(Graphics g, int ASCII, int x, int y) {
    // caractere espaco ou nao imprimivel ou o caractere>255 nao sao desenhados
    if (ASCII < 33  ||  ASCII > 255) return;
    g.clipRect(x, y, charLarg, charAlt);   // altera clip pra tamanho do caractere.
alter clip (smaller. charLarg x charAlt)
    // desenha o caractere dentro do retangulo. (cIndex - 32) * charLarg = posicao do
caractere
    // draw char inside clipRect
    g.drawImage(imgFont, x - ((ASCII - 32) * charLarg), y, GRAPHICS_TOP_LEFT);
    // reset o clip - full screen, ou seja, volta o normal
    g.setClip(0, 0, LARG, ALT); // this instruction is very important! (turn back clip)
  }



  protected void drawString(Graphics g, byte []s, int x, int y) {
    int cx = x;   // posicao inicial.  initial position.
    // loop atraves de todos os caracteres na string. loop through all chars in s
    for (int a = 0; a < s.length; a++) {
      drawCharASCII(g, s[a], cx, y);
      cx += charLarg;   // va para a proxima posicao para desenhar. go to next position
    }
  }



  // desenha score int em x,y (ASCII de 48~57).  ex: 000777 (6 digitos - max 8)
  // this one is very useful and fast and no String creator
  // although score is int, it draws short or byte too
  protected void drawScoreInt(Graphics g, int score, int digitos, int x, int y) {
    if (digitos<=0 || digitos>8) return;
    int d = 1;
    int a;
    for (a = 1; a < digitos; a++)  d *= 10;
    a = x;   // posicao inicial
```


(C) Copyright Nokia 2014. All rights reserved.

```
    for (int i=0; i<digitos; i++) {
      drawCharASCII(g, (score/d) + 48, a, y);  // +48 pra coincidir com tabela ASCII
      a += charLarg;   // va para a proxima posicao para desenhar. go to next position
      score -= d*(score/d);
      d /= 10;
    }
  }
}
```


(C) Copyright Nokia 2014. All rights reserved.