Building a Java ME Canvas based calendar/date picker

Introduction

Here's a calendar component usable on J2ME Canvas. You can use it, for example, as a simple and intuitive date picker.

You can see this component in action here: Calendar Widget in action.

Source code: CalendarWidget class

This class represents the Calendar itself.

```
import java.util.Calendar;
import java.util.Date;

import javax.microedition.lcdui.Canvas;
import javax.microedition.lcdui.Font;
import javax.microedition.lcdui.Graphics;

public class CalendarWidget
{
}
```

Let's define some initial variables to hold labels. We'll use in our calendar: month and week day labels

```
// defining array of string.
static final String[] MONTH_LABELS = new String[]{
  "January", "February", "March", "April", "May", "June", "July", "August", "September",
"October", "November", "December"
 };
static final String[] WEEKDAY_LABELS = new String[]{
  "M", "T", "W", "T", "F", "S", "S"
 };
```

Now we'll define some appearance-related properties. This properties can be customized to make the date picker appear as you want it to look like.

```
/* starting week day: 0 for monday, 6 for sunday */
public int startWeekday = 0;

/* elements padding */
public int padding = 1;

/* cells border properties */
```


(C) Copyright Nokia 2014. All rights reserved.

```
public int borderWidth = 4;
public int borderColor = 0x0000ff;

/* weekday labels properties */
public Font weekdayFont = Font.getDefaultFont();
public int weekdayBgColor = 0x0000ff;
public int weekdayColor = 0xffffff;

/* header (month-year label) properties */
public Font headerFont = Font.getDefaultFont();
public int headerBgColor = 0x0000ff;
public int headerColor = 0xffffff;

/* cells properties */
public Font font = Font.getDefaultFont();
public int foreColor = 0x000000;
public int bgColor = 0x9999ff;
public int selectedBgColor = 0xffff00;
public int selectedForeColor = 0xff0000;
```

Now we'll define some internal properties that we'll use to handle size/time properties.

```
/* internal properties */
int width = 0;
int height = 0;
int headerHeight = 0;
int weekHeight = 0;
int cellWidth = 0;
int cellHeight = 0;

/* internal time properties */
long currentTimestamp = 0;
Calendar calendar = null;
int weeks = 0;
```

Now, let's define a simple constructor, accepting a Date instance as an argument

```
public CalendarWidget(Date date)
{
 calendar = Calendar.getInstance();

 //we'll see these 2 methods later
 setDate(date);

 initialize();
}
```

Then, we'll define some useful methods to set and get the selected date

```
public Date getSelectedDate()
{
 return calendar.getTime();
}
public void setDate(Date d)
```


(C) Copyright Nokia 2014. All rights reserved.

```
{
 currentTimestamp = d.getTime();

 calendar.setTime(d);

 //weeks number can change, depending on week starting day and month total days
 this.weeks = (int)Math.ceil(((double)getStartWeekday() + getMonthDays()) / 7);
}
public void setDate(long timestamp)
{
 setDate(new Date(timestamp));
}
```

Here we will use two methods to initialize calendar properties. This methods will be called by constructor, and should be called each time you change some public properties (like padding, fonts or border width) so that sizes can be recalculated correctly .

```
void initialize()
{
 //let's initialize calendar size
 this.cellWidth = font.stringWidth("MM") + 2 * padding;
 this.cellHeight = font.getHeight() + 2 * padding;

 this.headerHeight = headerFont.getHeight() + 2 * padding;
 this.weekHeight = weekdayFont.getHeight() + 2 * padding;

 this.width = 7 * (cellWidth + borderWidth) + borderWidth;
 initHeight();
}
void initHeight()
{
 this.height =
  headerHeight + weekHeight +
  this.weeks * (cellHeight + borderWidth) + borderWidth;
}
```

These two methods will be used to get, respectively, the number of days and the start week day of the current month.

```
int getMonthDays()
{
 int month = calendar.get(Calendar.MONTH);

 switch(month)
 {
 case 3:
 case 5:
 case 8:
 case 10:
  return 30;
 case 1:
                int year = calendar.get(Calendar.YEAR);
                return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0) ? 29 :
28;
 default:
  return 31;
 }
```


(C) Copyright Nokia 2014. All rights reserved.

```
}
int getStartWeekday()
{
 //let's create a new calendar with same month and year, but with day 1
 Calendar c = Calendar.getInstance();

 c.set(Calendar.MONTH, calendar.get(Calendar.MONTH));
 c.set(Calendar.YEAR, calendar.get(Calendar.YEAR));
 c.set(Calendar.DAY_OF_MONTH, 1);

 return (c.get(Calendar.DAY_OF_WEEK) + 5) % 7;
}
```

Now we must handle key events, to allow user to select the date they want. We handle left/right keys shifting current date by one day, while up/down keys will shift the current date by 7 days.

```
public void keyPressed(int key)
{
 switch(key)
 {
 case Canvas.UP:
  go(-7);
  break;
 case Canvas.DOWN:
  go(7);
  break;
 case Canvas.RIGHT:
  go(1);
  break;
 case Canvas.LEFT:
  go(-1);
  break;
 }
}
void go(int delta)
{
 int prevMonth = calendar.get(Calendar.MONTH);

 setDate(currentTimestamp + 86400000 * delta);

 //we have to check if month has changed
 //if yes, we have to recalculate month height
 //since weeks number could be changed
 if(calendar.get(Calendar.MONTH) != prevMonth)
 {
  initHeight();
 }
}
```

Now that we've finished with utility functions, we can finally paint our date picker.

```
public void paint(Graphics g)
{
 //painting background
 g.setColor(bgColor);
```


(C) Copyright Nokia 2014. All rights reserved.

```
 g.fillRect(0, 0, width, height);

 //painting header (month-year label)
 g.setFont(headerFont);
 g.setColor(headerColor);
 g.drawString(MONTH_LABELS[calendar.get(Calendar.MONTH)] + " " +
calendar.get(Calendar.YEAR), width / 2, padding, Graphics.TOP | Graphics.HCENTER);

 //painting week days labels
 g.translate(0, headerHeight);

 g.setColor(weekdayBgColor);
 g.fillRect(0, 0, width, weekHeight);

 g.setColor(weekdayColor);
 g.setFont(weekdayFont);

 for(int i = 0; i < 7; i++)
 {
  g.drawString(WEEKDAY_LABELS[(i + startWeekday) % 7],
   borderWidth + i * (cellWidth + borderWidth) + cellWidth / 2,
   padding,
   Graphics.TOP | Graphics.HCENTER
  );
 }

 //painting cells borders
 g.translate(0, weekHeight);

 g.setColor(borderColor);

 for(int i = 0; i <= weeks; i++)
 {
  g.fillRect(0, i * (cellHeight + borderWidth), width, borderWidth);
 }
 for(int i = 0; i <= 7; i++)
 {
  g.fillRect(i * (cellWidth + borderWidth), 0, borderWidth, height - headerHeight -
weekHeight);
 }

 //painting days
 int days = getMonthDays();
 int dayIndex = (getStartWeekday() - this.startWeekday + 7) % 7;

 g.setColor(foreColor);

 int currentDay = calendar.get(Calendar.DAY_OF_MONTH);

 for(int i = 0; i < days; i++)
 {
  int weekday = (dayIndex + i) % 7;
  int row = (dayIndex + i) / 7;

  int x = borderWidth + weekday * (cellWidth + borderWidth) + cellWidth / 2;
  int y = borderWidth + row * (cellHeight + borderWidth) + padding;
```


(C) Copyright Nokia 2014. All rights reserved.

```
  //if this is the current day, we'll use selected bg and fore colors
  if(i + 1 == currentDay)
  {
   g.setColor(selectedBgColor);
   g.fillRect(
    borderWidth + weekday * (cellWidth + borderWidth),
    borderWidth + row * (cellHeight + borderWidth),
    cellWidth, cellHeight);
   g.setColor(selectedForeColor);
  }

  g.drawString("" + (i + 1), x, y, Graphics.TOP | Graphics.HCENTER);

  //if this is the current day, we must restore standard fore color
  if(i + 1 == currentDay)
  {
   g.setColor(foreColor);
  }
 }
 //let's translate back!
 g.translate(0, - headerHeight - weekHeight);
}
```

Sample usage

Here's how you can initialize and customize your Canvas based date picker:

```
calendar = new CalendarWidget(new Date());

calendar.headerFont = Font.getFont(Font.FACE_PROPORTIONAL, Font.STYLE_BOLD,
Font.SIZE_LARGE);
calendar.weekdayFont = Font.getFont(Font.FACE_PROPORTIONAL, Font.STYLE_BOLD,
Font.SIZE_MEDIUM);
calendar.weekdayBgColor = 0xccccff;
calendar.weekdayColor = 0x0000ff;
calendar.headerColor = 0xffffff;

calendar.initialize();
```

Resources

You can download full source code here:

- CalendarWidget.java: CalendarWidget source code
- CalendarCanvas.java: sample usage of CalendarWidget within a Canvas

(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.