Adding sound to SVG animation with XML parsing in Java ME

Overview

The Scalable 2D Vector Graphics API (JSR-226) conforms to SVG Tiny 1.1 and does not provide support for multimedia. The multimedia functionality as described by SVG Tiny 1.2 is supported by the Scalable 2D Vector Graphics API 2.0 (JSR-287) which is currently not supported by Nokia devices.

The multimedia limitation on JSR-226 can be circumvented by using XML parsing which is provided by the Web Services API (JSR-172).

Below is an example application which utilises the described XML parsing solution for playing audio: reading the <audio> tag from the XML file (here: the SVG content file **content3.svg**) and its attributes from an SVG file are enabled by using XML parsing. Once the attributes have been retrieved, the Mobile Media API (JSR-135) is used to implement the audio playback functionality for the application.

The source file below includes key code snippets from the application. The full code together with a compiled Java MIDlet can be downloaded from the packaged ZIP file linked at the bottom of this page.

The original SVG example (without the XML parsing solution) which has been used for making this XML parsing example can be found in Nokia Developer Java Developer's Library: Java Developer's Library 3.6 > Source codes for examples > Scalable 2D Vector Graphics - HelloWorldMidlet, LoadStaticMidlet, and SvgAnimatorMidlet

Source file

SvgAnimatorMidlet.java

```
import java.io.InputStream;
import javax.microedition.lcdui.*;
import javax.microedition.media.*;
import javax.microedition.midlet.*;
import javax.microedition.m2g.*;
import javax.xml.parsers.*;

public class SvgAnimatorMidlet extends MIDlet implements CommandListener {
 public SvgAnimatorMidlet() {
   SVGImage svgImage = null;
    try {
         InputStream svgStream =getClass().getResourceAsStream("content3.svg")
         svgImage = (SVGImage)( SVGImage.createImage( svgStream, null ) );
         }
    catch ( Exception e ){
        e.printStackTrace();
         }
   svgAnimator = SVGAnimator.createAnimator( svgImage );
   svgAnimator.setSVGEventListener( new MySvgEventListener() );
   Canvas c = (Canvas)(svgAnimator.getTargetComponent());
   Display.getDisplay(this).setCurrent( c );
   c.setCommandListener(this);
   parseData();
  }

 // Implements the XML parser for playing audio

 public void parseData() {
    try {
         SAXParserFactory factory = SAXParserFactory.newInstance();
         SAXParser saxParser = factory.newSAXParser();
         InputStream is = getClass().getResourceAsStream("content3.svg");
         saxParser.parse(is,new XmlHandler(this));
         }
    catch(Exception ex) {
         System.out.println("parsing exception");
         System.out.println(ex);
         }
  }
 public void startApp() {
   svgAnimator.play();
  }
}
```

XmlHandler.java

```
import java.io.IOException;
import java.io.InputStream;
import org.xml.sax.*;
import org.xml.sax.helpers.*;
import javax.xml.parsers.*;
import java.util.*;
import javax.microedition.media.Manager;
import javax.microedition.media.MediaException;
import javax.microedition.media.Player;

/* Implements the XML handler for processing audio tag and
its attributes from the SVG file, sets and starts the audio player by
using the retrieved attributes */
```


(C) Copyright Nokia 2013. All rights reserved.

```
public class XmlHandler extends DefaultHandler{
 private SvgAnimatorMidlet midlet;
 private Player p;
 private static final String audio = "audio";
 private static final String xlink = "xlink:href";
 private static final String type = "type";
 String attribute1="";
 String attribute2="";

 public XmlHandler (SvgAnimatorMidlet midlet){
   this.midlet = midlet;
 }

 public void startDocument() throws SAXException {
  // No implementation
 }

 public void startElement(String uri, String localName, String qName,
   Attributes attributes) throws SAXException{

   if(audio.equals(qName))
    {
     attribute1 = attributes.getValue(xlink);
     attribute2 = attributes.getValue(type);
     tagStack.push(qName);
     play();
    }

 public void play()
  {
    try {
         InputStream is = getClass().getResourceAsStream(attribute1);
         p = Manager.createPlayer(is, attribute2);
         p.setLoopCount(5);
         p.start();
   }catch (IOException ioe) {
   }catch (MediaException me) {
  }

 public void endElement(String uri, String localName, String qName,
   Attributes attributes) throws SAXException{
    tagStack.pop();
 }

 public void endDocument() throws SAXException{
  //No implementation
 }
}
```

content3.svg

```
<?xml version="1.0" standalone="no"?>
<svg width="176" height="178" stroke="none" baseProfile="tiny"
xmlns:xlink="http://www.w3.org/1999/xlink" timelineBegin="onLoad">

<audio xlink:href='background_music.wav' volume='7' type='audio/X-wav'
 begin='mybutton.mouseover' end='mybutton.mouseout' repeatCount='indefinite'/>
```

Sample application

File:SVGExample.zip

See also:

JSR 172: XML Parsing Example

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.