

---

## SVG PLAYER – INTERACTIVE PICTURES

Vladimir Batagelj, University of Ljubljana

---

### What is SVG ?

SVG - *Scalable Vector Graphics* is a new 2D-graphics markup language based on XML. The development of SVG started in October 1998; the last version of SVG definition was published on 4<sup>th</sup> September 2001 already as a W3C Recommendation. SVG is compatible with other web standards: HTML, XML Namespace, Xlink, Xpointer, CSS 2, DOM 1, Java, ECMA/Javascript, Unicode, SMIL 1.0, ... It allows us to include in HTML documents pictures described by their structure – composition of curves, shapes, text and bitmaps. Since the SVG viewer is not integrated yet into most web browsers we need, to view SVG pictures, to install it as a plug-in. An excellent SVG plug-in for Windows was produced by Adobe.

The SVG pictures are not static (as standard bitmaps GIF, TIFF, JPEG, PNG). The SVG viewer provides options to zoom in (to see details) and out (to see global view), to move the picture, to search for text, ... Besides this, using built-in animation capabilities or Javascript program support, the pictures can be made alive and interactive. To get some impression about SVG see the following examples:

*Kaleidoscope*: <http://www.burningpixel.com/svg/Kaleid.htm>,

*Vienna - Social patterns and structures*: <http://www.karto.ethz.ch/~an/cartography/vienna/>,

*Europe*: <http://www.carto.net/papers/svg/eu/index.html>,

*Pilat Informatique Educative*: <http://perso.wanadoo.fr/pilat/>

*PC Technology in motion*: [http://www.usbyte.com/index\\_SVG.htm](http://www.usbyte.com/index_SVG.htm),

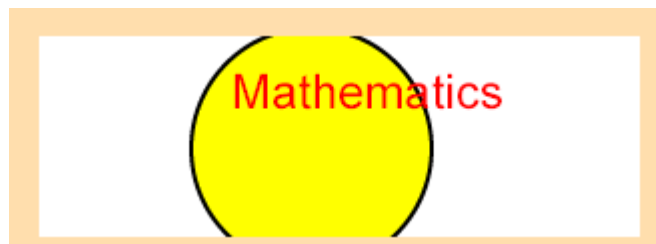
*How to write hiragana characters*: <http://www.kenjikojima.com/HiraganaQuiz/>.

SVG pictures can be produced by drawing tools. In Windows we can use Adobe Illustrator, Corel Draw, WebDraw (by Jasc) and Mayura. But special programs for visualization of obtained data/results will produce most SVG pictures. The main SVG application areas are: data visualization, presentations (like Power Point), maps (GIS), technical layouts, and educational illustrations.

### A simple example in SVG

Here is a simple example of picture description in SVG

It creates yellow circle with black border containing red inscription “Mathematics”.



To insert a SVG picture into a HTML document we use the simple example we have:

tag. For the picture from our

The attribute `src` determines the location (URL) of the SVG file; `xmlns` becomes important in advanced applications using Javascript or Java. The attributes `width` and `height` are obligatory and determine the size of rectangle in which the picture is rendered. The value of `type` is the MIME-type of the file - for SVG file it can be `image/svg+xml` or `image/svg`. The attribute `onmouseover` directs the user that has not a SVG viewer installed on his computer, to the web site from which he can obtain a viewer.

## SVG player

Using the `SVG` tag we can partition a SVG picture to several parts. In the `SVG` tag two attributes `visibility` and `display` are available to control the visibility of the corresponding part. The difference between them is explained in SVG documentation. We shall use the `display` attribute. Setting it to `none` we switch the visibility of the part off; setting it to `inline` (or some other value) we switch the visibility of the part on. Using Javascript this can be done interactively allowing to the user to select the parts to be displayed.

To support the visibility controlled SVG pictures I prepared the *SVG player* - a collection of Javascript functions for controlling the value of `display` attribute in parts of the SVG picture. The *SVG player* is given in the listing of the file `svgplayer.js`. The Javascript functions and SVG picture are related by two arrays `picList` and `picIndex` embedded as `script` in SVG picture file. The array

`picList` contains a list of names of picture parts; the array `picIndex` is the inverse of `picList` – for each name returns the corresponding index. For example:

```
<svg xml:space="preserve" width="300" height="50" viewBox="0 0 200 50">
<script><![CDATA[
  PicList[0] = "shape1";  PicIndex["shape1"] = 0;
  PicList[1] = "shape2";  PicIndex["shape2"] = 1;
  ...
]]></script>
...
<g id="shape1" style="display:inline">
  <rect x="5" y="5" width="40" height="40" style="&sty0;" /></g>
<g id="shape2" style="display:none">
  <circle cx="75" cy="25" r="20" style="&sty1;" /></g>
...
</svg>
```

The picture control HTML file combines *SVG player* ( `svgplayer.js` ) and SVG picture and using HTML form tags provides the interactive control of the visibility of parts of the picture.

```
<EMBED WIDTH="600" HEIGHT="300" SRC="./picture.svg" NAME="PicA"
  PLUGINSOURCE="http://www.adobe.com/svg/viewer/install/">
<FORM NAME="control">
<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="2" WIDTH="100%">
<TR><TD><INPUT type="button" value=" first " onClick="setAllPics('PicA','none');
  setPic('PicA',this.form,0,'inline');"></TD>
<TD><INPUT type="button" value=" last "
  onClick="setAllPics('PicA','none'); setPic('PicA',this.form,5,'inline');"></TD></TR>
<TR><TD><INPUT type="button" value=" next " onClick="nextPic('PicA',this.form);"></TD>
<TD><INPUT type="button" value=" prev " onClick="prevPic('PicA',this.form);"></TD></TR>
<TR><TD><INPUT type="button" value=" showA " onClick="setAllPics('PicA','inline');"></TD>
...
</FORM>
```

File:

}

Because of the limited space available for this paper we can not give here a detailed explanation of the *SVG player* logics. The hints given here should be sufficient to understand the examples available on the internet. A ZIP file with the last version of *SVG player* and related example files is available at <http://vlado.fmf.uni-lj.si/pub/SVG/SVGplayer/> . A test application of SVG player displaying all its capabilities can be found at <http://sio.edus.si/list/1/svg/svg03.htm> .

### Some applications of *SVG player*

One of the simplest applications of *SVG player* is to combine several bitmap pictures into a single SVG picture as its parts. Then, using the *SVG player*, we can display selected pictures in the same place on the page. An example of this you can see at <http://vlado.fmf.uni-lj.si/pub/CONF/DSI.01/kuk.htm> .

At <http://sio.edus.si/list/1/svg/svg04.htm> you will find the page with SVG visualization, using the *SVG player*, of the step by step construction of the perpendicular (from a point to a line).

Based on the ideas used in *Logo2PS* and *SVG player* I prepared for *MSWLogo* a collection of logo commands *Logo2SVG* that allows user to save a trace of the Logo turtle as a (visibility controlled) SVG picture <http://vlado.fmf.uni-lj.si/educa/logo/logo2svg/> .

The option to produce (visibility controlled) SVG visualizations we implemented also in our program *Pajek* (<http://vlado.fmf.uni-lj.si/pub/networks/pajek/>) for analysis and visualization of large networks. Some examples are available at: <http://vlado.fmf.uni-lj.si/pub/networks/pajek/examples/> .

### References

1. Vladimir Batagelj: *Logo to Postscript*. Proceedings of Eurologo'97. Budapest, 1997, p. 333-341. <http://vlado.fmf.uni-lj.si/educa/logo/logo2ps/>
2. V. Batagelj: *Logo to SVG*, Proceedings of Eurologo'01, Linz, Austria, 2001, p. 229-236.
3. The World Wide Web Consortium - W<sup>3</sup>C: <http://www.w3.org>
4. XML: <http://www.w3.org/XML/>
5. W<sup>3</sup>C/SVG: <http://www.w3.org/TR/SVG/index.html>
6. HTML: <http://www.w3.org/MarkUp/>
7. XML Namespace: <http://www.w3.org/TR/REC-xml-names/>
8. Xlink: <http://www.w3.org/TR/WD-xlink>
9. Xpointer: <http://www.w3.org/TR/WD-xptr>
10. CSS 2: <http://www.w3.org/TR/REC-CSS2/>
11. DOM 1: <http://www.w3.org/TR/REC-DOM-Level-1/>
12. ECMA/JavaScript: <http://www.ecma.ch/ecma1/stand/ecma-290.htm>
13. Java: <http://java.sun.com/>
14. Unicode: <http://www.unicode.org/>
15. SMIL 1.0: <http://www.w3.org/TR/REC-smil/>
16. Adobe SVG Viewer: <http://www.adobe.com/svg/viewer/install/>
17. Adobe Illustrator: <http://www.adobe.com/products/illustrator/main.html>
18. Corel Draw: <http://www.corel.com/svg>

19. Jasc Software, WebDraw: <http://www.jasc.com/webdraw>

20. Mayura: <http://www.mayura.com/>

**Author**

Prof. Vladimir Batagelj  
University of Ljubljana, FMF, Department of Mathematics  
Jadranska 19, 1000 Ljubljana, Slovenia  
Email – [vladimir.batagelj@uni-lj.si](mailto:vladimir.batagelj@uni-lj.si)  
WWW – <http://vlado.fmf.uni-lj.si/>