

libavg

Was ist libavg?

- Open-Source Authorensystem
- Pendant zu svg (Scalable Vector Graphics) für pixelorientierte Grafik
- ‚Multimedialer Mixer‘
- Ziel: Alternative zu Macromedia Director

Was kann libavg?

- Strukturierte Layoutbeschreibungen von Szenen (xml) in Präsentationen umsetzen.
- Konkret: Bilder, Videos und Text auf dem Bildschirm ‚mischen‘.
- Transparenzunterstützung.
- Dynamische Veränderungen der Szene über Skripte.

Statische Szenen

Leere Szene

```
<?xml version="1.0"?>
<!DOCTYPE avg SYSTEM "../../doc/avg.dtd">
<avg id="emptyavg" width="800" height="600"
      fullscreen="false" bpp="16">
</avg>
```

- 2 Zeilen xml-Einordnung.
- avg-Knoten, in dem allgemeine Szenenparameter angegeben werden.
- Innerhalb des avg-Knotens stehen alle anderen Szenenelemente.

Bitmaps

```
<image id="testing" x="0" y="0" z="1"
        width="199" height="199" opacity="0.5"
        href="rgb24.tif"/>
```

- x und y geben die Position der Bitmap in Pixeln an.
- Die z-Werte verschiedener Elemente bestimmen die Reihenfolge, in der die Elemente gemalt werden.
- width und height geben Breite und Höhe der Bitmap an.
- opacity ist die Transparenz.
- href ist der Dateiname.
- Alpha-Kanäle (z.B. in tiffs oder pngs) werden unterstützt.

Videos

```
<video id="clogo1" x="0" y="0" z="1"  
      loop="true" opacity="1.0"  
      width="352" height="288"  
      href="c-logo_wire.mpg"/>
```

- x, y, z, width, height, opacity, href wie Bitmaps.
- loop gibt an, ob das Video als Endlosschleife gespielt werden soll.
- Videos laufen erst los, wenn sie über Scriptbefehl gestartet werden.

Text

```
<words id="cbasetext" x="20" y="50" z="1"  
      opacity="1" size="30" font="arial"  
      color="FF8000"  
      text="Willkommen an Bord der c-base"/>
```

- x, y, z, opacity wie gehabt.
- font ist der Name eines Truetype-Fonts, der auf dem Rechner installiert ist.
- color ist eine Farbangabe in typischer html-Schreibweise.
- size ist die Größe in Pixel.
- text ist der eigentliche text, der angezeigt werden soll.

avg

```
<avg id="mainavgnode" width="640" height="480"
  fullscreen="true">
  <avg id="nestedavg" x="0" y="100" z="3"
    opacity="1" width="400" height="100">
    <image id="nestedimg1" x="0" y="0" z="1"
      href="rgb24.tif"/>
    <image id="nestedimg2" x="200" y="0" z="2"
      href="rgb24alpha.tif"/>
  </avg>
  <image id="mainimg" x="0" y="0" z="2"
    width="400" height="300"
    href="rgb24.tif"/>
</avg>
```

- avg-Knoten können weitere avg-Knoten enthalten.
- Innere avg-Knoten definieren rechteckige Bereiche auf dem Bildschirm, in denen sich weitere Elemente befinden können.
- Positionen von Elementen sind relativ zur Position des umgebenden avg-Knotens

excl

```
<excl id="switch">
  <image id="img1" x="0" y="0" z="1"
    width="100" height="100"
    href="rgb24.tif"/>
  <image id="img2" x="100" y="0" z="2"
    width="100" height="100"
    href="rgb24.tif"/>
</excl>
```

- Nur ein Kind eines excl-Knoten wird angezeigt.
- Welches, kann über javascript bestimmt werden.
- Sinnvoll für Rollovers, oder um zwischen unterschiedlichen Layouts umzuschalten.

Dynamische Szenen

- Steuerung über separate .js-Datei.
- Kompletter Sprachumfang von Javascript steht zur Verfügung.
- In den Skripten können die meisten Attribute von Szenenelementen geändert werden.

Grundstruktur

```
use("../avg.js");  
  
var ok = AVGPlayer.loadFile("schleuse1.avg");  
if (!ok) {  
    print("Szenenbeschreibung nicht vorhanden");  
}  
AVGPlayer.play(25);
```

Ereignisse

mousedown.avg:

```
<image id="testmousedown" x="0" y="0"  
      z="1" href="rgb24.tif"  
      onmousedown="handleClick();" />
```

mousedown.js:

```
function handleClick()  
{  
    print ("click");  
}
```

- Skripte können aufgerufen werden, wenn Ereignisse auftreten.
- Benutzereingaben: onmousedown, onmouseup, onmouseover, onmousemove, onmouseout.
- Zeitsteuerung: setTimeout und setInterval.

Auf Szenenelemente zugreifen

```
var node = AVGPPlayer.getElementByID  
          ("clogovideo");  
node.play();  
node.x = 100;
```

Welche Attribute und Funktionen im einzelnen verfügbar sind, steht unter <http://www.libavg.de/reference/html/index.html>.

Interna

- Kern von libavg ist die display engine.
- Für die Anzeige wird DirectFB eingesetzt, libavg benötigt also einen funktionierenden Framebuffer.
- Der js-Interpreter und das Komponentensystem stammen aus Mozilla.
- Bilder werden über paintlib geladen.
- Für Videos ist libmpeg3 verantwortlich.

Weiterentwicklung

- PPC Linux support
 - Touchscreen support
 - OpenGL support (Elemente skalieren, rotieren, schneller auf heutigen Rechnern,...)
 - DirectFB Hardwarebeschleunigung
 - Besserer Font/Text support. Layout? Schräger Text?
 - Netzwerkunterstützung
 - Soundunterstützung
-
- Stylesheets
 - Mac OS X
 - Windows
 - Vektorgrafik
 - ...