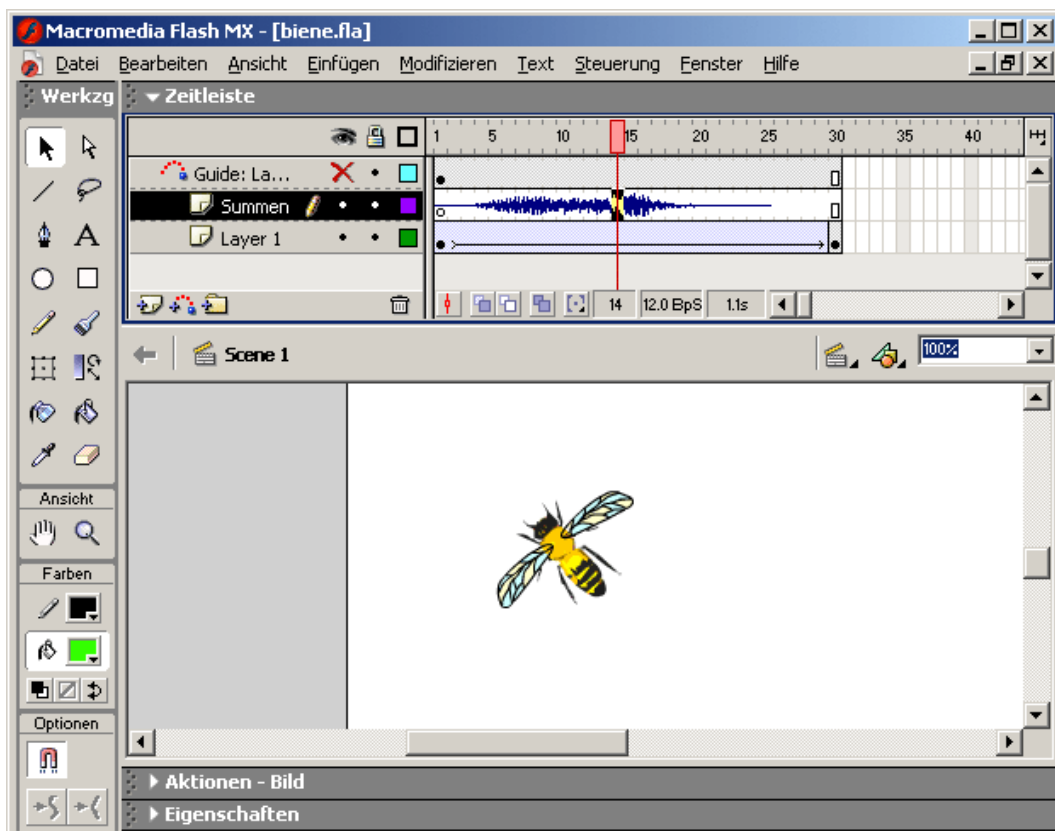


Kapitel 10

2D-Grafik im Web

10.1 Macromedia Flash

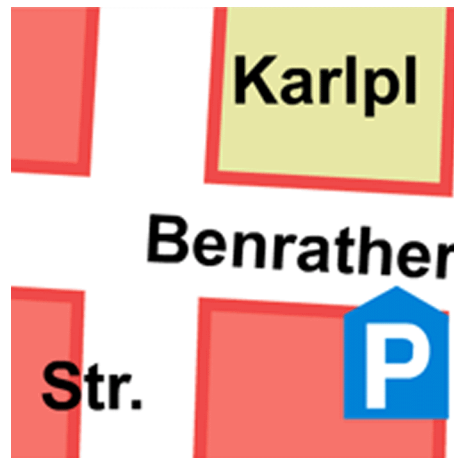
Macromedia Flash ist ein Werkzeug zum Editieren und Animieren von zweidimensionalen Vektorgrafiken. Durch sein kompaktes Speicherformat ist es besonders geeignet für plakative, animierte Grafiken im Internet. Zum Abspielen in einem Web-Browser ist das Macromedia-Flash-Plugin erforderlich.



Vektorgrafikwerkzeug Macromedia Flash MX



Gesamtansicht des Stadtplans



Teilansicht des Stadtplans

Rasterfreies Zoomen bei Dateien im Flash-Format durch Vektorgrafik

```

<HTML>
  <HEAD>
    <TITLE>Macromedia Flash</TITLE>
  </HEAD>
  <BODY BGCOLOR="f9d1a1">
    <CENTER>
      <H1>Eingebetteter Flash-Film</H1>
      <EMBED
        src="vogel.swf"
        width=700
        height=200
        Loop="True"
        Play="True"
        BGColor="f9d1a1"
        Quality="Autohigh"
        Scale="showall"
        SAlign=" "
      >
      <P>vogel.swf, 6 K
    </CENTER>
  </BODY>
</HTML>

```

Flash mit Einzelbild-Animation

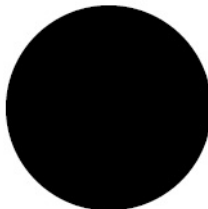
10.2 SVG

Im Gegensatz zum proprietären Binärformat Flash handelt es sich bei SVG (Scalable Vector Graphics) um einen offenen Standard eines Vektorgrafikformates auf XML-Basis. Entwickelt wurde dieses Format von einem Verbund von Firmen, darunter Adobe, Apple, Autodesk, BitFlash, Corel, HP, IBM, ILOG, Macromedia, Microsoft, Netscape, OASIS, Quark, RAL, Sun, Visio, W3C, Xerox.

SVG ...

- kann nach Text durchsucht werden
- erlaubt Textgestaltung
- kann Objekte zeitlich koordiniert bewegen
- enthält Eventhandling
- verfügt über photoshopartige Filter-Effekte
- ist sowohl unkomprimiert als auch komprimiert einsetzbar
- erfordert zum Abspielen ein Plugin
- verlangt Rechenleistung auf der Clientseite

Die SVG-Datei `kreis.svg` platziert einen Kreis mit Radius 80 in die Mitte der Zeichenfläche:



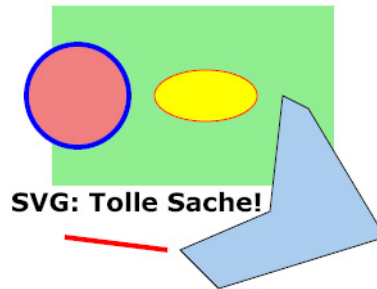
SVG-Screenshot: Kreis

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
  "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg width="200" height="200" xmlns="http://www.w3.org/2000/svg"
  xmlns:xlink="http://www.w3.org/1999/xlink" >
  <circle cx="100" cy="100" r="80" />
</svg>
```

SVG-Quelltext: Kreis

Die SVG-Datei `basics.svg` enthält die grafischen Objekte Gerade, Rechteck, Kreis, Ellipse, Polygon sowie Text. Die Objekte erhalten als Attribute ihre Koordinaten sowie Angaben zur Farbe der Füllung, Farbe der Umrandung und Stärke der Umrandung.



SVG-Screenshot: Gerade, Rechteck, Kreis, Ellipse, Polygon, Text

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
  "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg width="310" height="270" viewBox="0 0 310 270"
xmlns="http://www.w3.org/2000/svg"
  xmlns:xlink="http://www.w3.org/1999/xlink" >

  <title>Gerade, Rechteck, Kreis, Ellipse, Polygon, Text</title>

  <line    x1="50" y1="210" x2="130" y2="220"
    stroke="red" stroke-width="4" />

  <rect    x="40" y="30" width="220" height="140"
    fill="lightgreen" stroke-width="12" />

  <circle  cx="60" cy="100" r="40"
    fill="lightcoral" stroke="blue" stroke-width="4px" />

  <ellipse cx="160" cy="100" rx="40" ry="20"
    fill="orange" stroke="red" stroke-width="1" />

  <polygon points="220,100 240,110 300,210 170,250 140,220 210,190"
    fill="#abcdef" stroke="#000000" stroke-width="1"/>

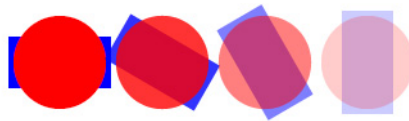
  <text    x="8" y="190"
    style="font-family:verdana; font-size:20px; font-weight:bold">
    SVG: Tolle Sache! </text>

</svg>
```

SVG-Quelltext: Gerade, Rechteck, Kreis, Ellipse, Polygon, Text

Die SVG-Datei `gruppierung.svg` fasst mit dem `<g>`-Element mehrere Objekte zusammen und referenziert sie später über das `xlink:href`-Element (erst dann werden sie gezeichnet). Weiterhin finden vier Transformationen statt. Die erste besteht aus einer einzelnen Translation. Die zweite und dritte bestehen aus einer Translation, gefolgt von einer Rotation. Die vierte wird beschrieben durch Angabe von sechs Zahlen $(a\ b\ c\ d\ e\ f)$, welche eine 3×3 -Transformationsmatrix beschreiben mit erster Spalte $(a\ b\ 0)^T$, zweiter Spalte $(c\ d\ 0)^T$ und dritter Spalte $(e\ f\ 1)^T$.

Obacht: Transformationen beziehen sich auf das Koordinatensystem des Objekts. `transform=rotate(45) translate(200, 0)` bedeutet daher zunächst eine Drehung im Uhrzeigersinn um 45 Grad und dann eine Verschiebung um 200 Pixel längs der Diagonalen nach rechts unten.



SVG-Screenshot: Gruppierung, Wiederverwendung, Transformation

```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
  "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg width="400" height="100" viewBox="0 0 400 100"
xmlns="http://www.w3.org/2000/svg"
  xmlns:xlink="http://www.w3.org/1999/xlink" >

  <title>Gruppierung und Wiederverwendung</title>

  <defs>
    <g id="vorlage">
      <rect x="-40" y="-20" width="80" height="40" fill="blue" />
      <circle cx="0" cy="0" r="36" fill="red" />
    </g>
  </defs>

  <use xlink:href="#vorlage"
    transform="translate( 80 50)" />

  <use xlink:href="#vorlage"
    transform="translate(160 50) rotate(30)" fill-opacity="0.8"/>

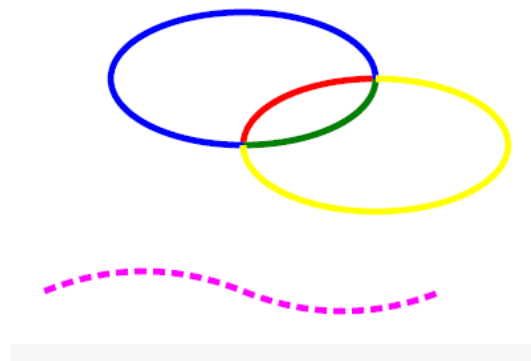
  <use xlink:href="#vorlage"
    transform="translate(240 50) rotate(60)" fill-opacity="0.5"/>

  <use xlink:href="#vorlage"
    transform="matrix(0.0 -1.0 1.0 0.0 320 50)" fill-opacity="0.2"/>

</svg>
```

SVG-Quelltext: Gruppierung, Wiederverwendung, Transformation

In der Datei `kurven.svg` werden unter Verwendung des `path`-Element Kurven gezeichnet. Die vier Kreisbögen (Kommando A) beginnen jeweils bei (300,100) haben einen x-Radius von 100 und einen y-Radius von 50 und enden bei (200,100). Sie unterscheiden sich durch die vier Kombinationen von *large-arc-flag* und *sweep-flag*. Die erste kubische Bezierkurve (Kommando C) beginnt bei (50,260) und hat als weitere Kontrollpunkte (100,240), (150,240) und (200,260). Mit dem Kommando S wird der erste Kontrollpunkt der nächsten kubischen Bezierkurve automatisch errechnet durch die beiden letzten Kontrollpunkte der Vorgängerkurve. Daher reicht nun die Angabe zweier weiterer Punkte (300,280) und (350,260). Über das `stroke-dasharray`-Attribut wird die Kurve gestrichelt gezeichnet: auf 9 gefüllte Pixel folgen 5 leere Pixel.



SVG-Screenshot: Kreisbögen und Bezierkurve

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
  "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg width="500" height="300" xmlns="http://www.w3.org/2000/svg"
  xmlns:xlink="http://www.w3.org/1999/xlink" >

  <title>Kurven</title>

  <path fill="none" stroke="red" stroke-width="5"
    d="M 300 100 A 100 50 0 0 0 200 150" />

  <path fill="none" stroke="green" stroke-width="5"
    d="M 300 100 A 100 50 0 0 1 200 150" />

  <path fill="none" stroke="blue" stroke-width="5"
    d="M 300 100 A 100 50 0 1 0 200 150" />

  <path fill="none" stroke="yellow" stroke-width="5"
    d="M 300 100 A 100 50 0 1 1 200 150" />

  <path fill="none" stroke="magenta" stroke-width="5" style="stroke-dasharray: 9,5;"
    d="M 50 260 C 100 240 150 240 200 260 S 300 280 350 260"/>

</svg>
```

SVG-Quelltext: Kreisbögen und Bezierkurve

In der Datei `laufrad.svg` wird durch vierfache Wiederverwendung eines Viertelrades ein Speichenrad erzeugt. Das Speichenrad erhält durch `animateTransform` eine horizontale Verschiebung und eine Rotation.



SVG-Screenshot: Animation von Transformationen

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN"
  "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
<svg width="400" height="150" xmlns="http://www.w3.org/2000/svg"
  xmlns:xlink="http://www.w3.org/1999/xlink" >

  <title>Laufrad</title>

  <defs>
    <g id="rad">
      <path id="viertel" fill="white" stroke="black"
        d="M 10,10 L 40,10 A 30,30 0 0,1 10,40 L 10,10 " />
      <circle cx="0" cy="0" r="50" fill="red" stroke="black" />
      <circle cx="0" cy="0" r="6" fill="white" stroke="black" />
      <use xlink:href="#viertel" x="0" y="0" transform="rotate( 0)" />
      <use xlink:href="#viertel" x="0" y="0" transform="rotate( 90)" />
      <use xlink:href="#viertel" x="0" y="0" transform="rotate(180)" />
      <use xlink:href="#viertel" x="0" y="0" transform="rotate(270)" />
    </g>
  </defs>

  <use xlink:href="#rad" x="-50" y="75">

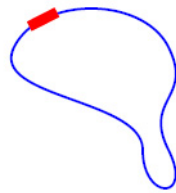
  <animateTransform xlink:href="#rad"
    attributeName="transform"
    type="translate"
    from="-50,0" to="500,0"
    begin="0s" dur="6s"
    repeatCount="indefinite"/>

  <animateTransform xlink:href="#rad"
    attributeName="transform"
    type="rotate"
    from="0" to="360"
    begin="0s" dur="3s"
    additive="sum"
    repeatCount="indefinite"/>

</svg>
```

SVG-Quelltext: Animation von Transformationen

Die Datei `bezier.svg` definiert eine Bezierkurve durch das `path`-Element. Dabei wird zunächst mit `M` der Startpunkt festgelegt, dann folgen durch `C` drei weitere Stützpunkte, die zusammen mit dem Vorgänger eine kubische Bezierkurve beschreiben. Durch `S` werden jeweils der dritte und vierte Stützpunkt für eine weitere kubische Bezierkurve definiert; automatisch übernommen werden als erster Stützpunkt der letzte Kurvenpunkt und als zweiter Stützpunkt die Verlängerung der Geraden durch die beiden letzten Kurvenpunkte. Weiterhin wird durch `animateMotion` das rote Rechteck längs der Bezierkurve bewegt. Hierbei entsteht die resultierende Kurve durch Verknüpfung der Pfad-Koordinaten mit den Objektkoordinaten.



SVG-Screenshot: Bezier-Kurve als Animationspfad

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
  "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg width="260" height="200" viewBox="0 0 260 200"
  xmlns="http://www.w3.org/2000/svg"
  xmlns:xlink="http://www.w3.org/1999/xlink" >

  <title>Bezier-Kurve mit Pfadanimation</title>

  <rect id="kasten" x="-12" y="-4" fill="red" width="24" height="8" />

  <path id="kurve" stroke="blue" stroke-width="2" fill="none"
    d="M 40,60
      C 60,20 140,20 160,60
      S 140,100 160,140
      S 140,180 140,140
      S 20,100 40,60" />

  <use xlink:href="#kasten" x="0" y="0" />
  <animateMotion xlink:href="#kasten"
    begin      ="0s" dur="5"
    rotate     ="auto"
    fill       ="freeze"
    calcMode  ="linear"
    repeatCount="indefinite" >
    <mpath xlink:href="#kurve" />
  </animateMotion>

</svg>
```

SVG-Quelltext: Bezier-Kurve als Animationspfad

In der Datei `hyperlink.svg` wird durch Anklicken des grünen, abgerundeten Rechtecks die Webseite `http://www.inf.uos.de` aufgerufen.



SVG-Screenshot: Hyperlink

```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
  "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="200" height="200" xmlns="http://www.w3.org/2000/svg"
  xmlns:xlink="http://www.w3.org/1999/xlink" >

  <title>Hyperlink</title>

  <a xlink:href="http://www.inf.uos.de/">
  <rect x="50" y="50" rx="8" ry="8" fill="green" width="100" height="100"/></a>
  <text x="80" y="105" font-size="20" fill="white" >GO !</text>

</svg>
```

SVG-Quelltext: Hyperlink

In der Datei `synchronisation.svg` wird durch Klicken der roten Kugel eine Bewegung gestartet und nach 4 Sekunden wieder gestoppt. Weiterhin wird 4 Sekunden nach Beginn der Bewegung die Wave-Datei `signal.wav` abgespielt.



SVG-Screenshot: Synchronisation mit Audio

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN"
  "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
<svg xmlns:a="http://www.adobe.com/svg10-extensions" width="400"
height="100" xmlns="http://www.w3.org/2000/svg"
  xmlns:xlink="http://www.w3.org/1999/xlink">

  <title>Synchronisation</title>

  <circle id="kugel" cx="30" cy="50" r="25" fill="red"/>
  <rect id="wand" x="325" y="0" width="25" height="100" fill="blue" />

  <animateMotion xlink:href="#kugel"
    begin="kugel.click" dur="4"
    path="M 0,0 270,0"
    fill="freeze" />

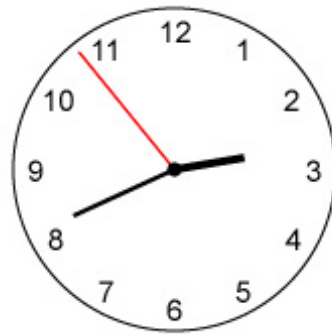
  <a:audio xlink:href="signal.wav" begin="kugel.click+4s"></a:audio>

</svg>
```

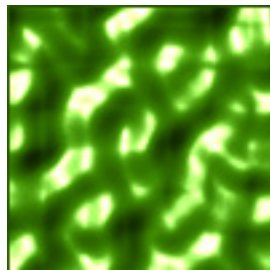
SVG-Quelltext: Synchronisation mit Audio



animierte Ellipsen



Systemzeit mit Javascript übertragen



Photoshopartige Filter

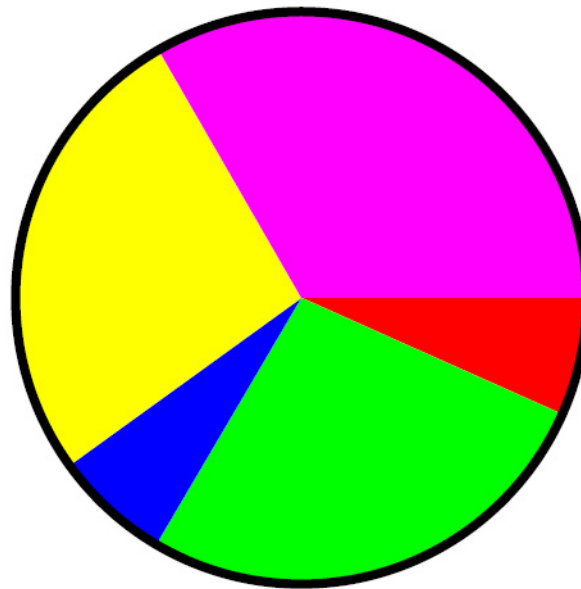
HIT ME!

Button mit Interaktion



Schatteneffekt auf Pixelgrafik

In Kombination mit PHP können die Ergebnisse einer Datenbankquery dynamisch zu einer SVG-Grafik aufbereitet werden. Der folgende Screenshot zeigt eine Tortengrafik als Visualisierung der Lehrbelastung der Professoren aus der Datenbank UNI aus der Vorlesung Datenbanksysteme im SS 2003.



dynamisch erzeugte SVG-Grafik nach Datenbankquery