Learning to Code with SVG

Lesson Plan: Coding a Frog in SVG on a 600 by 600 grid

Objective: Hands-on learning of SVG by drawing a frog with basic shapes; rectangles, circles and ellipses.
Introduce path element commands, M, L, & Q, to draw a lines and curves and CSS property: stroke-linecap.

Lab Time: Approximately 1 hour, not including Lecture time. Students should copy and paste lines, and then change the attribute values to greatly reduce typing time and typos.

Age range: 4-8th grades, or any age student unfamiliar with SVG

Requirements: Familiar with a simple text editor like Notepad on Windows, or Text Wrangler on a Mac.
Familiar with copy and paste shortcuts (ctrl-c and ctrl-v on windows and command-c and command-v on a Mac).
Ability to save file with a .svg extension.
Should be familiar with adding <circle> and <ellipse> elements in SVG

Resources: http://steamcoded.org/lessons/grid600x600.svg
http://steamcoded.org/images/curve1.svg
http://steamcoded.org/images/curve2.svg
http://www.w3schools.com/svg/default.asp
https://www.w3.org/TR/SVG11/
Free eBook for the iPad *Learn SVG Interactively*, by Jay Nick

Lecture: SVG elements consist of tagnames, attributes, and their values in the form <tagname attribute="value"> The style attribute consists of CSS name:value pairs separated by a semi-colon. Each name is separated from its value with a colon. For example, style="name:value;name:value;" Like SVG, CSS is a web standard for styling web technologies including SVG and HTML.

Discuss how to draw a curve with the path command. The path d attribute consists of a list of commands, i.e. M (moveto), L (lineto), Q (Quadratic Bezier Curve), etc. Note that path commands that are uppercase letters are absolute coordinates and lower case letters are relative coordinates, so case does matter.

To draw the curve in this lesson students should understand that a curve can be drawn with 3 points: starting and ending points of the curve and a control point not on the curve that controls it's shape. Demonstrate how a curve is drawn by using the resources for curve1 and curve2 above - which show how the curve is drawn using straight line segments using just 3 points.

In the instructions below, the starting point for the smile is (220,170) - so M (moveto) start. The Q (curve) takes 2 points, first the control point (300,280), then the end point (380,170) The path command should be of the form <path d="MstartQcontrol,end" /> so the resulting code is <path d="M220,170Q300,280,380,170" />

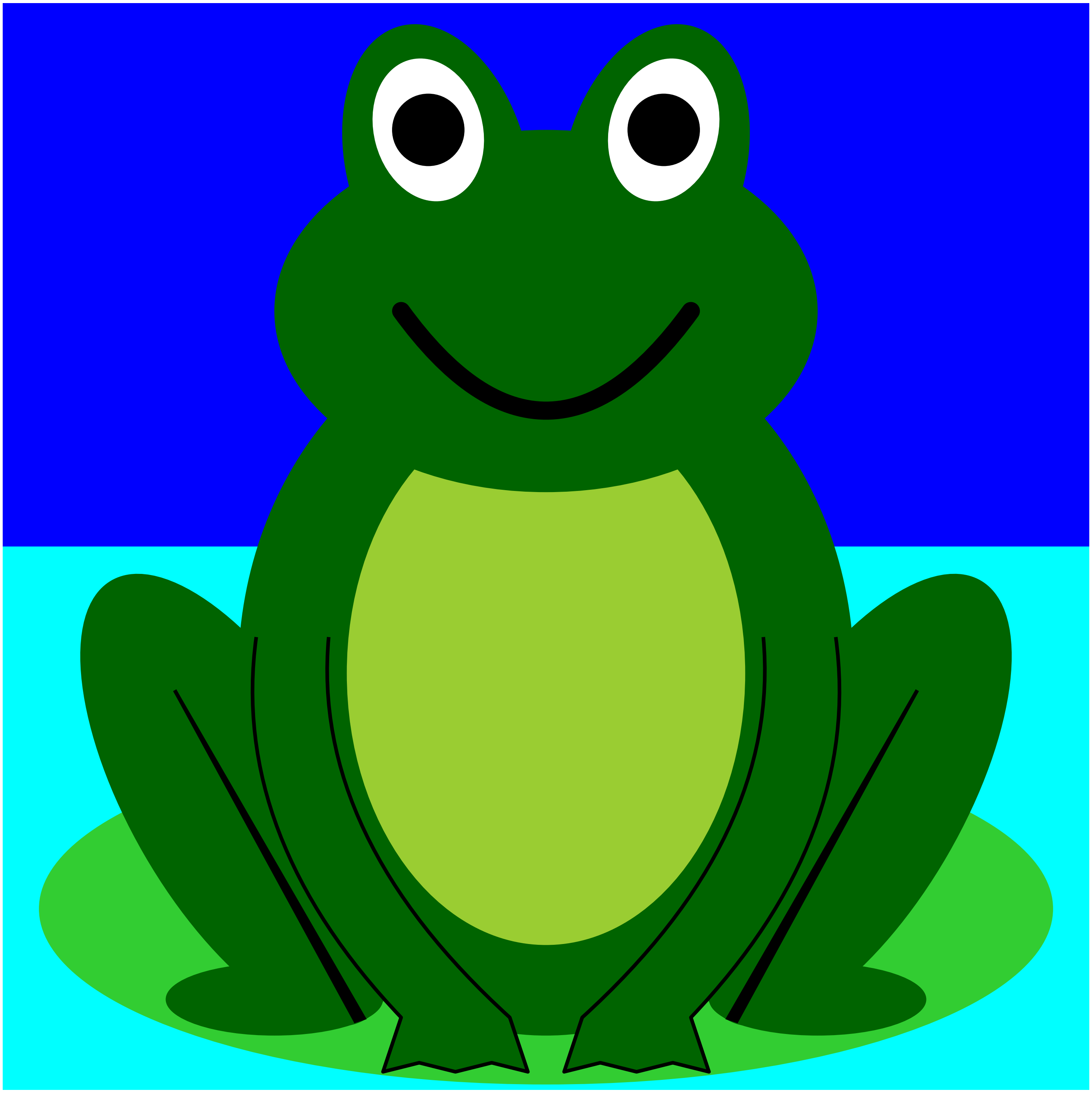
Procedure: Have students get an SVG template with 600x600 grid from:
http://steamcoded.org/lessons/grid600x600.svg.txt
Copy the code and paste it into a text editor.
Save the file as Frog.svg then open in a browser. Keep the text editor and browser windows open.

Add SVG elements where indicated using the instructions on page 4 (see below). **Important:** Students should save the file and refresh the browser after adding each SVG element to their file.

When complete, change the style attribute of the first <g> element from "display:initial" to "display:none" which hides the grid Then change the style style attribute of the second <g> element from "opacity:0.5" to "opacity:1"

Take Away: Students should feel comfortable adding circles and ellipses when creating SVG images.
Students should understand that a Quadratic Bezier Curve can be drawn with 3 points, starting and ending points of the curve, and a control point not on the curve, but controls the curves shape.

Additional Activity: Students can modify the y value of the control point of the curve to create a frown or bigger/smaller smile. Or change other values to distort the smile and have fun.



Coding a Frog in SVG on a 600 by 600 grid

To get started copy the code of this image into your editor: http://steamcoded.org/lessons/grid600x600.svg.txt and save the file as Frog.svg and open the file in a browser.

In the editor, add the SVG elements (per instructions below) where indicated in the SVG code, i.e. inside the second grouping (<g> element).

Syntax Examples:
A rectangle starting at (0,0) with a width of 600 and a height of 600 would have the following code:
<rect x="0" y="0" width="600" height="600" />

An circle centered at (235,70) with a radius of 20 would have the following code:
<circle cx="235" cy="70" r="20" />

An ellipse centered at (300,500) with an x-radius of 280 and a y-radius of 97 would have the following code:
<ellipse cx="300" cy="500" rx="280" ry="97" />

1: Add another default style - fill:darkgreen; - to the group with style="opacity:0.5;" so it looks like:
<g style="opacity:0.5;fill:darkgreen;">

2: Draw an rectangle starting at (0,0), width: 600, height: 600, style="fill:blue;"

3: Draw an rectangle starting at (0,300), width: 600, height: 300, style="fill:aqua;"

4: Draw an ellipse at (300,500), x-radius: 280, y-radius: 97, style="fill:limegreen;"

5: Draw an ellipse at (150,550), x-radius: 60, y-radius: 20

6: Add a group <g> element with attribute: transform="rotate(30,470,440)"
<g transform="rotate(30,470,440)">
</g>
This rotates the objects within the group (child elements) by 30 degrees about point (470,440)

7: Add an ellipse at (130,440), x-radius: 60, y-radius: 140 as a child element of the group in step 6

8: Add a <path> element as a child element of the group in step 6 with attributes:
d="M130,370L130,579L125,578z"
style="fill:black;stroke:black;stroke-width:2px;"

9: Copy the code from steps 5-8 and paste on the next line, then change:
ellipse attribute cx, from "150" to "450"
transform from "rotate(-30,130,440)" to "rotate(30,470,440)"
ellipse attribute cx, from "130" to "470"
path attribute d, from "M130,370L130,579L125,578z" to "M470,370L470,579L475,578z"

10: Draw an ellipse at (300,370), x-radius: 170, y-radius: 200

11: Draw an ellipse at (300,370), x-radius: 110, y-radius: 150, style="fill:yellowgreen;"

12: Draw an ellipse at (300,170), x-radius: 150, y-radius: 100

13: Draw an ellipse at (240,90), x-radius: 50, y-radius: 80 with attribute:
transform="rotate(-15,240,90)"
This rotates the ellipse -15 degrees about its center

14: Draw an ellipse at (360,90), x-radius: 50, y-radius: 80 with attribute:
transform="rotate(15,360,90)"
This rotates the ellipse 15 degrees about its center

15: Draw an ellipse at (235,70), x-radius: 30, y-radius: 40, with attributes:
transform="rotate(-15,235,70)"
style="fill:white;"

16: Draw a circle at (235,70), radius: 20, style="fill:black;"

17: Copy the code from steps 15-16 and paste on the next line, then change:
ellipse attribute cx, from "235" to "365"
transform from "rotate(-15,235,70)" to "rotate(15,365,70)"
circle attribute cx, from "235" to "365"

18: Add a <path> element with attributes:
d="M220,170Q300,280,380,170"
style="fill:none;stroke:black;stroke-width:10px;stroke-linecap:round;"

19: Add a <path> element with attributes:
d="M140,350Q125,460,220,560l-10,30l20,-5l20,5l20,-5l20,5L280,560
Q170,460,180,350"
style="stroke:black;stroke-width:2px;stroke-linejoin:round;"
Note the lower case L is a line with relative coordinates and is not the number 1

20: Add a <path> element with attributes:
d="M460,350Q475,460,380,560l10,30l-20,-5l-20,5l-20,-5l-20,5L320,560
Q430,460,420,350"
style="stroke:black;stroke-width:2px;stroke-linejoin:round;"

When complete, change the style attribute of the first element from "display:initial" to "display:none" which hides the grid. Then change the style style attribute of the second element from "opacity:0.5" to "opacity:1"

Coding a Frog in SVG on a 600 by 600 grid

Answer Sheet

Common mistakes are missing double quote marks around attribute values, missing space between attributes, missing the start < and ending /> symbols, and inserting a closing '/' on the opening <g> tag.

```
 1:   <g style="opacity:0.5;fill:darkgreen;">

 2:   <rect x="0" y="0" width="600" height="600" style="fill:blue" />

 3:   <rect x="0" y="300" width="600" height="300" style="fill:aqua" />

 4:   <ellipse cx="300" cy="500" rx="280" ry="97" style="fill:limegreen" />

 5:   <ellipse cx="150" cy="550" rx="60" ry="20" />

 6:   <g transform="rotate(-30,130,440)" />
      </g>

 7:   <ellipse cx="130" cy="440" rx="60" ry="140"" />

 8:   <path d="M130,370L130,579L125,578z"
      style="fill:black;stroke:black;stroke-width:2px;" />

 9:   <ellipse cx="450" cy="550" rx="60" ry="20" />
      <g transform="rotate(30,470,440)">
         <ellipse cx="470" cy="440" rx="60" ry="140" />
         <path d="M470,370L470,579L475,578z"
            style="fill:black;stroke:black;stroke-width:2px;" />
      </g>

10:   <ellipse cx="300" cy="370" rx="170" ry="200" />

11:   <ellipse cx="300" cy="370" rx="110" ry="150" style="fill:yellowgreen;" />

12:   <ellipse cx="300" cy="170" rx="150" ry="100" />

13:   <ellipse cx="240" cy="90" rx="50" ry="80" transform="rotate(-15,240,90)" />

14:   <ellipse cx="360" cy="90" rx="50" ry="80" transform="rotate(15,360,90)" />

15:   <ellipse cx="235" cy="70" rx="30" ry="40" transform="rotate(-15,235,70)"
      style="fill:white;" />

16:   <circle cx="235" cy="70" r="20" style="fill:black;" />
```

17:

```
<ellipse cx="365" cy="70" rx="30" ry="40" transform="rotate(15,365,70)"
style="fill:white;" />
<circle cx="365" cy="70" r="20" style="fill:black;" />
```

18:

```
<path d="M220,170Q300,280,380,170"
style="fill:none;stroke:black;stroke-width:10px;stroke-linecap:round;" />
```

19:

```
<path d="M140,350Q125,460,220,560l-10,30l20,-5l20,5l20,-5l20,5L280,560
Q170,460,180,350" style="stroke:black;stroke-width:2px;stroke-linejoin:round;" />
```

Note the lower case L is a line with relative coordinates and is not the number 1

20:

```
<path d="M460,350Q475,460,380,560l10,30l-20,-5l-20,5l-20,-5l-20,5L320,560
Q430,460,420,350" style="stroke:black;stroke-width:2px;stroke-linejoin:round;" />
```

Note the lower case L is a line with relative coordinates and is not the number 1