PART 1: Graphics File Formats

Common Graphics Files

• .BMP
• .JPG / .JPEG
• .GIF
• .SVG

One of these things is not like the others...

• .BMP
  • Bitmap (Native to Microsoft Windows)
  • Uncompressed, all pixels stored
  • Four color modes:
    – 24-Bit (16,777,216 colors)   3 bytes/pixel
    – 256 Color                   1 byte/pixel
    – 16 Color                    ½ byte/pixel
    – 2 Color (Monochrome)        1 bit/pixel
    – First stores colors directly, last three use a Palette
  • Not a good choice for the Web due to size

• .JPG / .JPEG
  • Joint Photographic Experts Group (pronounced “J-Peg’)”
  • 16,777,216 possible colors (24-bit color)
  • Compressed, uses “lossy” compression
    – Converting a .BMP to .JPG changes some pixels
    – Increases overall compression by doing so
    – Changes won’t be noticed in “noisy” images
  • Great for photographs
  • Terrible for text, cartoons, line art

• .GIF
  • Graphics Interchange Format (CompuServe 1987, 1989), pronounced “JIFF”
  • Up to 256 maximum colors, but uses a palette
  • Compressed, uses “lossless” compression
  • Supports transparency, simple loop animations
  • OK but not great for photographs
  • Great for text, cartoons, line art
  • Patent entanglements diminished use 1995-2010
. PNG

- “Zillions” of colors: supports up to 48-bit color
- Supports both lossy & lossless compression
- Supports transparency (no animations, sorry)
- Great for photographs, text, cartoons, line art
- Free from patent entanglements
- Increasing support by graphics packages

What’s Common About These?

- All formats so far (.BMP, .JPG, .GIF, .PNG) are pixel-based formats.
- Creatable by painting programs:
  - Windows Paint (comes with Windows)
  - Mac Paintbrush (free download)
  - Photoshop (not free)
- Painting new object over old changes the old
- Scaling up shows a bad case of “jaggies”

. SVG

- Scalable Vector Graphics
- NOT a pixel based graphics format
- IS a vector based graphics format
- Objects have separate existence from images
- Created by drawing programs, not painting
- Looks good when scaled up or down in size
- Text Coding similar to HTML (actually XML)
- Not all browsers fully support .SVG (e.g., I.E.)

PART 2: Bézier Curves

Bézier Curves

- A Bézier Curve (named after Pierre Bézier) is:
  - A Piecewise,
  - Parametric,
  - Cubic,
  - Polynomial.

Polynomial

- A simple polynomial is an equation, with:
  - An independent variable,
  - A series of terms based on powers of that variable,
  - Simple numeric coefficients for those terms (no sines and cosines, no calculus, no fancy stuff, just numbers).
  - \( f(x) = \cdots + ax^4 + bx^3 + cx^2 + dx + e \)
Cubic

- Maximum power of any term is 3.
- \( f(x) = ax^3 + bx^2 + cx + d \)
- For Bézier curves this means that there are at most two changes in direction.

Parametric

- \( y \) is not a function of \( x \), locking graph to the coordinate axes, but...
- \( x \) and \( y \) (and \( z \) if we go to 3 dimensions) are now functions of a new independent variable, the parameter, often called \( t \).
- For Bézier curves this means that the curve can be oriented anywhere in the plane or in space – and is not locked to the coordinate axes.

Parametric (continued)

- \( x(t) = a_xt^3 + b_xt^2 + c_xt + d_x \)
- \( y(t) = a_yt^3 + b_yt^2 + c_yt + d_y \)
- \( z(t) = a_zt^3 + b_zt^2 + c_zt + d_z \)
- You can extend this method to even higher dimensions, even if you can’t visualize the results!

Piecewise

- If you want a complicated shape, you have to stick a bunch of Bézier curves end-to-end, but...
- ...you have to be careful to get one curve to flow smoothly into the next!

Points

- Four points define a Bézier Curve
  - Two end points and two control points,
  - Each end point associated with one control point.
- The curve goes off infinitely in each direction, but we are interested only in the convex hull:

Control Lines

- Each end point and its associated control point form a control line, where...
- ...the Bézier curve is tangent to each control line at the end point
Joining Two Bézier Curves

- But control lines aren’t the same:

Joining Two Bézier Curves

- But control lines ARE the same:

Summary

- When joining two Bézier curves end-to-end, make sure that the:
  - Second control point of the first curve,
  - The common end points, and
  - The first control point of the second curve...
- ...are all in a straight line...
- ...and the first curve will blend smoothly into the second!