Filed on: [Indigo]<Artwork>TiogaArtwork>3.2>TA.doc
Last edit: July 6, 1982 5:22 pm

Tioga Artwork

Richard J. Beach
July, 7, 1982

Introduction

A prototype for integrating text and figures within a Tioga document exists. Tioga Artwork represents a successful experiment in graphical style. Artwork which can be integrated presently includes Griffin pictures, AIS scanned images, and the results of executing JaMGraphics programs, along with text captions.

The graphical style experiment attempted to provide a similar model of separating form from content as exists in many text formatters. In our graphical style model, geometry represents the content and its rendering represents the form. Graphical style is defined using the Tioga style mechanisms and can describe colors, line weights, filled or outlined areas, pen styles, and shadow styles.

A Tioga document with both text and graphics can be typeset using a version of the Tioga Typesetter. This version of the typesetter incorporates TiogaArtwork to render graphical nodes of a Tioga document. In a symmetric fashion, TiogaArtwork uses the typesetter to compose the text within captions.

Incorporating AIS Images into Documents

As an introductory example of using TiogaArtwork, consider the following instructions for incorporating a sampled image into a document. Note that if someone were to modify PressScreen to produce an AIS file, then the resulting screen file could both be printed on a Press printer and be included in a Tioga document.

Create a text node in the document where the image is to be placed. Use the Tioga Break command to make this a separate node. Enter the local filename of the AIS image as the text of the node.. The node will appear in the document window as normal text so it is necessary to make the node special for TiogaArtwork. Use the EditTool to set the ArtworkImage property for this node: 1) enter ArtworkImage in the `Property name:` field, 2) enter `TRUE` in the `Property value:` field, 3) select the node containing the image filename, and 4) bug the `Set` button in the `Property` section of the EditTool.

Typeset the document using the TiogaArtwork version of the Tioga typesetter.

Incorporating Griffin Illustrations into Documents

To incorporate Griffin pictures in Tioga documents, the Griffin file first must be converted to TiogaArtwork. Retrieve the Griffin file onto the local disk. Using the Tioga Artwork Viewer, enter the filename in the box and bug Griffin -> TiogaArtwork! button.

Create a text node in the document where the Griffin picture is to be placed. Use the Tioga Break command to make this a separate node. Enter the local filename of the converted Griffin picture, which will be the Griffin picture name with the extension ".Artwork" as text within the node. To make this

node special for TiogaArtwork, use the EditTool to set the ArtworkFileName property for this node: 1) enter ArtworkFileName in the `PropertyName:` field, 2) enter `TRUE` in the `PropertyValue:` field, 3) select the node containing the converted filename, and 4) bug the `Set` button in the `Property` section of the EditTool.

Typeset the document using the TiogaArtwork version of the Tioga typesetter.

Software Configuration

At present, TiogaArtwork is bound up with the Tioga typesetter. The package name is TATS. When this is run from the UserExec, two viewers are created: Typesetter and Tioga Artwork Viewer. The Typesetter user interface is identical to the Tioga Typesetter.

The Tioga Artwork Viewer provides three buttons and a GetSelection menu button. When a button is selected and Tioga Artwork is busy, the button will appear grey with black lettering. Messages describing the current state of Tioga Artwork appear below the buttons at the bottom of the viewer. All buttons are documented with explanations which appear when the button is bugged with the middle mouse button.

Tioga Artwork Document Structure

The document structure expected by Tioga Artwork reflects both the Tioga document structuring capabilities and graphical clustering of picture objects. A design criteria for this prototype was to use only text nodes within Tioga documents, so as to avoid any modifications to the Tioga text editor. Therefore all the geometry descriptions are represented either as text which are JaMGraphics commands or filenames of artwork files. To distinguish text which represents graphical objects, properties are assigned to such nodes.

Converted Griffin pictures will contain the necessary properties to be rendered successfully. Other nodes created by hand which did not originate as converted files will have to have such properties added manually. The Tioga EditTool provides this functionality.

The provision of graphical style utilizes the same style mechanisms as Tioga and its typesetter. Since Tioga provides for extensible style parameters, most graphical style parameters are extensions of the Tioga styles. Tioga Artwork renders the picture according to these graphical style parameters. Format names applied to a node indicate the style parameters in force when the node is rendered.

A common expectation in working with illustrations for documents is the ability to group parts of a picture and manipulate the group. Positioning and scaling are two common operations applied to subpictures within an illustration. Positioning of textual captions is another operation. The tree structure of Tioga documents provides a natural representation of such hierarchial structure within illustrations. Therefore Griffin pictures which contain clusters of objects are represented as nested Tioga nodes. Each Griffin object is made relative to an origin of (0,0) with its containing node supplying the appropriate positioning commands. Clusters of objects supply further positioning commands for the group. Furthermore, Tioga Artwork provides a nested display context for each level in the tree. Thus positioning and scaling transformations apply only to the nodes contained within the Tioga subtree and no explicit management of display contexts is needed.

Tioga Artwork Properties

Properties distinguish nodes which contain nontextual interpretation of their contents. Tioga Artwork uses the following properties for graphical illustrations: Artwork, ArtworkImage, ArtworkFileName, ArtworkPath, BoundingBox, and Origin. An artwork node must have at least one of the three properties: Artwork, ArtworkImage or ArtworkFileName. Properties are generated automatically during the Griffin to TiogaArtwork conversion, but must be manually applied for all other artwork nodes.

The property Artwork with the value TRUE, in the absence of either the ArtworkImage or ArtworkFileName property, means that the node contains JaMGraphics commands. Normally the positioning and scaling information nodes have this property.

The property ArtworkImage with the value TRUE means that the node contains the local filename of an AIS image file. Tioga Artwork will interpret the text as a local filename and cause the image to be drawn.

The property ArtworkFileName with the value TRUE means that the node contains the local filename of an artwork file (hopefully different from any containing filename!). Tioga Artwork will interpret the text as a local filename and render the artwork contained therein. This mechanism allows the inclusion of converted Griffin illustrations with great ease.

The property ArtworkPath with the value TRUE means that the node contains solely geometry information. The path definition relies on the three JaMGraphics commands: .moveto, .lineto, and .curveto. The graphical style parameters determine how this path should be rendered. A Format name is required to define which style rule is in force when the path is rendered.

The property BoundingBox with the value "xmin,ymin,xmax,ymax" defines the bounding box of the graphical object. The Griffin to TiogaArtwork conversion automatically creates this property. However, the information is not used by Tioga Artwork at present. It should be helpful in speeding up the layout procedure required by the Typesetter.

The property Origin with the value "x,y" defines the centre of rotation for the graphical object. The Griffin to TiogaArtwork conversion creates this property with the value 0,0. The intention is to define a centre of rotation which might not be simply the lower left corner of the bounding box. For example, a circular object should have its origin as the centre of the circle.

Graphical Style Parameters

The graphical style parameters provide extensions to those style parameters implemented by Griffin. The definition of most graphical style parameters resides in the file BasicGraphics.Style.

Color Styles

Color style is defined in terms of Hue, Saturation and Brightness values. Note that CedarGraphics refers to Brightness as Value; Tioga implements color styles with Brightness and must change before TiogaArtwork can conform to this standard. Color values are in the range [0..1]. BasicGraphics.Style provides named colors to simplify color specification. These names originate from the Griffin Color Chart. Colors within style rules can be specified in three ways: 1) three real numbers for hue, saturation and brightness, each in the range 0.0 to 1.0; 2) a named color from BasicGraphics.Style; 3) a computed value such as "50 percent areaSaturation" to specify a lighter color. For each color style, mumble, there are three separate style parameters which can be accessed: mumbleHue, mumbleSaturation, and mumbleBrightness.

The style parameter textColor defines the color of text. At present, only the Tioga Typesetter interprets this style parameter when creating a Press file. Viewers might display colored text on the color viewer someday.

The two style parameters areaColor and outlineColor define the color of graphical objects. The pathType style determines the choice between areaColor and outlineColor styles.

Path Styles

The pathType style has three possible states: filled, outlined, and filled+outlined. A filled path appears as a colored area using the areaColor style parameter. An outlined path appears as an uncolored (the background color will show through) area surrounded by a line colored with the outlineColor. A filled+outlined path appears as a colored area surrounded by a line.

The lineWeight style defines the width of the outline measured in points (1/72 of an inch). Fractional values are accepted. The default value is a single pixel line (essentially zero points).

Pen Styles

The penStyle parameter determines the shape of the pen which renders the outline. The default pen is round, but other choices are square, rectangular, italic, and elliptical. An italic pen refers to the flat

nibbed pens used in calligraphy. Pens may be parameterized by lineWeight, penSlant, penWidth, and penHeight described below.

The penSlant style parameter determines the angle of the pen measured in degrees from the horizontal counter-clockwise. Obviously this does not apply to round pens. The default is no slant.

The penWidth and penHeight parameters together define the aspect ratio of the pen. Obviously these parameters do not apply to round, square or italic pens. The ratio is applied to the lineWeight to determine the actual pen width and height. The default values are both 1.

Shadow Styles

Once the style information is separated from the graphical information, derived styles become possible. Shadows are created by applying special styles to the graphical object prior to rendering the actual object. Two kinds of shadows are offered: drop shadows and offset shadows.

The shadowType style parameter defines the type of shadow desired: none, drop or offset. The default is none. A drop shadow is drawn by following the path of the object with an italic brush. An offset shadow is drawn by drawing the object offset by some distance.

The shadowDirection style parameter determines the general direction for the placement of the shadow. Four directions are provided: upLeft, upRight, downLeft, and downRight. The default is downRight.

The shadowAngle style parameter provides fine tuning of the shadow direction. The default is 45 degrees. The angle in degrees is measured from the horizontal counter-clockwise.

The shadowWeight style parameter defines the width of the shadow. For drop shadows this is the width of the italic pen, and for offset shadows it is the width of the outline drawn around the shadow. The width is measured in points (1/72 of an inch) and fractional values are accepted. The default value is 1 point.

The shadowOffsetAmount style parameter determines how far the offset shadow should be placed from the original. The distance is measured in points (1/72 of an inch) and fractional values are accepted. The offset shadow is placed shadowOffsetAmount points at shadowAngle degrees in the shadowDirection direction. The default value is 12 points.

The shadowPathType style parameter applies to the offset shadow. The same choices as pathType apply here: filled, outlined, and filled+outlined. The default is filled.

The shadowAreaColor style definition provides the shaded color for the shadow. This is either the drop shadow color of the italic pen, or it is the filled color for offset shadows which have either a filled or filled+outlined shadowPathType.

The shadowOutlineColor applies to the outline of the offset shadow when it has the outlined or filled+outlined shadowPathType.

Expected Geometry Commands

Tioga Artwork expects the geometry definition to be in terms of JaMGraphics commands. Very distinct sets of commands are generated during the conversion of Griffin pictures to Tioga Artwork. Layout commands provide positioning and scaling: .translate, .scale, and .rotate. Path commands define the geometric outline of the object: .moveto, .lineto, and .curveto. Commentary within the converted Griffin pictures begins with a percent sign, %, which JaM accects as a comment delimiter.

The graphical style rendering assumes that nodes with the property ArtworkPath will contain path definitions. This path may be executed several times to achieve the stylistic effects specified by pathType and shadowType. Only the JaMGraphics commands .moveto, .lineto, and .curveto have been tested to ensure that they generate the appropriate behaviour. Other drawing commands may be added to TACallig.Mesa as the demand arises.

Any other node with the Artwork property but without the ArtworkPath property is simply passed to JaMGraphics for interpretation. Commands which modify the display context will be in effect for all nodes in the subtree spawned by that node in the Tioga document. That is, each subtree inherits the

display context of its parent as its own.

The typesetter requires Tioga Artwork to layout the size of each piece of artwork. At present, for a lack of a more clever technique, the artwork is interpreted twice: once for layout sizing, and once to render it. The layout pass relies on a passive bounding box device to gather bounding box information without displaying anything. Thus a figure within Tioga Artwork must be capable of being rendered twice.

Captions within Artwork

Tioga Artwork relies on the typesetter to compose text captions. This provides both the high-quality text formatting algorithms implemented in the typesetter and the identical style mechanism for defining caption styles within the document. Captions within artwork must meet two criteria: 1) they must be nodes within an artwork branch, and 2) they must not have any artwork property (any of Artwork, ArtworkImage, ArtworkFileName).

The Format name of the node determines how the typesetter will render the caption. The type family, face and size are determined by the format style rule. Horizontal justification is defined by the lineFormatting style parameter within the margins established by the style rule. For most situations, a zero leftIndent provides consistent placement. Until the lineLength style parameter is interpreted by the Typesetter, you will have to compute your own rightIndent: 6.5 inches minus your intended line length gives the value for rightIndent.

Tioga Artwork will place the caption at location (0,0) in the current display context. Thus it is expected that a surrounding artwork node will do an appropriate .translate to establish the caption anchor position. In the absence of any vertical justification style parameters (not yet defined by the typesetter or Tioga), the caption is placed with the top of the caption at y=0. Hence the caption descends in the negative y direction. The horizontal placement is determined by the lineFormatting style: flushLeft implies place at (0,0) and draw to the right,
flushRight implies place the caption to the left of (0,0), and centred implies position the centre of the caption width at (0,0).

Some Layout Tips

Background Boxes

Drawing a box (.moveto ... .lineto ... .lineto ... .lineto ... .lineto, not via .drawbox) with an appropriate style (filled pathType and some areaColor) provides a background for the figure. Shading used in some magazines and textbooks can be accomplished this way. For printing on white paper, choose a light color; for preparing a 35mm slide or video tape frame, choose a dark color, traditionally a dark blue. Position the box so the enclosed artwork is placed as you like it.

Scaling the Artwork to Fit

At present, the layout process is iterative and manual. However, the best technique is a combination of experimenting with JaMGraphics and interating the typeset document. If you experiment with JaMGraphics, then you can immortalize your scaling and positioning parameters in the root node of the artwork branch. If necessary, create a new artwork node and use the Tioga Nest command to place the artwork branch underneath. That new node is the place to insert the layout directives.

Including Output From JaM Programs

The complication here is the artwork dictionary used to redefine the graphics primitives: .moveto, .lineto, and .callig. Also, you must preload the modules that register your JaMGraphics commands. Since the artwork dictionary is in your way when you wish to execute your JaM program, use the following technique to suspend the artwork dictionary and later resume it: ".end what-ever-your-JaM-program-invocation-is artwork .begin".

Be prepared for the fact that Tioga Artwork executes the artwork twice: once for layout size determination and a second time for rendering.