

Inter-Office Memorandum

To	Griffin Users	Date	October 28, 1981
From	Maureen Stone	Location	Palo Alto
Subject	How to use Griffin	Organization	ISL

Filed on: [Indigo]<Griffin>Release>HowToUseGriffin.press

What is Griffin?

Griffin is a stand-alone interactive illustrator for creating full-page color illustrations composed of lines, curves, filled areas, and captions. Illustrations are constructed on the screen using the mouse and menus, and can then be printed at high resolution on a color press printer such as the Puffin or Pimlico.

To run Griffin, you must have an Alto II with at least 2 banks (128K) of memory and Mesa6 prompts, a Dolphin or a Dorado. Run the command file [Indigo]<Griffin>Release>NewGriffin.cm, which will retrieve the following files:

Griffin.run: the illustrator itself.

Griffin.fonts: contains a selection of fonts.

ColorChart.griffin: contains a sample of all the Griffin colors.

Circle.griffin: contains a closed curve which approximates a circle.

To run Griffin, you must have Griffin.run on your disk, along with Griffin.fonts or at least one strike font (for the menus), and a current fonts.widths. To enter the illustrator, type:

Griffin RETURN

To exit from Griffin, bug 'Quit' on the master menu. A menu will be displayed asking for confirmation ('Yes' or 'No').

Fonts

Fonts are chosen from a menu that lists all the fonts in Griffin.fonts. To add to this list, simply load the desired strike fonts onto your disk and start Griffin. Griffin will use the information in the strike font plus the information in fonts.widths to make new entries in Griffin.fonts. There is currently no way to delete fonts other than deleting Griffin.fonts and starting over.

Griffin tries to make displayed and printed text strings look the same by using the information in fonts.widths to position text on the screen. The positioning parameters are set when the font is entered into Griffin.fonts. Therefore, it is important that you generate Griffin.fonts using the version of fonts.widths that matches your printer. In general, if printed captions are mispositioned, rebuilding Griffin.fonts with the correct version of fonts.widths should fix the problem.

Color Griffin

There is a version of Griffin which will run on a Dorado with a color monitor. Color Griffin may be found on [Indigo]<Griffin>Color>Griffin.image. You can use the command file above to retrieve the auxillary files from [Indigo]<Griffin>Release>.

Objects

The basic graphical unit in Griffin is the *object*. Single objects may be grouped in a *cluster*, which behaves as a single object for most operations. The basic unit to be printed or saved is the *illustration*, which may be composed of any number of objects.

There are three types of object in Griffin: shapes, captions, and control objects. *Shapes* consist of straight and curved lines, either singly or *linked* end-to-end into more complex figures, and may be either *closed* (end points linked) or *open* (end points not linked). Closed shapes may be filled with a solid color. *Captions* are composed of text typed from the keyboard, which may be rotated by any multiple of 90 degrees.

Control objects that may appear on the Griffin screen include *menus*, *control points*, and the *control grid*. Control points are used in defining the geometry of shapes, and are placed and deleted with the mouse. The grid is provided as an aid in placing objects and control points precisely.

Every object has a *geometry* and a *style*. The geometry of shapes is defined by placing control points and fitting lines or curves through them. For captions, the geometry consists of an anchor position and a text string. Style attributes include such things as color, line thickness, font, etc. The style of an object can be changed without affecting the geometry, and vice versa.

Objects have a definite order of *overlap*. A newly created object is always placed at the top; operations are available for explicitly changing and controlling the overlap order of the objects in an illustration. Filled shapes are opaque, obscuring those below them in the overlap order. Shapes that are not filled are transparent, except on the border.

Menus

All Griffin commands are entered by menu selection. The keyboard is used only for typing captions and file names. Menu items are selected by clicking down over the desired item on the menu with any of the three mouse buttons. Outside of the menus, the three mouse buttons have different functions, described below under 'Mouse Operations'.

Menus appear on top of the picture being drawn; they can be moved around to provide access to all parts of the picture. The *master menu*, always visible on the screen, controls the display of other menus used for various specialized purposes. Each item on the master menu is a *toggle* that can be turned on and off by repeated clicking with the mouse. Turning one of these toggles on causes the associated menu to appear on the screen; when the toggle is turned off, the menu vanishes. Some of the second-level menus in turn include toggles that control even more specialized *submenus*.

Some menus, known as *one-of-many menus*, have the property that only one item on the menu may be selected at a time. Selecting any item on such a menu automatically deselects the previously selected item.

The following table summarizes the various menus available in Griffin and the toggles that control them.

Menu	Toggle	On menu
Master menu	Always present	- - - -
Edit menu	Edit'	Master menu
*Shape menu	Edit'	Master menu
*Spline type menu	Spline Type'	Edit menu
Object menu	Objects'	Master menu
File menu	Files'	Master menu
Transform menu	Transform'	Master menu
Overlap menu	Overlap'	Master menu
View menu	View'	Master menu
Style menu	Style'	Master menu
*Thickness menu	Shape Style'	Style menu
*Color control menu	Shape Style'	Style menu
*Line color menu	Color Style'	Style menu
*Area color menu	Color Style'	Style menu
*String rotation menu	Text Style'	Style menu
*Character rotation menu	Text Style'	Style menu
*Font menu	Text Style'	Style menu
*Anchoring menu	Text Style'	Style menu
*One-of-many menu		

User Messages

User messages (prompts, warnings, etc.) are displayed as a special menu containing the message in inverted video (white on black). Bugging this menu makes it disappear. It is always possible to continue running Griffin after the appearance of one of these messages.

Mouse Operations

Griffin uses each of the three mouse buttons for two different operations, depending on whether you hold down the SHIFT key while pressing the button. The following table summarizes the mouse operations:

Mouse button:	Left	Middle	Right
Without SHIFT key	Move object	Place control point	Select object
With SHIFT key	Copy object	Delete control point	Deselect object
Draw-through	--	All in rectangle (delete only)	Select/deselect all in rectangle
Objects affected	Shapes Captions Clusters Control objects (move only)	--	Shapes Captions Clusters

Selecting and Deselecting Objects

Many Griffin operations apply to one or more objects *selected* in the illustration. To select an object, bug it with the right mouse button; to deselect a selected object, bug it with the right button while holding down the SHIFT key. If the object is part of a cluster, the entire cluster is selected or deselected.

Drawing diagonally through a rectangle (button down, move, button up) selects/deselects all objects entirely contained in the rectangle. If any object in a cluster would be selected/deselected in such a draw-through operation, the entire cluster is selected/deselected.

Selection is indicated by a small square token displayed on each selected object. For shapes, the selection token is displayed at some point on the edge. For captions, it is displayed at the anchor point. If the object is completely obscured (either by overlying objects or by the selection token itself), the token appears as a slashed box. A selected cluster is indicated by a square token with a small letter 'c' in it, displayed on the topmost object (in overlap order) of the cluster.

Move and Copy

Any Griffin object may be moved by pressing down the left mouse button over any point on the object, moving the mouse, and releasing the button at the new location. (A menu must be moved by its *frame*, the grey border surrounding the text.) The object is repositioned so that the point on the object that was under the down stroke of the mouse is moved to the location of the up stroke. If the object is part of a cluster, the entire cluster is moved as a unit. Holding down the SHIFT key during this operation causes a *copy* of the object (shape, cluster, or caption) to be placed at the indicated position, leaving the original object undisturbed. The copy is placed at the top of the overlap order; if the original object was selected before the copy operation, the selection transfers to the new copy.

Placing and Deleting Control Points

Control points are displayed as small crosses. They are placed by clicking the middle mouse button

and deleted by clicking the same button while holding down the SHIFT key. Drawing through a rectangle with the shifted middle button deletes all control points in the rectangle. All control points on the screen are ordered in the sequence in which they were placed. Any new control point goes after all the previous ones, and there is no way to change the order. There is currently no good way to determine the order of the existing control points other than to plot a curve through them.

Creating and Modifying Objects

Shapes (graphical objects) are created and modified with the commands on the *edit* and *shape menus*, in conjunction with control points placed on the screen with the mouse. These menus are both controlled by the 'Edit' toggle on the master menu. Other attributes of the object to be created are specified on the various *style menus* (see 'Styles', below).

Drawing Shapes

To create a graphical object, place a series of control points in order with the middle mouse button, make sure the desired style attributes have been selected, then bug 'Draw' or 'Close', for an open or closed shape, respectively. The type of shape to be drawn (straight-line or curved) is specified with the 'Line' or 'Curve' option on the shape menu. [For the aficionado, the menu item 'Spline Type' expands a one-of-many menu of various methods of curve fitting. These methods are explained in greater detail in the section 'Curve Fitting' at the end of this document.] The *color control menu* (see below) determines whether a closed object will be filled, outlined, or both.

For linked objects with more than one link, place the control points for each link separately, select 'Line' or 'Curve' for that link, then bug 'Draw Link' for all but the last link. After placing the control points for the last link, bug 'Draw' or 'Close'. 'Back Link' is used to back up one link while editing.

The special-purpose 'Box' command on the edit menu allows you to draw rectangles easily by placing just two control points, representing the endpoints of the rectangle's diagonal.

Creating Captions

To create a caption, place a control point, then type the caption. After the first character is typed, the control point will turn into a blinking caret. The backspace key backs up one character; the 'backward' key (BW on the Alto II keyboard, top-blank on the Alto I keyboard) erases the last word of the caption; the DEL key erases the entire caption. Any mouse operation terminates the caption.

Captions are displayed on the screen using the information in the file Fonts.Widths. Each character is positioned near as possible to where it would go on the printed page. Therefore, the caption will be displayed such that the total length relative to other objects in the picture is correct when the picture is printed. However, the character spacings on the display will look funny because of the difference between the fonts used on the printer and the fonts used on the alto. The characters should look fine when printed.

Modifying Shapes and Captions

To modify an object, select it, then bug *Modify*' on the edit menu. If more than one object is selected, the topmost selected object will be the one modified.

For a shape object, *Modify*' erases the object (or the last link of a linked object) and leaves its control points visible. The current shape and color styles (see below) are set to the style of the object being modified. You may now add, delete, or move control points, just as though you were drawing the object for the first time.

When you modify a caption, a caret appears at its end, as though you had just typed it in. The current text styles (see below) are set to the style of the caption being modified. You can then edit the caption in the normal way.

Manipulating Objects

The operations described in this section are used for object-level manipulations within an illustration. All of these operations appear on the *object*, *view*, and *overlap menus*, which are controlled, respectively, by the *Objects*', *View*', and *Overlap*' toggles on the master menu.

Deleting Objects

The *Delete*' command on the object menu deletes all objects currently selected. The last deletion can be canceled by bugging *Undo*'.

Selecting/Deselecting the Entire Illustration

For convenience, every object in the illustration can be selected or deselected by bugging *Select All*' or *Deselect All*' on the object menu. These operations apply only to objects on the current view (see *Alternate View*', below).

Grid

A *control grid* is available to aid in precise placement of objects and control points. The display of the grid is controlled by the *Grid*' toggle on the view menu. The grid is displayed as a pair of horizontal and vertical axes, each of which can be moved independently. Whenever the grid is visible, all mouse operations (*Move*, *Copy*, *Place* and *Delete* control point), are rounded to the nearest grid point.

Refreshing the Display

Refresh' on the view menu erases and replots the display.

Clusters

A *cluster* is a group of objects (shapes and captions) that are moved, copied, transformed, selected, and deselected together. That is, any of these operations performed on one item in the cluster is automatically applied to the entire cluster. A selected cluster is indicated by a single select token; however, every object in the cluster is considered selected.

To create a cluster, select all the objects that are to be part of the cluster (possibly including other clusters), then bug Cluster' on the object menu. The user message New cluster made'' will appear. To unmake a cluster, select it and bug Uncluster'. The user message Cluster unmade'' will appear. There is only one level of clustering, so a cluster of clusters makes one large group. That is, Uncluster' applied to such a cluster results in a group of single objects, not a group of clusters.

Changing the Overlap Order

Objects have a definite order of overlap, whether or not they actually cover each other up. Objects are displayed and printed in bottom-to-top order. To change the position of an object in the overlap order, select the object and bug the desired item on the overlap menu: Top', Bottom', Up-One', or Down-One'. If more than one object is selected, they retain their relative positions as they move. For example, if five objects are selected and you bug Top', the selected objects will become the top five objects.

Alternate View

Griffin allows you to work on two separate *views* within the same illustration. In effect, you have two separate pages to work on. The view you are currently working on is called the *current view*; the other is the *alternate view*. For example, you might create the background for a picture and place it on the alternate view, keeping only the foreground visible on your working view. This would speed up editing of the foreground by simplifying refresh operations. When the background and foreground were both completed, you would *merge* them into a single illustration. As another example, the alternate view might be used for selectively merging pieces of two different files. This could be done by reading each file into a separate view, then transferring objects from one view to the other.

All operations work identically on both views. To change from one view to the other, bug Switch' on the view menu. This will erase the screen and display all objects on the other view. Control objects (menus, control points, and grid) are displayed in the same positions in either view.

To transfer objects from one view to the other, select the objects to be transferred and bug Transfer' on the object menu. The selected objects are erased from the screen and transferred to the alternate view. Bugging Merge' on the view menu causes all objects currently on the alternate view to be transferred to the current view and displayed on the screen.

Only objects that are visible on the current view are included when an illustration is written out on a press file. All objects on both views are included when the illustration is saved on a Griffin file.

Transforming Objects

Griffin offers a variety of *transformations* that may be applied to an object. All transformation operations appear on the *transform menu*, controlled by the 'Transform' toggle on the master menu. Only shape objects transform fully. For captions, only the anchor point is transformed

To transform one or more objects, select the objects to be transformed, place the transformation control points (details below under the various transformations), and bug the appropriate item on the transform menu. The selected object(s) are erased, transformed, and redisplayed. If 'Use Copy' is selected on the transform menu, the original object is left undisturbed and a *copy* is created, placed on top in the overlap order, transformed as specified, and displayed. If the requested transformation would flatten the object in such a way that its original shape could not be recovered (for example, scaling it by zero), a warning message is displayed and the transformation is not performed.

X or Y Scaling

For X or Y scaling (ScaleX', ScaleY'), two or three control points are required. The points are projected onto the specified dimension (the other dimension is ignored). If three points are placed, the object is transformed so that the distance from the first to the second control point along the specified dimension is scaled into the distance from the first to the third point. If only two points are placed, the overall size of the object in the specified dimension is scaled to the distance between the control points in that dimension. The first control point is used as the origin for the scaling operation.

Full Scaling

For full scaling (Scale'), two or three points must be placed. In effect, both X and Y scaling (see above) are done simultaneously with respect to these control points. For example, you can reduce the size of an object by a factor of 2 by placing the first two control points on a 45 degree diagonal, and the third on the same diagonal, midway between the first and the second. If only two control points are placed, they define the new bounding box of the object. The first control point is used as the origin for the scaling operation.

Rotation

Rotation (Rotate') requires two or three control points. The first point is the center of rotation. If three points are placed, the object is rotated so that the direction from the first to the second control point is mapped into the direction from the first to the third point. If only two points are placed, they represent an absolute angle relative to the horizontal.

Full Mapping (Scaling and Rotation)

For the full mapping transformation (Map'), two to six points may be placed. Two points specify a simple translation (move) to be applied to the selected object(s). Four points specify a mapping from the line defined by the first two points into the line defined by the third and fourth. Objects are scaled and rotated, but not deformed by this mapping. Three points may be used as a shorthand for the four-point map, in which the two lines are defined by the first and second points, and the first and third. For six points, the triangle defined by the first three points is mapped into the triangle defined by the second three. Five points are a shorthand for six points, with the two

triangles defined by points one-two-three and one-four-five.

Styles

Every object in a Griffin illustration has a *style* that defines various attributes of the object's appearance when displayed or printed. Styles are specified with the *style menu*, controlled by the 'Style' toggle on the master menu. The 'Color Style', 'Shape Style', and 'Text Style' toggles on the style menu in turn control the display of specialized *style submenus* for setting the style attributes of an object.

The style submenus 'remember' their settings, even when they are not currently displayed on the screen. The state of all these menus together defines the *current style*, which is applied to any newly created object. All objects carry all style attributes for example, every shape object has a font style. It is only possible to operate on the entire style for an object, not just one attribute. For instance, there is no way to change the color of several captions in one operation without also setting them all to the same font.

To read the style of an object, bug 'Indicate' on the style menu. The current style is set to the style of the selected object, and all visible style submenus are updated accordingly. If more than one object is selected, the style of the topmost selected object becomes the current style.

To change the style of an object, select the object, set the desired attributes on the style submenus, then bug 'Apply' on the style menu. All selected objects are given the current style, as defined by the current settings of the style submenus.

Shape Styles

Shape styles include line thickness and whether a closed object is filled, outlined, or both. They are specified with the *thickness menu* and the *color control menu*, both controlled by the 'Shape Style' toggle on the style menu. The line thickness (given in Alto pixels, where one pixel=32 micras) applies to all lines and outlines. All lines have round endpoints. Outlined objects are larger than unoutlined objects by half the thickness of the outline.

Color Styles

Every Griffin object has two color attributes, one for the outline and another (applicable only to closed shapes) for area fill. These attributes are specified with the *line color* and *area color menus*, both controlled by the 'Color Style' toggle on the style menu.

Griffin provides a 'palette' of colors that are intended to be generally useful for all types of illustration. A color reference sheet is available on the file [Ibis]<Griffin>ColorChart.press. It is recommended that you keep a hardcopy of this color chart near your workstation for reference while using Griffin. [There will be variations in the colors between Puffin and Pimlico printers, especially black, browns, and dark reds.] All colors except black and white display as half-tone grey patterns. These patterns were chosen to give some indication of the brightness of the color, but are not unique. For example, red and green display as the same grey pattern.

[Because of changes in the color chart since the first alpha-release, it is possible for a Griffin file to contain objects with color values that are not on the menu. These objects will display in a 50% grey pattern, and will continue to print in the same colors as when they were created. Indicate' applied to such an object will set the color menu to black, so the color cannot be copied.]

Text Styles

Text styles include font, anchoring position, and rotation. They are specified with the *font*, *anchoring*, *string rotation*, and *character rotation menus*, all of which are controlled by the 'Text Style' toggle on the style menu. Because text display is based on the hardcopy widths of the characters, the spacing of characters on the screen may appear a bit strange. However, the overall length of each caption is represented accurately, so that shapes and captions can be correctly aligned.

The font to be used for a caption is specified by selection on the font menu. This menu includes the names of all the fonts on your disk. To add to this list, simply load the desired strike fonts onto your disk and start Griffin. Griffin will use the information in the strike font plus the information in *fonts.widths* to make new entries in *Griffin.fonts*. There is currently no way to delete fonts other than deleting *Griffin.fonts* and starting over.

Captions may be *anchored* at the left, right, or center by bugging the appropriate option on the anchoring menu. The designated anchor point is positioned over the control point when the caption is typed in. When a caption is selected, the select token appears at the anchor point.

String rotation controls the direction in which the *caption itself* is oriented that is, the direction in which characters are added as text is typed from the keyboard. The items 0 d', 90 d', 180 d', and 270 d' on the string rotation menu represent normal (right), up, left, and down, respectively.

Character rotation controls the direction in which the *characters within the caption* are oriented. The options on the character rotation menu have the same meanings as the corresponding items on the string rotation menu. Character rotations may require rotating a font. This takes several minutes (up to a significant part of an hour for very large fonts). Once the font is rotated, it is stored in the file *Griffin.fonts*, where it can be reused. Thus it is only time-consuming to use a rotated font the first time.

Of the 16 possible combinations of string and character rotation, only half are useful in practice. The 4 cases where the string rotation is the same as the character rotation, and the 4 cases where the string rotation equals the character direction plus 270 (producing a vertical column of characters), produce readable text. The other 8 cases are not generally useful.

Filing

All filing operations are done with the *file menu*, controlled by the Files' toggle on the master menu. The file menu contains a list of file names, an empty slot, and the functions Press', Save', and Restore'. The list of file names includes all files on your disk with the extension .griffin'.

Creating a Press File

To make a press file of the current illustration, type the desired press file name (without an extension) into the empty slot on the file menu, or select one of the existing names on the menu; then bug Press'. The extension .press' is automatically appended to the name supplied, and a press file with that name is written containing the current illustration. This operation may take several minutes. Pictures with many lines and outlined areas will take the longest time to format.

The press file will contain only those objects that are visible on the current view when the file is made. This does not include runtime objects (control points, menus, or grid), objects not on the current view, or objects that are completely off the screen. Objects that are partially on the screen will not print correctly in this version of Griffin.

Saving a Griffin File

To save the current illustration in editable form on a Griffin file, type the desired Griffin file name (without an extension) into the empty slot on the file menu, or select one of the existing names on the menu; then bug Save'. The extension .griffin' is automatically appended to the name supplied, and a Griffin file with that name is written containing the current illustration. If a file of the specified name already exists, a menu is displayed asking for confirmation (Yes' or No'), before overwriting the file.

The Griffin file will contain *all* objects in the current illustration when the file is made. This does not include runtime objects (control points, menus, or grid), but *does* include objects not on the current view and objects that are completely off the screen. Only one illustration may be saved in each Griffin file.

Reading a Griffin File

To read in an illustration from a Griffin file for further editing, select the name of the desired file on the file menu; then bug Restore'. The extension .griffin' is automatically appended to the name selected, and the illustration stored on the Griffin file with that name is read in and displayed on the screen. Objects read from the Griffin file are *added to* (they do not *replace*) any objects already on the screen when the file is read.

If Griffin Gets Full

Griffin is a memory resident system. Even on the D-machines, it uses only as much memory as is available on an Alto. This means that there is a limit to the number of objects which can be in a Griffin picture. The number of objects depends on the type and complexity of the objects. As the system fills up, the response to mouse operations becomes slow. There seems to be a drastic slowdown just before the system becomes completely full. Once there is no more space to store objects the following message is displayed:

Griffin is out of space. Save your file!

It is strongly recommended that the illustration be filed right away, before any more operations are performed. Once the illustration is safely filed, the picture can be edited, although no more objects can be added. The error message will be displayed each time the system is unable to get enough space to perform an operation. In general, save your big files often. While every effort is made to give you a chance to save your work once Griffin fills up, it is not always possible.

If Griffin Breaks

When Griffin detects an error condition, it attempts to write a file `GriffinError.Griffin'` containing the picture you were working on at the time the error occurred. Griffin will then exit to the Alto Executive and print the following message:

// Griffin error. File GriffinError.Griffin written

Some errors may make it impossible to write the file, in which case Griffin will attempt to exit gracefully to the Alto Executive with the following message:

// Fatal Griffin error. No file could be written

There is always the possibility of something catastrophic happening, which could cause the system to freeze up (cursor does not track the mouse) or even hit the dreaded boot lights. If this occurs, the boot button is your only recourse. Do not confuse some of Griffin's long pauses with fatal errors. Working pauses are indicated by the busy cursor (that funny object is a griffin), which will still track if you move the mouse. Long pauses are caused by such operations as writing files (several minutes, and you should hear the disk), generating press files (several minutes, plus the screen may break up a bit), and rotating fonts (up to an hour for large fonts).

All errors, comments, criticism, plaudits, etc., should be directed to Stone.PA. Please include a pointer to `GriffinError.Griffin'` if you have one, and a description of what you were doing when the system broke.

Curve Fitting

This section describes the various curve fitting methods available in Griffin. In general, deciding what method of curve fitting to use is a matter of personal preference. It is not necessary to understand all five methods to use the illustrator. The default fitting method corresponds to the menu item `Natural: Arc Length'`, and is adequate for most pictures. This section tries to give some description and guidelines for using the various methods. It isn't really as hard as it sounds. Experimentation is strongly encouraged.

There are five methods of curve fitting available in Griffin. These include two types of *natural*

spline (arc length and unit length), which smoothly connect a set of control points; a non-interpolating (does not draw through the control points) spline called a *B-spline*; an interpolating but only first-order continuous (less smooth) curve called a *Catmull-Rom curve*; and a very different type of curve called a *Bezier curve*, which uses the control points as end points (on the curve) and tangent points (off the curve) in an alternating fashion. There is a figure on [Indigo] <Griffin>Release>SplineSampler.press which shows the five curve types. This figure must be printed on a press printer.

B-splines, Catmull-Rom curves, and Bezier curves are *local* curves. That is, a given section of the curve is affected only by the control points near that section. As a result, these curves are more symmetrical and more modifiable (since changes at one end don't affect the other end of a long curve) than natural splines. Each section of the curve is completely defined by four control points. For the B-spline and Catmull-Rom curves, moving one control point will affect four sections of the curve. For Bezier curves, moving one control point will affect only one or two sections of the curve.

Natural splines and Catmull-Rom curves are defined by placing control points along the desired curve. The algorithm will draw a curve through the points. Think of these points as "handles" to control the slant of the curve at the end conditions. (The Catmull-Rom curve does not go through the first and last control points; by placing double points at the beginning and the end, you can make the curve continue out to the endpoints.) The two types of natural spline generate similar-looking curves, especially if the control points are evenly spaced. In general, the unit mesh version will give sharper turns at the control points. The Catmull-Rom curves, in comparison, look very square. However, because they are local curves, they may be easier to use for some shapes.

The B-spline goes *near* all the control points but through none of them. It is defined by placing control points *near* the desired path; the curve follows the *direction* of the points, but does not pass through them. B-splines are local curves and are very smooth and symmetrical, but it is quite a bit harder to predict from the positions of the control points where the curve will fall. Placing the control points in some regular pattern using the grid will produce very attractive curves for borders or symmetrical shapes.

Each section of a Bezier curve is defined by the *end points* plus two points off the curve (*tangent points*). The control points must be placed in the following order: end point, tangent point, tangent point, end point. The line from the first end point to the first tangent point is tangent to the curve (describes the slope or slant) at the end point. Similarly, the line from the second tangent point to the second end point is tangent to the curve at the second end point. If more than four control points are placed, the first four define the first section of curve, the fourth point plus the next three define the second section of curve, and so on. The sections are connected, but not smoothly. To make a smooth connection between sections, it is necessary to line up the tangents of the two sections. That is, there must be three control points in a line at the join point. Bezier curves are useful for drawing a short section of curve when the tangents at the end conditions are important (for example, a curve that points horizontally from the side of a box into the side of another box). Bezier curves can also be used to make symmetrical or repeated shapes with sharp breaks in them.

Because Catmull-Rom curves and B-splines do not go through the first and last control points, these curves cannot be used in this version of Griffin to make objects with multiple links. Objects are also restricted to using only one method of curve fitting for the entire object. Thus it is illegal to change the curve-fitting method in the middle of a multiply-linked object.

Griffin Menu Summary

The following is a summary of all menus use in Griffin and their contents

Master menu always present on screen

Edit: Toggle display of edit and shape menus
 Objects: Toggle display of object menu
 Style: Toggle display of style menu
 Files: Toggle display of file menu
 Transform: Toggle display of transform menu
 Overlap: Toggle display of overlap menu
 View: Toggle display of view menu
 Quit: Exit from system [*displays confirmation menu*]

Edit menu controlled by Edit' toggle on master menu

Draw: Draw or complete open shape
 Close: Draw or complete closed shape
 Draw Link: Draw segment of linked object
 Modify: Edit object [*erases object, leaving control points visible*]
 Back Link: Edit last link [*erases link, leaving control points visible*]
 Box: Draw rectangle [*requires 2 control points*]
 Spline Type: Toggle display of spline type menu

Shape menu (one-of-many) controlled by Edit' toggle on master menu

Line: Draw object with straight lines
 Curve: Draw object with spline curves

Spline type menu (one-of-many) controlled by Spline Type' toggle on edit menu

Natural: Arc Length: Natural splines, arc length parameterized [*default*]
 Natural: Unit Mesh: Natural splines, unit mesh parameterized [*same as Draw*]
 Bezier Curve: 4 points determine a curve which goes through the first and fourth point
 B-Spline: 4 points determine a curve which does not interpolate any point. Cannot be used in linked objects.
 Catmull-Rom Curves: 4 points determine a curve which goes through the second and third points. Cannot be used in linked objects.

Object menu controlled by Objects' toggle on master menu

Delete: Deletes selected object(s)
 Undo: Cancel last deletion
 Select All: Select all objects
 Deselect All: Deselect all objects
 Cluster: Make selected objects into a cluster
 Uncluster: Break selected objects out of clusters
 Transfer: Move selected object(s) to alternate view

File menu controlled by Files' toggle on master menu

Press: Write press file
 Save File: Write Griffin file
 Restore: Read Griffin file
 Empty menu slot: [*Used for supplying new file names*]
 File name 1
 File name 2
 .
 . [*Names of all .griffin' files on disk*]
 .
 File name n

Transform menu controlled by Transform' toggle on master menu

Map: Map 1, 2, or 3 points into 1, 2, or 3 points.
 ScaleX: Scale in X dimension
 ScaleY: Scale in Y dimension
 Scale: Scale in both dimensions

Rotate: Rotate around a point

Use Copy: Apply transformation to copy [*leaves original object undisturbed*]

Overlap menu controlled by 'Overlap' toggle on master menu

Top: Move to top of overlap order

Bottom: Move to bottom of overlap order

Up One: Move up one position in overlap order

Down One: Move down one position in overlap order

View menu controlled by 'View' toggle on master menu

Refresh: Erase and repaint display

Grid: Toggle display of control grid

Merge: Merge all objects in alternate view into current view

Switch: Switch to alternate view

Style menu controlled by 'Style' toggle on master menu

Indicate: Set current style to style of selected object

Apply: Apply current style to selected object

Color Style: Toggle display of line and area color menus

Shape Style: Toggle display of thickness and color control menus

Text Style: Toggle display of font, anchoring, string rotation,
and character rotation menus

Thickness menu (one-of-many) controlled by 'Shape Style' toggle on style menu

1: Set line thickness to 1 Alto pixel

2: Set line thickness to 2 Alto pixels

3: Set line thickness to 3 Alto pixels

4: Set line thickness to 4 Alto pixels

Color control menu (one-of-many) controlled by 'Shape Style' toggle on style menu

Filled: Fill closed objects with current area color

Outlined: Outline closed objects with current line color

Both: Outline and fill closed objects with current line and area colors

Line color menu (one-of-many) controlled by 'Color Style' toggle on style menu

Lines: [*Inactive text*]

List of available color names

Area color menu (one-of-many) controlled by 'Color Style' toggle on style menu

Areas: [*Inactive text*]

List of available color names

Font menu (one-of-many) controlled by 'Text Style' toggle on style menu

Font name 1

Font name 2

.

. [*Names of all fonts in Griffin.fonts*]

.

Font name n

Anchoring menu (one-of-many) controlled by 'Text Style' toggle on style menu

Left: Anchor caption at left

Centered: Anchor caption at center

Right: Anchor caption at right

String rotation menu (one-of-many) controlled by 'Text Style' toggle on style menu

0 d: Caption reads from left to right

90 d: Caption reads from bottom to top

180 d: Caption reads from right to left
270 d: Caption reads from top to bottom

Character rotation menu (one-of-many) controlled by 'Text Style' toggle on style menu

0 char d: Orient characters normally
90 char d: Orient characters facing top of page
180 char d: Orient characters upside-down
270 char d: Orient characters facing bottom of page