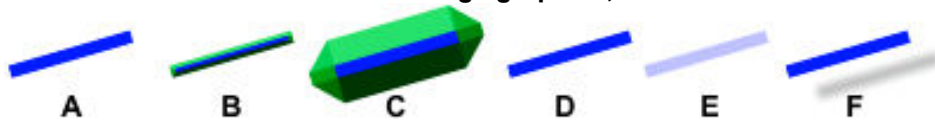


Tutorial: Gauge graphics, Part 2



Let's look at a line and see how it reacts to the same edits that were done to object borders in Part 1. The above diagram illustrates a blue line with a 2-pt width. Image **A** is the original line.

Line **B** shows an interior bevel. The beveled edge received a default color of white. This was changed to green to make it more easily seen. Had the bevel been moved further into the interior of this line, the blue color would have been covered entirely by the green.

Line **C** is the exterior version of the beveled edge. Lines can be beveled, borders cannot.

Line **D** had curvature applied, but it retained square corners. This edit is not applicable to lines.

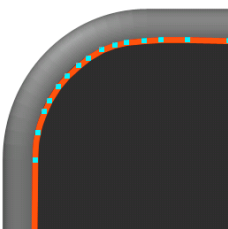
Line **E** illustrates a Flat transparency.

Line **F** shows a Wall shadow. This is perhaps the most useful of these six line techniques. The shadow can be moved as desired. It can also receive all shadow edits.

A trick using a wall shadow can sometimes be employed to cast a shadow on a gauge that would be impossible to achieve otherwise. The next series of illustrations will demonstrate this handy technique. The body of the demo gauge will be used as the example.

If you recall from the discussion in Part 1 when this body was developed, it was stated that no shadow was being applied next to the rim even though the rim was simulated to be higher than the rest of the gauge body. Had a wall shadow been used on this object, it would have been cast outside of the gauge.

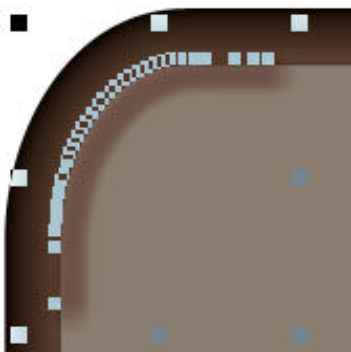
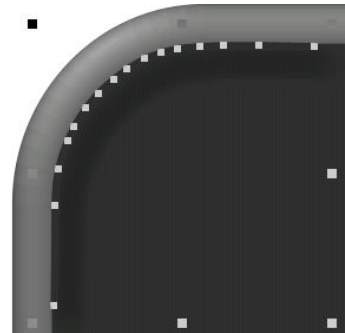
The rolled rim is the beveled edge of the original object, a rectangle. The rim cannot be made to produce a shadow independently of the object to which it belongs. Any shadow cast by the body would appear as shown at right. The Wall shadow would fall onto the panel of the aircraft, not the body of the gauge.



If a rim shadow had been desired, the following method would have been brought into play. A line made with the Pen Tool would have been placed directly against the rim. It would be positioned wherever a shadow was wanted.

At left, that line is shown in red for illustration. Once drawn and positioned properly, it could then be changed to the same color (80% black) as the body of the gauge. That would camouflage the line causing it to disappear from view. Though not visible, it would still retain its shadow-making property.

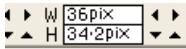
A Wall shadow could then be cast onto the body of the gauge as shown at right (look closely). The shadow could be edited for color, if needed, and shifted in position until it lay in a position directly against the rim.



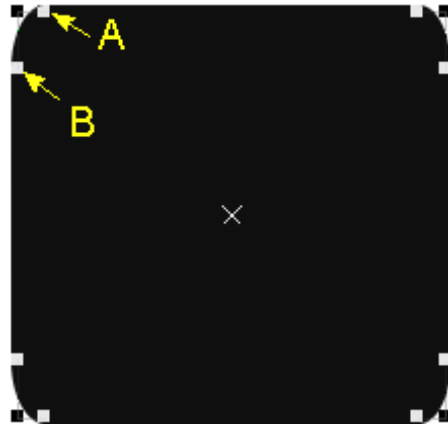
While the above example is not very effective because the dark color of the gauge conceals the shadow, this technique may have application on gauges that use lighter colors.

At left, a light brown gauge uses the same technique. The color of the line casting the Wall shadow closely matches the color of the rim making it hard to detect. But you can see the shape of this line from all the selection points. The shadow from this line falls next to the rim adding additional height. The shadow was also changed from the default black to a dark brown. A shadow such as this example is very effective in increasing the 3D effect of a gauge element.

Returning to the Heading Button



With the properties of lines and borders behind us, we can return to developing the Crosswind button. It begins by drawing a rectangle to a size that appears to be in proportion to the body of the gauge. At left, the rectangle can be seen. Black 90-percent is used and no border was required for this step. Let's take a closer look at the outline of the Heading Button. At right, the shape of the button is a black rectangle with curved corners. But look closely at the corners. They are elliptical arcs (part of an ellipse). Selection point **A** is much nearer the corner than point **B**. The result is a rounded rectangle with an appearance that differs with the rounded corners that were used on the body of the gauge. If you recall, that entity used curved corners that had true arcs that were circular (part of a circle).



Why were asymmetrical corners desired for the button and how were they achieved? They were designed to lend a bit of variety to the gauge. The buttons were not to look as though they were smaller, cloned copies of the gauge body. This concept is not a biggie as designing techniques go and most sim pilots would likely not notice. But, elliptical arcs do allow another editing technique to be explained. Curved corners and rectangles (or squares) relate something like this:

If a rectangle is drawn to size and a curvature is then applied to the corners, they will become rounded using circular arcs. The initial arc will use the default Curvature Ratio of 0.2.

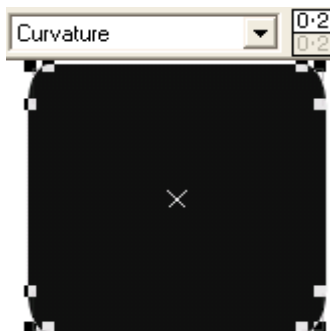
If the rectangle is then resized disproportionately (more in one direction than the other), the pair of arc selection handles in each corner will assume unequal distances from their corner. The following diagram will demonstrate this relationship.



Drawing 1 is the original rectangle.

Drawing 2 shows the default, rounded corners. At this step they are circular arcs.

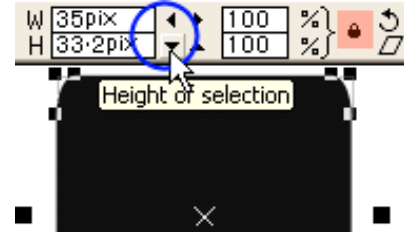
Drawing 3 contains the rectangle that has been extended in the Y-direction only. Now, the arc selection handles are located at different distances from the corners. The corners are no longer circular arcs but have become segments of an ellipse.



Using the asymmetrical technique just discussed, the Crosswind button is transformed as shown at left. Actually, this is not the button, but it will simulate the cutout in the body of the gauge through which the button will protrude. As such, the shape at left is made slightly larger than the button. It is the mother shape. All other elements for the Crosswind Button will either be derived from it or be stacked atop it.

The first element to be derived will be a smaller version of the black rectangle. This new element will form the main body of the button. It must be made smaller than the black rectangle, otherwise, the black cutout could not be seen.

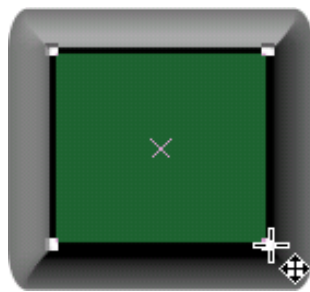
To produce a smaller version of the rectangle requires a few simple steps. First, the rectangle is selected, then a clone is made. The clone is then selected and reduced in size. That process is shown at right.



The dimensions of the new rectangle are only one-pixel smaller in both the X and Y directions. A click on the little arrows in the blue circle will accomplish that feat quickly. Just remember to unlock the "aspect ratio" inside the red area. Otherwise, a change in the X-value will automatically reset the Y-value to a proportional dimension calculated by XaraX. We want to force the proportions to the needs of the demo gauge. Those may not be in perfect symmetry.

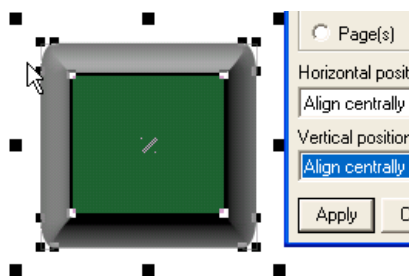


The drawing above shows the results of the next three steps. The color was changed to 50% black. The cloned rectangle was then given a Chiselled beveled edge. The bevel Size was adjusted to 5.3-pixels. That produced an acceptable height simulation to the button.



The next element that is needed is the unlighted, green face. It will sit atop the button giving the appearance that the button is not active. This element will be drawn by eye using the flat, top portion of the bevel as the guide.

At left, you can see the green rectangle. Note that this element will use a border. Here, the border (80% black) has the default width of .5-pts. That is too prominent and will be reduced to a value of .38, a value derived by eye. As stated earlier, the border will provide for a more distinctive visual divide between the green and the gray of the button. Perhaps other colors, such as a dark green, would also have worked.



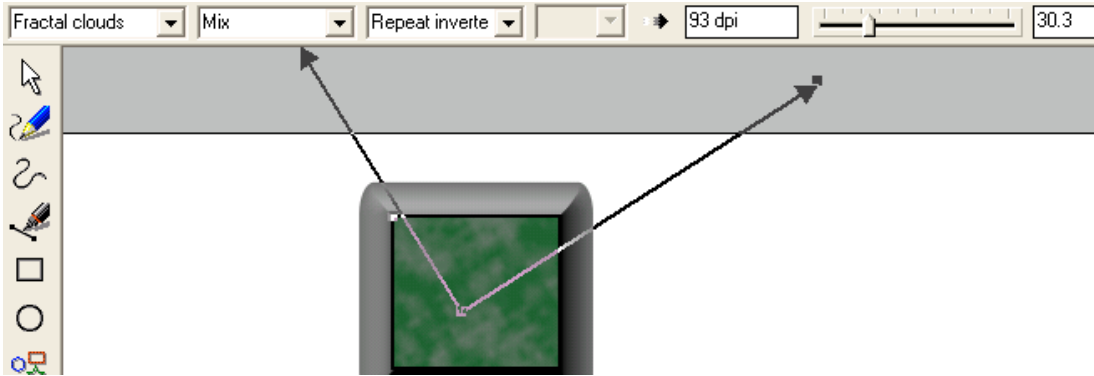
Once the green element is sized properly to fit evenly on the bevel, the two parts will be aligned. With both elements selected, they are stacked centrally in both the horizontal and vertical directions using the Object Alignment box. At left, the two elements have been centered.

Both elements must be de-selected so that only the green element can be selected for further modification. And what will that modification be?

The statement was made in the introduction to this tutorial that a "dirty" gauge was preferred over a pristine gauge. Now is the time for some mud slinging. The major reason for this is to impart a sense that the gauge was not manufactured yesterday. It has been in place on the panel for some time and gets used often enough to have received a smudge or two from a greasy finger. Perhaps a chip in the paint would be appropriate. There are many ways of aging a gauge to make it look "broken in". However, this effect can quickly get out of hand and grow cartoonish. A gauge should not look as if the pilot retrieved it from a Dumpster. At right, the button is chipped around the edge and the label has peeled off. Would you fly this airplane? That button may control the ejection seat. If so, you will quickly learn not to push unmarked buttons.



For the Pattern Gauge, the final size of the instrument will be small when installed into a panel. Any aging effects will probably not be individually recognizable. Instead, they each will contribute to the feeling that every control is not graphically perfect, that the gauge has accumulated some frequent flyer miles.



To provide the button with a bit of wear and tear, the Transparency Tool was summoned. The settings used and their effect are shown above. The two black arrows allow the blotchy pattern to be varied in size and direction. By editing this effect separately for each button, the clouds will vary in pattern and not look like copies of each other.

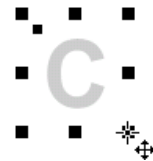
On the far right side of the above graphic, the transparency slider is set to 30.3-percent. This value allows the gray base color to show through the green. The transparency value that is set is based on one's judgment of what looks appropriate. The size of this gauge when installed into a panel will reduce the visual impact of this effect. It is suggested that effects such as this be viewed at reduced zoom so their appearance can be judged more accurately. A zoom of 100% is shown at left. Even that is quite a large size for a gauge control. Even so, the black border and the Fractal clouds have begun the process of dissolving into only a subconscious awareness.

The last element needed to complete the button in its OFF condition is the letter "C". That letter clearly identifies the button as the Crosswind heading. This single piece of text will occupy the very top of this control. Instead of delving into the Text Tool at this time, the letter C will be constructed and positioned on the button with only the essential details being described. Later in this tutorial, text will be covered in greater detail.



At left, the letter is shown atop the green rectangle. It was constructed using the Text Tool found at the left side of the XaraX screen. The font was Arial set to the Bold style. No border was used around the text object. After typing, the letter was sized appropriately to fit onto the face of the button.

Sizing is done by adjusting the selection handles just like other objects in XaraX. At right, the cursor is shown at the lower, right corner. Once sized correctly, the object color was changed from the default black to 20% black, or 80% white. A letter color of 100% white may not be suitable since it is subject to the direct touch of a pilot's finger. White graphics in real life would gradually be toned-down over time from lack of cleanliness. The same may or may not be true of other text on a gauge. They may be required to look worn, or not, depending upon the design of the gauge itself. That judgment must be made separately for each text on every gauge.



As a rule, I lean toward a pure white color for text that would be printed onto the body of a gauge. If a shadow falls over it, pure white will help simulate that effect better than would a darker text color. An example is seen at right.

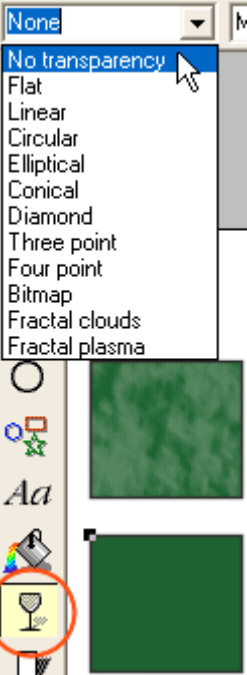
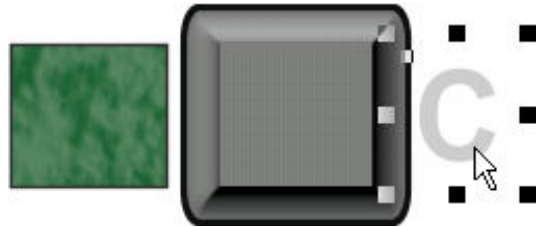
The last two letters of "Gain" lie partially within the shadow of the knob. The letters will enhance that shadow if transparency is applied to the entire word, more for those letters that lie in the shadow than those that do not (a Linear transparency).



After designing the letter "C" to suit the size of the button, it can then be aligned centrally on the stack using the Object Alignment box. The four elements (cutout, bevel, green face, letter) can be selected and grouped together. That will prevent any accidental movement that can occur unnoticed as other work is done in the same workspace. Grouping completes the button for the OFF state. After grouping, a clone is made of the group. This reproduces every element in that group. The cloned group will then be ungrouped to make available the various elements inside. Those will be used as elements for the ON state of the button. Of course, some will require modification to achieve the illuminated effect. That will be the next topic discussed.

After ungrouping, the top two elements are moved aside for further editing. The others (black cutout, gray bevel) are fine as is.

One additional element will be needed. That will be a clone of the green rectangle. It will occupy the third position in the drawing hierarchy. Why this is done will become clear as the elements are assembled. For a refresher showing all the elements for both states of this button, look back at page 14.

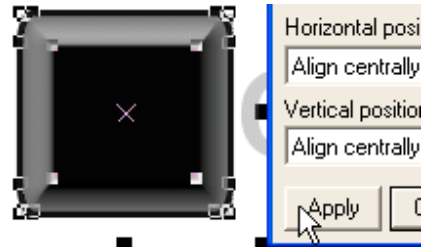


First, let's clone the green rectangle, remove the transparency, color it 90% black, then center it atop the gray button.

At left, the cloned green rectangle (lowest element in picture) was first selected. Then the Transparency Tool was chosen (inside red circle). These actions notified XaraX that the selected object is about to receive further work on its transparency, therefore, make all tools available.

Next, the Transparency shape inside the drop-down box is set to No transparency. That removed the Fractal clouds and reverted the rectangle back into its original color of green. The black border remains.

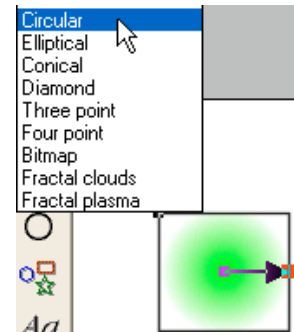
The third step is to exchange the green color of the clone for 90% black. The final step will center the black clone atop the gray button. These two steps are illustrated at right.



Work can now begin on the original green rectangle. The Fractal cloud transparency is also removed from that element. Doing so will overcome the quirk in XaraX that was mentioned earlier. After removing the transparency, a bright green color is applied that will simulate the ON state of the button. That color is shown at right.

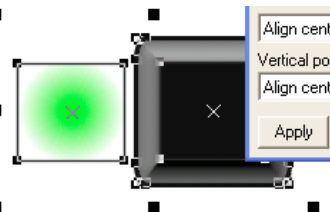


The green can now be given a Circular transparency to simulate the glow of an internal light bulb. That effect is seen at left. When this element is placed atop the black clone, the gradient of the transparency between the center of the rectangle and the border will allow the black to show. That contrast adds to the intensity of the glowing bright green color. No one will mistake whether this button is on or off.

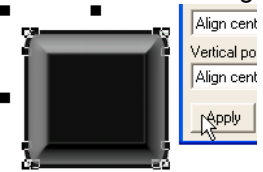


Finally, the green rectangle must be placed on and aligned to the other elements sitting on the "illuminated" stack.

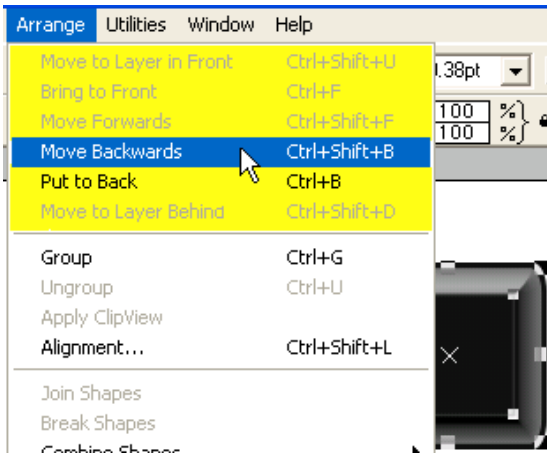
Using the Selection Arrow, a selection rectangle is drawn around all the elements that are to be centrally aligned. All those elements are shown in their



selected state in the illustration at right. A press of the "Apply" button will slide the green element onto the stack.



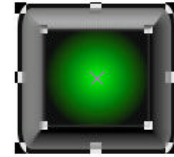
In the diagram at left, the aligned elements are shown positioned atop each other...whoops! What happened to the green rectangle? It must be buried somewhere within the stack of elements. How on earth can it be found and dug out? It's the black clone's fault. If you remember, it was created from the green rectangle. That gave it a higher priority in the draw order, so it now sits at the top of the stack. Logic would suggest that greenie is hiding directly under the black clone. What to do?



Tools that are found inside the Arrange menu can shuffle the drawing order. The area in yellow shows those aids. With the black rectangle selected, the only choices XaraX makes available are Move Backwards and Put to Back. The other choices are grayed because the black rectangle already occupies the Front (top) drawing order.

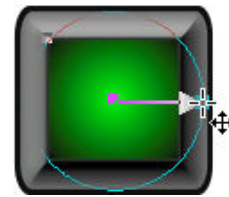
The black rectangle can be moved one layer down with each application of the Move Backward tool. In this case, one application will position it below the green rectangle.

At right, we can see the green rectangle has once again surfaced to the top of the



stack. We now can observe the full illuminated effect for the first time. This makes it possible to determine whether any further work is necessary. As is, it looks a bit too dark around the edges. Let's edit the transparency so that the glow extends further to the edge.

The green rectangle was selected and the Circular transparency was extended further from the center of the button. That increases the glow area and suits our purpose better. You can see that the glow at right has now assumed more realism.



The last element to be added to the button is the text. All elements are selected by drawing a selection rectangle (shown in red, at left) around all objects.

The Object Alignment box can then be used to position the letter centrally on top of the stack. That placement is shown at right. However, the gray text color looks inappropriate for an illuminated button. I prefer to make letter darken when lighted



from below. Instead of growing brighter from such a light source they would be thrown into silhouette.

The letter "C" is selected, then given an object color as shown in the last illustration at right. This is a dark green chosen from the color pallet.

With the text in place, all elements should be grouped together. This step completes the illuminated state of this button. However, we are not finished with the two Crosswind buttons. They will be used as "mother" buttons. They will beget all other heading buttons including the large U-R-F control. Creating a family of buttons will be discussed next.



Cloning buttons

Once a single heading button is finished, it becomes an easy task to produce the others. That is what cloning is all about. Of course, each button requires it's own individuality such as text identity. And the Fractal cloud transparency on each OFF button will be changed to lend visual variety. However, one rather major modification is required. That will be the largest heading button serving as the Final/Runway/Upwind control. It is much wider than the other buttons and contains three pieces of text.

As mentioned earlier, a cloned group contains all objects within that group. So, let's clone the OFF Crosswind button three times. The same will be done for the ON state of that button. Those clones will provide a basis for the Downwind, the Base, and the Upwind/Runway/Final buttons. For simplicity, only the Downwind button will be demonstrated. The Base button is fashioned using similar steps. So too, with the ON state of these controls. However, creating the larger Runway button will require editing in greater depth and that process will be illustrated.



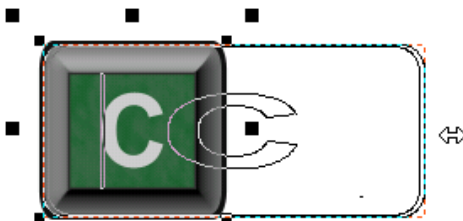
First, ungroup the stack to gain access to the text. Text is selected using the Text Tool shown at left. Swipe it across the letter to notify XaraX that additional work is to be performed on the text object. Then type the letter "D". The new letter will replace the old. However, not all letters are the same width. Larger letters, especially "M" and "W" have larger aspect ratios than other letters of the alphabet.

Because of this difference, it is necessary to re-align (horizontally, not vertically) the new letter. Once done, regroup the stack. This single editing step is performed on both the ON and OFF stacks. The final results are seen at right.



Making the largest heading button

This 3-in-1 heading button can also be made from the Crosswind button. The clone of the Crosswind button is stretched in the X-direction. This is done on all elements in the stack at the same time. How far to stretch? Until the width look appropriate for the room devoted to it on the body of the gauge. It can always be fine-tuned, if necessary, near the end of the design phase.



At left, the stretching operation is underway. All elements in the stack are receiving this edit including the text. You can see from the outline of the letter that the aspect ratio has changed because the letter now looks

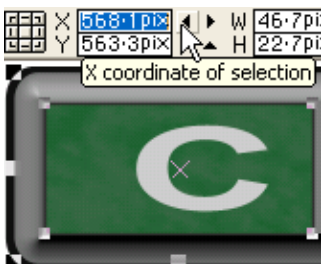
disproportionately wide.

All elements in this stretching action do not expand the same number of pixels. At right, you can see the original Crosswind button and the stretched version. Look closely at the black cutout surrounding both buttons. The one on the right is a bit wider. Also, the green



rectangle on the stretched graphic no longer covers the face of the button. What's the reason?

The percentage of stretching is the same for all elements on the button. In this case, the stretch increased the overall width of the button by about 88-percent. It grew from 36 pixels to 67.7 pixels in width, a gain of 31.7 pixels. The green rectangle also increased an identical 88-percent. It grew from 24.8 pixels to 46.7 pixels, an increase of only 21.9 pixels. That was not enough growth, pixel-wise, to keep the flat area on the button covered.

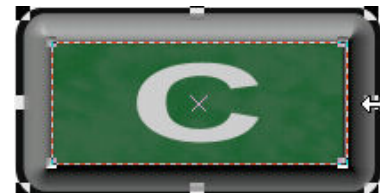


No problem, the bald spot is easily concealed. This is done by first selecting the button, then ungrouping the stack (if still grouped) to gain access to the text and the green rectangle. Both will require further modification.

After selecting the green rectangle, it is moved left until it is again even with the beveled edge of the button. That is shown at left. This process is best performed using the X-coordinate box at the top of the XaraX screen. Clicking the little black arrow or typing values into the window will move the green rectangle accurately. Using the mouse for this job is not suggested. That could introduce a change in the Y-value as well. No movement up/down is wanted.

At right, the green rectangle has been extended to the right until it now completely covers the gray area underneath.

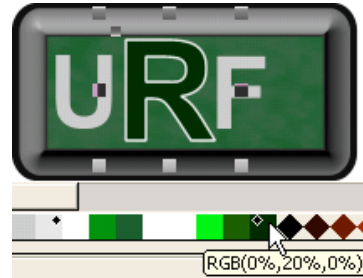
The text could be changed back to its original appearance by editing the aspect ratio using the tools available for text editing. But in this case, it is probably easier to delete it and clone the letter from another button. Three new letters will be needed to identify this button; U, R, and F. Whipping the three letters into shape will be our next task.





At left, the cloned letters have been edited to read correctly. The "R" was selected and increased in height and width to make it stand out from the other two letters. Its prominence signals that this button is an input button that must be activated before the gauge will accept runway heading data. Not only will the letter be more prominent, but the color parameters for that piece of text will also be changed.

That edit can be seen at right. The letter was given a gray border having a width of .38-pts, plus it received the dark green color seen in the pallet. At this point the three letters are randomly placed. They need an adjustment in both the horizontal and vertical directions.



This will be a three-step process. First, every letter on the button was selected and centered vertically. The centering effect is seen below in the image on the left.

Then, the letter "U" was moved until the green space to the left of that letter equaled the space above and below the letter. This is done by eye. The letter "F" also is adjusted in a similar way. The center image



shows this horizontal shifting of both end letters.

Step 3 will require all three letters to be selected. Hold down the SHIFT key for multiple selections. Using the

Object Alignment box shown in the illustration above, the Distribute equi-spaced alignment is chosen. This tool will even the distance between all selected objects using the outer two objects as the boundary. The result of equal spacing is shown at right. XaraX is very helpful when multiple objects must be formatted equal distance in either a horizontal or vertical arrangement.



The black cutout surrounding the button will not be modified even though it is a bit larger than the original one-pixel measurement used for all the other buttons. The appearance of a slightly larger cutout will work for this control because the button itself is larger.

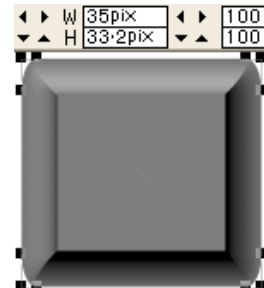
When all heading buttons in both the ON and OFF state are finished, attention can shift to the Turn Buttons shown below. Their construction will be detailed and new techniques introduced. The major differences between these buttons and the Crosswind are their custom shape. They are two separate objects, not just a dark line drawn to divide a single object. Doing that would have produced a similar effect and would have been much simpler. But it would have prevented a very helpful technique from being demonstrated. While not necessary for the Turn Buttons, this technique may become an important tool in your gauge design arsenal.



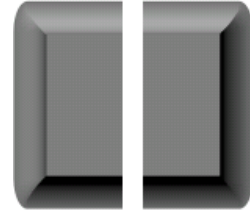
Turn Buttons

The design for these buttons requires that both elements, when mounted side-by-side, share the same footprint as other buttons. A clone of the Crosswind button can therefore initiate the process.

At right, the clone was ungrouped so all elements except the bevel could be deleted. That lone item is shown at right. Inside the data windows above the bevel, the size is shown. The width value (35) is the number of greatest significance and will be used for a bit of simple math. The height value shown in the window will also be used, but for now, it is only noted for reference. The W (width) shows the number of pixels in the X-direction. The H (height) is the number of pixels in the Y-direction.



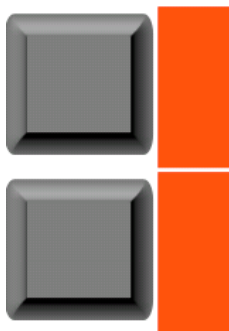
The game plan is to divide the cloned bevel from top to bottom near its center, so that it looks similar to the illustration at right. The split will not take place at the exact center because some visual separation must be retained between the two buttons. As mentioned, their combined footprint should be no wider than the Crosswind Heading Button (35-pixels).



Splitting will create two separate turn buttons, one for the left, one for the right. Those halves will be developed further in order to make simulations for the illuminated and non-illuminated states.

The button will be divided using the Subtract Shapes tool inside the Arrange/Combine Shapes menu. This tool deletes the subtracted portion of the object leaving a new shape. Because the beveled edge in the two halves illustrated above have different variations in shading around the edge, two clones of the gray bevel will be needed. Both will be split in such a way that each will delete a different half. This will preserve the continuity of shading.

To subtract shapes, two objects are needed. We already have the beveled portion of the button, so let's construct the second shape, a red rectangle. The subtraction object can be made any color, with or without a border. Those have no effect on the subtraction process.



What is most important is the W-dimension for the red rectangle. It must be exactly one-half the W-dimension of the bevel. As shown earlier, that element measures 35 pixels in width. Half of that value is 17.5. XaraX handles this division easily because it works in tenths of a pixel.

For the H-dimension, the rectangle should be a few pixels longer than the bevel, which measured 33.2-pixels. For this subtraction, I will construct a red rectangle measuring 17.5 x 38 pixels. At left, the bevels and rectangles are ready to be positioned in preparation for the subtraction process. The top pair will produce the Left Turn button; the bottom pair will create the Right Turn button.

To position the bevel and matching rectangle so that the splitting operation produces the desired result requires aligning either the left or right sides of each pair of object, depending on which half you want to keep.

Tip: When using the Subtract Shapes tool, the drawing order of the two shapes determines which shape is the subtracting shape. The shape on top will be the one that is subtracted and the one that will be deleted.

At right, the previous tip is demonstrated.

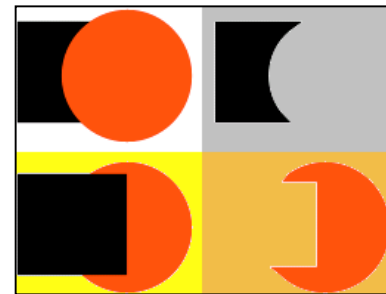
White: The red circle occupies the top position.

Gray: The overlap of the circle was subtracted from the square, then vanished leaving an object with a new shape.

Yellow: The black square holds the highest drawing order.

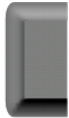
Gold: The overlap of the square was subtracted from the oval, then disappeared leaving a wrench shape.

With the above property in mind, we can align each bevel with their splitting rectangle and adjust the drawing order as necessary. The Object Alignment will be used for this task.



At left, the top pair was positioned using the right side of both objects as the alignment point. That places the left side of the rectangle directly along the center of the bevel. If you remember, we want some separation between the two halves of the Turn Buttons. To achieve this, the red rectangle is selected, then shifted one pixel to the left using the arrow key on the keyboard. That movement will decrease the width of the final object by a pixel. After subtraction the button will measure $(35 / 2) - 1 = 16.5$ pixels in width.

The bottom pair of elements was aligned using the left side of each object. That locates the right side of the rectangle directly over the center of the bevel. It will be shifted one pixel to the right to gain a total separation between the two buttons of two pixels. Their combined footprint will then equal 35 pixels; $(16.5 + 16.5 + 2)$.

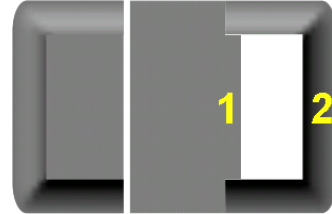


The result of the subtraction process is shown at left. The bevels are now separated into two objects of the appropriate width. When the final placement on the gauge takes place, they will be juxtaposed with the two-pixel separation.

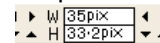


The objects at left look like simple beveled elements that were cut down the middle. But if you select one of the buttons, you would see in the Arrange menu that it is actually a grouped object. Subtraction produced the grouping automatically. A grouping of what, you may ask?

To answer that question, let's ungroup the Right Turn button and see what individual parts may lie inside. At right, the illustration shows that the button consisted of two parts. They are a gray, flat base (part 1) with a beveled edge (part 2) on top. Both turn buttons must be ungrouped for the next step. That will make part 1 accessible for additional editing.



The gray base (part 1) is selected and set to 90% black. That gives the button a two-tone effect just like the Crosswind button. The black area will provide the dark, underlying layer for the face of the button in the same manner as used on the Crosswind Button. Except, in this case, the turn buttons will be blue.

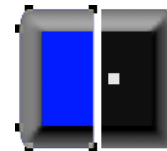


The black color has been applied to both buttons in the graphic at right. By selecting both buttons, their combined width will show in the data window. If needed, one of the buttons can be moved so the W-value reads 35 pixels.



The two black bases (part 1) can also be cloned in preparation for designing the face of each button. After cloning, a suitable blue color is then applied to the clone. That task is shown at left. The blue clone obviously occupies the top spot in the drawing order. That is why you cannot see the objects below it. While it is still selected, let's tell it where to go.

A single application of the Move Backwards tool was required to position the blue face as shown at right. It now lies below the beveled edge, but above the black base of the button. Just the right spot to work its magic.



With 50% Flat transparency, the blue will lose a bit of its vibrancy while still giving some indication of color. That condition will simulate the button when in the OFF mode. The two buttons with applied transparency are shown at left. They do need text labels and an arrow. The text will not be demonstrated because that technique is a repeat of the text edit on the Crosswind Button. However, the construction of the arrow will be discussed.

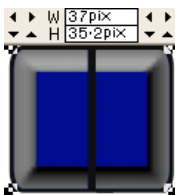
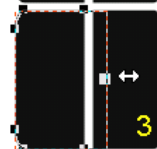
Before illustrating how to make an arrow in XaraX, let's finish up the two buttons by placing a black cutout around the pair. That will blank out the white space between them. What is the best way to create the cutout so its shape parallels the edges of both buttons? If you answered cloning, then give yourself an A+.



In drawing 1 at right, the black base on both buttons was selected, then cloned. That produced the two elements in drawing 2. To cover the white space between them, the left element was selected. The handle on the right side of that selection was pulled to stretch the element until it covered the white space. That process is shown underway in drawing 3.



This modification effectively covered the white gap as shown at left. Both elements were then grouped so they can be more easily handled as a single element. The cutout will be resized so that a one-pixel band will show around both Turn buttons.

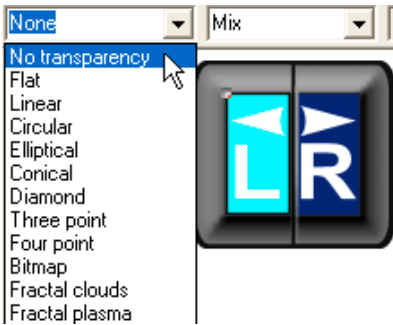


At bottom left, the black cutout was stacked behind the two Turn Buttons using the Put to Back tool inside the Arrange menu. While still selected, the values inside the data windows were change to those shown here. That increased the cutout size so that a 1-pixel band surrounds both buttons. You can see the cutout is now 37-pixels wide and 35.2-pixels high.

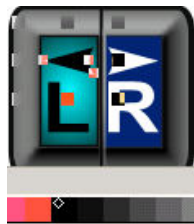
Both buttons should be grouped together, even without the text. That will allow them to be aligned centrally above the black cutout (which is also a group).

Illuminating the buttons is the final step. For that, I will pretend the text and arrows have already been placed on the face of each button.

Illuminating these buttons will proceed in a similar way to that used on the Crosswind control. The basis for each control is cloned copies of the non-illuminated Turn Buttons. The dark blue face of each clone will be selected and turned to a lighter shade of blue. The color chosen for this effect is cyan. It can be seen in the graphic at right.



To achieve the full vibrancy of this color, the Flat transparency must first be removed. That effect can be seen at left. The illumination from an interior bulb is then achieved by applying a new type of transparency set at 100-percent. That will be the Elliptical type which will match the elongated shape of each button. That effect can be seen at right. One last step is needed; to alter the white of the text and arrow.



That is a two-step process. The first action will be to select the text and arrow, then turn them 100% black. That process is underway at left. The pure black will form the foundation for a transparency. If you recall, bright colors can be toned-down by placing a black element directly under them. When a bright color is then given transparency, the black will penetrate the color causing it to grow darker.

Next, each of the black elements is cloned. The clones will occupy the top of the stack. As such, they can easily be selected by clicking them with the Selection Arrow. After selecting, the clones are given a blue color. That step is seen at right. The bright blue needs to be darkened a shade or two. After all, the effect desired is a simulated silhouette of the text and arrow but without losing all the blue color.



The text and arrow will receive a Flat transparency type. The percentage of transparency was adjusted until the elements looked "right". The illustration at left uses a percentage of 30.

One element perhaps not readily apparent to the eye rests on top of the Right Turn Button. Although small, it contributes to the overall realism. To see this element better, the illustration at right was zoomed.

The red arrow points to the beginning of a gray line drawn from one beveled edge to another. It is about one-pixel in width. The gray line adds a highlight separating the edge of that button from the shadow area that lies between the two buttons.

The gray color was chosen to match the highlight area at the top of the bevel because that is where the line begins. To tone-down the gray as it reached the shadows along the bottom of the button, it was given a Linear transparency. This allowed the line to blend in more evenly with those shadows, though not perfectly.

Now that you are aware of this line, if you take another look at the graphic above, the line will be seen more easily. This is a perfect example of adding effects that do not beg for attention, yet they enhance the overall feeling of reality.

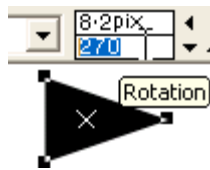
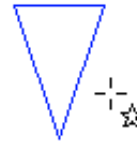
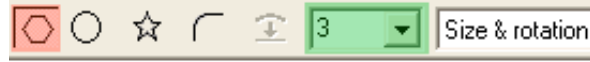
How are the directional arrows on each button produced? That topic was promised earlier and now will be delivered. In doing so, more experience will be gained in using the QuickShape Tool.



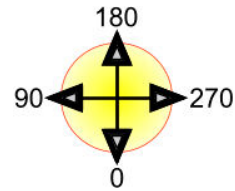


Let's look closely at one of the directional arrows. The shape at left is partly a product of the QuickShape Tool found in the toolbar along the left side of the XaraX screen. A basic triangular shape was drawn, then rotated to face to the right. It was edited to a size that fit the Right Turn Button. Color was applied to the arrow before creating a pleasing curve on the end. But, let's back up and go through these steps in more detail.

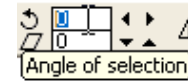
At right, the QuickShape info bar shows the choices that were used to construct the initial arrow. The Create polygon button (in red) is pressed and the the number of sides (green area) is set to 3. The triangle was drawn by the mouse into the shape shown at right. After releasing the mouse button, the triangle filled with black. A border is not needed around this object. Also, by default, the arrow is drawn pointing downward, a zero-degrees orientation.



To rotate it 90-degrees so that it points to the right, a value of 270 is typed into the Rotation window seen at left. A value of -90 also works. A compass rose for rotational values in XaraX uses the angle layout shown at right. These values are in effect ONLY when the QuickShape Tool is active.



Tip: The angle of rotation is not to be confused with another angular setting, the Angle of selection illustrated at right. Input into that window will rotate the arrow relative to its current position. That position is always zero-degrees. Positive values rotate the selection counterclockwise, negative values rotate clockwise. For example, if a value of 270 were entered, the arrow at right would turn counterclockwise 270-degrees. It would end up pointing left. These properties are in effect ONLY when the Selection Arrow is active.



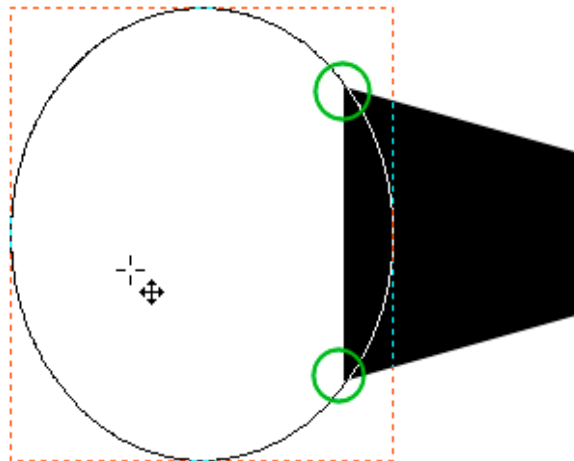
The color of the arrow for this demonstration will be left at default black for ease of viewing. The last step in arrow creation will be to change the tail-end of the arrow into a concave curve. This is not an essential edit, it just adds a bit of directional strength to the object.

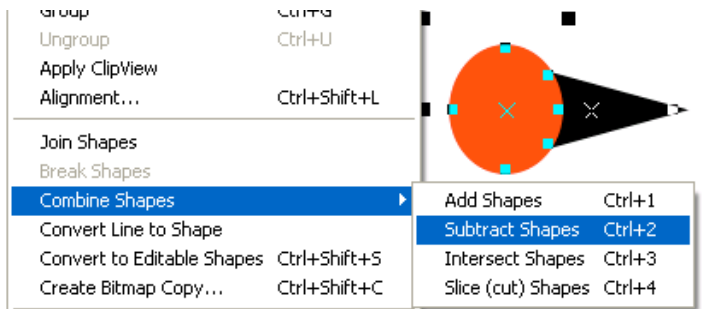
The tool used to make the curve will be the Subtract Shapes tool inside the Arrange/Combine Shapes menu. Since that tool was used a short time ago, your familiarity with it may give you an idea of what the subtracting shape should be. An oval will work quite nicely.

As you have learned, to subtract shapes you must have two shapes with which to work. The first shape will be the black arrow. The subtracted shape will be an oval of appropriate size. What size is that? The magnified drawing below will make that clear.

The oval was drawn using a radius that would form a curve which coincided with the two end corners of the arrow (inside green circles). The oval extended far enough into the arrow to produce a visible curved effect after the subtraction process. Rough alignment between both objects is done by eye, then fine tuned using the X and Y coordinate windows. Because values can be entered into those windows to the nearest 1/10th pixel, placement can be very accurate.

The black oval outline seen at right is the way that XaraX draws objects whenever they are moving. That speeds up the redraw process by temporarily eliminating the unnecessary fill color.





With both objects selected and the subtracting element occupying the top-draw position, the shapes are subtracted. The resulting arrow is shown at right.



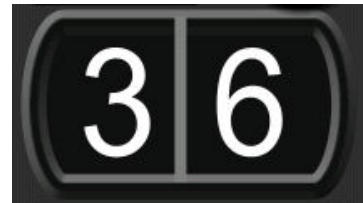
To make the left-pointing arrow, a clone is first made, then flipped horizontally.

The Turn Buttons will not

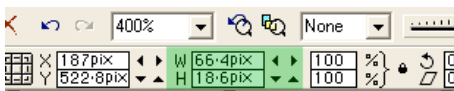
be given any aging or blemishes. They are too small to effectively use such effects. At this juncture, the Heading Buttons are complete.

Heading window

This window serves a dual roll. It both receives and displays information. The information is in the form of numerical digits. These are not XaraX graphics; instead, they are generated by XML. The graphics discussed here will be the rim outlining the window and the divider between the digits. The divider breaks the rectangular shape into two separate windows so they can receive heading information independently of each other. The color chosen for the digits was pure white. For maximum readability, the background of the window was set at 90% black.



Before beginning the development of this element, the approximate amount of room available on the body of the gauge was determined. That was done by drawing a trial rectangle. This rectangle was adjusted in size until it fit within the controls that would surround it. This gave an idea of what the size of the window would be so the next edit could be applied effectively. The trial rectangle measured about 66 x 37 pixels in size. This dimension is the initial working size.



An edited version of the trial rectangle is shown at left. The height has been reduced in half to 18 pixels. Those parameters can be seen in the data windows. Why was the rectangle reduced in height?

The reason was to attain asymmetrical curvature for the corners. When the rectangle is given curved corners, then adjusted back to the original height (37-pixels), the curves will no longer retain their default circular arc. Let me demonstrate.

At right, the rectangle was selected and given a Curvature. The Primary curvature ratio was changed from the default 0.2 to the .65 value shown in the data window. This increased the arc at the corners until they produced circular ends on the rectangle. The curves shown here are true arcs.



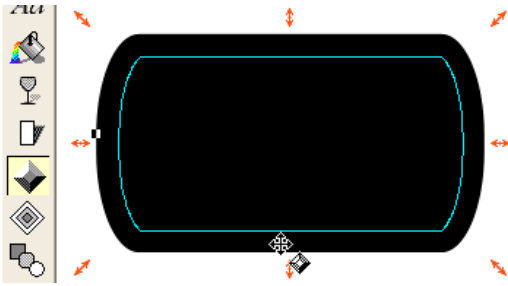
When the rectangle is edited back to the original height (37 pixels) the arcs will assume

asymmetry producing the graphic below.



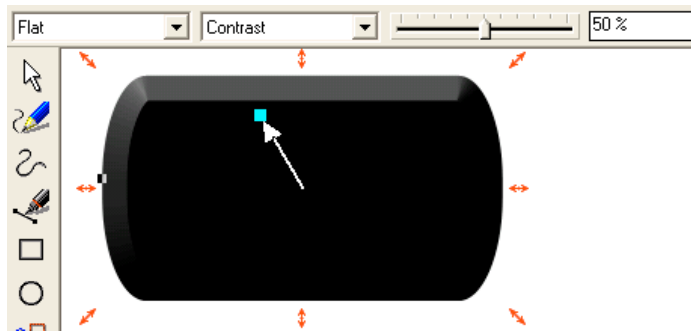
This is the outline I wanted for the window. Why not just a rectangular window with square corners, or slightly rounded corners? Visual variation and interest is the best answer I can think of.

The black rectangle at left will form the base for the window. It must be altered from a flat, lifeless ink spot into metal that reflects light and simulates depth. It needs a rim job. To edit the rectangle, the Bevel Tool will be used. An inside bevel will produce the 3-D effect without changing the footprint of the rectangle.



With the rectangle selected and the Bevel Tool active, the red arrows were pulled toward the center of the bevel. The cyan outline shows the width of the bevel during this process. When the width looks about right, releasing the mouse button will terminate the editing process.

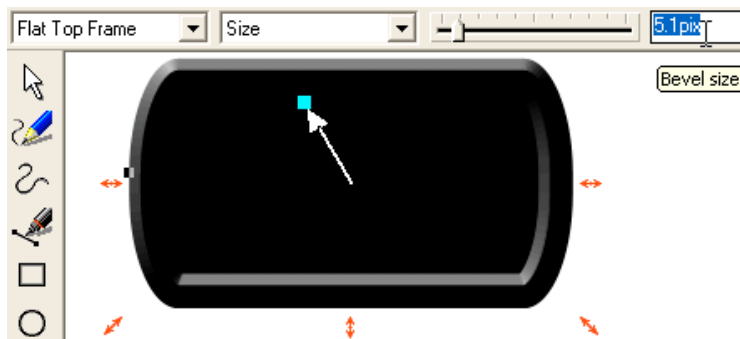
Upon release, XaraX applies the beveled effect as shown in the next illustration. The Flat bevel with a 50% Contrast are the default values. A Flat bevel always raises the surface of the object.



A sunken appearance is the desired effect. The bevel type must be changed to achieve that result.

Tip: The Bevel Tool must be active to see the parameter information for a selected bevel (type, contrast, etc.).

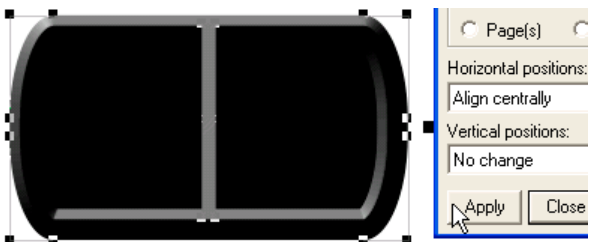
If the Selection Arrow is active, the information displayed at the top of the screen will change to show the parameters of the selected object, such as size, screen coordinates, etc.



The bevel has been changed to Flat Top Frame in the illustration at left. The bevel size was adjusted slightly to a width of 5.1 pixels. The Contrast was also raised from the 50% default value to 77%. This produced a more pronounced contrast between the rim of the window and the background. This gave a better

visual separation in the region at the lower, right side of the window. The last element needed to complete the window is a divider. This will have the effect of making two smaller windows inside the rim. A thin, vertical rectangle is used to make the divider, but a line would also work.

At right, this element can be seen. It contains the initial dimensions shown in the two data windows. The color chosen was a custom gray that matched the gray highlight showing along the raised rim. The divider will begin and end in those gray areas so the height may also need to be edited to span from the top rim to the lower rim.



To place the divider in the exact center of the window, the Object Alignment box is once again called into use. The result of that action is shown here. No shadow is applied to the divider because it would be ineffective against the dark background. When the XML code is running, the digits will display

centered within each window at a size that is easily readable but will not crowd the rim.

This completes the Pattern Gauge Heading Window.

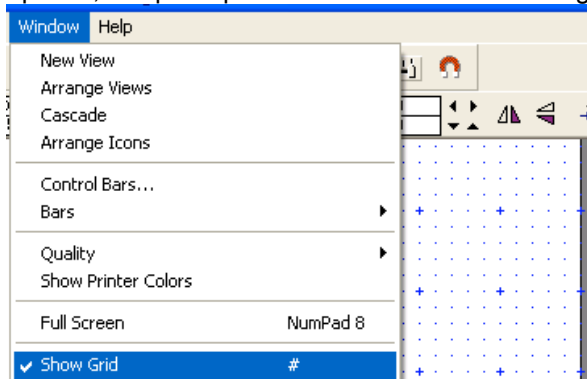
Assembling all the elements

When this phase of gauge design begins, the fun stuff turns deadly fun. You finally get to see all elements coming together for the first time. The carefully conceived layout will now prove your genius or that noun your wife uses. Assembly is the time to finalize all individual elements and to decide their placement on a gauge.

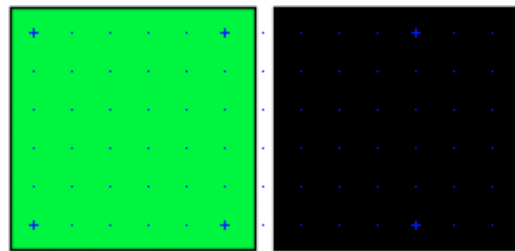
In some ways, the tedious work is now beginning. With all controls in place, changes will surely suggest themselves. This may be a simple altering the size of a control to better fit the allotted space or adjusting a troublesome shadow. Sometimes, color choices will change. But, all this is normal. If the change requires a major revision of an element, denial will likely set in, at least temporarily. In such situations, a deep breath, and a final push is the only course of action.

A large part of assembly is alignment and sizing. Buttons and such must be grouped and fit together with some semblance of order. No one wants to place their simulator lives in the hands of a gauge that looks to be assembled by a drunken sailor. Consequently, alignment techniques will play a large roll in this section of the tutorial.

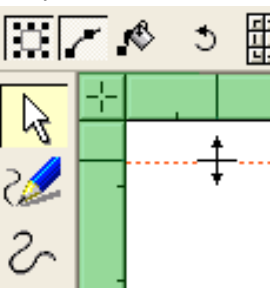
XaraX has many features that assist in alignment. One has already been used several times for various elements of the demo gauge. That is the Object Alignment box. With its many options, it is perhaps the most valuable of the alignment tools, at least, in the initial design phase.



Other aids are a grid that is available for the XaraX workspace. This dot pattern is turned on/off in the Windows menu. The grid is more easily seen in lighter colors versus dark as shown below.

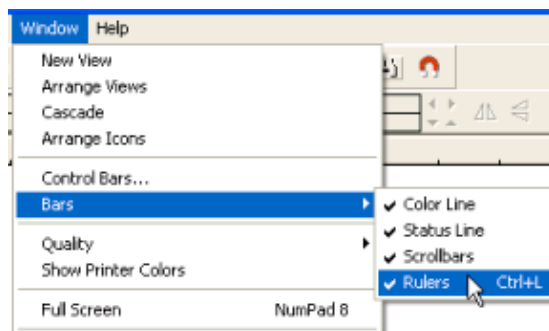


Snaps to this grid and to objects are also available in the Window menu. This feature is helpful at times to assist in the accurate placement of objects.



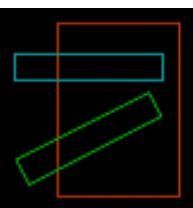
Another very helpful alignment tool is called Guides. These are dotted lines, usually red, that run vertically or horizontally across the workspace. A horizontal guide is shown at left. These entities occupy their own layer, so as not to interfere with drawn objects. Guides can be moved about with the Selection Arrow or have a value typed into their Properties box. They also can use Snaps.

To create a guide requires that the Rulers be active. They are activated in the Windows menu as seen in the graphic below.

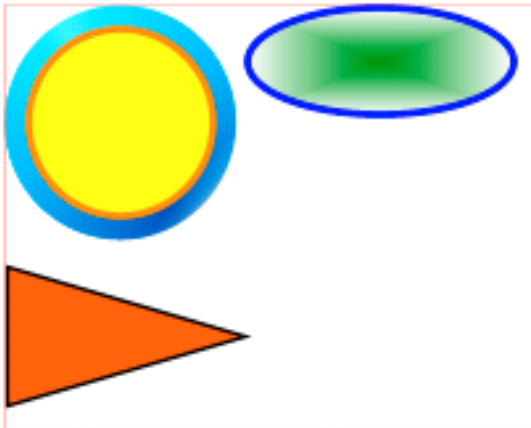


These bars (green areas, left) run along the top and left edges of the workspace. Guides can be pulled "off" the ruler bars in a way that is similar to the alignment guides in Paint Shop Pro.

A technique also frequently used employs a tool I call an Alignment Rectangle or "A-Rex".



This is a constructed rectangle having a transparent interior and a thin (0.1 pixel) border. The border color, usually a bright red, green, or blue, is chosen so that it can be easily seen during an alignment process. Objects are aligned along the straight edges of the A-Rex. They can be sized and rotated as needed.



The thin sides of an A-Rex make perfect reference points when aligning objects vertically or horizontally. The arrangement at left show the circle and triangle aligned vertically. The circle and oval are aligned horizontally. This example uses one A-Rex, but multiple Alignment Rectangles are often in place at the same time during assembly.

Most A-Rex alignments are performed by eye. But, for minute adjustments, coordinate values for an object can be entered in the appropriate data window. For unusual alignments, A-Rexes can be circular or other shapes.

While an A-Rex will permit path-alignment of multiple objects, they can also serve as

spacers. Because XaraX allows the sizing of a rectangle so accurately, objects place against opposite sides can be spaced at distances to the nearest 1/10th pixel.



The canvas on which the various elements of the Pattern Gauge will be assembled is the body of the gauge, at left. All control buttons and the display window must fit within this area in some logical manner without overcrowding. Overcrowding only contributes to a decrease in visibility and usability of the various controls.

The Power Button (grouped) will begin the assembly process. It is placed in the upper right corner of the gauge. As you can see at right, an A-Rex was constructed to act as a spacer between the right rim and the

button (the body of the button, not its shadow). A clone of this spacer will allow comparison on the opposite side of the gauge for the controls that will be placed there.

As evident in this illustration, extra space was allowed above the button because text will eventually occupy that area. How much space? That answer is not known at this time; therefore, all assembly is just a "trial fit" for now. Adjustments will be made as necessary before the process is completed.



Located on the top row with the Power Button are the two Turn Buttons, also grouped together as one unit. Every object along this row will be aligned using their bottom edge. This can be done using another A-Rex, or a Guide as seen in the illustration below. Not recommended for

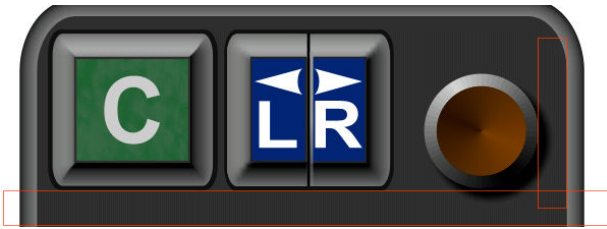
these alignments would be the Object Alignment box.

If that tool were used for this purpose, the Power Button would be aligned improperly. The reason for this has to do with the shadow that was applied to that object. The Object Alignment tool considers a shadow as part of the object. That tool would align the lower edge of the shadow with the bottom of the Turn Buttons. That would cause the body of the Power Button to look higher than would be appropriate.



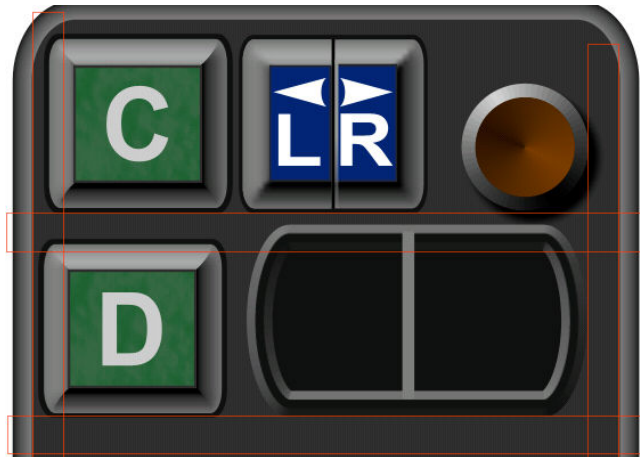
The Turn Buttons have a one-pixel cutout around the entire control. That is the element that was used to position the red Guide.

this process is over. Jot down the parameters of a shadow before deletion, and use the same values to reapply it after the alignment step.



Continuing, the Crosswind Button is placed in the top, left corner of the gauge. In the example at left, an A-Rex was used for horizontal reference. They are drawn wider than the gauge body to make them more easily selected if they need to be moved.

At right, the second row containing the Downwind Button and the Heading Window have been added. Those are aligned along their bottom using another A-Rex. The window was space away from the rim of the gauge using the same A-Rex as the Power Button. A clone of that spacer has also been positioned on the left side of the gauge to check the balance along that edge. As you can see, additional adjustments will be needed to achieve equal space along that side.



In the illustration at left, the remaining two buttons have been located on the gauge. These clearly show overcrowding and do not clear the rim.

Now that all elements are on stage, we can begin the process of blocking locations that look more balanced and even. This process will combine a bit of logic with trial and error. Stirred into the mixture will be a dusting of compromise.

The first obvious change will be to edit the size of the Heading Window. That element, as shown above, would look better if reduced to the same height as the Downwind Button. That

would also keep the spacing between rows one and two even across the width of the gauge.



Though not essential, this would improve the appearance of the gauge. The idea comes to mind to also adjust the width of that window. It will be edited to match the width of the large button below it. The new window is resized and shown at left.

The remaining steps involved in the assembly process are just a matter of push and pull until balance is achieved. Each action will not be documented. Instead, the result of the final assembly is shown here. Each blue A-Rex and red Guide was left in place so the reader can see how each was used to get the desired alignment or to resize the Heading Window.

A few additional explanations are in order. The text above the Power Button was aligned with the top of the other two buttons on the upper row. The letters were required to be of sufficient height to provide good readability, but fit into the space between the button and the Guide. The ability to use a larger font size was the major reason the Power Button was made smaller than all the other buttons.

The distances between all rows and the rim spacing at the top and bottom of the gauge are all equal in value. That was one nice result caused by lowering the height of the Heading Window. The four horizontal A-Rex spacers are also equal. That too resulted from editing the Heading Window. As you can see, the window now aligns at both ends with the Runway Button below it. Overall, the layout of the gauge is satisfactory and lends a very unified look to the controls. I will stamp this assembly finished.



The graphic at left represents the non-illuminated condition of the gauge. This is the image used by Flight Simulator as the background for the gauge. It will be seen when an aircraft panel is loaded in which the gauge is installed. The file name for the BMP will appear in the panel.cfg file of that aircraft. FS will be instructed not only which image to display, but where and at what size the display is to be drawn on the panel.

Earlier in this tutorial, creating an illuminated state for each of these buttons was demonstrated. That technique will now be put to practical use. The illuminated controls have an exacting relationship to their unlighted twin. That kinship is detailed next.

With flight simulator running and an appropriate command is detected by the XML code, the illuminated BMP will be placed directly over its unlit counterpart. That is done so fast that it gives the appearance the button has received electrical power and has thus become active.

What is needed is a technique to insure the lighted controls match the exact size of the unlighted buttons and that they are positioned directly over them when directed to do so. That is an essential requirement to achieve a smooth, visual transition from one state to the other. Otherwise, a control may appear to jump during this changeover.

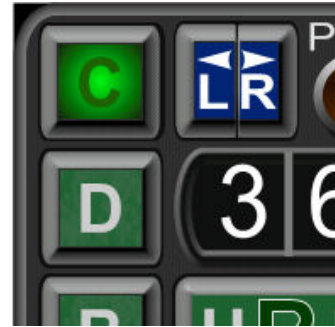


There are several ways to achieve exactness between the ON buttons and the OFF buttons. But, since XaraX is being demonstrated, the tools available inside that application will be pressed into service. Besides, they make it easy while maintaining vector graphic quality.

A summary of this process would read something like this: The entire OFF gauge is cloned. That includes all controls sitting in their final locations on the body of the gauge. From this clone, each individual button will be edited to its ON state. This will be done using the techniques already covered.

Because each control was grouped as they were completed, they must now be ungrouped on the clone. This allows access to every element inside the button. Those elements will be then be edited, as needed, to turn the control into the ON state. The techniques demonstrated earlier will be used for this. Care is taken that NO ELEMENT IS MOVED during this process. It is vital that the ON clone be identical to the OFF control except for the illuminated face.

At right, the Crosswind Button from the cloned gauge is used as an example. The button was selected, ungrouped, then edited into the illuminated state. At that point, all elements on that button would again be grouped for safety.



Another button is then chosen. It too would be edited in the same manner. This process is repeated until all controls that require illumination are finished. At that point, we will have produced the graphic shown at left. Each lighted control now occupies the same group of pixels relative to the body of the gauge as does its unlighted cousin. This entire graphic; buttons, window, and gauge body is grouped for safety.

Part 3 will begin with text. Although few text objects occupy the demonstration gauge, those that do will be discussed. Special effects commonly seen on gauges such as text on a curve will also be covered. Plus, many new tools and techniques will be detailed as gauge assembly begins.