Chapter 1: Documentation of Liberty BASIC Wire Frame Functions

Tom Nally -
steelweaver52

Return to Chapter 1: Introducing the Liberty BASIC Wire Frame Library

Contents

- Function FF.LBWF.LBWFVersion$()

Set Up the Viewing Environment

- **Function FF.LBWF.Camera(CamX, CamY, CamZ, VCtrX, VCtrY, VCtrZ, ZoomFac)**

In order to create a wire model view, the programmer must establish the point in space where the camera resides, and then identify the point in space where the camera points.

- CamX, CamY, CamZ - The x-, y- and z-coordinates of the camera location.
- VCtrX, VCtrY, VCtrZ - The x-, y- and z-coordinates of the point in space where the camera is pointed.
- ZoomFac - The zoom factor. Start out by assigning ZoomFac = 1. If the image is too large or too small, adjust accordingly.

Top of Page

- **Function FF.LBWF.ScreenCenter(ScrCenterX, ScrCenterY)**

This function establishes the "screen center". Essentially, the screen center is the point in your Liberty BASIC graphicbox that corresponds to the point defined by (VCtrX, VCtrY, VCtrZ).

- ScrCenterX - Usually, this is the x-coordinate of the center of the graphicbox
- ScrCenterY - Usually, this is the y-coordinate of the center of the graphicbox

Occasionally, the programmer does not want to plot the 3D image at the center of the graphicbox. In that case, move ScrCenterX and ScrCenterY accordingly.

Top of Page

Creating Boxes and Cylinders

Version 0.5 of the *LBWF Library* provides functions for the creation of six different objects. Below, I will provide the documentation for creating boxes and cylinders. In subsequent chapters, I will provide documentation for additional objects.

- **Function FF.LBWF.CreateBox(BoxName$, xdim, zdim, boxheight, BoxColor$)**

As the reader could guess, this function creates a *box*. Understand, however, that the data for the box is merely created in memory. The box will not be plotted in the graphicbox until a function is called which draws the object.

- BoxName$ - All LBWF objects must be assigned names. Names are handled as strings. The Library recommends that the programmer use object names without spaces.
- xdim - The dimension of the box in the x-direction
- zdim - The dimension of the box in the Z-direction
- boxheight - The height of the box.
- BoxColor$ - All objects must be given a color, passed to the function as a string. See the Liberty BASIC help files for proper ways to designate colors.

So, where are boxes -- and other objects -- plotted? The default location of a box is centered on the y-axis, with the bottom of the box resting on the X-Z plane. All objects are originally positioned similarly. Elsewhere, I will identify the function that allows the programmer to move any objects she creates.

Top of Page

- **Function FF.LBWF.CreateCylinder(CylName$, radius, numSides, cylheight, CylColor$)**
- CylName$ - The name of the cylinder, sent to the function as a string.
- radius - The radius of the cylinder
- numSides - The number of sides that you want the cylinder to have. Not that in the example above, the cylinder has 20 sides.
- cylheight - The height of the cylinder.

- CylColor$ - The color of the cylinder, sent to the function as a string.

Top of Page

Moving an Object

- **Function FF.LBWF.TranslateObject(ObjectName$, transX, transY, transZ)**

To *translate* an object means to move the object without rotating it. LBWF provides a single translation function to move an object in any or all of the x-, y-, or z- axes.

- ObjectName$ - The name of the object to be translated. This must be the name used when the object was created.
- transX, transY, transZ - The number of units to move the object in the x-, y- and z- directions.

Note that this function will always move the object relative to the object's current position. Note also that the object will not *appear* to move until a subsequent draw command is issued.

Top of Page

Drawing, Hiding, and Showing Objects

- **Function FF.LBWF.DrawObject(ObjectName$)**
- ObjectName$ - The name of the object to be drawn. This must be the name used when the object was created.

The DrawObject function draws the object identified as an argument. When the function is called, the object will be drawn regardless of whether its visibility property was set to "hidden" by the **FF.LBWF.HideObject()** function.

Top of Page

- **Function FF.LBWF.DrawAllObjects()**

This function requires no arguments. The function will draw all objects, except for those whose visibility property is set to "hidden" by the **FF.LBWF.HideObject()** function.

Top of Page

- **Function FF.LBWF.HideObject(ObjectName$)**

- ObjectName$ - The name of the object to be hidden. This must be the name used when the object was created.

When this function is called, the object is not *hidden* instantly. Rather, the function sets the visibility property of the object to "hidden". Then, the object will *not* be drawn the next time the **FF.LBWF.DrawAllObjects()** function is called.

Top of Page

- **Function FF.LBWF.ShowObject(ObjectName$)**

- ObjectName$ - The name of the object to be shown. This must be the name used when the object was created.

When this function is called, the object is not *shown* instantly. Rather, the function sets the visibility property of the object to "show" or "shown". Then, the object will *not* be drawn the next time the **FF.LBWF.DrawAllObjects()** function is called.

Top of Page

Miscellaneous Functions

- **Function FF.LBWF.ShowAxes()**

By default, the axes are not visible. This function ensures that the axes will be drawn when the FF.LBWF.DrawAllObjects() function is called. Function FF.LBWF.ShowAxes() requires no arguments. I like to call this function immediately after calling the Camera and ScreenCenter functions.

When the axes are drawn, the blue line represents the x-axis, the red line represents the y- axis, and the dark green line represents the z- axis.

Top of Page

- **Function FF.LBWF.HideAxes()**

If the programmer does not want the axes to appear the next time the FF.LBWF.DrawAllObjects() function is called, she may use Function FF.LBWF.HideAxes(). This function requires no arguments.

Top of Page

- **Function FF.LBWF.LBWFVersion$()**

This function will return the current version of the *LBWF Library*. It requires no arguments. With the publication of Newsletter 134 (reprinted here as Chapter 1: Introducing the Liberty BASIC Wire Frame Library), the current version of the library is 0.5.

Top of Page

Tom Nally
Steelweaver52@aol.com

Note: This linked article accompanies Chapter 1: Introducing the Liberty BASIC Wire Frame Library*, which originally appeared in the Liberty BASIC Newsletter, Issue #134. It is reprinted here with the permission of the author. -* JanetTerra