GraphWorX64 3D
Training Manual

Version 10.60

© 2011 ICONICS, Inc. All rights reserved worldwide. This document may not be reproduced or copied in any manner without written authorization from ICONICS, Inc.

The information contained within this document is subject to change without notice. ICONICS, Inc. does not guarantee the accuracy of the information.

GENESIS64, Hyper Historian, Pocket GENESIS, BizViz and their respective modules, OPC-To-The-Core, and Visualize Your Enterprise are trademarks of ICONICS, Inc.

Windows, Windows XP, Windows 2000, Windows 2003 Server, Windows CE, Windows NT, Windows XP-64, Windows 2003-64, Windows Vista, Windows Server 2008, Windows 7 and Windows Server 2008 R2 are registered trademarks of Microsoft Corp. in the United States and other countries.

Other product and company names mentioned herein may be trademarks or registered trademarks of their respective owners.

Document Number: TM-GWX64-3D-1060

Visualize Your Enterprise™

ICONICS LICENSE AGREEMENT

YOU SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS!

Opening and using the enclosed software for any purpose indicates your acceptance of the terms and conditions of this license agreement. If you do not agree with them you should return all software, documentation and copy protection keys within seven days of shipment unopened and your money will be refunded.

ICONICS provides this program and licenses its use in the United States, Puerto Rico, or internationally. You assume the responsibility for the selection of the program to achieve your intended results, and for the installation, use and results obtained from this program.

LICENSE

You are granted a personal license to use this program under the terms stated in this Agreement. You may: 1) Install and use the program on a single machine. 2) Make a single archival back-up copy of the program for the sole purpose of supporting your use of the single program on a single machine. 3) You may not use, copy, modify, or transfer the program, or transfer any copy, in whole or in part, except as expressly provided in this license, or with a written contractual agreement with **ICONICS,** Inc.

TERM

The license is effective until terminated. It will terminate if you fail to comply with any term or condition of this Agreement. You agree, upon such termination, to destroy the program and all copies that were made from it and to promptly return any copy protection key(s) to **ICONICS.**

LIMITED WARRANTY

ICONICS WARRANTS THE CD-ROM OR PHYSICAL DISKETTES, ON WHICH THE PROGRAM IS FURNISHED, AND PHYSICAL DOCUMENTATION TO BE FREE OF DEFECTS IN MATERIALS AND WORKMANSHIP UNDER NORMAL USE FOR A PERIOD OF NINETY (90) DAYS FROM THE DATE OF DELIVERY TO YOU AS EVIDENCED BY YOUR RECEIPT. THIS PROGRAM IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION.

ICONICS SPECIFICALLY DISCLAIMS ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, IMPLIED WARRANTIES OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN PARTICULAR, WITH RESPECT TO ANY PARTICULAR APPLICATION, USE OR PURPOSE, IN NO EVENT SHALL **ICONICS** INC. BE LIABLE FOR ANY OTHER COMMERCIAL DAMAGE, INCLUDING BUT NOT LIMITED TO SPECIAL, INCIDENTAL, CONSEQUENTIAL OR OTHER DAMAGES. **ICONICS** DOES NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE PROGRAM WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE PROGRAM WILL BE UNINTERRUPTED OR ERROR FREE.

LIMITATION OF REMEDIES

ICONICS' entire liability and your exclusive remedy shall be the replacement of any CD-ROM or diskette not meeting **ICONICS**' "Limited Warranty" which is returned to **ICONICS** with a copy of your receipt within the warranty period. The remedy for breach of this warranty shall be limited to replacement and shall not encompass any other damages including but not limited to loss of profit, special, incidental, consequential, or other similar claims arising out of the use or inability to use such program even if **ICONICS** has been advised of the possibility of such damages, or for any claim by any other third party.

GENERAL

This Agreement will be governed by the laws of the Commonwealth of Massachusetts. Should you have any questions concerning this Agreement, you may contact **ICONICS** in writing at:

ICONICS, Inc.
100 Foxborough Boulevard
Foxborough, MA 02035

You acknowledge that you have read this agreement, understand it and agree to be bound by its terms and conditions. You further agree that it is the complete and exclusive statement of the Agreement between you and **ICONICS** which supersedes any proposal or prior Agreement, oral or written, and any other communications between you and **ICONICS** relating to the subject matter of the Agreement.

ICONICS RETURN POLICY

All sales are final (NO RETURNS) unless one of the following conditions is applicable: Program is returned in its original packaging material within 7 days after shipment. In such a case, a full refund is provided, less all incurred shipping and handling costs. Program is returned within 30 days from the original shipment date. In such a case a 20% restocking fee is applicable. No returns are allowed on OPC ToolWorX or ActiveX ToolWorX.

Visualize Your Enterprise™

Contents

1. Introduction to 3D

1.1. OBJECTIVES 1-1
1.2. GETTING STARTED 1-2
1.3. GENDEMO AND THE LEARNING CENTER 1-2
1.4. CREATING 3D CONTENT 1-4
REVIEW: INTRODUCTION TO 3D R1-1

2. The 3D Space

2.1. OBJECTIVES 2-1
2.2. INTRODUCTION TO THE 3D SPACE 2-1
2.3. THE 3D RIBBON 2-5
2.4. NAVIGATING IN A 3D VIEWPORT 2-12
LAB: NAVIGATING THE 3D SPACE L2-1
REVIEW: THE 3D SPACE R2-1

3. Primitives

3.1. OBJECTIVES 3-1
3.2. INTRODUCTION TO PRIMITIVES 3-1
3.3. COMMON PRIMITIVE PROPERTIES 3-2
3.4. THE PLANE PRIMITIVE 3-5
3.5. THE POLYGON PRIMITIVE 3-6
3.6. THE CUBE PRIMITIVE 3-6
3.7. THE SPHERE PRIMITIVE 3-7
3.8. THE CYLINDER PRIMITIVE 3-8
3.9. THE CONE PRIMITIVE 3-9
3.10. THE DONUT PRIMITIVE 3-10
3.11. THE PIPE PRIMITIVE 3-10
3.12. THE TERRAIN PRIMITIVE 3-12
3.13. ANNOTATIONS 3-13
LAB: ICTONICS TANK L3-1
REVIEW: PRIMITIVES R3-1

4. Manipulating Objects

4.1. Objectives 4-1
4.2. Opening a 3D Object 4-1
4.3. Selecting 3D Objects 4-2
4.4. Coordinates in 3D 4-5
4.5. Arranging 3D Objects in Space 4-7
4.6. Simple Transformations 4-11
4.7. Advanced Transformations and Mirroring 4-15
4.8. Duplicating Objects 4-20
4.9. Applying Materials to Object Surfaces 4-25
LAB: MANIPULATING OBJECTS L4-1
REVIEW: MANIPULATING OBJECTS R4-1

5. 3D Animation

5.1. Objectives 5-1
5.2. Dynamics 5-1
5.3. Position Dynamics 5-2
5.4. Visibility Dynamics 5-8
5.5. Collision Detection 5-11
5.6. Animating Views at Runtime 5-13
LAB: 3D DYNAMICS L5-1
REVIEW: 3D ANIMATION R5-1
LAB: ICTONICS PRODUCTION LINE L6-1

A. Glossary **A-1**

B. Answers **B-1**

Introduction to 3D

1

GraphWorX64's 3D feature gives you the special ability to create dynamic displays that provide a more powerful and compelling user experience, especially when interacting with dynamic machine-based controls and three-dimensional sites and objects.

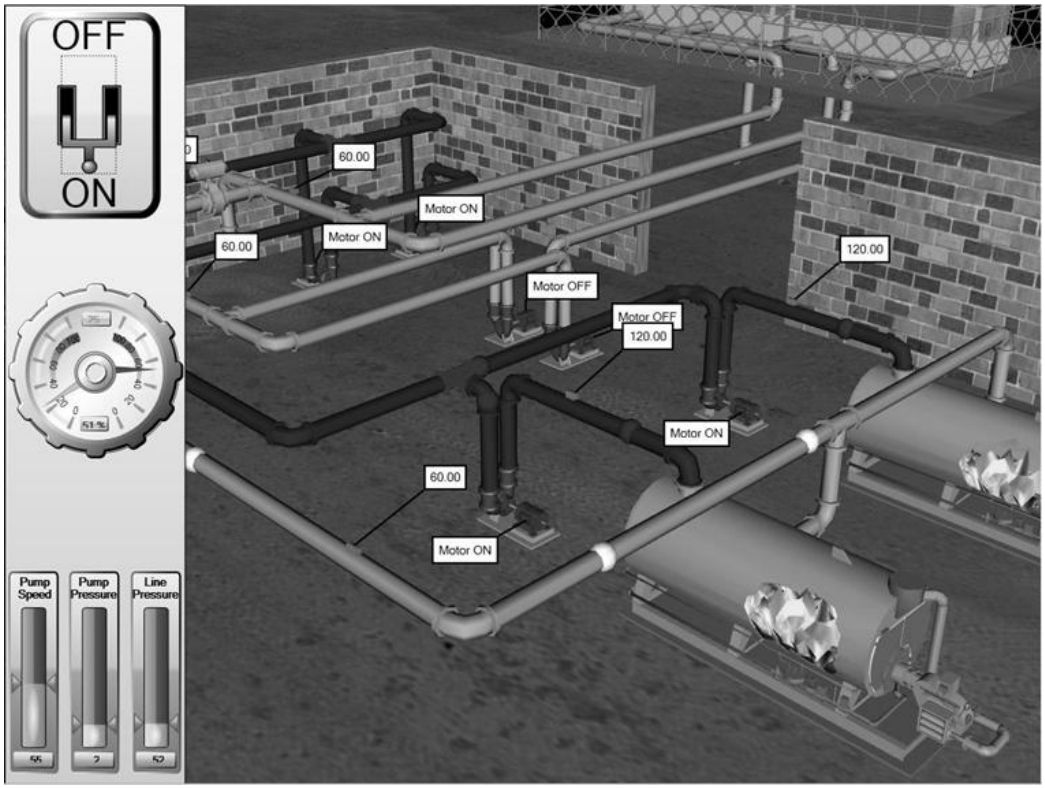

Figure 1.1 - A 3D Display Created in GraphWorX64

1.1. Objectives

After completing this chapter, you should understand:

- The minimum requirement to run GraphWorX64
- The technology behind how GraphWorX64 works
- The uses for GraphWorX64 3D objects
- The main features of GraphWorX64 3D
- The file format supported for 3D graphicsObjective 2

1.2. Getting Started

With 3D graphics, you can create displays that let users:

- Visualize real-time data in a three-dimensional realm, using animation to reflect actual data originating from an OPC tag.
- Do virtual deconstructions of machines to look inside for problems.
- Move virtually, for example, through a reactor or inside a machine.
- Monitor a site remotely, jumping from floor to floor, from building to building, or from street to street.
- Manipulate objects and take action remotely and then monitor progress.

GENESIS64 is developed from the ground up using C# and .Net Framework 3.0. Because of this, we can take advantage of the Windows Presentation Foundation (WPF), which is a graphical subsystem in the .Net Framework 3.0. It uses a markup language called Extensive Application Markup Language (XAML), which allows us to import 3D models saved in this format. Together, .Net Frameworks 3.0 and XAML allows us to generate and display superior graphics and animations in GraphWorX64.

In order to take full advantage of the technologies used in GraphWorX64, we recommend that your machines have at least the minimum specifications listed in Table 1.1.

Table 1.1 - Machine Specifications

	Recommended	Minimum
64-bit Processor Speed:	2.26GHz	2 GHz
RAM:	6 GB	2 GB
Video Card:	Dedicated	Dedicated
Onboard memory:	512 MB	256 MB
Resolution:	1280x1024 with 32 bit color	1024x768 with 32 bit color
Support:	DirectX 9, 10 or above	DirectX 9

1.3. GenDemo and the Learning Center

There are several industry examples in GenDemo, which is installed with GENESIS64. These demos, highlighted in Figure 1.2, are stored in the GenDemo library as GraphWorX64 files. You can use them as a reference point when you begin to create your own displays. You can view these displays by opening them in GraphWorX64 and analyze their configurations. The GenDemo also has a number of 2D and 3D Industry Case Studies that you can look at as well.

Figure 1.2 - GraphWorX64 3D Animation Provided in GenDemo

The GenDemo also comes with a Learning Center where you can learn more about the 3D features of GraphWorX64. The Learning Center is interactive and lets you explore each feature and even provides examples. Figure 1.3 shows an example of a Learning Center lesson.

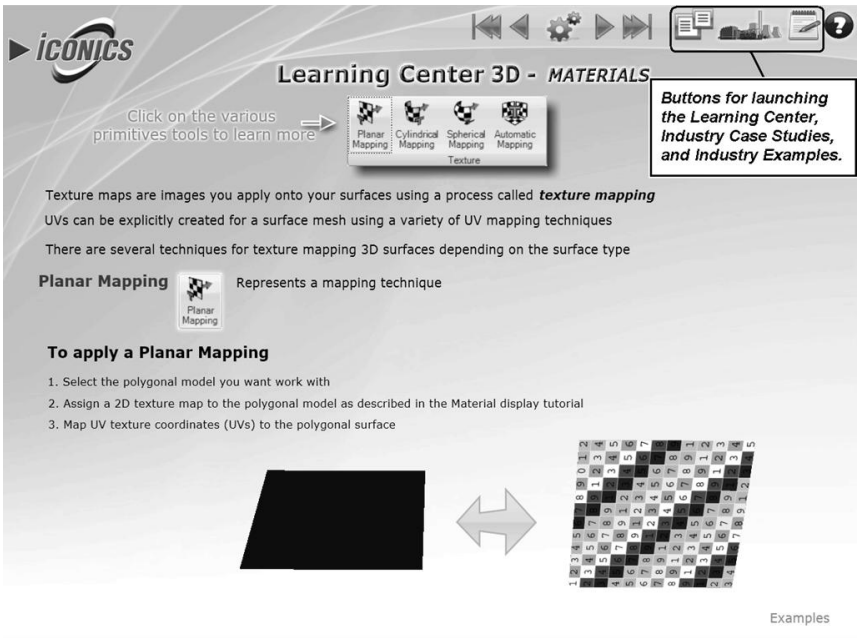

Figure 1.3 - Using the Learning Center to Learn More about 3D Features

1.4. Creating 3D Content

GraphWorX64 provides a wide range of display options. One of the advanced features is the ability to provide graphics in a three dimension space. This feature uses many of the same concepts that are used by its 2D counterpart, and for this reason, you should familiarize yourself with building 2D displays before continuing. This course will focus on the advanced 3D features.

The 3D graphics that you use in displays can:

- Model 3D objects and add textures and images to their surfaces
- Animate objects based on real-time or historical data
- Create 3D objects that dynamically interact
- Add events and scripting to 3D objects
- Overlap a 3D layer of objects with a 2D layer of objects

When creating 3D contents, a number of factors will determine the complexity of your 3D objects. For example, the required refresh rate and speed could be compromised if a machine's capability cannot adequately handle a complex graphic. You should always keep performance in mind when creating 3D contents. For now, let us take a look at the different ways we can use to obtain 3D contents.

You can create 3D contents using a few different ways. For example, you can:

- Draw your own 3D objects from scratch, using primitives available in GraphWorX64
- Use predefined objects from the GraphWorX64 3D Symbols Library. You can modify objects in the Symbols Library and save them for their modified use. You can even add your modified object back into the Symbols library.
- Import existing models from external applications that support XAML, COLLADA or 3DS file format

We will describe each of these techniques below. For the remainder for the course, we will focus on creating 3D contents by drawing them from primitives.

1.4.1. The Symbols Library

You can always start by using a 3D symbol from the GraphWorX64 Symbols Library. The Symbols Library has sub-folders for 3D objects in most of its sections (shown in Figure 1.4) that represent models for use in different industries.

Some symbols, such as robots, have animated features programmed into them, which you can use or change; others, such as pipe segments, have no animation.

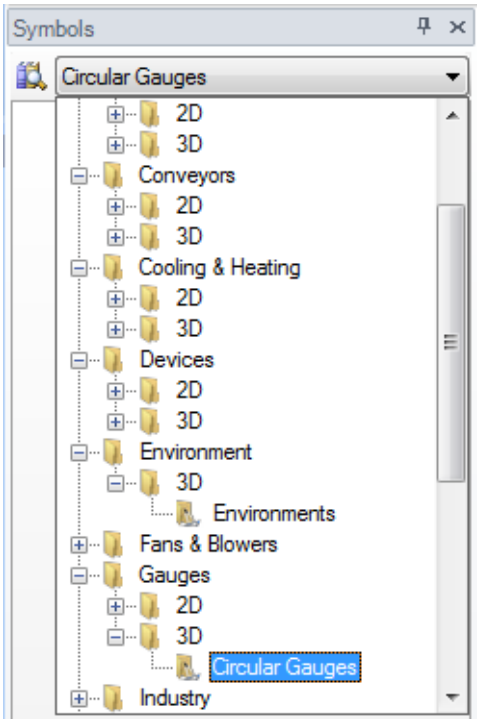

Figure 1.4 - The 3D Symbols Library

For other vertical markets, additional 3D symbols libraries are available from ICONICS for an additional fee. For example, if you specialize in machine building, you can get a specialized symbols library that contains symbols geared specifically to your industry. Contact your ICONICS representative for more information.

Once you have downloaded the supplemental 3D symbols from ICONICS, you should add them to the symbols library by adding a new category. Once you do this, you will be able to see the symbols in the library and they are ready for you to use.

To use a symbol, simply drag-and-drop one of these symbols into your design surface. From here, you can edit the symbol by adding or removing parts or even associate dynamics to it to fit your needs.

1.4.2. Primitives for Creating Objects from Scratch

You can create basic objects from scratch by using the primitive 3D structures that are available in GraphWorX64. These shapes include cones, spheres, cylinders, and cubes, which you can combine to create complex illustrations or models, like the one shown in Figure 1.5. The objects you produce in this way can be very effective for your GraphWorX64. We will look more closely at primitives later in this course.

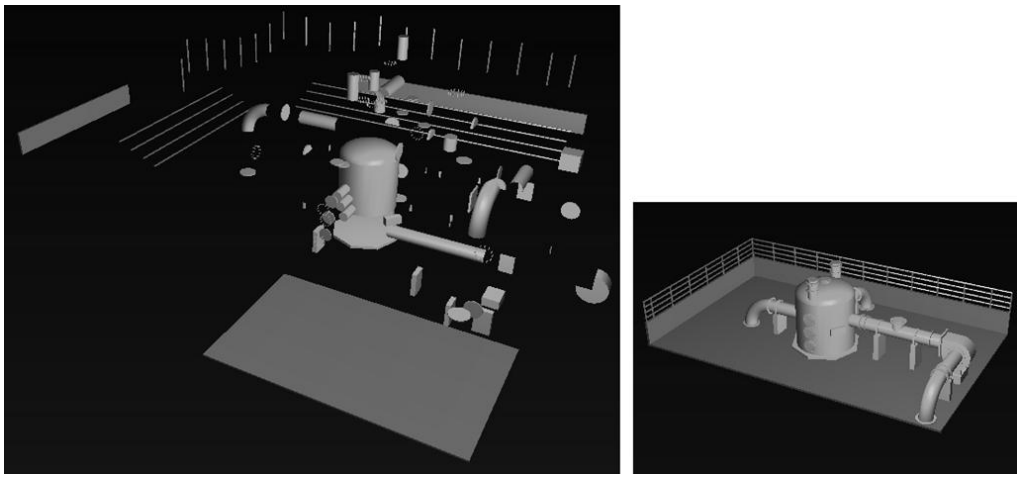

Figure 1.5 - Shapes (Left) Used to Create a Tank (Right)

1.4.3. Importing Models

It is likely you will import a majority of your models from files created using other tools, such as the station model shown in Figure 1.6. Approximately 90% of the graphics formats that are available on the market can be imported directly into GraphWorX64, or converted to XAML and then imported. Let us look at the file format requirements.

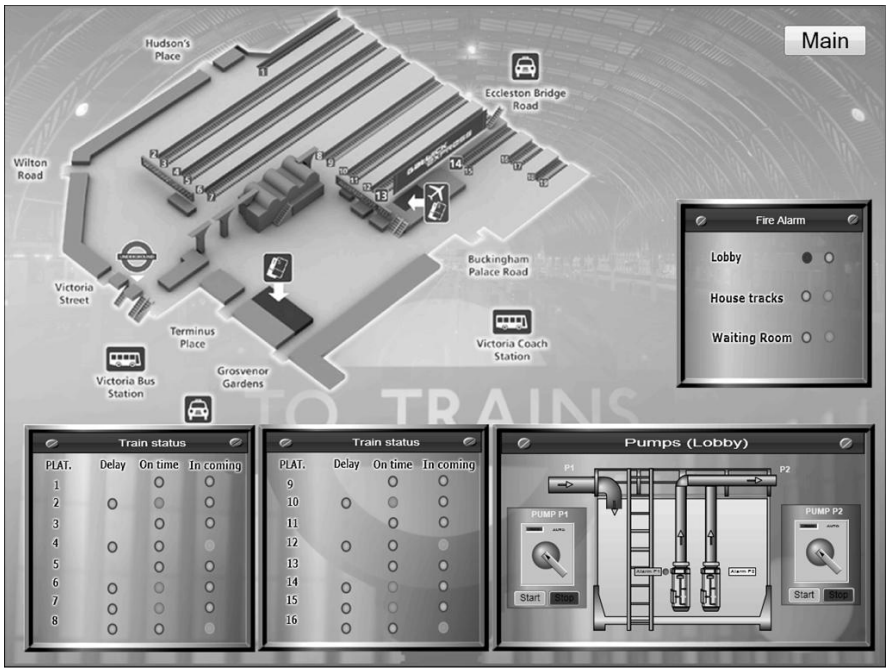

Figure 1.6 - A 3D View of a London Underground Train Station with Controls

Natively, GraphWorX64 3D graphic are saved in the XAML format. You can import 3D graphics files using the built-in importer for XAML, COLLADA (an open standard format) and 3DS (3D Studio, a royalty-free and prolifically-used tool). No additional external software or tools are needed in order to import XAML, COLLADA or 3DS files. Any model that is created in XAML, COLLADA or 3DS format can be imported into GraphWorX64 for use in a 3D display. A myriad of such 3D models can be found on the internet to get you started.

But what do you do if your 3D content is in some other format?

Exporting files to XAML format using external 3D content software: One solution is to use your external 3D content software to export files into XAML format, making it possible to move models into GraphWorX64. There are a number of free plug-ins you can use to **export** from external 3D content programs into the XAML format. In particular, there are free plug-ins for Autodesk® Maya® and for Blender. Note that you must own a copy of the external program in order to export a model from it.

Migrating (converting) files to XAML format: If you do not have the external program to export files into XAML format, there are external utilities that you can purchase to **migrate** files from different file formats into XAML. One of these is PolyTrans from Okino, which has partnered with ICONICS to supply converters for an additional fee. You can contact Okino through their web site (www.okino.com) and make sure you tell them you are using ICONICS GraphWorX64. Contact your ICONICS representative for more information.

1.4.3.1. Performance Considerations

A model that is created in a high-fidelity CAD program typically contains details, such as nuts and bolts, that are not necessary for use in real-time applications such as GraphWorX64. Such details typically are not monitored, their inclusion in the model does not provide feedback to the operator and, in most cases, there is no harm in removing them from the 3D model that you will be using in GraphWorX64.

It is important to eliminate these unneeded details wherever possible because they add to the number of polygons in a model, which can become a burden for the graphic card and can slow down the runtime performance of GraphWorX64. For this reason, before you export any model, review the model and look at the number of polygons contained in it, and eliminate all details in the model that will not have a use in any GraphWorX64 display that you will be creating.

Review: Introduction to 3D R1

1.1. Chapter Objectives

You should now know and understand the following topics:

- The minimum requirement to run GraphWorX64
- The technology behind how GraphWorX64 works
- The uses for GraphWorX64 3D objects
- The main features of GraphWorX64 3D
- The file formats supported for 3D graphics

1.2. Summary

GraphWorX64 is built on .NET 3.0 technologies. For 3D graphics formats, GraphWorX64 uses WPF (Windows Presentation Foundation) API standard for graphics and animation, and XAML (Extensible Application Markup Language) file format. .NET scripting extends the usefulness of GraphWorX64 3D objects beyond the features that are available in GraphWorX64.

GraphWorX64 3D uses many of the concepts that are used by 2D features, except that GraphWorX64's 3D gives you tools for creating dynamic displays that provide a more powerful and compelling user experience, especially when interacting with dynamic machine-based controls and three-dimensional sites and objects. The ICONICS GenDemo has several industry examples that you can view and even use in your own 3D creations.

You can create 3D contents in a few different ways. You can draw your own 3D objects from scratch, using primitive structures such as cones, spheres, cylinders, and cubes, which you can use to create simple but effective 3D views. You can also use predefined objects from the GraphWorX64 3D Symbols Library. You can modify objects in the 3D Symbols Library and save them for their modified use. You can also purchase a specialized symbols package for your particular vertical market by contacting you ICONICS representative.

It is more likely that you will import predefined models from external XAML editors. Any model that is created in XAML format can be imported into GraphWorX64 for use

in a 3D display. There are several importers and converters available for converting your models to XAML.

1.3. Terms

Primitives WPF XAML

1.4. Test Your Knowledge

Before continuing to the next chapter, make sure you can answer the following questions.

1. Which of the following contains 3D examples and objects that are already made?

   a. The GraphWorX64 3D Symbols Library
   b. The ICONICS GenDemo has industry examples
   c. Models you have that are in XAML format can be imported
   d. All of the above
   e. None of the above

2. True or False: There is no scripting feature available in GraphWorX64 3D.

3. True or False: You can create 3D objects from scratch by using 3D primitives.

4. You can import any model that is in XAML format. What can you do if your models are not in XAML format?

   a. Import them into GraphWorX64 and GraphWorX64 will convert them automatically
   b. Use a converter to convert them, and then import them into GraphWorX64
   c. Redraw them in GraphWorX64 using primitives
   d. Send them to ICONICS for conversion

5. True or False: You can have 2D and 3D contents on the same GraphWorX64 display.

6. True or False: Including detailed models in your 3D will not affect runtime performance in any way.

The 3D Space 2

The purpose of this chapter is to show you how to work within a 3D space, or commonly known as the 3D viewport. You must understand how to navigate and move in the 3D space before you can work with 3D objects in it.

2.1. Objectives

After completing this chapter, you should have an understanding of:

- The 3D viewport
- Activating and deactivating a 3D viewport
- Orientation and the X, Y, and Z axes
- Perspectives
- Grid options
- The 3D viewport's ribbon
- How to navigate in a viewport

2.2. Introduction to the 3D Space

GraphWorX64 can display objects in a two-dimensional (2D) or three-dimensional (3D) space, but the objects in each type of space are separated into their own separate sets of coordinates.

A 2D space is a flat surface and is the default dimension for a GraphWorX64 display. You can use techniques such as shading or offsets to mimic three-dimensionality in a 2D space, but all of the objects are still defined in GraphWorX64 as 2D objects with an X and Y position.

You can add a 3D space to a 2D display by clicking on the **3D View** button in the Shapes section of the Home tab in the GraphWorX64 ribbon. This action creates a 3D space called a ***3D viewport*** or ***3D space***. When you work with 3D objects, you work inside this viewport on the GraphWorX64 design surface. The viewport is a sub-area in a 2D space; the 3D viewport opens up when working with a 3D object or when working in a 3D view. You can add multiple viewports on your design surface. Technically, the 3D viewport is an object in GraphWorX64, and like any other object in GraphWorX64, it appears in the Explorer.

You will notice a few differences when working in the 3D space compared to working in a 2D space. First, in 3D you are working with an additional dimension – depth – which is handled by a third axis, the Z-axis, for locating objects. To reduce confusion, an ***axis legend*** in the 3D space helps you determine which direction you want to work with in the display. This also helps when you need to change an object's size or move an object in a particular direction.

You will also notice that when you are working in the viewport, the ribbon changes to a specialized 3D ribbon. This ribbon contains options and actions that are available only when you are working in the 3D space. We will discuss the different options available here later.

Although the 3D space is quite different to work with than the 2D space, with time and lots of practice, it will become second nature to work with it. Let us now see how we can add a viewport to our display.

2.2.1. Adding a 3D Space to Your 2D Display

There are two ways to add a 3D space to your GraphWorX64 display. First, you can add one simply by adding any 3D object to a 2D work surface. By adding such an object, you automatically add the 3D viewport because the object needs the 3D space to exist.

The simplest way to do this is by dragging a 3D model from the symbols library to your display. Figure 2.1 shows a 3D boiler that has been added from the symbols library. Notice that the viewport is there but it is not yet activated.

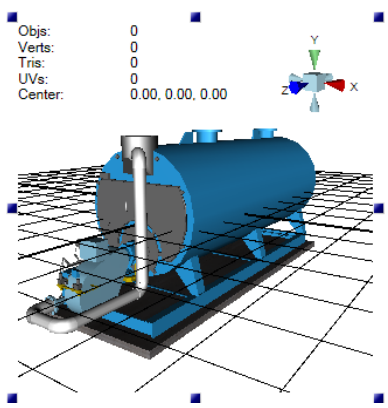

Figure 2.1 - 3D Object in an Inactive Viewport

You can double-click on the 3D object to activate the viewport. Once the viewport is activated, a thick purple line surrounds the boundaries of the viewport. An activated viewport should look like the one in Figure 2.2 – notice how the selection boxes that surround the tank have changed to a thick line.

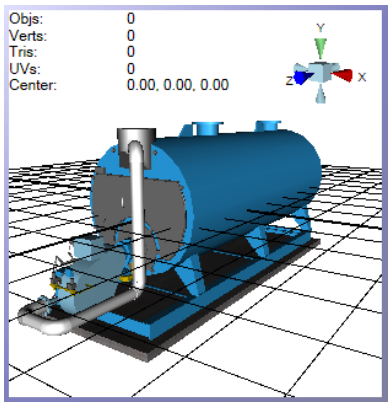

Figure 2.2 - 3D Object in an Activated Viewport

Alternatively, you can create a viewport by inserting an empty 3D view. You can do this by clicking on the 3D View button in the Shapes section of the Home tab in the GraphWorX64 ribbon.

Once you select to create a viewport, you then click in the design surface and hold down the mouse button while you drag the mouse to create the desired viewport size, and then release the mouse button. Now, you should have an active viewport on your display, and the GraphWorX64 ribbon switches to the 3D ribbon, as shown in Figure 2.3.

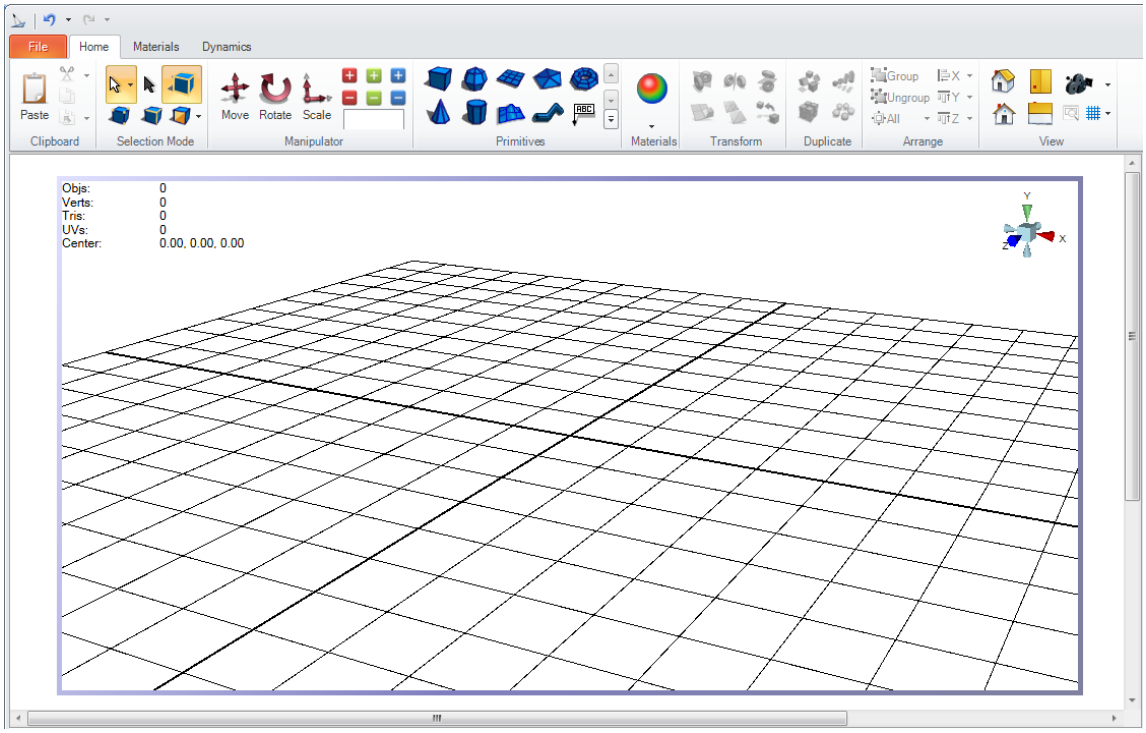

Figure 2.3 - The Activated 3D Viewport with the 3D Ribbon at the Top

2.2.2. The Color-Coded Axis Legend

The thing to remember about a 3D viewport is that it has three dimensions, not two. The three dimensions are represented by a color-coded *axis legend* in the viewport that labels the X-, Y-, and Z-axes, shown in Figure 2.4. Knowing the direction of each of these axes is fundamental to navigating in the viewport and understanding how to use the transformation options.

Because you can manipulate the perspective in the viewport, the orientation is always changing. At the top right corner of the viewport, the legend always maintains the true orientation of the three axes as you manipulate the viewport.

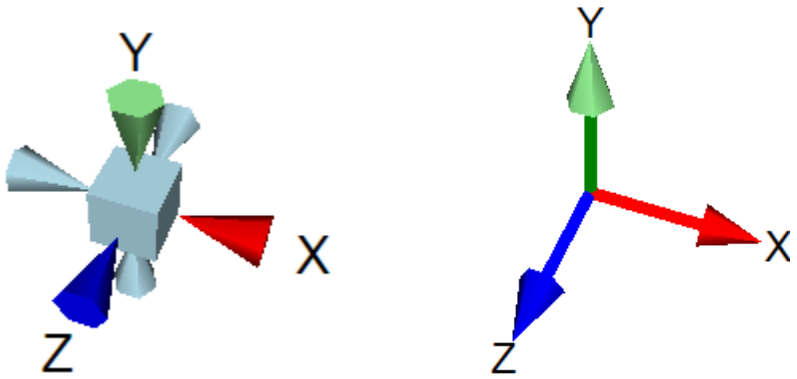

Figure 2.4 - The Color-Coded Axis Legend in Cube Form (Left) and Standard Form (Right)

In the Cube form of the legend, each axis is labeled with a small colored pointer, the tip of which points to the axis' plane. In Figure 2.4, you can see the X with its *red* pointer tip that identifies the plane as the right side of the center cube; this is the X-axis orientation. A *green* pointer for the Y axis and a *blue* pointer for the Z axis point to the other two planes. Note that there is a Standard form of the legend you can use instead.

If you wish to hide the axis legend, you can go to the viewport's properties and change the ShowAxis option in the Content section from True to False. In the same section, you can change the Axis Type from Cube (the default) to Standard if you like.

2.2.3. Deactivating a Viewport

While you are working in a viewport, you will not be able to interact with objects outside of it, nor will you be able to add 2D graphics to the display. You will need to deactivate the viewport to go back to your 2D space.

To deactivate a viewport, double-click outside of the viewport boundary in the design surface. This lets you toggle into and out of a 3D viewport, and is especially helpful

when you want to work with both 2D and 3D features in the same GraphWorX64 display.

You can also deactivate a viewport by pressing the **Esc** key. Keep this in mind for those times when the viewport takes up the entire screen and you cannot click outside its boundaries.

2.3. The 3D Ribbon

The 3D environment has its own ribbon that appears when the 3D viewport is active. When the viewport is inactive, the normal GraphWorX64 ribbon returns. The 3D ribbon includes the Home, Materials, and Dynamics tabs, along with the File menu as shown in Figure 2.5.

Figure 2.5 - The 3D Ribbon's Home Tab

> **Note:** One noteworthy difference between the 3D ribbon and the normal GraphWorX64 ribbon is that you cannot save a file from within the 3D viewport; you must inactivate the viewport if you want to save the GraphWorX64 display.

Let's take a look at the tabs on the 3D ribbon.

2.3.1. The File Menu

On the 3D ribbon, the *File* menu contains options to start a new 3D space (clearing the 3D space), import and export 3D objects, set some options for the 3D view control, optimize the scene and display scene information. Let's look at these last two options.

2.3.1.1. 3D Object Information and Performance

Once you are in a 3D viewport, the File menu has a **Scene Info** button that hides and shows the scene information at the top left of a 3D view, as illustrated in Figure 2.6.

The following information displays for the objects that are currently selected in the 3D viewport. This information is important only because these are factors that can affect runtime performance:

- **Objs**: The number of selected objects

- **Verts**: The number of vertices in the selected object(s). The number of vertices on an object affects memory. This number tells you relatively how many polygons are used to create an object and polygons consume memory on the video card. For example, a rough-surfaced sphere with 20 points consumes less memory on the video card than a refined sphere that has 10,000 points. We will talk more about vertices in a later chapter.

NOTE: A 256 MB video card can hold about one million polygons.

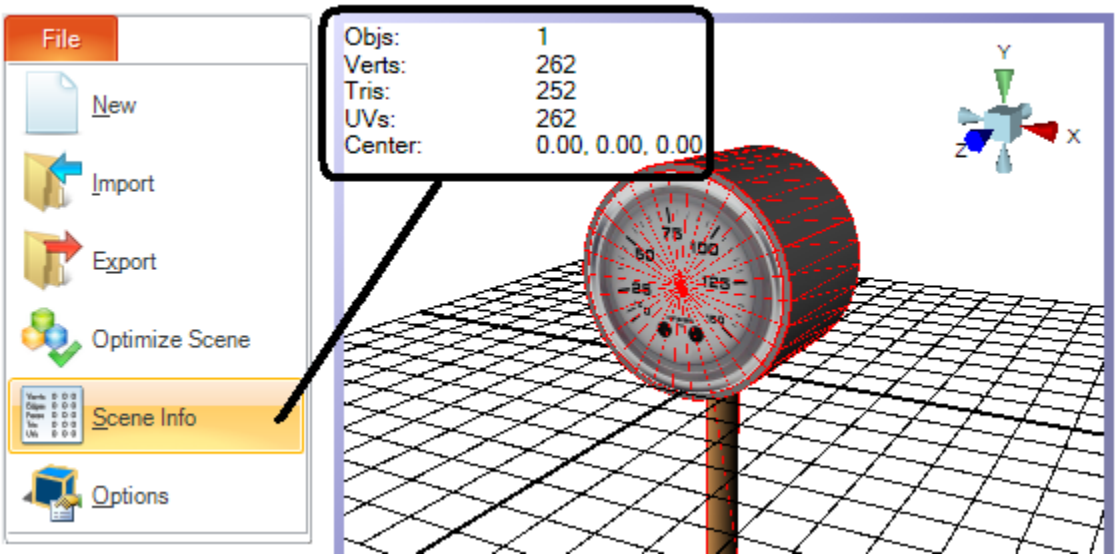

Figure 2.6 – The File Menu (Left) and Scene Information (Right)

- **Tris**: The number of triangles in the selected object(s). Its impact is the same as the number of vertices.
- **UVs**: The number of texture coordinates in the selected object(s). Texture can potentially consume a lot of memory. We will talk more about texture in a later chapter.
- **Center**: The X, Y, and Z coordinates of the center point of the current selection.

If you think that the number of vertices, triangles, and UVs is affecting performance of the display, the File menu also has an Optimize Scene button that you can click to make the scene more efficient by reducing the file size and improving performance.

2.3.2. The Home Tab

The **Home** tab of the ribbon contains commands and options for creating and manipulating 3D graphics, as shown earlier in Figure 2.5. The Home tab contains several sections, most of which are used when you are creating a 3D object in the

viewport. Two of the sections on the Home tab are designed to help you navigate and understand the scene in a view. These two tab groups are **Options** and **View**. Let's look at each of these sections.

2.3.2.1. Changing the Perspective

The **View** section, shown in Figure 2.7, provides shortcuts to different perspectives in the 3D viewport. The four left-most buttons in this section let you *snap* to a specific perspective of the objects in the viewport.

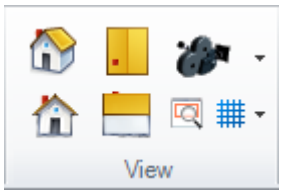

Figure 2.7 - The View Section

NOTE: The Camera button is used for creating movie-like behavior. We will talk about it later when we discuss animating your 3D objects.

There are two types of perspectives – 3D orthographic. A ***3D perspective***, which you get by clicking the Perspective icon, most closely resembles human vision, where objects appear to recede into the distance creating a sense of depth and space, as shown in Figure 2.8. In the perspective view, you can use the mouse to rotate, pan, and orbit to see objects from any point of view.

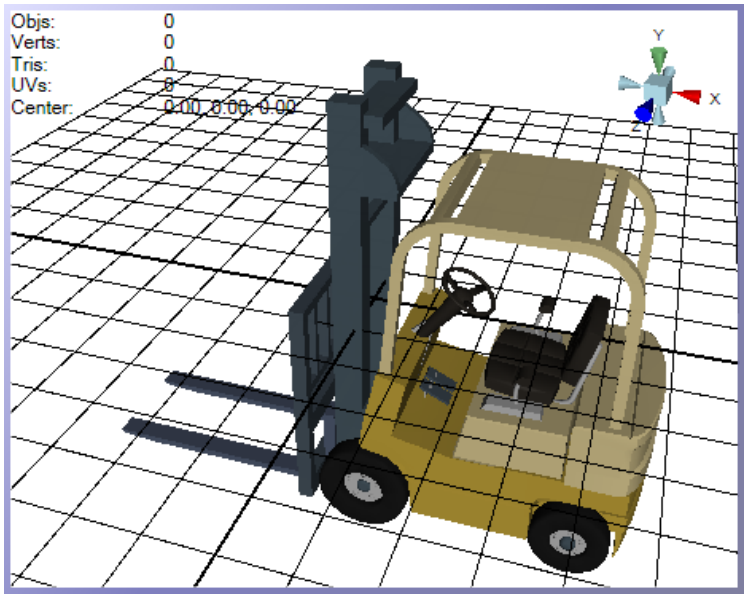

Figure 2.8 - A Perspective View of a Forklift

Orthographic perspectives are almost like a 2D representation of the scene without depth or a horizon line. All lines in the model are parallel to one another. An orthographic view is a straight-on view from a particular axis, such as from ***above*** (the *top* view, from the Y axis), from the ***front*** (from the Z axis), and from the ***side*** (from the X axis).

For example, you might want to view a building from above so that you can place condensers, solar panels, and vents accurately on its roof. Or you might want to view a switchboard from the front to accurately position switches on the panel.

Click on the Top icon to see the object from above, as shown in Figure 2.9.

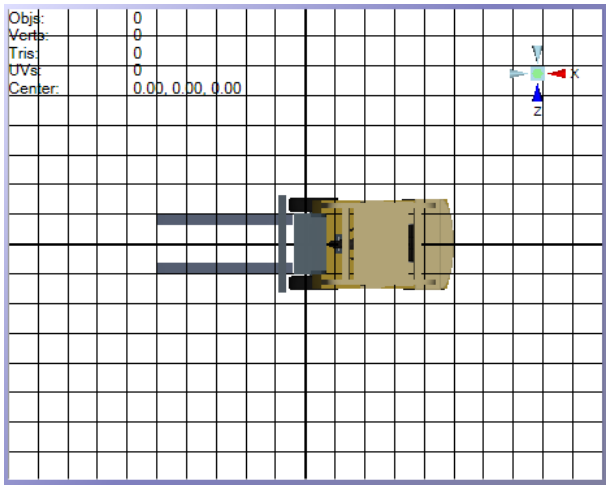

Figure 2.9 - Top View of the Forklift

Click on the Front icon to see the object from the front, as shown in Figure 2.10. The line that you see through the forklift is the front view of the XZ plane's grid, which intersects the forklift in this example.

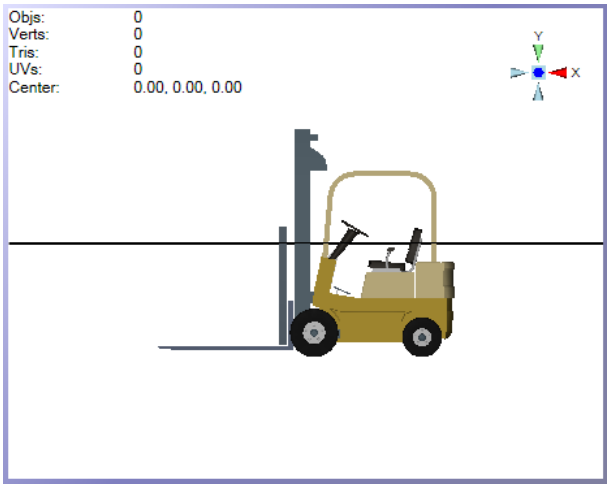

Figure 2.10 - Front View of the Forklift

Click on the Side icon to see the object from the side, as shown in Figure 2.11. Again, the line you see through the forklift is the side view of the XZ plane's grid. In Section 2.3.2.3, "Changing the Grid", we will look more closely at ways to show and hide this and two other grids in the 3D viewport.

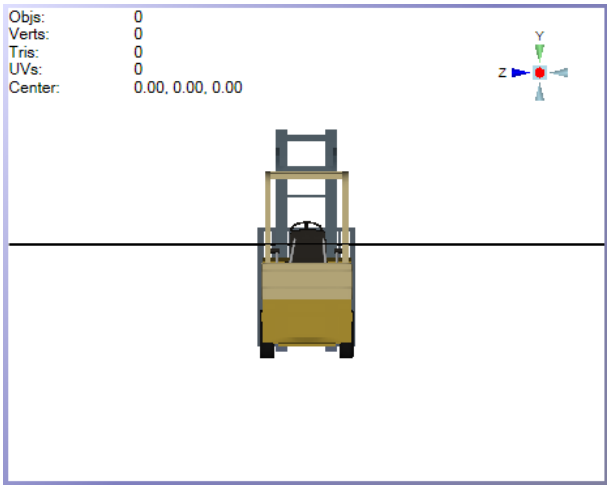

Figure 2.11 - Side View of the Forklift

2.3.2.2. Centering an Object in the Scene

Clicking the Center View icon in the View section automatically centers the selected object in the view so that you have clear visibility to it. It will even zoom in if the object is small, or zoom out is the object is large. This is helpful when you simply need to get a better look at an object but don't want to bother using the navigating tools.

2.3.2.3. Changing the Grid

The Grid button gives you several options for creating a grid that you can use for drawing and aligning components of a graphic object. In addition to snapping objects to the grid structure that displays in the scene, you can also define the grid itself. A good example of what a grid looks like was shown earlier in Figure 2.9. When you click on the drop-down arrow of the Home ribbon's Grid icon, a menu drops down as shown in Figure 2.12. Here, you can turn on the Snap to Grid and show any of three grids.

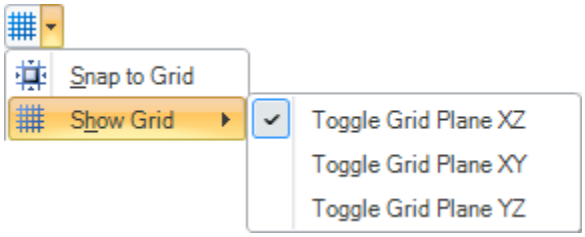

Figure 2.12 - Grid Options

When you click on the **Grid** button's **Options** menu item, you see the Grid Options dialog shown in Figure 2.13. The same options are available in the Properties explorer of the viewport when you are working in the 2D space.

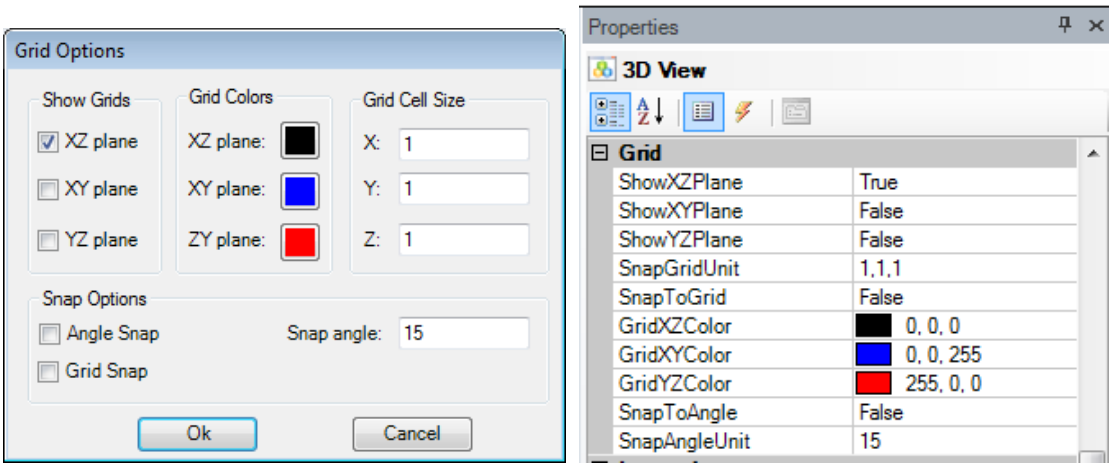

Figure 2.13 - The Grid Options Dialog (Left) and the Grid Properties for the Viewport (Right)

In the dialog box, you can choose the grid's cell size and colors. You can also snap to the grid, or snap to an angle of your choosing.

You can also show the grid planes for the XY, XZ, and YZ axes, as demonstrated in Figure 2.14.

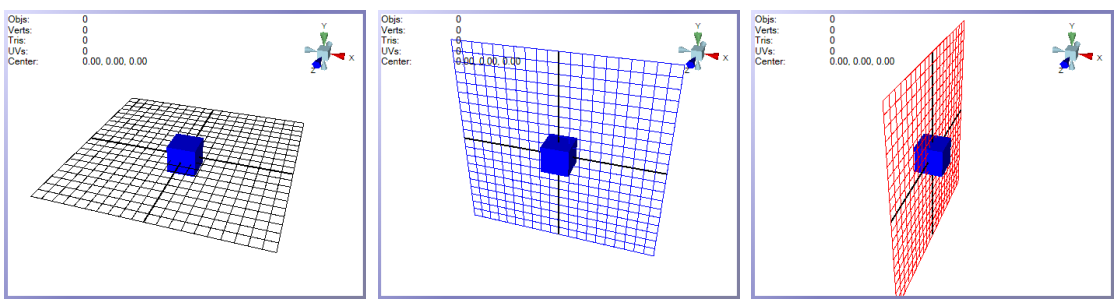

Figure 2.14 - The Show Grids option for the XZ (Left), XY (Center), and YZ (Right) Planes

The grid planes are color coded so that you can easily distinguish each plane when displaying them all at the same time, as shown in Figure 2.15. Displaying grid planes helps you to orient yourself in the 3D viewport so that you can rotate an object and look at it from alternate perspectives while maintaining an understanding of the orientation in the 3D space. You can show any one, any two, or all three grids at any time.

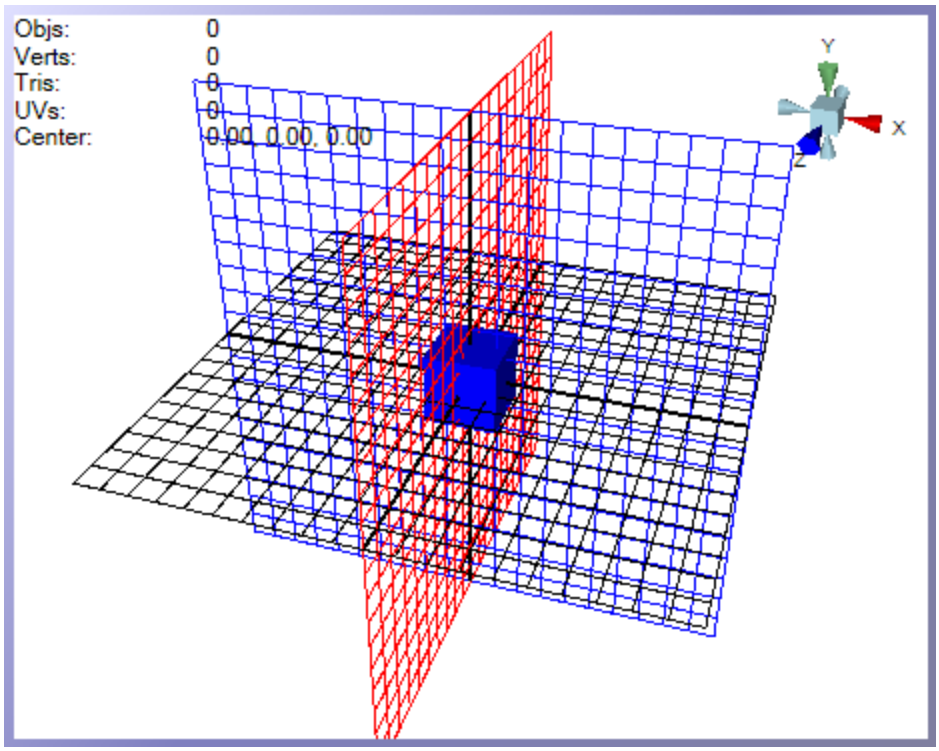

Figure 2.15 - All Three Grids Displayed at Once

2.3.3. The Materials and Dynamics Ribbons

The **Materials** tab of the ribbon, shown in Figure 2.16, contains options for changing the color and appearance of your 3D objects in the viewport. The **Dynamics** tab of the ribbon, also shown in Figure 2.16, contains *dynamics* which, as you know from working in 2D, are options for connecting your 3D objects to real-time data to animate them. The Dynamics tab contains most of the options that you see in the 2D Dynamics tab and their function is about the same, too. We will look more closely at the options on these ribbon tabs later.

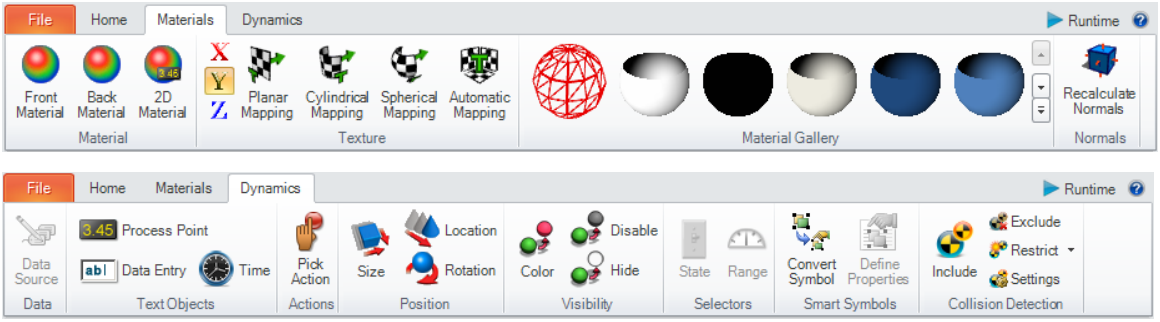

Figure 2.16 - The Materials (Top) and Dynamics (Bottom) Tab of the Ribbon

2.4. Navigating in a 3D Viewport

Once you have created a 3D space, it is crucial that you know how to navigate in this space. Knowing how to work with the space will save you time and effort when you are building your 3D model. For example, it may be much easier to look at the model from a top view to add details to a roof of a house then from the 3D perspective view.

To navigate your way through a 3D space, you will need to take advantage of the Explorer and use your mouse. In the next few sections, we will detail how each can help you navigate and manage your 3D space.

2.4.1. The Explorer

When working with 3D objects in GraphWorX64, you will find the Explorer to be especially helpful. It has a lot of little tools that allow you to quickly interact with your design and increase your productivity. The Explorer, shown in Figure 2.17, gives you an easy way to drill down into graphics and access information. Its hierarchical outline shows the content of a display in a tree-like structure, with branches that you can expand and collapse depending on the level of details you want to see. You can move objects in the Explorer, select them, and group them more easily when there are many objects on the display.

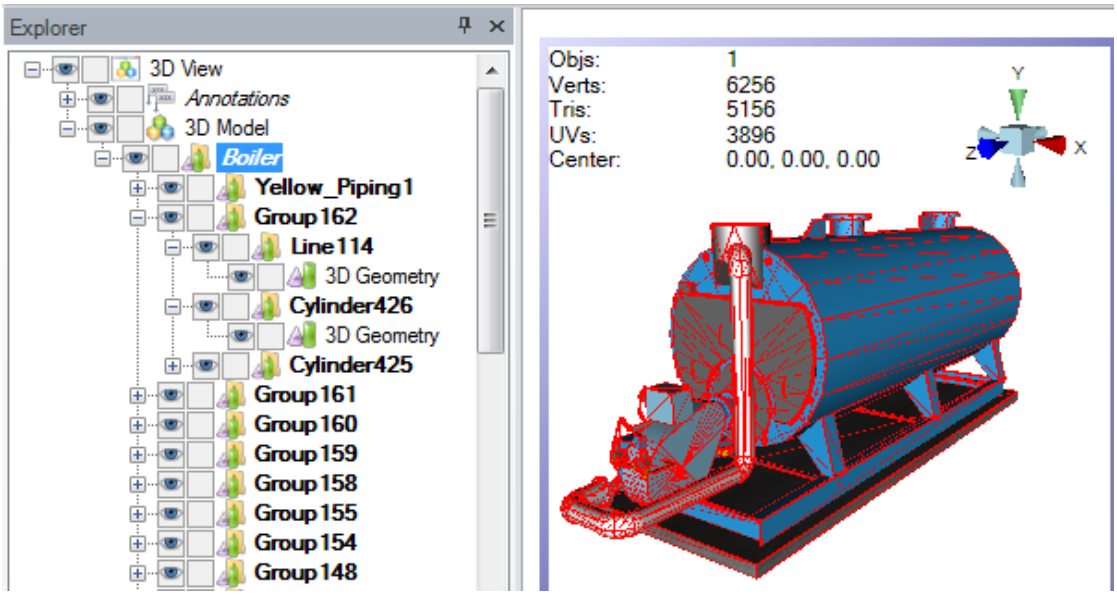

Figure 2.17 - Select Objects in the Explorer

There are three types of geometries used in 3D GraphWorX64, and you will see them displayed in the Explorer (and in the example shown in Figure 2.17). The three types are **3D Model**, **3D Group**, and **3D Geometry**. Any 3D object created in GraphWorX64

is a *3D Model*. A 3D Model can contain a 3D Group or 3D Geometry, while a 3D Group can contain only a 3D Geometry – no other combinations are possible. The 3D Model and 3D Group are geometries for grouping objects. 3D Geometry is an object imported from an external application.

In the Explorer, you can quickly change the placement of an object from one group to another by dragging it, and you can quickly reorganize objects and groups. Once you have completed the design and definition of an object, you can then use the Explorer to lock it to prevent further editing.

2.4.2. Mouse Actions

In addition to the static View buttons on the 3D Home ribbon, you can use the mouse to navigate in a 3D viewport and glide through the scene changing the perspective however you like. You will need a three-buttoned mouse, or a mouse with a wheel. It works to your advantage if you have a mouse with a wheel. Figure 2.18 illustrates how to use the mouse buttons to zoom, pan, and orbit. Let's take a closer look at these features.

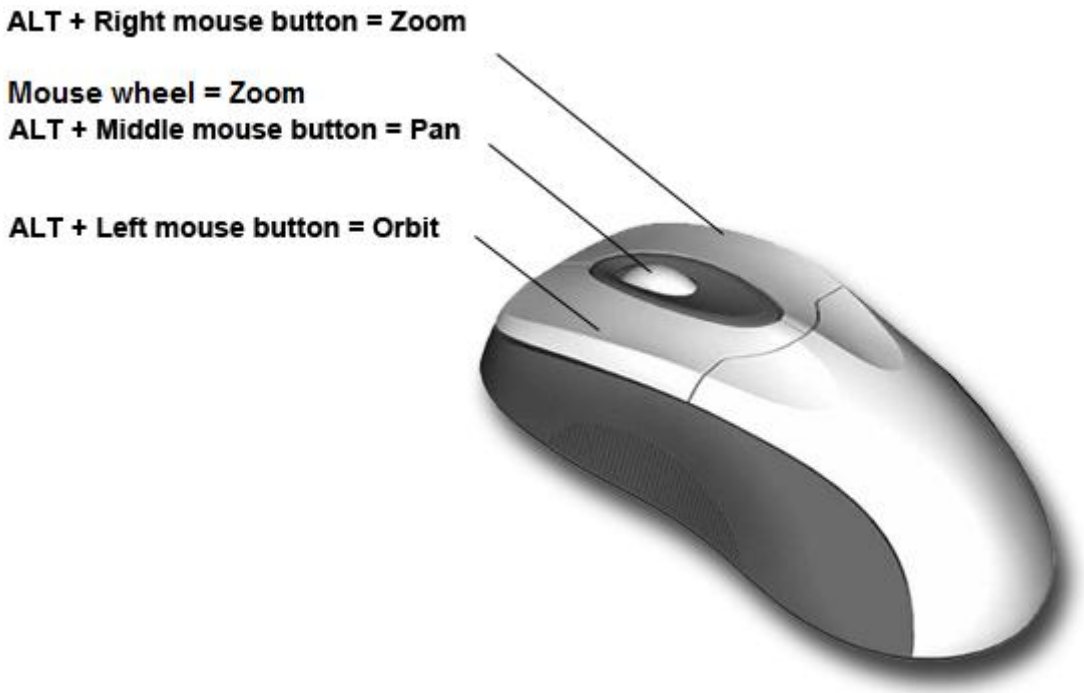

Figure 2.18 - Mouse Buttons Used for Navigating 3D Views

2.4.3. Orbiting

Orbiting lets you change the perspective so that you can see an object from different angles. Hold down the **Alt** key, then hold down the ***left*** mouse button and drag the mouse in any direction to tumble or rotate the camera's view around a particular center of interest to achieve a higher or lower vantage point, or a different side angle, as demonstrated in Figure 2.19.

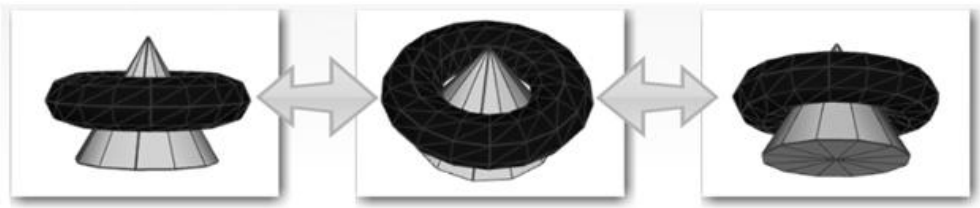

Figure 2.19 - How Orbiting Can Change Your Perspective

NOTE: The camera's view is the viewpoint from which you can see the 3D object.

2.4.4. Panning

Panning lets you scan the width and depth of your scene. Hold down the **Alt** key, then hold down the ***middle*** mouse button (or the ***wheel***) and drag the mouse in any direction to move the camera up, down, or sideways in relation to the scene, as demonstrated below in Figure 2.20.

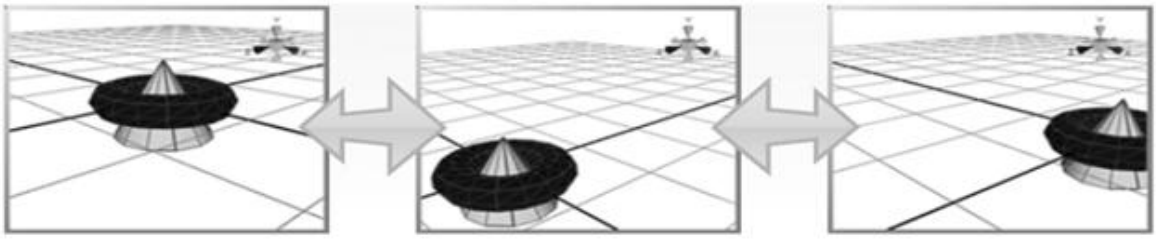

Figure 2.20 - How Panning Can Change Your Perspective

Panning lets you scroll to areas that are beyond the boundaries of the 3D viewport. The example in Figure 2.21 shows only one corner of a campus of buildings. To see other parts of the campus you could zoom out but you would lose the detail; panning would let you see the detail by simply scrolling to other areas of the campus.

Figure 2.21 – Viewport in which the Display Extends beyond the Boundaries of the Viewport

2.4.5. Zooming

Zooming lets you view the items in your scene closer or farther away. Hold down the **Alt** key, and then hold down the ***right*** mouse button and drag the mouse in any direction to zoom, as demonstrated below in Figure 2.22. Zooming in and out in this way can be tricky because of the speed with which the zoom happens. Alternatively, if your mouse has a wheel, ***rolling the mouse wheel*** backward and forward will zoom in and out in a more controlled way.

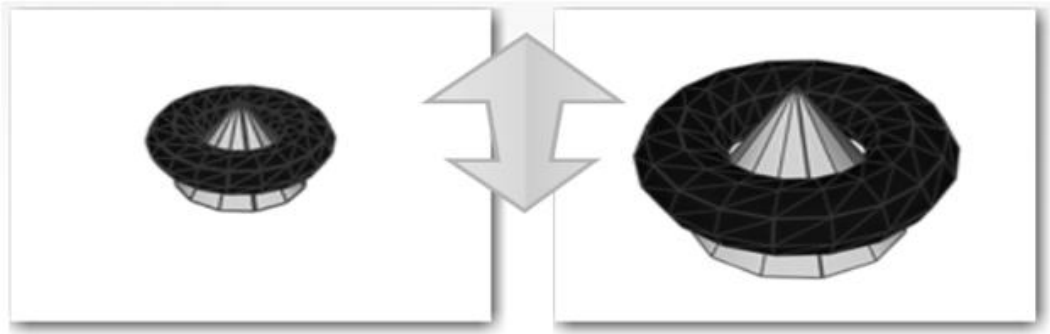

Figure 2.22 - How Zooming Out and in Can Change Your Perspective

2.4.6. Seeing a 3D View from Any Angle

Once you have mastered the use of your mouse to zoom, pan, and orbit, you can see your view from any angle. Viewing objects from different angle gives you the perspectives you need for lining up objects and moving them into place. Figure 2.24 and Figure 2.24 give you an idea of the usefulness of being able to move freely around in a 3D viewport. These examples show how changing the perspective helps you to find and fix misalignments.

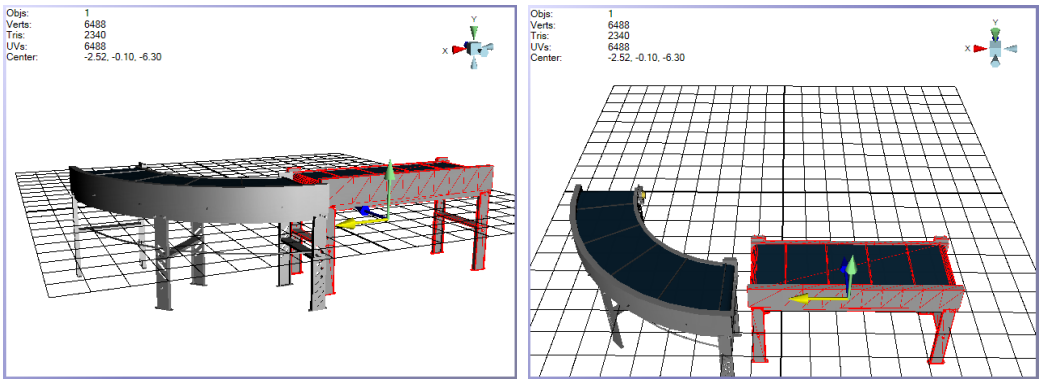

Figure 2.23 – Identifying Misalignments

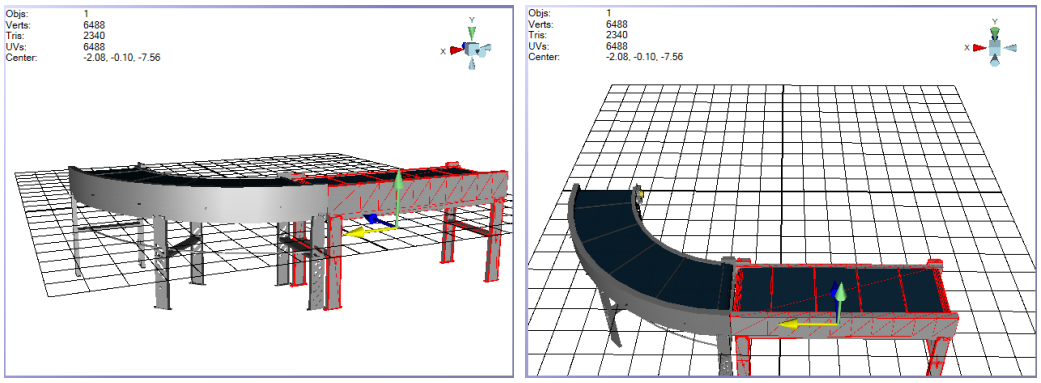

Figure 2.24 – The Same Two Perspectives with the Objects Aligned

Lab: Navigating the 3D Space L2

Throughout this course, we will try to gain some experience with the tools available in GrpahWorX64 3D environment. At the end of the course, we will create a 3D display for icTonics.

Estimated time to complete this lab: 20 minutes

2.1. Task Checklist

In this lab, you will complete the following tasks:

- Insert a 3D object from the Symbols Library
- Practice selecting different parts of an object
- Practice navigating through the 3D space

2.2. Support Files

You will need the following files to complete this lab:

- icTonicsTemplate.tdfx

2.3. Working with an Object in a 3D Space

Before we start building a 3D display for icTonics, we will take some time to get to know how to work in a 3D space.

1. Open GraphWorX64 from Start → Programs → ICONICS GENESIS64 → GraphWorX64 → GraphWorX. Alternatively, you can open a GraphWorX64 display in the Workbench.

2. Open a new display.

3. Click on the 3D View button in the Shapes section of the Home tab of the ribbon.

4. In the work area of the display, click in the upper left-hand corner and drag the mouse to the lower right-hand corner. You should now have a 3D viewport in the work area as shown in Figure L2.1.

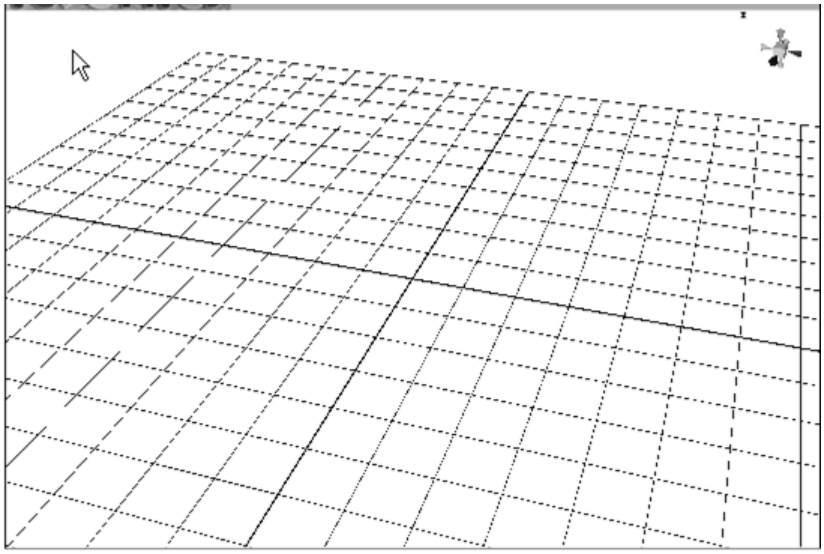

Figure L2.1 - A 3D Viewport in a Display

5. Save the display as "Practice1.gdfx". You should remember to save the displays occasionally as we go along.

   Note: The Save feature is not available while the 3D viewport is active. To inactivate the viewport, double-click outside its boundaries. Then you can save the file using File - Save. To reactivate the 3D viewport, double-click inside its boundaries.

6. Change to the Top view by clicking on the Top button in the View section of the Home tab in the 3D ribbon.

7. Open the Symbols Library and expand the Conveyors > 3D folder and double-click on Conveyors.

8. Select the Conveyor Continuous Turntable symbol and drag it into your viewport.

9. Go into the Object Explorer; you'll see that you now have a 3D Model under the 3D View. Change the name of the 3D Model to "Turntable" so that we can refer back to it with ease later.

10. For now, that is all we will add to the display. Select the Perspective button to change the view into perspective. Your turntable should look very different from the top view.

11. Select the other views to look at the turntable from different angles. This will give you a better understand of what it looks like. Make sure you end in the perspective view.

12. Now, we are going to rotate the view freely so that you can look at the turntable from any angle. Hold down the ALT key, and drag the mouse around inside the 3D viewport. This will let you change the perspective to see the turntable in any way that you want. While you rotate the 3D viewport, take note of how the axis legend always orients itself to the X, Y, and Z axes.

13. You can also zoom in and out in the viewport. Scroll the mouse toward yourself to zoom out and scroll away from yourself to zoom in. Alternately, you can use the zoom slider at the bottom right of the Workbench to zoom in and out.

14. In the Explorer, expand Turntable > 3D Model > 3D Group and you will see many different 3D Geometries inside this group. Try selecting the different geometries in the tree. The selected object will be highlighted in red on the display. If you don't see anything, try zooming in or rotating to a different angle in the 3D viewport to see if you can find the selected object.

15. Double-click outside the viewport and save the display.

Review: The 3D Space R2

2.1. Chapter Objectives

You should now know and understand the following topics:

- The 3D viewport
- Activating and deactivating a 3D viewport
- Orientation and the X, Y, and Z axes
- Perspectives
- Grid options
- The 3D viewport's ribbon
- How to navigate in a viewport

2.2. Summary

When you work with 3D objects in GraphWorX64, you work in a viewport on the design surface. The viewport is a sub-area that opens up when working with a 3D object in a 2D space, or when working in a 3D view. You can have multiple viewports in a display.

You activate a viewport by double-clicking on it; you deactivate it by double-clicking outside the viewport. When the viewport is activated, the 3D ribbon appears; when the viewport is inactivated, the regular GraphWorX64 2D ribbon appears. Although the 3D ribbon is similar to the 2D ribbon, it contains different options that are available only inside the 3D space. For example, scene information provides details that will help you improve the runtime performance of the display.

The thing to remember when working inside the viewport is that it has three dimensions which are represented by three axes in the viewport: the X-, Y-, and Z-axes. A legend in the viewport always shows the true orientation of the axes.

Buttons on the Home tab of the 3D ribbon let you change the perspective of the scene in the viewport. You can also use the mouse to orbit around objects, pan from one place to another, and zoom in to see detail and zoom out for a bird's eye perspective inside the viewport.

It is important to keep the Explorer open while you work. This allows you to easily select the 3D objects that you want to edit. It also allows you to easily group and regroup different objects.

2.3. Terms

2D	3D	Axis	Camera View
Grid	Orbit	Pan	Perspective
Plane	Scene	View	Viewport
Zoom			

2.4. Test Your Knowledge

In this chapter, we discussed how you can work within a 3D space. Before continuing to the next chapter, make sure you can answer the following questions.

1. How many viewports can you open in a GraphWorX64 display?

   a. Only one
   b. It depends on the size of the display
   c. Five
   d. Unlimited

2. True or False: You can use the mouse to tumble and scan the scene in the viewport.

3. What is the quickest way see a scene from above?

   a. Press Alt + the left mouse button to orbit above the scene
   b. Define a keyboard shortcut
   c. Click on the Top button on the Home ribbon
   d. Just press Enter
   e. All of the above
   f. None of the above

4. Is there a way to move left and right in the scene?

   a. Yes, use the orbit feature on the mouse.
   b. Yes, use the pan feature on the mouse.
   c. Yes, use the zoom feature on the mouse.
   d. No, there is no way to move left and right in the scene.

5. What is the purpose of the legend that displays in the top right corner of the viewport?

   a. To show the orientation of the X, Y, and Z axes
   b. To identify the planes you are working with in the 3D space
   c. To help you orient yourself after you have changed the perspective of the scene
   d. All of the above
   e. None of the above

6. What is the easiest way to tell the number of objects you have in a viewport?

   a. Count the objects by looking at the viewport.
   b. Count the objects in the Explorer.
   c. Select the view in the Explorer and look at Objs property in the Scene Information.
   d. Go to Tools → Count objects.
   e. There is no way to tell.

7. True or False: There is a button on the 3D Home ribbon that centers the selected object in the view.

8. What grid features are available in a 3D space?

   a. You can show or hide a grid
   b. There is one grid for each plane
   c. You can use the Snap to Grid feature to align objects in the space
   d. There is a Snap to Angle feature for use when rotating objects
   e. You can change a grid's cell size
   f. You can change a grid's color
   g. All of the above
   h. Some of the above; circle all that are true: A B C D E F

9. How do you know if the 3D viewport is activated:

   a. The axis legend displays in it
   b. The viewport has a solid blue border
   c. The 3D ribbon appears
   d. All of the above
   e. A and C
   f. B and C
   g. A and B

10. True or False: You can hide the scene information that appears in the upper left corner of a 3D viewport.

Primitives 3

The purpose of this chapter is to show you how to create a 3D object using the primitive shapes in GraphWorX64. Once you have a grasp of the fundamental concepts presented here, you can apply the same concepts to pre-fabricated models that you import or use from the 3D Symbols Library.

3.1. Objectives

After completing this chapter, you should have an understanding of:

- The primitive shapes that are available in GraphWorX64 3D
- The properties of primitives
- Annotations

3.2. Introduction to Primitives

Primitives are three-dimensional geometric shapes that are the building blocks for creating complex geometric objects, which you can use in GraphWorX64 to create 3D objects from scratch. GraphWorX64 primitive shapes are: the cube, the sphere, the plane, the polygon, the donut, the cone, the cylinder, terrain, and the pipe .

Many 3D modelers begin with primitives as a basic starting point for their models. This technique is referred to as *primitive-up modeling*. You can modify the basic shape of a primitive and change the attributes of basic primitives to make them more complex or less complex.

Use the **Primitives** section, shown in Figure 3.1, in the **Home** tab of the 3D ribbon to insert or draw different basic 3D shapes.

Figure 3.1 - Primitives Section

Note: We will look at the annotation primitive at the end of this chapter.

Once you have added one of these shapes to the 3D space, you can use its properties to create the look you need. You can also move, rotate, and scale the shape and apply images, textures, and colors to the surface of the shape.

3.3. Common Primitive Properties

Let us begin by looking at the common properties that most shapes have, and then we will talk about the primitives individually.

Once you create a primitive, in the primitive's Properties explorer you will find two sections: **Appearance** and **Common**. The *Common* section of the primitive's properties is shown in Figure 3.2. In this section, you can enter a name and description for the shape. If you are working in Advanced application mode, you can also specify the description that shows in the object's tooltip, custom data, whether the shape is visible in the 3D scene, and whether the shape is locked in the display. All primitives have these same common properties.

Common	
(Name)	
Description	
DescriptionInToolTip	Default
CustomData	
Visible	True
Locked	False

Figure 3.2 - Common Properties of a Cylinder (in Advanced Application Mode)

Other properties for the primitives are in the *Appearance* section. The most frequently changed properties are the **Material** and the **BackMaterial** properties, which determine the color and texture of the primitive's outside and inside surfaces. You can change these properties by using the Materials tab on the 3D ribbon.

There are other Appearance properties that primitives commonly have, too. We will look at each of these properties next.

3.3.1. Segments for Height, Width, and Sides

Primitive shapes have many properties that you can control. **Height**, **width**, and the number of **sides** are Appearance properties whose use is quite obvious, but in GraphWorX64 3D you can also control the **number of segments** that make up the height, width, and sides of an object. The number of segments can determine an

object's shape. For example, an object with fewer segments may have discernable sides compared to one that has many segments which makes it appear smooth or rounded. While more segments give you a smoother surface, fewer segments give you faster rendering.

For example, the shapes in Figure 3.3 are both spheres. The one on the left has ten sides and four height segments, while the one on the right has 30 sides and 10 height segments. Notice the vast difference between them because of the number of segments.

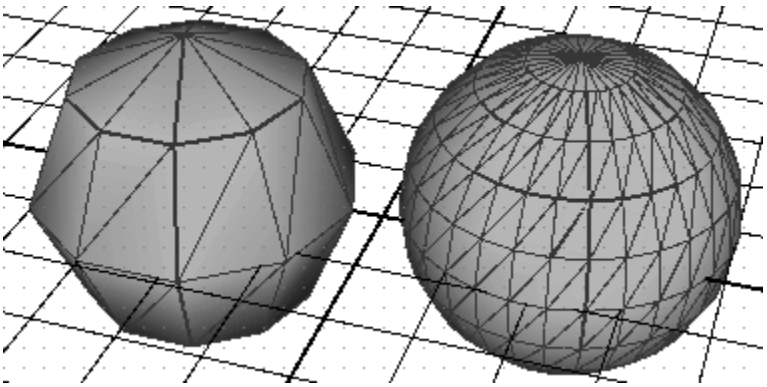

Figure 3.3 - Two Spheres with Different Numbers of Sides and Height Segments

Tip! Adding segments adds points and surfaces to the object. And adding points and surfaces adds memory consumption. If you raise the number of surfaces on a sphere from 60 to 500, the sphere consumes more memory. It is important that you understand this if you are animating in real-time, because the power needed to animate an object increases with the quality of the graphics. Higher quality means that more power is required. This means you have to make decisions about the number of segments you use in a model and consider how much detail you can have without overwhelming runtime performance.

3.3.2. Slice Angles

For many primitives, you can define a **Slice Angle** property to create a wedge shape or to create a shape with a wedge taken from it, as shown in Figure 3.4. For the Slice Angle property, you define the degree of angle you want for the shape. The range you can enter is from 0 (no shape) to 360 (a complete shape). Figure 3.4 shows two cylinders with different slice angles to give the illusion that one is a wedge that has been removed from the other. You can define a slice angle for cylinder, sphere, polygon, donut, and cone primitives.

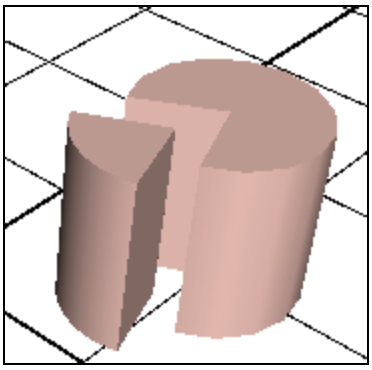

Figure 3.4 - Two Cylinders: Slice Angles of 80 Degrees (Left) and 280 Degrees (Right)

3.3.3. Caps

The **Cap Top** and **Cap Bottom** properties determine whether a shape is closed off at the top or bottom. If the cap property is set to **True**, the shape appears to be solid or *capped*, like the wedge in Figure 3.5; set to **False**, the shape appears to be empty or has no cap. Typically, a cap set to False reveals the inside of the shape. Since by default the inside is *transparent*, an uncapped shape will reveal invisible insides, like the uncapped cylinder in on the left in Figure 3.5. You can give it an inside color by using the shape's **BackMaterial** property, as shown on the right in Figure 3.5.

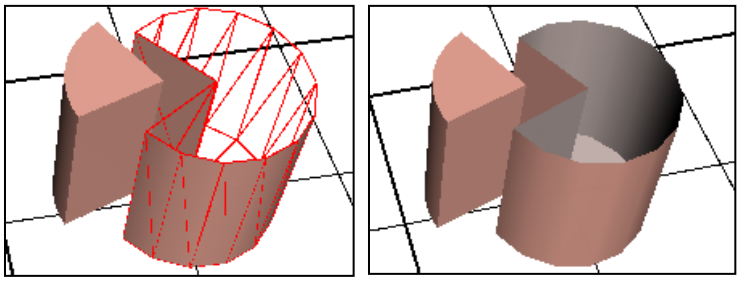

Figure 3.5 - Capped and Uncapped Cylinder Shapes

3.3.4. The Radius

Some 3D primitives have a **Radius** property. The radius defines the width of a round or spherical shape. But some shapes have *two* Radius properties, usually called **Radius1** and **Radius2**. For cylinders, donuts, and polygons, defining a *second* radius adds a hole in the middle of the shape and defines the radius of the hole. For cones, the two Radius properties identify the radii of the top and bottom of the cone.

> **NOTE:** Where there are two Radius properties, the larger value defines the outer radius, and the smaller value defines the inner radius.

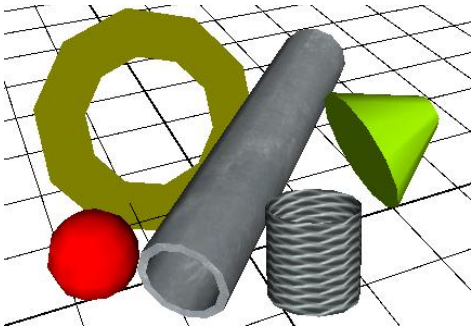

Figure 3.6 - Shapes that Have Radius Properties

Now that we have seen the basic properties of each primitive in GraphWorX64 3D, let's look at each primitive and the shapes they can create. We will start with the simpler primitives, and work toward the more complex ones.

3.3.5. Preferences for Primitives' Properties

You may find that there are some primitives whose properties you are always changing. For example, maybe you are always changing the Material property of the Pipe primitive. Or you are creating a model that requires all Spheres to share the same number of segments. You can set default properties for most types of primitives by using the Preferences Explorer.

3.4. The Plane Primitive

The plane is simply a flat surface in a three-dimensional space. It is a two-dimensional shape that has a front and a back. You can set different colors for each side. Add it to your 3D view by clicking on the plane icon in the Primitives section.

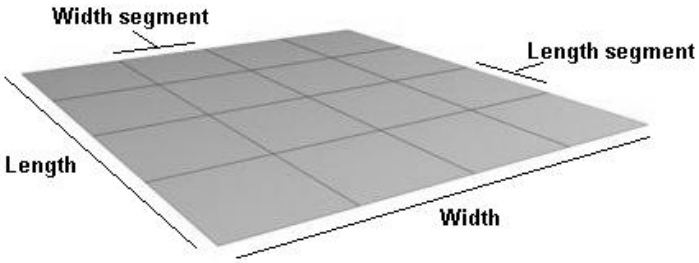

Figure 3.7 - The Plane Primitive

The plane primitive has an additional property you should be aware of: the Direction property. Changing the **Direction** property changes the XYZ coordinates of the plane in the 3D space. By default, the Direction property is 0, 1, 0. When you change the coordinates of the Direction property, you change the angle of the plane in the space.

You can accomplish the same thing by rotating the plane using the Rotate button in the Manipulator section of the Home tab, as you will see in the next chapter.

3.5. The Polygon Primitive

Like the plane, the polygon is a two-dimensional shape that has a front and a back; it is a shape that is confined within a plane. You can set different colors for each side of the polygon. Inside its plane, the polygon has three or more sides; the more sides it has, the rounder its appearance becomes.

The polygon's shape is defined by three-dimensional points (*vertices*) and the straight lines that connect them (*edges*). The interior region of the polygon is called the *face*. Add it to your 3D view by clicking the polygon icon in the Primitives section. A polygon can take on any number of shapes, including the ones in Figure 3.8.

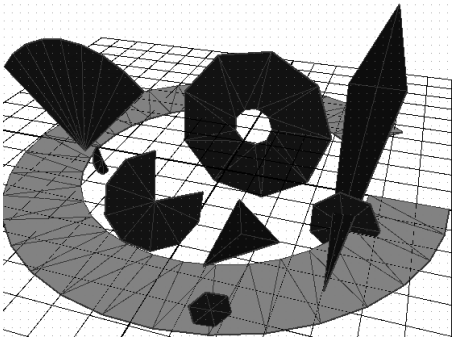

Figure 3.8 - Polygons

3.6. The Cube Primitive

The cube is a box shape. Add it to your 3D view by clicking on the cube icon in the Primitives section. Use its properties to create the cube shape you want.

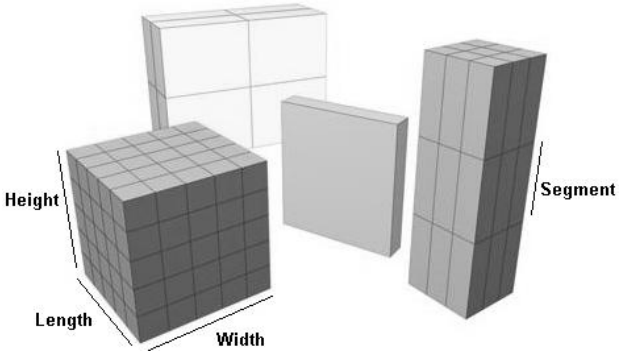

Figure 3.9 - The Cube primitive

3.7. The Sphere Primitive

You can add a sphere primitive to your 3D view by clicking on the sphere icon in the Primitives section. The sphere is essentially a globe or ball shape, but the shapes you create with a sphere do not necessarily have to be round. By varying the numbers of sides and height segments, you can change the sphere to create shapes like those shown in Figure 3.10.

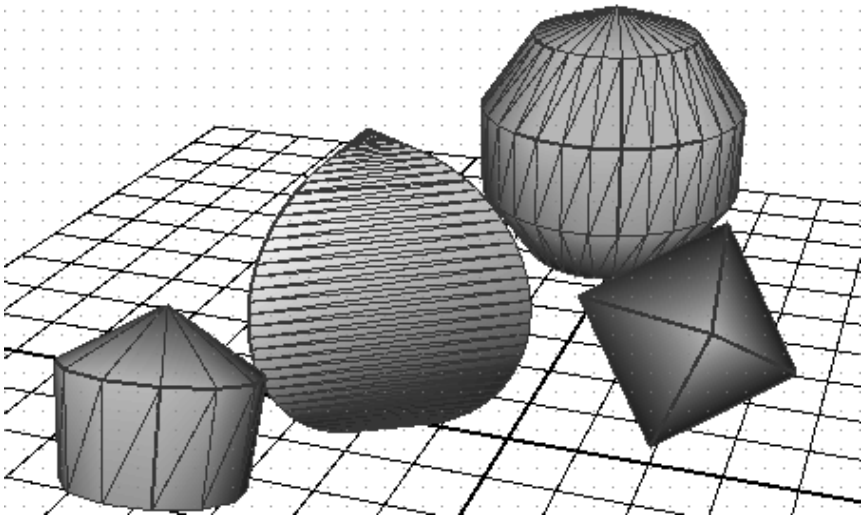

Figure 3.10 - Sphere Shapes with Different Heights and Number of Sides

In addition to the properties mentioned at the beginning of this chapter, a sphere has properties called **Hemisphere** and **Hemisphere2**, which define the two hemispheres of the sphere. Use these properties to truncate a sphere's shape. The range for each hemisphere property is from 0 to 1. Figure 3.11 shows three sphere primitives and their different hemisphere properties.

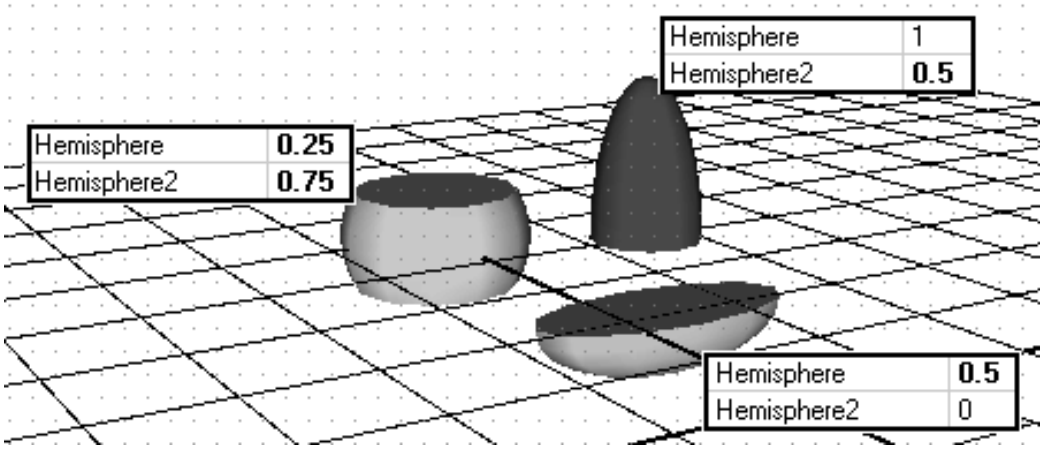

Figure 3.11 - Three Sphere Primitives with Various Hemisphere Properties

3.8. The Cylinder Primitive

The cylinder typically looks like a supporting column in a building, but depending on its number of sides it can also look more like a pentagon or octagon.

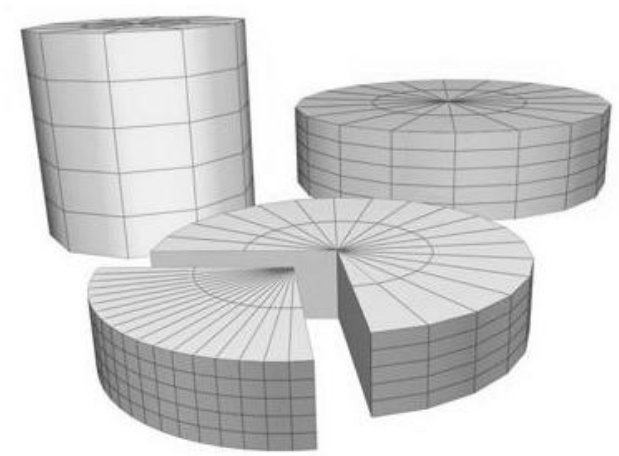

Figure 3.12 - The Cylinder Primitive

You can add it to your 3D view by clicking the cylinder icon in the Primitives section. You can change the properties of a cylinder to make it look like the examples in Figure 3.13. Note that one of the shapes in Figure 3.13 is made up of two merged cylinders. You can merge primitives by moving them into the same space and grouping them. We will look at how to group objects in the next chapter.

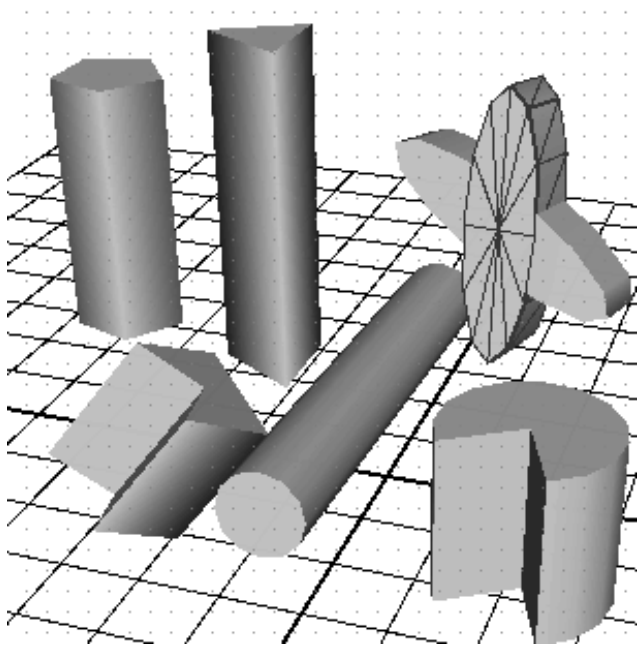

Figure 3.13 - Samples Using the Cylinder Primitive

You can also create a tube in your 3D view by using the cylinder primitive and setting the Radius2 property. The tube primitive is similar to a cylinder except that it has a hole through its middle and can be used to create arcs. Figure 3.14 illustrates some examples of a tube shape. Note, however, that if you are creating pipes, there is a pipe primitive, which we will look at soon.

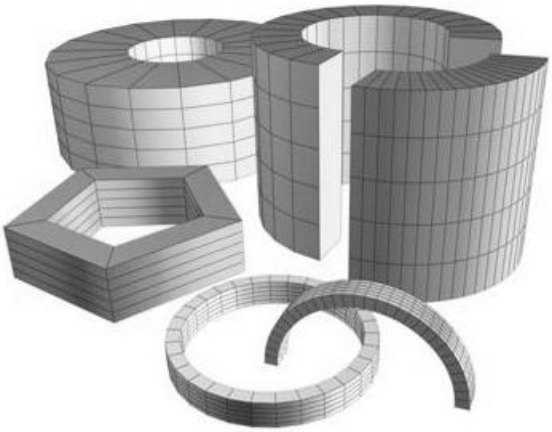

Figure 3.14 - Some Tubes created using the Cylinder Primitive

3.9. The Cone Primitive

The cone primitive is similar to a cylinder except that its sides are not parallel. Figure 3.15 shows some traditional examples of a cone shape.

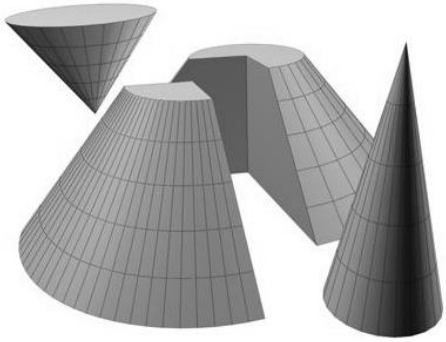

Figure 3.15 - The Cone Primitive

You can add a cone primitive to your 3D view by clicking the cone icon in the Primitives section. Figure 3.16 shows some examples of the variety you can create.

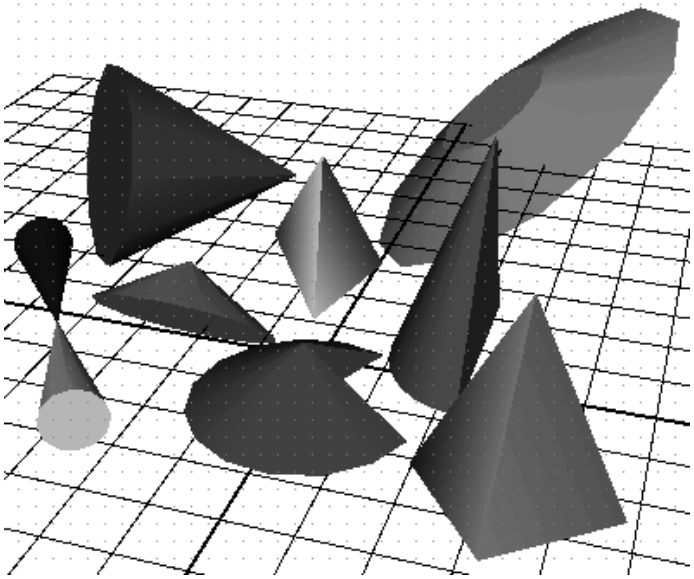

Figure 3.16 - Additional Cone Primitives

3.10. The Donut Primitive

You can add a donut primitive to your 3D view by clicking the donut icon in the Primitives section. The donut, typically a ring shape, can be reshaped to look quite different. Figure 3.17 illustrates some examples.

Note: The donut primitive is sometimes called a *torus* in the GraphWorX64 interface.

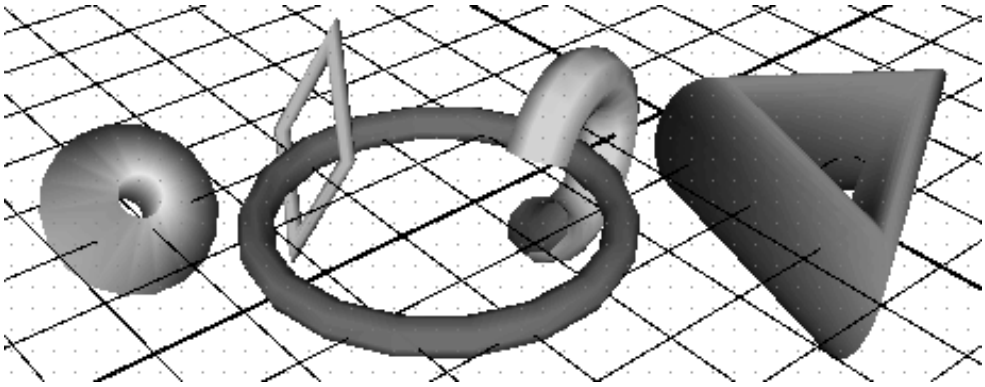

Figure 3.17 - Five Donut Primitives

3.11. The Pipe Primitive

You can add a pipe primitive to your 3D view by clicking the pipe icon in the Primitives section. The pipe primitive is unique in that you can easily manipulate it to create turns and bends. Figure 3.17 illustrates some examples.

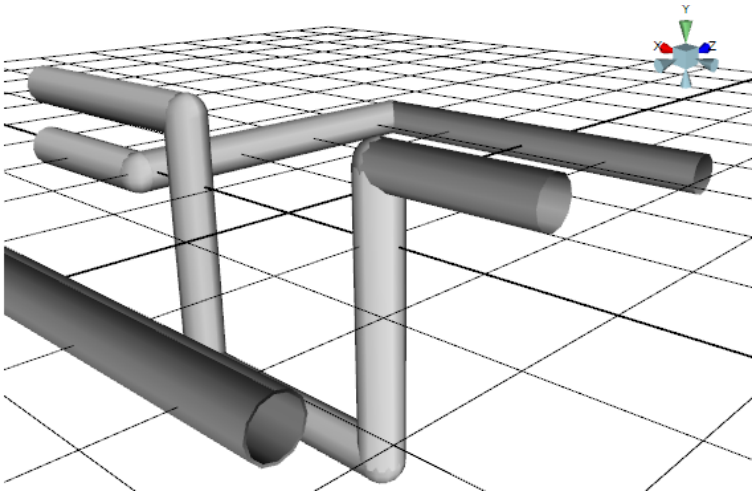

Figure 3.18 - A Few Pipes

The pipe primitive works a lot like the pipe control in 2D GraphWorX64. Double-clicking anywhere in the pipe creates a handle with a Move manipulator that you can grab for creating a bend, shown in Figure 3.19. Each end of the pipe also has a Move manipulator so you can move the ends easily. We will look at the Move manipulator in more detail in the next chapter.

You can use the Snap to Grid feature to control the angles at which the pipe bends. The grid's cell size will affect the length of pipe between bends if you are using the Snap to Grid feature. Each bend's XYZ coordinates are written to the pipe primitive's **Points** property. You can manually adjust the coordinates to specify their exact coordinates and to force pipes to align exactly inside the 3D grid.

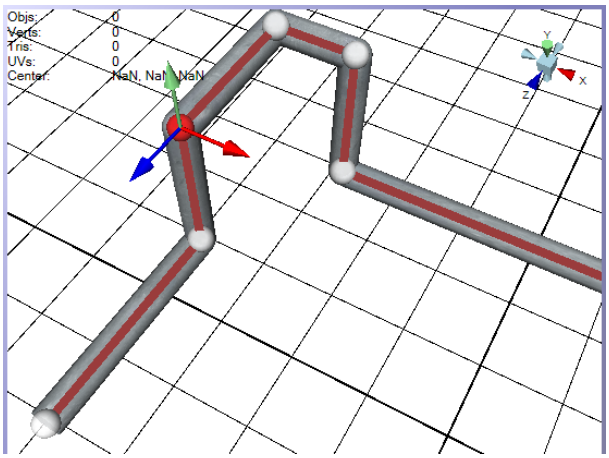

Figure 3.19 - Bending a Pipe

You can also easily animate an object so that, during runtime, the object moves along the path of a 3D pipe. We will look at this particular feature later when we work with 3D dynamics.

3.12. The Terrain Primitive

By default, the Terrain primitive looks like the terrain of hills or mountains, like the example in Figure 3.20; you can use it to achieve a highly-faceted surface that reflects light and creates shadows. Beyond that, it has very limited use in this release of GraphWorX64. With that in mind, let us take a quick look at its current capabilities.

You can add a Terrain primitive to your 3D view by clicking on the terrain icon in the Primitives section. Terrain is a two-sided shape that has a front and a back; you can set different colors for each side.

Note that you can create individual terrains and connect them in a view.

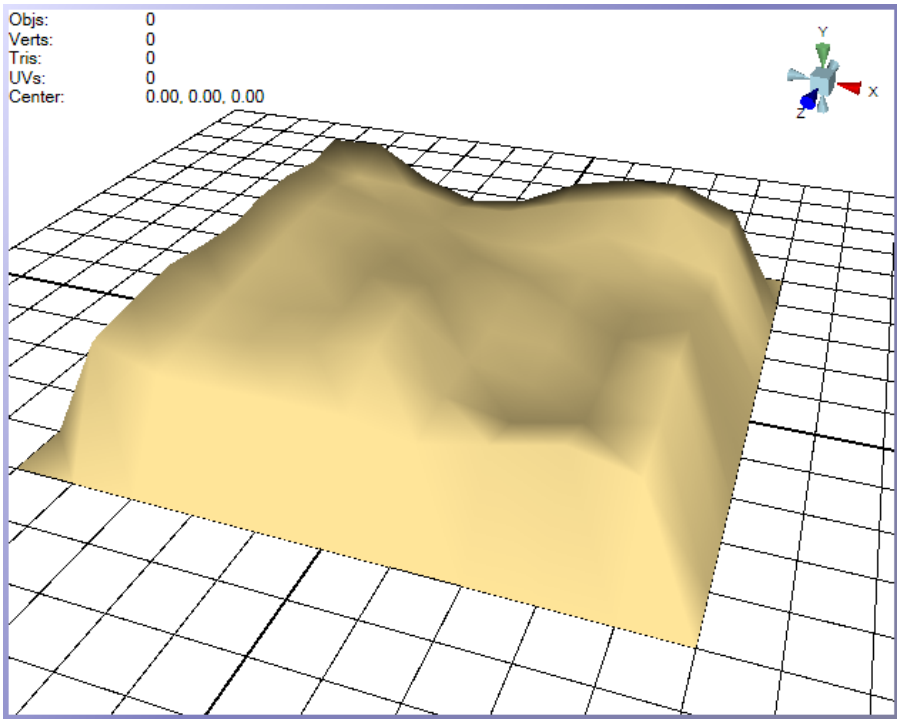

Figure 3.20 - The Terrain primitive in its Default Shape

You can use the primitive's **ElevationBitmap** property to import a bitmap that GraphWorX64 can use to generate the terrain you want. The elevation bitmap generates the terrain's *relief profile*, but it does not paint the terrain. To paint the terrain, you must apply a *texture* to it – the texture can be a bitmap of the actual terrain, if you have it.

If you do not have an elevation bitmap, you can generate a random terrain. Use the **RandomSeed** property to enter a number that generates different random profiles for the terrain. The **WidthSegments** and **LengthSegments** properties also affect the detail of the terrain profile. Figure 3.21 shows the terrain primitive from above so you can see the segments and how they help to define the surface.

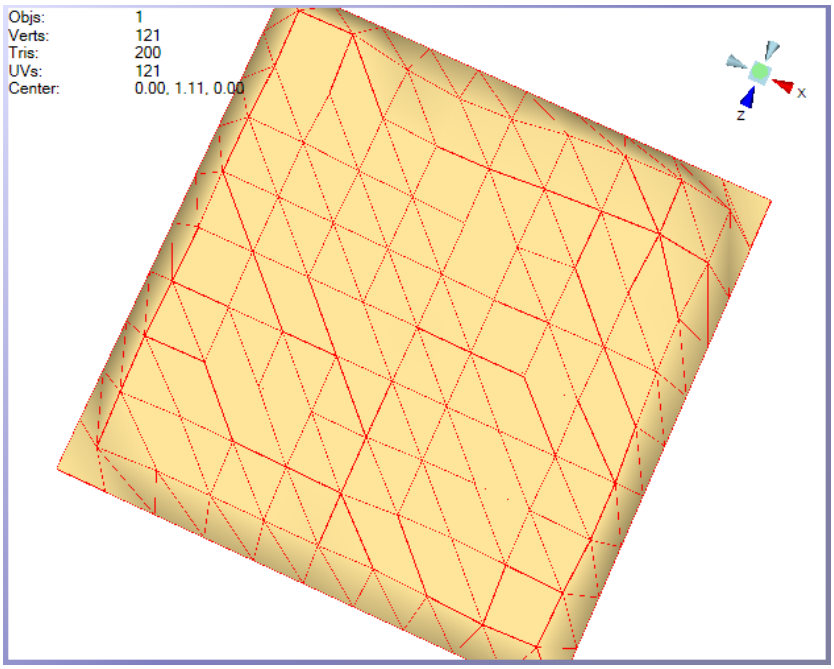

Figure 3.21 - A Terrain's Top View, Showing its Segments

The **VerticalDisplacement** property sets the maximum vertical disparity of peaks to valleys when randomly generating the terrain. The default value of the **BordersDisplacement** property is False, which prevents the edges of the terrain from being displaced vertically. The **Direction** property is made up of the X, Y, and Z coordinates for the angle of the terrain in space. You can type coordinates for the property, or you can use the Rotate button on the Home tab, which we will look at more closely in the next chapter.

3.13. Annotations

An annotation is a graphic label that remains in place when a 3D display or object rotates or moves in a 3D scene. Each annotation is anchored to the object that it labels, as shown in Figure 3.22. For example, you may want to be able to easily monitor a gauge on a tank. So you can create an annotation for the tank, then add a gauge (from the Symbols Library) to the annotation and create a pointer to an animated tank object, which the gauge monitors; when runtime users rotate the animated tank, the gauge remains steady so that the users can always read it.

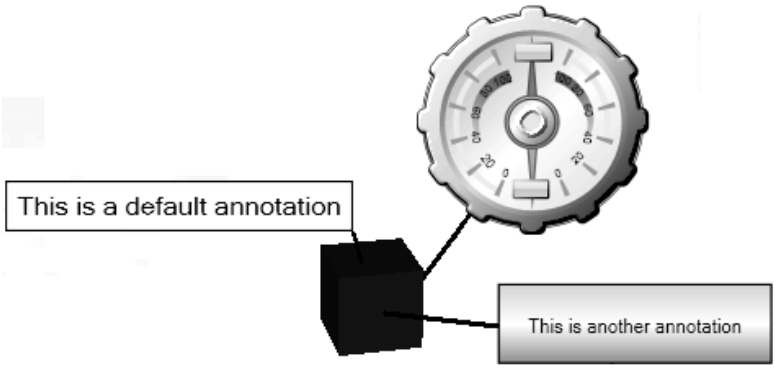

Figure 3.22 - Some Simple Annotations

NOTE: Annotations appear in their own special section in the tree structure in the Explorer, and not as children of the objects they're attached to, as shown in Figure 3.23. Notice the *Annotations* branch, and that the two annotations are not part of the cube and sphere primitives in the Explorer.

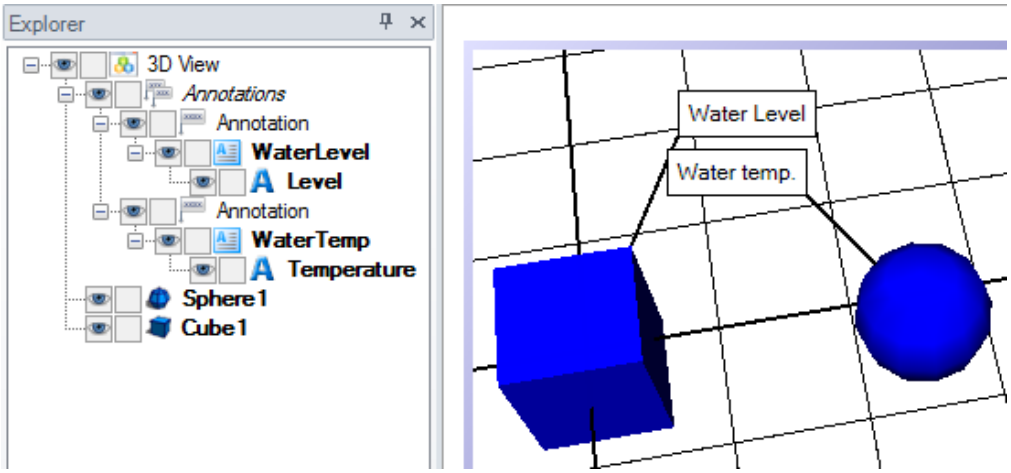

Figure 3.23 - Annotations in the Explorer

You can put anything you like into an annotation. For example, the annotation can have a gauge, but it can also have an animated object, a trend, and even an alarm viewer. You are not limited to what you put into the annotation.

You can create annotations so that their display measures are culture- and language-sensitive. Using one display with one set of annotations, you can set up the annotations so that, for example, a temperature gauge shows readings in Celsius to users in Europe and in Fahrenheit to users in the United States. The same display's annotations can also be language-sensitive to users.

To create an annotation, select the object you want to annotate, and then click on the Annotate icon in the Primitives section on the Home tab. This places an annotation in the 3D view and attaches the annotation to the selected object, as shown in Figure 3.24.

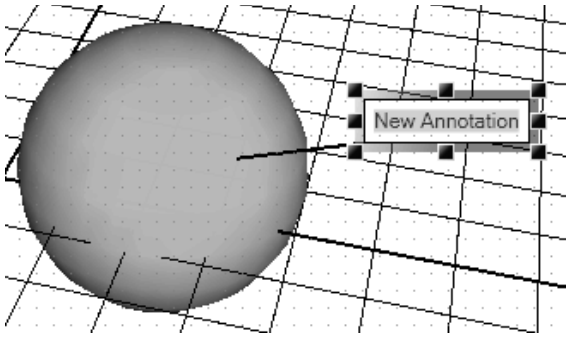

Figure 3.24 - An Annotation Added to a Sphere Primitive

Type the text of the annotation or add an object, such as a 3D symbol from the Symbols Library, a GraphWorX64 object, or an AlarmWorX64 object. Configure any objects in the annotation as you would configure them for any other display.

Editing an annotation can be confusing at first. Until you become accustomed to annotations, the following tips may be of some help:

- To exit from an annotation editing mode, press the **Esc** key or ***double-click*** outside the annotation. This will return you to the 3D editing mode that you were in before you edited the annotation.
- To edit an annotation after it has been created, double-click the annotation content.
- If you single-click an unselected annotation, the move manipulator appears, as shown in Figure 3.25; you can move the annotation as you would any other object. Note that this behavior happens only when the annotation is not currently selected. If you want to move an annotation that is selected, exit the annotation's editing mode and then single-click the annotation.

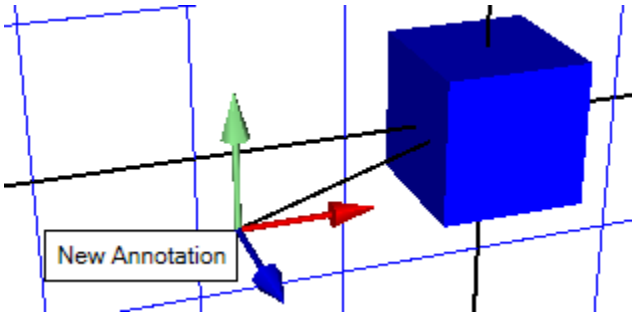

Figure 3.25 - Moving an Annotation

- An annotation has two sets of properties. Select the *border* of the annotation to set **Annotation** properties; select the *inside area* (for example, the text) to set **Label** properties.

- When you are editing an annotation, the ***2D*** GraphWorX64 ribbon displays. The space inside the annotation is 2D, but you can add a 3D symbols and 3D objects to it, which will create a 3D viewport inside the annotation.
- As shown above, all annotations appear in the Annotations section in the Explorer. When selecting an annotation from the Explorer, you can see which object it is attached to by looking at the annotation's **AssociatedObject** property.

Among the many properties you can set for an annotation are the AnchorPoint and Position properties, shown in Figure 3.26. The **AnchorPoint** identifies the coordinates on the object where the line is anchored; the **Position** identifies the coordinates on the annotation box. These properties are visible on the **Annotation** properties only (when you single-click an annotation that's in edit mode).

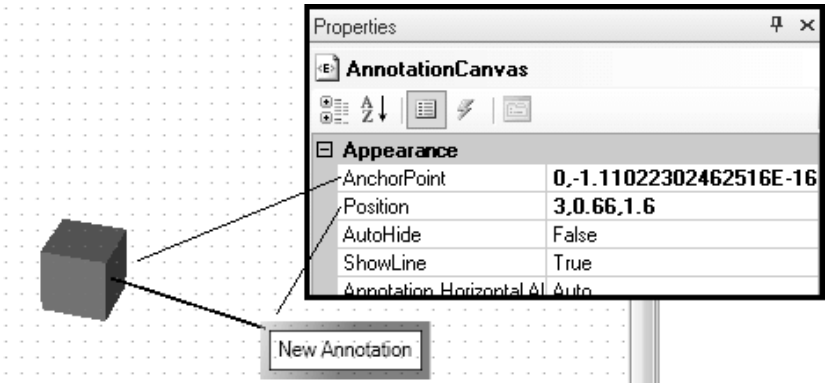

Figure 3.26 - Anchor Properties of AnchorPoint and Position

NOTE: If changes to these properties are not reflected immediately on the graphic, rotate the scene around and the line position will update.

Lab: icTonics Tank L3

During this lab, we will practice working with the different primitives. We use 3D primitives to create a tank for the icTonics 3D display that we will build at the end of the course.

Estimated time to complete this lab: 20 minutes

3.1. Task Checklist

In this lab, you will complete the following tasks:

- Insert primitives
- Work with slice angles
- Change the Hemisphere properties

3.2. Support Files

You will need the following files to complete this lab:

- Practice1.gdfx

3.3. Inserting Primitives

In this lab, we will insert the primitives necessary to build our tank. In the next lab, we will look at how to put these parts together to create the tank.

1. Open the Practice1.gdfx display if you closed it from the previous lab.
2. Double-click on the 3D viewport to activate it. We will clear out the objects in the viewport first. Click on the File menu button, and select New.
3. From the Home tab of the ribbon, add a Cylinder primitive to the display. This will become the body of our tank.
4. Notice that in the Explorer, you now have a Cylinder object. In the Properties explorer, go to the Appearance section.

5. Set the Radius2 property to 0.4 and set the Slice Angle property to 270. This will give us an opening on the tank so we can see what is in it. The properties of the object should look similar to Figure L3.1.

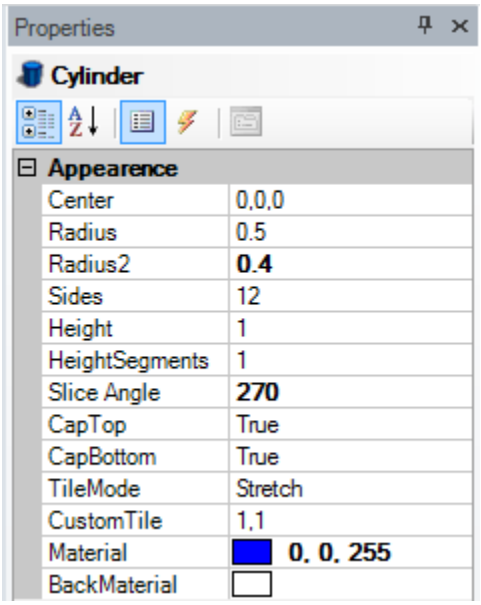

Figure L3.1 - Cylinder Properties

6. Next, add a Sphere primitive onto your display. The center for this primitive is the same as your cylinder. If you know how to, you can move the object aside to see it better and manipulate it, or you can just leave it there and manipulate its properties. We will learn more about moving objects around in the next lab.

7. In the sphere's Appearance properties, change the Hemisphere field to 0.5. You should now have something that looks like a bowl.

8. Insert the cone primitive onto the display and change the Height of the cone to 0.5.

9. You should now have three primitives in your display. Save the display.

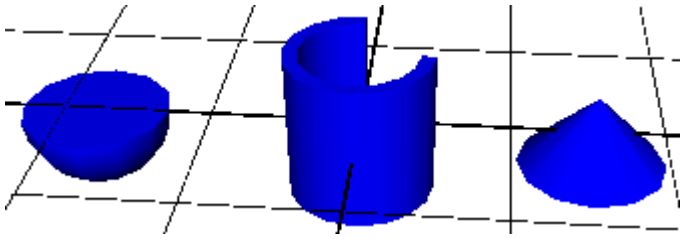

Figure L3.2 - Inserted Primitives

Review: Primitives R3

3.1. Chapter Objectives

You should now know and understand the following topics:

- The primitive shapes that are available in GraphWorX64 3D
- The properties of primitives
- Annotations

3.2. Summary

Primitives are three-dimensional geometric shapes that are the building blocks for creating complex geometric objects in GraphWorX64 3D views. Primitives that are available are the cube, sphere, plane, polygone, donut, cone, cylindar, terrain, and pipe. You can modify the basic attributes of primitives to change their shapes and appearance.

Use the Primitives section of the Home tab in the 3D ribbon to insert basic 3D shapes.

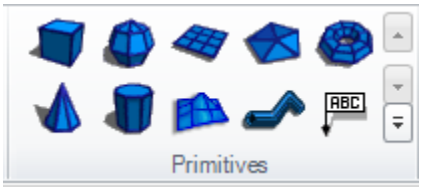

Figure R3.1 - Primitives Section

Primitive shapes have many properties that you can control. **Height**, **width**, and the number of **sides** are the most obvious properties of any shape, but in GraphWorX64 3D you can also control the number of **segments** that make up the height, width, and sides of an object. The number of segments can determine an object's shape, controlling if the shape has discernable sides or is smooth or rounded in appearance. More segments give you a smoother surface, however, fewer segments give you faster rendering. Adding segments adds points and surfaces to the object and consumes more memory.

For many primitive shapes, you can define a **Slice Angle** property to create a wedge shape or to create a shape with a wedge taken from it. The **Cap Top** and **Cap Bottom** properties determine whether a shape is closed off at the top or bottom. Some shapes have a **Radius** property. The radius defines the width of a round or spherical shape. But

some shapes have two Radius properties, usually called **Radius1** and **Radius2**. For cylinders, donuts, and polygons, defining a second radius adds a hole in the middle of the shape and defines the radius of the hole. For cones, the two Radius properties identify the radii of the top and bottom of the cone.

An annotation is a graphic label that remains in place when a 3D display or object rotates or moves in a 3D scene. Inside an annotation is a 2D scene where you can use any 2D GraphWorX64 feature to add any objects that are available in 2D, including symbols from the Symbols Library, 3D viewports, and controls.

3.3. Terms

Annotations	BackMaterial Property	Caps
Cone Primitive	Cube Primitive	Cylinder Primitive
Donut Primitive	Edges	Face
Hemisphere	Material Property	Pipe Primitive
Plane Primitive	Primitives	Polygon Primitive
Radius	Segments	Slice Angle
Sphere Primitive	Terrain Primitive	Vertices

3.4. Test Your Knowledge

In this chapter, we discussed the primitives that are available in 3D GraphWorX64. Before you continue in this course, make sure you can answer the following questions.

1. What is significant about the number of segments in a primitive?

    a. The number of segments refines the detail of a primitive
    b. The number of segments can change the shape of a primitive
    c. The number of segments can affect the rendering speed
    d. All of the above
    e. None of the above

2. How do you define the inside color of a shape?

    a. Turn off caps
    b. Use the BackMaterial property
    c. Shapes have no inside color
    d. Use the InsideColor property

3. If a primitive has two Radius properties, what can be the function of the second Radius property?

   a. It adds another object
   b. It alters the pivot point
   c. It creates a hole in some objects and defines the radius of the hole
   d. It defines the radius of the other end of a cone
   e. Both A and B
   f. Both C and D
   g. None of the above

4. Which of the primitives are in two-dimensions only? (Choose all that apply)

   a. Sphere
   b. Cube
   c. Cylinder
   d. Plane
   e. Cone
   f. Donut
   g. Polygon
   h. Terrain
   i. Pipe
   j. None of these

5. What is the purpose of an elevation bitmap?

   a. To create a fourth dimension in a 3D view
   b. To create a texture for a Terrain primitive
   c. To create a background image for a 3D view
   d. To create the relief definition of a Terrain primitive

6. What is it about annotations that makes them useful in 3D?

   a. Annotations always face the user, regardless of the user's perspective
   b. Annotations label objects
   c. Annotations can display dynamic objects and symbols
   d. Annotations can display any ICONICS object
   e. All of the above
   f. None of the above

7. Which property do you use to create a wedge-shaped primitive?

   a. The BackMaterial property
   b. The Cut menu item
   c. The Hemisphere property
   d. The Slice Angle property
   e. None of the above

8. True or False: By default, the inside surface of a 3D primitive is the same color as the object's outside surface.

9. True or False: You can merge and group primitive shapes to form new, complex shapes.

10. True or False: In the object Explorer, each annotation is listed in the tree structure as a component of the object it is anchored to.

Manipulating Objects 4

The purpose of this chapter is to show you how to manipulate 3D objects to create a static 3D scene. There are several 3D features that let you move, resize, rotate, arrange, duplicate, and transform objects with precision.

4.1. Objectives

After completing this chapter, you should be familiar with:

- Opening a 3D object
- Ways to select 3D objects
- Ways to arrange 3D objects
- Grouping and ungrouping objects
- 3D coordinates
- The pivot point
- Ways to manipulate objects by moving them, rotating them, scaling them, shaping them, transforming them, and duplicating them
- Applying colors and other materials to object surfaces

4.2. Opening a 3D Object

To open a 3D object, expand the Object Explorer to display the object from an existing view or from the Symbols Library, and then drag the 3D object onto the design surface, as shown in Figure 4.1. Note that some objects contain animation and dynamics may take a few seconds to open.

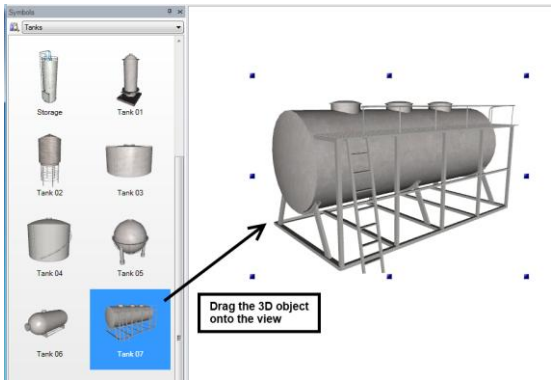

Figure 4.1 - Dragging a Symbol onto the Design Surface

Once you have dragged an object onto the view, you may want to move it, rotate it, or give it a different scale. Before we get to that, you must be able to select it in the 3D viewport. So let us talk briefly about how to select 3D objects and their components.

4.3. Selecting 3D Objects

In GraphWorX64 3D, there are multiple pointers available that let you select objects as well as faces, vertices, edges, and triangles. You can even use a cursor that selects types of objects, too.

Before you can select an object, a Select button must be activated. The Select buttons are located in the Selection Mode section of the Home tab, as shown in Figure 4.2. Then you select the object in the Object Explorer or by clicking on it in the viewport. In a complex model, it may is easier to select an object or component in the Object Explorer.

NOTE: If an object is locked, you cannot select it in the Object Explorer. An object is locked if its Locked property is true or if in the Explorer the lock icon appears next to the object.

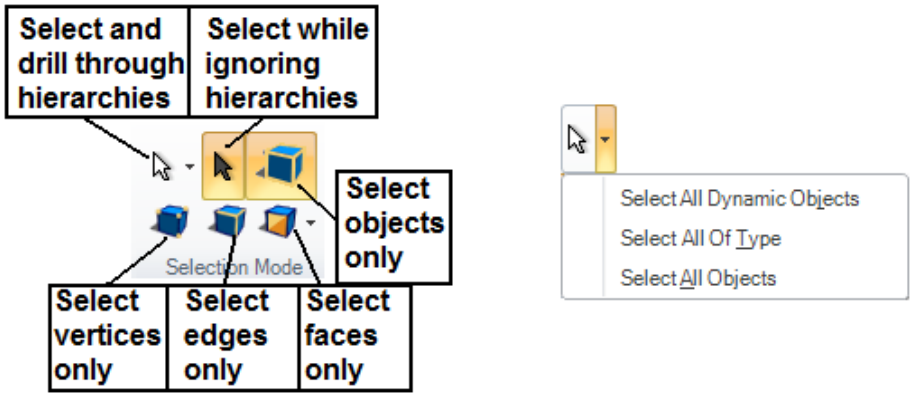

Figure 4.2 - Selection Modes in the 3D Viewport (Left) and the Hierarchical Selector (Right)

But before you can select anything, you need to select the right cursor. There are two sets of Selection Mode icons you need to be aware of. As you can see in Figure 4.2, the two ―arrow" icons give you the option of using using a cursor that does or doesn't drill through hierarchical groups when you click in the viewport. *One of these two arrow icons must be active in order for you to select.*

Of the remaining four ―cube" icons, it is likely that you will most-frequently use the top icon to select objects only. If you are designing models for use in the 3D scene, then you may want to use the three icons on the bottom row. We will look at them in a minute, but for now it is only important that you know that *one of these four cube icons must also be activated.*

4.3.1. Selecting Components

Because a 3D object typically contains a complex construction of many small graphical objects, selecting it or its components can be a tricky task. To edit a 3D object, you typically need to manipulate components or smaller parts of a bigger object such as a leg on a tank or a step on a ladder. For this reason, the ability to select this component is very important.

By expanding objects in the Explorer, you can single out the component you want and select it in the Explorer tree; you will see it selected in the viewport. Conversely, if you do not know the name of a component, you can select it in the viewport, which also selects it in the Explorer. You can see a selected object easily because it is highlighted with a red frame, as illustrated in Figure 4.3. You can also use the hierarchical selection icon shown in Figure 4.2 to drill down to lower components in the viewport.

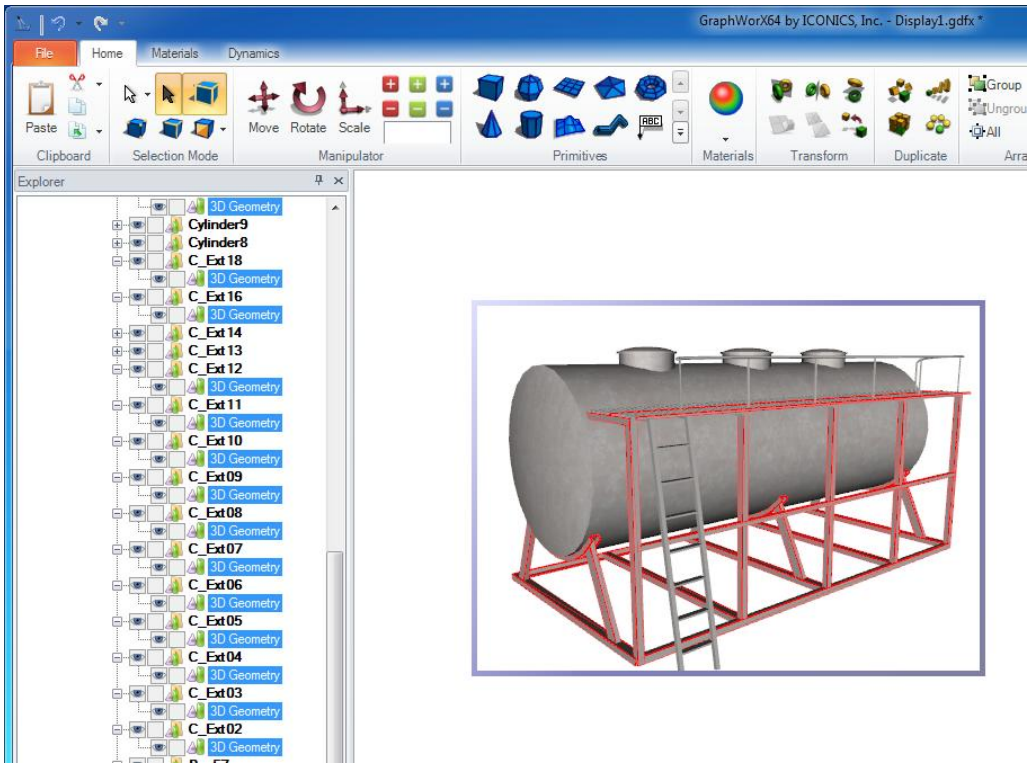

Figure 4.3 - Selected Components in the Explorer (Left) and the Viewport (Right)

Since you may want to be able to select objects and sub-elements easily from the Object Explorer, it is important that you name objects in a way that will be helpful for selecting them. If you give them all the same name (such as Component 1, Component 2, etc.), working with them in the Explorer could become a nightmare. But if you give them meaningful names (such as Handrail, Leg Support 1, etc.), selecting and maintaining them using the Explorer will be much easier.

4.3.2. Selecting Multiple Objects

You can select multiple objects in the Explorer or in the viewport. To select multiple objects in the Explorer, you hold down the **Ctrl** key while clicking on each object, or hold down the **Shift** key while clicking the first and last objects in a range.

If you want to select the objects in the viewport, you must hold down the **Shift** key while clicking on each object you want to select.

When you select multiple objects or sub-elements, the last selected object is the ***key object***, as shown in Figure 4.4. Some tools in the 3D viewport use the key object to determine what to do with the selection. For example, when you transform multiple objects, the transformation uses the *key* object's pivot point.

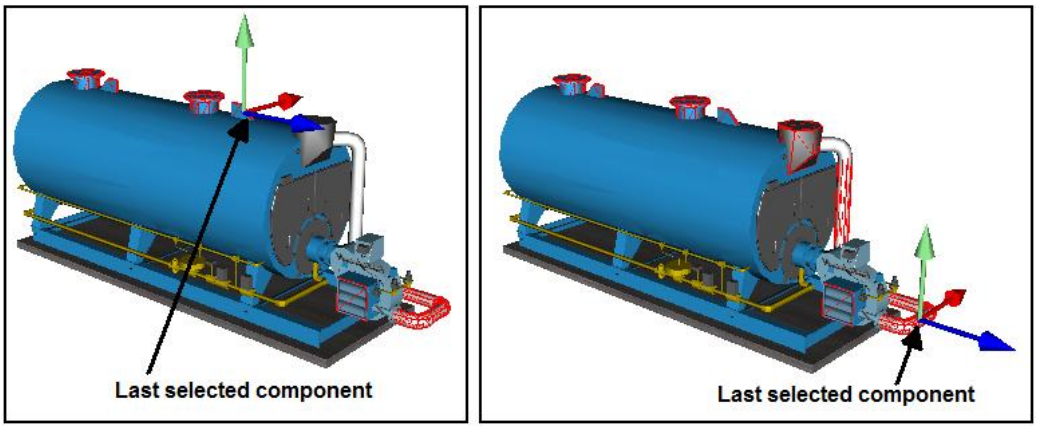

Figure 4.4 - Two Examples of a Highlighted Key Object

4.3.3. Selecting Sub-elements

When creating complex objects and models, you may need access to object faces, vertices, edges, and triangles (or segments). To edit these sub-elements of an object, you must first be able to select them. To do this, you can click on the selection modes that best suits the work you need to do, and then click on the object in the viewport to make selections in that mode. An example of each selection mode is given in Figure 4.6.

- **Select Objects** : The default, this mode selects entire objects, and is the most commonly used mode.

- **Select Vertices Only** : This mode allows you to select individual vertices on a 3D object. This lets you move or delete vertices to change an object's appearance. This mode is intended for use by advanced model designers.

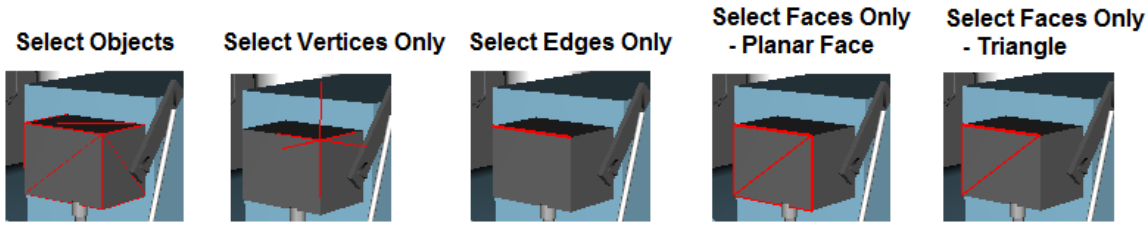

Figure 4.5 - Illustrations of Selected Sub-elements

- **Select Edges Only** : This mode is similar to the Select Vertices Only button. It lets you select individual edges on a 3D object so you can move or delete an object's edges. This mode is for advanced model designers.

- **Select Faces Only** : This mode lets you select individual faces of 3D objects. It is actually a drop-down button (see Figure 4.6) which also lets you choose if you want to select *planar faces* or *triangular sections of faces*. This mode can be particularly helpful if you want to shape a terrain on your own: just start with a flat terrain and select triangle faces to be raised or lowered depending on your needs, to create hills, mountains, lakes, rivers, seas, etc. This mode is also used for advanced model design.

Figure 4.6 - Face selection types

Note: You can also use the Divide Face button to have greater control on an object's face. We'll look at this feature later in this chapter.

4.4. Coordinates in 3D

The most basic visual entity in the 3D space is the ***point***. The point has no size, but it does have a location. To determine the location of a point, we first establish an arbitrary point in space as the origin using the traditional X, Y, and Z axes of geometry. Let's take a closer look at how coordinates work.

4.4.1. X, Y, and Z Axes

In a 3D viewport, we can say that a point's location is:

- A number of units along the X axis
- A number of units along the Y axis
- A number of units along the Z axis

The location of a point is identified using the three dimensions of the X-axis, Y-axis, and Z-axis, as shown in Figure 4.7.

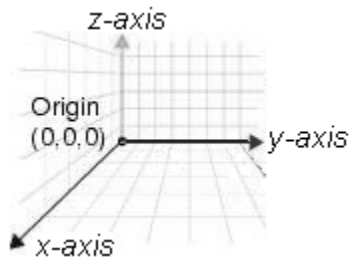

Figure 4.7 - The X-, Y-, and Z-axes

A number represents the point on each axis, and the point in space is expressed as the three axis numbers in XYZ sequence. The point of *origin* is always expressed as (0, 0, 0). From this origin we can count off in space to give us the 3D coordinates of a point in space.

So, looking at the example in Figure 4.8, a point 4 units along the X-axis, 6 units along the Y-axis, and 2 units along the Z-axis, the origin has the XYZ coordinates of (4, 6, 2). Where the three axis coordinates intersect, we have the coordinates in the 3D space.

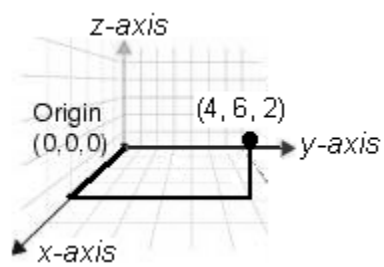

Figure 4.8 - The Coordinates (4, 6, 2) are Plotted in the 3D Space

This example shows points in space between the point of origin and us. To specify points on the *opposite* side of the point of origin (for example, going away from us), we use negative numbers; for example (-5, -2, -1).

4.4.2. World Space, Object Space, and Local Space

3D positions and transformations exist within coordinate systems called spaces. ***World space*** is the coordinate system for the entire scene. Its origin is at the center of the scene. Figure 4.9 shows the world space axes.

Object space is the coordinate system from an object's point of view. The origin of object space is at the object's pivot point, and its axes are rotated with the object, as shown in Figure 4.9.

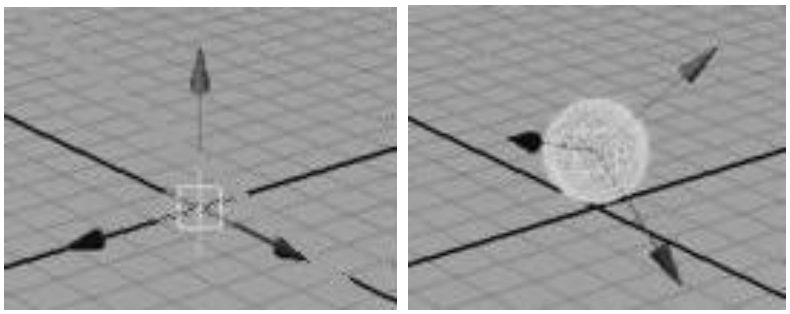

Figure 4.9 - World Space (Left) and Object Space (Right)

Local space is similar to object space, but it uses the origin and axes of the object's parent node in the hierarchy of objects. This is useful when you have not transformed the object in question, but the object belongs to a group that is transformed.

4.5. Arranging 3D Objects in Space

Arranging objects in the 3D space can be a challenging task. GraphWorX64 offers a series of tools for arranging objects. These tools are conveniently located in the Home tab of the ribbon in the Arrange section, shown in Figure 4.10. Let's take a look at these tools.

Figure 4.10 - The Arrange Section

4.5.1. Grouping and Ungrouping

Creating groups lets you manage a selection of graphical components as one entity, all at once rather than one at a time. For example, you can assign properties to a group, select a group in the Object Explorer as shown in Figure 4.11, and manipulate a group. Furthermore, you can lock a group, hide a group, and specify other properties and dynamics for a group.

To group objects together, select the objects, and then click on the Group button. Once the objects are grouped, you can select them as a group by selecting the group's name in the Explorer; you cannot select a group directly in the viewport. You will find each group in the Object Explorer with its objects nested under it in the navigation tree, as shown for the group called Tank 10 in Figure 4.11. To ungroup an object that is grouped, select the group in the Object Explorer and then click on the ungroup button.

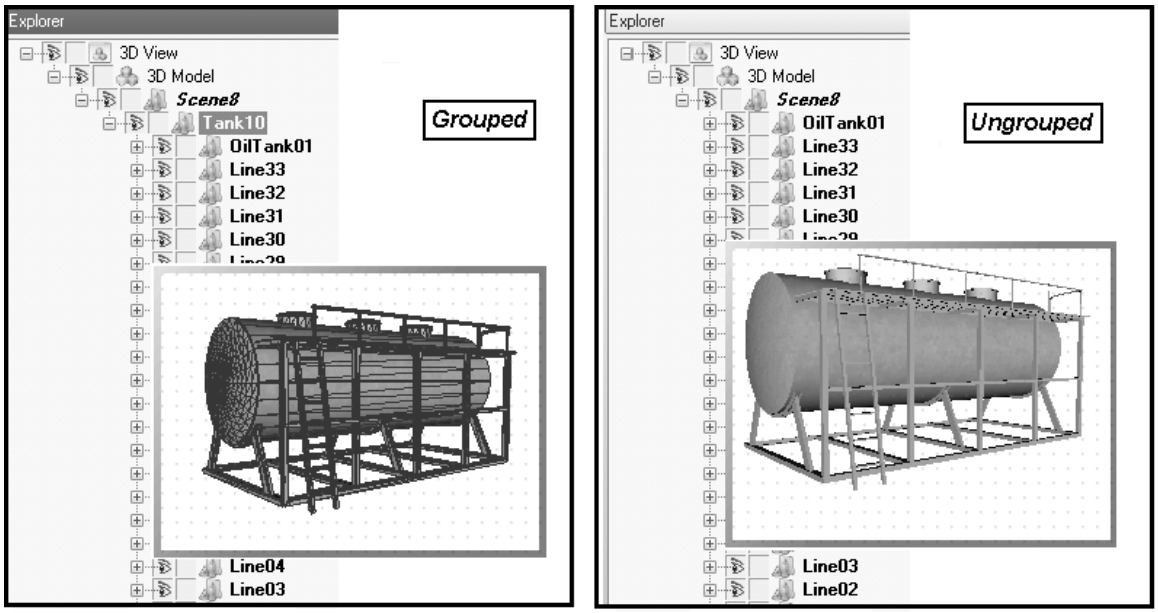

Figure 4.11 - Grouped (Left) and Ungrouped (Right) Components

4.5.2. Centering and Resizing Multiple Objects at Once

From the Arrange section you can also center selected objects and make them the same size. After you select the objects, click on the **All** button to drop dowm the menu shown in Figure 4.12.

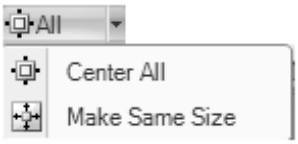

Figure 4.12 - Align All Button

Selecting **Center All** aligns the selected objects around the *key* object's pivot point, as show in Figure 4.13.

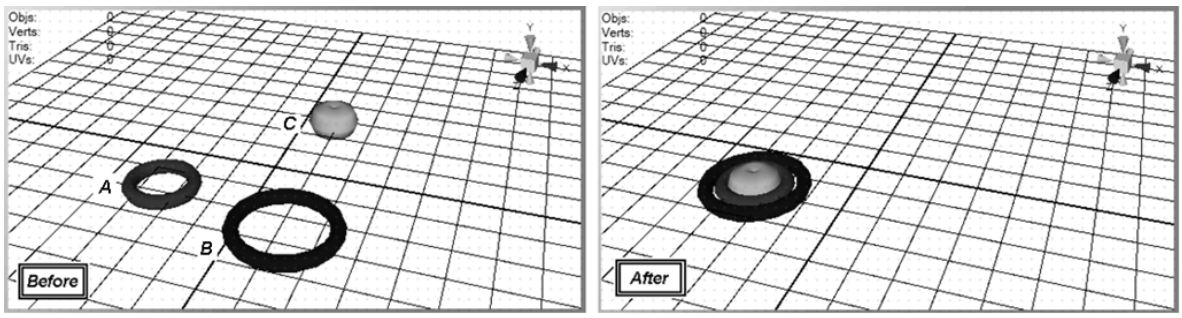

Figure 4.13 - Objects Are Centered Around the Key Object, which Is Ring A

Clicking **Make Same Size** resizes the objects so that their dimensions – that is, their height, width, and depth – match the *key object's* dimensions. The two examples in Figure 4.14 use the same shapes shown in the "Before" part of Figure 4.13.

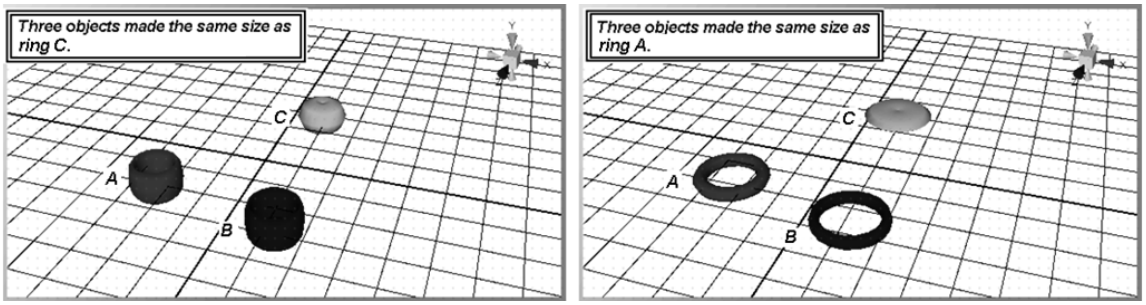

Figure 4.14 - Rings Are Resized to the Key Objects, Ring C (Left) and Ring A (right)

4.5.3. Aligning Objects

It is possible to align two or more objects in the 3D space along one or more axes. To align objects, first select the objects that you want to align; the last object that you select will be the *key object*, meaning all other objects will be aligned to its coordinates. After selecting all the objects and determining the key object, select the axis that you want to align to by clicking on its drop-down menu. The drop-down menu for each axis is shown in Figure 4.15.

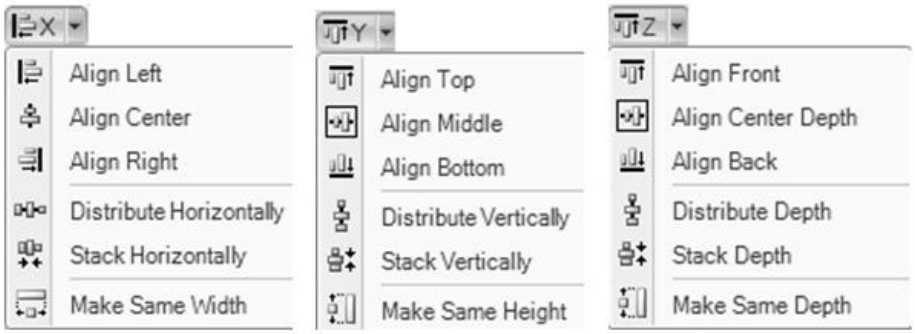

Figure 4.15 - Align Menus for X, Y, and Z

It is also possible to ***stack*** the objects in a pile horizontally along the X-axis, vertically along the Y-axis, or in depth along the Z-axis. In this case, there is no key object. You should first place the objects in their stack order, and then select the option to place the objects adjacent to each other along the selected axis.

You can also ***distribute*** the objects evenly along the axis. In this case, you will need to first place the ~~"~~end" objects at the proper location. This will define the range of space to distribute the selected objects. Using this option will make sure that the space between each of the selected object is identical.

Furthermore, you can ***resize*** the objects so they are the same height, width, or depth as the key object. Note that these selections change the actual shape of each object. This makes it easy to uniform the size of objects, especially when you need to use them to make one bigger object later. If you don't want to change the objects' shapes, you can use transformations, which we'll look at next.

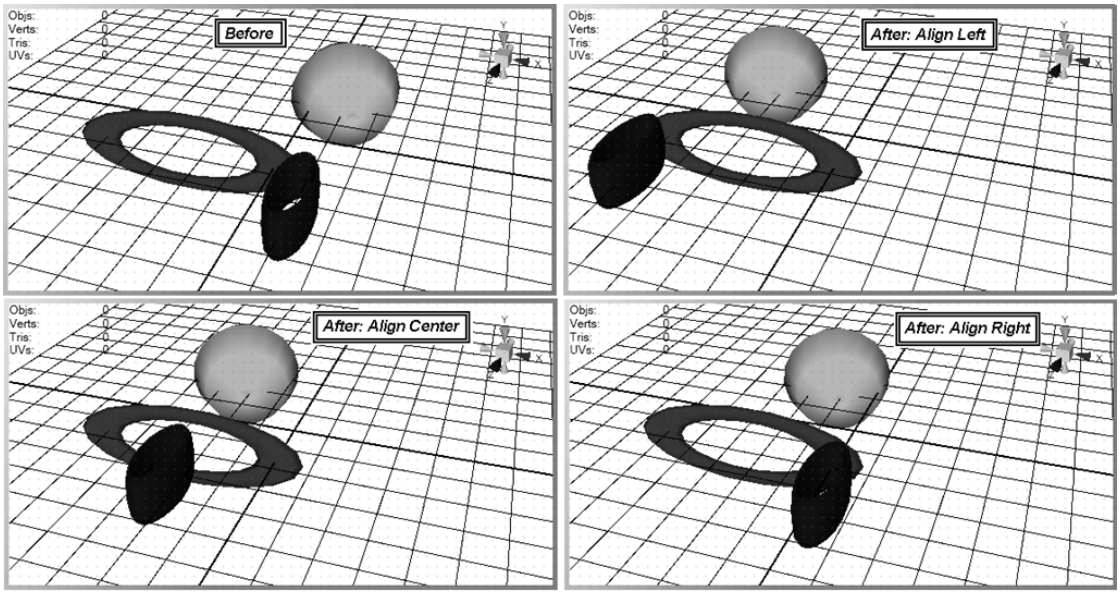

Figure 4.16 - Different Alignments along the X-Axis

4.6. Simple Transformations

Transformations change an object's position, size, and orientation without changing its shape. In terms of 3D graphics, *transforming* means changing an object by moving, rotating, and/or scaling it.

In GraphWorX64, the transformations you make to an object are saved in a transform node. That is, GraphWorX64 remembers that the object is rotated 32,0,5 degrees and moved -3,6.2,7 centimeters from its original position. When you group objects together, each group remembers its own transformations. This lets you create hierarchical animations easily.

Transformations happen in relation to an object's (or component's) pivot point, and take place along or around either the world axis, the object's axis, or a local axis. So let's start by looking at the pivot point of objects.

4.6.1. The Pivot Point

An object's pivot point represents the exact location of the object in space. All transformations of an object are relative to the object's pivot point. By default, the object's pivot point is at its center point which is (0,0,0), but you can change the pivot point of an object by changing the object's **Center** property.

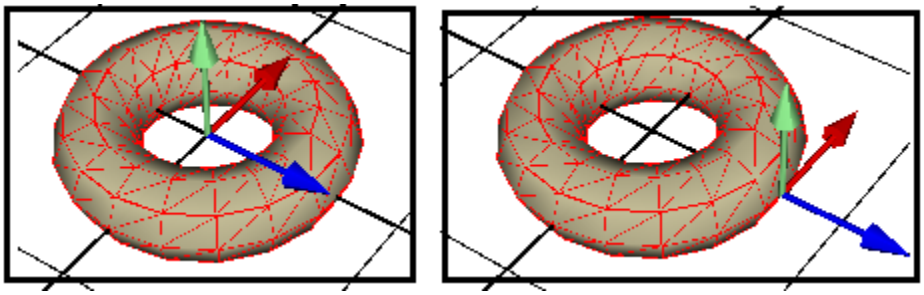

Figure 4.17 - Object with a Pivot Point at its Center (Left) and Moved off Center (Right)

Understanding an object's pivot point is important if you are going to rotate or scale the object. So let's take a look at how you can rotate and scale objects, as well as move them.

4.6.2. Manipulating an Object

The tools for moving, rotating, and scaling objects are the buttons in the Manipulator section on the Home ribbon, as shown in Figure 4.18. Each manipulative action can be

done using the mouse, but you can also use the keyboard and the plus and minus buttons in the Manipulator section.

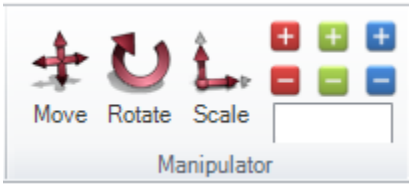

Figure 4.18 - The Manipulator tab group

When a manipulator is active and one or more objects or sub-elements are selected, a manipulator is displayed at the *pivot point* of the selected object, as shown back in Figure 4.18. If *multiple* objects or sub-elements are selected, the manipulator is displayed in the pivot point of the key object (that is, the last selected object), as shown back in Figure 4.4. Let us take a look at each manipulator.

4.6.2.1. The Move Manipulator

The **Move** manipulator lets you reposition the selected object(s) along with its pivot point in the 3D space. Before you can use the Move manipulator, you must first activate it. Although clicking the Move button activates the move manipulator, it is best to press the default hotkey of **W** to activate it. When activated, the move manipulator icon appears on the selected object, as shown in Figure 4.19. The manipulator's handles point in the directions of the three axes—X, Y, and Z.

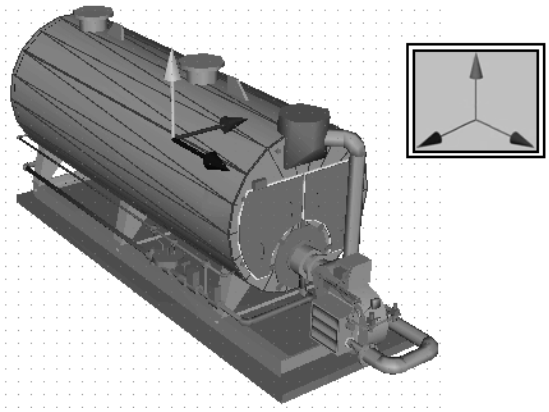

Figure 4.19 - The Move Manipulator on a Tank

There is more that one way to use the manipulator once it is activated. Let's look at each way.

To use the mouse: Click on a handle on the manipulator. This chooses the direction in which you want to move the selected object(s). Notice that the handle colors

correspond to the axis colors on the axis legend. Once a handle is activated, it turns yellow. Click and drag the yellow handle to move the object in the activated direction.

> ***Tip!*** If no direction handle is selected, you can move the object freely along all three axes by holding down the middle mouse button or wheel.

To use the keyboard: Alternatively, you can use keyboard keys to move the selected object(s). The text box in the Manipulator section of the Home ribbon (shown in Figure 4.20) determines how far the object moves when you press a key:

- Pressing the Left and Right arrow keys moves the object along the X axis.
- Pressing the Up and Down arrow keys moves the object along the Y axis.
- Pressing the Page Up and Page Down keys moves the object along the Z axis.

To use the + and - buttons: You can also click the plus and minus buttons in the Manipulator section of the Home ribbon, shown in Figure 4.20. The text box determines how far the object moves when you click a button. The *red* buttons move the object(s) along the X-axis, the *green* buttons along the Y-axis, and the *blue* buttons along the Z axis. (The colors correspond to the colors on the axis legend.)

Figure 4.20 - Button Colors on the Home Tab (Left) Correspond to Those in the Axis Legend (Right)

> ***Tip!*** For more precision in where the object moves, you can click on the Transform button and enter the precise movements you want to make. We will look at the Transform button later in this chapter.

4.6.2.2. The Rotate Manipulator

The **Rotate** manipulator lets you rotate the selected object(s) in the 3D space around the pivot point, as shown in Figure 4.21.

Figure 4.21 - Rotation around the Pivot Point

Before you can use the Rotate manipulator, you must first activate it. Although clicking the Rotate button activates the manipulator, you can also press the default hotkey of **E** to activate it. When activated, the Rotate manipulator is represented by a gyroscope that grips the selected object(s), as shown in Figure 4.22. Each bar of the gyroscope represents one of the three axes (X, Y, and Z); each bar's color identifies the axis along which it will rotate.

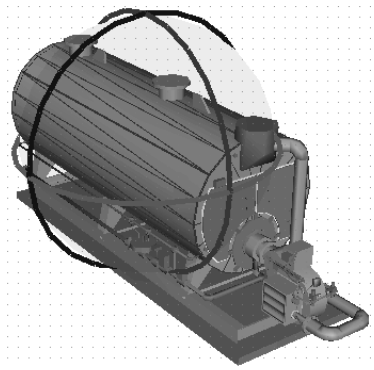

Figure 4.22 - The Rotate Gyroscope on a Tank

With your **mouse**, you can click and drag a bar to rotate the object along one of the three main axes. If you click inside the gyroscope without selecting a bar, you can rotate along all three axes at the same time (this is called *free rotation*). You can also use the **keyboard keys** and the **+ and - buttons** to rotate the selected object(s) in the same way you use them with the Move manipulator.

> ***Tip!*** For more precision in the angle to which the object rotates, you can click the Transform button and enter the precise relation angle you want to make. We will look at the Transform button later in this chapter.

4.6.2.3. The Scale Manipulator

Use the **Scale** manipulator to resize the selected object(s) to scale. This feature scales the object out from or in toward the pivot point, as shown in Figure 4.23.

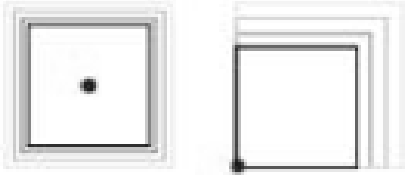

Figure 4.23 - Scale in Relation to the Pivot Point

Before you can use the Scale manipulator, you must first activate it. Although clicking the Scale button activates the manipulator, it is best to press the default hotkey of **R** to

activate it. Once activated, the Scale manipulator appears over the selected object(s), as shown in Figure 4.24. A small cube appears on the intersection of the axes as well as at the end of each axis – there are four cubes in total.

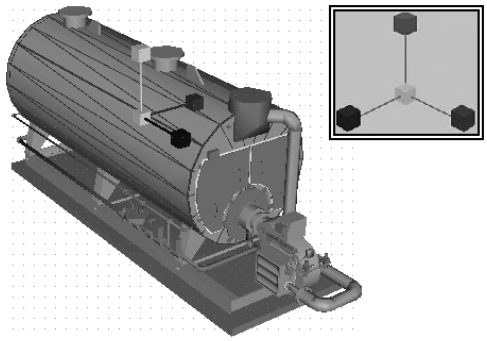

Figure 4.24 - The Scale Manipulator on a Tank

With your **mouse**, you can click any cube that lays on a scale manipulator axis and then, dragging the mouse, you can scale the object(s) along that axis (this is called *constrained scale*). Clicking on the cube at the intersection of the axes lets you scale along all axes at the same time (this is called *free scale*). You can also use the **keyboard keys** and the **+ and - buttons** to scale the selected object(s) in the same way you use them with the Move manipulator.

> ***Tip!*** To use a more precise measurement to scale the object, use the Transform button and enter the precise change you want to make.

4.7. Advanced Transformations and Mirroring

An alternate way to move selected 3D objects is by using the advanced transformation buttons in the Transform section in the Home tab.

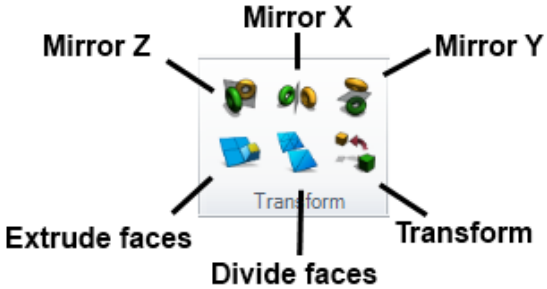

Figure 4.25 - The Transform Section

The Transform section contains six options that let you move the selected object in relation to the X axis, Y axis, or Z axis, and extrude or divide faces. Let's take a look at each of these options.

4.7.1. Transform Button

Clicking the **Transform** button in the Transform section lets you move, rotate, or scale the selected object(s) by numeric precision along any of the three axes. This button provides the *same* changes that are available with the **Move**, **Rotate**, and **Scale** buttons in the Manipulator tab group, except with the Transform button you can be more precise. When you click this button, the dialog box appears as seen in Figure 4.26, where you can explicitly state the changes you want by entering numeric changes.

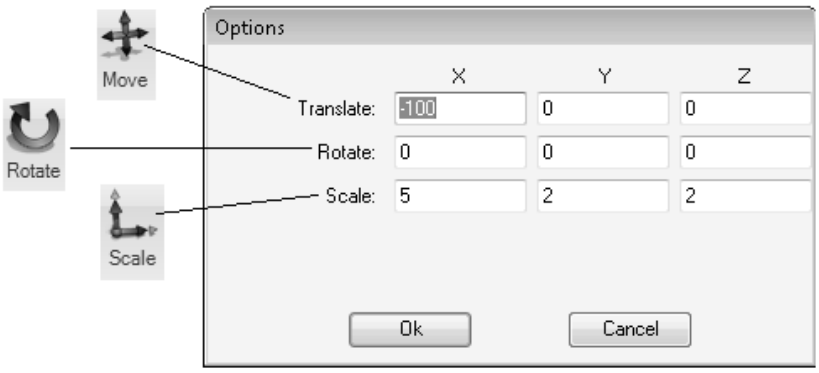

Figure 4.26 - Transform Options

The **Translate** fields indicate the number of units by which you want to move the pivot point in each of the axes. Specify the number of units to move the object along the X-, Y-, and/or Z-axis.

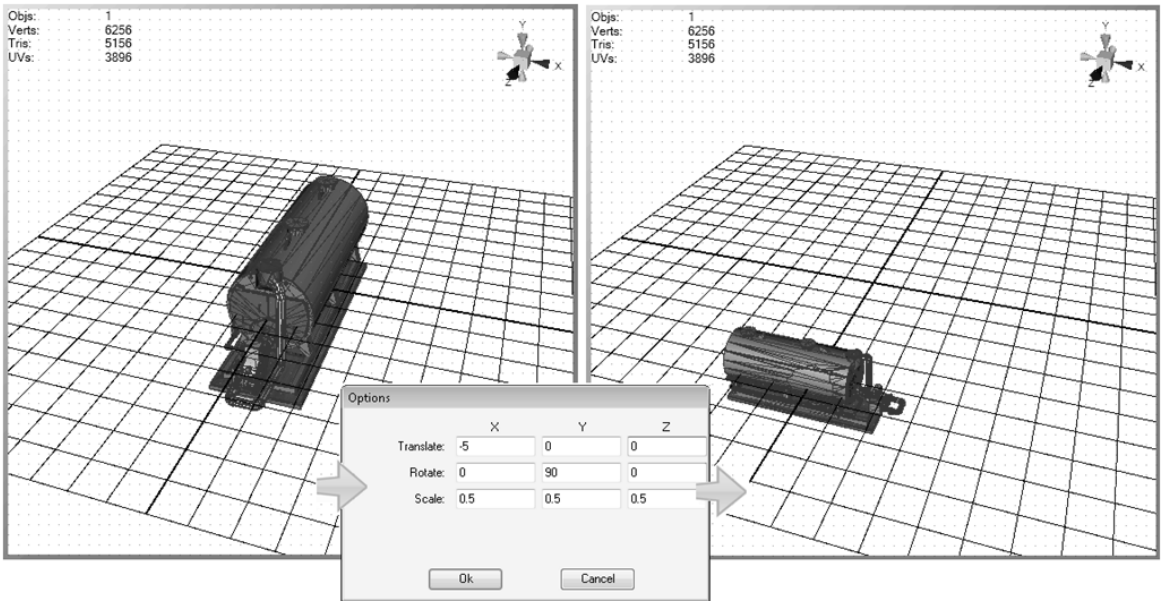

Figure 4.27 - The Transform Options and Effect

To rotate the selected object(s) use the **Rotate** fields. Rotations are expressed in angles from 0 to 360 for each axis; the selection rotates on its pivot point along the X-, Y-, and/or Z-axis.

Scale indicates the factor by which you want to increase or decrease the size of the selected object(s) along each axis. A value of 1 indicates that you want the object(s) to remain the same size, a value of 2 doubles the object(s) in scale, and a value of .5 reduces the object(s) by half.

4.7.2. Mirror X Button

Using the **Mirror X** button shifts the selected object to its mirrored position along the X axis. The object is repositioned as shown on the button itself. An example using the tank object is shown in Figure 4.28.

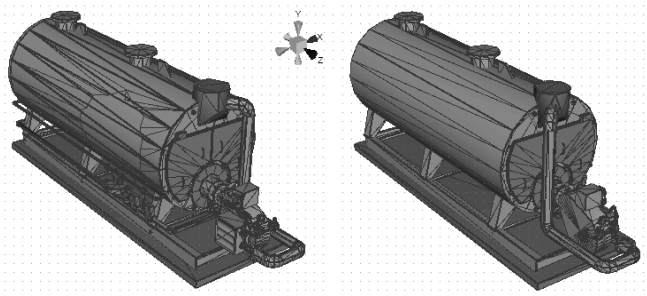

Figure 4.28 - Mirroring along the X-axis

4.7.3. Mirror Y Button

Using the **Mirror Y** button flips the selected object to its mirrored position along the Y axis. The object is repositioned as shown on the button itself. An example using the tank object is shown in Figure 4.29. In this example, the tank turns upside down.

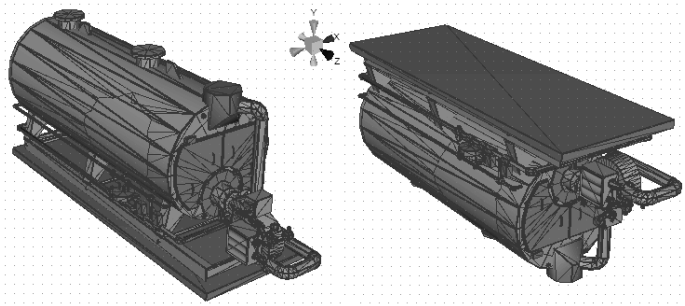

Figure 4.29 - Mirroring along the Y-axis

4.7.4. Mirror Z Button

Using the **Mirror Z** button shifts the selected object to its mirrored position along the Z-axis. The object is repositioned as shown on the button itself. An example using the tank object is shown in Figure 4.30. In this example, the tank flips end to end.

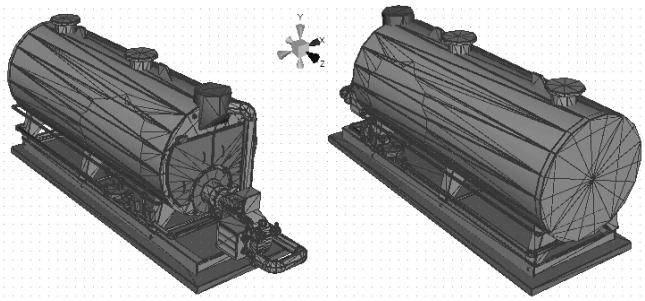

Figure 4.30 - Mirroring along the Z-axis

4.7.5. Extrude Face Button

You can *extrude* a face of an object (as shown in Figure 4.31) by using the **Extrude Face** button . To use this button, you must click the Face selection button (in the Selection Mode section on the Home ribbon), then click on the plane where you want the extrusion to be created. Then you can click the Extrude Face button.

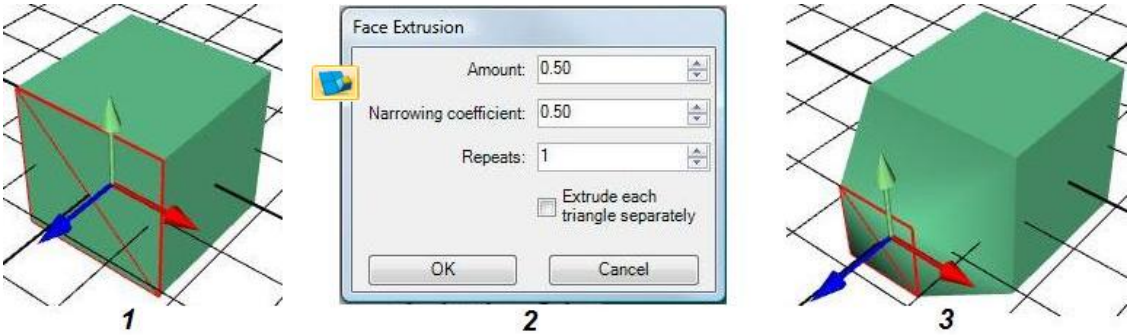

Figure 4.31 - An Example of the Face Extrusion Process

A dialog will appear where you can configure the extrusion. Specify the Amount by which you want the face to extrude from the object. For example, .50 makes the extrusion protrude from the object by 50% of the object's depth. In the Narrowing Coefficient field, specify the amount by which you want it to narrow. For example, .50 reduces the extruding region by 50%.

In the Repeats field, specify the number of times you want the extrusion to repeat. For example, if you enter 2, the Amount and Narrowing Coefficient will be applied once,

and then it will be applied a second time to the face. If you want each triangle in the face to extrude independently, put a checkmark in the Extrude Each Triangle Separately checkbox.

You may want to experiment with this option a little bit to give yourself an idea of how you might use it. For example, you might use it on a cylinder to create a smokestack. One thing you will notice is that once you add the extrusion, the object changes to a 3D Model with a 3D Geometry component. You can see this best in the object Explorer, shown in Figure 4.32. What might have started out as a Cube would become a 3D Model with a 3D Geometry component.

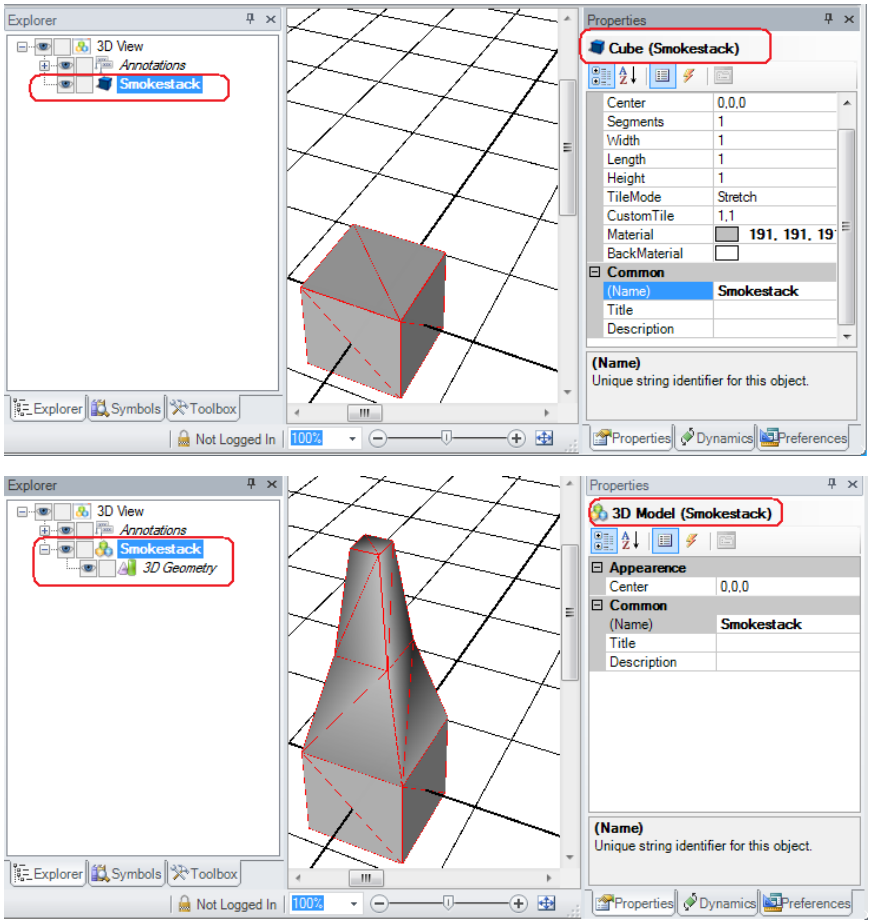

Figure 4.32 - Example of a 3D Model Created by Adding an Extrusion

Any properties that you applied to the cube before applying the extrusion are retained, but as a 3D Model it will now have a different set of properties; as such, you may not be able to undo the properties you set before doing the extrusion. For this reason, you should be aware of the number of segments you are using before applying an extrusion so that you don't lock yourself into having unnecessary detail.

4.7.6. Divide Face Button

You can click the **Divide Face** button to divide the selected face of an object into more segments or vertices. The button becomes available only when the Face selection button is selected in Selection Mode section on the Home ribbon.

This is a good feature to use when you want to increase the number of segments for a 3D Model. Back when we reviewed primitives, we talked about how you can increase the number of segments on the face of a primitive by specifying the number in the primitive's properties. But some types of objects, such as 3D models, don't have a property for the number of segments; instead, to increase the number of segments, you can use this button.

Figure 4.33 shows how a 12-triangle face becomes a 36-triangle face after clicking this button. Because the face is made of triangles, the number of triangles on the face triples. This gives you more segments, but it also adds to the size and processing time of the display, so you'll want to use this feature with discretion.

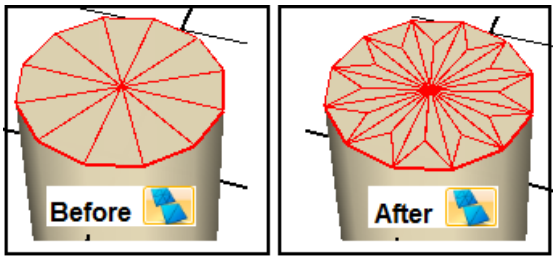

Figure 4.33 - An Example of the Face Division Process

4.8. Duplicating Objects

Use the **Duplicate** section on the Home ribbon to copy a selected object multiple times. This feature can create a pattern with the copies in space and is especially useful when you need to create an object that has repeated features (such as rows of buttons) or sections (such as a spiral staircase) that form a pattern.

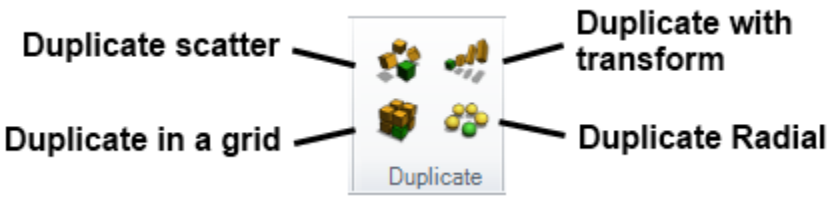

Figure 4.34 - The Duplicate Section on the Home Ribbon

When making multiple duplicates, as you do with these buttons, you can move, rotate, and scale duplicate objects. Duplicates that are created are positioned in relation to the position of the original object's location and use the original object's pivot point.

4.8.1. Duplicate Scatter

The **Duplicate Scatter** button allows you to create copies of the selected object(s) that are randomly scattered throughout the 3D viewport. When you click this button, the Duplicate Options dialog box displays, as shown in Figure 4.35; here you can set numerical options for the object(s) duplication.

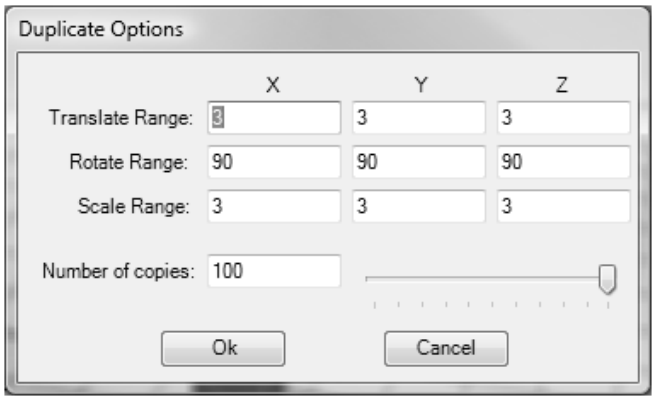

Figure 4.35 - Duplicate Scatter Options

You can specify a range within which each copy can move (Translate), rotate, and scale on the X, Y, and Z axes.

The range of values is used to *randomize* the position, orientation, and scale of each duplicated object relative to the original. Because the values identify the limits of the randomness, *larger values* allow for larger ranges, and therefore will result in increased randomness in the copies. By contrast, *smaller values* will result in less variation from the original, and gives more restrictions in how the duplicates differ. The more copies you want to make, the longer it will take to complete the operation. Figure 4.36 shows how you might use the Duplicate Scatter option to create a cityscape.

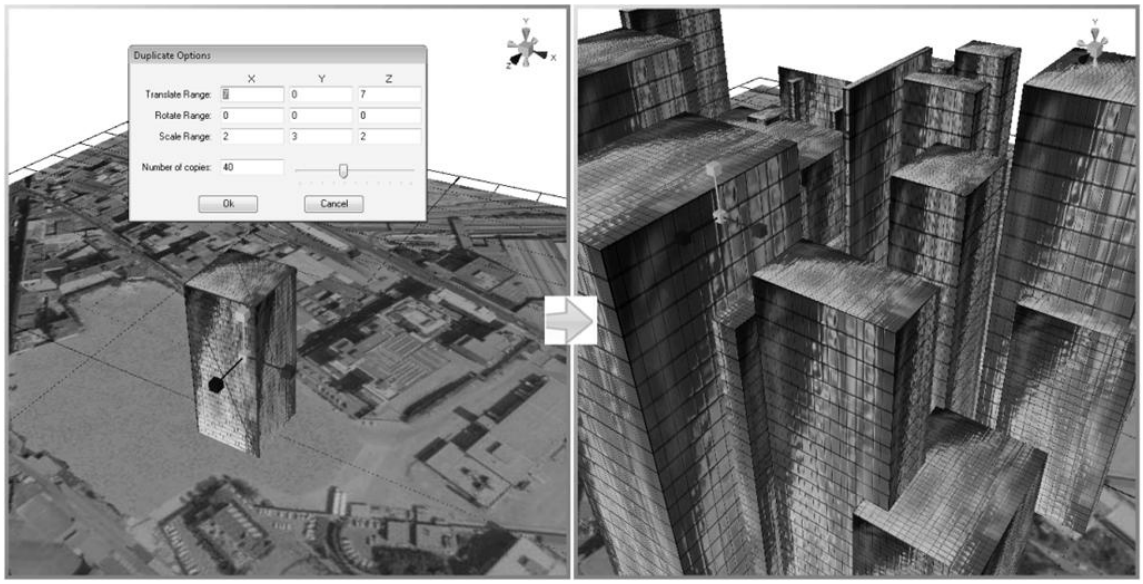

Figure 4.36 - Using Duplicate Scatter to Create a Cityscape

4.8.2. Duplicate Array

Aside from duplicating objects with randomized values, you can also duplicate them in an orderly fashion, as shown in Figure 4.38.

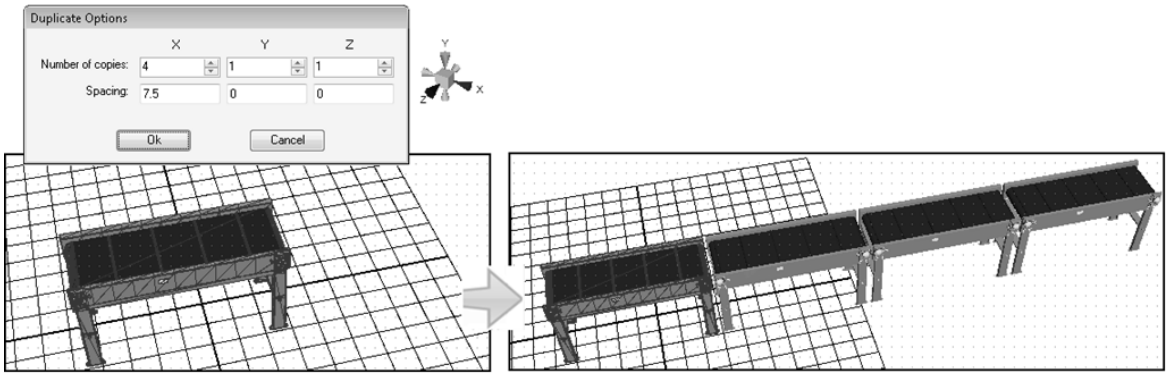

Figure 4.37 - Using Duplicate Array to Create a Conveyor

Use the **Duplicate Array** button to create a regular 3-dimensional array of duplicates of the selected object(s). Once you click on the button, the option dialog in Figure 4.38 displays so you can set numerical options for the object(s) duplication.

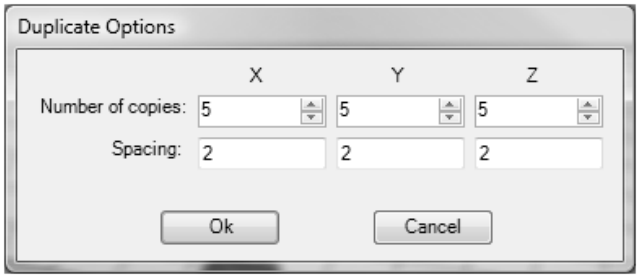

Figure 4.38 - Duplicate Array Options

You can specify the number of copies to be created on each axis, as well as the spacing of the objects, as shown in Figure 4.38. Spacing refers to the number of spaces in the grid and determines the location of copies along the axis. Simply count the spaces in the grid to determine how many spaces you want each copy to be offset from each other.

Figure 4.37 (shown earlier) demonstrates how you can use Duplicate Array to create a conveyor by copying along the X-axis. Figure 4.39 shows its use for creating warehouse shelving by making copies along the X-axis and the Y-axis.

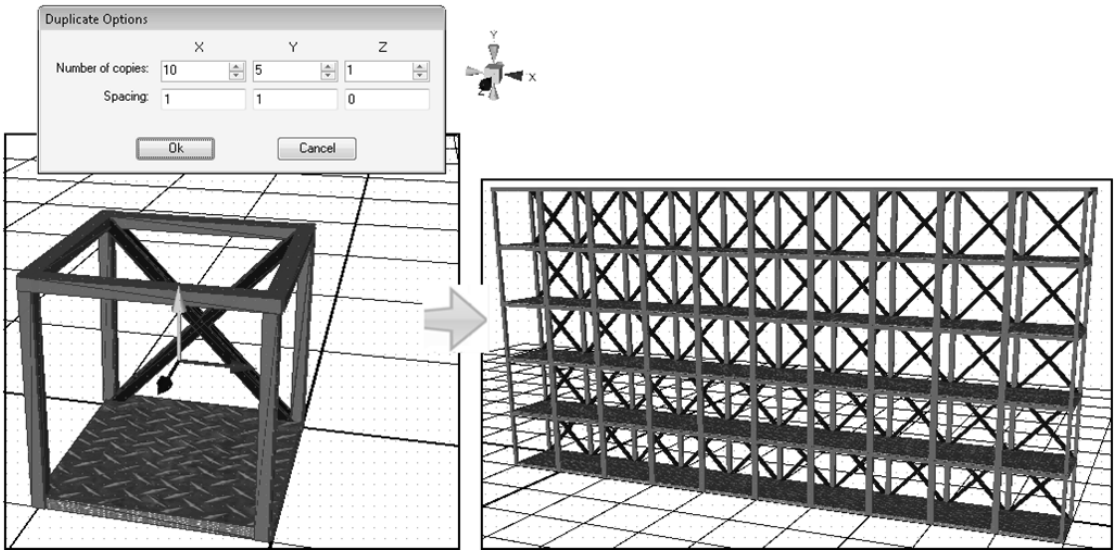

Figure 4.39 - Using Duplicate Array to Create Warehouse Shelving

4.8.3. Duplicate Radial

You can use the **Duplicate Radial** button to create a radial array of the selected object(s). When you click this button, a dialog displays so you can set numerical options for the object(s) duplication.

You can specify the degree or ratio along each axis (0 to 360) as well as the center of rotation in the form of an offset from the center of the key object. By offsetting copies

along the right axis, you can create a spiral effect. Figure 4.40 shows how you can use Duplicate Radial to create spokes on a wheel.

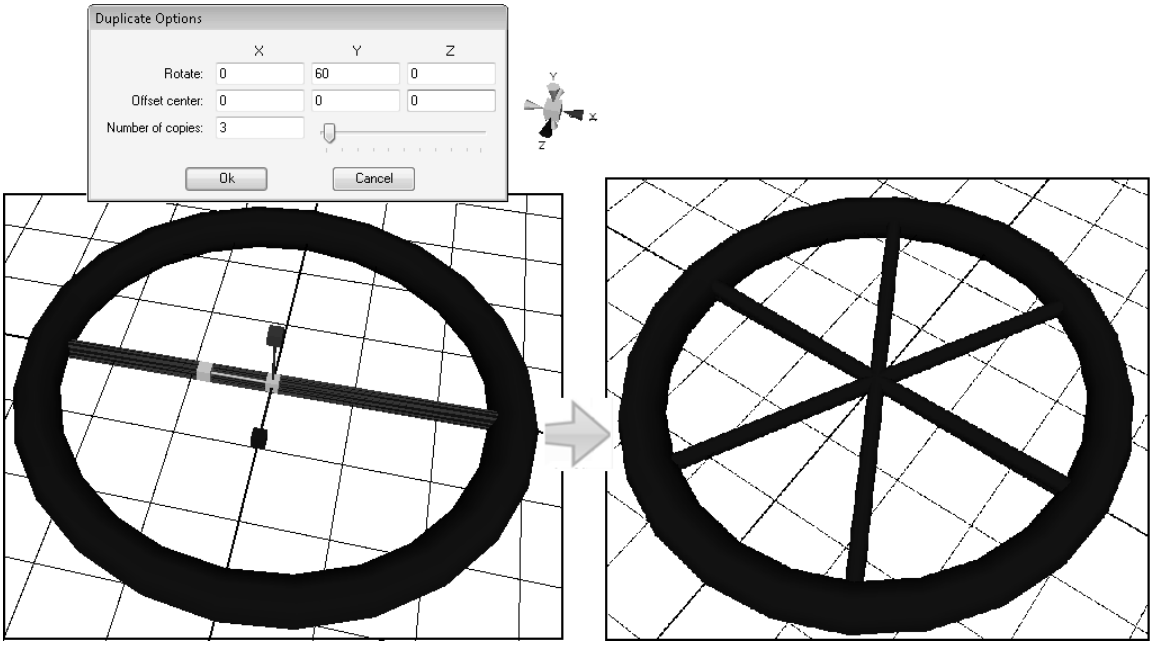

Figure 4.40 - Using Duplicate Radial to Create Wheel Spokes

4.8.4. Duplicate with Transform

Click the **Duplicate with Transform** button to create a number of copies of the selected object(s) and apply a numerical transformation to each of them. Once you click this button, a dialog appears so you can set numerical options for the object(s) duplication, as shown in Figure 4.41.

Each copy will be moved, rotated, and scaled based on the values you enter. The movements, rotations, and scaling will occur along the axis or axes you specify. For example, if you want each copy to move up one and rotate 30 degrees around the Y-axis, enter 1 in the Translate Y box, and enter 30 in the Rotate Y box. Figure 4.41 provides an example of how you might use the Duplicate with Transform function to create a helix.

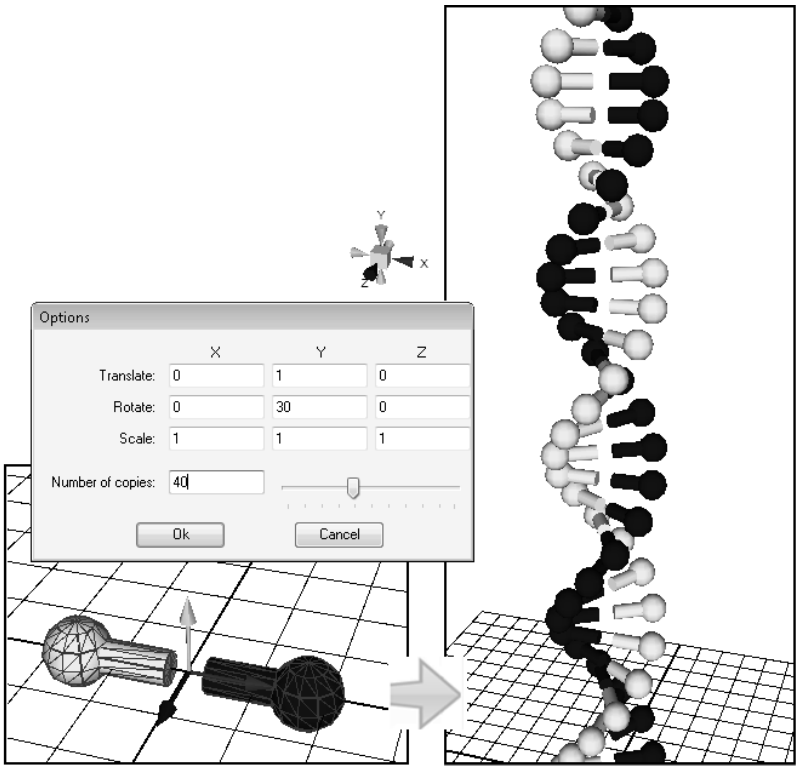

Figure 4.41 - Using Duplicate with Transform to Create an Upward Spiral on the Y-axis

4.9. Applying Materials to Object Surfaces

In creating objects, you'll want to apply color and texture, called ***materials***. Materials are colors or images that you can use on an object to make it look like it is created with certain materials such as wood or metal.

Each primitive or object has properties called **Material** and **BackMaterial**. Clicking on the ellipsis [...] button of each of these properties brings up a dialog that allows you to apply color or an image to the object. There are other properties in this dialog as well, which we will discuss later.

Using the Properties Explorer to set the material allows you to set it for only the selected primitive. If you want to apply the same material to a *group* of primitives or objects, you can use the **Materials** tab of the ribbon (shown in Figure 4.42) to apply a ***facet*** to an object's front or back. Let's look at each section of the Materials tab.

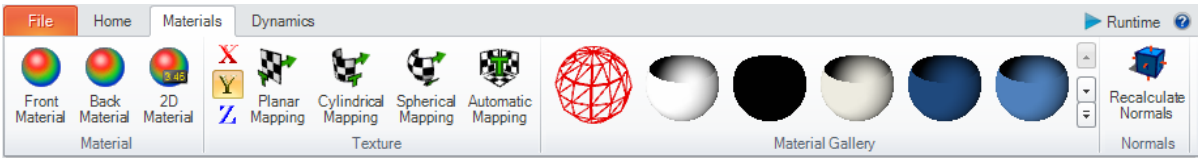

Figure 4.42 - The Materials Ribbon

4.9.1. Front and Back Material

You can use the the **Front Material** and **Back Material** buttons to change the color and property of an object's surface. The surface of any object has a front and a back, but where you find this surface is not always obvious. For that reason, let us look at the two primitives in Figure 4.43: a three-dimensional donut shape and a two-dimensional polygon.

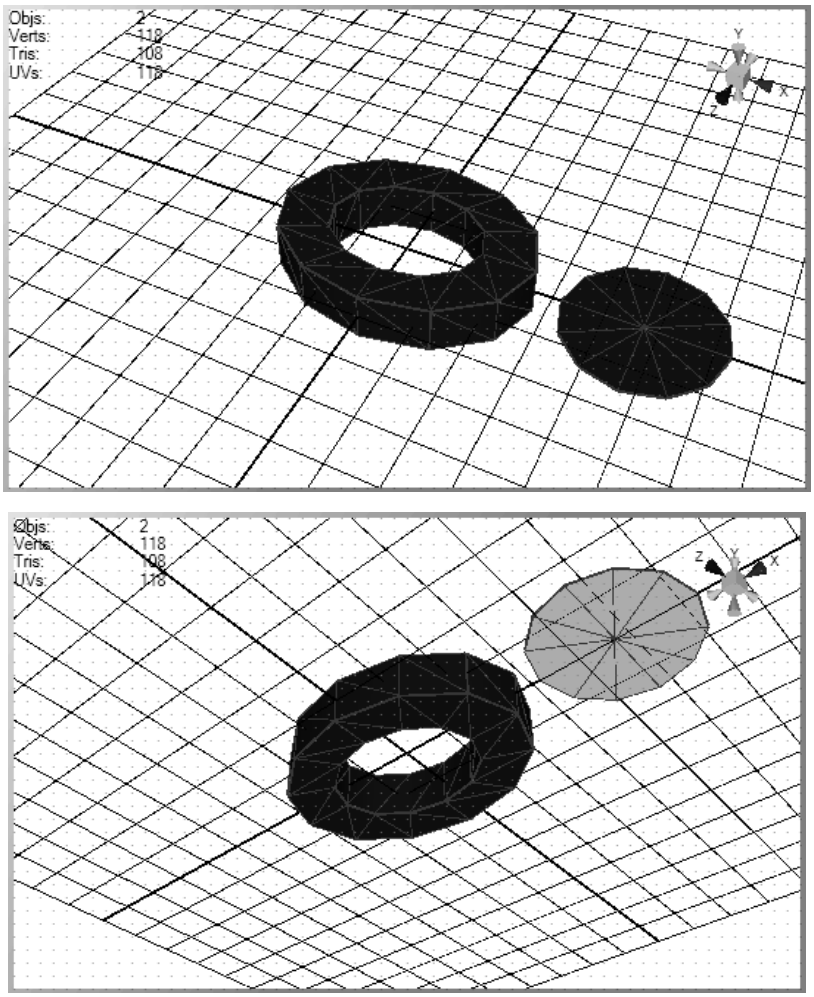

Figure 4.43 - Two Primitives shown from Above (Top) and Below (Bottom)

Each object has a **Material** property that determines its *front* color, and a **BackMaterial** property that determines its *back* color. The figures in the example have identical Material property values, which give them a blue front color, and identical BackMaterial property values, which give them a green back color.

In Figure 4.43, you can see the polygon's back color when you look at it from below. The polygon, which is two-dimensional, has only two sides. But for a three-dimensional shape, *its back side is its inside*, and the only way we can see the inside of

the donut shape is to remove a cap. In Figure 4.44, the donut's bottom cap has been removed, showing its green inside from beneath.

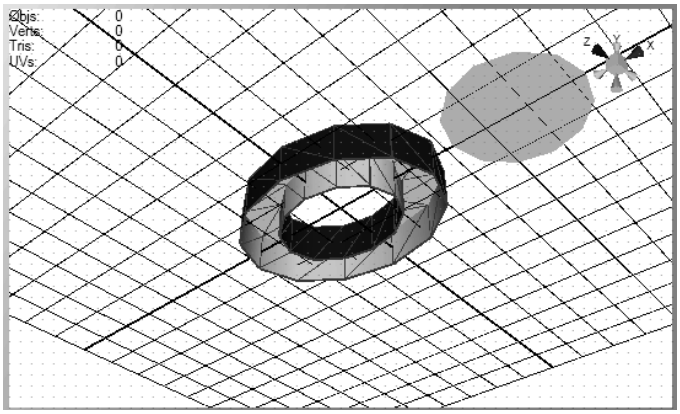

Figure 4.44 - The Donut's Bottom Cap is Removed to Reveal Its Inside

Clicking the **Front Material** and **Back Material** buttons on the Materials tab opens a dialog, shown in Figure 4.45, so you can set multiple material values for an object:

- **Diffuse color** is the color that the object *reflects*
- **Ambient color** is the light that *surrounds* the object
- **Emissive color** makes the object appear to *emit light* of that color
- **Specular color** and **intensity** are used for light *highlights* on the object's surface

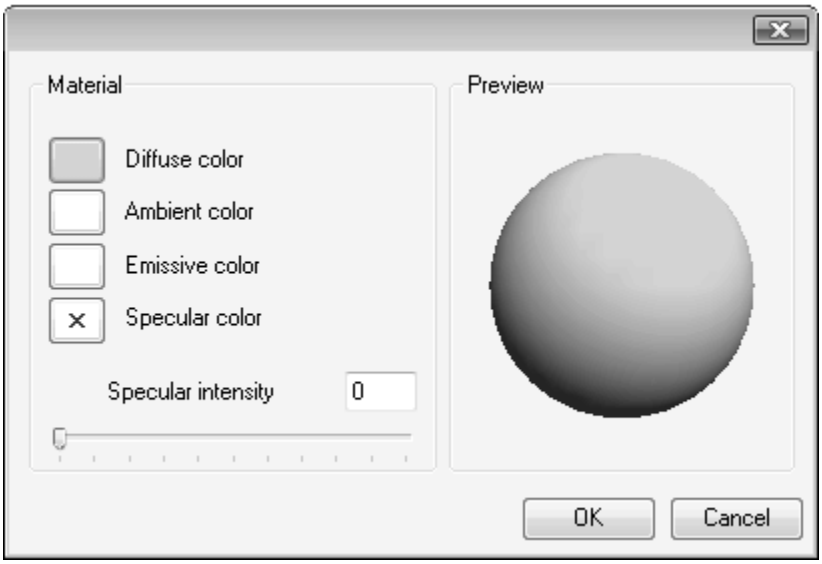

Figure 4.45 - Setting Values for the Front and Back of an Object or Shape

If you click on any of the buttons in the Material dialog, a separate dialog displays your options. Each of the four buttons opens a dialog that displays the **Solid Color** tab as shown in Figure 4.46, which lets you choose the color and opacity. The *diffuse* and

specular dialogs also let you set a gradient using the **Gradient** tab and use an image (such as a .jpg file that shows brick work or corregated metal) using the **Image** tab.

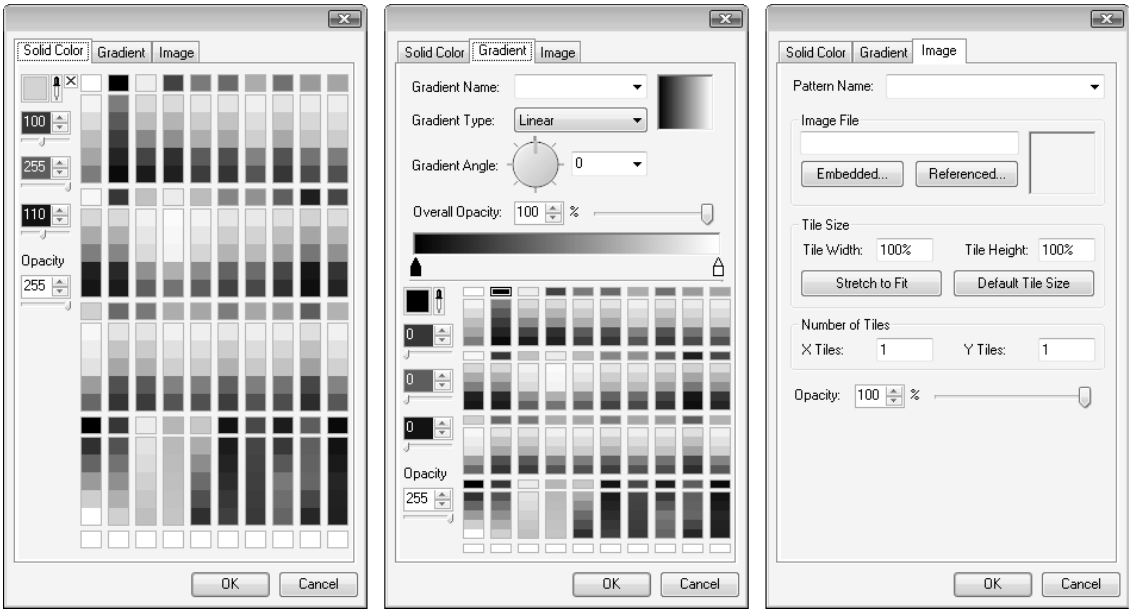

Figure 4.46 - Setting Color Values for Front and Back Materials

4.9.2. Texture

The ***texture*** of an object affects how material wraps around it. In general, GraphWorX64 is smart enough to know how to wrap a material around an object. For example, if you are applying a material to a sphere, it will know to use the Spherical Mapping. However, there may be instances where you want to wrap the material around the object in another way.

In such cases, you can use the buttons in the **Texture** section to map the surface of the object to define how an image draws on it. The surfaces available are plane, cylinder, sphere, or automatic, and they can be applied along the X, Y, or Z axis. Choosing the automatic option allows GraphWorX64 to apply the texture as it sees fit.

> **NOTE:** Applying a texture to a primitive changes it into a 3D Geometry in a 3D Model. Once you do this, you will lose the ability to edit the properties of a primitive because the primitive no longer exists.

4.9.3. The Material Gallery

The **Material Gallery** is a repository of 3D materials and textures you can apply to the surface of a design. You can add shortcuts to materials that you use often. For example, you may create a wall with the outside material being brick and the inside material

being wood. You can save the materials applied to this –wall" into the Material Gallery. This way, when you create another wall later, you can simply click on the material in the gallery to apply both the Material and BackMaterial to the new wall.

If you no longer need materials that you added to the gallery or if you want to start over with a clean gallery, you can delete all materials in the gallery, or reset the gallery so that it only contains the default materials.

4.9.4. 2D Material

Other than the common materials for 3D objects, GraphWorX64 allows you to have dynamic materials to be applied to your object. These are the so-called *2D Materials*. The concept behind this is similar to those of annotations, where the difference is that a two-dimensional process point or other 2D display or object will be wrapped over the 3D object. This is handy if you want to show a data source value over one of your objects (as shown in Figure 4.47).

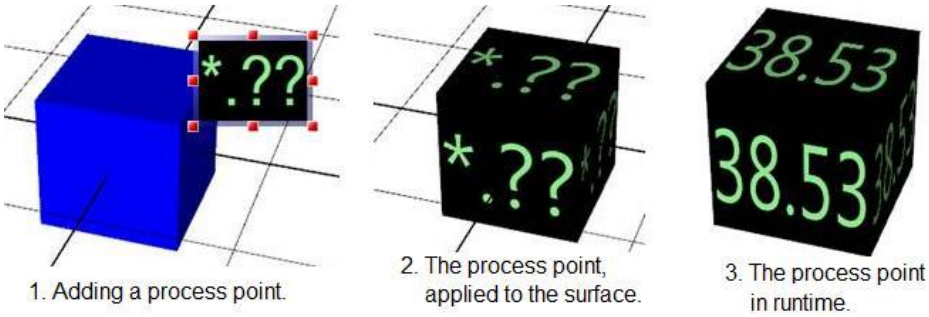

Figure 4.47 - Using a 2D Material on a Cube Object

Here is how you use a process point or 2D material the surface of a 3D object:

1. In the 3D viewport, select the 3D object you are applying the process point to.
2. From the Materials ribbon, click the 2D Material button. An annotation box opens next to the selected object.

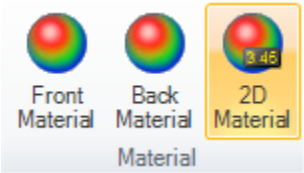

Figure 4.48 - The 2D Material Button in the Ribbon Bar

3. Insert the 2D object into the annotation and configure it as needed. You can configure it just as you would any 2D display.

4. Double-click outside the annotation editor (or press the Esc key on your keyboard). The annotation box closes and the 2D material is applied to the object's surface.

5. If the 2D object has a data source, go into Runtime to see it work.

4.9.5. Normals

In a GraphWorX64 3D viewport, you can click the Recalculate Normals button to recalculate the way in which light bounces off the triangles of the selected object. You might use this button after working extensively with a model. Select the object, then click this button.

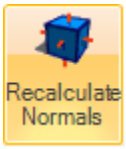

Figure 4.49 - Recalculating Normals

Lab: Manipulating Objects L4

In this lab, we will finish creating the tank that we started in lab 3 as well as gain more experience in manipulating 3D objects.

Estimated time to complete this lab: 20 minutes

4.1. Task Checklist

In this lab, you will complete the following tasks:

- Create a tank for icTonics
- Apply a material to the tank

4.2. Support Files

You will need the following files to complete this lab:

- Practice1.gdfx

4.3. icTonics Tank

In this lab, we will put together the icTonics storage tank.

1. Open Practice1.gdfx if you closed it from the previous lab.
2. Select the Cylinder object and click on the Rotate Manipulator.
3. Rotate the cylinder around the Y-axis (the green one) so that the opening is facing you.
4. Change your view to the Front view by clicking on the Front button in the View section of the Home tab. Zoom in or out, if you need to, so you ocan see the three objects easily.
5. Select the Sphere object and click on the Scale Manipulator.

6. Reduce the size of the sphere on its Y-axis so that the dip of the bowl is shallower.

Figure L4.1 - Resized Y-axis on Sphere

7. Select the Cone, and click on the Move Manipulator.
8. Move the cone to a location over the cylinder.
9. Do the same with the Sphere, except you want to move it below the cylinder.
10. Select all three primitives. Make sure that the key object is the Cylinder.
11. In the Arrange section of the Home tab, click on the down arrow for the X axis and select Align Center.
12. Do the same for the Z axis, except select the Align Center Depth option.
13. Now do the same for the Y axis, except choose Stack Vertically. Your tank should look similar to Figure L4.2.

Figure L4.2 - Tank

14. With all three objects still selected, click on the Materials tab and click the Front Material button.
15. Click on the Diffuse color option. In the dialog that comes up, go to the Image tab.
16. Click on the drop-down list in the Pattern Name field select Metal Gray.
17. Click on OK until you are back in the 3D viewport. Your tank should now look similar to Figure L4.3.

Figure L4.3 - Tank with Material Applied

18. With all three objects still selected, click on the Group button in the Arrange section on the Home tab of the ribbon.

19. You will see that the three objects are now under a 3D Model in the Explorer. Change the name of the 3D Model to "Empty Tank 3D".

20. Copy and paste this tank into the GENESIS64TrainingSymbols'symbol library. Name the tank in the symbols library "Empty Tank".

4.4. Advanced Lab

If you have some time, here is a bit more practice in manipulating objects. Note that this is an advanced lab, and you will be challenged by having less-detailed instructions. In this section of the lab, we will create a Rubik's cube.

1. Start a new GraphWorX64 display and create a new 3D space in it.

2. Select the Cube primitive to create a cube.

3. Click on the "Duplicate the selected objects in a grid" button.

4. We want to create 3 copies in each direction and we can leave the spacing between each cube to 1.2 since the size of the cube is 1.

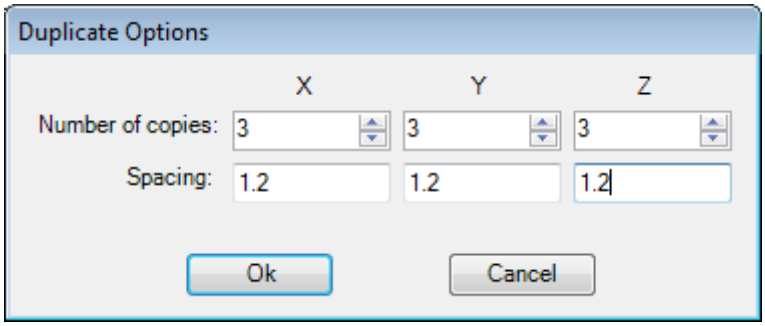

Figure L4.4 - Duplicate Options Settings

Tip! You may want to rename the cubes in the Explorer.

5. Now insert a plane, and change the color to red. Align this plane to the upper left cube farthest away from you by using the align options in the Arrange section.

6. Duplicate the plane. Select the duplicate in grid option and give it the settings as shown in Figure L4.5.

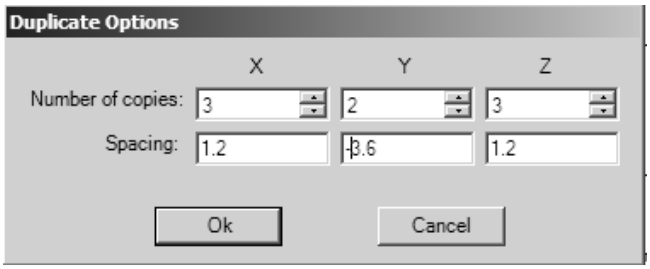

Figure L4.5 - Duplication of Planes

Tip! You may want to rename the planes and change the orientation of the planes a bit to match up a bit more to the cubes.

7. Select the planes that are on the bottom of the cube, and change the color to Orange.

8. Repeat steps 5-7, except the planes should be on the X and Z axis and each side of the Rubik's cube should have a different color.

9. Your finished Rubik's cube should look similar to Figure L4.6.

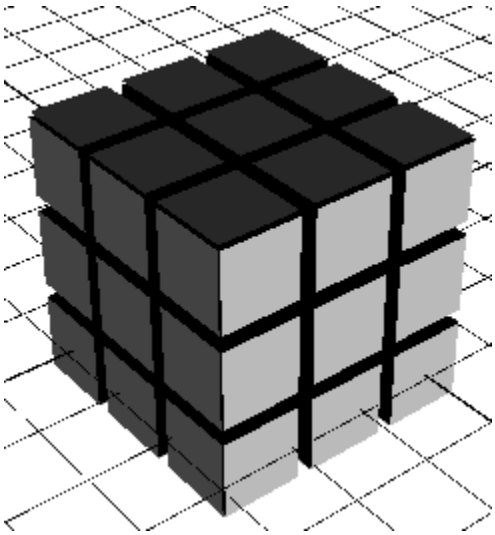

Figure L4.6 – Rubik's Cube

Review: Manipulating Objects R4

4.1. Chapter Objectives

You should now know and understand the following topics:

- Opening a 3D object
- Ways to select 3D objects
- Ways to arrange 3D objects
- Grouping and ungrouping objects
- 3D coordinates
- The pivot point
- Ways to manipulate objects by moving them, rotating them, scaling them, shaping them, transforming them, and duplicating them
- Applying colors and other materials to object surfaces

4.2. Summary

To open a 3D object, expand the Explorer to display the object in an existing view or from the Symbols Library, and then drag the 3D object onto the work area.

Before you can select an object, selection modes must be activated in the Selection Mode section on the 3D ribbon's Home tab. By expanding objects in the Explorer, you can single out 3D objects for selection, including sub-elements and groups, and select them in the Explorer tree. You can also drill down to sub-elements if you are using the hierachy selection icon.

You can also select multiple objects at a time. When you select multiple objects or sub-elements, the last selected object is the *key object*. Some 3D tools use the key object to determine what to do with the selected objects. For example, when you align multiple objects, they align to the *key object*.

You can align objects along the X, Y, or Z axis. You can group and ungroup objects. You can center multiple objects and resize them using the key object's size as the basis.

An object's 3D coordinates identify the object's location in the scene by using one coordinate point in the 3D space. To set an object's 3D coordinates, you must be aware

of the orientation of the X, Y, and Z axes. *World space* is the coordinate system for the entire 3D scene; its origin is at the center of the scene. *Object space* is the coordinate system from an object's point of view; the origin is the object's pivot point, and its axes rotate with the object. *Local space* is similar to object space, but it uses the origin and axes of the object's parent node in the hierarchy of objects.

Pivot points control how objects rotate and scale. All transformations of an object are relative to the object's pivot point. You can move the pivot point of an object by changing the object's Center property.

The Move manipulator moves the selected object(s) in the 3D space; the pivot point moves along with the object. The Rotate manipulator rotates the selected object(s) in the 3D space around the object's pivot point. The Scale manipulator lets you resize the selected object(s) to scale out from or in toward the object's pivot point. The Transform section on the Home tab of the ribbon contains four options that let you move a selected object in relation to the X axis, Y axis, or Z axis.

Use the Duplicate section on the Home tab to copy a selected object multiple times. This feature can create a pattern with the copies in space and is useful if you have an object that has repeated features (such as rows of buttons) or sections (such as a spiral staircase) that form a pattern.

If you are creating 3D models, some additional tools may be helpful. Three selection modes lets you select vertices, edges, faces, and triangles. The Divide Face button lets you multiple the number of triangles on a face so you can refine the surface. And you can reshape a primitive by using the Extrude Face feature.

You can also apply materials, colors, and effects to the front and back (or outside and inside) of objects. You can create your own Material Gallery of often-used materials. You can apply images to surfaces and use the Texture buttons to apply images. You can even apply 2D dynamics and objects to the surface if you like.

4.3. Terms

3D Coordinate	Ambient Color	Diffuse Color
Emissive Color	Extrude	Key Object
Local Space	Material	Object Explorer
Object Space	Pivot Point	Rotate
Scale	Specular Color	Texture

4.4. Test Your Knowledge

In this chapter, we discussed ways to manipulate and change objects in the 3D space. Before you continue to the next chapter, make sure you can answer the following questions.

1. How can you select a sub-element of a 3D object?

   a. Click on it in the Explorer
   b. Click a selection mode for sub-elements then click on the object in the 3D space
   c. Ctrl + click on it in the 3D space
   d. All of the above
   e. A and B
   f. B and C

2. What key must you hold down in order to select multiple objects in the 3D space?

   a. The Alt key
   b. The Ctrl key
   c. The Shift key
   d. Both the Alt + Ctrl keys

3. What is <u>NOT</u> true about the key object?

   a. It is the last object you select when selecting multiple objects
   b. It is the invisible object that resides at the center of viewport
   c. It is the object whose pivot point will be used if you do a transformation
   d. It is the object to which all other selected objects will be aligned if you do an alignment

4. True or False: You can align selected objects to any axis.

5. True or False: You cannot have negative X,Y,Z coordinates.

6. How can you change the location of the selected object(s) pivot point?

   a. Change the object's Center property
   b. Change the object's PivotPoint property
   c. Select the object and use the move manipulator to move the pivot point
   d. You cannot change the location of the pivot point

7. There are keyboard hotkeys you can use for manipulating selected objects along the X, Y, and Z axes. What are they?

   a. The number keys on the keypad
   b. The arrow keys and the Page Up and Page Down keys
   c. The function keys F5 through F10.
   d. Ctrl + the arrow keys

8. What do the Mirror X, Mirror Y, and Mirror Z buttons on the Home tab do?

   a. Make mirrored copies of the selected object(s)
   b. Move the selected object(s) to a mirrored position
   c. Make a ghost copy at a mirrored position
   d. There are no such buttons on the Home tab

9. What can the buttons in the Duplicate section do in addition to duplicating the selected object(s)?

   a. Scatter the duplicates
   b. Transform the duplicates
   c. Arrange the duplicates into a grid
   d. Arrange the duplicates in a spiral or radiated effect
   e. All of the above
   f. None of the above

10. True or False: 2D Material option lets you use a process point as material on an object.

3D Animation 5

Animation is what gives a 3D scene its power. Animation can provide operators with data-driven information that they can potentially interact with, as well as time-driven animation that can provide instruction or visual cues. Without the data behind the graphics that provide animation, graphics in a 3D scene are little more than pretty pictures on a screen.

Data-driven animation is animation that is driven by data coming from a device. Real-time data can animate a 3D object to give you a real-time human-machine interface. An example of data-driven animation would be a needle gauge that fluctuates with temperature or speed, or a tank animation in which the water level rises and falls with the level that is in the tank.

Alternatively, you can use **time-based animations** for constructing animations that work like movies. With a time-based animation, you decide the timing and sequence for the object's movement in the display. Time-based animations are useful for assembly or repair instructions, or for training or modeling sequences. **Combining these two types of animation** in any 3D scene gives you the control you need for the three-dimensional human-machine interface over the fourth dimension of time.

In this chapter, we will focus on the dynamics that you can use to animate your displays.

5.1. Objectives

After completing this chapter, you should know how to:

- Animate objects to mimic real-world objects at runtime
- Integrate the user experience into an object's animation
- Use views to provide a movie-like experience at runtime
- Detect virtual collisions in animated objects

5.2. Dynamics

You can create data-driven and time-based animations by using dynamics, much like you would in 2D GraphWorX64 displays. Dynamics in a 3D scene, however, take on the third dynamic of depth. It is the Dynamics tab of the 3D ribbon that contains options for connecting your 3D objects to real-time data and animating them.

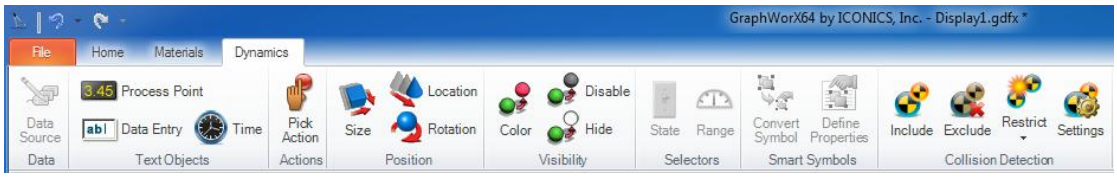

Figure 5.1 - The Dynamics Tab of the Ribbon

The Dynamics tab contains most of the buttons that you see in the 2D environment and, for the most part, they work the same way in the 3D environment as they do in the 2D environment.

On the 3D Dynamics tab, the **Data Source**, **Text Objects** (that is, the Process Point, Data Entry, and Time text objects), and the **Pick Action**, all work the same way they work in 2D. **Smart Symbols** and **Selectors** are the same as in 2D, too. Since you should already have an understanding of these tools, in this 3D course we will focus on the **Position** and **Visibility** dynamics and their 3D uses.

> **NOTE**: If you are not familiar with these tools or would like a review, please refer to your GENESIS64 Core training manual and the GraphWorX64 Expanded manual and the related help files.

5.3. Position Dynamics

You can set up a dynamic that repositions or resizes objects during runtime by using the buttons in the Position section of the Dynamics tab. Let's take a look at how you make these dynamics work in 3D.

Figure 5.2 - The Position Dynamics

5.3.1. The Size Dynamic

The **Size** dynamic can alter the size of an object based on the data source value. In 3D, the **Size** dynamic makes the size of an object change along any or all axes. Use the Size dynamic to define a 3D object that changes size with the real-time value of its associated data point. For example you can make a water level rise and fall, or a balloon expand and contract.

To add a Size dynamic to an object, select the object and then click on the **Size** button.

The properties for 3D are like the properties for 2D although there are some differences to accommodate the 3D environment. Figure 5.3 shows the properties for the Size dynamic with the drop-down image for the **SizeMode** property. You can use this property to specify the directions in which the object's size changes. The colors on the icon match the colors on the viewport's axis legend. By clicking on the arrows, you identify the axes and directions in which the objects are to resize. The directions you choose will display in the SizeMode field as **Up, Down** (x-axis); **Left, Right** (y-axis); and **Back, Front** (z-axis).

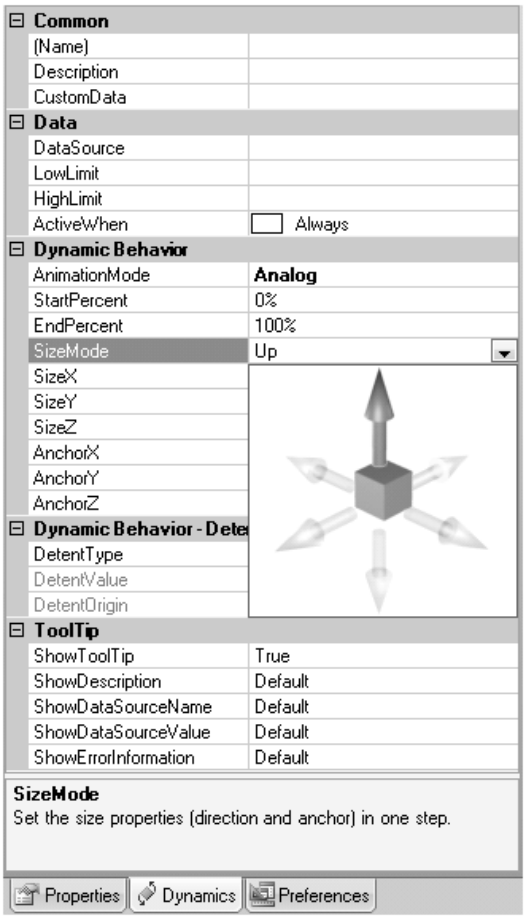

Figure 5.3 - Size Dynamic Properties

5.3.2. The Location Dynamic

A **Location** dynamic moves or translates an object based on the data source. The Location dynamic makes an object move along a path that you create. You can use it to move an object, such as a box along a conveyer belt.

To add a Location dynamic, select the object, then click on the **Location** button on the Position section on the Dynamics tab. This creates a Location dynamic on the object. Initially a path from the selected object to an oblique form appears, as shown in Figure 5.4.

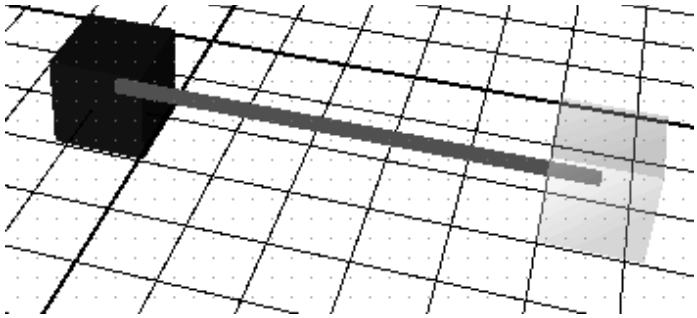

Figure 5.4 - Location Dynamic

You can select the oblique form that marks the *ending point* of the object's path and move the ending point to the object's final location.

If there are turns or bends in the path, you can click on the path to create a turning point. When you click on the path, a white ball appears on the path. When you select the white ball, it turns red and a move manipulator appears. You can grab the stopping point with your mouse and drag it along the path, or move it to form an angle. These behaviors and the move manipulator are shown in Figure 5.5.

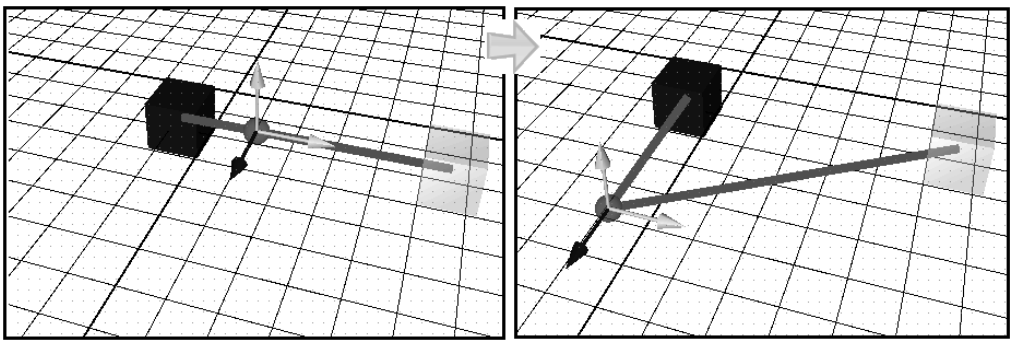

Figure 5.5 - Dragging the Stopping Point (Left) and Changing the Path (Right)

In doing this, you can establish the entire path. Figure 5.6 shows the path that was created for a conveyor in a plant.

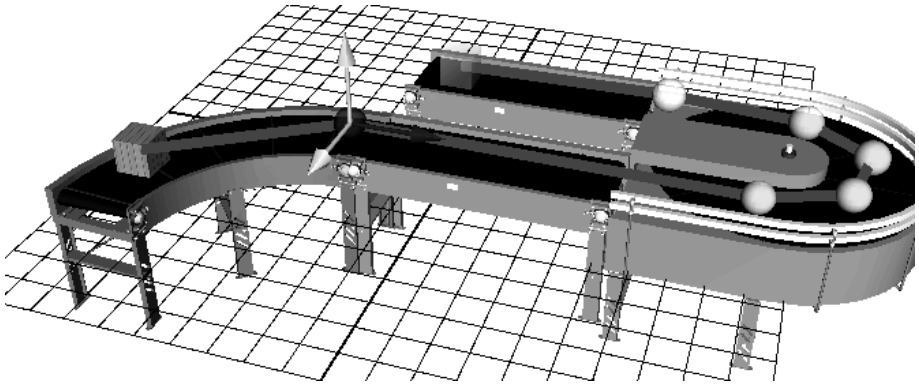

Figure 5.6 - Using the Location Dynamic to Create a 3D Conveyor

The path gets automatically written to the Location dynamic's **3DPath** property. The Location dynamics properties are shown in Figure 5.7.

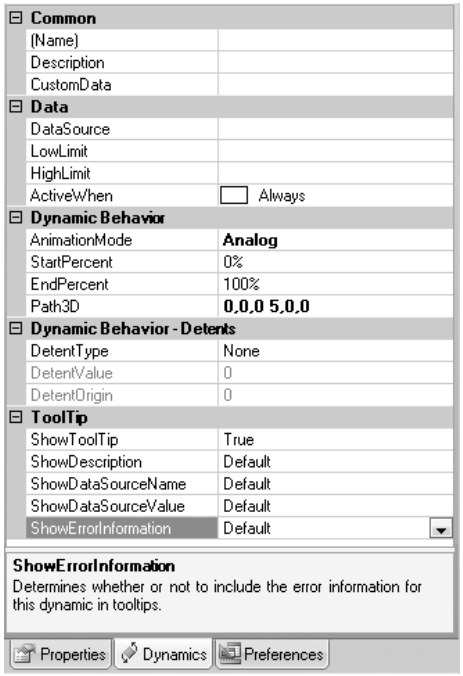

Figure 5.7 - Properties for the Analog Location Dynamic

5.3.2.1. Using a 3D Pipe Primitive as the Path

You can also easily animate an object so that, during runtime, it moves along the path of a 3D pipe primitive, which we looked at in an earlier chapter. This lets you show, for example, water moving though a pipe or people moving in a tunnel. To use a pipe as the path, you must first create the 3D pipe that you want the object to move along; you must create the entire pipe structure so that the path is complete. Only then should you use it as an object's path.

When you are ready to create the path, select the object you want to animate, then click the Location dynamic button on the Dynamics ribbon. Initially, a path from the selected object to an oblique form of the object appears, as is normal for a Location dynamic. But this time, *you're going to ignore this path.* Instead, you then must select both the object and the 3D pipe primitive.

When both are selected, right-click to display a pop-up menu, then select **Copy Path to Dynamic Location**. This adds the pipe's path to the object's location dynamic that will move the object to one end of the pipe object. The path gets automatically written to the Location dynamic's 3DPath property so you don't have to set the 3DPath property manually. The object and the 3D pipe primitive will remain in the Explorer as two separate objects.

5.3.3. The Rotation Dynamic

The **Rotation** dynamic makes an object rotate around a pivot point. It can rotate a shape through space at an angle you specify based on the data source. You can use it to spin objects like gears, or to orbit one object around another object like a planet orbits around the sun. As with other dimension dynamics, rotations can be discrete or analog.

To add this dynamic, select the object and click on the **Rotation** button on the Dynamics tab. When you do this, the object's appearance changes slightly and a pivot point appears at the center of the object, as shown on the left in Figure 5.8. You can change the location of the pivot point, as well as choose the axis and the starting and ending angle to determine how the object moves.

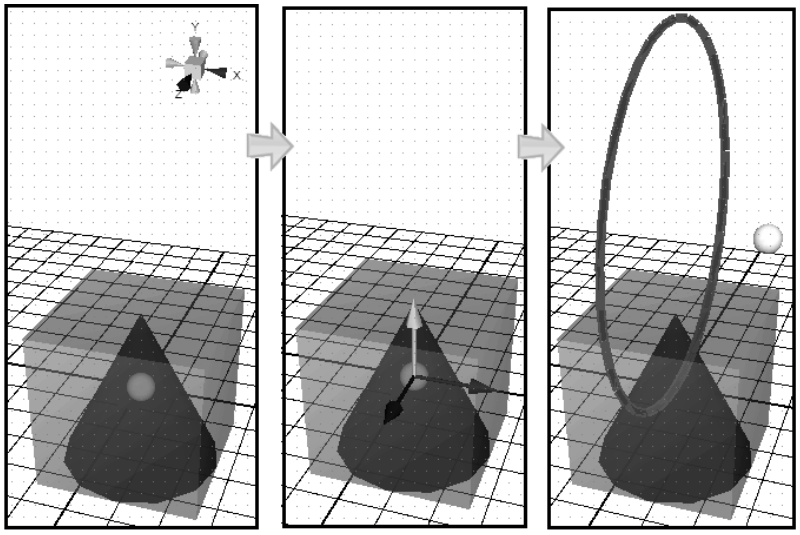

Figure 5.8 - Moving the Pivot Point for a Rotation around the X-axis

If you leave the pivot point inside the object, the object will spin. But if you offset the pivot point outside the object, the object will orbit instead. Figure 5.9 illustrates some examples of pivot point locations.

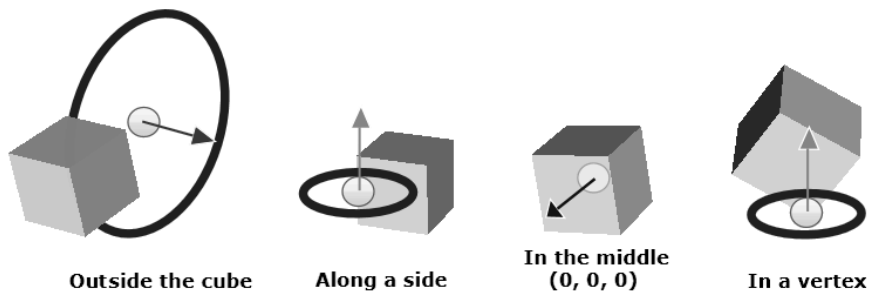

Figure 5.9 - Sample Pivot Point Locations

You must also specify which **axis** which the object will rotate around. Figure 5.10 illustrates how the rotation occurs when the pivot point is inside the object. You choose the axis using the object's Dynamic properties.

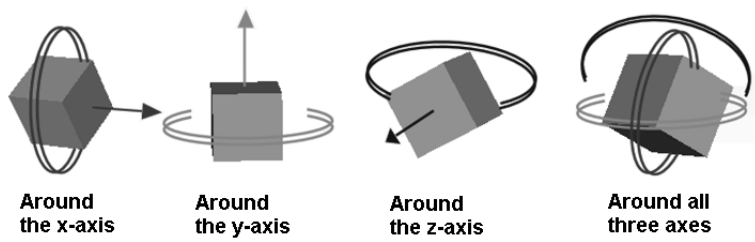

Figure 5.10 - Rotations around the Axes

Figure 5.11 shows the rotation path of a cube that rotates around the y-axis using an offset pivot point.

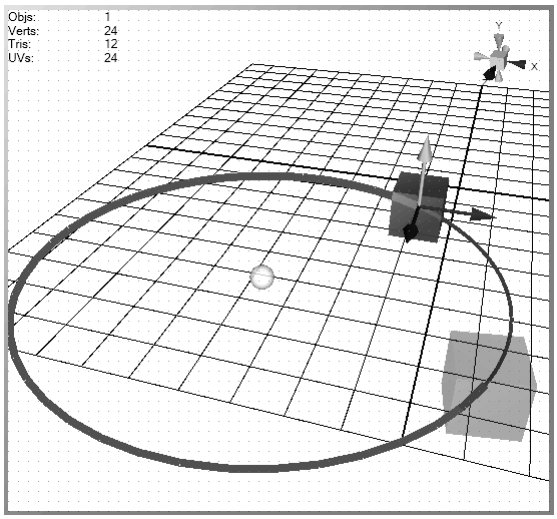

Figure 5.11 - Rotation Clockwise around the Y-axis Using a Pivot Point Offset

Figure 5.12 shows properties of an object's Rotation dynamic and highlights the Axis3D property in which the axis around which rotation will occur is being selected.

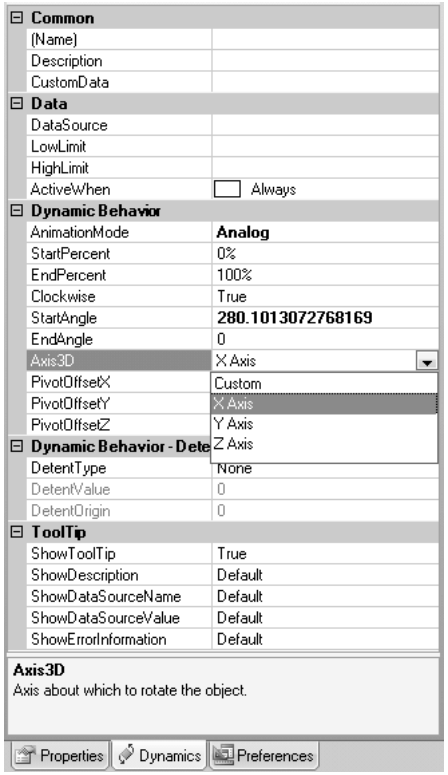

Figure 5.12 - Rotation Dynamic Properties for a 3D Object

5.4. Visibility Dynamics

Three Visibility dynamics are also available on the **Dynamics** tab. Each of these dynamics controls how the object will look in runtime depending on the value of the given data source. Let us take a look at each.

Figure 5.13 - Visibility Dynamics

5.4.1. Color Dynamic

The **Color** dynamic makes the object's color change in runtime, reflecting changes in the real-time value of a data point. Static colors can also be valuable – for example, they can reflect an object's state – while alternating colors can create visual animation effects that display indicators of rates. Figure 5.24 shows the Color dynamic's properties.

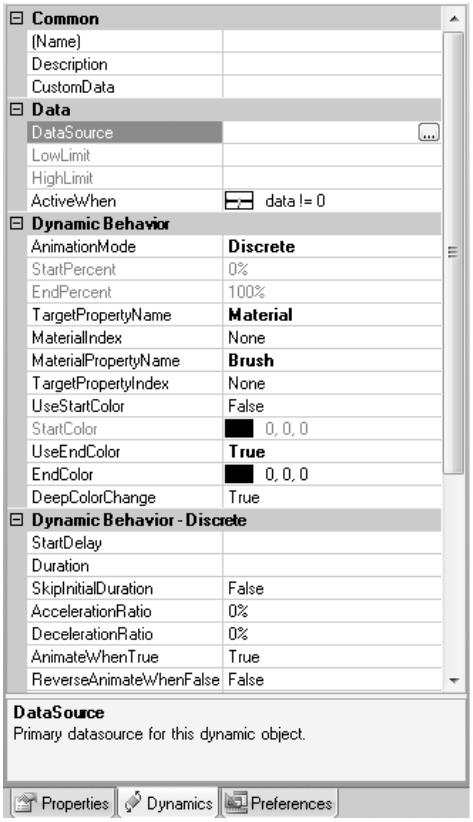

Figure 5.14 - Properties of the Color Dynamic

5.4.2. Hide Dynamic

The **Hide** dynamic conditionally hides and shows an object during runtime. You can use it to hide a disabled button, or you could use it with a machine to "peel" layers away to reveal inner workings, or use it to remove floor levels in a high-rise so that each floor can be viewed. A hidden set of objects can be used to illustrate a display state, remove items you do not want the user to be able to see, change the picture or graphic that is in view, or create animation effects.

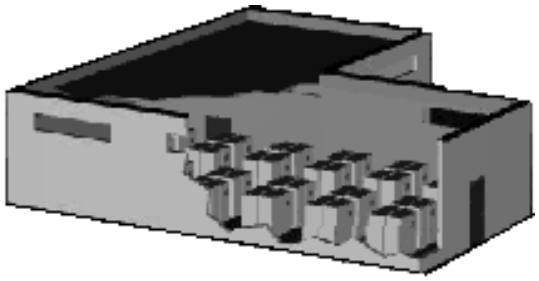

Figure 5.15 - A Building with Part of Its Outside Wall Hidden

At runtime, the Hide dynamic removes the 3D object from view or returns a hidden object so that it is shown. You should be aware, however, that hidden objects are not disabled; when an object is hidden it is still available programmatically, but it is not visible on the screen.

Common	
(Name)	
Description	
CustomData	
Data	
DataSource	
LowLimit	
HighLimit	
HideWhen	data != 0
Dynamic Behavior	
AnimationMode	**Discrete**
StartPercent	0%
EndPercent	100%
ExplicitHide	True
ExplicitShow	False

Dynamic Behavior - Discrete	
StartDelay	
Duration	
SkipInitialDuration	False
AccelerationRatio	0%
DecelerationRatio	0%
AnimateWhenTrue	True
ReverseAnimateWhenFalse	False
AutoReverse	False
RepeatCount	1
FillBehavior	HoldEnd
FreezeWhenNotAnimating	False
Flash	
FlashRate	
HideWhenNotFlashing	False
ToolTip	
ShowToolTip	True
ShowDescription	Default
ShowDataSourceName	Default
ShowDataSourceValue	Default
ShowErrorInformation	Default

Figure 5.16 - Hide Dynamic Properties

5.4.3. Disable Dynamic

The **Disable** dynamic deactivates any actions that are associated with a particular object. Disabling an object is a great way to ensure that program logic in your displays is enforced, or to disallow an action that you do not want the user to be able to perform. For any multi-step action, the Disable dynamic is always the last dynamic because it deactivates actions.

At runtime, a disabled object remains in view, but does not respond to keyboard or mouse input. You can disable objects based on a data source, a value you set, or the user (for example, you can make sure a non-administrator cannot use a particular button).

You might use this dynamic to create a button that is disabled when a value is within a specific range; like a Release Valve button that is disabled when the water pressure is within normal range.

Common	
(Name)	
Description	
CustomData	
Data	
DataSource	...
LowLimit	
HighLimit	
DisableWhen	data != 0
ToolTip	
ShowToolTip	True
ShowDescription	Default
ShowDataSourceName	Default
ShowDataSourceValue	Default
ShowErrorInformation	Default

Figure 5.17 - Properties of the Disable Dynamic

5.5. Collision Detection

The Collision Detection System lets you know if two or more objects collide in a 3D View control. There are a few reasons why you might use this feature. You may want to make sure that your objects do not hit each other during runtime operations, because although the real objects are not colliding, the operator may get harmed or distracted by the 3D representation of the environment. Another reason to use it is to try to prevent actual collisions due to moving parts.

Collision Detection is a test of dynamic and animated objects in a 3D viewport of a GraphWorX64 display. It tests any objects, dynamic or static, that you believe could collide in a 3D GraphWorX64 display. For example, if the display simulates a system of conveyors and robot arms, you can test the objects that make up the conveyors and robot arms, the objects that interact with the conveyors and robots, and the static objects, such as building walls or ovens, around the conveyors and robot arms.

The collision detection test is only interested in the geometries of registered objects and doesn't know anything else about them. Collision detection doesn't even know about the location, rotation, and size dynamics that are associated with objects. It simply reads the current geometries of the registered objects in each rendering loop and tries to find a collision between them. When it finds a collision, it reports that collision using TraceWorX, GenEvents, or by changing the color of objects in the display when they collide.

In setting up a collision test, you identify the objects to be tested, and then identify pairs of objects that you suspect may collide. To identify the 3D objects to be included in the

test, select the object(s) in the 3D viewport, then click the **Dynamics** ribbon's **Include** button. Repeat this step until all of the objects you want to include have been selected. (If you include an object by mistake, you can select it and click the Exclude button.)

Next, you click the **Settings** button on the **Dynamics** ribbon. This opens the Collision Detection dialog box, shown in Figure 5.18. The objects you have selected appear in the **Registered Bodies** box on the left.

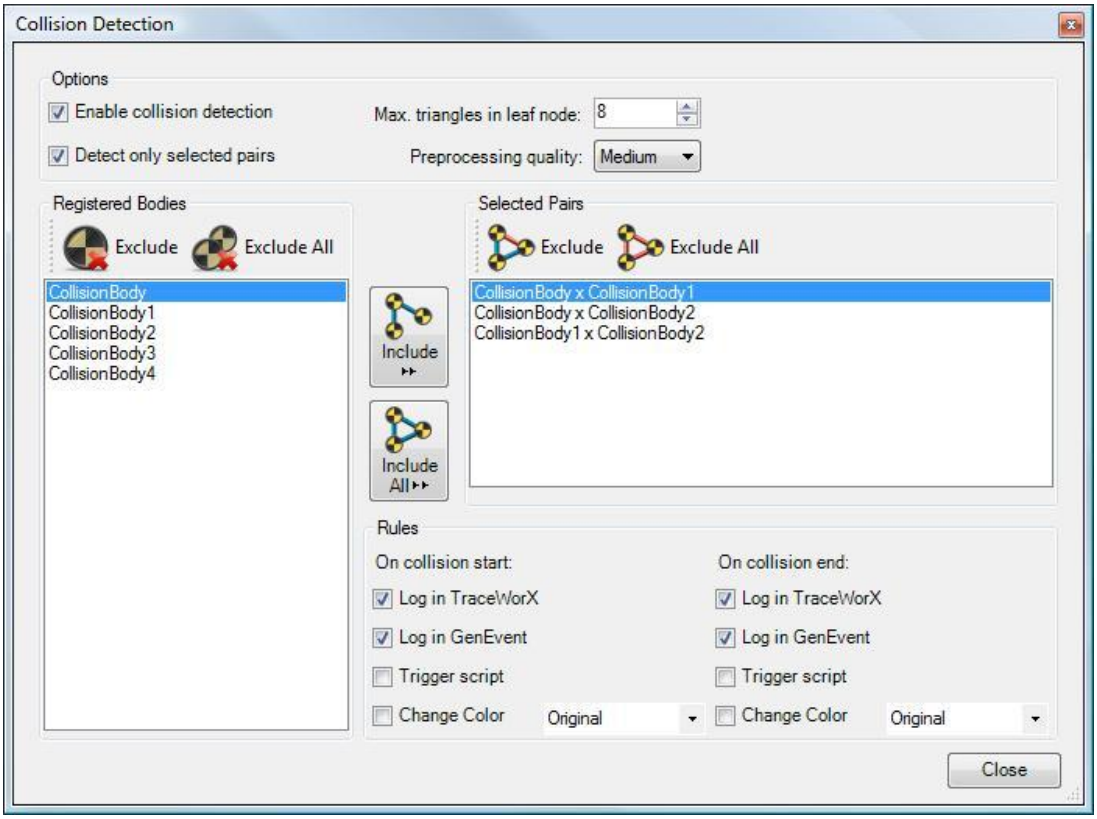

Figure 5.18 – Collision Detection Dialog

To enable fields in the dialog box, put a checkmark in the **Enable Collision Detection** checkbox. Then you want to test every possible pair of objects for collision detection by making sure the **Detect Only Selected Pairs** checkbox is unchecked ☐; or select ☑ the Detect Only Selected Pairs checkbox then click the **Include All** button.

Alternatively, to test selected objects for collision, select the objects and click the Include button. The included objects appear as pairs in the **Selected Pairs** box on the right. In the **Rules** box, specify the how you want collisions to be reported. You can choose between logging the collision in TraceWorX, posting an event to GenEvent, triggering a script to perform a custom action or changing the object color for visual identification.

When you are done, click **Close**, then enter Runtime mode to begin the test. When your test is over, you can review the test result. If you selected to report the collisions to TraceWorX or GenEvent, Figure 5.19 and Figure 5.20 show how the reports may look.

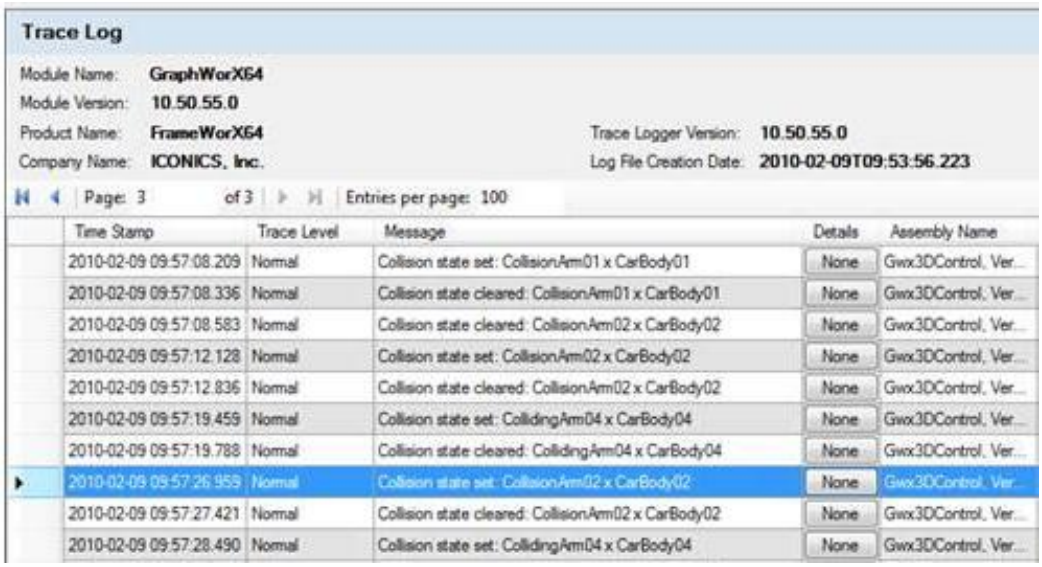

Figure 5.19 - Collisions Logged in TraceWorX

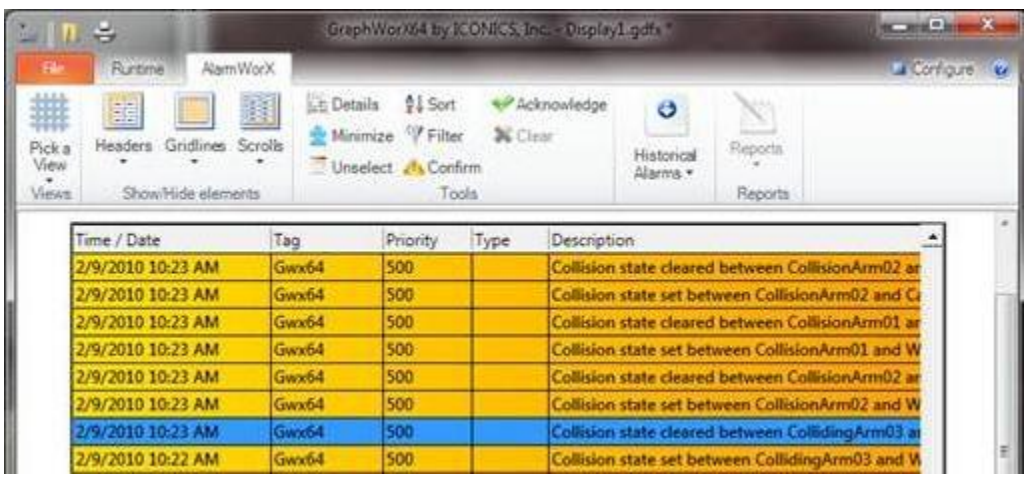

Figure 5.20 - Collisions Posted to GenEvent

5.6. Polygon Reduction

You may need to use a special 3D model or purchase one already built from another source. When models are built using third party software, they are generally built to represent objects in great detail. In a human machine interface (HMI) project, many of the details become unnecessary and reducing the model's complexity will decrease demands on hardware to rend it. In these cases, you need to reduce the details on the object. This can be as easy as reducing the number of polygons in the model.

Create any object or import it from a file. If you have your 3D object inside 3D View and it is now selected you can go to the Advanced tab in the GraphWorX Ribbon Menu and press the Make Reducible button.

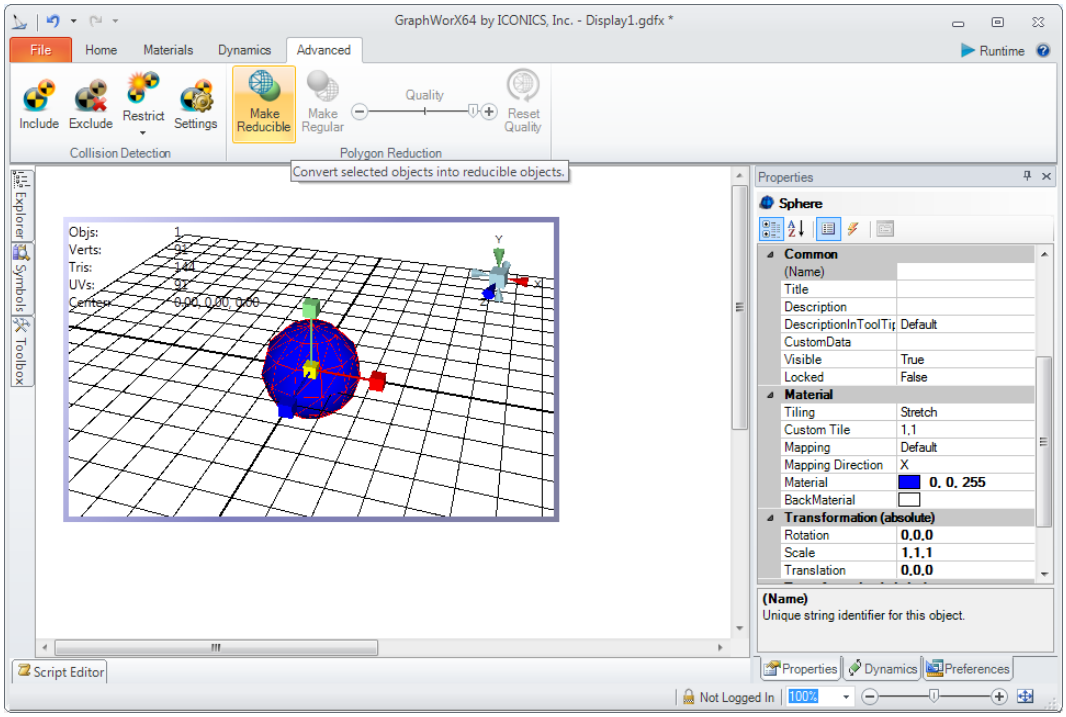

Figure 5.21 - Collisions Posted to GenEvent

Once your object is transformed to a reducible state, you can use the minus-plus slider to change the level of reduction. Press the Reset Quality button to return to your original setting. If you want to return your object to Regular, select it and press the Make Regular button.

5.7. Animating Views at Runtime

Views give you added power during configuration and runtime because they provide quick and easy access to certain areas of your 3D scene. The **Camera** button in the **View** section of the Home tab is used for saving views in a 3D scene. A saved view in 3D is like a saved view anywhere else in GENESIS64 – it is a saved location in a scene that *you* want to be able to go to easily during configuration of the scene or that you want *runtime users* to be able to access quickly during runtime.

For example, if a scene includes a room full of machines, you might save a view of each machine's controls. Then in runtime, users can right-click on the 3D view and select any other view they want to go to in order to focus in on certain information. 3D animation takes the user to that scene in a movie-like sequence. The animation helps

the user locate the view in the overall scene. You can turn off the animation feature so that views simply display without the camera simulating movement from one part of the scene to another.

Let us look at an example, using a nuclear power plant's campus. You can try this yourself by opening GenDemo and browsing to GENESIS64 Examples and select the Nuclear Plant example. Once the display opens, you can right click on the display to show a list of views, like the example in Figure 5.22.

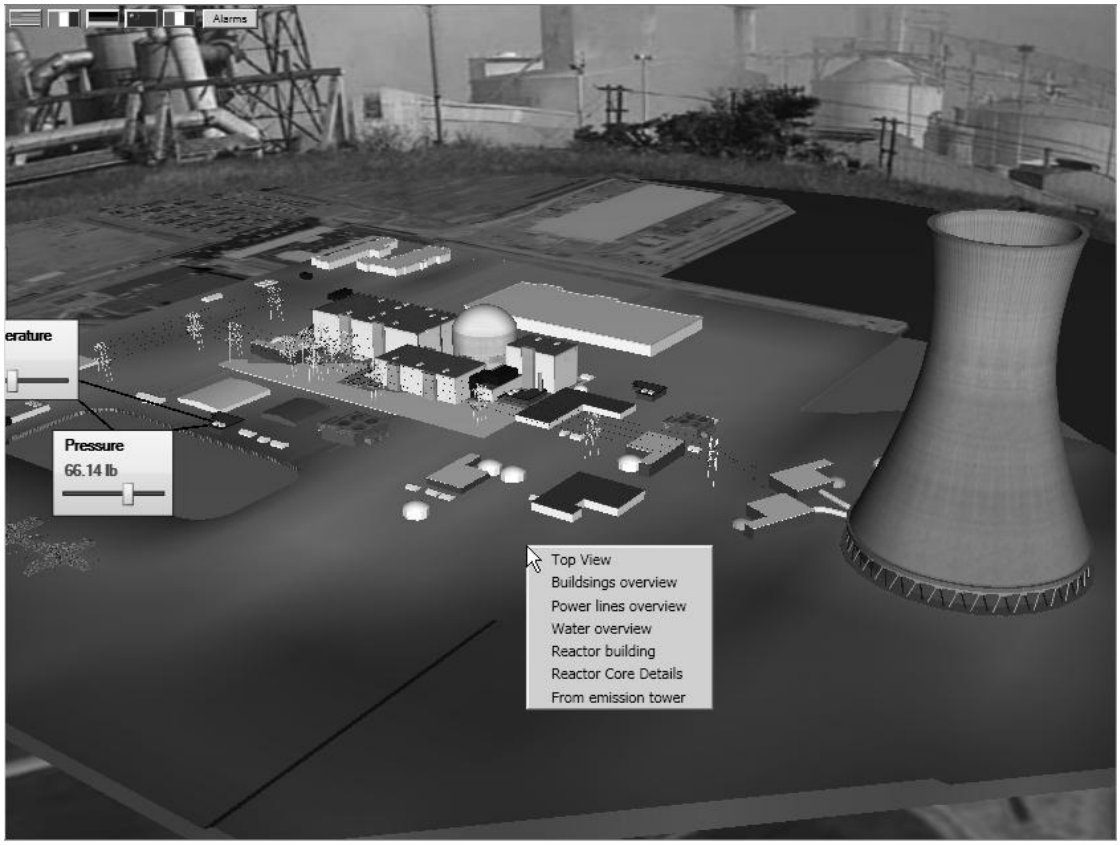

Figure 5.22 - Runtime View Menu

You can select any view in the list and it will bring you to a different location of the scene, providing focus on a different part of the plant. Figure 5.23 shows a few of the views available in the 3D scene shown in Figure 5.22.

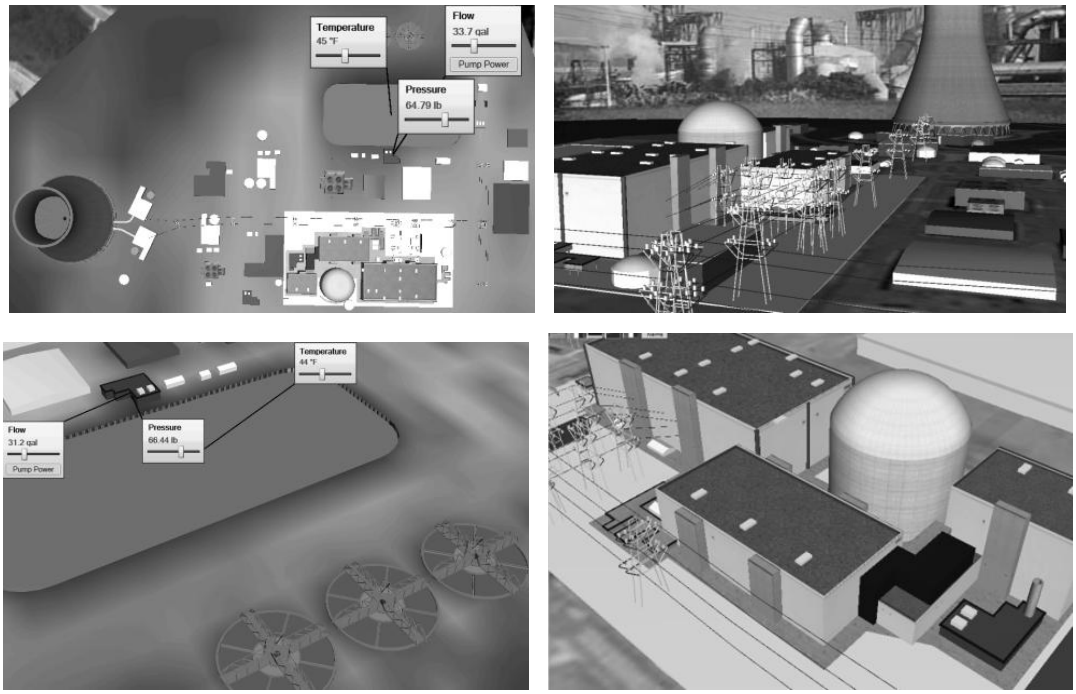

Figure 5.23 - Different Views of the Nuclear Power Plant

5.7.1. Managing Views

In order to change your view in runtime, or create a movie-like scene, you must first create a library of views you will need in runtime. To save a view, you will first need to manipulate the scene to show you the particular information you are interested when changing to the view. You can zoom into a particular section of the 3D viewport to see something more clearly.

Once you are satisfied with what you see on screen, click on the down arrow on the Camera views button. This will give you a drop-down menu with an option to add the current view.

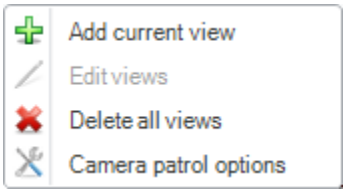

Figure 5.24 - Camera View Drop-down Menu Options

Select the **Add Current View** option from the drop-down menu. The *Custom Views* dialog displays as shown in Figure 5.25.

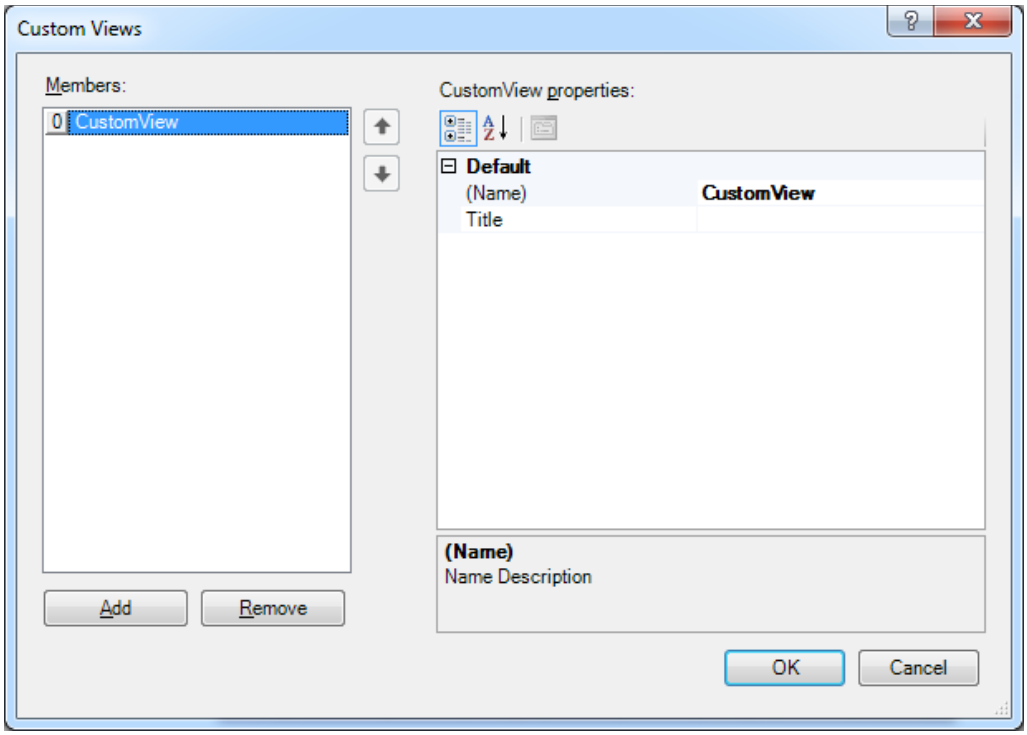

Figure 5.25 - Add a View

Give the view a name and the title you want to use to identify this view and click on OK to save the view and close the dialog. When you click the Camera views drop-down button again, you will see the view in the menu.

In this dialog, you can also change the order of your views by selecting the view, and clicking on the up or down button to reorganize the order of your views. When you are done, you can click on OK to save your changes. If you change the order of your views using this dialog, you can see the changes reflected in the Camera views dropdown menu.

There is also an option on the Camera views drop-down list that allows you to delete all views that you have created. This option resets the views you have in the display by clearing it. If you want to delete just one view instead of all of them, you can use the Edit Views menu item.

> ***Tip!*** A view is the coordinates of the camera position in a scene; it is not "aware" of the objects it highlights or displays. Therefore, if you change the location of an object, the view does not move with it. If you want the view to show the moved object, you must change the view, or delete the old view and create a new one for the location where the object is now.

5.7.2. Controlling the Runtime Behavior of Views

As mentioned previously, you can create a movie-like scene using views. To do this, you will use the **Camera Patrol Options**, which allows you to define your "movie" settings. When you select the **Camera Patrol Options** from the Camera view's drop-down menu, the Camera Patrol Options dialog appears, as shown in Figure 5.26.

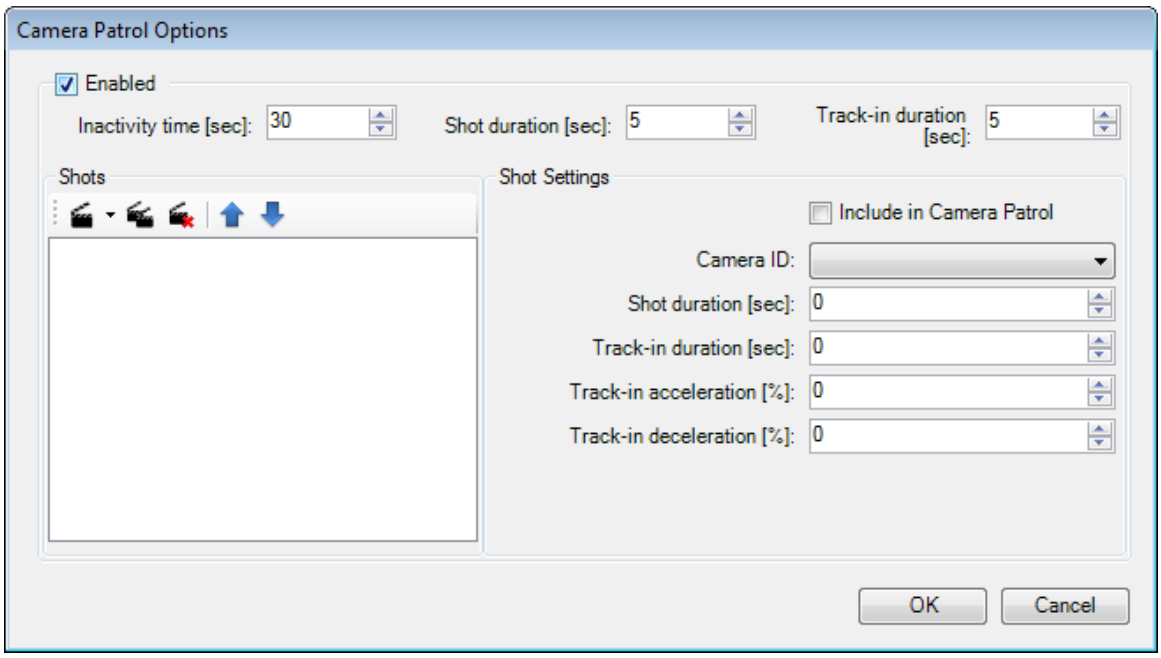

Figure 5.26 - Camera Patrol Options

You can set up a camera patrol that automatically walks through the views during runtime by putting a checkmark in the **Enabled** checkbox. In runtime, after the user is inactive for a specified period of time (that is, the number of seconds you specify in the **Inactivity time (sec)** field), the camera patrol will automatically cycle through the views in the specified sequence. The **Shot duration (sec)** says how long the specified view will remain visible before moving to the next one. **Track-in duration** allows you to specify how long the animation between two views should last (0 means no animation). You can also specify an acceleration/deceleration for the camera movement if you want.

The settings in the above part of the dialog are used as the default settings for each new shot added to the time line, however you can customize each detail for each single shot.

In the following example we have added our three views in the Camera Patrol Shot list to create six different shots. Each of them has a custom duration and Track-In timing.

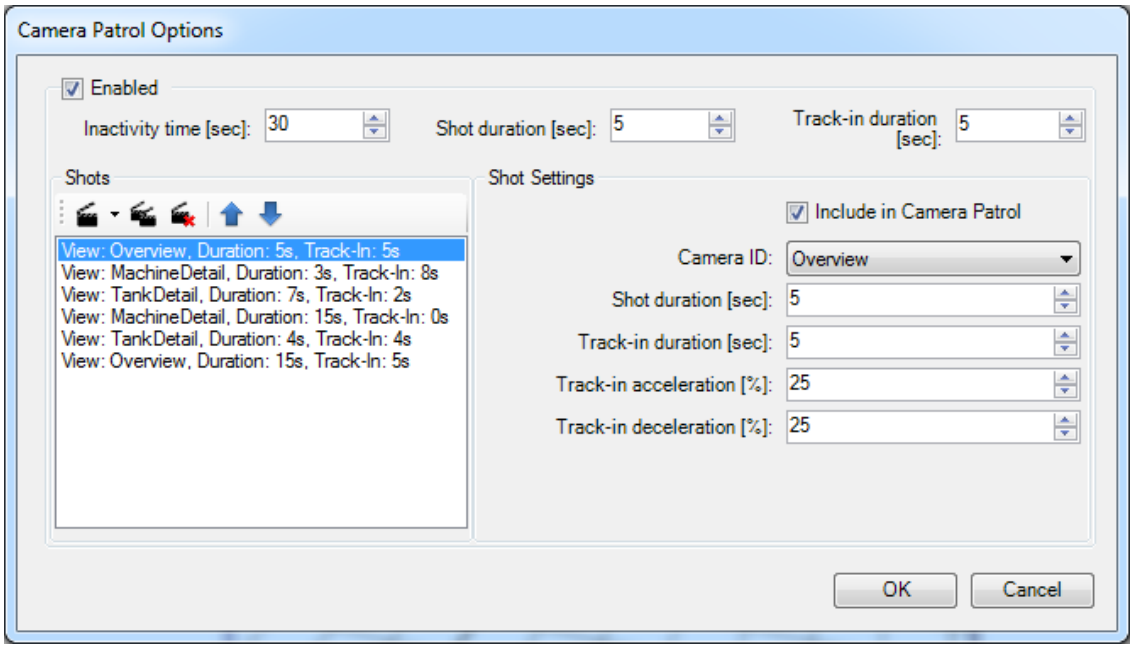

Figure 5.27 - Camera Patrol Options

After you have created your time line, you can always get back to this dialog to add more shots, remove some, sort them or change their properties according to your needs.

Lab: 3D Dynamics *L5*

In this lab, we will explore the different dynamics available in the 3D environment, which will give you some practice in creating 3D animations. We will have similar animations for the icTonics 3D display.

Estimated time to complete this lab: 20 minutes

5.1. Task Checklist

In this lab, you will complete the following tasks:

- Add a Size dynamic
- Add a Location dynamic
- Add Hide dynamics
- Use discrete animation to reveal hidden information over time

5.2. Support Files

You do not need any supporting files to complete this lab

5.3. 3D Dynamics

We will first explore the different Position dynamics before we create animations.

1. Create a new GraphWorX64 display and name it “Practice2.gdfx”.
2. Add a 3D viewport that is the size of the display by clicking on the **3D View** button in the Shapes section of the Home tab, then drawing the viewport in the display with you mouse.
3. Add a Cube primitive onto the display.
4. In the 3D ribbon, select the Dynamics tab and click on the Size button in the Positions section.
5. In the Dynamics Explorer, set the data source to “localsim:sine”.

6. In the Dynamics Behavior section of the Dynamics Explorer, change the Size Mode so that the cube grows in all directions.

7. Change the StartPercent field to 30%. Your Dynamics Properties should look similar to Figure L5.1 when you are done.

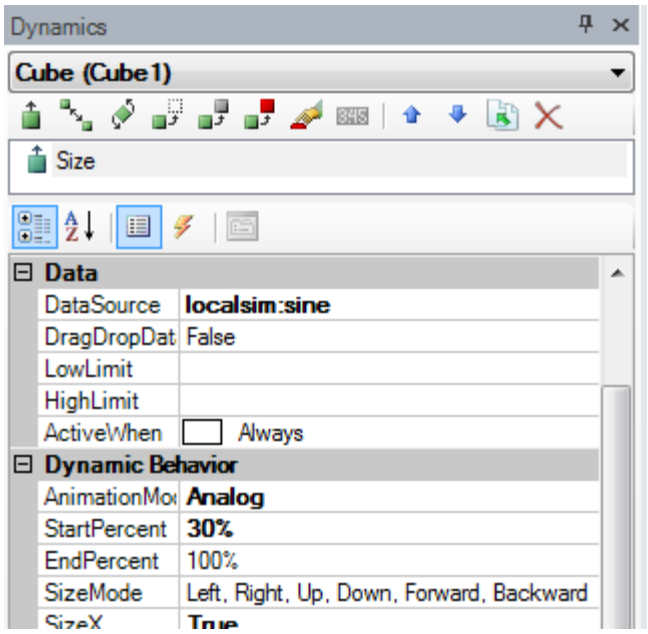

Figure L5.1 - Dynamics Properties for the Size Dynamic

8. Save the display and go into runtime. You should see the cube growing and shrinking in accordance with the data source value.

9. Go back into configuration mode and add a Location dynamic to the cube by selecting the cube, and then clicking on the Location button in the Dynamics tab.

10. Set the data source of the Location dynamic to the “localsim:sine” again.

11. Select the gray oblique cube in the display. Click on the X-axis arrow and drag the gray cube further away from its original location.

12. There is nothing else that we need to change. The Dynamics Explorer for the Location dynamic should look similar to Figure L5.2.

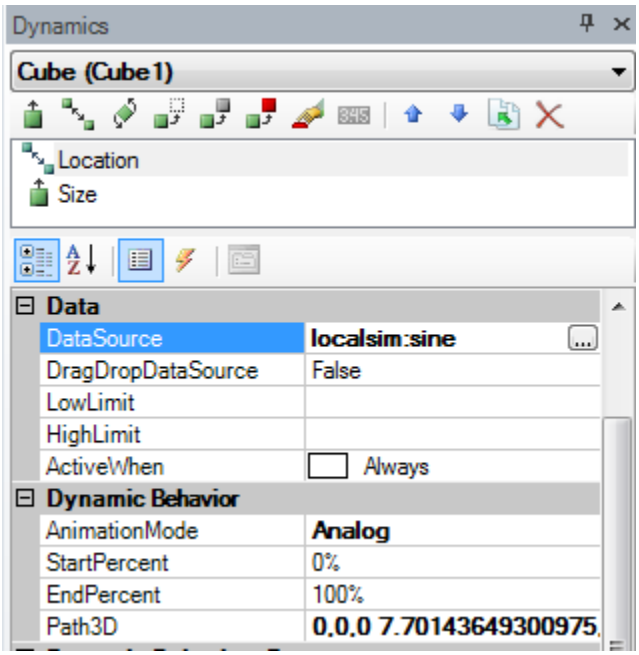

Figure L5.2 - Dynamics Properties for the Location Dynamic

13. Save the display and go into runtime again. You should now see the cube grow as it moves toward the end destination and shrink as it goes back to the original location.

5.4. Discrete Animation

In the previous section, we practiced adding Analog animations. In this section, we will practice adding Discrete animations. You will see the difference between the two.

1. Move the cube away from the center of the viewport if it is still in the center to make space for our next practice exercise.

2. In the Symbols library, select _Vertical Markets → Food & Beverages → 3D → Food Boxes and add the Food Box symbol into the viewport.

3. Rotate the box so that the face of the box is facing you as shown in Figure L5.3.

Figure L5.3 - Box Food

4. Resize the object so that it is smaller than the cube we added earlier.

5. Add a Sphere to the display and make it look like a dome by changing the Hemisphere2 properties to 0.5.

6. Add a Location dynamic to the dome. Move the end location so that it is over the dome.

7. In the Dynamics Explorer, change the Location dynamic's AnimationMode in the Dynamics Behaviors section to Discrete.

8. Set the Duration to 3000 (3 seconds).

9. Next you are going to change the data source to a simulation tag. You can create the tag by typing "Activate" as the name.

(Name)	Activate
Mode	None (Local Variable)
DataType	Double-Precision Float
LowLimit	0
HighLimit	100
InitialValue	0

Figure L5.4 - New Simulation Tag

10. Your Location dynamic's properties should look similar to Figure L5.5.

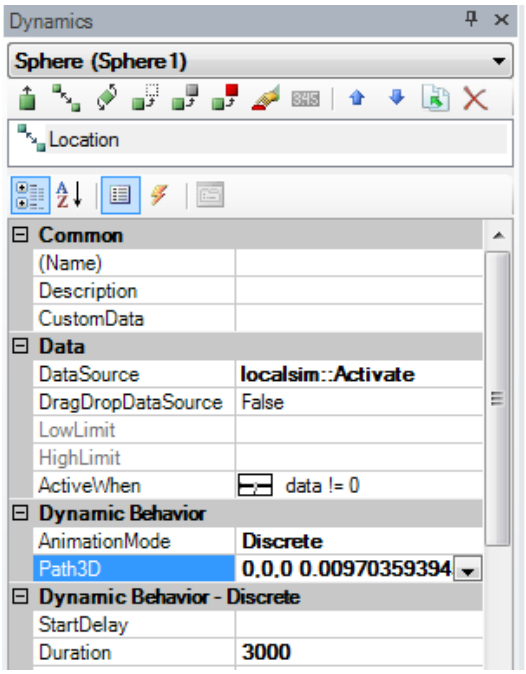

Figure L5.5 - Location Dynamics Properties

11. Add a Hide Dynamic to the Dome and set the DataSource to "localsim::Activate".

12. Make sure the AnimationMode is Discrete and change the StartDelay field to 3000 and the Duration to 1000. The properties for the Hide dynamic should look similar to Figure L5.6.

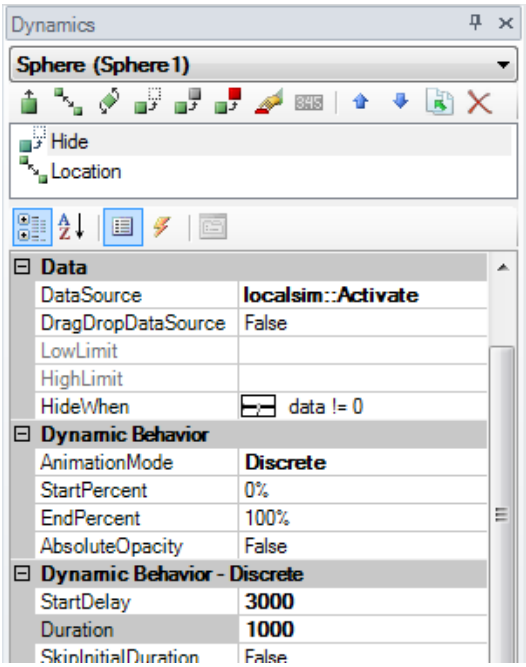

Figure L5.6 - Hide Dynamic Properties

13. In the 2D space, create a button called “Activate”.

14. Change the Command to Write Value. Set the data source to “localsim::Activate” and change the OnUpValue field to 1.

15. Go into runtime and click on the Activate button. You should see the dome slowly move up and disappear.

16. Go back into configuration mode and, in the 3D viewport, duplicate the Sphere two times and move them above the original one.

17. Change the radius of the domes to 0.9, 1.2, and 1.5 respectively. Your display should look similar to Figure L5.7.

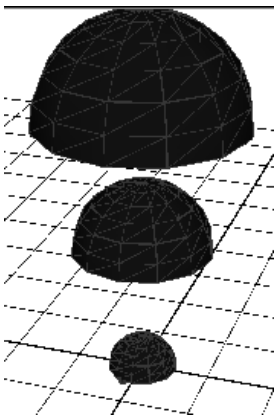

Figure L5.7 - Three Domes

18. On the middle dome, go to the Location dynamic and add a StartDelay of 3000. Then go to its Hide dynamic and change the StartDelay to 6000.

19. On the bottom dome, go to the Location dynamic and add a StartDelay of 6000. Then go to its Hide Dynamic and change the StartDelay to 9000. This will allow the three domes to move up and disappear in sequence.

20. Confirm that the top dome’s Location dynamic doesn’t have a StartDelay, and its Hide dynamic has a StartDelay of 3000.

21. In the object Explorer, rename the domes “TopDome”, “MiddleDome”, and “BottomDome”.

22. Select all three Spheres with the bottom dome as the key object, then align them in the center on the X and Z axes.

23. Also, select the Food Box symbol with the three spheres and align the bottom on the Y axis.

24. At this point, your three spheres and the Food Box object are aligned. You should now work with these objects using the Explorer; otherwise, it would be almost impossible to select the objects that are inside the biggest dome.

25. Select the Food Box symbol and the bottom dome and align them in the center on the X and Z axes.

26. Save the display and go into runtime. Click the Activate button and watch the results.

27. You may want to re-set the end locations of the domes so that the bigger domes do not hide the smaller ones when they go into the hide state.

28. After you re-set the end locations, save the display and go into runtime to see the results.

Review: 3D Animation R5

5.1. Chapter Objectives

After completing this chapter, you should know how to:

- Animating objects to mimic real-world objects at runtime
- Integrating the user experience into an object's animation
- Using views to provide a movie-like experience at runtime
- Detect virtual collisions in animated objects

5.2. Summary

Graphics themselves would be nothing if it were not for the data behind them that animate the graphics. There are two types of animation to the HMI integrator: data driven animation and time-based animation.

Data-driven animation is bound to a particular data that is coming from a device. Real-time data can animate the 3D object to give you a real-time human-machine interface. Use **time-based animations** for constructing animations that work like movies. With a time-based animation, you decide the timing and sequence for the object's movement in the display. Time-based animations are useful for assembly or repair instructions.

The **Dynamics** tab of the ribbon contains options for connecting your 3D objects to real-time data and animates them. On this tab, the **Data Source**, **Text Objects** (that is, the Process Point, Data Entry, and Time text objects), and the **Pick Action**, all work the same way as they would in 2D. **Smart Symbols** and **Selectors** are the same as 2D, too.

The **Size**, **Location**, and **Rotation** dynamics in the **Position** section are similar to their 2D equivalents, but have their own unique nuances in 3D mode. The same is true of the **Color**, **Disable**, and **Hide** dynamics in the **Visibility** section. The **Collision Detection** system lets you know if two or more objects have the potential to collide in a 3D view control, helping you to avert actual collisions in the process.

The **Camera views** button in the **View** section on the **Home** tab of the ribbon is used for saving views in a 3D scene. When users select a view, the 3D animation can

take the user to that view in a movie-like sequence. You can control how that movie-like sequence unfolds.

5.3. Terms

Camera View Collision Detection Dynamics Saved View

5.4. Test Your Knowledge

This is the concluding chapter of this course. In this chapter, we discussed the different dynamics available in GraphWorX64 3D as well as camera views. You should now be able to answer the following questions about 3D animations in GraphWorX64.

1. How do the Data Source, Text Objects, Pick Action, Smart Symbols, and Selectors work differently in 3D than they do in 2D?

   a. They have voice animation
   b. They have limited function in 3D
   c. There is no difference in how they work
   d. These dynamics are not available in 3D

2. True or False: You cannot use animation with real-time data.

3. What do you need to do if you want to change a 3D object's size in runtime after you add the size dynamic to it?

   a. Use a script to change the object's size
   b. Set the dynamic's SizeMode property
   c. Start the recorder and use the Manipulators on the Home tab of the ribbon
   d. Nothing, the object knows what to do automatically

4. How do you make an object spin in place?

   a. Use the Rotation dynamic and leave the pivot point inside the object
   b. Change the orientation of the X, Y, and Z axes.
   c. Use the Location dynamic and set the Spin property to True
   d. Use the Rotation dynamic and set its Spin property to True

5. An assembly line has been reconfigured and you are rearranging the objects in a 3D view to mimic the new assembly line configuration. In the older version of the 3D view, runtime users had access to saved camera views that focused

on key workstations in the assembly line. Will those views still be useful for runtime users?

a. Yes, a view follows the object wherever you move it.
b. Only if the workstations did not move. If the workstation objects moved, you must delete the old camera views and create new ones for the objects at their new locations.
c. No, you must recreate views every time you make any change to a 3D view.
d. None of the above.

6. True or False: To create a path that bends, you must use two Location dynamics and group them.

7. True or False: The Color dynamic cannot reflect real-time data in 3D.

8. True or False: Hidden objects are always disabled.

9. True or False: At runtime, a disabled object remains in view, but does not respond to keyboard or mouse input.

10. True or False: The Camera Patrol option configures runtime to cycle through the camera views after a specified duration of user inactivity.

Lab: icTonics Production Line L6

You should now have all the tools you need to create a 3D display for icTonics' beverage production line. This display will help icTonics' operators to better visualize the production floor for control.

Estimated time to complete this lab: 120 minutes

6.1. Task Checklist

In this lab, you will complete the following tasks:

- Generate a 3D display for icTonics
- Insert 3D viewport
- Use the Symbols Library
- Use Primitives to create objects
- Use Dynamics to create a live display
- Create Smart Symbols

6.2. Support Files

You will need the following files to complete this lab:

- icTonicsTemplate.tdfx

6.3. icTonics Beverage Tank

In this lab, we will create one production line on the icTonics Production floor. First, we will create the mix tank from which we use to fill our beverage bottles.

1. Open a new GraphWorX64 display and apply the icTonics template to it. You can save the display as "icTonics US Facility.gdfx". You should periodically save the display from now on.

2. Insert the "Empty Tank" you created in the previous lab from the Symbols Library. This opens the tank inside a 3D viewport.

3. Flip the tank around the Z axis so that the Sphere object is on the top.

4. Double-click outside the 3D viewport to deactivate it, then grab and drag its handles to resize it so that it takes up the entirety of the white space below the banner.

5. Reactivate the 3D viewport. Then add a new cylinder to the 3D display and change its radius to 0.4.

6. Align the new cylinder to the center of your tank and stack it on top of the sphere that's part of your tank.

7. Add a Size dynamic to the newly-added cylinder and set the data source to the Foxboro tank 1 tank level tag. Move the tank aside; you are done working with it for a while.

8. Add another cylinder with a radius of 0.05, to create a post. Then add another cylinder perpendicular to it, giving it the same radius and a height of 0.3. It should look like Figure L6.1.

Figure L6.1 - Perpendicular Cylinders

9. Select the horizontal cylinder and click the Transform icon that duplicates the selected objects and applies a transformation. Set the transform properties as shown in Figure L6.2 and click OK.

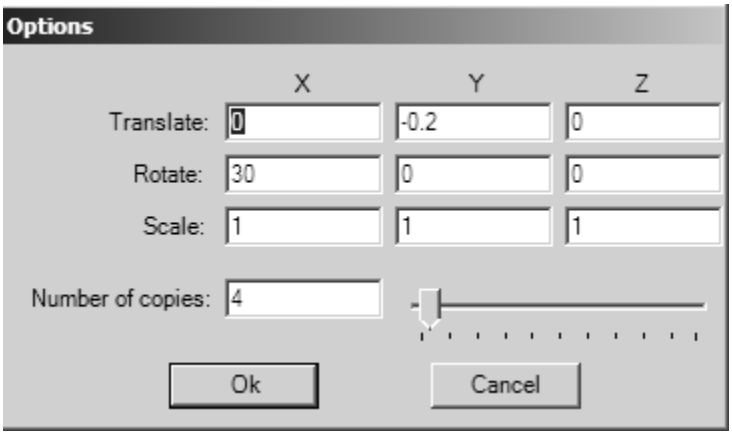

Figure L6.2 - Transform Properties

10. You should now have a post with spokes that is similar to Figure L6.3.

Figure L6.3 - Spokes

11. Now slide each spoke so that one end is attached to the post. Your finished object should look similar to Figure L6.4.

Figure L6.4 - Churning Blade

12. Group the six cylinders together and name it Churning Blade.

13. Add a Rotation dynamic to the Churning Blade and change the data source to the ChurningBladeStatus tag in Foxboro Tank 1. You may want to change the pivot point of the rotation dynamics to the center of the middle pole.

14. Change the dynamic's AnimationMode to Discrete and change the Axis3D to the Y axis.

15. Set the duration to 5000 and change the RepeatCount to Infinite.

16. Depending on how your spokes are placed, you may want to change their rotation direction using the Clockwise property.

17. Also, add a color dynamic to the Churning Blade. Again, set the data source to the ChurningBladeStatus tag in Foxboro Tank 1.

18. Change the color dynamic's ActiveWhen property to **data == 0** and set the EndColor to red.

19. Now align the Churning Blade group to the center of the tank.

20. Add some dynamics to the Sphere of the EmptyTank3D object so that the top of the tank will move away when you click on it.

HINT: You may want to use a Toggle Value Pick Action with a local simulation tag to control when the top moves away and when it comes back.

21. Now group all the tank objects together and call it “Tank”.

22. On the Dynamics tab of the ribbon, click on the Convert Symbol in the Smart Symbols section.

23. Click OK to the window that pops up.

24. In the Edit Smart Properties Dialog, change the name of the properties accordingly and click on OK.

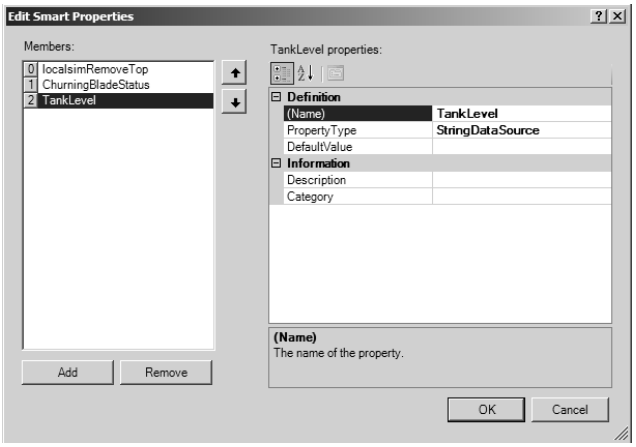

Figure L6.5 - Smart Properties

25. Copy and paste the smart symbol into the Symbols library and name this symbol “Smart Tank”.

26. Your finished tank should look similar to Figure L6.6.

Figure L6.6 - Smart Tank with Churning Blades Inside

27. We will set the Tank aside for now. You can hide the tank by clicking on its eye icon in the Explorer so that it’s not in the way.

6.4. icTonics Production Line

Next, we are going to create the production line for the beverages, including the conveyors, the bottles, the tank, and how they work together.

1. In the Symbols library, go to Conveyors → 3D → Conveyors.
2. Drag the "Conveyor Angle 45" and "Conveyor Straight" into the 3D viewport. Since these symbols are made up of multiple parts, it will be much easier to work with them using the Explorer.
3. Duplicate each of the conveyors once and align the conveyors so that they look like the production line in Figure L6.7.

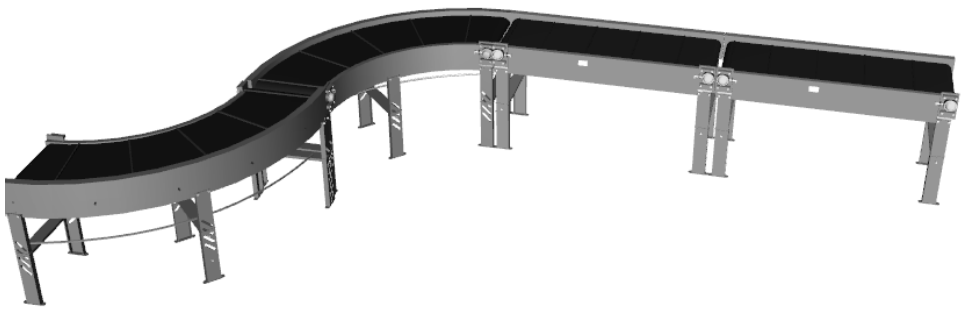

Figure L6.7 - Conveyor Line

Hint: You can rotate the conveyor to the angle that you need. You can also use the align option to move the conveyors to the correct location or one that's close to what you need, then fine tune it. You can also try changing the orientation and zoom level of the viewport for easier manipulation.

4. Drag the "Empty Bottle" out from the GENESIS64TrainingSymbols and stack it on top of the conveyor.
5. Duplicate the Empty Bottle seven times so that there are 8 empty bottles all together. Place these bottles on the conveyor as shown in Figure L6.8.

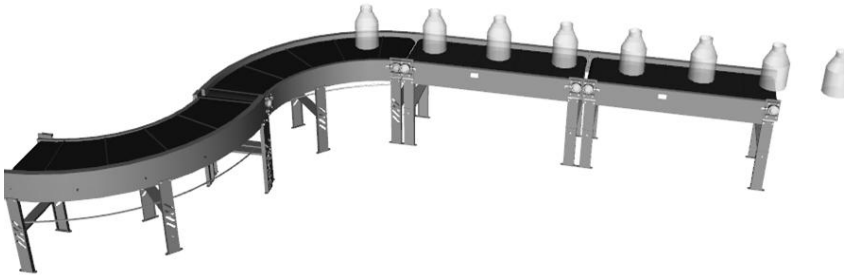

Figure L6.8 - Conveyor with Empty Bottles

6. Drag the “Full Bottle” out from the GENESIS64TrainingSymbols and stack it on top of the conveyor. Again, duplicate this bottle so that it looks like Figure L6.9.

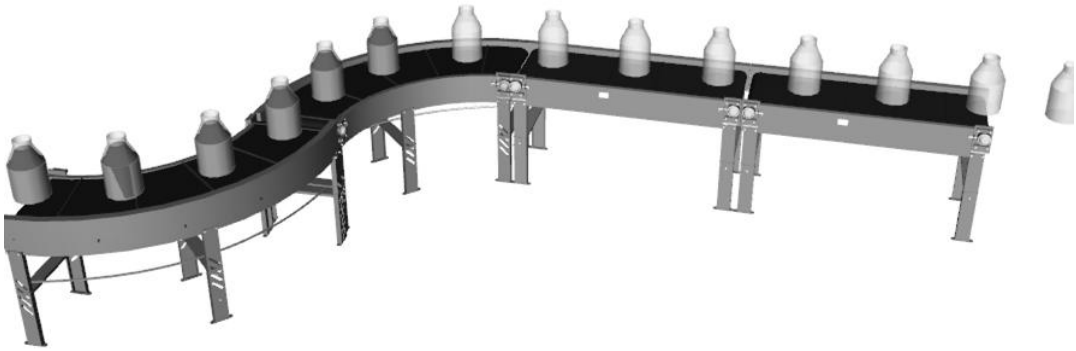

Figure L6.9 - Conveyor with Bottles

7. For each of the bottles, add a Location dynamic that moves each bottle to the next bottle’s position. For all of these dynamics, you should use the same simulated Data Source (let’s call it “StartConveyor”). The Animation mode should be Discrete and the Duration should be set to 1000 milliseconds. Figure L6.10 shows the properties of the Location dynamics for one of the bottles. All bottles should have similar dynamic properties.

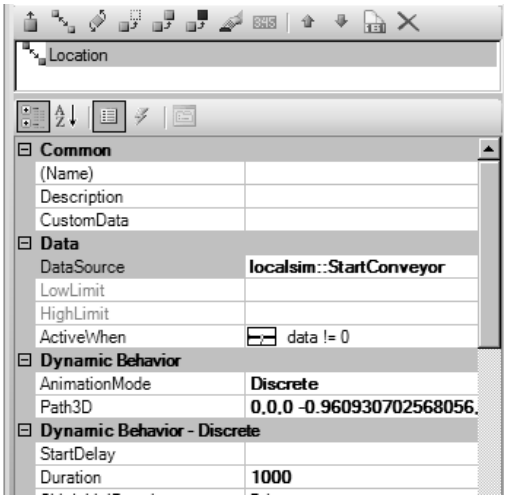

Figure L6.10 - Location Dynamics Properties

8. Select the Full Bottle that’s next to the Empty Bottle and duplicate it. This will be the bottle where we add dynamics to simulate the filling of it.

9. Remove the Location dynamic from this bottle and add a Hide dynamic to it.

10. Add the data source for a simulation tag (let's call it "StartFill") and set it to hide the bottle when the data source equals 0 (zero). Do not forget to make sure that the AnimationMode is Discrete.

11. Now drill down into the symbol and find the fill of the bottle (the actually liquid inside).

12. Add a Size dynamic to this fill. You should select the same data source you used for the Hide dynamic here. Set the AnimationMode to Discrete, set the ActiveWhen property to when data does not equal zero, and set the Duration to 2000 (milliseconds).

13. Now add the tank back into the display by clicking the empty icon box in the Explorer where the Tank's eye icon should be. Align the tank to above the first bottle that is filled with liquid.

14. Add a cylinder with a radius of 0.2 and a Height of 3 below the tank as shown in Figure L6.11. This will extend into the bottle.

Figure L6.11 - Tank with Extension Tube

15. Since we are making the tube extend to the bottle, let us put a Location dynamic on the tube. Again, the data source should be a simulation tag (let's call it "ExtendTube"). Set the AnimationMode to Discrete and the Duration to 1000 milliseconds. Also, set the SkipInitialDuration to True and the ReverseAnimationWhenFalse to True.

16. At this point, your display should be nearly complete. We will just add one more thing to it.

17. In 2D mode, add a control board to the display. The control board consists of a text object that reads "Control Board" and a rectangular back plate.

18. The Control board should also have four radio buttons called: Start Conveyor, Extend Tube, Fill Bottle, and Retract Tube.

19. Give each radio button a GroupName (let's call it "Production").

20. For the Start Conveyor radio button, set the Command to Write Value. The data source should be the one you set for the Location dynamics of all the bottles. If you used the suggested tag, it is the simulation StartConveyor tag. Set the radio button's OnCheckedValue to 1.

21. The Extend Tube and Retract Tube radio buttons are very similar. They should have the same data source, which is the one you use for the Location dynamics for the tube. If you used the suggested tag, it is the simulation ExtendTube tag. For the Extend Tube radio button, set the OnCheckValue to 1; for the Retract Tube tag, set the OnCheckValue to 0 (zero).

22. In the Retract Tube radio button, add two more Pick actions. One of the Pick actions should reset (to a value of 0) the conveyor moving tag (StartConveyor) and the other should reset the bottle filling tag (StartFill).

23. For the Fill Bottle radio button, set the data source to the one you used for the Size dynamic. If you used the suggested tag, it is the simulation StartFill tag.

24. Your display should now look similar to Figure L6.12.

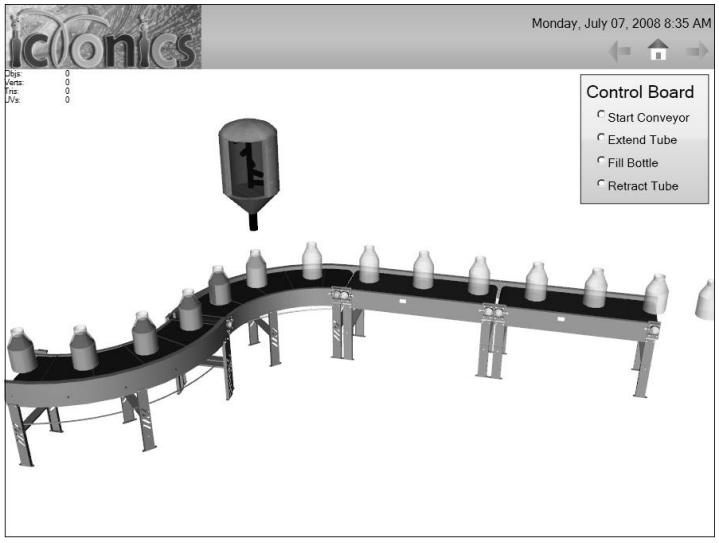

Figure L6.12 - Finished Display

25. Save the display if you have not saved it and go into runtime. Try clicking on each of the radio buttons in order and you should see your conveyor move, your bottle fill, and move away.

Glossary

2D stands for two-dimensional. In GraphWorX64, 2D is an environment in which you can create objects that have two dimensions: height and width.

3D stands for three-dimensional. In GraphWorX64, 3D is an environment in which you can create objects in three-dimensional spaces that have a height, width, and depth.

3D Coordinates identify a point in the 3D space plotted as (*X*, *Y*, *Z*), where *X, Y,* and *Z* are the locations on the X, Y, and Z axes.

Ambient Color is the color that surrounds an object.

Annotations are graphical or textual labels that remain in place when a display's object rotates or moves.

Axis is a line along which one of the X, Y, or Z coordinates orient an object in a *scene* in the 3D space. The three axes measure height, width, and depth in a *scene*.

BackMaterial Property is the property that identifies the color, texture, or pattern of the back side of a two-dimensional primitive or the inside of a three-dimensional primitive.

Camera View is the current *perspective* of a *scene*. A *scene* can be looked at from any perspective. The camera view is the currently viewed perspective.

Cap is a property that determines whether the primitive is closed off at the top or bottom (or at its ends).

Collision Detection is a test of dynamic (animated) and static objects in a 3D viewport which you believe could collide.

Cone Primitive is a funnel-shaped object used in 3D for creating a variety of graded shapes, including pyramid or umbrella shapes.

Cube Primitive is a box-shaped object used in 3D for creating shapes with rectangular sides.

Cylinder Primitive is a column-shaped object used in 3D for creating rounded wheels, straight tubes, drum shapes, or shapes with sides such as pentagons or even triangular shapes. To make bendable tube shapes, you would use the *Pipe Primitive* instead.

Diffuse Color is the color that an object surface reflects.

Donut Primitive is a ring-shaped object whose shape you can change by altering its properties. A donut primitive can be given rounded or straight sides.

Dynamics are animation actions that you can apply to a 3D object that dictates the behavior of the object in runtime. They include process and data entry points, pick actions, and selectors, and they allow you to change the location of the object or alter the color or visibility of the object in runtime.

Edges are the lines that connect the *vertices* of a shape's face.

Emissive Color makes an object appear to emit light of that color.

Extrude is extra dimension you can add to a *face* through a protrusion or elevation of the surface.

Face is the surface of a shape. It is made up of *vertices* and *edges* and the triangles that the vertices and edges create.

Grid is a spatial ruler that is not part of the scene but guides your placement of objects in a *scene*.

Hemisphere is a property of the *Sphere primitive*. It ranges from 0-1 (top to bottom), which allows you to create shapes such as domes and slices.

Key Object is the object that is the reference point when you are applying manipulations to multiple objects. When you are trying to manipulate a group of objects, the last object selected is the *key object*.

Local Space is similar to *Object Space*, but the origin of the space comes from the object's parent.

Material is the color or image applied to a surface of an object. The Material Gallery consists of shortcuts to preferred colors and images. Material can be applied to the front and back of a 2D object or the front and inside of a 3D object.

Material Property is the property that identifies the color, texture, or pattern of the front side of a two-dimensional primitive or the outside of a three-dimensional primitive.

Object Explorer is a feature of 3D viewport that lists objects and groups in the *scene*. It is frequently easier to select an object by clicking it in the Object Explorer rather than clicking on it in the scene itself. For that reason, using easy-to-understand names for objects is critical to using the Object Explorer effectively.

Object Space is the coordinate system from an object's point of view. The origin is the object's *pivot point* or center.

Orbit allows you to tumble or rotate around the *scene* to view from any perspective. To orbit, you can hold down the ALT key and the left mouse button, and then drag the mouse to navigate through the *scene*.

Pan allows you to scan and scroll through the width and depth of the *scene* in the *viewport*. To pan, you can hold down the ALT key and the middle mouse button or wheel, and then drag the mouse in any direction to pan the scene.

Perspective is a point of view from which you can view and work with in a *scene*. In terms of GraphWorX64, you can view a *scene* from a 3D perspective, or from an orthographic perspective. The orthographic perspectives available are front view, side view, and top view.

Pipe Primitive is a tool for adding pipes and tunnels to a 3D display. The pipe primitive is unique in that you can manipulate it to create turns and bends, and it can be used as the pathway for a 3D Location dynamic.

Pivot Point is the point around which manipulations such as scale, rotate, and move are performed. This is also the Center property in an object's properties explorer.

Plane is a two dimensional area along any two axes. The available planes are XY, XZ, and YZ.

Plane Primitive is a four-sided, flat-surfaced shape that has a front and a back.

Polygon Primitive is a shape that has a front and a back that is confined in a plane. It is like a *plane primitive* except that it can have any number of sides.

Primitives are basic geometric shapes you can use to assemble a simple but effective 3D object in GraphWorX64.

Radius is a property that determines the length of any line from the center of a primitive to its perimeter. Some primitives have two radii. For shapes like donuts and cylinders, the larger radius specifies the outside perimeter of the shape and the smaller radius specify the inner "hole". For a cone, the two radius properties identify the radii of the top and bottom of the cone.

Rotate is a 3D feature that lets you reorient an object in space around its pivot point.

Saved View is a *scene* that you have configured and saved so that you can return to it with ease later. In runtime, users can transition from view to view in a movie-like sequence. Views can also work like a screen saver that runs movie-like sequences when the computer is idle.

Scale is a 3D feature that lets you resize an object in relation to the location of the object's pivot point. If the object's pivot point is at its center, scaling does not change the object's basic shape. Offsetting the pivot point can scale one or two dimensions of the object, distorting its shape.

Scene is the entire visible content of a 3D viewport.

Segments are the units that make up a primitive's height, width, and sides. They are also called triangles. The number of segments can determine an object's shape or appearance. More

segments give you a smoother surface and require more memory, while fewer segments give you less detail but faster rendering.

Slice Angle is a property that allows you to create rounded shapes with convex and concave angles. For example, you can create a wedge shape.

Specular Color creates highlights on an object's surface.

Sphere Primitive is an orb-shaped primitive used in 3D for creating ball and egg shapes, as well as sided spherical object shapes.

Terrain Primitive is a highly-faceted 3D shape that looks like the terrain of hills or mountains. It is a two-sided shape that has a front and a back. You can use its ElevationBitmap property to import an elevation bitmap for generating the terrain's relief profile.

Texture affects how material wraps around an object. The texture surfaces available in 3D GraphWorX64 are plane, cylinder, sphere, or automatic.

Torus refers to the *Donut Primitive*.

Transform is a 3D graphics term for changing the location, orientation, and/or size of an object.

Translate is a 3D graphics term for moving an object in any direction.

Vertices are the three-dimensional points of a shape's surface. The number of vertices is determined by the number of *segments* used to create the surface.

View is a selectable object that encapsulates the entirety of a 3D *scene*. It is essentially a snapshot of a scene in the viewport. Creating a view for a scene gives you and runtime users quick and easy access to that scene at any time.

Viewport is the window into a 3D scene in a GraphWorX64 display. You can maintain the 3D portion of a display only through a viewport.

World Space is the coordinate system for the entire *scene* in the 3D *viewport*.

WPF stands for Windows Presentation Foundation. It is a graphical subsystem in the .Net Framework 3.0. Graphics and animations in GraphWorX64 use this system.

XAML stands for Extensible Application Markup Language, which is used by *WPF*. This is the native GraphWorX64 file format for saving 3D models.

Zoom allows you to move closer to or farther away from the objects in a *scene*. To zoom in or out, you can scroll the wheel on your mouse. Alternatively, you can hold down the ALT key and the right mouse button, and then drag the mouse to zoom in or out of the scene.

Answers

Questions / Chapters	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Chapter 1	D	False	True	B	True	False				
Chapter2	D	True	C	B	D	C	True	G	F	True
Chapter3	D	B	F	D, G	D	E	D	False	True	False
Chapter4	E	C	B	True	False	A	B	B	E	True
Chapter5	C	False	B	A	B	False	False	False	True	True

OPC Automation at your fingertips

100 Foxborough Blvd., Foxborough, MA 02035
508.543.8600 • FAX 508.543.1503 • BBS 508.543.3881
www.iconics.com • E-mail info@iconics.com

© 2011 ICONICS, Inc. All Rights Reserved.
Specifications subject to change without notice.