WITNESS 14 3D Development

Floating Point Changes

WITNESS 14 is built using the Visual Studio 2013 and the Simulation library project has been changed to use *fp:precise*. Older versions of Visual Studio (version 6.0 and before) used the *fp:fast* compiler option as the default for Floating-Point behaviour.

In the past, while upgrading from different compiler versions we have maintained the use of this *fp:fast* option. Since Microsoft Visual Studio 2005, the default option is no longer *fp:fast* but *fp:precise*.

WITNESS 14 now uses the *fp:precise* option for Floating-Point behaviour. This will enable better reproducibility for future compiler upgrades. More information can be found here: http://msdn.microsoft.com/en-us/library/e7s85ffb(v=vs.120).aspx

From the Release Notes for WITNESS 14

"WITNESS 14 is built using Microsoft Visual Studio 2013 to take advantage of new and enhanced features. This has allowed the use of a more precise option for floating point calculations. In comprehensive testing there are some very small differences in calculations, in the magnitude of 10-6 and smaller. The vast majority of models will be completely unaffected by this, but where calculations are to many decimal places and where logic has been constructed that takes advantage of tiny numerical differences then changes to the results of models may be observed. "

See also

Previous floating point changes	https://sites.google.com/site/lannerwitness/updates/visualstudiofloatingpointstuff
What Every Computer Scientist Should Know About Floating-Point Arithmetic	http://docs.oracle.com/cd/E19957-01/806-3568/ncg_goldberg.html

Targeting XP in VS2013

See: http://blogs.msdn.com/b/vcblog/archive/2012/10/08/windows-xp-targeting-with-c-in-visual-studio-2012.aspx for more details

3D Command Reference

The 3D command reference is a document detailing the data commands published by WITNESS for 3D visualization. These commands are serialized in XML and passed between WITNESS and the Virtalis rendering application (W3DMod.exe) using a TCP socket connection.

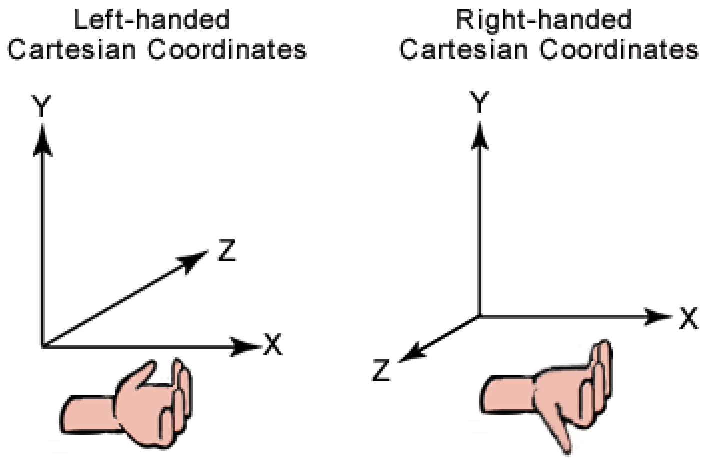
As of the time of writing, the latest version can be found here:

`"\\develop\Public\Products\witness\DevCycles\Witness14.0\Cycle2 - 3D Release\Docs\3D`

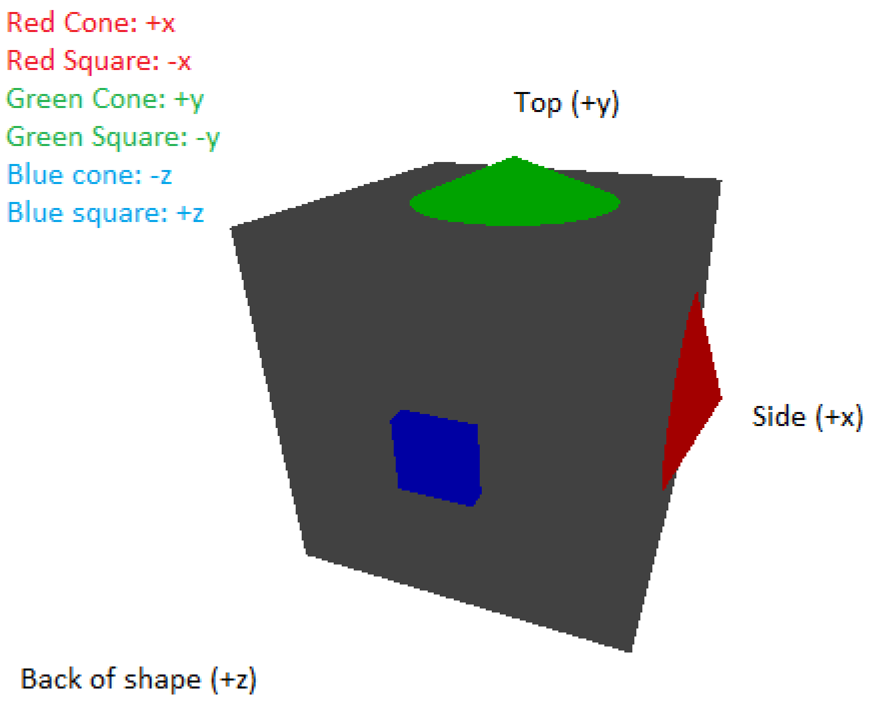
Command Reference 18.docx"

3D Coordinates and Roll Pitch and Yaw in WITNESS

3D graphics applications use two types of Cartesian coordinate systems: left-handed and right-handed. In both coordinate systems, the positive x-axis points to the right, and the positive y-axis points up. You can remember which direction the positive z-axis points by pointing the fingers of either your left or right hand in the positive x-direction and curling them into the positive y-direction. The direction your thumb points, either toward or away from you, is the direction that the positive z-axis points for that coordinate system. The following illustration shows these two coordinate systems.



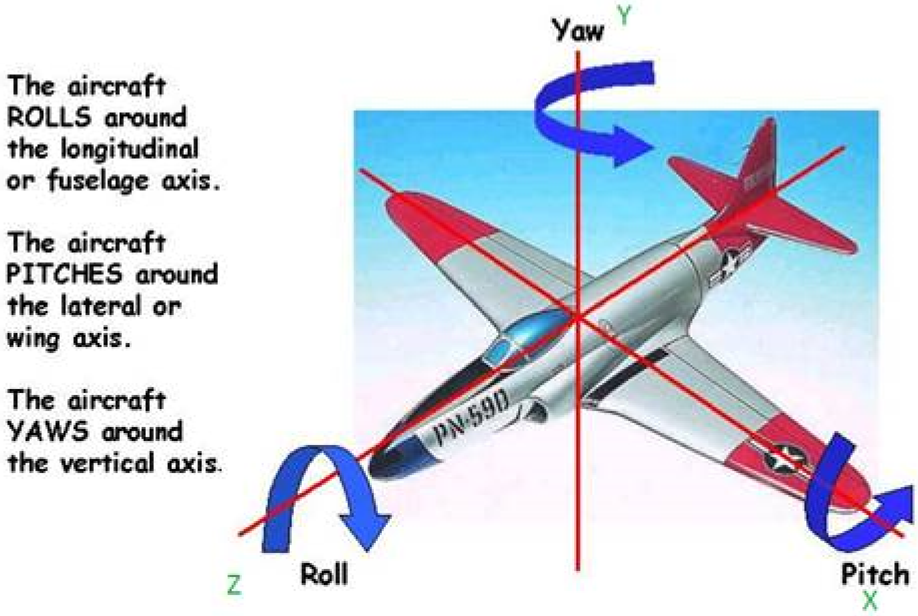
WITNESS/Virtalis 3D uses the Right Hand Rule to describe the 3D coordinate axis. Essentially this means that positive z is facing out from the screen. When positioning 3D shapes, for travelling along paths or positioning in queues, we consider the front face to be the face that is pointing along the negative z axis.



From the 2D, we map the 2D plan view xy plane to 3D xz plane. Y is vertical in 3D.

Roll - Is a roll is a rotation about the axis in and out of the screen (Z)
Pitch - Is a rotation about the horizontal axis (X)
Yaw - Is a rotation about the vertical axis (Y)

In summary:


The aircraft ROLLS around the longitudinal or fuselage axis.
The aircraft PITCHES around the lateral or wing axis.
The aircraft YAWS around the vertical axis.
Yaw
Y
PN-59D
Z
Roll
Pitch
X

Related item: 3D Queue position information and command send to the 3D Server see also Witness W3D Queue Implementation.docx

3D Queue Positioning Explained

See the follwoing two documents:

- Witness W3D Queue Implementation.docx
- Part Queuing Positions Overview 01.docx