Plant Simulation 3D Tutorial

1. User Interface
2. 3D Representation
3. Animations
4. Improving Performance
1. **User Interface**
   - The Ribbon User Interface
   - 3D Properties
   - Create In 3D
   - Graphic Structure
   - Write SimTalk Code More Easily

2. 3D Representation
3. Animations
4. Improving Performance
The Ribbon User Interface (1)

Contextual tab
Categories
Panels

The Ribbon User Interface (2)

Commands for 3D are located both on the general tabs ...

... and on the 3D contextual tabs.
3D Properties

Provides access to virtually all 3D properties of an object:
- Transformation
- Animation Paths
- Graphic Groups
- Extrusion Settings
- ...

Shortcut: Ctrl+3

Create In 3D

The dialog Create in 3D

- Controls whether an object may be created in 3D
- If CreateIn3D=false in the origin, inheritance of the setting is enforced!
- Don’t switch this off if you don’t need to. Normally it is better and more flexible to configure the visibilities settings instead.
- SimTalk: attribute CreateIn3D : boolean
- Accelerator: Alt H3P3C
Graphic Structure

The dialog Graphic Structure

- Corresponds to the 2D dialog Structure
- Displays the internal 3D structure of an object
- Provides different visualizations for
  - External graphic groups (red boxes)
  - Internal graphic groups (yellow boxes)
  - Invisible graphics (grayed out)
  - Animatable objects (lucid 3D icon)
  - State objects (traffic lights)
- Provides fast access to graphic inheritance and visibility settings
- Accelerator: Alt 3ER

Write SimTalk Code More Easily

Ctrl+C ... Ctrl+V

Ctrl+V
Plant Simulation 3D Tutorial

3D Representation

1. User Interface

2. 3D Representation
   • Graphic Groups
   • 3D Representation Settings
   • Work with Graphics and Graphic Groups in SimTalk
     • Example

3. Animations
4. Improving Performance
Graphic Groups

- The graphics of each and every object are partitioned into uniquely named graphic groups.
- There is always the graphic group named default.
- Graphic groups do not contain other graphic groups.
- A graphic group can be toggled visible or invisible.
- A graphic group can be marked as part of the internal or external representation of an object.

- The default graphic of some objects like conveyors, variables, etc. are generated by Plant Simulation.
- You cannot edit these generated graphic groups and they do not interact with graphic inheritance.

3D Representation Settings (1)

- The tab Visibility
  - Specifies which graphical content makes up an object's outside representation

- The tab Graphics
  - Determines graphic inheritance.
  - Lists all graphic groups and displays whether they are generated or not.
  - Sets if graphic groups are internal, external, visible, or invisible.

- Internal graphics
  - Are useful for decoration or organizational purposes.
  - Are only visible from the outside together with the contained objects.

- External graphics
  - Are visible from the outside as long as the graphic is visible.
  - Internally only temporarily visible for modelling
3D Representation Settings (2)

<table>
<thead>
<tr>
<th>Graphic group visibility from inside an object</th>
<th>Graphic group visibility from outside an object</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Marked as Visible?</strong></td>
<td><strong>Marked as Internal?</strong></td>
</tr>
<tr>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

*For modelling purposes, external graphic groups can be switched visible. This setting can be made for each window separately and it is not stored in the model.

Modify Graphics and Graphic Groups With SimTalk

- Plant Simulation provides a number of attributes and methods to work with 3D graphics and graphic groups.
- Compare the Plant Simulation help topic The Parameters GraphicPath and GraphicGroupName.
Example

The tutorial model shows the assembly of several engine parts to a complete engine ...
Plant Simulation 3D Tutorial

1. User Interface
2. 3D Representation

3. Animations
   - Animatable Objects
   - Create an Animation
   - Work with Animatable Objects and Animations in SimTalk
   - Animated Robots

4. Improving Performance

Animatable Objects (1)

Create a simulation object, e.g. a SingleProc, using an existing graphic (only in frames)

Create an animatable object using an existing graphic

Accelerator: Alt 3ECV

Accelerator: Alt 3ECO
**Animatable Objects (2)**

Creating an Animatable Object

- Split off a graphic part as a new animatable object.
- Define the object position, e.g. as rotation center for animations.
- Each animatable object can be animated or transformed.
- Each animatable object provides its own set of graphic groups.

Create an Animation (1)

Making an Object Animate the MUs it Carries

- You can modify (using the buttons Visualize, Extend, Edit..., Delete) non-generated or add (using the button Add...) additional animation paths.
- Each object has rules determining which animation paths it uses.
- To use another animation path on (the object) \textit{obj}, you can redirect an animation to it with \textit{obj.MUAnimations.Path.redirectTo(...)}. You can override this redirection with another call or cancell it with \textit{obj.MUAnimations.Path.cancelRedirection()}.
- To pass the control over the MUs of \textit{obj} to one of its animatable Objects, assign the relative path to this object to the animation object attribute (SimTalk: \_3D.AnimationObject). You can also set this value on the context menu of the target animation object.
Create an Animation (2)

Making an Object Animate Itself

- You can modify self animation paths and their contents the same way you do with MU animations.
- Self animations have to be started, paused or stopped using SimTalk.
- MUs have no self animation. If you want to add an animation to an MU that is controlled by itself and not by its location, you can separate the whole MU graphic into an animatable object and animate that object instead.

Create an Animation (3)

Demonstrate a simulation with camera flights

- Frames offer you the possibility to animate a camera. This feature can be used to create a demonstration of your simulation.
- You can modify camera animation paths and their contents the same way you do with MU and self animations.
- Camera animations have to be started, paused or stopped using SimTalk.
Work with Animatable Objects and Animations in SimTalk

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>obj.getObject(index: integer)</td>
<td>Path extension to the animatable object in obj at index.</td>
</tr>
<tr>
<td>obj.getObject(name: string)</td>
<td>Path extension to the animatable object called name in obj.</td>
</tr>
<tr>
<td>obj.MUAnimations</td>
<td>Path extension to obj's MU animations</td>
</tr>
<tr>
<td>obj.SelfAnimations</td>
<td>Path extension to obj's self animations (not for all object types)</td>
</tr>
<tr>
<td>obj.CameraAnimations</td>
<td>Path extension to obj's camera animations (only for Frames)</td>
</tr>
<tr>
<td>obj.[Self</td>
<td>Camera]Animations.Animation.schedule</td>
</tr>
<tr>
<td>obj.[Self</td>
<td>Camera]Animations.scheduleRotation(...)</td>
</tr>
<tr>
<td>obj.[Self</td>
<td>Camera]Animations.pauseAnimation</td>
</tr>
</tbody>
</table>

Animated Robots

For animated robots, we already have fully prepared augmented graphics:

Simply exchange the graphic of your PickAndPlace robot.

Accelerator: Alt 3EX
Improving Performance

1. User Interface
2. 3D Representation
3. Animations
4. Improving Performance
   • General Advice
   • Why Cutting Graphic Inheritance Often is a Bad Idea
   • Retrospective: Why Did We Build Our Engine MUs The Way We Did?
   • Building a Configurable Object with Animatable Objects
   • Graphic Optimization
   • Generated Graphics
Avoid cutting the graphic inheritance – especially for MUs!

- This happens if
  - You explicitly switch off the graphic inheritance
  - You change the structure of an object graphic with:
    - Group/Ungroup
    - Optimize
  - You add or remove graphics
  - You change any part of an object graphic:
    - *The material of that cube in the corner looks better if it's yellow.*
    - *Move the drill to the side a bit.*
    - *That wall should be a bit higher.*
  - ...

Why Cutting Graphic Inheritance Often is a Bad Idea (1)

Suppose we have a gray object with an orange and a green version ...

Subtyping with graphic modifications
(graphic inheritance will be cut!)

Subtyping with predefined graphic groups

Total size about 13 MB
Total size about 3 MB
**Why Cutting Graphic Inheritance Often is a Bad Idea (2)**

- Inherited graphics take up less disk space - but disk space is not the issue …

<table>
<thead>
<tr>
<th>Do it for Your Simulation</th>
<th>Do it for Yourself</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depending on the size of the graphic, cutting graphic inheritance itself may take time:</td>
<td>Your own effort in managing a number of slightly different graphics throughout the model grows, too.</td>
</tr>
<tr>
<td>◦ Plant Simulation has to duplicate the entire object graphic, there is no partial inheritance!</td>
<td>In general it is easier to search one special folder with your classes instead of searching an entire model.</td>
</tr>
<tr>
<td>◦ Many graphic duplicates make it more difficult for the graphics card to optimize render tasks.</td>
<td></td>
</tr>
</tbody>
</table>

**Retrospective: Why Did We Build Our Engine MUs The Way We Did?**

Instead of toggling graphic groups visible at the assembly stations, we could have modified the graphics of the *Engines*. But …

- Every process step would have changed the **entire** graphic – not just that part we knowingly modified.
  - The simulation certainly would have slowed down a lot.

- Every single *Engine* would have noninherited graphics as soon as either the Engine leaves its source or at the latest, when it passed the first assembly station
  - The memory consumption would have become **huge** – even for a simulation as small as this.
Building a Configurable Object with Animatable Objects

Possible Reasons
- Specifying every possible variant is either impossible or not feasible.
- A large number of variants can be explained as all of the combinations of several simpler variations
- You might also want to animate some sub-graphics.

How?
1. Select the sub-graphic that is to be configurable.
2. Split that graphic off as an animatable object.
3. Add all necessary graphic variants to that object and set up your desired default transformation.

Graphic Optimization

The Dialog Optimize Graphic
- Problem: You import a large .jt file and everything starts to slow down and performance becomes sluggish.
- Possible solution: Optimize your graphic.
- If you want to split off animatable objects, subdivide the object first, then optimize the parts.
- Prefer to optimize class graphics, not object graphics!
- Compare the graphic complexity before and after an optimization
- SimTalk: _3D.optimizeGraphic(...)
- Accelerator: Alt 3EO
To improve render performance, avoid/reduce:

- Large numbers of sub-graphics, e.g.
  - Roller conveyors
  - Rail conveyors with crossbars
  - Conveyors with a large number of legs
  - Chute conveyors with a large number of chute rails
- Curve segments

Thank you for your attention!

Dipl.-Inf. Peter Komarek
Siemens PLM Software