Edit Scene Component Example

**Note:**

This document shows you how to edit scene component by some examples. For complete configuration, see document “Configuration”.

In this document, we assume that configuration data is place at the webapps\ROOT\data directory of TOMCAT installation directory. When we discuss any files or directories, we assume their directory starting from that file directory.

All configuration files are text files, in which any content between /\* and \*/ or following // are comments. The comments have no meaning, and can be deleted from configuration files; their function is only for improving readability.

If an item in configuration file is a file name, its directory is that of the configuration file.

## Preparation

Read document “Front-end Development Example” and know how to do front-end development.

Put the following code in a HTML file. Don't forget Modifying the web server URL.

<html>

<head>

<script

type="text/javascript"

src="http://localhost:8080/interface.jsp?function\_name=construct\_render\_object">

</script>

<script type="text/javascript">

var render\_object;

function body\_onload()

{

construct\_render\_object(

document.getElementById("my\_canvas"),

"NoName","NoPassword",

"chinese",

"test","",

"","",5,[],

function(my\_render\_object)

{

render\_object=my\_render\_object;

document.title=render\_object.title;

}

);

}

</script>

</head>

<body onload="body\_onload();">

<canvas id="my\_canvas" tabindex="0" width=1250 height=500></canvas>

</body>

</html>

## A Scene Assembly File and A Component Assembly File

Enter into directory “data\assemble\test\_part\assemble\part”, You will see three files:

* assemble.txt
* movement.assemble
* SLDASM.assemble

The file assemble.txt is Scene Assembly File, In it you will see:

no ..\camera.assemble

no movement.assemble

yes SLDASM.assemble

The content in Scene Assembly File is some records, each record has two items. The first item is a Boolean value (yes or no, true or false). When engine create a component list, the first item identifies whether or not components are shown in component list. The second item is Component Assembly File, whose content defines components that exist in scene, and where the components are located.

File SLDASM.assemble is Component Assembly File. The content in a Component Assembly File is a component tree. The first item in a tree node is component name, whose function is to identify a component. The second item in a node is part name; it identifies which part data this component exploits to render it. The third item in a node is component location. It is a 4\*4 matrix (16 floating number). The component location identifies relative location to its parent node.Following third item is 0 or more parameter. Part driver decide how many parameters exist here. Some part drivers need no parameter, while others may need many parameter. The last item is child number. If this node is a leaf, this item is zero. If this item bigger than zero, it means this node is a sub tree and following child number is the definition of the sub tree.

The following code the content in file SLDASM.assemble (Note: all content between /\* and \*/ are comment and has no meaning. When engine read data from files, they will ignore all contents between /\* and \*/ ):

/\* 1:name \*/ display\_component\_root

/\* 1:type \*/ display\_part\_root

/\* 1:location \*/ 1.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 1.0

/\* 1:child\_number \*/ 5

/\* 2:name \*/ display\_component

/\* 2:type \*/ display\_part

/\* 2:location \*/ 1.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 1.0

/\* 2:child\_number \*/ 0

/\* 2:name \*/ display\_component\_1

/\* 2:type \*/ display\_part\_1

/\* 2:location \*/ 1.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 1.0

/\* 2:child\_number \*/ 0

/\* 2:name \*/ display\_component\_2

/\* 2:type \*/ display\_part\_2

/\* 2:location \*/ 1.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 1.0 0.0 1.0 0.0 0.0 1.0

/\* 2:child\_number \*/ 0

/\* 2:name \*/ display\_component\_3

/\* 2:type \*/ display\_part\_3

/\* 2:location \*/ 1.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 1.0

/\* 2:child\_number \*/ 0

/\* 2:name \*/ display\_component\_4

/\* 2:type \*/ display\_part\_4

/\* 2:location \*/ 1.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 1.0

/\* 2:child\_number \*/ 0

This Component Assembly File has six components, one is parent, the others are its five children. Each component has a component name, part name, location, and child number. If child number is zero, it is leaf component, otherwise it is non-leaf component.

The location has 16 floating numbers, it is a 4\*4 matrix. It indicates relative location from its parent, its absolute location is multiplication of all matrixes of all its forefathers and itself.

In this example, all matrix are identity matrix like this .

1.0 0.0 0.0 0.0

0.0 1.0 0.0 0.0

0.0 0.0 1.0 0.0

0.0 0.0 0.0 1.0

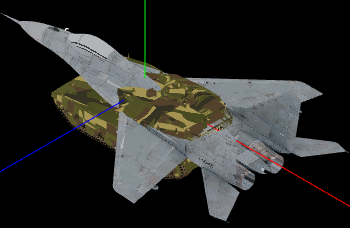
## Modify Component Part

Modify component part name of display\_component to names in following table, browser HTML file by chrome, firefox,opera, .etc. Images in following table will be displayed.

|  |
| --- |
| obj\_format\_file\_military\_part\_0 |
| obj\_format\_file\_military\_part\_1 |
| obj\_format\_file\_military\_part\_3 |
| obj\_format\_file\_military\_part\_4 |
| bridge\混色杆件\结点板\_编号\_22甲.SLDPRT |
| bridge\混色杆件\结点板\_编号\_218\_原编号\_18.SLDPRT |
| bridge\混色杆件\加强支杆\_编号\_251\_原编号\_101.SLDPRT |
| bridge\混色杆件\畸零短弦杆\_编号\_25.SLDPRT |

## Modify Component location

Set part name of display\_component to obj\_format\_file\_military\_part\_1, set part name of display\_component\_1 to obj\_format\_file\_military\_part\_4, two components are located at same location . Browser HTML file，you will see following image.



Component location is decided by a 4\*4 matrix with 16 floating numbers. To increase height of plane, you can increase height of display\_component(plane) to 1000 by the following matrix.

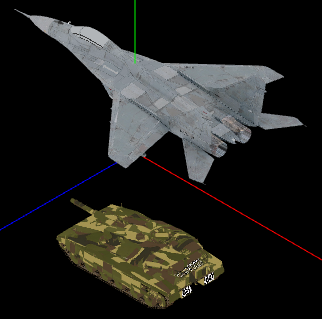
1.0 0.0 0.0 0.0

0.0 1.0 0.0 0.0

0.0 0.0 1.0 0.0

0.0 1000.0 0.0 1.0

Browser HTML file，you will see following image.



To rotate 90 degree around X aixs, you can employ the following matrix.

1.0 0.0 0.0 0.0

0.0 0.0 /\* cos \*/ 1.0 /\* sin \*/ 0.0

0.0 -1.0 /\* -sin \*/ 0.0 /\* cos \*/ 0.0

0.0 1000.0 0.0 1.0

Browser HTML file，you will see following image.



To rotate 90 degree around Y aixs, you can employ the following matrix.

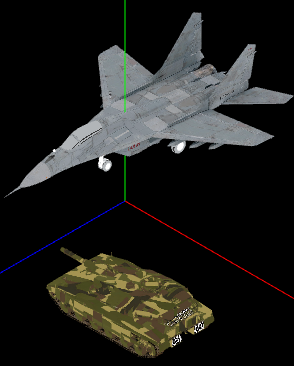
0.0 /\* cos \*/ 0.0 -1.0 /\* -sin \*/ 0.0

0.0 1.0 0.0 0.0

1.0 /\* sin \*/ 0.0 0.0 /\* cos \*/ 0.0

0.0 1000.0 0.0 1.0

Browser HTML file，you will see following image.



To rotate 90 degree around z aixs, you can employ the following matrix.

0. 0 /\* cos \*/ 1. .0 /\* sin \*/ 0.0 0.0

-1. .0 /\* -sin \*/ 0. 0 /\* cos \*/ 0.0 0.0

0.0 0.0 1.0 0.0

0.0 1000.0 0.0 1.0

Browser HTML file，you will see following image.

