

inVRs - Feature #48

Normalized Entities

05/23/2011 01:07 PM - sam g

Status: New	Start date: 05/23/2011
Priority: Normal	Due date:
Assignee:	% Done: 0%
Category:	Estimated time: 0.00 hour
Target version:	
Resolution:	
Description	
<p>it's quite common that you have models, which have a different scale. Therefore you can set the scale option within entities.xml. However this is an annoying task, if you have to start from scratch.</p> <p>Therefore the idea to add an option or pseudo loader for models, which ensure that the model will get normalized (boundingsphere center: 0,0,0, r: 1) after the model has loaded.</p> <p>This would help to simplify the configuration process, because you only have to take care about the relations between the objects, not the individual sizes of the objects</p>	

History

#1 - 05/23/2011 01:36 PM - Christoph Anthes

A new class of entities is not really required to solve this issue, or better to implement this feature. In general I would integrate such a feature during loading entities as an additional entity attribute. Two approaches could be implemented:

- 1) Changing the scale of the geometry representing the entity. This would involve scene graph specific adaptation of the actual entity geometry during the loading process. I would not go for that option since it is scenegraph specific and would have to be implemented for each underlying scenegraph.
- 2) Changing the scale of the entity type. The scale attribute of the type could be easily adapted during loading. The loaded entity type could be set to a normalized scale.

In the entity types configuration as a subtag of entityType that could be configured via XML as:

```
<normalizeTransformation x="true|false" y="true|false" z="true|false">
```

It has to be watched that the scaling is performed typically on one axis, since few entities are really cubic. So the scaling axis has to be provided.

#2 - 05/25/2011 03:45 PM - sam g

Christoph Anthes wrote:

A new class of entities is not really required to solve this issue, or better to implement this feature.

was never my idea to implement it like this.

In general I would integrate such a feature during loading entities as an additional entity attribute. Two approaches could be implemented:

1) Changing the scale of the geometry representing the entity. This would involve scene graph specific adaptation of the a actual entity geometry during the loading process. I would not go for that option since it is scenegraph specific and would have to be implemented for each underlying scenegraph.

but it would be more transparent to inVRs. e.g. by using a Pseudo-Loader like OpenSceneGraph (.scale, .translate,...) and in Addition, I don't know how different entiyTypes (Physics) will react on such changes.

2) Changing the scale of the entity type. The scale attribute of the type could be easily adapted during loading. The loaded entity type could be set to a normalized scale.

In the entity types configuration as a subtag of entityType that could be configured via XML as:

```
<normalizeTransformation x="true|false" y="true|false" z="true|false">
```

It has to be watched that the scaling is performed typically on one axis, since few entities are really cubic. So the scaling axis has to be provided.

I don't understand what you mean by only one axis. The original idea was to ensure that the Boundingsphere of the whole model has a radius of 1 and it centered at the origin. Therefore it will automatically adjust to the largest axis value.

You further have to keep in mind, that with your suggestion, it is possible to have multiple or even none true parameter settings (x=true, y=true, z=true). If you want to go for that, you have to specify the exact algorithm and order. Besides that you have to clarify, which is the order, if you also want to change the representation settings. which comes first,....

Furthermore, which is not described, you possible have to translate the model, too. e.g. if the model is not at the center.