

Solipsis

Building Guide

1. System configuration

Minimal system configuration

- 1Gb of physical memory
- 250 Mb of free disk space
- Windows® XP or Vista
- DirectX® 9.0 or OpenGL® 2 compatible graphic card with 128Mb memory
- Internet connection

Recommended system configuration

- 2Gb of physical memory
- 250 Mb of free disk space
- Windows® XP or Vista
- DirectX® 9.0 or OpenGL® 2 compatible graphic card with 512Mb memory
- Internet connection

2. *Links*

lexiExporter plugin

<http://www.ogre3d.org/wiki/index.php/LEXIExporter>

oFusion exporter plugin

<http://www.ofusiontechnologies.com>

3. General requirements

A. Softwares/Plugins

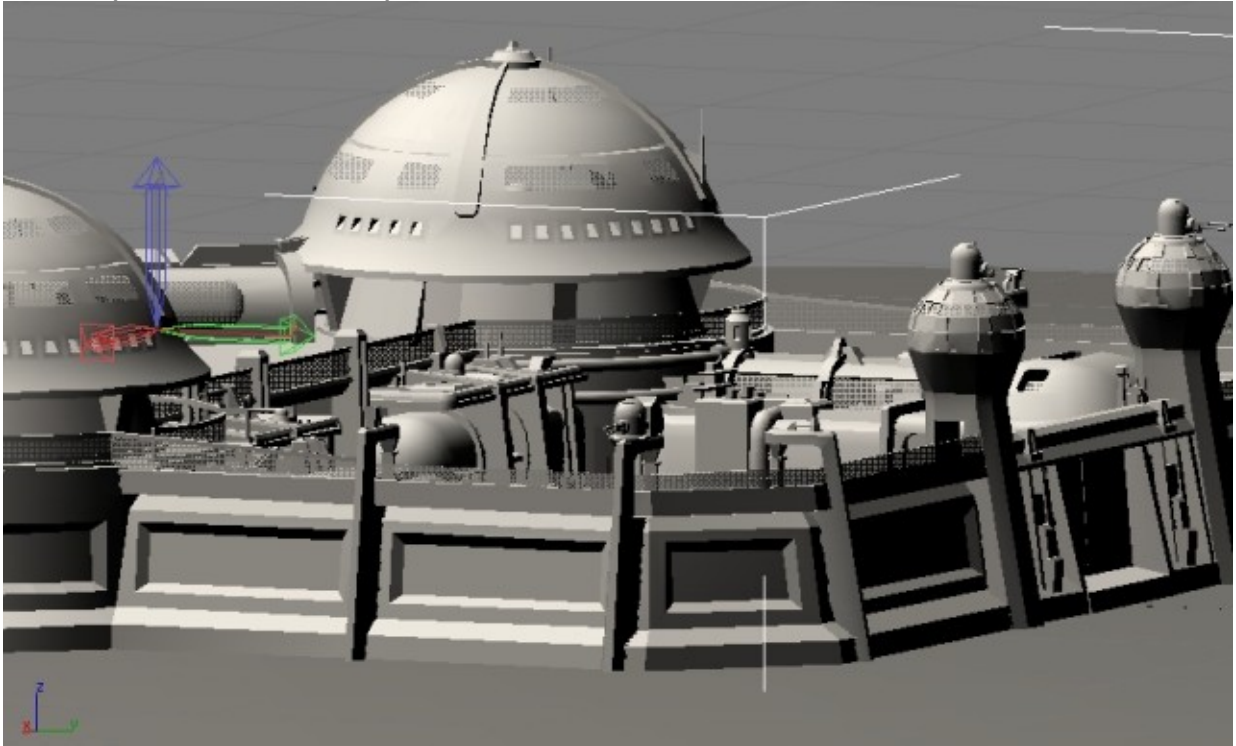
- Your favorite 3D modeling software (3DS Max®, Maya®, Blender®)
- Your favorite Ogre exporter plugin (LexiExporter for 3DS Max®, oFusion for 3DS Max®).

For instance lexiExporter is exporting correctly avatars, oFusion has some troubles with bounding box computed/exported into the .mesh file.

B. Axis/Scale

Models must respect the standard meter unit (scale).
(0, 0, 1) is 1 meter vector on the UP-axis (Z in 3DSMax)

In **3DS Max**, (X_{max} , Y_{max} , Z_{max}) with UP axis = Z axis



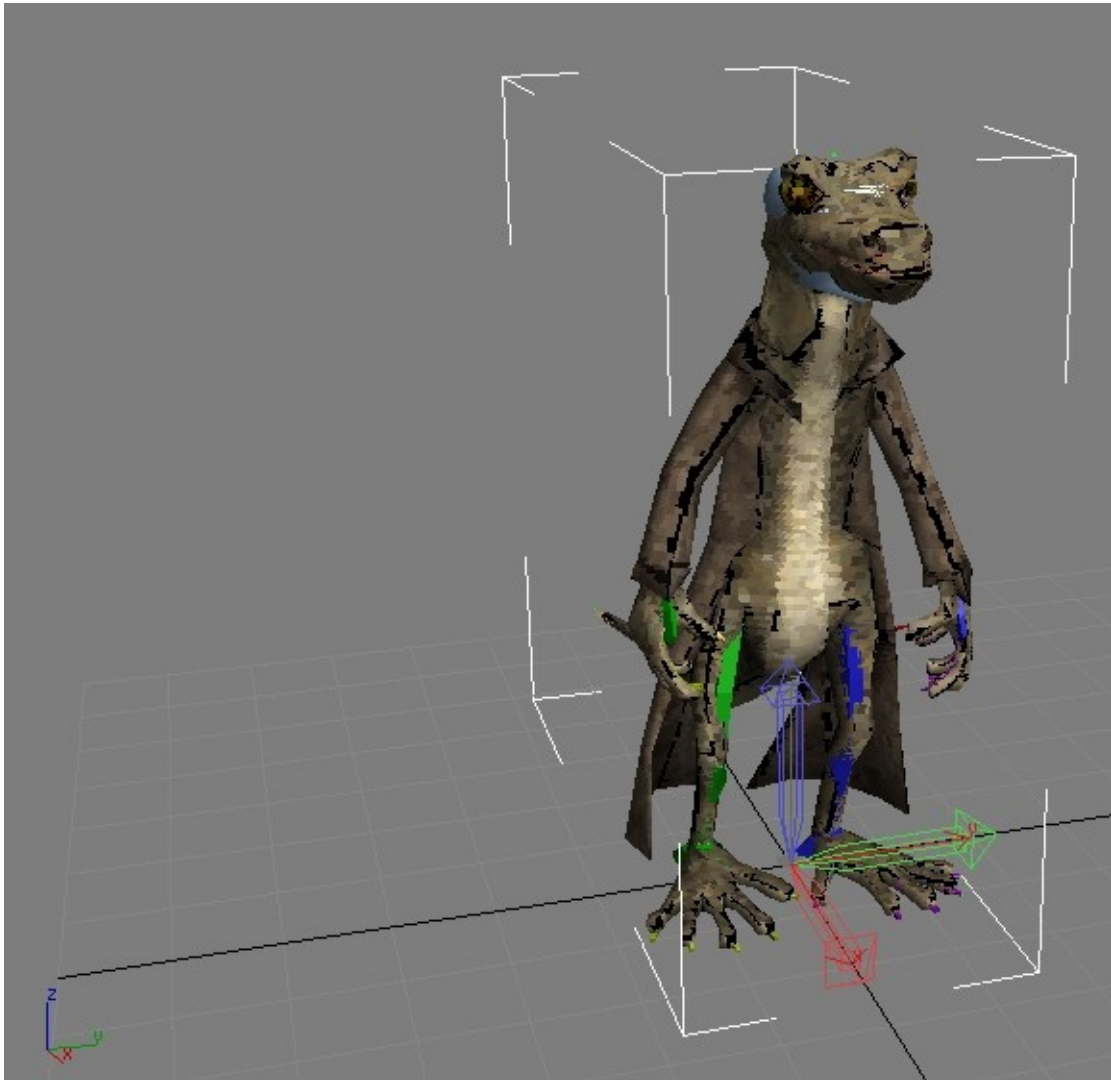
In **Solipsis (Ogre engine)**, (X , Y , Z) = (X_{max} , Z_{max} , $-Y_{max}$) with UP axis = Y axis



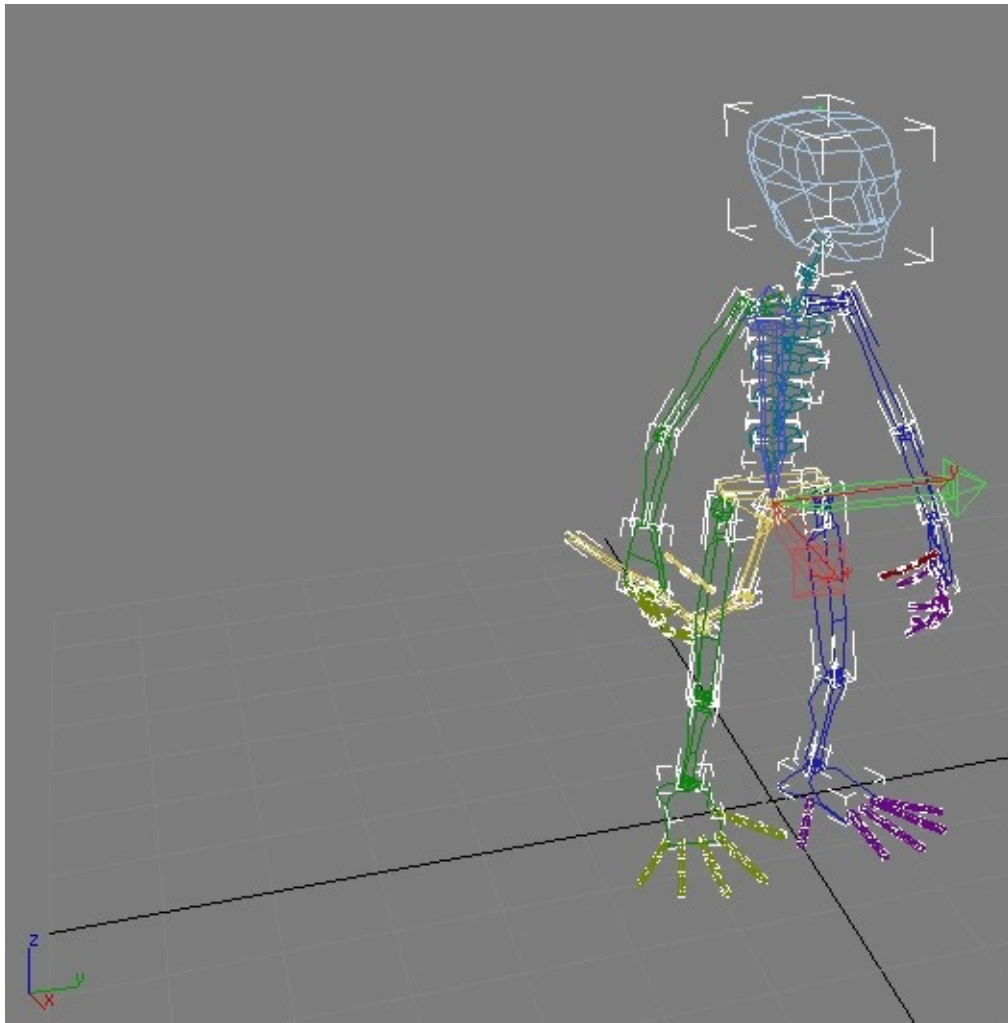
4. Avatar building

A. Requirements

- Avatar must respect the standard meter unit (scale), basic avatars are about 1.5 m.
- Avatar is facing the X axis ie. Eye vector= $(1, 0, 0)$
- Pivot on mesh must be centered and just below feet

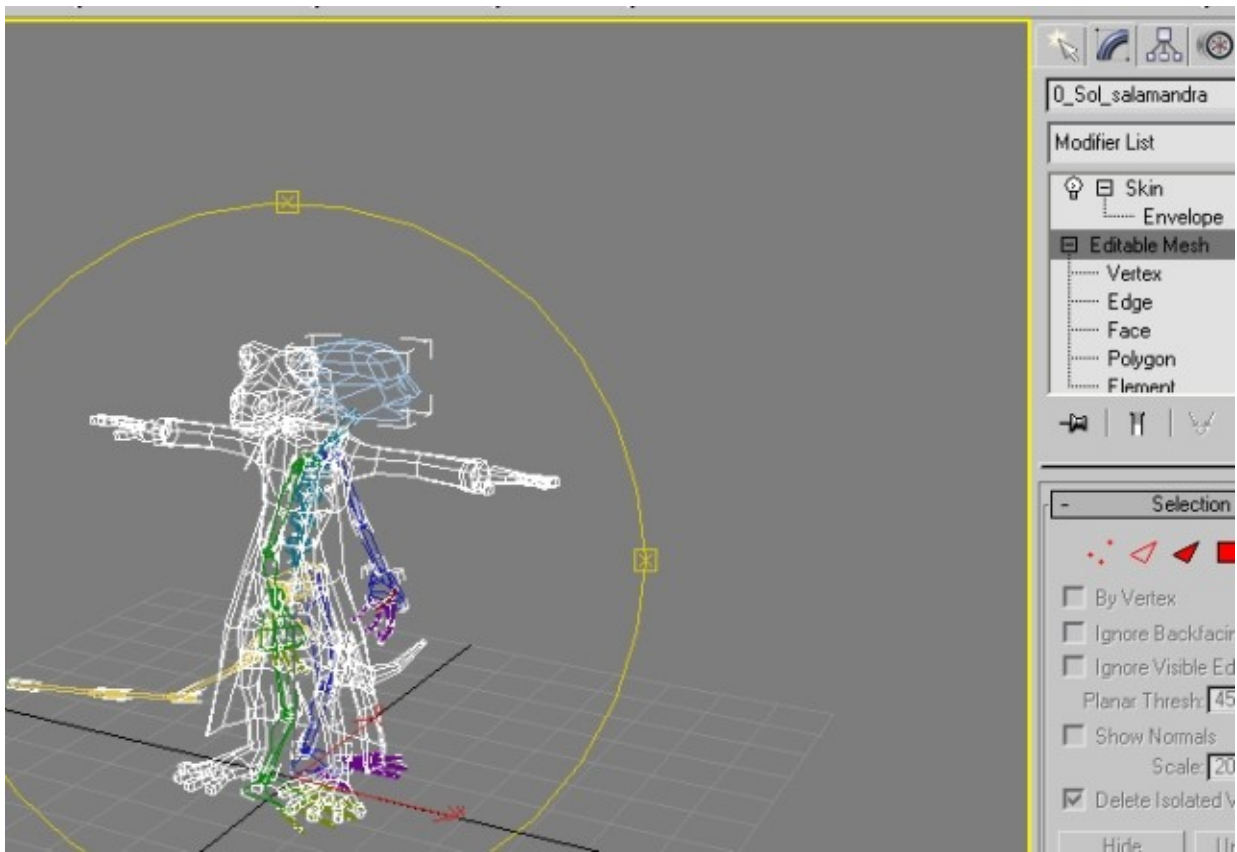


- Pivot on skeleton must be centered on the hip joint



- All animations are embedded into the same .max file, each animation will be described into the exporter plugin dialogs. Animations are : Idle, Walk, Run, Fly, Swim

oFusion users note : Take care about initial pose of skinned mesh according to the rigged skin because oFusion take this pose into account on bounding box computation



B. Step by step

1. Create your avatar

mesh, materials, textures, skinning, rigging, animations ...

2. Export your avatar (eg. with lexiExporter in 3DS Max) into 1 dedicated directory

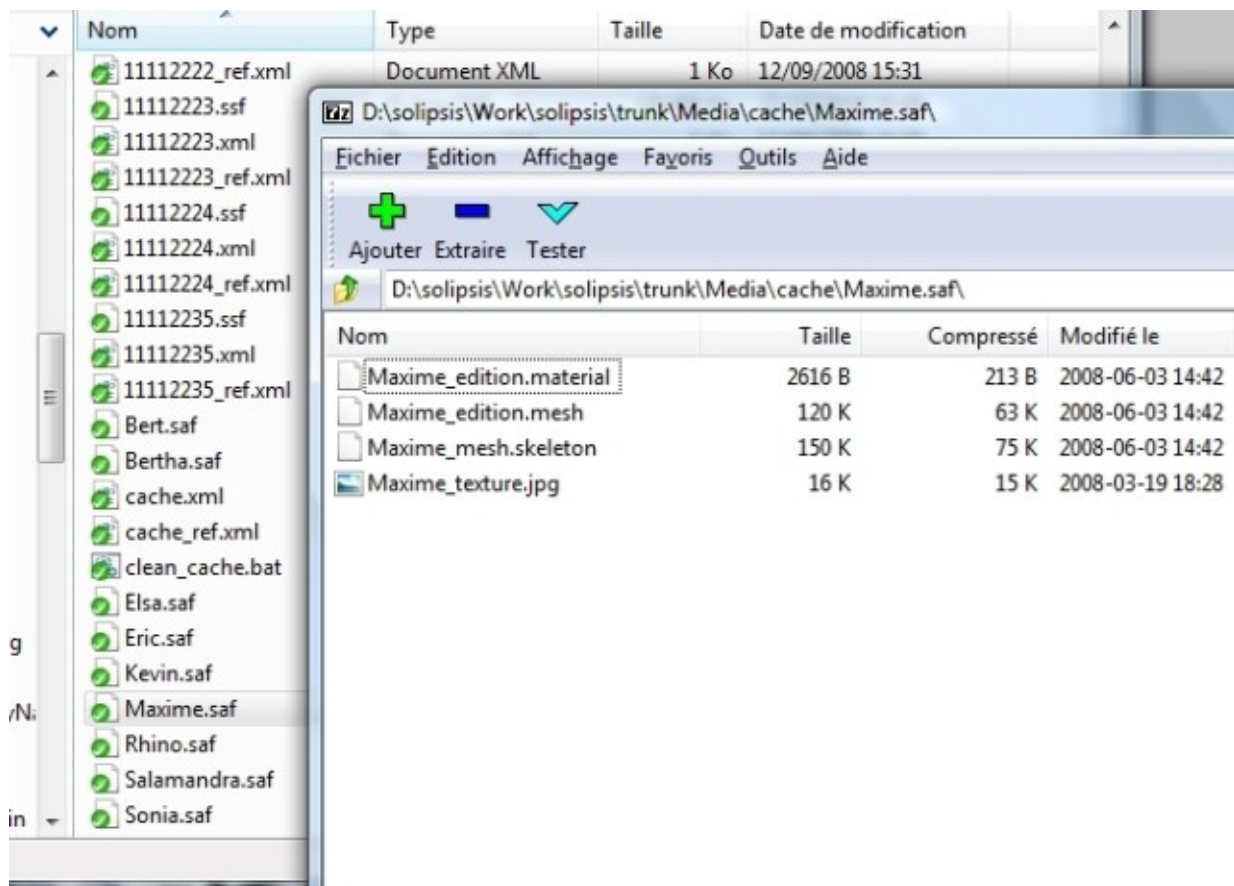
You will have all associated files (mesh, skeleton, materials, textures)

3. Build your Solipsis Avatar File (.saf)

The SAF file is a simple ZIP file renamed in SAF and contains all exported files (<myavatar> stands for the filename of your avatar) :

- <myavatar>_edition.mesh : the OGRE mesh file
- <myavatar>_mesh.skeleton : the OGRE bones/animations file
- <myavatar>_edition.material : the OGRE material file
- .bmp, .jpg, ... : textures files

Example Maxime.saf avatar file contains :



C. Integration

Simply copy your SAF avatar file into your local Solipsis Navigator directory, commonly into C:\Program Files\Solipsis\Media\cache. Restart your navigator and you will see your avatar into the Avatar editor list.

5. Scene building

A. Requirements

- Scene must respect the standard meter unit (scale)

B. Step by step

1. Create your scene

meshs, materials, textures ...

2. Create the collision mesh of your scene

floor, walls, stairs, ...

Keep it as simple as possible, avoid too much tessellation

Give it a specific name (COLLISION_mesh for example)

Choose the position where new incoming avatars will be dropped (the entry gate)

3. Export your scene (eg. with oFusion in 3DS Max) into 1 dedicated directory

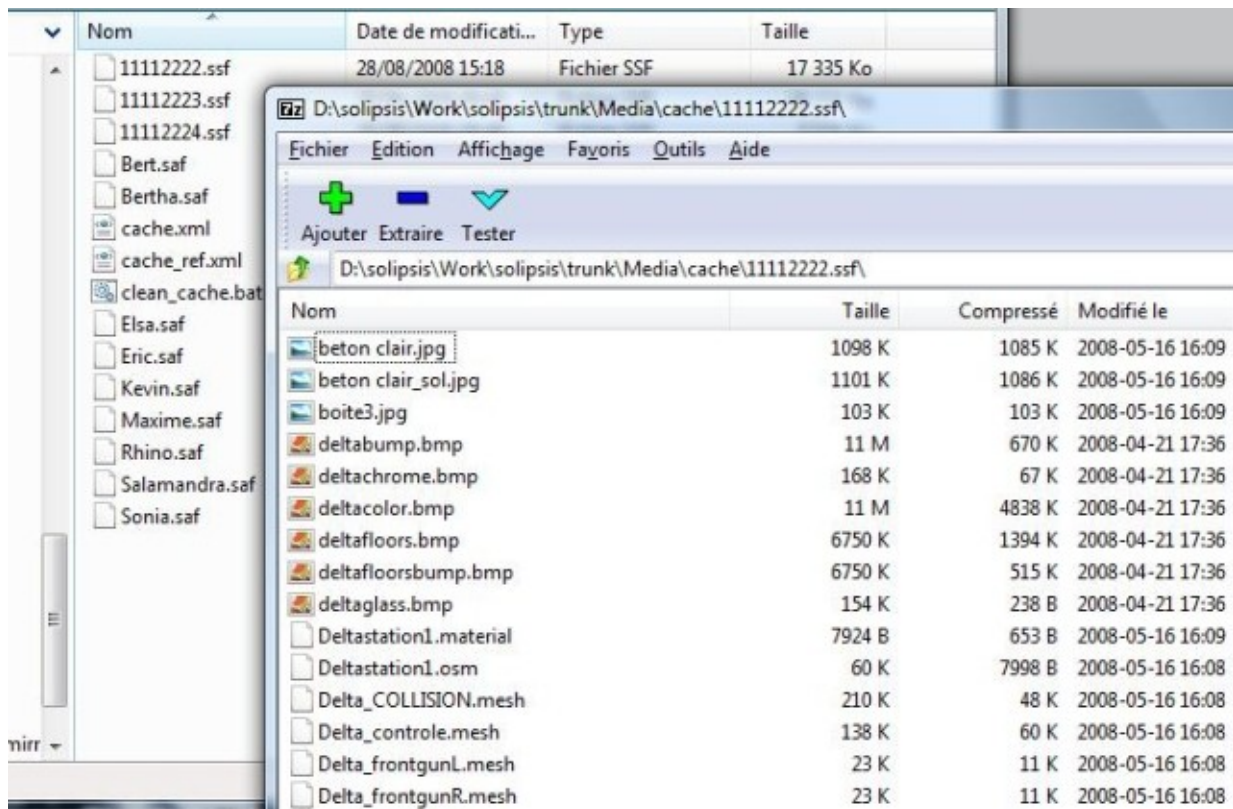
You will have a .OSM file with all associated files (meshs, materials, textures)

4. Build your Solipsis Scene File (.ssf)

The SSF file is a simple ZIP file renamed in SSF and contains all exported files :

- .osm : the oFusion scene file
- .mesh : OGRE mesh files
- .material : the OGRE material file
- .bmp, .jpg, ... : textures files

Example 11112222.ssf scene file contains :



C. Integration

1. Create the XML scene description file

The final step is the creation of the XML file describing your scene.
In this example the XML file 11112222.xml looks like :

```
<node>
  <entity uid="11112222" owner="00000001" type="1" name="Deltastation1" version="00000000">
    <position x="18" y="-58" z="133" />
    <orientation x="0" y="0" z="0" w="1" />
    <aabb>
      <min x="0" y="0" z="0" />
      <max x="0" y="0" z="0" />
    </aabb>
    <content>
      <sceneContent>
        <entryGate gravity="true">
          <position x="17" y="-50" z="115" />
        </entryGate>
      </sceneContent>
      <lod level="0">
        <sceneLodContent mainFilename="Deltastation1.osm" collision="Delta_COLLISION" />
        <files>
          <file name="11112222.ssf" version="00000000" />
        </files>
      </lod>
    </content>
  </entity>
</node>
```

11112222 is the unique node identifier (*nodeId*) and 11112222 is the unique entity identifier (*entityUID*).

Owner is the node identifier of the creator (avatar).

Adjust if necessary **position** and **orientation** of the scene.

Define the **entryGate** attribute : position of new incoming avatars and the flag indicating if the gravity is activated.

Define the **mainFilename** attribute with the OSM file present in the SSF archive.

Define the **collision** attribute with the name of the collision mesh present in the SSF archive.

Finally define the SSF archive file into the **files** tag.

2. Server administrator adds the SSF file

The server administrator can now integrate the scene into the dedicated RakNet server. He must copy the SSF file and the <nodeId>.xml file into the server Media/cache directory of the server.

To simplify clean up of cache, copy the <nodeId>.xml file into <nodeId>_ref.xml file and update the clean_cache.bat.

Finally launch the RakNet server with argument -s <nodeId> and the dedicated port -p <port>. To define it into the WorldsServer, edit the file WorldsServer\uiworlds.html and insert your new server like :

eg. uiworlds.html for RakNetServer.exe -p 8556

```
...
selectWorlds.innerHTML += "<option value='127.0.0.1:8556'>MyScene</option>";
...
```