Best viewed at 1024x768
or greater

If you use a wide screen
monitor, this document will
not be correctly displayed.
Please refer to the
offline pdf copy
(Meta2tr_Guide_Book.pdf)

CONTENTS

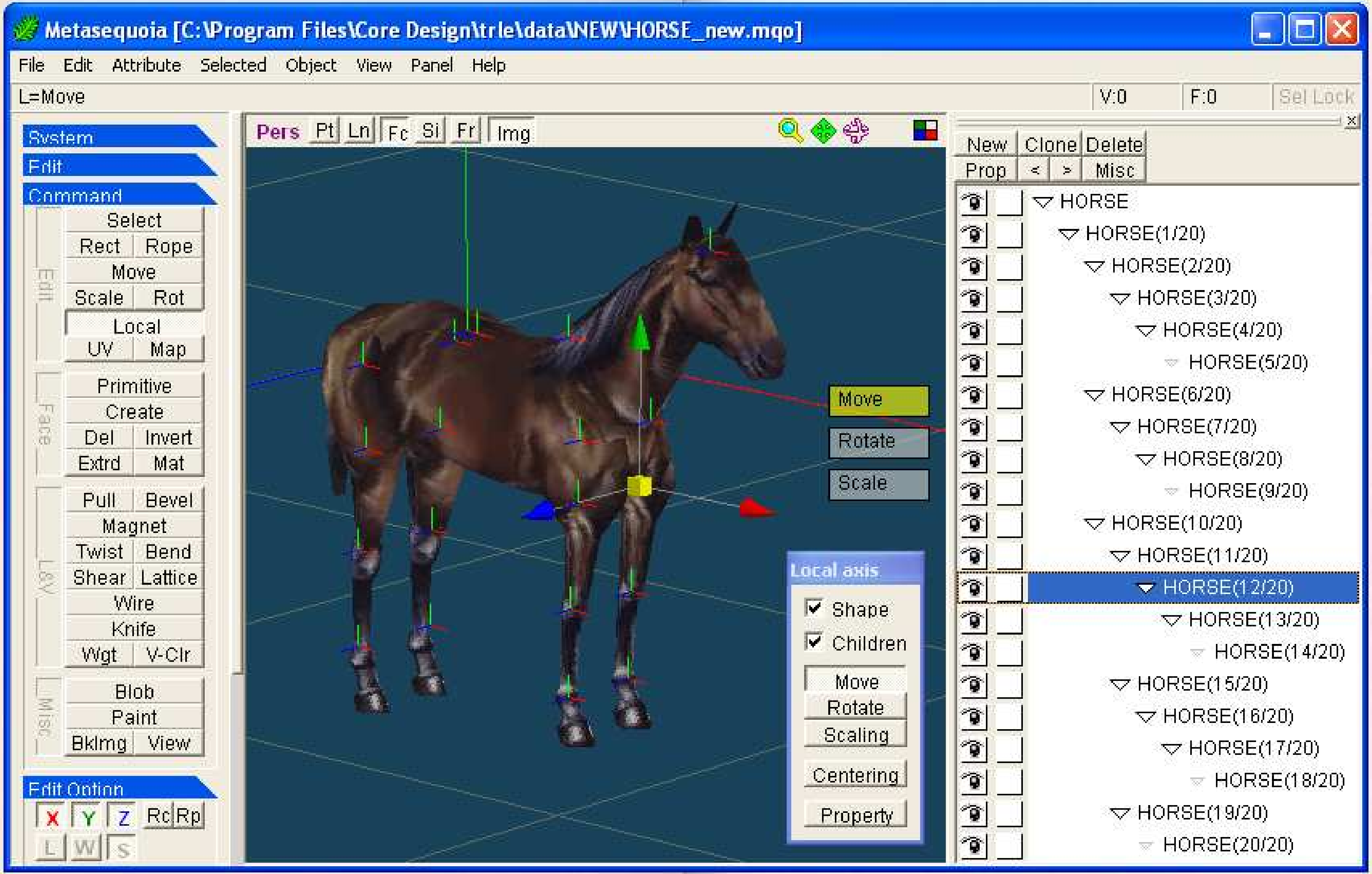
What's new

Mesh trees........................1
Vertex numbers....................4
Lara's Mesh tree..................5
Automatic vertex renumbering.....7
Doubled skin joint meshes.......10
Collision boxes.................11

General

What is meta2tr?................15
What are uv maps?...............17
Preparation in NGLE.............19
Workflow that works.............27
AnimTexRanges...................33
Pixel shader for Metaseq........35
Converting Sketchup to MQO......37
Reprojecting uvmaps in Blender..39

Credits and Disclaimers.........41



Meshtrees are extracted for movables that have one.

Click LOCAL to edit the tree. You can move the pivot points, to match them to a new model.

Mesh tree notes:

- Lara's mesh tree is set in LARA only, see P.5.
- You cannot add or remove "bones" in the tree.
- If, during editing, you have rotated or scaled individual meshes, meta2tr will not be able to import the movable. You can reset these parameters by opening the Property window in the local panel, and setting:
  - local scales to 1.000
  - local angles to 0.000

Vertex numbers.

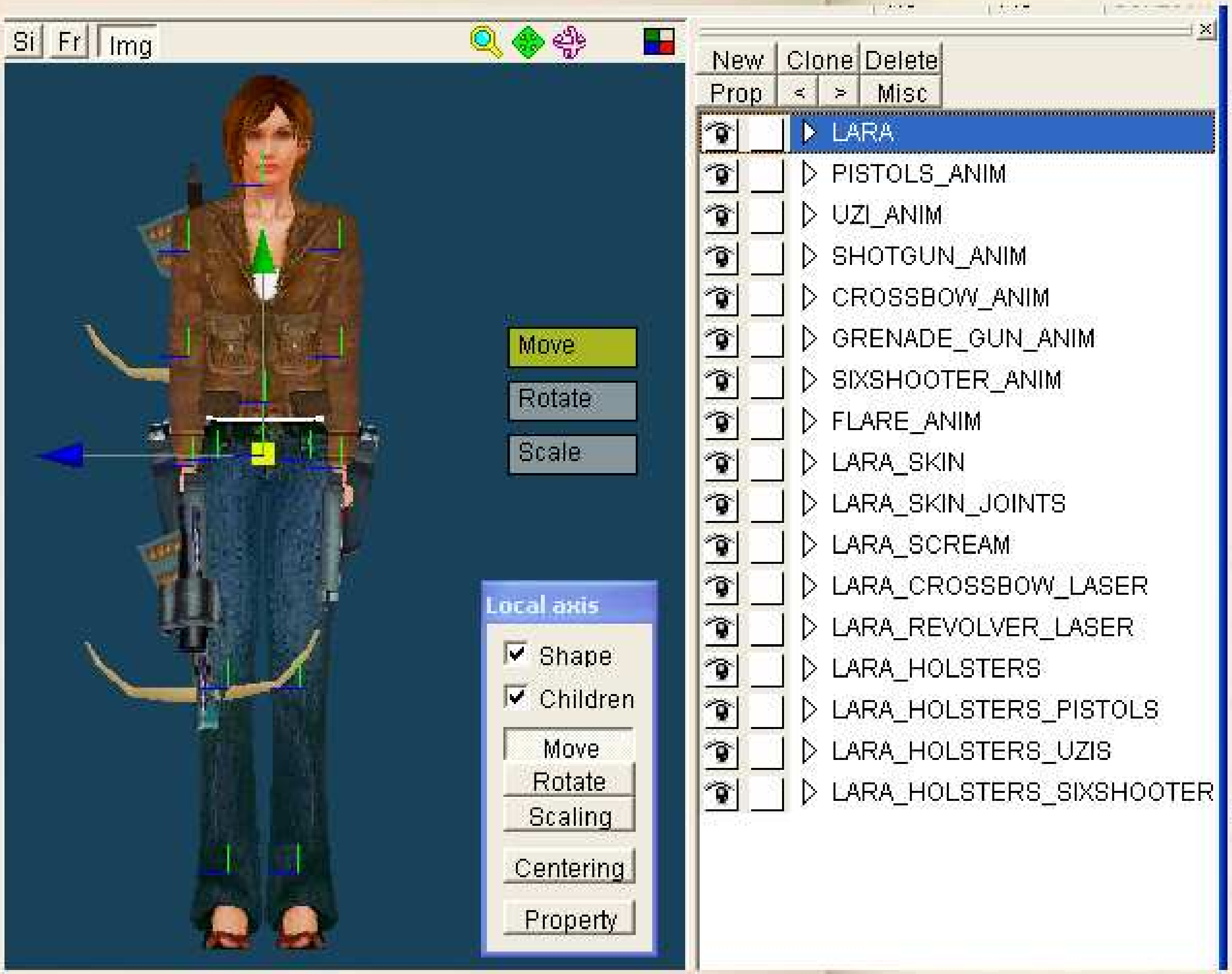
All statics and movables are exported with weights that show the original vertex numbers in the TR4 file.

To see the weight values, click "Wgt", then the vertex. -0.01 = vertex 0, -0.02 = vertex number 1 etc.

This is just for information. Since these values are negative, they are not imported and can be left in the mqo file.

To set a vertex number, enter a positive weight. For example +0.05 = vertex 4, +0.06 = 5, etc.

Automatic vertex renumbering for Lara's skin joints is explained on page 7.



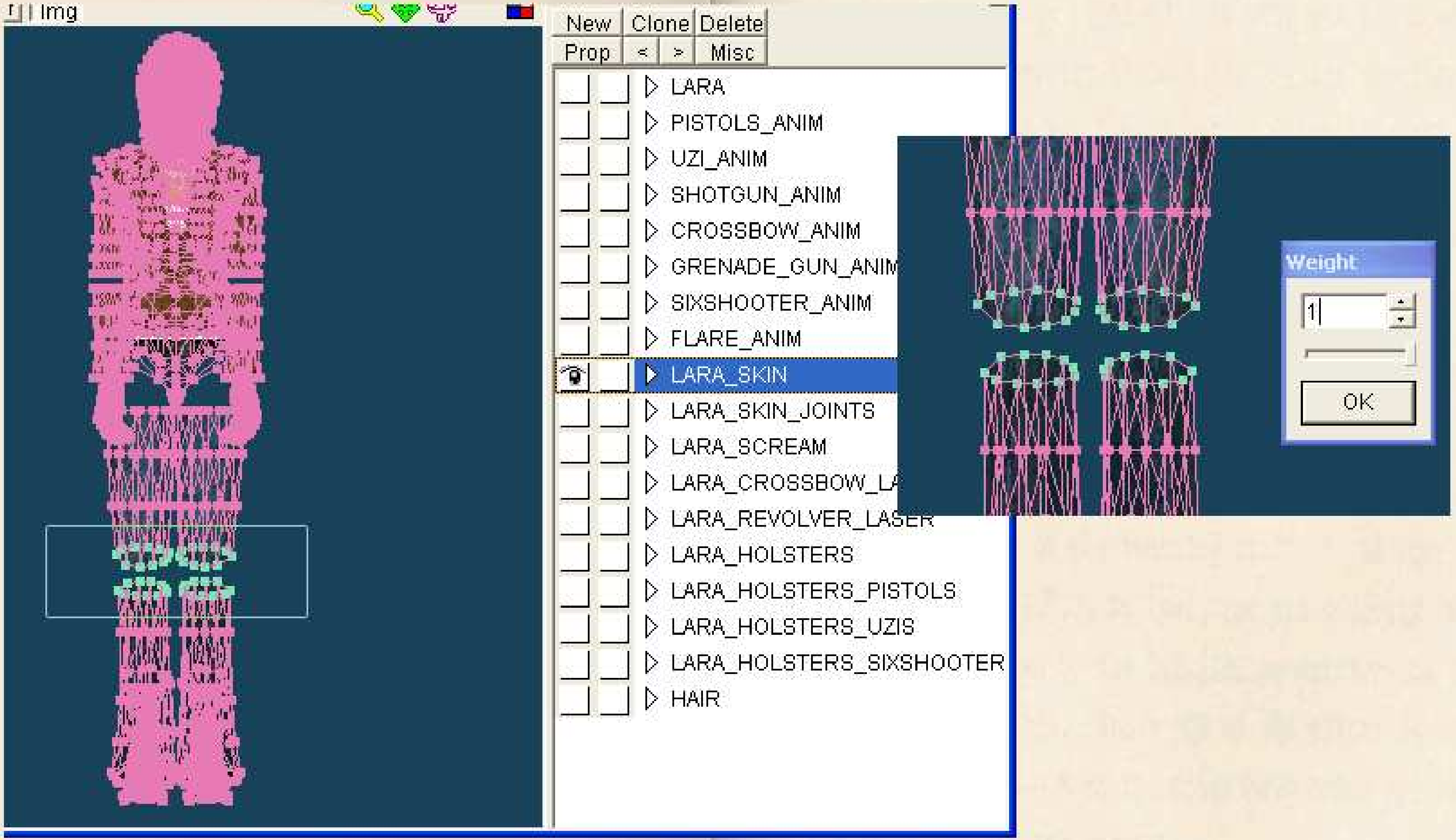
LARA
PISTOLS_ANIM
UZI_ANIM
SHOTGUN_ANIM
CROSSBOW_ANIM
GRENADE_GUN_ANIM
SIXSHOOTER_ANIM
FLARE_ANIM
LARA_SKIN
LARA_SKIN_JOINTS
LARA_SCREAM
LARA_CROSSBOW_LASER
LARA_REVOLVER_LASER
LARA_HOLSTERS
LARA_HOLSTERS_PISTOLS
LARA_HOLSTERS_UZIS
LARA_HOLSTERS_SIXSHOOTER

Lara is made from all of these objects.

They all have a hierachy in Metaseq, but only LARA has the mesh tree (you can see the pivot points in the image.)

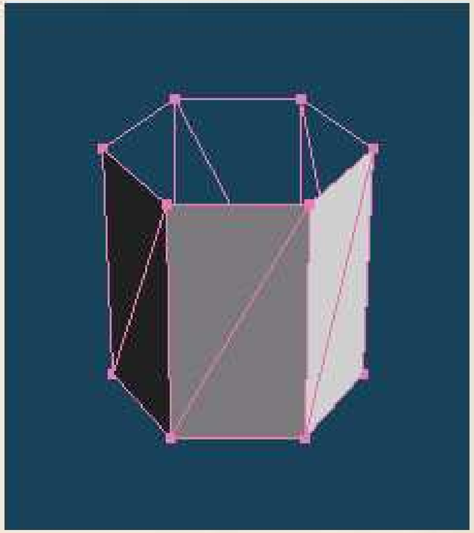
Lara's mesh tree is in the LARA. object.

All LARA-related movables, including MESHSWAP and vehicule versions of Lara, do not have a mesh tree in the MQO file since it is defined in LARA.



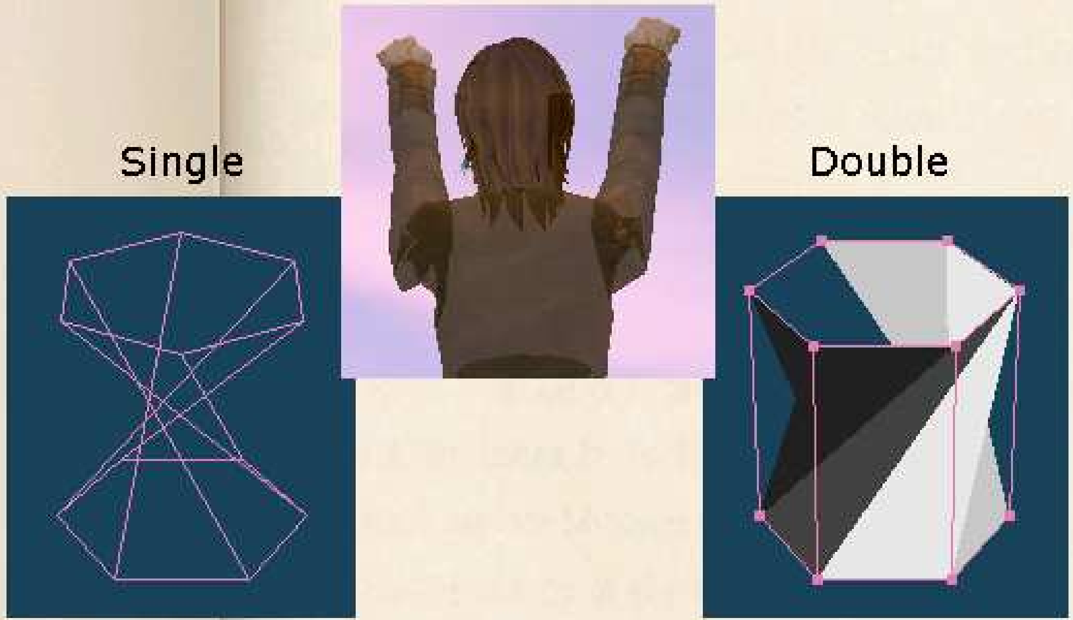
Automatic vertex numbering for the meshes that interface with the skin joints is possible, just select all the joining vertices and set their weight to +1.00.

For the Hair hole, you need specific vertex weights, namely +0.38, +0.39, +0.40 and +0.41. (0.38 = vertex 37, etc). Notice that the weights need to be positive numbers to be imported.



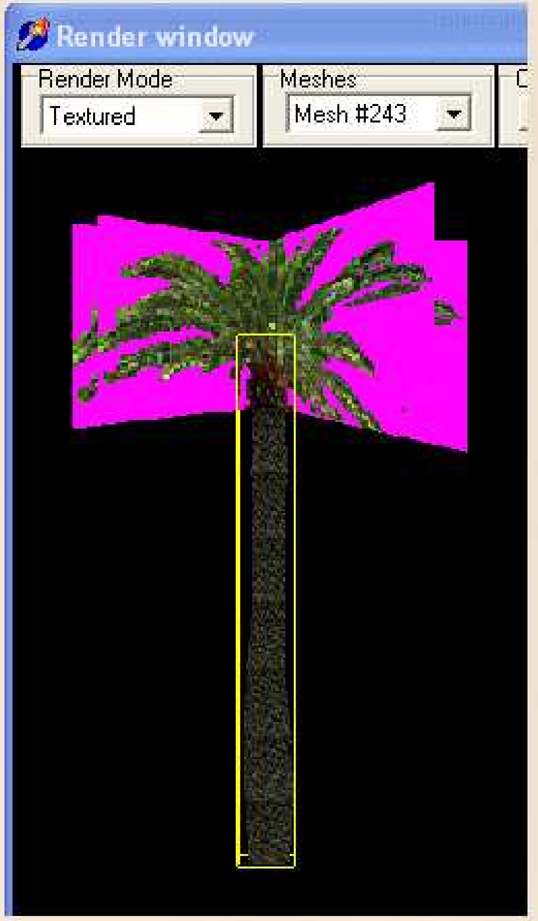
A normal skin joint is a tube of triangles with up to 11 vertices each side.

Use "Align vertices" to get perfect alignment between the vertices shared between the LARA_SKIN object and the joints.



The 3 images on the right show a technique which avoids collapse of the joint at large rotations.

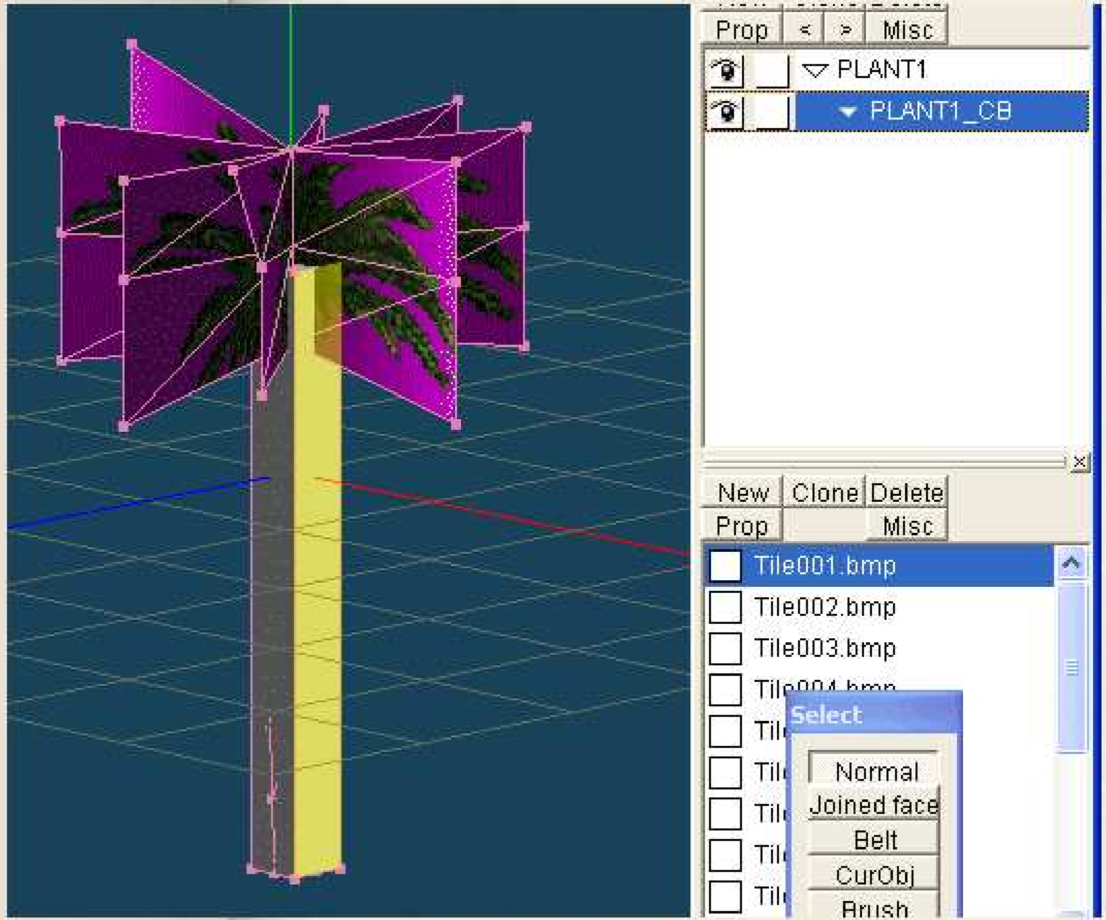
Basically a second twisted set of faces is added, they untwist when the joint is rotated. You can see the effect by comparing the left and right shoulders.



Render Mode: Textured

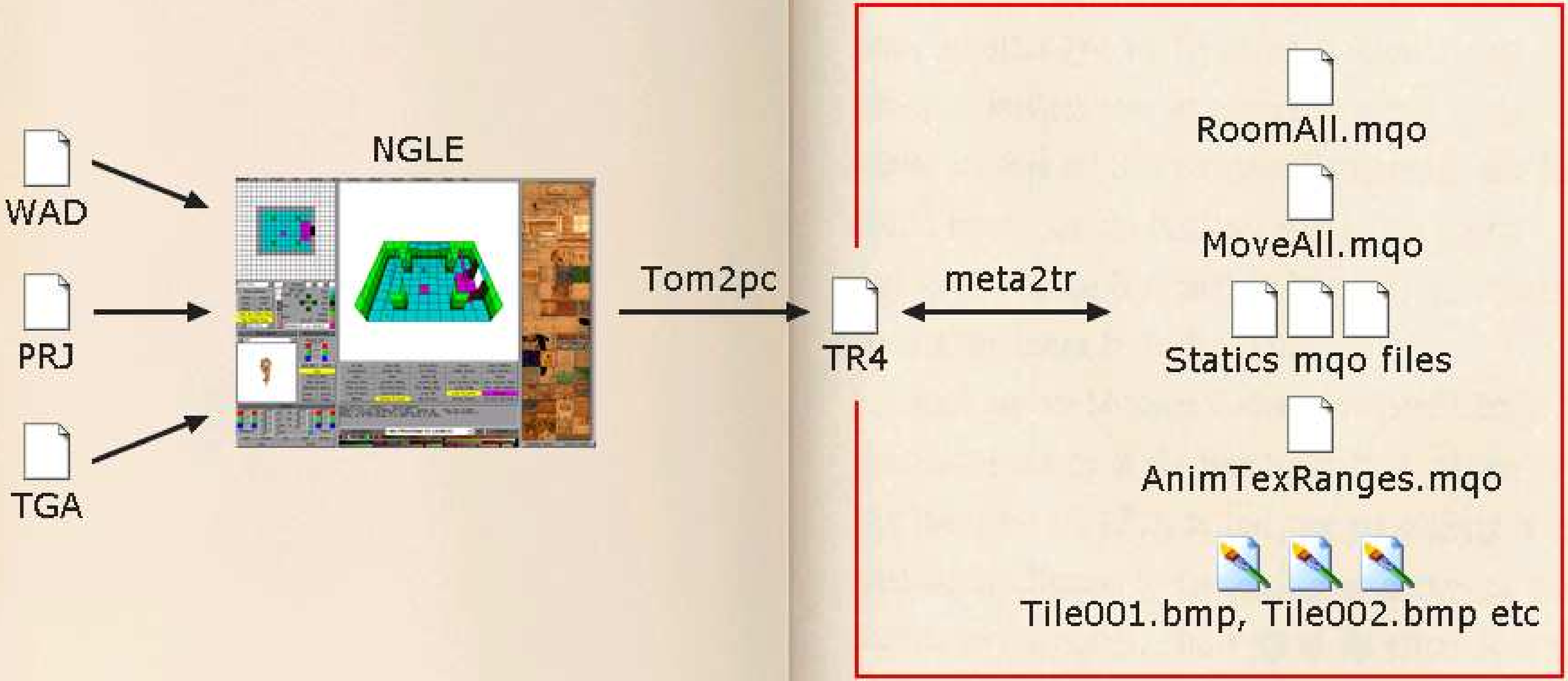
Meshes: Mesh #243

Collision boxes for statics can be resized in Metasequoia, so you don't have to return to Strpix to do this.



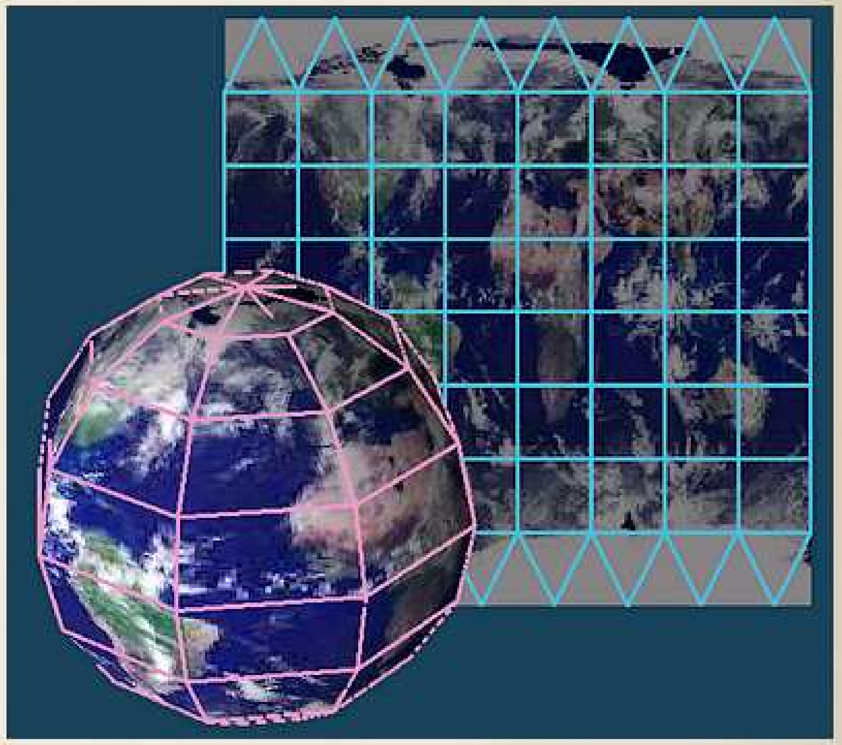
Just edit the CB object. The visibility box in automatically adjusted to the boundaries of the static object's mesh.

General.



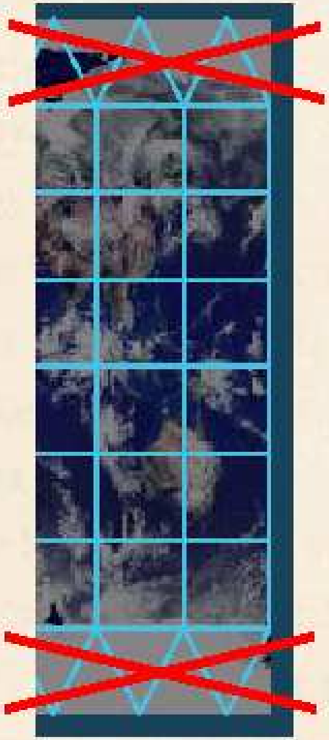
On the left side is the normal method used to produce levels. meta2tr converts between TR4 files and Metasequoia files (MQO and BMP) You can edit the MQO and BMP files, and inject them back into the level.

Main uses:
- make rooms less block-shaped
- add realistic shadows
- real uv-mapping
- import external 3d files



A UV-map maps the faces of a 3D mesh to a 2D texture. Modern texturing tools can unwrap faces of the mesh onto the texture space, this makes painting the mesh surface more intuitive.

Unfortunately, the current WAD file



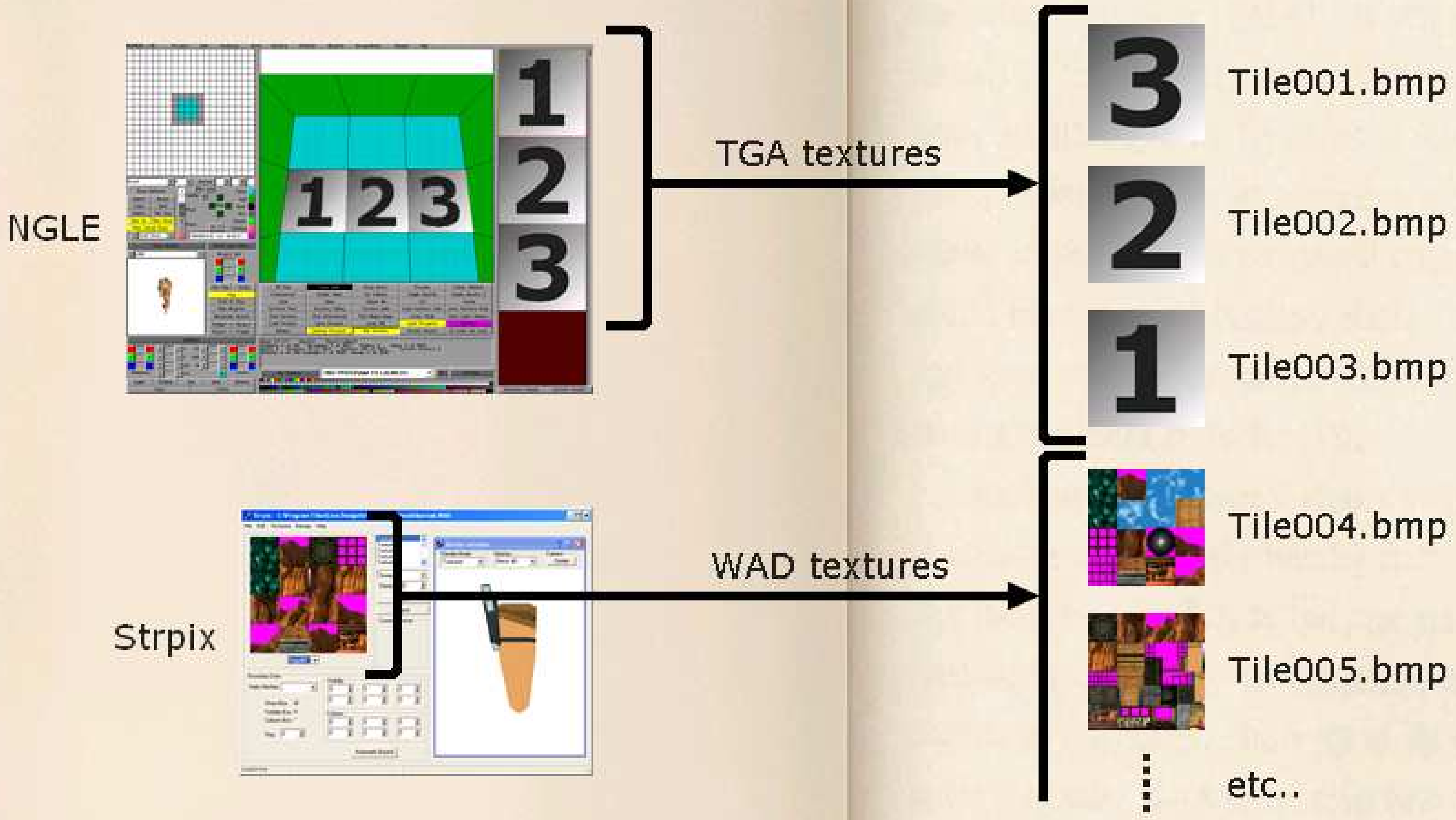
format can only accept textures that fit into perfect boxes, hindering the many objects available for download from being added to a Wad.

However, meta2tr does import real uv maps directly into the TR4 file.

Whether you are replacing Lara or adding a tree to the room mesh any new model will need to have it's textures added to the level.

Check the demo project file: the level has a lot of empty 256x256 textures in the TGA texture set, they are reserving space for use later, either in rooms, or for movables or statics.

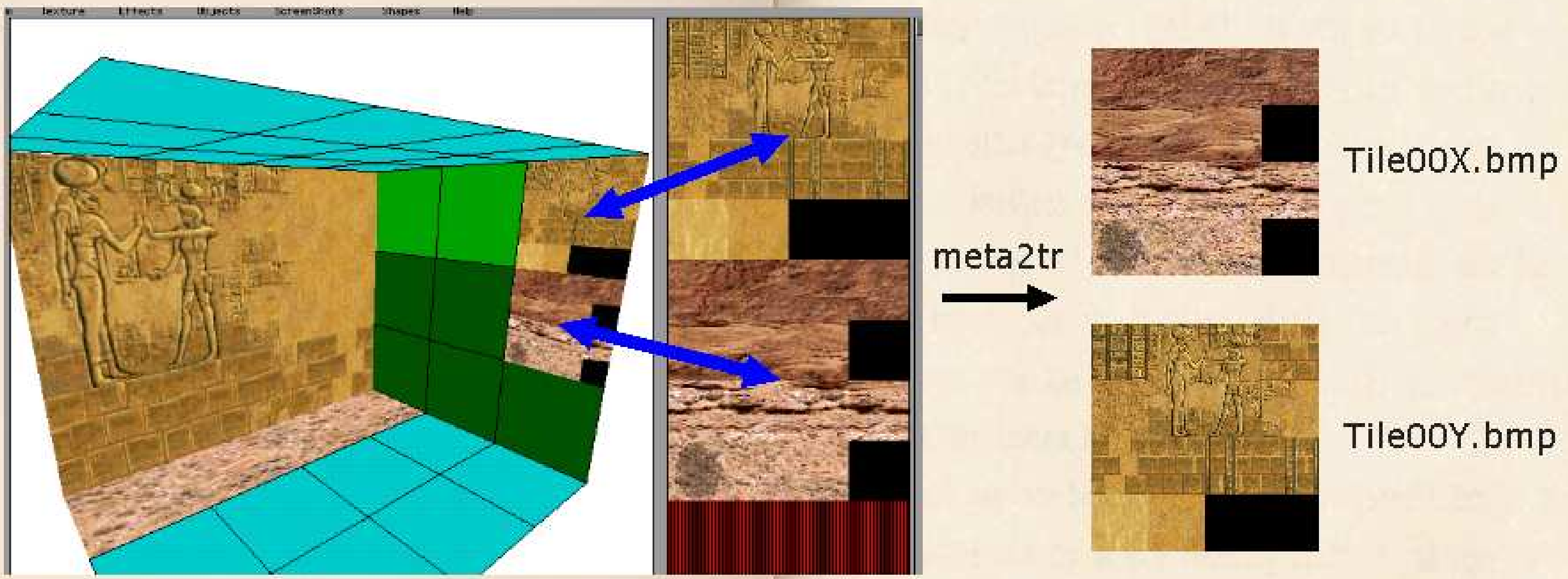
When I extract the level, the first 19 bmp files always contain the room textures plus my empty tiles.



Here's the rule: the first bmp files always contain the textures from the TGA texture set that have been placed on the map, and that are not bumpmapped.

This is why all the empty 256x256 textures are placed on the map in the demo level. (Select "Big Textures" in NGLE to do this).

When using Tomp2pc, you will see a value for RTPages. It is the number of bmp files holding the TGA textures.



For rooms, you may find it useful to group textures in 256x256 areas in the TGA file, then place the 256x256 groups as a SINGLE texture somewhere in a room that Lara cannot get to. The bmp files produced by meta2tr will then contain the same groups.

In the example above, the first 256x256 area in the TGA contains textures for a wall. The second 256x256 area contains rock textures.

The room was textured on the left side using normal-sized textures. Then, on the right, each group was placed as a 256x256 texture (blue arrows) which means the bmp files will be identical to the TGA areas.

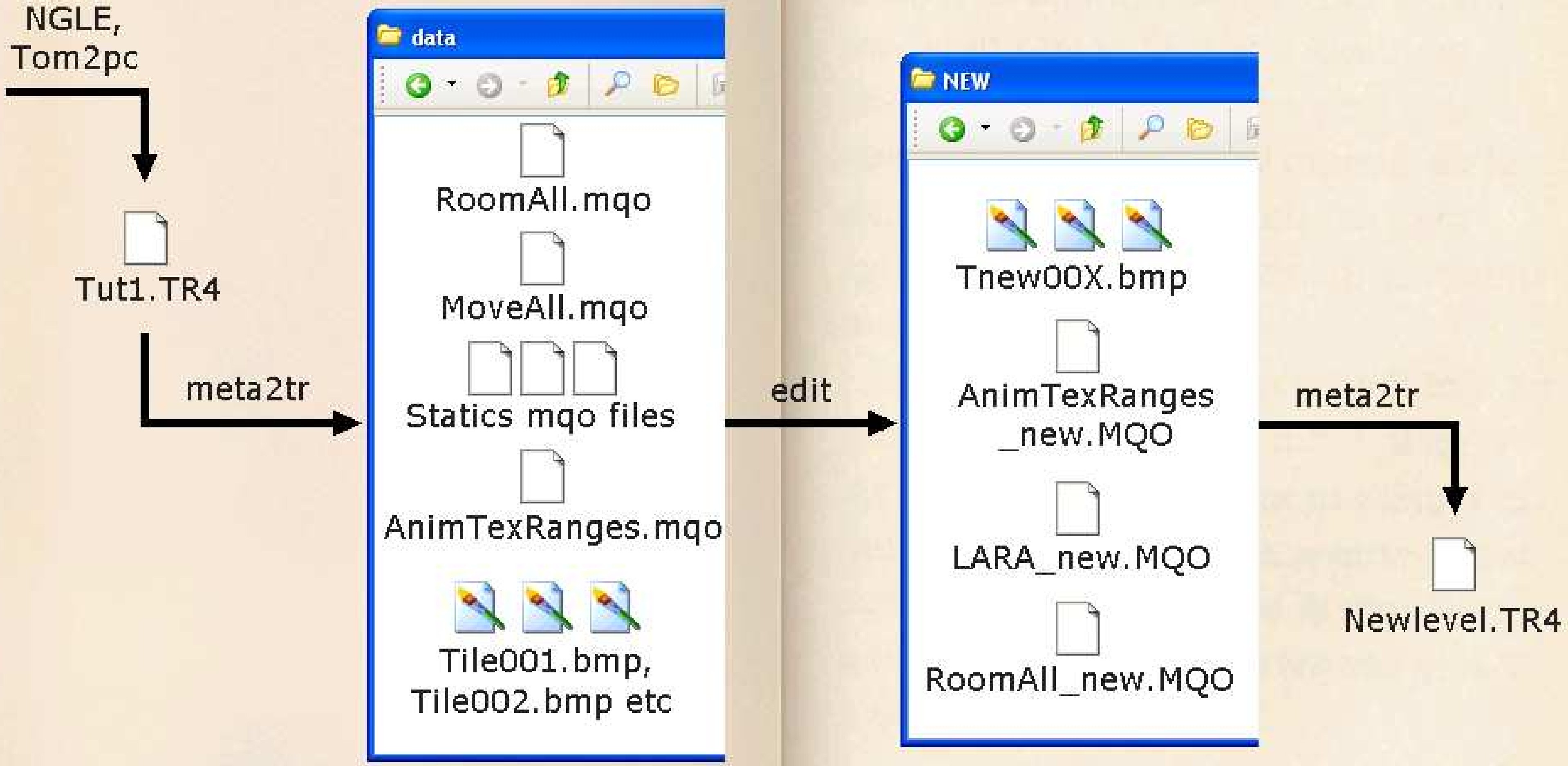
In this way a bmp file in Metaseq can represent a "material" (wall, rock, etc)

Collision comes from NGLE. Some modifications to the room mesh (eg rounded walls) may need collision to be prepared in NGLE. See the meta2tr web site for this.

You can later return to NGLE for any operation, but remember, adding objects to your Wad or rooms to your map will require you to:

- reserve textures for it
- texture it with Metasequoia

Recommended workflow



NGLE,
Tom2pc

Tut1.TR4

meta2tr

data

RoomAll.mqo

MoveAll.mqo

Statics mqo files

AnimTexRanges.mqo

Tile001.bmp,
Tile002.bmp etc

edit

NEW

Tnew00X.bmp

AnimTexRanges
_new.MQO

LARA_new.MQO

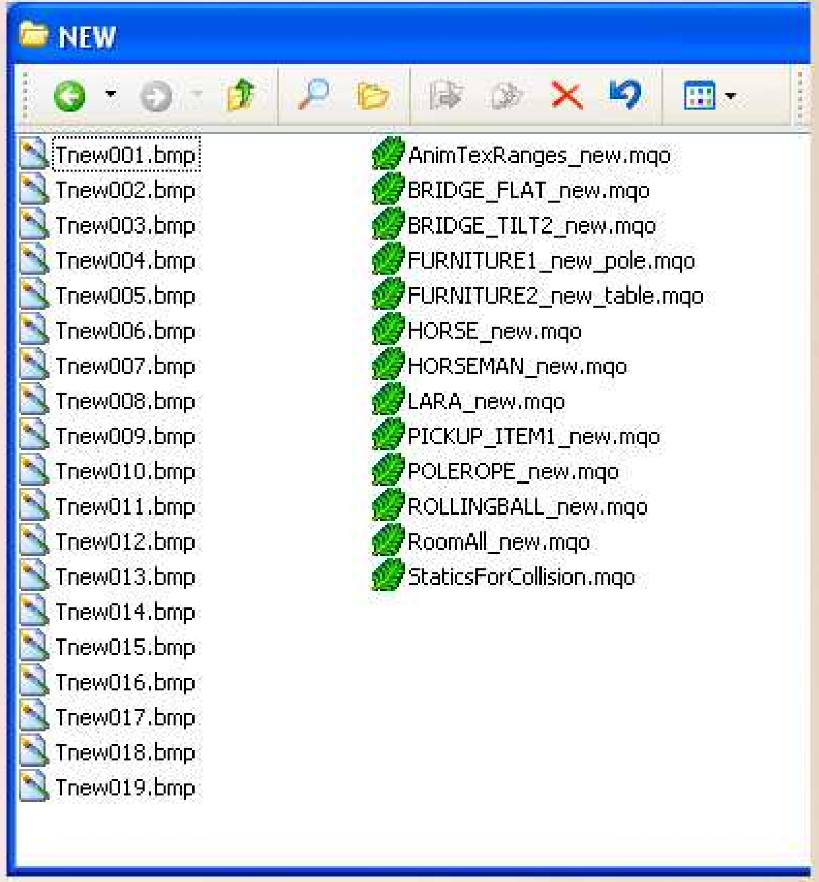
RoomAll_new.MQO

meta2tr

Newlevel.TR4

1) With meta2tr in the data folder, drag the TR4 level file onto the the meta2tr icon. This "unpacks" the level into DATA

2) Now create a sub-directory in DATA called NEW. It will hold all of the edits, and becomes the visual reference for the level.



Tnew001.bmp
Tnew002.bmp
Tnew003.bmp
Tnew004.bmp
Tnew005.bmp
Tnew006.bmp
Tnew007.bmp
Tnew008.bmp
Tnew009.bmp
Tnew010.bmp
Tnew011.bmp
Tnew012.bmp
Tnew013.bmp
Tnew014.bmp
Tnew015.bmp
Tnew016.bmp
Tnew017.bmp
Tnew018.bmp
Tnew019.bmp
AnimTexRanges_new.mqo
BRIDGE_FLAT_new.mqo
BRIDGE_TILT2_new.mqo
FURNITURE1_new_pole.mqo
FURNITURE2_new_table.mqo
HORSE_new.mqo
HORSEMAN_new.mqo
LARA_new.mqo
PICKUP_ITEM1_new.mqo
POLEROPE_new.mqo
ROLLINGBALL_new.mqo
RoomAll_new.mqo
StaticsForCollision.mqo

3) Copy all the MQO/BMP files in DATA to NEW.

4) Edit in Metasequoia. This is where the reserved tiles start to be used. (see P.6) To prevent losing your edits after future updates in NGLE,

any edited object should only use the NGLE tile range. (this is explained on P.19).

5) For any object or room that needs updating in the level, drag it from NEW onto the meta2tr icon in DATA. Don't forget that the visibility flag can be used to select single items, which makes import faster.

Notes:

- You don't need to add "new" to the file names.

- I put movables and statics into separate MQO files, but you don't have to.

- I dedicated tile ranges to objects. This means the object can easily be reused in a different level.

You can return to NGLE for any operation, but remember, adding objects to your Wad or rooms to your map requires you to integrate the room/object into NEW, and to texture it, using tiles from the NGLE range.

If you have returned to NGLE, and used tom2pc to create a new version of the TR4 file, then:

1) Drag the TR4 file in DATA onto the meta2tr icon. This produces new BMP and MQO files in DATA, but they do not replace the files in NEW, which are the reference for your edits.

2) Make sure all of the items in the mqo files in NEW have visibility on

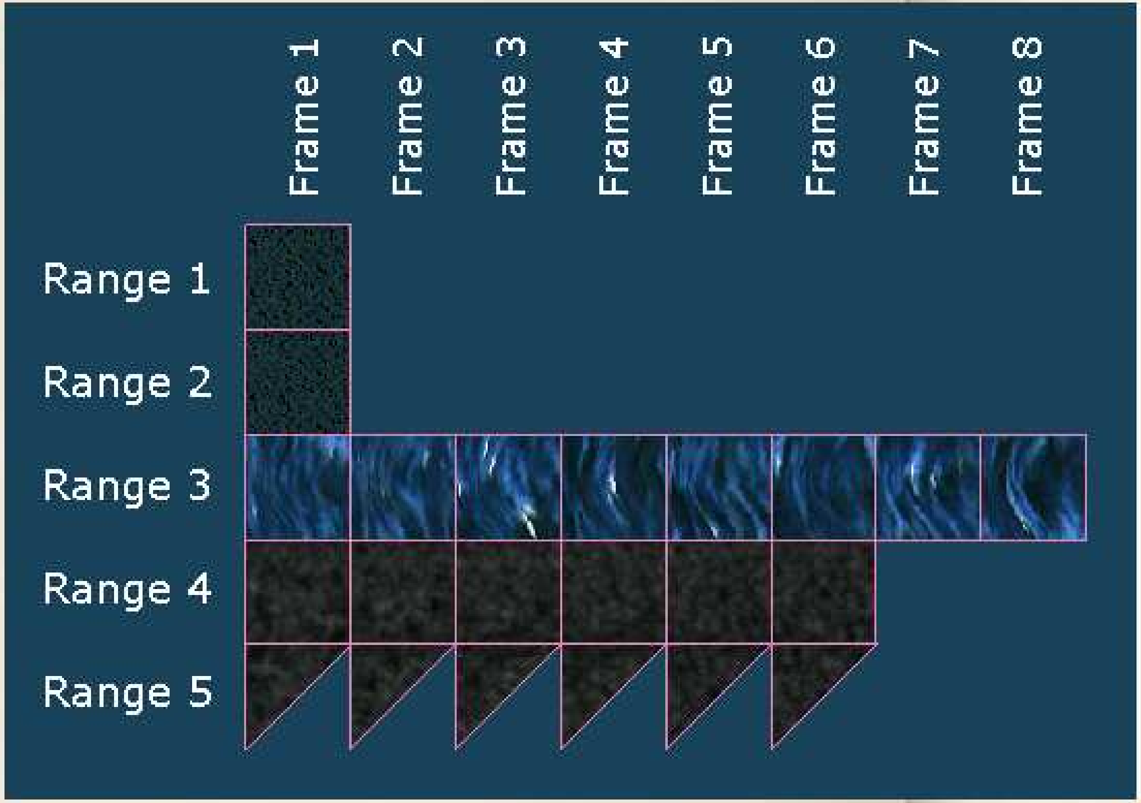
3) Drag, one by one, all the edited MQO files in NEW onto the meta2tr icon.

4) Drag all the tiles from the NGLE range in NEW onto the meta2tr icon. You can simultaneously drop upto about 30 tiles.

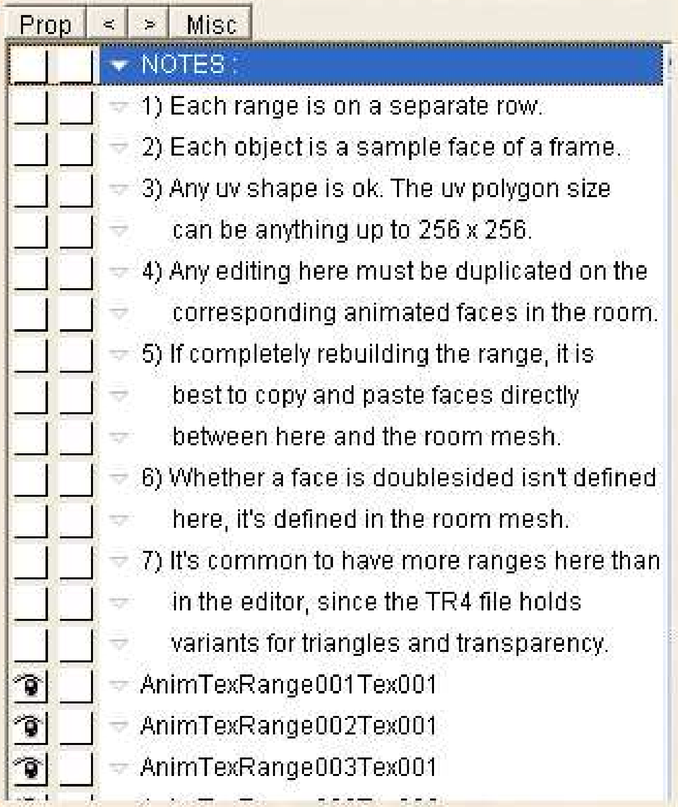
This builds a new level (Newlevel.tr4) based on the newer TR4 file, having the edits in NEW.

An update in NGLE may reorder textures and tiles in DATA, but by reinjecting everything back into the level, the order is restored to what it was before. This way the textures stay in sync with the MQO files, so you don't lose any work.

Thanks to Psiko for this awesome idea! This is the basic foundation of his HyperSquare project and he will have more detail in his tutorials.



AnimTexRanges contains the animated texture ranges in the TR4 file.

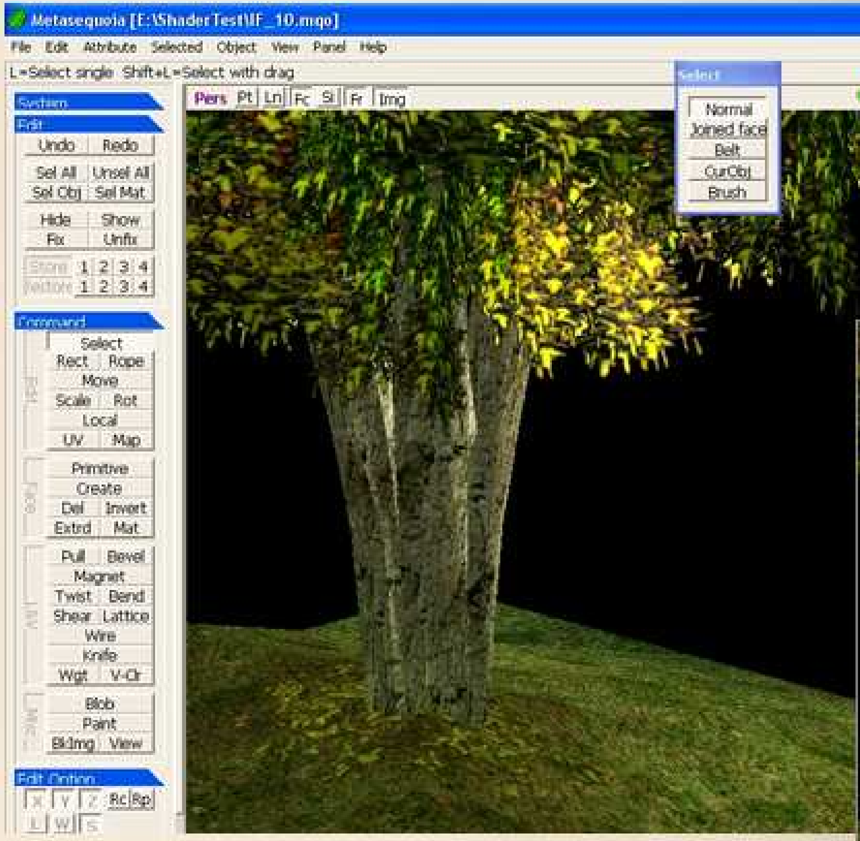


NOTES :

1) Each range is on a separate row.
2) Each object is a sample face of a frame.
3) Any uv shape is ok. The uv polygon size can be anything up to 256 x 256.
4) Any editing here must be duplicated on the corresponding animated faces in the room.
5) If completely rebuilding the range, it is best to copy and paste faces directly between here and the room mesh.
6) Whether a face is doublesided isn't defined here, it's defined in the room mesh.
7) It's common to have more ranges here than in the editor, since the TR4 file holds variants for triangles and transparency.

AnimTexRange001Tex001
AnimTexRange002Tex001
AnimTexRange003Tex001

See the notes above for more.



In Metasequoia

Recent versions of Metasequoia can accept pixel shaders, if your graphics card supports PS 2.0.

The meta2tr download includes a modified blinn.fx shader file.

In game.

Copy it to the HLSL directory in Metaseq and restart Metasequoia, and you will have room lighting exactly like it is in the game.

This makes editing lighting quicker and easier.

The most reliable method I found to convert Sketchup to MQO was using FBX converter from Autodesk.

1) In Sketchup, remove any groups.

2) Save as Google Earth 4 (*.kmz)

3) Change the extension to ZIP.

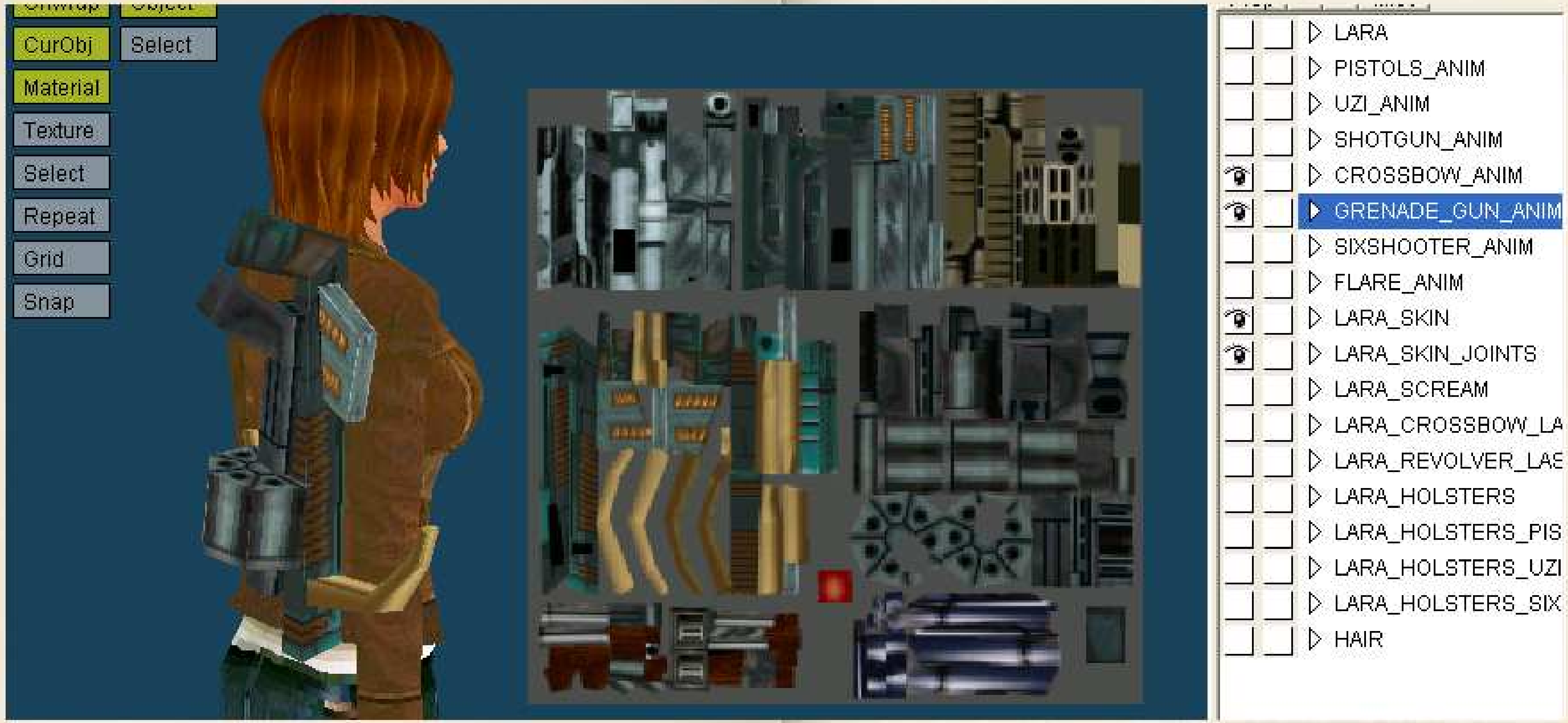
4) Unzip to a directory

5) In FBX convertor, convert the .dae file to .fbx

6) In FBX convertor, convert the .fbx files to .3ds or .obj

7) Open the .3ds or .obj file in Metasequoia.

If the original object didn't use a uv map, you will have to create one.



When adding an object to a level, it is useful to have a set of tiles dedicated just to that object, this makes it easier to use the object in a different level.

The Lara that I added to this demo level did not have weapons so I decided to remap the original ones to a single tile.

To do this automatically, I used Blender 2.49b. To import the meshes, I modified the script mqo_import.py for Blender, it is included in the download.

Note that this script doesn't accept MQO files with vertex weights. Select all, click "Wgt" and enter 0.00 to remove them.

Credits.

Core Design, for the Tomb Raider game.

Psiko, for beta testing

R.Stone, Popov, Rgbold, TRWad, Turbo Pascal, Dr. Willard, Yuri Zhivago and Iceberg, for the TR4 documentation and the various editing tools they built.

O.Mizno, for Metasequoia.

Paolone, for NGLE

Many others that provided free textures, free 3d models, free software that made meta2tr and this demo possible.

Disclaimers.

This level was not made and is not supported by Core Design Ltd. and Eidos Interactive.

meta2tr is free software. Use at your own risk. Back up your files.