

Blender Jedi Academy Plugin Suite Manual

Willi “mrwonko” Schinmeyer

December 23, 2014

Contents

1	Purpose of this document	3
2	Changelog	3
3	Installation	4
4	ASE	4
4.1	ASE Exporter	4
4.2	ASE Importer	5
5	ROFF	5
5.1	ROFF Exporter	5
5.2	ROFF Importer	5
6	MD3	6
6.1	MD3 Exporter	6
7	Ghoul 2 (GLM/GLA)	6
7.1	Features of the File Format	7
7.2	Limits	7
7.3	Scene Setup	7
7.3.1	Hierarchy	8
7.3.2	Custom Properties	8
7.4	GLM Import	8
7.5	GLA Import	10
7.6	GLM Export	10
7.7	GLA Export	10
7.8	animation.cfg Export	11

1 Purpose of this document

This manual is supposed to explain the various Blender plugins for Jedi Academy file formats I've worked on. It is not supposed to explain how to use Blender, there's plenty of documentation on that already.

2 Changelog

0.1 (2012-10-28)

Initial Release for Blender 2.64. Includes:

- ASE Exporter (static geometry)
- ASE Importer (static geometry)
- GLA Exporter (animations)
- GLA Importer (animations)
- GLM Exporter (animated (player)models)
- GLM Importer (animated (player)models)
- MD3 Exporter (static/per-vertex animated geometry)
- ROFF Exporter (rigid body animation)
- ROFF Importer (rigid body animation)

0.2 (2013-03-13)

Mostly a bugfix release:

- Fixed an implicit int-to-bool conversion in the GLM importer that made Blender 2.65+ cringe (i.e. fixed GLM import in 2.65+)
- Possibly fixed a bug that might've lead to incorrectly imported tags on GLM files
- Fixed a bug where new skeletons/animations (not based on existing ones) would not export correctly due to an invalid inverted base matrix.
- Fixed a bug where ROFF import did not work due to looking for files with an .ase extension
- Fixed a bug where ROFF export would append .ase instead of .rof
- Added UV support to ASE import (possibly broke import for some faces in very rare cases? Report if there's a missing face on import.)
- ROFF Export scale defaults to 100% now
- Wrote this manual (yay!)

0.2.1 (2013-03-18)

Hotfix

- Fixed last sequence being 1 frame to long in animation metadata export
- Fixed `skeleton_root`'s transformation being ignored (e.g. translation or scale relative to `scene_root`)

0.2.2 (2014-12-23)

Hotfix

- Fixed roff exporter not exporting calculated transformations (e.g. from constraints) correctly.

3 Installation

Open the User Preferences (File/User Preferences or Ctrl+Alt+U), go to the Addons tab and press there “Install from File” button. Select the zip archive containing the addon. If you wish to manually install the addon, refer to the Add-Ons section of the Blender Manual for the correct paths.

Once installed, don't forget to enable the addon. It's located in the Import-Export category and called “Import-Export: Jedi Academy Import/Export Tools”. Then click Save User Settings so it'll still be activated the next time you start Blender.

4 ASE

The ASE file format (Ascii Scene Export) is a plaintext model format that is widely used in Jedi Academy due to its simplicity and 3ds Max's ability to export it out of the box.

The Plugin supports only a subset of the features the ASE format has, but so does Q3Map2. Everything important is there, unless normals are important to you.

4.1 ASE Exporter

Located in “File/Export/JA ASE (.ase)” or invoked by entering “Export JA ASE (.ase)” in the Search Window opened by default with Space.

Things to keep in mind:

- All mesh objects in the current scene will be exported.
- The meshes have to be UV mapped.
- The meshes should be triangulated – although no error will be raised if they're not, only the first 3 vertices of each face will be exported.
- The texture used will be the name of the first material on the object. Multi-material objects are not supported.

- Normals are not exported.
- Models do not need to be split at UV seams, Q3Map2 will take care of that.

4.2 ASE Importer

Located in “File/Import/JA ASE (.ase)” or invoked by entering “Import JA ASE (.ase)” in the Search Window opened by default with Space.

Not much to say about this one... Imports meshes with uv coordinates from an .ase file, ignoring pretty much everything else, as does Q3Map2. Textures get applied as material names.

5 ROFF

ROFF is short for Rotation Origin File Format. And it’s pretty much just that: It saves the rotation and origin/position of an object each frame, relative to the previous position/rotation. Can be used to animate rigid objects like brushes (or bmodels, more specifically) via a script (like the boulder in t3_rift), as well as for camera paths.

One thing to note is that the framerate is saved as milliseconds per frame, so rounding may occur. The files shipped with Jedi Academy all have 50ms/frame, which is exactly 20 FPS.

5.1 ROFF Exporter

Located in “File/Export/JA ROFF (.rof)” or invoked by entering “Export JA ROFF (.rof)” in the Search Window opened by default with Space.

Features:

- Exports location/rotation of the currently selected object over time, so select exactly one object.
- Uses Blender’s start/end frames to determine the animation’s length.
- Uses Blender’s FPS setting from the Render Settings.
- Converts 1:1 between Blender Unit and Jedi Academy’s Units, unless you supply a different scale in the export settings.

5.2 ROFF Importer

Located in “File/Import/JA ROFF (.rof)” or invoked by entering “Import JA ROFF (.rof)” in the Search Window opened by default with Space.

Features:

- Imports location/rotation for the currently selected object over time, so select exactly one object.
- Sets Blender’s start/end frames to match the animation’s length.
- Sets Blender’s FPS setting in the Render Settings to match the one in the file.

6 MD3

Originally created for Quake 3, the MD3 format is for models, with the possibility of per-vertex-animations (which make for huge files). It notably uses fixed-point coordinates for the vertices, resulting in the requirement that models need to stay within the $[-512; 512)$ range and only have a precision of $1/64$ unit, and support so-called Tags to mark locations on the model (e.g. muzzle flash position/rotation on a weapon).

If you want animated map models, this should be the format of your choice.

6.1 MD3 Exporter

Located in “File/Export/JA MD3 (.md3)” or invoked by entering “Export MD3” in the Search Window opened by default with Space.

Acknowledgement: I did not originally write this exporter, I improved upon one I found online which was at the time maintained by Xembie, although I believe he did not initially start it. His version has since been updated and the current version is available on

<http://sourceforge.net/projects/md3exporter/>

Features:

- Exports *animated* MD3 models. So if you don’t want animations, make sure to set the Blender End Frame to the same as the Start Frame or your model will turn out needlessly large.
- Looks for a custom property called “md3shader” on the *objects* to determine the texture to use. (Make sure not to put the custom property on the mesh.)
- Exports only selected objects.
- Mesh objects get exported as meshes.
- Empty objects get exported as tags.
- Can be scaled and offset using the corresponding Export settings.
- The “MD3 Name” Export setting should be set to the path of your model including the extension, e.g. “models/map_objects/imperial/field_post.md3”.
- Make sure your model does not exceed the $[-512; 512)$ range or you’ll get a struct.error on export.

7 Ghoul 2 (GLM/GLA)

The Ghoul 2 format is Raven Software’s skeleton-based model/animation file format used for playermodels in Jedi Academy and its other Quake 3 Engine games. GLM files are Ghoul 2 Models, which are saved separately from the GLA files, the Ghoul 2 Animations, which can be shared amongst models.

7.1 Features of the File Format

- Levels of Detail (LOD) – less expensive versions of the model for use at greater distances and on slower machines.
- Tags – placeholders marking locations on the model, e.g. where weapons should be held. Represented by single triangle meshes so they can be skinned to the armature.
- Skins – not a Ghoul 2 feature per se, but tightly connected: A file describing what textures to put on which surface.

7.2 Limits

Due to the way data is saved there are some inherent limits for Ghoul 2 files:

- The Mesh must be triangulated.
- Every Vertex has to be weighted to the skeleton.
- A Mesh can reference no more than 32 bones.
- Any single vertex can reference no more than 4 out of those 32 bones.
- Names must be no longer than 64 characters (bones, objects etc.).
- UV coordinates are per-vertex, meaning that any single vertex can only have one UV coordinate. This necessitates splitting the mesh at UV seams, which can for example be achieved by also marking seams as “sharp” and using the “Edge Split” modifier to split at those sharp edges.
- Bones must not be scaled. The file format only saves location and rotation and applying scale anyway while lead to (visually) broken animations.
- Bone must not move further than 512 units away from their base pose.
- Objects that are to be exported need the Ghoul 2 Custom Properties
- LODs are similar: Lower Levels of Detail (with higher indices) can have no additional Meshes, and the Meshes that are present will use the same texture as the ones on LOD 0.

Jedi Academy imposes some additional limits, most notably a maximum of 1000 vertices per object.

7.3 Scene Setup

The exporter uses a very specific setup to determine the Ghoul 2 specific settings. The Importer creates such a hierarchy, so you can reexport an imported model without any changes.

7.3.1 Hierarchy

The root object is called “scene_root” and will likely be an Empty. Its children are the Armature, called “skeleton_root”, if any, and the different LODs, name “model_root_[LOD]”, where [LOD] is the Level of Detail starting at 0 for the most detailed one. The actual hierarchy below that only matters for LOD 0, because the others mirror it. “model_root_0” should have a single child, which corresponds to the root object in the GLM, and below that the hierarchy is up to you, as far as the exporter is concerned (for proper dismemberment in Jedi Academy you’ll need to adhere to a certain pattern though, look at existing models for reference).

7.3.2 Custom Properties

To be able to match the objects across LODs they need the same name. This is impossible in Blender, object names are unique, so the exporter uses a custom property instead. (In theory a prefix would be possible as well, this solution is a leftover from the olden days when names were severely limited in length.) For easier management of the custom properties the Plugin adds a new Panel to the Object Properties called “Ghoul 2 Properties”. For an object without the required Custom Properties it consists of a “Add G2 properties” button. Press it to add the properties.

Note that for all LODs except the first one (LOD 0), only the name is used since the properties are the same across all LODs, which is why the matching of objects is necessary in the first place. This especially means that the surfaces will use the same texture!

There are the following properties:

name The surface’s name, for matching across LODs.

shader The material to apply to this surface. Ignored in favor of skin files in Jedi Academy.

tag Whether this surface is a Tag, i.e. represents a position, e.g. where a weapon should be held. The most important part about a Tag are the first 3 vertices, although there has to be at least one triangle. Vertex 2 is the origin, Vertex 0 the direction of the X-Axis and Vertex 1 the direction of the Y-Axis, although some math is involved to guarantee they’re perpendicular. In Raven’s models, the short leg is the X Axis, the long one the Y Axis. (Unless they screw up, as in R2D2’s case.)

tag Whether this surface should be hidden. Mostly ignored though – Mod-View looks for _off suffixes in the names while Jedi Academy consults the skin files.

7.4 GLM Import

Located in “File/Import/JA Ghoul 2 model (.glm)” or invoked by entering “Import JA Ghoul 2 model (.glm)” in the Search Window opened by default with Space.

Settings

Skin If you'd like to load the textures from a skin file, enter its name here. If the model is called "model.glm" and you'd like to use "model_blue.skin", just enter "blue".

Guess Textures Since Jedi Academy uses skin files, the actual textures specified in the GLM are ignored. It's probably for that reason that their first character is replaced by the end-of-string symbol. Guessing the original first character is not so hard if the path is "?odels/players/..." though, so tick this box to make the importer try to figure it out. You're advised to just use the Skin setting instead though.

Base Path The importer tries to load the textures and apply them to the model, as well as the matching .gla. To be able to do that, it needs to know where to look. It will try to examine the path of the opened file, looking for "GameData" in it. If you're using a non-standard folder layout that may fail, so you can enter a custom path to be prepended to textures when searching.

.gla override GLM files reference the GLA they're for (for Jedi Academy's playermodels that's "models/players/_humanoid/_humanoid.gla") and the importer loads the skeleton as well. Here you can set a different GLA to use instead, using Jedi Academy's base-relative notation (e.g. "models/players/_humanoid_kor1/_humanoid_kor1.gla"). This will only work if the number of bones matches.

Scale A scale to apply, since using a 1:1 conversion from Jedi Academy units (roughly an inch each) to Blender (typically a meter) results in huge models. This only rescales the scene_root, and since export is done relative to that this setting will automatically be ignored when exporting.

Skeleton changes Jedi Academy uses a Skeleton/Armature that's a little strange in places. For example the lower arms are not children of the upper arms, making animation quite the pain. The plugin can automatically correct the default skeleton in that regard if you select "Jedi Academy _humanoid" here. Animations will still be properly applied and exporting works with a changed hierarchy, too – the plugin will do the required math.

Animations If you want to import animations, make sure not to set this to "None". Importing all frames takes forever though, and Blender can't handle it very well either – that's because in GLA files every frame is a keyframe and there's over 20000 of them in Jedi Academy's _humanoid.gla. For that reason there's an option to import only a range.

Start frame If you choose to only import a range of frames (and make sure to choose that in the Animations setting if this is your intention, this setting on its own won't do anything!), this sets at which frame to start, counting from 0.

Number of frames If you choose to only import a range of frames, this sets how many to import.

7.5 GLA Import

Located in “File/Import/JA Ghoul 2 skeleton/animation (.gla)” or invoked by entering “Import JA Ghoul 2 skeleton (.glm)” in the Search Window opened by default with Space.

Settings

Refer to the GLM Import settings, since these are a subset of them.

7.6 GLM Export

Located in “File/Export/JA Ghoul 2 model (.glm)” or invoked by entering “Export JA Ghoul 2 model (.glm)” in the Search Window opened by default with Space.

Settings

Base Path See GLM Import Base Path setting.

.gla name Which skeleton this model uses, since internally no bone names are used, just indices, so the exporter has to look them up. Use Jedi Academy’s base-relative notation, e.g. “models/players/_humanoid/_humanoid.gla”. Some models, like weapons, have no skeleton - leave this empty in that case.

7.7 GLA Export

Located in “File/Export/JA Ghoul 2 skeleton/animation (.gla)” or invoked by entering “Export JA Ghoul 2 Skeleton & Animation (.gla)” in the Search Window opened by default with Space.

Settings

Base Path See GLM Import Base Path setting.

gla name The filename of the gla you’re exporting. The exporter will probably be able to guess this based on the Base Path so you can leave it empty.

gla reference If you’re not creating a completely new skeleton, but merely new animations for an existing one, you have to make sure the indices of the bones match. Well, you don’t have to make sure yourself, the exporter can do that, but you’ll have to tell it which skeleton to match (matching done by bone name), and that’s what this setting is for.

7.8 animation.cfg Export

Located in “File/Export/JA Ghoul 2 animation markers (.cfg)” or invoked by entering “Export JA Ghoul 2 Animation metadata (.cfg)” in the Search Window opened by default with Space.

GLA files only contain tons of frames. In order for the game to know which ones belong to which animation a separate file is used – the animation.cfg. To automatically create that, add a Time Marker in the Timeline (default short-cut: M) and rename it to the name of the animation starting there (default: CTRL+M), then use this operator.

Settings

Offset If you plan to merge your animation with an existing one, the animation starts will be off by the number of frames of the file you’ve merged yours into. That’s what this offset is for: It will be added to the frame numbers in the exported file.