Assimsoft Milkshape Plugins, "The Next Generation" ;)

Created by Chris Graham (gameRevolt) for Assimilation Software.

www.assimsoft.com

Introduction

The Milkshape Plugins are a set of open source plugins intended for working with models from and between various Star Trek games; they include a number of importers and exporters, and a simple "set length" tool to work with the common scale that these work at.

Full source code is given out along with this distribution so as to empower anyone with the skills required to be able to enhance my work. This distribution is released under the GNU Public License (see license.txt).

About Assimsoft

A group of gaming fans called 'Assimilation Software' have come together create free, open source 3D video-games and the technology behind them. We are developing our core 'assim engine' product and a trilogy of games in a universe based on the Star Trek TV show around it in an open environment where anyone with the initiative and skills to make things happen can become a member of our development team.

If you want to be a part of this, we are always looking for help... visit our website at http://www.assimsoft.com/. If you don't want to be a team member but still want to support our cause and our efforts, spread the world about us, consider making a donation to team members you feel are doing a good job (hardware, money, books, favours), or just hang out with us at our forums.

Personally speaking I've spent an enormous amount of time supporting other Trek games for free when I should have been working on our own, so if you appreciate this please reread that donation bit again ;).

Issues

Texturing

The exporters all ensure that textures are in the appropriate format for their game. There are two main kinds of texturing:

- SFC, where textures and lightmaps are in separate maps, of bmp (or pcx) format
- Armada / Bridge Commander, were textures and lightmaps are combined into a single tga file with an alpha channel

There is a special tool that allows you to also toggle between these formats from within milkshape, so you don't have to rely on the lightmap being mapped to the alphamap of milkshape making things look transparent (milkshape doesn't support separate lightmaps, so this is the hack that must be used). Also if you are working with SFC you may want to switch to the alphamap method if you prefer to work in that texture formatting scheme.
Whatever the situation, be very careful with your textures: they may be overwritten during conversions. Always make sure you are working on the correct texture files. Keep backups.

Hardpoints and nodes

The Starfleet Command and Armada plugins support this. Nodes are like a superset of hardpoints and are used for Armada. Both these use the Milkshape bones/joints/skeletal system to do things.

Joint viewing in milkshape can be turned on and off by the "show skeleton" option in the joints tab. To set the current joint being edited, click on it in the joints tab. To add new joints use the joint tool in the model tab and use the orthographic views to position it. New joints will become children of the currently selected joint... in other words, from a ST: Armada point of view, they are underneath it in the hierachy (the sfc importer doesn't care about joint hierachy). If you delete a joint (pressing the delete key over it whilst in the joints list), all children of it will be deleted.

As of R3, a hardpoint importer tool (available from tools menu) is provided that can import hardpoints from sfc models, nodes from armada model, or hardpoint dump files from assimview. Note: when importing a .pnt (assimview dump file), you can change the radius entry in it to load the file at different scales (open the file in notepad).

Scaling

All the plugins work to a common scale of 10 milkshape units to one metric metre, and compensate automatically when you import or export.

There is a special scaling tool that you can use to view and set the length of your ship, and sets the width and height automatically to be in proportion.

By convention, the models should face the camera. So when you load up a model you should be seeing its front.

Format Support

Armada (all editions)

Importing

Due to the concerns of the community, this is only available by moderated request. Please consult "Lord Apophis", "Phantom", or "Voyager009" if you would like it.. do not ask me, as I have delegated the responsibility to those who know community members better.

For those with the importer...

The importer will set up a skeletal structure and material naming conventions to make it export almost exactly as it was imported.
The importer could be used to convert A2 models to A1 models, as long as any A2-material-name-tags are removed. It could also be used for porting to other games.
The importer is quite dedicated in finding your textures so this shouldn't be an issue.
There is one problem that I have been unable to fix where for some unknown reason some meshes are flipped. I really haven't a clue as to how this could happen.

Actually there is a cut down importer in this release that is only able to load stock models. It is useful for seeing the format the exporter works at.

Exporting

There is a common exporter for both Armada I and II, where you select which game you want from the file-type dropdown menu when you save.

Make sure you only use uncompressed 24/32-bit TGA files for textures as thats what Armada I/II requires.

Changing the lighting/texturing

To force the lighting model for different materials to be other than the default "lambert" model, prefix "!lambert_", "!constant_" or "!phong_" to the name of the material. I myself am not certain on how the different models work, and how the settings affect them. I leave this as an experiment to you.

To force a different texturing type for meshes, put "_!" followed by the texturing type required, at the end of the material name. This is our equivalent of the specially named child points that the 3dsmax exporter requires for customisation.

Valid texturing types...

- default - default
- additive - Use additive blending
- translucent - Semi transparent
- alphathreshold - Use for objects using alpha channel 'cut outs'. Alpha channels will have hard edged 'threshold' but objects will be drawn quickly.
- alpha - Uses entire alpha channel. Object will require sorting, so will have performance implications.
- wireframe - Use wireframe graphics.
- opaque - (Armada 2 only) Use the alpha channel for glow, not transparency. This seems to have the same effect as "default" for Armada 2.
- noalpha - (Armada 2 only) A special one put in by me that sets the default texturing type but also turns off glow, so the alpha channel has no effect. Normally, If you have a texture without an alpha channel, Armada 2 will treat it as if it had a fully white alpha channel. Therefore, the texture would glow: but not when the texturing is set to "noalpha".

Ok, so like most you are probably confused by that. I will give you some examples of material names and their effect...

- !lambert_whatever_!noalpha - The material uses lambert lighting, and the noalpha texture type. (This will only work with Armada 2). The "noalpha" type works the same as the "default" type but the alpha channel is turned off.
- mynormaltexture - Nothing special here as no tags have been embedded in the name. It will use the default lighting model (=lambert) and the default texture type (=default).
- !phong_crazy - Will use the default texture type and the phong lighting model.
- crazyby_!additive - Will use the default lighting model and the additive texturing type.

I know this is complex, but its more than you need.. just use standard 32 bit textures with a black alpha channel, or a custom alpha channel, with all the defaults.

Node Hierachy

Node hierarchy is supported by the Milkshape skeletal system, and any nodes that have the same name as a mesh will tie in with the mesh in the hierarchy. The type of node a joint is exported as depends on its prefix...

- Sprite, s_ (where the rest of the name is the name Armada understands for the required sprite)
- Hardpoint or general node, h_
- Emitter, e_
- Mesh, m_
- LOD Control, l_

Actually to be more precise, a node named "m_mymesh" will link up to a mesh called "mymesh". If a linking cannot be established the exporter will give an error.

Linking will only be attempted if a skeletal structure exists. If it does not, an old style "arbitary structure" export will be done.

You do not need to worry about the skeletal structure affecting the positions or orientations of meshes, as the plugins compensate for this automatically.
As nodes in Armada have directions as well as positions, you will need to set them if you are considering things like hardpoints or sprites. If you look carefully you will see the currently selected joint has little coordinate axes drawn that show how it is pointing. The exporter will try its best to work with these but it might be necessary to use assimview (another assimsoft product) to get them exactly as wanted. This is because Milkshape works in angles, whilst Armada and Assimview work in vectors.. the convertion is messy.

Really the best thing to do to understand Armada models is to use the importer as a guide.. it imports so that it would be exported almost exactly as it originally was.

Hot tip: Make your original mesh without a node hierachy and export it, then use the hardpoint tool to import the default node hierachy that was created for it back in. You can then edit that hierachy further inside milkshape.

Birth of the Federation

To get to the hob model files, rename the largest .res file to a .zip file and unzip it as you would normally unzip a zip file.

Not all .hob files will load perfectly, but very little is known about this file format.

Elite Force 2

(Previously released as part of the eliteforce.com tools)

This will import .tan non-skeletal meshes from Elite Force 2, or skb skeletal meshes. It can also import ska animations for a specific skb mesh thats already loaded (pretty broken currently).

Because you are not importing a tik file, you will be asked what texturemap should be loaded for each surface in the file. To find out which one, find the corresponding .tik file for the model being loaded, open it in notepad, and you will see lines that show you which ones to use. Alternatively, if you put all the textures, tiki files and model files associated with whatever you are importing in the same folder, the importer is clever enough to assemble things together without bothering you.

DDS textures are automatically converted to tga files upon loading.

This is an early alpha release, and many things are broken.

There is also a uncompilable EF2 exporter in the source code that I have put on hold. It might be interesting for some.

Bridge Commander

Importing

The Bridge Commander ship import should be reliable, but the importer it only supports part of what the game is capable of. There is one remaining issue with the importing process: the meshes within models are not always aligned properly to each other. In other words, sometimes ship components may be rotated at strange angles or moved to odd positions. Importing bridges works to an extent, however it is particularly affected by the problem of mesh non-alignment. I hope that if someone finds a simple ship model that has the mesh non-alignment issue, then they can tell the community what model has the problem and the maths gurus out there can try and figure out what is wrong. The importer creates a 'structure.txt' file upon loading a model that contains the mathematical transformational information that must be employed. Currently the importer 'kind of works' by using this data to build up transformation matrices which are compounded together in a tree-like-way and then applied to vertex positions. If the solution is not in this data then nif specs are included with this pack. Most people won't know what I am on about and for most people this is not going to be a big issue!

On importing a model with textures that are stored internally, they are converted to external .nif textures. External .nif textures are converted to .tga files.

Textures are searched for in the model directory, and in a directory 'high'.

Exporting

I am informed this works as it should do.

Homeworld (all editions)

This is a basic importer that should work acceptably well.

Star Fleet Command (all editions)

If there is a lightmap, it is placed as the "alphamap" for milkshape. This will make the mesh pretty much invisible in milkshape, but when exported the alphamap will be used as a lightmap for SFC. You will also need to increase the emittance in order to make the texture visible.

Exporting

Any bones in the Milkshape model will be exported as SFC hardpoints.

Make sure you only use BMP/PCX files as that's what SFC needs.

Star Trek Invasion

This is to import models from this Playstation game. First they must be extracted from the archive using my special toolkit...

http://www.assimsoft.com/incoming/small-releases/trk.zip

There are loads of tools here allowing you to view videos, art and so on. This isn't easy to use, and I don't really support it -- if you study hard enough you'll work it out in the end ;)

The models aren't reassembled properly so it is up to you to get their mesh components realigned and get rid of lods you don't want.

Also texturing is very patchy indeed. This is really just an experimental importer that others might wish to play with and/or improve.

Klingon Academy

Klingon Academy is not directly supported, however Milkshape does come with lwo plugins that work to a limited success. I hope to have an improved importer released at some stage that works better, supports KA directory structure, ad supports LWS files transparently (so you only import the LWS file). I've talked to the creator of the current Milkshape LWO plugin and he seems enthusiastic in implementing the improvements necessary, so it might be done without too much of my help :D.

Converting between games

The plugins are tuned to be able to convert almost seamlessly between Bridge Commander, Armada and Starfleet Command. The texturing differences are bridged with the automatic texture convertion feature, and between Armada and SFC there is even almost compatible hardpointing. Scale is unified between all plugins. Some lighting tweaks will need to be done, as the games support different lighting models.

Changes

Things have changed substantially from the initial set of plugins. These are the major changes...

- hardpointing for sfc and armada!
- new lightmap flip tool
- fixed bug in bc exporter that meant texture filenames were messed up
- fixed sfc import bug where "emissive" was not set right (making all black)
- merged armada exporters to a single plugin
- automatic texture conversion
- new docs
- scaling tool
- nif importer
- sod importer
- homeworld importer
- botf importer
- star trek invasion importer
- fixed problem with sfc importer texturing when shared vertices had different texturing

Contact Details

If you have any questions or just want to chat, visit us at...

http://www.assimsoft.com/

Or go directly to our forums...

http://www.assimsoft.com/forums/

Or contact me (I much prefer forum posts to personal emails as they can't be forgotten as easily, and others can answer them)...

gameRevolt@assimsoft.com

If you have problem meshes, please upload them to the root directory of http://assimsoft.com/uploads.php and contact us.