Neverblender Documentation

December 17, 2016

Contents

1 Installation

The only requirement is Blender 2.70 or newer. You can install neverblender like every other add-on from blenders User Preferences under the Add-Ons tab. Select Install from file at the bottom and select the downloaded zip file.

Alternatively you can install manually by unzipping the contents of the downloaded zip and placing the neverblender folder into the scripts/addons directory of your Blender directory.

2 Import & Export

2.1 Import

Import Geometry Import the geometry from the mdl file.

Import Walkmesh Attempts to import a walkmesh. If the imported model is a placeable, the script will look for a **.pwk* file in the same folder. If the model is a door, it will look for a **.dwk* file. If the model is a tile, it will read the walkmesh directly from the **.mdl* file.

Import Smooth Groups Import smooth groups as sharp edges.

Import Animations Import animation from the mdl file. Animations are added to the imported geometry. If no geometry has benn imported, the script will try to add animations to already existing objects in blender.

Materials None = No materials or textures will be imported. Single = Neverblender will attempt to merge similar materials to reduce clutter. Multiple = Each object will get it's own material, even if this results in multiple identical materials.

Image Search Search for textures in subdirectories. This might take a significant amount of time depending on the number of files.

2.2 Export

Export Options

Export Animations ported regardless of this setting.

Export Walkmesh Attempts to export a walkmesh. The type of exported walkmeshes depends on the objects classification.

Export Smooth Groups Convert sharp edges to smooth groups.

Apply Modifiers Apply Modifiers before exporting.

3 Editing

There are three types of objects which can be exported as mdl: Empties, Meshes and Lamps. Other objects like curves and surfaces will have to be converted to meshes before attempting to export them.

3.1 Empties

Empties appear in mdl files as Dummies. Like in blender they are used to group objects. In addition there are special types of Dummies which for example indicate locations for spells effects.

3.1.1 Dummy

Simple Empties/Dummies with a parent are either used to group objects or indicate a special location for the engine. If a Dummy doesn't have a parent it is considered to be a *Rootdummy* instead of a simple Dummy and the available options will change.

The following types of Dummies are available:

None Simple dummy without any special purpose.

Ground Indicates the ground level for spells/ visual effects

Impact Impact target for most spells/ visual effects.

Head Hit Spells/ visual effects

Head Head location for Spells/ visual effects

Hand Point of origin for (some) spells. If the placeable is using a spell/ visual effect this will be the point of origin most of the time.

Use 1 Use Node for placeables or doors. Upon receiving a use command a character will move to the closest of the two use nodes.

Use 2 Use Node for placeables or doors. Upon receiving a use command a character will move to the closest of the two use nodes.

They are not mandatory. If a particular Dummy is missing from the mdl, the engine will use (0,0,0) as the default location for that Dummy.

Selecting the dummytype is not enough to make them work, they need to have the correct name or to be precise the correct suffix. The export script will try to generate a valid name depending on the Dummys type, but it is recommended to do it manually to avoid naming conflicts.

You can use the button next to the Dummytype selector to generate a working name. It will take the dummy's name and append the needed suffix for the selected Dummytype.

3.1.2 Rootdummy

Each mdl requires at least one Empty: The *Rootdummy*. All children of a *Rootdummy* are considered to be part of the same mdl. All objects which are to be exported have to be a descendant of the *Rootdummy*.

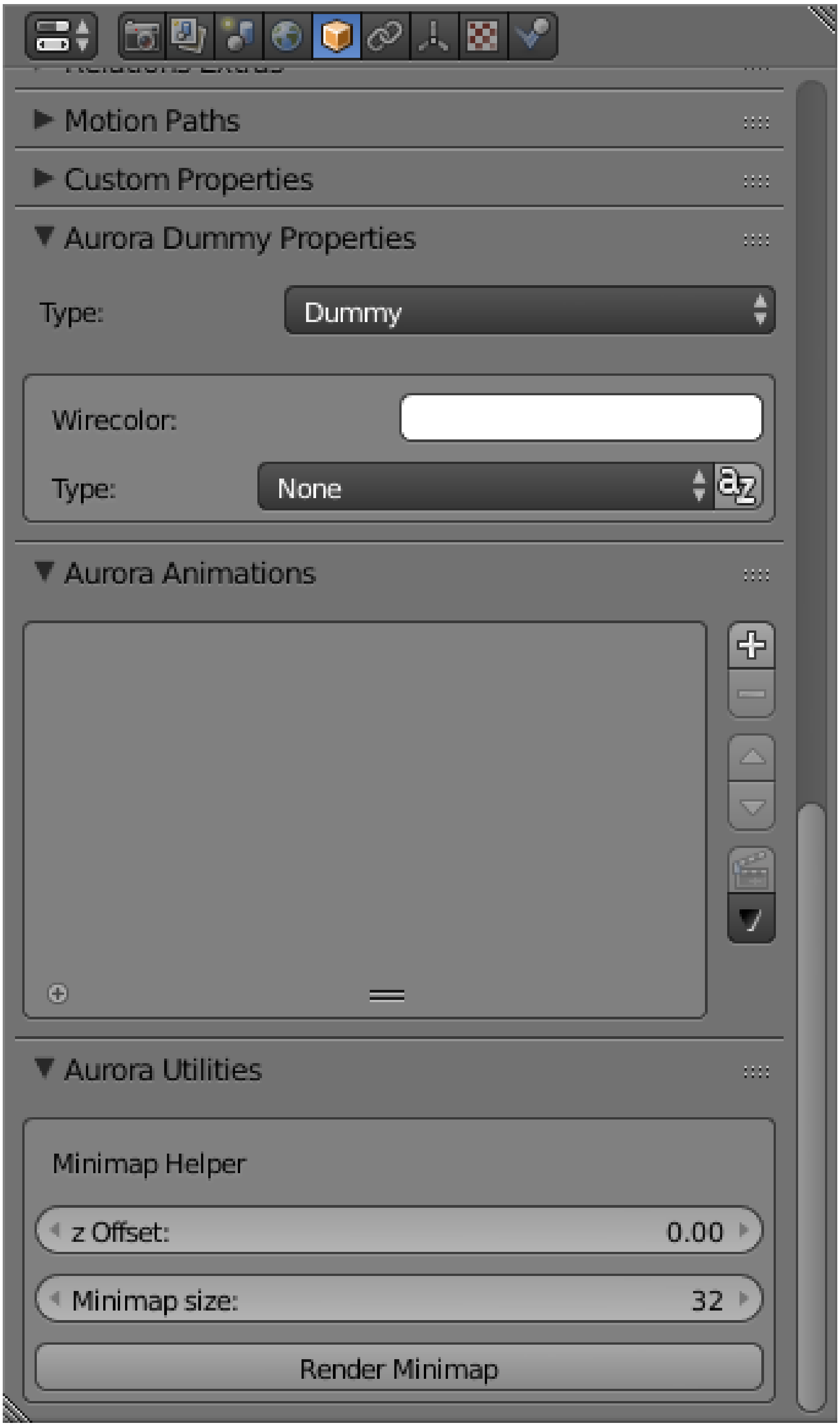
An Empty with the type Dummy and without a parent will autmatically be considered a Rootummy. The *Rootdummy* must have the same name as the mdl file (minus the files extension). Rootdummys hold additional information about the model and the Aurora Property panel will change accordingly.

Classification TODO

Supermodel References another mdl file. All animations will be available in this mdl.

Animationscale Unknown

It is also used to store the beginning and end frames of specific animations (walk, run, sit,. . .). It is not necessary to select the Rootdummy to access these values, selecting a child of the Rootdummy will do.



Type: Dummy

Wirecolor:

Type: None

z Offset: 0.00

Minimap size: 32

Figure 1: Dummy Property Panel

3.1.3 Reference Node

The purpose and usage of these nodes is unknown. Supposedly it can be used to reference other mdl files. Reference nodes can be found in some spells and most of the time references fx_ref.

3.1.4 Patch Node

The purpose and usage of these nodes is unknown. They occur in some spells, but they seem to behave excatly like normal dummy nodes, i.e they have the same attributes.

3.2 Meshes

3.2.1 Trimeshes

Wirecolor This is unused in-game. You can safely ignore it. This probably defines the colour for the object's wireframe in 3dsmax.

Self-illumination color Makes the mesh seem to glow. It does not act as a light source however. This will not be visible in blender's viewports or renderers.

Ambient Color OpenGl material property. The Aurora Engine uses per object ambient light, which you can define here. Blenders ambient light is defined on a per scene basis. Therefore this property not be visible in blender's viewports or renderers.

Shininess A matching .txi file for the texture and a .env file are necessary

Tilefade Only available for tiles. Controls wether this mesh will turn invisible to clear the view.

None This mesh will always be visible.

Fade The object will fade.

Base Unknown

Neighbour The object will fade along with meshes in neighbouring tiles.

Render Controls wether this objects should be rendered in-game. Meshes can still cast a shadow without being rendered.

Shadow Controls wether this objects should cast a shadow.

Beaming Probably unused.

Inherit Color Probably unused.

Rotatetexture Only available for tiles. Auto-Rotates textures so the uvs are rotated the same way to avoids seams between tiles.

Transparencyhint Helps the engine to prioritize transparent meshes, similar to a z-buffer. If you have multiple meshes with transparent textures and have issues like flickering try changing this value.

Smoothgroup Controls how or wether at all to create smoothgroups or shading groups.

Separate Each face will have its own smoothgroup. This results in no smoothing at all.

Single All faces belong to a single smoothgroup. Meshes will be smoothed.

Auto Auto generates smoothgroups, depending on the settings in blender and edges marked as sharp. Replicates blenders smoothing as closely as possible.

3.2.2 Danglymeshes

Danglymeshes are Trimeshes which are"bouncy" or "dangly". They are affected by wind, character movement and spells. A danglymesh has a set of weights which determine how far a vertex can be displaced from its original position.

A Danglymesh has all properties of a trimesh in addition to the following properties:

Dangle group The dangle group is a vertex group containing the weights of vertices for the danglymesh. You must select an existing vertex group.

Period

Tightness

Displacement

3.2.3 Skinmeshes

A Skinmesh has all properties of a trimesh in addition to the following properties:

Skinmesh description

List properties

3.2.4 Animeshes

Animmeshes are Trimeshes with animated UV textures. At the moment Animeshes will be imported as Trimeshes and thus lose their animations.

An Animesh has all properties of a trimesh in addition to the following properties:

List properties

3.2.5 Walkmeshes

Walkmeshes indicate where a character can walk. The type depends on the type of mdl you want to create.

You will need to select the appropriate type of the walkmesh.

Tileset Walkmesh for tiles. Assign Materials to each face to set the surface type of tile (stone, grass, unwalkable, ...). This will also affect footstep sounds and grass growth.

Door: Closed The walkmesh for the closed state of the door. Blocks movement.

Door: Open 1 The Walkmesh for the first open state of the door. Blocks movement.

Door: Open 2 The Walkmesh for the second open state of the door. Blocks movement.

Placeable Walkmesh for placeables. Blocks movement.

3.3 Emitters

Support for Emitters is rudimentary. Blenders particle systems is too different to allow a direct import, instead Emitters are imported as plain text. While all data will be retained, editing is difficult.

3.4 Lamps

Blender lamp poperties will be used for exporting a light. Exported properties are:

- Color
- Distance

Light type

There are some special light types, all of which are used to give builders the ability to select light color in the Toolset.

Mainlight 1 For Tiles only. Color can be changed in the Toolset.

Mainlight 2 For Tiles only. Color can be changed in the Toolset.

Sourcelight 1 For Tiles only. This is acually a burning flame. Color can be changed in the Toolset.

Sourcelight 2 For Tiles only. This is acually a burning flame. Color can be changed in the Toolset.

Default Default type, can always be used. Blenders lamp properties will be used (color and distance)

Wirecolor

This is unused in-game. You can safely ignore it. This probably defines the color for the object's wireframe in 3dsmax.

Priority

Unknown. Might control when the light source casts a shadow.

Ambient Only

This controls if the light is only an ambient light source or if it is directional as well.

Fading

Unknown. Might activate some kind of distance fall off for the light.

Shadows

Determines if this light is capable of casting shadows.

Is Dynamic

Unknown.

Affect Dynamic

This controls whether this light affects dynamic objects, i.e. characters. Disabling this will prevent this light from casting shadows with dynamic objects, but it will in turn improve performance.

This is basically a less strict version of the *Shadows* setting.

Lensflares

Add a series of lensflares for this light.

Lensflares are unavailable for Mainlights or Sourcelights.

Texture Texture for this flare

Colorshift Unknown. Possibly color difference from the light color.

Size Size of the flares. This will scale the texture.

Position Distance from the lights origin.

4 Animations

Write something meaningful

5 Mdl types

5.1 Creatures

TODO

5.2 Doors

Placeable should have at least the following dummies:

Use1

Use2

Impact

They will ensure it can be used properly and spell/effects hit the right spot.

Possible Animations

closed Closed state of the door.

open1 First open state of the door. Doors in NWN can open/swing to both sides, depending from which side a player is opening it. Open 1 and Open 2 can be the same.

opening1 Transition from closed to open1

closing1 Transition from open1 to closed.

open2 Second open state of the door. Doors in NWN can open/swing to both sides, depending from which side a player is opening it. Open 1 and Open 2 can be the same.

opening2 Transition from closed to open2.

closing2 Transition from open2 to closed.

trans Unknown

die Animation when the placeable is destroyed. Most placeables omit this alltogether or sink into the ground.

dead Dead state. Most placeables omit this alltogether or have a single positionkey below ground level.

5.3 Effects

TODO

5.4 Placeables

Possible Animations

default Default state. It is recommended to add this if there are other animations present. If this is omitted and other animations are present a placeable can't be set to static in the toolset (else the game will crash)

on

on2off Transition from on to off state.

off

off2on Transition from off to on state.

close Closed state.

close2open Transition from closed to open state.

open Open state.

open2close Transition from open to closed state.

damage Animation when the placeable is damaged. Most placeables omit this animation or willshake for 0.1 seconds.

die Animation when the placeable is destroyed. Most placeables omit this alltogether or sink into the ground.

dead Dead state. Most placeables omit this alltogether or have a single positionkey below ground level.

5.5 Tiles

Tile Walkmesh (AABB)

The walkmesh must be parented to the the *Rootdummy* object. No overlapping faces are allowed. To specify the surface type, it is necessary to add the walkmesh materials to the object. You can add all walkmesh materials by clicking on the *Load walkmesh materials* Button in the Aurora mesh properties panel.

The walkmesh materials are then added as materials slots and accessible through the materials tab in the object properties.

The export script will create an aabb tree from the walkmesh during export and add it to the **.mdl* file. It will also create a matching **.wok* file.