

Animating a Cabinet for NWN2

Robert J. Hall

This tutorial will demonstrate how to animate a placeable for Neverwinter Nights 2 (NWN2) using Arbos' nwn2mdk project in the Windows OS. built as separate models so that their axis of rotation—the position of its hinges—is located at the origin.

Introduction

The NWN2 game engine implements animation through the Granny Software library. This package stores the information needed to perform specific animations of the part models in files with a '.gr2' suffix. These can be found under the lod-merged folders under Data where the NWN2 software is installed. The file names begin with the name of the MDB model to be animated, followed by the animation type. When no animation is specified, the file contains 'skeleton' information about the model.

Typical names of placeable animations include 'idle', 'open', 'plc_openidle', and 'close'. These files are used, for example, when calling the ActionPlayAnimation function to make a placeable animate.

The animation of the placeable is made possible via the nwn2mdk package, written by Arbos. This tool takes advantage of Blender's built-in Import/Export capability for FBX format files, which preserves the animation information. Thus, NWN2 MDB files accompanied with their associated Granny files, must be converted to FBX format for import to Blender using nwn2mdk. The model is later exported in FBX format then converted back to MDB files, along with a set of Granny files containing the animation information, using nwn2mdk.

For the purposes of this tutorial, the model only needs a skeleton file plus two animation files: the first for idling while the doors are closed and a second idle animation, tagged plc_idleopen, while the doors are swung open. Activating the model in game will cause the engine to interpolate between these two states when opening or closing the doors. For this to work properly, the doors must be

Software Requirements

I used the following free software packages for this project:

- [nwn2mdk](#) – version 0.4
- Blender – version 2.79
- [Blender MDB Import/Export Plugin](#)
- GIMP – version 2.8
- GIMP Normalmap plugin
- GIMP DDS plugin

The plug-ins are loaded into their respective software.

In order to perform this procedure, you will need a working knowledge of how to build and work with models in Blender. Expect it to take several weeks of practice to gain that level of familiarity.

Preparation

The following preparation steps are covered by tutorials available elsewhere, so I will cover them only briefly and note the steps I used.

For this tutorial, I wanted to create a cabinet with doors that could open and later close when activated by the PC. Hence, the cabinet needed to have both exterior and interior mesh surfaces. I chose the "[Metal office file cabinet set](#)" from CadNav, then trimmed out the parts I didn't need. Since the inside of the cabinet will be visible when the doors are open, I added a set of shelves to the interior.

In order to make it presentable, I revamped the model's UV mapping to a texture atlas format and gave it a custom texture set with normal and tint maps. The sections of the map are organized into the cabinet sides and two door packets. To keep the quality high, the UV maps for the two doors

and the four shelves were combined to re-use the same two sections of the texture map.

When the model was complete, I tested it in game to make sure the scale was right. The door handles proved to be too large and high off the ground compared to a human character, so I shrank the model down to 60% of the default size. Finally I added a C2 collision mesh and a walk mesh—the model shape is fairly simple so it doesn't need a C3 collision mesh.

The illustration below shows a textured 3D view of the cabinet in Blender. It's a very vanilla model that should be familiar to many office workers.

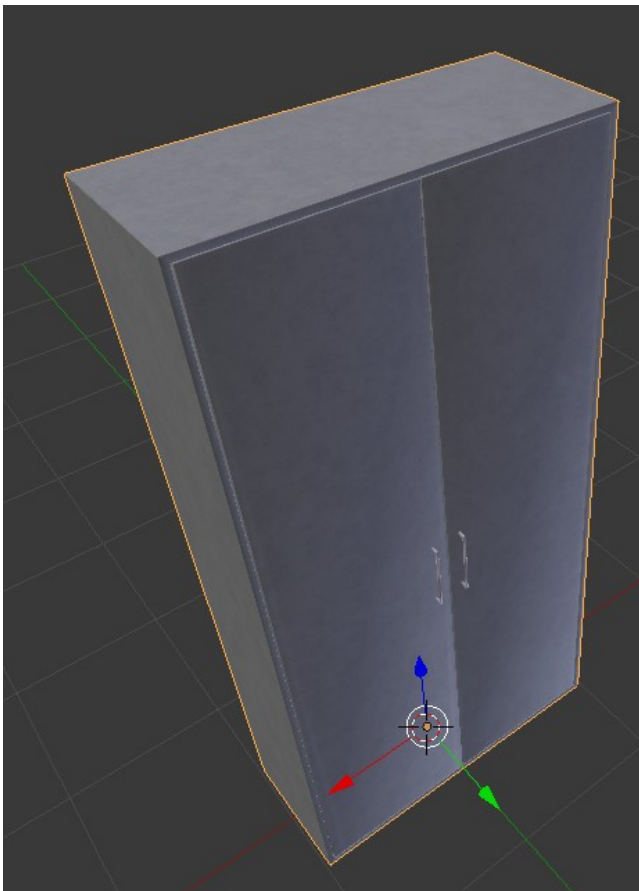


Fig. 1 – A view of the base cabinet model in Blender's 3D View.

Idle Animation

To implement the cabinet animation, the model is divided into three separate model files; one for the static part and two for the doors. The doors are going to pivot about their hinges, so they need to be positioned so that the outer pivoting edge is at the origin (coordinates 0, 0, 0). When the parts are

merged together, this puts the doors out of their correct alignment position with respect to the cabinet frame. However, this will be fixed by the animations.

The cabinet now consists of the following model parts:

- plc_md_cabinet05.mdb
- plc_md_cabinet05_01.mdb
- plc_md_cabinet05_02.mdb

Note how the latter two parts, which are the doors of the cabinet, have an underscore number at the end.

I followed steps 1-3 in the nwn2mdk wiki tutorial titled 'Tutorial: Animated Placeable' as follows:

- Drag-and-drop all three model files onto the nw2fbx.exe icon. This will create a merged file called 'plc_md_cabinet05.fbx' in the same folder.
- Open Blender, select the File menu, then click on Import and the FBX (.fbx) sub-menu. This will open the FBX import interface.
- Select the file 'plc_md_cabinet05.fbx' and click on the 'Import FBX' button. The cabinet model is now visible in the 3D view.
- In the 3D view, add a Plain Axis by typing SHIFT+a, then pick on Empty, Plain Axes. This will create an X/Y/Z axis at the origin.
- In the Outliner, rename the Empty object to the model name plus '.PIVOT'; hence 'plc_md_cabinet05.PIVOT'.
- In the 3D view, select the door frame mesh part 'plc_md_cabinet05' by right-clicking on the entry in the Outliner.
- In the 3D view, hold down the SHIFT key and right-click on the Plain Axis. This must be the last part selected.
- Type Ctrl-P and select the Object menu pick.

In the Outliner, the door frame mesh has been

under the `plc_md_cabinet05.PIVOT` hierarchy. Clicking on the '+' button will expand the view and show the child part.

The two door parts need to be made children under the door frame parent. In the 3D view, I right-click on a door, then SHIFT+right-click on the cabinet frame. With the parts selected in that order, I type Ctrl-P and choose 'Object'. The same process is repeated for the other door part.

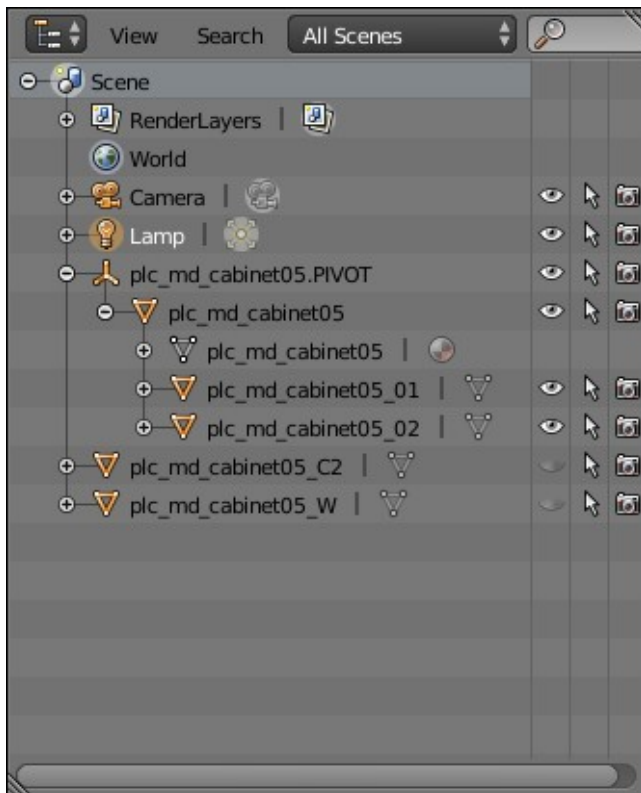


Fig. 2 – The Outliner shows the two door parts, `_01` and `_02`, positioned as children under the door frame parent. The door frame in turn is a child of the axis. The C2 mesh and the Walk mesh are kept separate.

At this point, I saved the model to a '.blend' file in case I need to start over.

The model is now ready to animate. To do so, I need to open the Timeline in one of the panels – in my case I swapped the UV Image Editor for the Timeline. This view represents the flow of time for the animation, with 24 frames per second of animation.

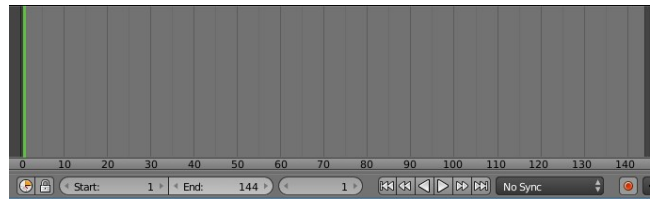


Fig. 3 – Sample timeline with the end point set to 144 frames; equivalent to 6 seconds of play.

Important: During testing, it was found that static animation behaved incorrectly if it had more than one frame. For this reason, the End point of the idle animation timelines is set to 1.

The first animation is for the IDLE mode with both doors closed. In the 3D View, the two door parts must be positioned in their closed position relative to the cabinet frame. This involved a series of shifting movements of the door parts, with closeup views being used to line them up correctly.

Important: The axis of rotation for each door part needs to be maintained at the same position relative to the mesh. For this reason, I only perform movement of the parts in Object Mode. Movements performed in Edit Mode will relocate the origin, resulting in incorrect animation.

With both door parts in place, I set the current frame to 1, move the cursor to the 3D view, select all components listed under `.PIVOT`, and click 'i' to insert a key frame. Since nothing is being moved for the animation, I just select 'Location'. The timeline will now show a frame at the start.

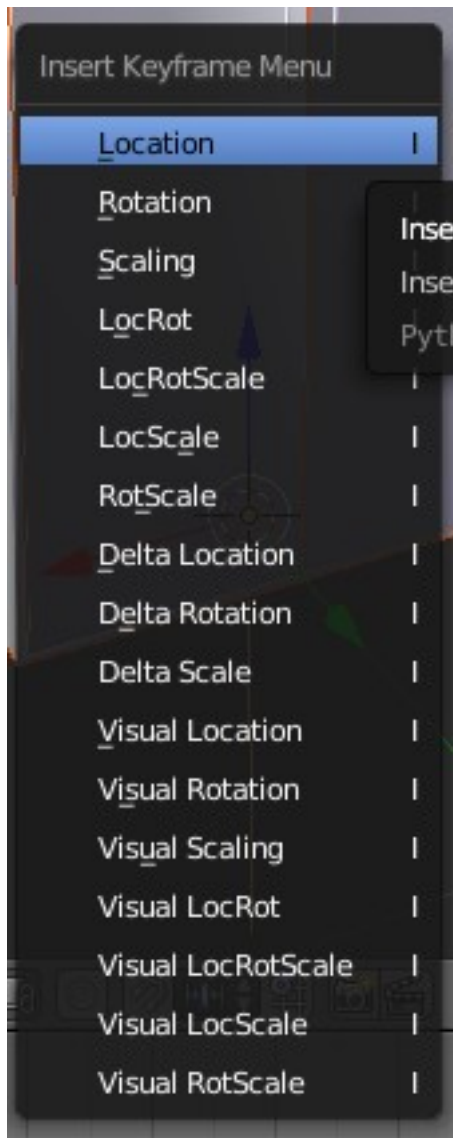


Fig. 4 – Insert Keyframe Menu

With the frame inserted, the timeline looks like the illustration below.

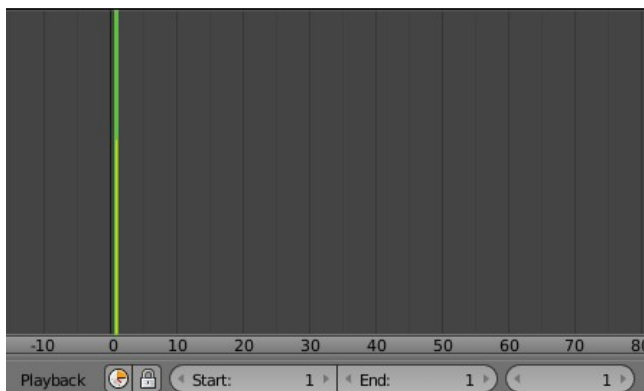


Fig. 5 – Completed idle animation timeline

It can play the animation now by clicking the Play Animation button, but it will do nothing. I

stopped the animation and returned to the first frame.

With the animation complete, the model can be exported to an FBX format file. I select the File menu, choose Export, then pick the FBX (.fbx) submenu.

The interface that appears has an 'Export FBX' panel in the lower left. Under the Geometries tab, I turn on the 'Tangent Space' check box. I click on the Animation tab and turn off everything except 'Baked Animation'. In addition, since there was no actual change in the model during the animation, the 'Simplify' field must be set to 0.00.

After double-checking the above, I use the browser panel at right to export the model to 'plc_md_cabinet05_idle.fbx'.

Open Idle Animation

The next step is to create the open idle animation. This follows the same process as the initial animation but with the doors in the open position. For this purpose, I need to rotate the two doors around the Z axis so that they are positioned at an angle to the hinges.

First, I need to clear the existing animation. In the 3D view, I right-click the model parts, press the space bar, type 'Remove Animation' in the input field, then enter. This will remove the frames from the Playlist panel.

For the left door, while remaining in Object Mode, I rotate it -110° about the Z-axis and position the inner edge near where the hinges would be located, making sure it just clears the cabinet sides. For the right panel, I do the same procedure with a $+110^\circ$ rotation. I check the positioning of each door in Edit Mode by selecting all faces, clicking 'n', and viewing the Transform coordinates in Global mode. The coordinates should roughly match, with the right door having the negative x-value of the left door. To change them, I first return to Object Mode.

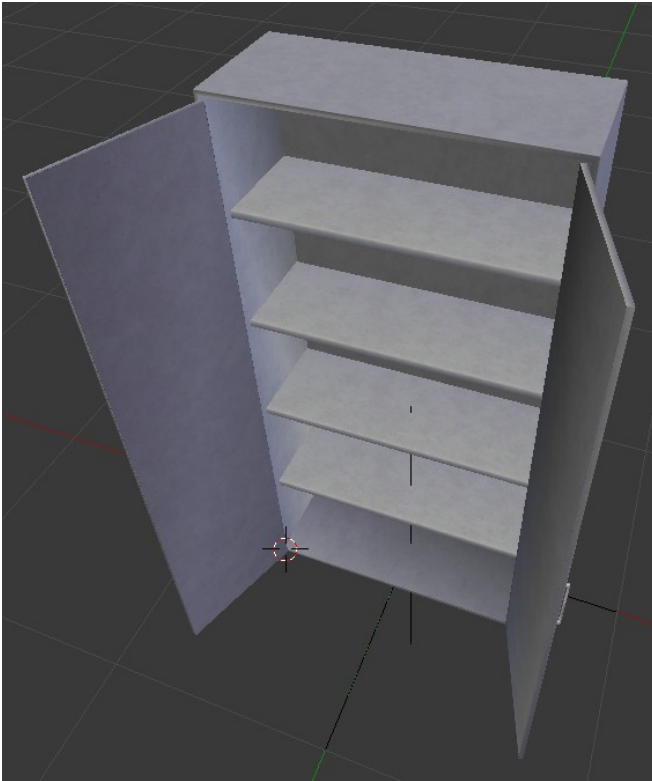


Fig. 6 – Blender model with doors open

The single-frame OPEN IDLE animation can now be set with the Playlist panel in the same way as the IDLE animation. The completed animation is now exported to the FBX format file named 'plc_md_cabinet05_plc_openidle.fbx'. Make sure to use the same Geometries and Animation tab settings that were used for the IDLE animation export.

Convert to MDB

The completed FBX files are in my folder, so I can individually drag-and-drop them onto the 'fbx2nw.exe' icon. This will create six new files:

- plc_md_cabinet05_idle.mdb
- plc_md_cabinet05_idle.skel.gr2
- plc_md_cabinet05_idle.anim.gr2
- plc_md_cabinet05_plc_openidle.mdb
- plc_md_cabinet05_plc_openidle.skel.gr2
- plc_md_cabinet05_plc_openidle.anim.gr2

The 'skel' files contain the model's skeleton in Granny file format. I only need to use one of these, so I rename 'plc_md_cabinet05_idle.skel.gr2' to:

- plc_md_cabinet05.gr2

The two 'anim' Granny files contain the door animations. For compatibility with NWN2, I rename these as follows:

- plc_md_cabinet05_idle.gr2
- plc_md_cabinet05_plc_openidle.gr2

The animation process is now complete.

Test in NWN2

The final step is to test the animation in the toolset. I modified my 'placeables.2da' file, using the row 631 PLC_MC_ARMY01 entry as a template. The fields for the new animated cabinet are set as follows:

LABEL	<i>MD_Cabinet05</i>
StrRef	****
PlaceableModelType	2
NWN2_ModelName	<i>plc_md_cabinet05</i>
NWN2_Skeleton	<i>plc_md_cabinet05</i>
AttachedModelName	<i>plc_md_cabinet05_??</i>
SoundAppType	61

The Attached Model Name is a "wild card" setting that will tell the game engine to attach the two door models. The 'SoundAppType' field is a row number in the 'placeableobjsnds.2da' file.

I copied the following files to my Override directory:

- plc_md_cabinet05.mdb
- plc_md_cabinet05_01.mdb
- plc_md_cabinet05_02.mdb
- plc_md_cabinet05.dds
- plc_md_cabinet05_n.dds
- plc_md_cabinet05_t.dds
- plc_md_cabinet05.gr2
- plc_md_cabinet05_idle.gr2
- plc_md_cabinet05_plc_openidle.gr2
- placeables.2da

In the NWN2 Toolset, I can now create a blueprint for the cabinet and test it in an area. If the cabinet doesn't show up, check that there are no typo

errors in the 'placeables.2da' entry.



Appendix A. Animating Doors

Doors can be animated using a technique similar to the method used to animate placeables. The sample metal grille door I wanted to build consists of two moving doors that opened by sliding to the sides. This requires three separate parts: a small base model to serve as a parent, and separate models for the left and right doors. The three models were named as follows:

- `plc_du_gate01.mdb`
- `plc_du_gate01_01.mdb`
- `plc_du_gate01_02.mdb`

The stub model '`plc_du_gate01.mdb`' consisted of a simple, square body mesh, which would be hidden in the ground, and a C2 collision mesh that would enclose the two doors in their idle state.

Within Blender, the left and right door parts are set up as children of the fixed stub model under a `plc_du_gate01.PIVOT` axis. Next, I generated the

following GR2 files using the technique described earlier:

- `plc_du_gate01.gr2`
- `plc_du_gate01_idle.gr2`
- `plc_du_gate01_dr_open1.gr1`
- `plc_du_gate01_dr_open2.gr2`

The first file contains the door skeleton, while the second is the idle animation with both doors closed. The 'open1' and 'open2' files are for the two possible open positions, depending on from which side the door is opened. Since the doors slid to the sides rather than swinging open, I made the 'open2' animation a clone of the 'open1' animation.

The completed door is configured for game use in the '`doortypes.2da`' file. I used the following fields for this door:

LABEL	Grate_Gate
PlaceableModelType	2
NWN2_ModelName	PLC_DU_GATE01
AttachedModelName	PLC_DU_GATE01_??
NWN2_Skeleton	PLC_DU_GATE01
StringRefGame	GrateGate
BlockSight	0
VisibleModel	1
SoundAppType	10
OpenTime	2

The 'AttachedModelName' setting is a 'Wild Card' string that includes the left and right door model parts. Since the grille door can be seen through, the 'BlockSight' field is set to zero. The 'SoundAppType' gives the row number in the 'placeableobjssnds.2da' file. Finally, the 'OpenTime' is set to two seconds.



The End