SOLDIER OF FORTUNE II

DOUBLE HELIX

ADDING A WEAPON OVERVIEW

BY RAVEN SOFTWARE

1.0 Adding a weapon to Single Player

1.1 Create a model

Model, skin, and animate a weapon in the software package of your choice. You'll eventually need the animations to be in separate .xsi files, so you'll probably want to use SoftImage, or Max in conjunction with our xsi exporter.

1.2 Create a .car file

With all of the xsi animation files in one folder, create a .car file that will allow the Carcass utility to turn your animations into a Ghoul2 model. Here's a sample .car file that would work with the M4 rifle:

```
$aseanimgrabinit
$aseanimgrab models/weapons/m4/m4menuspin.xsi
$aseanimgrab models/weapons/m4/m4altfire.xsi
$aseanimgrab models/weapons/m4/m4altreload.xsi
$aseanimgrab models/weapons/m4/m4crawl.xsi
$aseanimgrab models/weapons/m4/m4done.xsi
$aseanimgrab models/weapons/m4/m4fingeradjust.xsi
$aseanimgrab models/weapons/m4/m4fingerknuckles.xsi
$aseanimgrab models/weapons/m4/m4fire.xsi
$aseanimgrab models/weapons/m4/m4fire2.xsi
$aseanimgrab models/weapons/m4/m4fire3.xsi
$aseanimgrab models/weapons/m4/m4fire4.xsi
$aseanimgrab models/weapons/m4/m4idle.xsi
$aseanimgrab models/weapons/m4/m4pronetostand.xsi
$aseanimgrab models/weapons/m4/m4ready.xsi
$aseanimgrab models/weapons/m4/m4reload.xsi
$aseanimgrab models/weapons/m4/m4dryfire.xsi
$aseanimgrab models/weapons/m4/m4standtoprone.xsi
$aseanimgrab models/weapons/m4/m4proneready.xsi
$aseanimgrab models/weapons/m4/m4proneidle.xsi
$aseanimgrab models/weapons/m4/m4pronefire.xsi
$aseanimgrab models/weapons/m4/m4pronedone.xsi
$aseanimgrab models/weapons/m4/m4pronereload.xsi
$aseanimgrabfinalize
$aseanimconvertmdx_noask models/weapons/m4/m4base -makeskel
skeletons/weapons/m4/m4
```

A couple of important things to note about this .car file are that the last line dictates that the model will use models/weapons/m4/m4base.xsi as the base for the model and that Carcass will create the animation file for this model in skeletons/weapons/m4/m4.gla. When Carcass is run it will look for a folder named "base" in the current path and use that as a foundation for locating the .xsi files and for determining the destination of the .gla file. In our example we're doing all of this in C:\Program Files\Soldier of Fortune II - Double Helix\base so Carcass will expect the .xsi files to be in C:\Program Files\Soldier of Fortune II - Double Helix\base\models\weapons\m4 and it will create the animation file in C:\Program Files\Soldier of Fortune II – Double Helix\base\skeletons\weapons\m4\m4.gla.

1.3 Carcass your model

You'll need to use the Carcass utility to turn your animations into a Ghoul2 model and Carcass requires a folder named "base" somewhere in its path. In other words, you can't run Carcass from, say, c:\models\weapons\m4 but you can run it from c:\base\models\weapons\m4. When SOF2 is installed on your machine (the default location is C:\Program Files\Soldier of Fortune II - Double Helix) there will be a base folder created there…I suggest doing all of your work in there just to get up and running.

So let's assume for now you put your M4.car file in C:\Program Files\Soldier of Fortune II - Double Helix\base\models\weapons\m4 and you put the Carcass utility in C:\Program Files\Soldier of Fortune II - Double Helix\base. Run Carcass on this .car file by opening up a DOS window, going to C:\Program Files\Soldier of Fortune II - Double Helix\base\models\weapons\m4, and typing the "C:\Program Files\Soldier of Fortune II - Double Helix\base\carcass m4" command. Or, just put a .bat file in the same folder as the .car file and doubleclick it. Here are the contents of a sample called CarcassThis.bat (you can name this file whatever you want).

This assumes that the xsi animation files are in the C:\Program Files\Soldier of Fortune II - Double Helix\base\models\weapons\m4 folder, too.

```
c:
cd c:\Program Files\Soldier of Fortune II - Double Helix\base\models\weapons\m4
c:\Program Files\Soldier of Fortune II - Double Helix\base \carcass -forcebuild m4 >
carcassResults.txt
```

You don't have to add the "> carcassResults.txt" bit at the end, but if you're running from a .bat file and some error occurs during the Carcass process the window in which it's running will close before you get a chance to see the error message. The "> carcassResults.txt" means all output from the process will be entered into a file called carcassResults.txt for you to peruse at your leisure.

1.4 Final output (.glm, .gla, .frames)

Assuming the Carcass process worked properly you now have a Ghoul2 model (C:\Program Files\Soldier of Fortune II - Double Helix\base\models\weapons\m4\m4.glm) and its corresponding animation file (C:\Program Files\Soldier of Fortune II - Double Helix\base\skeletons\weapons\m4\m4.gla).

Carcass also spits out another very important file as a byproduct of creating the model… m4.frames. It will be created in the same folder as the .gla file. Make note of this file because you'll want to open it later on and add certain things to it.

A. Heck, let's do it now! Open your new m4.frames file in a text editor like Notepad. You can also open one of the original .frames files (the one for the M4, for instance, is in the same pk3 file as sof2.inview and is called m4.frames) as a reference. What you want to do in here is add the notetracks necessary to make certain things happen during certain weapon animations, like actually firing a bullet during a firing animation. For more information on notetracks see SoF2_ Weapons_FramesFile.doc. The most important notetrack you need to add to get your new weapon up and running is the **fire** notetrack. Look in the original m4.frames and see where the **fire** notetrack is. Copy that into your new .frames file, changing whatever frame numbers may be necessary since your weapon animations aren't identical to those in the original M4. At this point you can feel free to add whatever other notetracks you like (such as sounds).

B. Now you'll need to tell the game what to do with all of these animations. Assuming you've built some sort of rifle similar to the M4 you can peek into the sof2.inview file and copy the definition for the M4. Name your new definition something other than "M4", though…as an example, let's say your new weapon is an "AR15". You'll want to change some things since your weapon probably won't have any bolted-on models like the M4 does, but you can see how the animations you listed in your .car file get used by the weapon definition in the .inview file (see SoF2_Weapons_InviewFile.doc for more information).

C. Next you'll need to provide a definition of the weapon's behavior by opening the sof2.wpn file and copying the definition for the M4. Name your new weapon definition the same thing as in step 4…that'd be "AR15" in our example. You'll want to give it a different cvar and maybe use a different ammo type along with other bits of tweaking that make your weapon different from the original ones. For more information on how to create and manipulate weapon definitions in the sof2.wpn file see SoF2_Weapons_WpnFile.doc.

That's basically it! Now run the game, bring down the console and type **map pra2** (for instance) and when the map is loaded type **give all** at the console. You are now in possession of every weapon in the game including the one you just added.

There are several other niggling details pertaining to getting a weapon into the game – creating pickups in a map, displaying them in the menus, that sort of thing – but those are fairly trivial so I'll leave them as "an exercise for the reader".

2.0 Adding a weapon to Multiplayer

To add a weapon to multiplayer, you must first do all of the steps above. However, you must also modify the mp code to accept the new weapon. This document will not get into that at all, as if you have programming knowledge, it should be fairly straightforward. The thing to remember is that adding a weapon to single player is fairly dynamic while with multiplayer, you are actually changing the code and making it incompatible with other 'pure' servers out there.