

---

Subject: scripts.dll 3.2 WIP update

Posted by [jonwil](#) on Sat, 10 Feb 2007 09:21:29 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

This is what is currently in scripts.dll 3.2:

added code to coreshader.cpp to explicitly unset pixel and vertex shaders

replaced code to handle the "current Direct3D textures" array with new code so we can store Direct3D9 textures as well as Direct38 textures

split engine.cpp and shadereng.cpp up into multiple files and did cleanups/fixes on stuff that I cant remember anymore

cleaned up the way initalization of engine stuff and detection of if we are FDS or client is handled

cleanups to the way shaders are defined and such (especially the handling of ID3DXEffect)

shader state manager cleanups

fixed a bug in the code for calculating CRC32

bug fixes to the implementation of DynamicVBAccessClass and friends

working clone of the base code that renders a skin mesh

Improvements to the code for drawing non skin meshes

added code from bhs.dll into relevant places in new engine\_\* files

added new engine call to send a particular integer to the custom HUD code of a given player from a script

Making all calls to SetRenderState and SetTextureStageState made by renegade or by bhs.dll go

through the code in shaders.dll and the new state manager code

cloned ShaderClass::Apply

Cleanups to the definition of TextureMapperClass

added access to the current light direction vector

added code to make the screen fade rendering go through shaders.dll (the intent is that if shaders is off, screen fade works as it does now, if shaders is on, custom screen fade can be provided by post process shaders)

added access to the detection flags for SSE and 3DNow (i.e. code can see if the current machine has a CPU supporting either of those instruction sets)

fixed a bug with Get\_Armour\_Name

Cleaned up the definitions of myIDirect3D\* (as formerly defined in shadereng.h and now defined in engine\_d3d.h)

New code so that if your CPU supports SSE instructions, some matrix/vector math operations will be faster

Added various functions to various math classes

Fixed 2 bugs with the definiition of ScriptableGameObj which caused a crash on the RH8 LFDS

Corrected the definition of Get\_Vehicle\_Seat\_Count

Added definitions for reading the current player list

Changed Get\_GameObj\_By\_Player\_Name, Send\_Custom\_All\_Players, Steal\_Team\_Credits and the new Get\_Team\_Credits engine call to read the player list

added a new engine call float Get\_Team\_Credits(int team); //Count the total credits for a team

fixed a bug with Get\_GameObj

fixed a bug with the definition of PlayerDataClass that broke stuff on the LFDS

new scripts, JFW\_Kill\_Message and JFW\_Kill\_Message\_Display which handle displaying new kill messages. JFW\_Kill\_Message goes on all the objects and then sends a message to

JFW\_Kill\_Message\_Display to actually display the kill message.

added an improved version of JFW\_Resource\_Collector known as JFW\_Resource\_Collector\_2,

changes also by zunnie

new script JFW\_2D\_Sound\_Death\_Team which plays a 2d sound to the team when an object dies (e.g. for "unit lost" sounds)

new script JFW\_Vehicle\_Full\_Sound that, when created, starts a timer. When the timer goes off, it restarts the timer.

Also, when the timer goes off, if the vehicle is full, it plays a 2d sound for the team of the vehicle and displays a message for the team of the vehicle.

The message will be of the form <driver>'s <vehicle> is now full where <driver> is the name of the driver and <vehicle> is the name of the vehicle.

new script, JFW\_Vehicle\_Effect\_Animation. This basically plays an animation whenever there is at least one person in the vehicle.

For example, a spinning radar dish on top of a radar truck.

Added sound that is played when the radar spy zone triggers and gives the allies radar

Fixed a bug with JFW\_Infantry\_Force\_Composition\_Zone and

JFW\_Vehicle\_Force\_Composition\_Zone

new script, JFW\_Cash\_Spy\_Zone which is used to display the enemies cash reserves when the spy enters a silo or refinery spy zone

new script, JFW\_Power\_Spy\_Zone which brings down the enemy power for a period of time when the spy enters the power plant spy zone

Improvements to scripts by Kamuix

New scripts by Kamuix

added some scripts by zunnie

bumped version number to 3.2 and copyright year to 2007

slight improvements to the win32 build process and compiler options for all projects in the scripts.dll and bhs.dll

Changed the ExpVehFac scripts to call Enable\_Engine on flying units that are flying in (should make the rotor blades spin)

small typo fix to Reborn\_IsDeployableMech

cleanup to some parts of bhs.dll

Fixed a bug to do with the nickname exploit fixes that caused the LFDS to crash when people joined

Multisample Anti-Alias

Changes so that certain non-shaders code in shaders.dll gets run even with "shaders off" ("shaders off" basically means no loading databases and no creating shader objects)

Moved large parts of the custom hud code to shaders.dll

new hud.ini keyword to disable kill messages

Crashdumps are now output with sequential filenames much like screenshots

Changed the bhs.dll configuration dialog to be more consistent with the other configuration dialogs

Added new console command to check if a given client has a given file in their data folder (e.g. a map)

bug fixing to the handling of D3DRS\_ZBIAS to hopefully fix all the ZBIAS problems (i.e. flickering bullet holes etc)

bug fixing for npatch handling so that npatches will work again

turret lag fix improvements

fixed a bug with the cmsgp and cmsgt console commands

Added a hud.ini keyword so you can have 2 armour types that are unsquishable

fixed a typo with Set\_Obj\_Radar\_Blip\_Shape\_Player and Set\_Obj\_Radar\_Blip\_Color\_Player

fixed a bug causing the LFDS to crash anytime a player disconnected without properly leaving the

server

Added debug output to d3d8.dll to print if any functions are being called that dont have implementations.

Still to do:

New scripts to send messages to various players (one example would be a "when a player enters this zone, play an optional sound and send a message similar to what cmsgp does)

A new script which is a clone of JFW\_Blow\_Up\_On\_Enter but with a delay

Check that the spy power plant script actually works and if necessary, add any required hacks

Make sure that Set\_Obj\_Radar\_Blip\_Color and Set\_Obj\_Radar\_Blip\_Shape work correctly

Make sure that Disable\_All\_Collisions works correctly

Finish the aforementioned kill message scripts

Code to allow buildings to be brought back to life (requires code on the client)

Change the skin rendering code to spit out tangent and binormal data (required for the offset normal map shader)

Patch the bump map texture mapper to go through the new code and not to call SetRenderState or SetTextureStageState in d3d8.dll anymore

Investigate the RA\_xxx scripts which need changing for 994 and see if I can make any of those changes (or if I need NeoSaber to do it since they are his scripts)

Find a solution to the problem of multisample anti-alias not working correctly with the shadow render targets

Further improvements to the shader state manager

Add code to disable use of state blocks by ID3DXEffect

Finish the new shader stuff (such as the new EffectClass class)

Redo the shaders to use the new system

Add zbias to whichever texture mappers require it in order to fix the problem jerard was having with his doors

More SSE math code

Fix the normal map shader to use the correct light direction

Some of this stuff such as all the great shaders.dll work is being done by SaberHawk