Subject: final changelog for scripts.dll 3.2
Posted by jonwil on Mon, 19 Mar 2007 12:05:59 GMT
View Forum Message <> Reply to Message

Changelog for scripts.dll 3.2:
Improvements to the glass shader and normal map shader (the normal map shader now actually WORKS)
Major improvements and cleanups to shaders.dll including better performance across the board.
Anything you have heard about scripts.dll 3.2 being slower is a lie. Scripts.dll 3.2 may actually (on some graphics cards) be FASTER than stock renegade.
Complete rewrite of post process shader code (they are now called scene shaders)
Big changes to d3d8.dll and bhs.dll to go with the shaders.dll changes
Engine.cpp and shadereng.cpp are now split up into multiple files which means less code duplication and
easier to manage code.
Cleaned up the way initialization of engine stuff and detection of if we are FDS or client is handled
Fixed a bug in the code for calculating CRC32
Numerous bug fixes to the shader code
Numerous bug fixes to the shader related rendering code (which, for example, calculates Tangent and Binormal numbers)
Proper fix for all the zbuffer issues (esp on ATI cards)
Numerous bug fixes to engine calls (formerly in engine.cpp)
Numerous bug fixes to various scripts
Numerous other bug fixes
Cleanups to the definition of TextureMapperClass
Cleanups to the definition of the classes that handle vertex and index buffers
New class for cubemap textures
Several functions for shaders.dll to use including ways to get the current projection matrix and light vector
Definition of TextMessageEnum which will make it easier to see (when inside hooks like the chat hook) what sort of message it is
Code in shaders.dll to spit out appropriate D3DPERF calls for PIX to pick up
Code to spit out various debugging messages at various times
int Get_Harvester_Preset_ID(int Team); //Get the harvester preset ID for this team
bool Is_Harvester_Preset(GameObject *obj); //Checks if the object has the same preset used for the harvesters
Improvements to a whole bunch of math code (mainly used by shaders.dll) so that if your CPU has SSE, it will use
SSE instructions. This will make it faster.
Class definition for GameObjObserverTimerClass
Class definition for GameObjCustomTimerClass
Class definition for TransitionDataClass
Class definition for OffenseObjectClass
Class definition for PhysicalGameObjDef
Class definition for PhysObserverClass
Class definition for CombatPhysObserverClass
Class definition for PhysicalGameObj
Class definition for SpecialEffectsGameObjDef

Class definition for SpecialEffectsGameObj
Class definition for SimpleGameObjDef
Class definition for SimpleGameObj
Class definition for PowerupGameObjDef
Class definition for PowerupGameObj
Class definition for MuzzleRecoilClass
Class definition for ArmedGameObjDef
Class definition for ArmedGameObj
Class definition for CinematicGameObjDef
Class definition for CinematicGameObj
Class definition for BeaconGameObjDef
Class definition for BeaconGameObj
Class definition for C4GameObjDef
Class definition for C4GameObj
bool Get_Is_Powerup_Persistant(GameObject *obj); //Returns if this powerup is persistent
void Set_Is_Powerup_Persistant(GameObject *obj,bool Persist); //Sets if this powerup is persistent
bool Get_Powerup_Always_Allow_Grant(GameObject *obj); //Returns if this powerup is set to always allows grant
void Set_Powerup_Always_Allow_Grant(GameObject *obj,bool Grant); //Change if this powerup is set to always allows grant
int Get_Powerup_Grant_Sound(GameObject *obj); //Returns the sound that is played when this powerup is picked up
void Set_Powerup_Grant_Sound(GameObject *obj,int SoundID); //Set the sound that is played when this powerup is picked up
void Set_Vehicle_Is_Visible(GameObject *obj,bool visible); //works like Set_Is_Visible but for vehicles, makes them be ignored by Enemy_Seen
float Get_Team_Credits(int team); //Count the total credits for a team
extern SList<cPlayer *> *PlayerList; //Current player list
void Change_Team_2(GameObject *obj,int Team); //changes the team of a player given their GameObject without killing the player,passing anything other than 0 = Nod,1 = GDI will crash
int Get_Player_Type(GameObject *obj); //Get the player type of a player from the cPlayer
Bug fixes to JFW_Cinematic
New script JFW_Kill_Message_Display
New script JFW_Kill_Message
Bug fixes to JFW_Radar_Spy_Zone
Bug fixes to JFW_Radar_Jammer
Bug fixes to JFW_Sonar_Pulse
New script JFW_Resource_Collector_2
New script JFW_Low_Power_Message
New script JFW_Message_Send_Zone
New script JFW_Message_Send_Zone_Team
New script JFW_Message_Send_Zone_Player
New script JFW_Message_Send_Death
New script JFW_Message_Send_Death_Team
New script JFW_Message_Send_Custom
New script JFW_Message_Send_Custom_Team
New script JFW_Spy_Switch_Team

New script JFW_Spy_Vehicle_Ignore
New script JFW_2D_Sound_Death_Team
New script JFW_Vehicle_Full_Sound
New script JFW_C4_Sound
Fixed a bug with Get_Armour_Name
Fixed 2 bugs with the definition of ScriptableGameObj which caused a crash on the RH8 LFDS
Improved the math classes
Corrected the definition of Get_Vehicle_Seat_Count
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
fixed a bug with Get_GameObj
fixed a bug with the definition of PlayerDataClass that broke stuff on the LFDS
Improvements to scripts by Kamuix
New scripts by Kamuix
New 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 (makes the rotor blades spin)
small typo fix to Reborn_IsDeployableMech
Fixed a bug to do with the nickname exploit fixes that caused the LFDS to crash when people joined
Working multi-sample 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
Added a hud.ini keyword to disable kill messages
added new engine call to send a particular integer to the custom HUD code of a given player from a script
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)
Fixed bugs with NPatches to make them work again
Big improvements to the turret lag fix
fixed a bug with the cmsgp and cmsgt console commands
Added hud.ini keywords so you can have more armor types that are unsqiushable
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 don't have implementations
(which either indicates a bug or some weird case that is only ever used once in a blue moon or on specific weird hardware)
Bug fixes to JFW_Vehicle_Lock
New script JFW_Vehicle_Effect_Animation
Bug fixes to JFW_Repair_Zone_2

Bug fixes to JFW_Sell_Zone
Bug fixes to JFW_Infantry_Force_Composition_Zone
Bug fixes to JFW_Vehicle_Force_Composition_Zone
New script JFW_Cash_Spy_Zone
New script JFW_Power_Spy_Zone
New script JFW_Blow_Up_On_Enter_Delay
New script JFW_Conyard_Spy_Zone
Bug fixes to RA_Mine
Bug fixes to RA_Demolition_Truck_Improved
Bug fixes to RA_MAD_Tank_Improved
Bug fixes to RA_Conyard_Controller_Improved
Bug fixes to RA_Visible_Driver
New script RA_Thief_Improved
New script RA_Base_Defense_Powered
New scripts for Roleplay 2 written by Jerad2142
New shaders.dll hook called when a shutdown is happening (as opposed to a device reset)
New shaders.dll hooks for the HUD code
New shaders.dll hooks for direct3d related stuff
New shaders.dll hook to pass the screen fade manager rendering through shaders.dll
Improvements to the way network stuff is sent by bhs.dll
Fixed a bug in the cut/copy/paste code for edit controls
Added a new engine call so that if you are inside the ::Killed event and were killed by C4 or beacon, you can get the C4GameObj or BeaconGameObj that did the killing and find out stuff about it.
Added a hud.ini keyword to change the registry key that the WOL URLs (network status, news etc) are read from
Made 16 bit graphics mode work again
Fixed problems preventing single player from going past the first mission
Fixed it so that it wont crash when you alt-tab anymore

Also, we are looking into a fix to make Reborn_IsDeployableMech work that may get in for 3.2
And, NeoSaber is working on a new set of scripts for the RA:APB Nuke Silo that may get in for 3.2

Other than that, its mainly the final pieces of work on sdbedit.exe for the scene shaders plus a new example shaders.sdb file that need to be done before we release 3.2

Subject: Re: final changelog for scripts.dll 3.2
Posted by jnz on Mon, 19 Mar 2007 15:35:06 GMT
View Forum Message <> Reply to Message

hmm, is extern SList<cPlayer *> *PlayerList globalaly defined and always updated? or would i have to use an engine call to update it?

nice work

Subject: Re: final changelog for scripts.dll 3.2
Posted by CarrierII on Mon, 19 Mar 2007 16:44:23 GMT
View Forum Message <> Reply to Message

Woot at SSE instructions!

Can't wait.

Subject: Re: final changelog for scripts.dll 3.2
Posted by Jerad2142 on Mon, 19 Mar 2007 17:45:48 GMT
View Forum Message <> Reply to Message

So did the blast door transparency issue get fixed?

Subject: Re: final changelog for scripts.dll 3.2
Posted by StealthEye on Mon, 19 Mar 2007 18:39:25 GMT
View Forum Message <> Reply to Message

gamemodding, the list will always be up to date: It is the list used by (the) renegade (fds) internally.

Subject: Re: final changelog for scripts.dll 3.2
Posted by jnz on Mon, 19 Mar 2007 23:47:31 GMT
View Forum Message <> Reply to Message

yay, and thanks.