CScriptObjectSystem Reference

2004-09-13

Table of Contents

CScriptObjectSystem: C++ functions available in Lua script	

`CreateDownload`
`()`
	Usage:
	Gets a download object.
	
	Parameters: none
	
	Return:
	Returns a ScriptObject, the download object which has been created.
	
	Code Example:
	UI.BannerImageDownload = System:CreateDownload();

`LoadFont`
`(string)`
	Usage:
	Loads a font and makes it available for future-selection.
	
	Parameters:
	string: This is the name of font-xml-file (no suffix).
	
	Return: none
	
	Code Example:
	System:LoadFont("radiosta");

`ExecuteCommand`
`(string)`
	Usage:
	Executes a console command.
	
	Parameters:
	string: The name of the command to execute.
	
	Return: none
	
	Code Example:
	`System:ExecuteCommand(Sender.szCommand.." "..szText);`

CScriptObjectSystem: C++ functions available in Lua script	
LogToConsole (string)	**Usage:** Writes a string to the console. **Parameters:** string: The string to write to the console. **Return:** none **Code Example:** System:LogToConsole("SERVER:OnContact");
LogAlways (string)	**Usage:** Logs even with log verbosity set to 0 - without <LUA>. **Parameters:** string: The string to write to the logfile. **Return:** none **Code Example:** System:LogAlways(sText);
ClearConsole ()	**Usage:** Clears the console. **Parameters:** none **Return:** none **Code Example:** not used

CScriptObjectSystem: C++ functions available in Lua script

`GetConsoleKeyName`
`()`
	Usage:
	Gets the name of the key which is binded to show up the console.
	Parameters: none
	Return:
	Returns the name of the key or nil if failed.
	Code Example:
	System:GetConsoleKeyName()

`Log`
`(string)`
	Usage:
	Writes a message to the logfile and the console.
	Parameters:
	string: This is the string to write.
	Return: none
	Code Example:
	System:Log("Unrecognized idx type");

`Warning`
`(string)`
	Usage:
	Prints out a warning message to the logfile.
	Parameters:
	string: The warning message to print.
	Return: none
	Code Example:
	System:Warning("[AI] helicopter Gunship:UpdateServer can't find pointReinforce");

`Error`
`(string)`
	Usage:
	Prints out an error message to the logfile.
	Parameters:
	string: The error message to print.
	Return: none
	Code Example:
	System:Error(szError);

CScriptObjectSystem: C++ functions available in Lua script	
GetCurrTime ()	**Usage:** Gets the current time. **Parameters:** none **Return:** Returns the current time as a float value. **Code Example:** System:GetCurrTime()
GetCurrAsyncTime ()	**Usage:** Gets the current asynctime. **Parameters:** none **Return:** none **Code Example:** self.fActivateTime = System:GetCurrAsyncTime();
GetFrameTime ()	**Usage:** Gets the current frame time. **Parameters:** none **Return:** Returns a float value.
GetLocalOSTime ()	**Usage:** Gets the exact OS time, including the day, month etc... **Parameters:** none **Return:** Returns a ScriptObject, a table with the values for lua.

CScriptObjectSystem: C++ functions available in Lua script	

`DrawLabelImage`
`(vector3,`
` float,`
` int)`
	Usage:
	Draws a label with an image file.
	
	Parameters:
	vector3: The position where to draw the label.
	
	float: The size of the label as a float value.
	
	int: The texture id.
	
	Return: none
	
	Code Example:
	System:DrawLabelImage(pos, self.Properties.LabelSize, self.nTextureId);

`GetEntity`
`(int)`
	Usage:
	Gets an entity by id.
	
	Parameters:
	int: This is the entity id.
	
	Return:
	Returns an entity object on success, if failed, the function returns nil.
	
	Code Example: not used

`GetEntities`
`()`
	Usage:
	Gets all entities currently present in the level.
	
	Parameters: none
	
	Return:
	Returns a table filled with all entities currently present in the level.

`GetEntitiesInRadius`
`(CScriptObjectVector,`
` float)`
	Usage:
	Gets all entities in a specified radius.
	
	Parameters:
	CScriptObjectVector: The vector, as the center of the sphere.
	
	float: This is the length of the radius in which the function should check.
	
	Return:
	Returns a table filled with all entities contained in the specified radius. Otherwise, the function returns nil.
	
	Code Example: not used

CScriptObjectSystem: C++ functions available in Lua script

`GetTeamMembers`
`(int)`
	Usage:
	Gets all entity members of a certain team.
	DOES NOT WORK! THE BODY OF THIS FUNCTION IS COMMENTED OUT! IT ALWAYS WILL RETURN NIL AFTER A CALL!
	Parameters:
	int: The team id.
	Return:
	Returns nil.

`GetEntityByName`
`(string)`
	Usage:
	Gets an entity object by using its name.
	Parameters:
	string: The name of an entity in a map.
	Return:
	Returns a ScriptObject, meaning the entity as a table, or nil if it fails.
	Code Example:
	ent = System:GetEntityByName(self.Properties.AttachToUp);

`LoadAnimatedTexture`
`(string,`
` int)`
	Usage:
	Loads a set of textures to create an animated texture.
	Parameters:
	string: A path to a texture, including the texture name and file extension, like *.tga
	int: The number of frames (textures).
	Return:
	Returns != nil on success,
	nil if failed.
	Code Example:
	self.TexturesId[ind] = System:LoadAnimatedTexture(value,self.Properties.nFrames);

CScriptObjectSystem: C++ functions available in Lua script

`LoadTexture`
`(string,`
` int,`
` bool)`
	Usage:
	Loads a texture using a path.
	
	Parameters:
	string: The texture name and the corresponding path.
	
	int: [Optional] Load as Cube map or not.
	
	bool: [Optional] Clamp the texture or not.
	
	Return:
	Returns != nil on success,
	nil if failed.
	
	Code Example:
	texture = System:LoadTexture("Textures/Decal/Default.dds");

`LoadObject`
`(string)`
	Usage:
	Loads a *.CGF model file which contains geometry.
	
	Parameters:
	string: The name of the file we want to load.
	
	Return:
	Returns != nil on success,
	nil if failed.
	
	Code Example:
	geometry =
	System:LoadObject("Objects/Weapons/shells/smgshell.cgf"),

CScriptObjectSystem: C++ functions available in Lua script

```
DrawSprite

(string,
 vector3,
 int)
```

Usage:
Draws a sprite.
DOES NOT WORK! THE BODY OF THIS FUNCTION IS COMMENTED OUT! IT ALWAYS WILL RETURN NIL AFTER A CALL!

Parameters:
string: The name of the Image file.

vector3: A position.

int: A blending mode if != 0.

```
// possible render modes
  1:  R_BLEND_MODE__ZERO__SRC_COLOR
  2:  R_BLEND_MODE__SRC_COLOR__ZERO
  3:  R_BLEND_MODE__SRC_COLOR__ONE_MINUS_SRC_COLOR
  4:  R_BLEND_MODE__SRC_ALPHA__ONE_MINUS_SRC_ALPHA
  5:  R_BLEND_MODE__ONE__ONE
  6:  R_BLEND_MODE__DST_COLOR__SRC_COLOR
  7:  R_BLEND_MODE__ZERO__ONE_MINUS_SRC_COLOR
  8:  R_BLEND_MODE__ONE__ONE_MINUS_SRC_COLOR
  9:  R_BLEND_MODE__ONE__ZERO
 10:  R_BLEND_MODE__ZERO__ZERO
 11:  R_BLEND_MODE__ONE__ONE_MINUS_SRC_ALPHA
 12:  R_BLEND_MODE__SRC_ALPHA__ONE
 14:  R_BLEND_MODE__ADD_SIGNED      // Should it be 13?
```

Return: none

```
DeformTerrain

(CScriptObjectVector,
 float,
 int,
 bool)
```

Usage:
Creates a terrain deformation at the given point.
This function is called when a projectile or a grenade explodes.

Parameters:
CScriptObjectVector: This is the position of the explosion.

float: The size of the explosion.

int: [Optional] A texture id, a decal usually.

bool: [Optional] Deformable terrain or not. Set to true by default.

Return: none

Code Example:

```
System:DeformTerrain( pos, 8, Grenade.decal_tid );
```

CScriptObjectSystem: C++ functions available in Lua script

`ScreenToTexture`
`()`
	Usage:
	Renders the current screen to a texture. (Seems to be only implemented in the OGL renderer?)
	Parameters: none
	Return: none
	Code Example: not used

`LoadImage`
`(string,`
` bool,`
` bool)`
	Usage:
	Loads a texture from a file. Optionally allows to set the FT_NOREMOVE flag for the renderer when loading the texture.
	Parameters:
	string: This is the path to a texture.
	bool: [Optional] Should the texture be clamped or not? False by default.
	bool: [Optional] Should the texture be removable or not? False by default.
	Return:
	Returns != nil on success,
	nil if failed.
	Code Example:
	self.Indicator = System:LoadImage("Textures/Hud/Binocular/binoculars_targeting");

`FreeImage`
`(int)`
	Usage:
	Frees a texture again.
	Parameters:
	int: The id of a certain texture.
	Return:
	Returns != nil on success,
	nil if failed.
	Code Example:
	System:FreeImage(iTexture);

CScriptObjectSystem: C++ functions available in Lua script

```
DrawLine

(CScriptObjectVector,
 CScriptObjectVector,
 float,
 float,
 float,
 float)
```

Usage:
Draws a line between two points in 3d space.

Parameters:
CScriptObjectVector: The starting point.

CScriptObjectVector: The ending point.

// The color components
float: Red

float: Green

float: blue

float alpha

Return: none

Code Example: not used

```
Draw2DLine

(int,
 int,
 int,
 int,
 float,
 float,
 float,
 float)
```

Usage:
Draws a line in 2d space.

Parameters:

// Point 1
int: X Component of the point.

Int: Y component of the point.

// Point 2
int: X Component of the point.

int: Y component of the point.

// The color components
float: Red

float: Green

float: Blue

float: Alpha

Return: none

Code Example:
System:Draw2DLine(398, 300, 398-radius+1, 300, 0,0,0,1);

CScriptObjectSystem: C++ functions available in Lua script

```
DrawImage

(int,
 int,
 int,
 int,
 int,
 int)
```

Usage:
Draws a given texture image, with a blending mode if specified.

Parameters:
int: This is the texture id.

int: The x position(the screen is normalized at 800x600).

int: The y position(the screen is normalized at 800x600).

int: The image width.

int: The image height.

int: A blending mode.

Return: none

Code Example:
System:DrawImage(Binoculars.TID_Background, 0, 0, 800, 600, 4);

CScriptObjectSystem: C++ functions available in Lua script

Function	Description

`DrawImageColor`
`(int,`
` int,`
` int,`
` int,`
` int,`
` int,`
` float,`
` float,`
` float,`
` float)`
	Usage:
	Draws a given texture image, with a blending mode if specified. You can set colors for the image.
	Parameters:
	int: This is the texture id.
	int: The x position (the screen is normalized at 800x600).
	int: The y position (the screen is normalized at 800x600).
	int: The image width.
	int: The image height.
	int: A blending mode.
	// The color components
	float: Red
	float: Green
	float: Blue
	float: Alpha
	Return: none
	Code Example:
	System:DrawImageColor(FTBSniping.BreathBlurImg, 0, 0, 800, 600, 4, 1, 1, 1, mul);

CScriptObjectSystem: C++ functions available in Lua script

```
DrawImageCoords

(int,
 int,
 int,
 int,
 int,
 int,
 float,
 float,
 float,
 float)
```

Usage:
Draws a given texture image, with a blending mode if specified. You can set the uv-coordinates manually.

Parameters:
int: This is the texture id.

int: The x position (the screen is normalized at 800x600).

int: The y position (the screen is normalized at 800x600).

int: The image width.

int: The image height.

int: A blending mode.

// The uv components of the texture coordinates
float: U1

float: V1

float: U2

float: V2

Return: none

Code Example: not used

CScriptObjectSystem: C++ functions available in Lua script

```
DrawImageColorCoords
(int,
 int,
 int,
 int,
 int,
 int,
 float,
 float,
 float,
 float,
 float,
 float,
 float,
 float)
```

Usage:
Draws a given texture image, with a blending mode if specified. You can set the uv-coordinates manually.

Parameters:
int: This is the texture id.

int: The x position (the screen is normalized at 800x600).

int: The y position (the screen is normalized at 800x600).

int: The image width.

int: The image height.

int: A blending mode.

// The color components
float: Red

float: Green

float: Blue

float: Alpha

// The uv components of texture coordinates
float: U1

float: V1

float: U2

float: V2

Return: none

Code Example:
System:DrawImageColorCoords(RL.ZoomBackgroundTID, 0, 0, 400, 600, 4, 1, 1, 1, 1, fTexelWidth, 1-fTexelHeight, 1-fTexelWidth, fTexelHeight);

CScriptObjectSystem: C++ functions available in Lua script

```
DrawTriStrip
(int,
 int,
 table,
 float,
 float,
 float,
 float)
```

Usage:
Draws a trianglestrip with a texture on it.

Parameters:
int: This is the texture id.

int: A render mode.

// possible render modes

```
 1:  R_BLEND_MODE__ZERO__SRC_COLOR
 2:  R_BLEND_MODE__SRC_COLOR__ZERO
 3:  R_BLEND_MODE__SRC_COLOR__ONE_MINUS_SRC_COLOR
 4:  R_BLEND_MODE__SRC_ALPHA__ONE_MINUS_SRC_ALPHA
 5:  R_BLEND_MODE__ONE__ONE
 6:  R_BLEND_MODE__DST_COLOR__SRC_COLOR
 7:  R_BLEND_MODE__ZERO__ONE_MINUS_SRC_COLOR
 8:  R_BLEND_MODE__ONE__ONE_MINUS_SRC_COLOR
 9:  R_BLEND_MODE__ONE__ZERO
10:  R_BLEND_MODE__ZERO__ZERO
11:  R_BLEND_MODE__ONE__ONE_MINUS_SRC_ALPHA
12:  R_BLEND_MODE__SRC_ALPHA__ONE
14:  R_BLEND_MODE__ADD_SIGNED      // Should it be 13?
```

table: A table (struct in c++), including the following information for the triangle:

```
float x, y, z;
unsigned char c[4];
float u, v;
```

// The color components
float: Red

float: Green

float: Blue

float: Alpha

Return: none

Code Example: not used

```
SetWorldColorRatio
(float)
```

Usage:
Changes the world color ratio.

Parameters:
float: This is the world color ratio.

Return: none

Code Example:
System:SetWorldColorRatio(1);

CScriptObjectSystem: C++ functions available in Lua script	
SetGammaDelta (float)	**Usage:** Sets a gamma delta value for the renderer. **Parameters:** float: The gamma value. value: The value itself, string or number. **Return:** none **Code Example:** not used
DrawRectShader (string, float, float, float, float, float, float, float, float)	**Usage:** Draws a shader rectangle with passed position, size and colors. **Parameters:** string: The name of the shader tp draw. float: X-Position float: Y-Position float: Width float: Height // The color components float: Red float: Green float: Blue float: Alpha **Return:** none **Code Example:** System.DrawRectShader("ScreenDistort", 0, 0, 800, 600, 1, 1, 1, 1)
SetScreenShader (string)	**Usage:** Sets a screen shader. **Parameters:** string: This is the name of the shader to apply. **Return:** none **Code Example:** not used

CScriptObjectSystem: C++ functions available in Lua script	

`ShowConsole`
`(int)`
	Usage:
	Shows or hides the console.
	Parameters:
	int: Set to 0 by default. 0 closes the console, 1 brings it up.
	Return: none
	Code Example:
	System:ShowConsole(0);

`SetScreenFx`
`(string,`
` int)`
	Usage:
	Activates a screen effect.
	Parameters:
	string: This is the name of the effect.
	int: Enable (1) or disable (0) the effect.
	Return: none
	Code Example:
	System:SetScreenFx("NightVision", 1);

CScriptObjectSystem: C++ functions available in Lua script

```
SetScreenFxParamInt

(string,
 string,
 int)
```

Usage:
Sets a screen effect with additional parameters.

Parameters:
string: The name of the effect.

string: The effect parameter.

int: An integer value, passed as last parameter in c++ code:
m_pSystem->GetI3DEngine()->SetScreenFxParam(string, string, int)

The list of possible combinations of effects and parameters:

// effect
- ScreenFade:
    ScreenFadeTime // parameter
    ScreenPreFadeTime

- ScreenBlur:
    ScreenBlurAmount
    ScreenBlurColorRed
    ScreenBlurColorGreen
    ScreenBlurColorBlue

- FlashBang:
    FlashBangTimeScale
    FlashBangFlashPosX
    FlashBangFlashPosY
    FlashBangFlashSizeX
    FlashBangFlashSizeY
    FlashBangForce

Return: none

Code Example: not used

CScriptObjectSystem: C++ functions available in Lua script

`SetScreenFxParamFloat`

```
(string,
 string,
 float)
```

Usage:
Sets a screen effect with parameters.

Parameters:
string: The name of the effect.

string: The effect parameter(s).

float: A floating point value, passed as last parameter in c++ code:
m_pSystem->GetI3DEngine()->SetScreenFxParam(string, string, float)

The list of possible combinations of effects and parameters:

// effect
- ScreenFade:
  ScreenFadeTime // parameter
  ScreenPreFadeTime

- ScreenBlur:
  ScreenBlurAmount
  ScreenBlurColorRed
  ScreenBlurColorGreen
  ScreenBlurColorBlue

- FlashBang:
  FlashBangTimeScale
  FlashBangFlashPosX
  FlashBangFlashPosY
  FlashBangFlashSizeX
  FlashBangFlashSizeY
  FlashBangForce

Return: none

Code Example:
System:SetScreenFxParamFloat("ScreenBlur", "ScreenBlurAmount", self.hitdamagecounter/10.0);

`GetScreenFx`

`(string)`

Usage:
Checks if a screen effect is enabled or diabled.

Parameters:
string: The name of the screen effect to check for.

Return:
Returns an integer value (the state), otherwise nil.

Code Example:
local bFlashBangActive = System:GetScreenFx("FlashBang");

CScriptObjectSystem: C++ functions available in Lua script

`GetScreenFxParamInt`
`(string,`
` string)`
	Usage:
	Returns a screen effect parameter value.
	Parameters:
	string: The name of an effect.
	string: The effect parameter.
	Return:
	Returns the integer value for this effect parameter. Otherwise nil if failed (effect does not exist).
	Code Example: not used

`GetScreenFxParamFloat`
`(string,`
` string)`
	Usage:
	Returns a screen effect parameter value.
	Parameters:
	string: The name of an effect.
	string: The effect parameter.
	Return:
	Returns the integer value for this effect parameter. Otherwise nil if failed (effect does not exist).
	Code Example:
	local fFlashBangTimeScale = System:GetScreenFxParamFloat("FlashBang", "FlashBangTimeScale");

`SetScissor`
`(int,`
` int,`
` int,`
` int)`
	Usage:
	Set scissoring screen area.
	Parameters:
	int: The x value on screen.
	int: The y value on screen.
	int: The width value.
	int: The height value.
	Return: none
	Code Example:
	System:SetScissor(20, 20-9, 800-40, fFinalBoxHeight-8);

CScriptObjectSystem: C++ functions available in Lua script	

`GetCPUQuality`
`()`
	Usage:
	Checks the cpu quality and returns it.
	Parameters: none
	Return:
	Returns the cpu quality as an integer.
	0: // < 1.7 GHz (1.9 GHz = safety net) is minimum
	1: // < 2.6 GHz (2.5 GHz = safety net) is medium
	2: // < 3.0 GHz (2.9 GHz = safety net) is high
	3: // very high

`GetGPUQuality`
`()`
	Usage:
	Checks the video card quality and returns it.
	Parameters: none
	Return:
	Returns the gpu quality as an integer depending on the brand and product name. Check the c++ code for more details.
	0: minimum
	1: medium
	2: high
	3: very high

`GetSystemMem`
`()`
	Usage:
	Checks the size of total physical memory.
	Parameters: none
	Return:
	Returns the size of total physical memory in MB as an integer.

`GetVideoMem`
`()`
	Usage:
	Checks for the siize of available video memory.
	Parameters: none
	Return:
	Returns the size of the available video memory in MB as an integer.

CScriptObjectSystem: C++ functions available in Lua script	

`IsPS20Supported`
`()`
	Usage:
	Checks if pixel shaders version 2.0 are supported.
	Parameters: none
	Return:
	Returns 1 if true, 0 if false.

`IsHDRSupported`
`()`
	Usage:
	Checks if hardware supports high dynamic range rendering.
	Parameters: none
	Return:
	Returns 1 if true, nil if false.

`ActivateLight`
`(string,`
` bool)`
	Usage:
	Activates or disables a light.
	Parameters:
	string: The lightname.
	bool: Enable or disable the light.
	Return: none
	Code Example: not used

`SetSkyBox`
`(string,`
` float,`
` bool)`
	Usage:
	Sets a skybox shader effect.
	Parameters:
	string: The name of the shader to use.
	float: A blending time. // NOT SUPPORTED
	bool: “UseWorldBrAndColor“ // NOT SUPPORTED
	Currently the engine simply sets the shader and ignores the other two parameters.
	Return: none
	Code Example: not used

CScriptObjectSystem: C++ functions available in Lua script

`SetWaterVolumeOffset`
`(string,`
`float,`
`float,`
`float)`
	Usage:
	Sets an offset position to a certain water volume.
	Parameters:
	string: The name of a water volume.
	Return:
	Returns a warning string, if the volume was not found or not set.
	Code Example:
	System:SetWaterVolumeOffset(self.Properties.WaterVolume, 0, 0, 0);

`MeasureTime`
`(string)`
	Usage:
	Sets a named checkpoint for the profiler.
	Parameters:
	string: This is the name of the label.
	Return: none
	Code Example: not used

`IsValidMapPos`
`(CScriptObjectVector)`
	Usage:
	Checks if a certain point in the map is valid or not.
	Parameters:
	CScriptObjectVector: A table, containing x, y, z values, thus defining a position.
	Return:
	Returns a boolean, true = 1
	false = 0.
	Code Example:
	elseif (System:IsValidMapPos(pos) ~= 1) then
	...

`EnableMainView`
`(bool)`
	Usage:
	Toggles MainViewRendering.
	THIS FEATURE IS NOT IMPLEMENTED! THE FUNCTION RETURNS WITHOUT DOING ANYTHING, BUT SENDING A 0 ASSERT.
	Parameters:
	bool: Enable or disable MainViewRendering.
	Return: none
	Code Example: not used

CScriptObjectSystem: C++ functions available in Lua script	

`EnableOceanRendering`
`(bool,`
` bool)`
	Usage:
	Enables, disables OceanRendering.
	
	Parameters:
	bool: Render ocean or not.
	
	bool: Render shore or not.
	
	Return: none
	
	Code Example: not used.

`ScanDirectory`
`(string,`
` int,`
` int)`
	Usage:
	Scans a directory, there are three scan modes available.
	
	Parameters:
	string: The path, a folder.
	
	int: [Optional] The scan mode. Can be one of the following modes:
	
	#define SCANDIR_ALL 0
	#define SCANDIR_FILES 1
	#define SCANDIR_SUBDIRS 2
	
	int: [Optional] Search in a pack or not. Set to 0 by default.
	
	Return:
	Returns a ScriptObject, a table for lua.
	
	Code Example:
	local FileList = System:ScanDirectory("./"..szCutSceneFolder, SCANDIR_FILES);

`DebugStats`
`(bool)`
	Usage:
	Debug stats or not.
	
	Parameters:
	bool: Checkpoints enabled or not.
	
	Return: none
	
	Code Example: not used

CScriptObjectSystem: C++ functions available in Lua script	
`ViewDistanceSet (float)`	**Usage:** Sets the maximum view distance in the engine. **Parameters:** float: The maximum amount of view distance allowed. Minimum number is 20. If the passed value is smaller, it will be set to 20. **Return:** none **Code Example:** System:ViewDistanceSet(self.outsideViewDist);
`ViewDistanceGet ()`	**Usage:** Gets the current maximum of view distance. **Parameters:** none **Return:** Returns the value of the maximum allowed view distance as a floating point value.
`SetFogEnd (float)`	**Usage:** Sets the end distance of fog. Also sets the alpha value of the skybox accordingly. **Parameters:** float: The fog end value. **Return:** none **Code Example:** System:SetFogEnd(self.outsideEnd);
`SetFogStart (float)`	**Usage:** Sets the start distance of fog. **Parameters:** float: The fog start value. **Return:** none **Code Example:** System:SetFogStart(self.curStart);

CScriptObjectSystem: C++ functions available in Lua script	

`SetFogColor`
`(CScriptObjectColor)`
	Usage:
	Sets the color of the fog.
	Parameters:
	CScriptObjectColor: A table in lua, containg r/g/b color information. Could be looking like this:
	curColor = {0.0, 0.0, 0.0},
	Return: none
	Code Example:
	System:SetFogColor(self.curColor);

`GetFogEnd`
`()`
	Usage:
	Gets the fog end value.
	Parameters: none
	Return:
	Returns the fog end value as a floating point value.
	Code Example:
	self.outsideEnd = System:GetFogEnd();

`GetFogStart`
`()`
	Usage:
	Gets the fog start value.
	Parameters: none
	Return:
	Returns the fog start value as a floating point value.
	Code Example:
	self.outsideStart = System:GetFogStart();

`GetFogColor`
`()`
	Usage:
	Gets the fog color values.
	Parameters: none
	Return:
	Returns the fog color values.
	Code Example:
	self.outsideColor = System:GetFogColor();

CScriptObjectSystem: C++ functions available in Lua script	

`ApplyForceToEnvironment`
`(CScriptObjectVector,`
`float,`
`float)`
	Usage:
	Gets a list of all save-games.
	Parameters:
	CScriptObjectVector: A position where to apply the force.
	float: The radius of the force.
	float: The force value itself.
	Return: none
	Code Example:
	System:ApplyForceToEnvironment(pos, self.Properties.fBendRadius, self.Properties.fBendForce);

`GetWorldColor`
`()`
	Usage:
	Gets the current world color.
	Parameters: none
	Return:
	Returns a CScriptObjectColor object with color information, a table in lua.

`SetWorldColor`
`(CScriptObjectColor)`
	Usage:
	Changes the world color.
	Parameters:
	CScriptObjectColor: A table, containing color values.
	Return: none
	Code Example:
	System:SetWorldColor(self.outsideEnvColor);

`GetOutdoorAmbientColor`
`()`
	Usage:
	Gets the current outdoor ambient color value.
	Parameters: none
	Return:
	Returns a GetOutdoorAmbientColor object, containing the current outdoor color values, a table in lua.

CScriptObjectSystem: C++ functions available in Lua script

`SetOutdoorAmbientColor`
`(CScriptObjectColor)`
	Usage:
	Sets the outdoor ambient color values.
	
	Parameters:
	CScriptObjectColor: A table, containing, r/g/b color values.
	It could be looking like this:
	
	outsideAmbientColor = {0.0,0.0,0.0},
	
	Return: none
	
	Code Example:
	System:SetOutdoorAmbientColor(self.outsideAmbientColor);

`SetBFCount`
`(int)`
	Usage:
	Sets the current butterfly count.
	
	THIS CODE IS NOT WORKING ANYMORE! BUTTERFLIES ARE NOT SUPPORTED LIKE THIS BY THE 3DENGINE ANYMORE, PLEASE USE BOID ENTITIES INSTEAD.
	
	Parameters:
	int: The number of butterflies to set.
	
	Return: none
	
	Code Example:
	System:SetBFCount(Lerp(self.outsideBFlyNumber, self.Properties.BFlyNumber, fadeCoeff));
	
	IT IS USED LIKE THIS IN THE “Bfly.lua“, BUT THE NATIVE ENGINE CODE DOES NOT SUPPORT THIS FEATURE ANYMORE.

`GetBFCount`
`()`
	Usage:
	Gets the current butterfly count.
	
	THIS FUNCTION ALWAYS RETURNS 0!
	
	Parameters: none
	
	Return:
	Always returns 0.

CScriptObjectSystem: C++ functions available in Lua script	
`SetGrasshopperCount ()`	**Usage:** Sets the current grashopper count. THIS CODE IS NOT WORKING ANYMORE! GRASHOPPERS ARE NOT SUPPORTED LIKE THIS BY THE 3DENGINE ANYMORE, PLEASE USE BOID ENTITIES INSTEAD. **Parameters:** int: The number of grashoppers to set. **Return:** none **Code Example:** System:SetGrasshopperCount(Lerp(self.outsideGrasshopperNumber, self.Properties.GrasshopperNumber, fadeCoeff)); IT IS USED LIKE THIS IN THE “Grasshopper.lua“, BUT THE NATIVE ENGINE CODE DOES NOT SUPPORT THIS FEATURE ANYMORE.
`GetGrasshopperCount ()`	**Usage:** Gets the current grasshopper count. THIS FUNCTION ALWAYS RETURNS 0! **Parameters:** none **Return:** Always returns 0 and prints a warning to the logfile.
`SetGrasshopperCGF (int)`	**Usage:** This function sets a *.CGF model as grasshopper. THIS FUNCTION DOES NOT LOAD A MESH ANYMORE! IT ONLY PRINTS A WARNING TO THE LOGFILE. THIS FEATURE IS NOT SUPPORTED ANYMORE! **Parameters:** int: The id of a grasshopper entity. **Return:** none **Code Example:** System:SetGrasshopperCGF(self.id);

CScriptObjectSystem: C++ functions available in Lua script

`GetTerrainElevation`
`(CScriptObjectVector)`
	Usage:
	Gets the terrain elevation for a certain point.
	Parameters:
	CScriptObjectVector: The position to check at.
	Return:
	Returns a floating point value, representing the terrain elevation at the passed point.
	Code Example:
	local terrain = System:GetTerrainElevation(pos);

`SetSkyFade`
`(float,`
` float)`
	Usage:
	Sets the values for the two skyfade members in CScriptObjectSystem.
	Parameters:
	float: The value for
	m_SkyFadeStart; // when fogEnd less - start to fade sky to fog
	float: The value for:
	m_SkyFadeEnd; // when fogEnd less - no sky, just fog
	Return: none
	Code Example:
	System:SetSkyFade(self.Properties.xSkyStart, self.Properties.xSkyEnd);

`ActivatePortal`
`(CScriptObjectVector,`
` bool,`
` int)`
	Usage:
	Toggles a specified portal at a certain position. Recompute the sound occlusion to the opened or closed portal.
	Parameters:
	CScriptObjectVector: A position.
	bool: Enables or disables a portal.
	int: The entity id of the door / portal.
	Return: none
	Code Example:
	System:ActivatePortal(self:GetPos(),bOpen,self.id);

CScriptObjectSystem: C++ functions available in Lua script	

`DumpMMStats`
`()`
	Usage:
	Logs the current lua garbage collection count.
	Parameters: none
	Return: none
	Code Example: not used

`EnumDisplayFormats`
`()`
	Usage:
	Enumerates the possible display formats. Sets default values, if the renderer can not check propperly.
	Parameters: none
	Return:
	Returns a table with the possible display formats.

`EnumAAFormats`
`()`
	Usage:
	Enumerates the Full Screen AA modes.
	Parameters: none
	Return:
	Returns a table with the possible FSAA modes..

`IsPointIndoors`
`(CScriptObjectVector)`
	Usage:
	Checks if a point is indoors or not.
	Parameters:
	CScriptObjectVector: A position, a table in lua.
	Return:
	Returns a boolean, true or false. Set to false by default.
	Code Example:
	if (System:IsPointIndoors(Pos)) then
	...

`SetConsoleImage`
`(string,`
` bool)`
	Usage:
	Sets a new image for the console. Removes the old one.
	Parameters:
	string: The name of the new console image to set.
	bool: Remove the current image or not?
	Return: none
	Code Example: not used

CScriptObjectSystem: C++ functions available in Lua script

`ProjectToScreen`
`(CScriptObjectVector,`
` bool)`
	Usage:
	Projects a 3 dimensional vector to the 2 dimensional screen space.
	Parameters:
	CScriptObjectVector: A vector in 3d space.
	bool: [Optional] Use the modelview matrix for projection?
	Return: none
	Code Example:
	System:ProjectToScreen(vec);

`EnableHeatVision`
`(int)`
	Usage:
	Toggles the heat vision.
	Parameters:
	int: This is used like a boolean. 1 = true, 0 = false.
	Return: none
	Code Example: not used

`ShowDebugger`
`()`
	Usage:
	Shows the debugger.
	Parameters: none
	Return: none

`FrameProfiler`
`(bool,`
` bool,`
` string)`
	Usage:
	Sets the frame profiler.
	Parameters:
	bool: [Optional] Set to false by default. Enables or disables the profiler.
	bool: [Optional] Set to true by default. Shows the profiler output or not.
	string: [Optional] Set to ““ by default. Used as a prefix.
	For more, please check:
	ISystem::SetFrameProfiler(bool, bool, char*);
	Return: none
	Code Example: not used

CScriptObjectSystem: C++ functions available in Lua script	

`DumpMemStats`
`()`
	Usage:
	Dumps the memory usage statistics to the log.
	Parameters: none
	Return: none

`DumpWinHeaps`
`()`
	Usage:
	Dumps the windows heaps and dumps each of them with a special function to the log (using CryLogAlways()).
	Parameters: none
	Return: none

`Break`
`()`
	Usage:
	Writes an error string if WIN32 defined.
	CryError("CScriptObjectSystem:Break");
	Parameters: none
	Return:
	Returns none, does not call the error function if WIN32 not defined.

`DumpCommandsVars`
`(string)`
	Usage:
	Dumps the variables of a certain comman to the console.
	Parameters:
	string: [Optional] The command. Set to "" by default. Function returns, if no parameter was passed.
	Return: none
	Code Example: not used

`GetViewCameraPos`
`()`
	Usage:
	Gets the camera view position.
	Parameters: none
	Return:
	Returns a CscriptObjectVector, a table in lua.
	Code Example:
	local Indoor = System:IsPointIndoors(System:GetViewCameraPos());

CScriptObjectSystem: C++ functions available in Lua script

`RayWorldIntersection`
`(CScriptObjectVector,`
` CScriptObjectVector,`
` int,`
` int,`
` int,`
` int)`
	Usage:
	This function checks, if a ray with a certain position and a certain direction intersects with something.
	Parameters:
	CScriptObjectVector: The start position of the ray.
	CScriptObjectVector: The direction of the ray.
	int: The number of maximum hits.
	int: The number of entity types. (?)
	int: An entity with this id will be skipped.
	int: An entity with this id will be skipped.
	Return: none
	Code Example:
	local hits = System:RayWorldIntersection(pos, dest, 1,ent_static+ent_sleeping_rigid+ent_rigid+ent_independent+ent_terrain+ent_living,shooter.id);

`BrowseURL`
`()`
	Usage:
	Browses to a url. Platform dependant function.
	Does several security checks to make sure that this is really an url.
	Parameters:
	string: A url to browse to.
	Return: none
	Code Example:
	System:BrowseURL(UI.PageLoginDialog.szCreateAccountURL);

`IsDevModeEnable`
`()`
	Usage:
	This function checks if the game is running in devmode to make sure that some scripts like god-mode are allowed to be enabled.
	Parameters: none
	Return:
	Returns nil or != nil for true.
	Code Example: not used

CScriptObjectSystem: C++ functions available in Lua script

`RayTraceCheck`
`(CScriptObjectVector,`
`CScriptObjectVector,`
`int,`
`int)`
	Usage:
	Traces between two points to see if it collides with something.
	Parameters:
	CScriptObjectVector: Position one.
	CScriptObjectVector: Position two.
	int: Entity type to skip, pass the id here.
	int: Entity type to skip, pass the id here.
	Return:
	Returns true or false.
	Code Example:
	if (not System:RayTraceCheck(tbl.entity:GetPos(), safePos, self.id, tbl.entity.id)) then
	...

`SaveConfiguration`
`()`
	Usage:
	Saves the current configuration files.
	Parameters: none
	Return: none

CScriptObjectSystem: C++ functions available in Lua script

```
SetSystemShaderRenderFl
ags
(string,
 string,
 bool)
```

Usage:
Sets render flags for system shaders.

Parameters:
string: The name of the shader.

string: The name of a flag.

// Possible flags
DLD_TERRAIN_WATER, "DrawWater"
DLD_TERRAIN, "DrawTerrain"
DLD_DETAIL_TEXTURES, "DrawDetailTextures"
DLD_DETAIL_OBJECTS, "DrawDetailObjects"
DLD_FAR_SPRITES, "DrawFarSprites"
DLD_STATIC_OBJECTS, "DrawStaticObjects"
DLD_ENTITIES, "DrawEntities"
DLD_PARTICLES, "DrawParticles"
DLD_TERRAIN_LIGHT, "UseLights"
DLD_TERRAIN_FULLRES, "FullDetailTerrain"

bool: Enable or disable this flag? Works like a switch.

Return: none

Code Example: not