

DX9 Overlay API (de)

Erzeugt von Doxygen 1.8.7

Sam Mai 17 2014 19:26:58

Inhaltsverzeichnis

1	Datei-Verzeichnis	1
1.1	Auflistung der Dateien	1
2	Datei-Dokumentation	3
2.1	overlay.h-Dateireferenz	3
2.1.1	Dokumentation der Funktionen	4
2.1.1.1	BoxCreate	4
2.1.1.2	BoxDestroy	4
2.1.1.3	BoxSetBorder	4
2.1.1.4	BoxSetBorderColor	4
2.1.1.5	BoxSetColor	5
2.1.1.6	BoxSetHeight	5
2.1.1.7	BoxSetPos	5
2.1.1.8	BoxSetShown	6
2.1.1.9	BoxSetWidth	6
2.1.1.10	DestroyAllVisual	6
2.1.1.11	HideAllVisual	7
2.1.1.12	ImageCreate	7
2.1.1.13	ImageDestroy	7
2.1.1.14	ImageSetAlign	7
2.1.1.15	ImageSetPos	8
2.1.1.16	ImageSetRotation	9
2.1.1.17	ImageSetShown	9
2.1.1.18	Init	9
2.1.1.19	LineCreate	10
2.1.1.20	LineDestroy	11
2.1.1.21	LineSetColor	11
2.1.1.22	LineSetPos	11
2.1.1.23	LineSetShown	12
2.1.1.24	LineSetWidth	12
2.1.1.25	SetParam	12

2.1.1.26	ShowAllVisual	13
2.1.1.27	TextCreate	13
2.1.1.28	TextDestroy	13
2.1.1.29	TextSetColor	13
2.1.1.30	TextSetPos	14
2.1.1.31	TextSetShadow	14
2.1.1.32	TextSetShown	14
2.1.1.33	TextSetString	15
2.1.1.34	TextUpdate	15

Index	16
--------------	-----------

Kapitel 1

Datei-Verzeichnis

1.1 Auflistung der Dateien

Hier folgt die Aufzählung aller Dateien mit einer Kurzbeschreibung:

overlay.h	3
------------------	---

Kapitel 2

Datei-Dokumentation

2.1 overlay.h-Dateireferenz

Funktionen

- int **TextCreate** (const char *Font, int FontSize, bool bBold, bool bltalic, int x, int y, unsigned int color, const char *text, bool bShadow, bool bShow)
- int **TextDestroy** (int ID)
- int **TextSetShadow** (int id, bool b)
- int **TextSetShown** (int id, bool b)
- int **TextSetColor** (int id, unsigned int color)
- int **TextSetPos** (int id, int x, int y)
- int **TextSetString** (int id, const char *str)
- int **TextUpdate** (int id, const char *Font, int FontSize, bool bBold, bool bltalic)
- int **BoxCreate** (int x, int y, int w, int h, unsigned int dwColor, bool bShow)
- int **BoxDestroy** (int id)
- int **BoxSetShown** (int id, bool bShown)
- int **BoxSetBorder** (int id, int height, bool bShown)
- int **BoxSetBorderColor** (int id, unsigned int dwColor)
- int **BoxSetColor** (int id, unsigned int dwColor)
- int **BoxSetHeight** (int id, int height)
- int **BoxSetPos** (int id, int x, int y)
- int **BoxSetWidth** (int id, int width)
- int **LineCreate** (int x1, int y1, int x2, int y2, int width, unsigned int color, bool bShow)
- int **LineDestroy** (int id)
- int **LineSetShown** (int id, bool bShown)
- int **LineSetColor** (int id, unsigned int color)
- int **LineSetWidth** (int id, int width)
- int **LineSetPos** (int id, int x1, int y1, int x2, int y2)
- int **ImageCreate** (const char *path, int x, int y, int rotation, int align, bool bShow)
- int **ImageDestroy** (int id)
- int **ImageSetShown** (int id, bool bShown)
- int **ImageSetAlign** (int id, int align)
- int **ImageSetPos** (int id, int x, int y)
- int **ImageSetRotation** (int id, int rotation)
- int **DestroyAllVisual** ()
- int **ShowAllVisual** ()
- int **HideAllVisual** ()
- int **Init** ()
- void **SetParam** (const char *_szParamName, const char *_szParamValue)

2.1.1 Dokumentation der Funktionen

2.1.1.1 `int BoxCreate (int x, int y, int w, int h, unsigned int dwColor, bool bShow)`

Erstellt eine Box mit den angegebenen Parametern.

Parameter

<code>in</code>	<code>x</code>	Position X skaliert auf einer Auflösung von 800 x 600
<code>in</code>	<code>y</code>	Position Y skaliert auf einer Auflösung von 800 x 600
<code>in</code>	<code>w</code>	Breite der Box, skaliert auf einer Auflösung von 800 x 600
<code>in</code>	<code>h</code>	Höhe der Box, skaliert auf einer Auflösung von 800 x 600
<code>in</code>	<code>dwColor</code>	Farbe der Box
<code>in</code>	<code>bShow</code>	Sollte die Box nach dem Erstellen gezeichnet werden?

Rückgabe

Gibt die ID der Box zurück, -1 bei einem Fehler!

2.1.1.2 `int BoxDestroy (int id)`

Zerstört eine Box.

Parameter

<code>in</code>	<code>id</code>	ID der Box
-----------------	-----------------	------------

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

BoxCreate() (S. 4)

2.1.1.3 `int BoxSetBorder (int id, int height, bool bShown)`

Setzt eine Umrandung für die Box.

Parameter

<code>in</code>	<code>id</code>	ID der Box
<code>in</code>	<code>height</code>	Höhe der Umrandung.
<code>in</code>	<code>bShown</code>	true, falls die Umrandung gezeigt werden soll, false wenn nicht.

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

BoxCreate() (S. 4)

2.1.1.4 `int BoxSetBorderColor (int id, unsigned int dwColor)`

Setzt die Farbe der Umrandung einer Box.

Parameter

<i>in</i>	<i>id</i>	ID der Box
<i>in</i>	<i>dwColor</i>	Farbe für die Umrandung (ARGB-Code)

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

BoxCreate() (S. 4)

2.1.1.5 int BoxSetColor (int *id*, unsigned int *dwColor*)

Setzt die Farbe der Box.

Parameter

<i>in</i>	<i>id</i>	ID der Box
<i>in</i>	<i>dwColor</i>	Farbe der Box (ARGB-Code)

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

BoxCreate() (S. 4)

2.1.1.6 int BoxSetHeight (int *id*, int *height*)

Setzt die Höhe einer Box.

Parameter

<i>in</i>	<i>id</i>	ID der Box.
<i>in</i>	<i>height</i>	Höhe der Box.

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

BoxCreate() (S. 4)

2.1.1.7 int BoxSetPos (int *id*, int *x*, int *y*)

Setzt die Position einer Box.

Parameter

<i>in</i>	<i>id</i>	ID der Box.
<i>in</i>	<i>x</i>	Position-X, skaliert auf einer Auflösung von 800-600.
<i>in</i>	<i>y</i>	Position-Y, skaliert auf einer Auflösung von 800-600.

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

BoxCreate() (S. 4)

2.1.1.8 int BoxSetShown (int *id*, bool *bShown*)

Aktiviert / Deaktiviert eine Box temporär.

Parameter

<i>in</i>	<i>id</i>	ID der Box
<i>in</i>	<i>bShown</i>	true, falls die Box gezeigt werden soll, false wenn nicht.

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

BoxCreate() (S. 4)

2.1.1.9 int BoxSetWidth (int *id*, int *width*)

Setzt die Breite einer Box.

Parameter

<i>in</i>	<i>id</i>	ID der Box
<i>in</i>	<i>width</i>	Breite der Box

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

BoxCreate() (S. 4)

2.1.1.10 int DestroyAllVisual ()

Zerstört alle Overlays.

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

2.1.1.11 int HideAllVisual ()

Versteckt alle Overlays.

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

2.1.1.12 int ImageCreate (const char * *path*, int *x*, int *y*, int *rotation*, int *align*, bool *bShow*)

Erstellt ein Bild mit den angegebenen Paramtern.

Parameter

in	<i>path</i>	Relativer oder absoluter Pfad zur einer Bild-Datei.
in	<i>x</i>	X-Position des ersten Punktes, skaliert auf einer Auflösung von 800 x 600
in	<i>y</i>	Y-Position des ersten Punktes, skaliert auf einer Auflösung von 800 x 600
in	<i>rotation</i>	Rotation des Bildes.
in	<i>align</i>	Platzierung des Bildes.
in	<i>bShow</i>	true, falls das Bild beim initialisieren gezeichnet werden soll, false wenn nicht.

Rückgabe

Gibt die ID zurück, -1 bei einem Fehler.

2.1.1.13 int ImageDestroy (int *id*)

zerstört ein Bild.

Parameter

in	<i>id</i>	ID des Bildes, welches zerstört werden soll.
----	-----------	--

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

ImageCreate() (S. 7)

2.1.1.14 int ImageSetAlign (int *id*, int *align*)

Ordnet das Bild zu.

Parameter

in	<i>id</i>	ID des Bildes.
in	<i>align</i>	Zuordnung.

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

ImageCreate() (S. 7)

2.1.1.15 `int ImageSetPos (int id, int x, int y)`

Setzt die Position eines Bildes.

Parameter

<i>in</i>	<i>id</i>	ID des Bildes.
<i>in</i>	<i>x</i>	X-Position des ersten Punktes, skaliert auf einer Auflösung von 800 x 600
<i>in</i>	<i>y</i>	Y-Position des ersten Punktes, skaliert auf einer Auflösung von 800 x 600

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

ImageCreate() (S. 7)

2.1.1.16 `int ImageSetRotation (int id, int rotation)`

Setzt die Rotation eines Bildes.

Parameter

<i>in</i>	<i>id</i>	ID des Bildes.
<i>in</i>	<i>rotation</i>	Rotation des Bildes.

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

ImageCreate() (S. 7)

2.1.1.17 `int ImageSetShown (int id, bool bShown)`

Aktiviert / Deaktiviert ein Bild temporär.

Parameter

<i>in</i>	<i>id</i>	ID des Bildes.
<i>in</i>	<i>bShown</i>	true, falls das Bild gezeigt werden soll, false wenn nicht.

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

ImageCreate() (S. 7)

2.1.1.18 `int Init ()`

Initialisiert die API, muss aber nicht aufgerufen werden, da sie intern aufgerufen wird.

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

2.1.1.19 `int LineCreate (int x1, int y1, int x2, int y2, int width, unsigned int color, bool bShow)`

Erstellt eine Linie mit den angegebenen Parametern.

Parameter

<i>in</i>	<i>x1</i>	X-Position des ersten Punktes, skaliert auf einer Auflösung von 800 x 600
<i>in</i>	<i>y1</i>	Y-Position des ersten Punktes, skaliert auf einer Auflösung von 800 x 600
<i>in</i>	<i>x2</i>	X-Position des zweiten Punktes, skaliert auf einer Auflösung von 800 x 600
<i>in</i>	<i>y2</i>	Y-Position des zweiten Punktes, skaliert auf einer Auflösung von 800 x 600
<i>in</i>	<i>width</i>	Breite der Linie in Pixeln.
<i>in</i>	<i>color</i>	Farbe der Linie.
<i>in</i>	<i>bShow</i>	Sollte die Linie direkt nach dem Erstellen gezeichnet werden?

Rückgabe

Gibt die ID der erstellten Linie zurück. -1 bei einem Fehler.

2.1.1.20 `int LineDestroy (int id)`

Zerstört eine Linie.

Parameter

<i>in</i>	<i>id</i>	Linien-Objekt, welches zerstört werden soll.
-----------	-----------	--

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

2.1.1.21 `int LineSetColor (int id, unsigned int color)`

Setzt die Farbe einer Linie.

Parameter

<i>in</i>	<i>id</i>	ID der Linie
<i>in</i>	<i>color</i>	Farbe der Linie (ARGB-Code)

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

LineCreate() (S. 10)

2.1.1.22 `int LineSetPos (int id, int x1, int y1, int x2, int y2)`

Setzt die Position einer Linie

Parameter

<i>in</i>	<i>id</i>	ID der Linie
<i>in</i>	<i>x1</i>	X-Position des ersten Punktes, skaliert auf einer Auflösung von 800 x 600

<i>in</i>	<i>y1</i>	Y-Position des ersten Punktes, skaliert auf einer Auflösung von 800 x 600
<i>in</i>	<i>x2</i>	X-Position des zweiten Punktes, skaliert auf einer Auflösung von 800 x 600
<i>in</i>	<i>y2</i>	Y-Position des zweiten Punktes, skaliert auf einer Auflösung von 800 x 600

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

LineCreate() (S. 10)

2.1.1.23 `int LineSetShown (int id, bool bShown)`

Aktiviert / Deaktiviert eine Linie temporär.

Parameter

<i>in</i>	<i>id</i>	ID der Linie
<i>in</i>	<i>bShown</i>	true, falls die Linie gezeigt werden soll, false wenn nicht.

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

LineCreate() (S. 10)

2.1.1.24 `int LineSetWidth (int id, int width)`

Setzt die Breite einer Linie.

Parameter

<i>in</i>	<i>id</i>	ID der Linie
<i>in</i>	<i>width</i>	Breite der Linie in Pixeln.

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

LineCreate() (S. 10)

2.1.1.25 `void SetParam (const char * _szParamName, const char * _szParamValue)`

Setzt die Parameter für die API.

Mögliche Parameter für '*_szParamName*':

"use_window" Die API benutzt den Fensternamen, um einen Prozess zu finden. Standardmäßig auf 0.

"process" Name des Prozesses, wird ignoriert, wenn use_window auf 1 ist.

"window" Name des Fensters.

Parameter

in	<i>_szParamName</i>	Schlüsselparameter
in	<i>_szParamValue</i>	Wert für den ersten Parameter. Diese Funktion muss vor jeder anderen API-Funktion aufgerufen werden!

2.1.1.26 int ShowAllVisual ()

Zeigt alle Overlays.

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

2.1.1.27 int TextCreate (const char * *Font*, int *FontSize*, bool *bBold*, bool *bItalic*, int *x*, int *y*, unsigned int *color*, const char * *text*, bool *bShadow*, bool *bShow*)

Erstellt einen Text mit den angegebenen Parametern.

Parameter

in	<i>Font</i>	Name einer Font
in	<i>FontSize</i>	Größe der Font
in	<i>bBold</i>	Sollte die Font fett sein?
in	<i>bItalic</i>	Sollte die Font kursiv sein?
in	<i>x</i>	Position X skaliert auf einer Auflösung von 800 x 600
in	<i>y</i>	Position Y skaliert auf einer Auflösung von 800 x 600
in	<i>color</i>	Farbe des Overlays (ARGB-Code)
in	<i>text</i>	Text des Overlays
in	<i>bShadow</i>	Schwarzer Schatten um das Overlay?
in	<i>bShow</i>	Sollte der Text nach dem Funktionsaufruf gezeichnet werden?

Rückgabe

Gibt die ID des Overlays zurück, welche für alle weiteren Funktionen benötigt wird. Gibt -1 bei einem Fehler zurück.

2.1.1.28 int TextDestroy (int *ID*)

Zerstört einen Text.

Parameter

in	<i>ID</i>	TextID welche zerstört werden sollte.
----	-----------	---------------------------------------

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

TextCreate() (S. 13)

2.1.1.29 int TextSetColor (int *id*, unsigned int *color*)

Setzt eine neue Farbe für den Text

Parameter

<i>in</i>	<i>id</i>	Overlay-ID
<i>in</i>	<i>color</i>	Die neue Farbe als ARGB-Code

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

TextCreate() (S. 13)

2.1.1.30 `int TextSetPos (int id, int x, int y)`

Setzt den Text an eine angegebene Position

Parameter

<i>in</i>	<i>id</i>	Overlay-ID
<i>in</i>	<i>x</i>	X-Position skaliert auf einer Auflösung von 800 x 600
<i>in</i>	<i>y</i>	Y-Position skaliert auf einer Auflösung von 800 x 600

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

TextCreate() (S. 13)

2.1.1.31 `int TextSetShadow (int id, bool b)`

Aktiviert / Deaktiviert den Schatten eines Textes.

Parameter

<i>in</i>	<i>id</i>	Overlay-ID
<i>in</i>	<i>b</i>	true für das Aktivieren, false für das Deaktivieren

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

TextCreate() (S. 13)

2.1.1.32 `int TextSetShown (int id, bool b)`

Aktiviert / Deaktiviert das Overlay temporär.

Parameter

<i>in</i>	<i>id</i>	Overlay-ID
<i>in</i>	<i>b</i>	true für das Aktivieren, false für das Deaktivieren

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

TextCreate() (S. 13)

2.1.1.33 `int TextSetString (int id, const char * str)`

Setzen eines neues Textes für das Text-Overlay.

Parameter

<i>in</i>	<i>id</i>	Overlay-ID
<i>in</i>	<i>str</i>	Neuer Text als String

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

TextCreate() (S. 13)

2.1.1.34 `int TextUpdate (int id, const char * Font, int FontSize, bool bBold, bool bltalic)`

Aktualisieren des Overlays mit neuen Daten.

Parameter

<i>in</i>	<i>id</i>	Overlay-ID
<i>in</i>	<i>Font</i>	Neue Font als String
<i>in</i>	<i>FontSize</i>	Größe der Font
<i>in</i>	<i>bBold</i>	Sollte der Text fett sein?
<i>in</i>	<i>bltalic</i>	Sollte der Text kursiv sein?

Rückgabe

1 wenn erfolgreich, 0 wenn nicht.

Siehe auch

TextCreate() (S. 13)

Index

BoxCreate
 overlay.h, 4
BoxDestroy
 overlay.h, 4
BoxSetBorder
 overlay.h, 4
BoxSetBorderColor
 overlay.h, 4
BoxSetColor
 overlay.h, 5
BoxSetHeight
 overlay.h, 5
BoxSetPos
 overlay.h, 5
BoxSetShown
 overlay.h, 6
BoxSetWidth
 overlay.h, 6

DestroyAllVisual
 overlay.h, 6

HideAllVisual
 overlay.h, 6

ImageCreate
 overlay.h, 7
ImageDestroy
 overlay.h, 7
ImageSetAlign
 overlay.h, 7
ImageSetPos
 overlay.h, 7
ImageSetRotation
 overlay.h, 9
ImageSetShown
 overlay.h, 9
Init
 overlay.h, 9

LineCreate
 overlay.h, 9
LineDestroy
 overlay.h, 11
LineSetColor
 overlay.h, 11
LineSetPos
 overlay.h, 11
LineSetShown
 overlay.h, 12

LineSetWidth
 overlay.h, 12

overlay.h, 3
 BoxCreate, 4
 BoxDestroy, 4
 BoxSetBorder, 4
 BoxSetBorderColor, 4
 BoxSetColor, 5
 BoxSetHeight, 5
 BoxSetPos, 5
 BoxSetShown, 6
 BoxSetWidth, 6
 DestroyAllVisual, 6
 HideAllVisual, 6
 ImageCreate, 7
 ImageDestroy, 7
 ImageSetAlign, 7
 ImageSetPos, 7
 ImageSetRotation, 9
 ImageSetShown, 9
 Init, 9
 LineCreate, 9
 LineDestroy, 11
 LineSetColor, 11
 LineSetPos, 11
 LineSetShown, 12
 LineSetWidth, 12
 SetParam, 12
 ShowAllVisual, 13
 TextCreate, 13
 TextDestroy, 13
 TextSetColor, 13
 TextSetPos, 14
 TextSetShadow, 14
 TextSetShown, 14
 TextSetString, 15
 TextUpdate, 15

SetParam
 overlay.h, 12
ShowAllVisual
 overlay.h, 13

TextCreate
 overlay.h, 13
TextDestroy
 overlay.h, 13
TextSetColor
 overlay.h, 13

TextSetPos
 overlay.h, 14
TextSetShadow
 overlay.h, 14
TextSetShown
 overlay.h, 14
TextSetString
 overlay.h, 15
TextUpdate
 overlay.h, 15