Delta3D Version 2.4.0

dtGUI::

Reference Manual

Generated for Delta3D by Doxygen, Graphviz and MikTex

Sat Jan 9 23:28:47 2010

Contents

Chapter 1

Main Page

Delta3D is an Open Source engine which can be used for games, simulations, or other graphical applications.

The Delta3D framework exists as a number of modules, each sitting in its own library, enclosed within its own namespace. At the very core lies the dtCore library. This contains basic, low-level functionality which is mostly required for all 3D applications written in C++.

Around and alongside this sit other supporting libraries, such as dtUtil (containing reusable features which are useful for most applications), dtTerrain (for rendering terrain databases), dtGame, dtNet, etc.

Extensive online documentation is available from the Delta3D Docs section to help in using Delta3D.

The project's original reference guides generated by Doxygen from the source code may be viewed at the Delta3D API Documentation section.

To download source code, binaries, dependencies and sample datasets visit the Delta3D Downloads page.

For more about dependencies see the Delta3D Dependencies page.

The documentation you are looking at can be downloaded from www.3draum.ch.

Enjoy!

Chapter 2

Todo List

Chapter 3

Directory Documentation

3.1 src/dtGUI/ Directory Reference

Files

- file ceguiconnectionmanager.cpp
- file ceguikeyboardlistener.cpp
- file ceguimouselistener.cpp
- file ceguirenderer.cpp
- file ceguitexture.cpp
- file ceuidrawable.cpp
- file guiexceptionenum.cpp
- file hud.cpp
- file renderer.cpp
- file resourceprovider.cpp
- file scriptmodule.cpp

3.2 inc/dtGUI/ Directory Reference

Files

- file basescriptmodule.h
- file ceguiconnectionmanager.h
- file ceguiconnectionsignatures.h
- file ceguikeyboardlistener.h
- file ceguimouselistener.h
- file ceguirenderer.h
- file ceguitexture.h
- file ceuidrawable.h
- file dtgui.h
- file export.h
- file guiexceptionenum.h
- file hud.h
- file mainpage.h
- file renderer.h
- file resourceprovider.h
- file scriptmodule.h
- file widget.h

3.3 inc/ Directory Reference

Directories

- directory dtGUI

3.4 src/ Directory Reference

Directories

- directory dtGUI

Chapter 4

Namespace Documentation

4.1 dtCore Namespace Reference

4.1.1 Detailed Description

4.2 dtGUI Namespace Reference

The dtGUI namespace contains the functionality to render and control OpenGL graphical user interfaces.

Classes

- class BaseScriptModule
  An abstract interface for classes used by CEUIDrawable.
- class CEGUIConnectionManager
  this class is used to keep track of connections (event-callbacks) for any widget/EventSet created for/by a gui-object.
- class CEGUIKeyboardListener
- class CEGUIMemberSignature
  objects instantiated from this class describes an object-member-"signature"
- class CEGUIMemberSignatureBase
  Common interface for a member-callback-connection-signatures.
- class CEGUIMouseListener
  A mouse device listener to inject input to CEGUI.
- class CEGUIRenderer
- class CEGUISignatureStatic
  objects instantiated from this class describes a static-method's signature
- class CEGUITexture
  implementation of the CEGUI::Texture for OpenSceneGraph
- class CEUIDrawable
  A DeltaDrawable used to render CEGUI.
- class ExceptionEnum
- class HUD
  gui-class for rendering and managing widgets
- class Renderer
  Simple CEGui rendering class based on the CEGUI::OpenGLRender class.
- class ResourceProvider
  A simple CEGUI ResourceProvider based on the CEGUI::DefaultResourceProvider.
- class ScriptModule
  ScriptModule is the binding from CEGUI::Events to specific application callbacks.

Typedefs

- typedef CEGUI::Window Widget

Functions

- IMPLEMENT_ENUM (ExceptionEnum)

4.2.1 Detailed Description

The dtGUI namespace contains the functionality to render and control OpenGL graphical user interfaces. dtGUI uses `CEGUI` for high-level control of the widgets.

4.2.2 Typedef Documentation

4.2.2.1 typedef CEGUI::Window Widget

4.2.3 Function Documentation

4.2.3.1 dtGUI::IMPLEMENT_ENUM (ExceptionEnum)

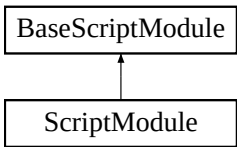
Chapter 5

Class Documentation

5.1 BaseScriptModule Class Reference

An abstract interface for classes used by CEUIDrawable.

`#include <inc/dtGUI/basescriptmodule.h>`Inheritance diagram for BaseScriptModule::



Public Member Functions

- BaseScriptModule ()
- virtual ~BaseScriptModule ()
- virtual bool executeScriptedEventHandler (const CEGUI::String &handlerName, const CEGUI::EventArgs &ea)=0

  Overload this function to handle Events triggered from a CEGUI::Window.

- virtual void executeScriptFile (const CEGUI::String &filename, const CEGUI::String &resourceGroup="")=0
- virtual int executeScriptGlobal (const CEGUI::String &function_name)=0
- virtual void executeString (const CEGUI::String &str)=0

5.1.1 Detailed Description

An abstract interface for classes used by CEUIDrawable. BaseScriptModule is mean to be implemented to provide support for Events triggered by the CEGUI Windows in a GUI scene. See also dtGUI::ScriptModule.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 BaseScriptModule () `[inline]`

5.1.2.2 virtual ~BaseScriptModule () `[inline, virtual]`

5.1.3 Member Function Documentation

5.1.3.1 virtual bool executeScriptedEventHandler (const CEGUI::String & *handlerName*, const CEGUI::EventArgs & *ea*) `[pure virtual]`

Overload this function to handle Events triggered from a CEGUI::Window. Parameters

handlerName the name of something to handle this event, typically a function name.

ea the CEGUI::EventArgs that can contain useful information about the CEGUI::Event that occurred.

Implemented in ScriptModule.

5.1.3.2 virtual void executeScriptFile (const CEGUI::String & *filename*, const CEGUI::String & *resourceGroup* = "") [pure virtual]

Implemented in ScriptModule.

5.1.3.3 virtual int executeScriptGlobal (const CEGUI::String & *function_name*) [pure virtual]

Implemented in ScriptModule.

5.1.3.4 virtual void executeString (const CEGUI::String & *str*) [pure virtual]

Implemented in ScriptModule.

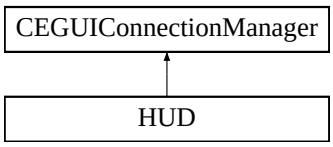
The documentation for this class was generated from the following file:

- basescriptmodule.h

5.2 CEGUIConnectionManager Class Reference

this class is used to keep track of connections (event-callbacks) for any widget/EventSet created for/by a gui-object.

`#include <inc/dtGUI/ceguiconnectionmanager.h>`Inheritance diagram for CEGUIConnectionManager::



Public Member Functions

- void Connect (CEGUI::EventSet *pEventSet, const std::string &sEventName, bool(*pFunction)(const CEGUI::EventArgs &))

  connects static function to an event

- template<class T >
  void Connect (CEGUI::EventSet *pEventSet, const std::string &sEventName, bool(T::*pObjectMember)(const CEGUI::EventArgs &), T *pObject)

  connects object-member from an event

- void Disconnect (CEGUI::EventSet *pEventSet, const std::string &sEventName, bool(*pFunction)(const CEGUI::EventArgs &))

  disconnects static functions from an event

- template<class T >
  void Disconnect (CEGUI::EventSet *pEventSet, const std::string &sEventName, bool(T::*pObjectMember)(const CEGUI::EventArgs &), T *pObject)

  disconnects object-member from an event

Protected Member Functions

- CEGUIConnectionManager ()

  constructs the a connection-manager

- virtual ~CEGUIConnectionManager ()

  destructs/clean up the connection-manager

5.2.1 Detailed Description

this class is used to keep track of connections (event-callbacks) for any widget/EventSet created for/by a gui-object.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 CEGUIConnectionManager () `[inline, protected]`

constructs the a connection-manager

5.2.2.2 ~CEGUIConnectionManager () `[protected, virtual]`

destructs/clean up the connection-manager

5.2.3 Member Function Documentation

5.2.3.1 void Connect (CEGUI::EventSet * *pEventSet*, const std::string & *sEventName*, bool(*)(const CEGUI::EventArgs &) *pFunction*)

connects static function to an event Parameters

pEventSet CETUIEventSet which includes the event

sEventName name of the event to be connected

pFunction addr of static function

5.2.3.2 void Connect (CEGUI::EventSet * *pEventSet*, const std::string & *sEventName*, bool(T::*)(const CEGUI::EventArgs &) *pObjectMember*, T * *pObject*) [inline]

connects object-member from an event Parameters

pEventSet CEGUIEventSet which includes the event

sEventName name of the event to be connected

pObjectMember (relative)addr of object member

pObject pointer to the instantiated object which owns the callback

5.2.3.3 void Disconnect (CEGUI::EventSet * *pEventSet*, const std::string & *sEventName*, bool(*)(const CEGUI::EventArgs &) *pFunction*)

disconnects static functions from an event Parameters

pEventSet CEGUIEventSet which includes the event

sEventName name of the CEGUIEvent

pFunction addr of static function

5.2.3.4 void Disconnect (CEGUI::EventSet * *pEventSet*, const std::string & *sEventName*, bool(T::*)(const CEGUI::EventArgs &) *pObjectMember*, T * *pObject*) [inline]

disconnects object-member from an event Parameters

pEventSet CEGUIEventSet which includes the event

sEventName name of the CEGUIEvent

pObjectMember (relative)addr of object member

pObject pointer to the instantiated object which owns the callback

The documentation for this class was generated from the following files:

- ceguiconnectionmanager.h
- ceguiconnectionmanager.cpp

5.3 CEGUIKeyboardListener Class Reference

`#include <inc/dtGUI/ceguikeyboardlistener.h>`

Public Member Functions

- CEGUIKeyboardListener (HUD *pGUI=NULL)
- bool HandleKeyPressed (const dtCore::Keyboard *keyboard, int key)
- bool HandleKeyReleased (const dtCore::Keyboard *keyboard, int key)
- bool HandleKeyTyped (const dtCore::Keyboard *keyboard, int key)

Static Public Member Functions

- static CEGUI::Key::Scan KeyboardKeyToKeyScan (int key)

  Determines the CEGUI scancode that corresponds to the specified Producer::KeyboardKey.

Protected Member Functions

- virtual ~CEGUIKeyboardListener ()

5.3.1 Constructor & Destructor Documentation

5.3.1.1 CEGUIKeyboardListener (HUD * *pGUI* = `NULL`)

5.3.1.2 ~CEGUIKeyboardListener () `[protected, virtual]`

5.3.2 Member Function Documentation

5.3.2.1 bool HandleKeyPressed (const dtCore::Keyboard * *keyboard*, int *key*)

5.3.2.2 bool HandleKeyReleased (const dtCore::Keyboard * *keyboard*, int *key*)

5.3.2.3 bool HandleKeyTyped (const dtCore::Keyboard * *keyboard*, int *key*)

Todo

what is this supposed to do? inject both keypressed and keyreleased to CEGUI::System? Haven't those already been injected with the other calls?

5.3.2.4 CEGUI::Key::Scan KeyboardKeyToKeyScan (int *key*) `[static]`

Determines the CEGUI scancode that corresponds to the specified Producer::KeyboardKey. Parameters

key the key to map

Returns the corresponding CEGUI key scancode

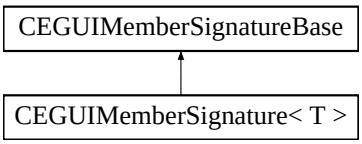
The documentation for this class was generated from the following files:

- ceguikeyboardlistener.h
- ceguikeyboardlistener.cpp

5.4 CEGUIMemberSignature< T > Class Template Reference

objects instantiated from this class describes an object-member-"signature"

`#include <inc/dtGUI/ceguiconnectionsignatures.h>`Inheritance diagram for CEGUIMemberSignature< T >::



Public Types

- typedef bool(T::* CALLBACK_FPTR)(const CEGUI::EventArgs &)

Public Member Functions

- CEGUIMemberSignature (CEGUI::EventSet *pEventSet, const std::string &sEventName, const CALLBACK_FPTR pMemberFunction, const void *pObject)

  construcs a connection-signature-object

- const CALLBACK_FPTR GetFunctionPtr () const

  returns the pointer to the object-member

- const void * GetObjectPtr () const

  returns the pointer to the object

- bool match (CEGUI::EventSet *pEventSet, const std::string &sEventName, const CALLBACK_FPTR pMemberFunction, const void *pObject) const

  check if one signature "matches" another (empty strings and NULL-Pointer's are Wildcards, means: "match any value")

- virtual bool operator== (const CEGUIMemberSignatureBase &x) const

  compares member-signatures

5.4.1 Detailed Description

template<class T> class dtGUI::CEGUIMemberSignature< T >

objects instantiated from this class describes an object-member-"signature"

5.4.2 Member Typedef Documentation

5.4.2.1 typedef bool(T::* CALLBACK_FPTR)(const CEGUI::EventArgs &)

5.4.3 Constructor & Destructor Documentation

5.4.3.1 CEGUIMemberSignature (CEGUI::EventSet * *pEventSet*, const std::string & *sEventName*, const CALLBACK_FPTR *pMemberFunction*, const void * *pObject*) [inline]

construcs a connection-signature-object

5.4.4 Member Function Documentation

5.4.4.1 const CALLBACK_FPTR GetFunctionPtr () const [inline]

returns the pointer to the object-member

5.4.4.2 const void* GetObjectPtr () const [inline]

returns the pointer to the object

5.4.4.3 bool match (CEGUI::EventSet * *pEventSet*, const std::string & *sEventName*, const CALLBACK_FPTR *pMemberFunction*, const void * *pObject*) const [inline]

check if one signature "matches" another (empty strings and NULL-Pointer's are Wildcards, means: "match any value")

5.4.4.4 virtual bool operator== (const CEGUIMemberSignatureBase &) const [inline, virtual]

compares member-signatures

Implements CEGUIMemberSignatureBase.

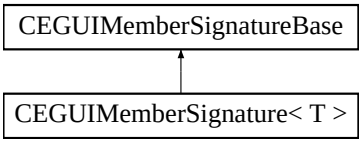
The documentation for this class was generated from the following file:

- ceguiconnectionsignatures.h

5.5 CEGUIMemberSignatureBase Class Reference

Common interface for a member-callback-connection-signatures.

`#include <inc/dtGUI/ceguiconnectionsignatures.h>`Inheritance diagram for CEGUIMemberSignatureBase::



Public Member Functions

- CEGUIMemberSignatureBase (CEGUI::EventSet *pEventSet, const std::string &sEventName)
  constructs a signature for an object-member-function
- const std::string & GetEventName () const
  returns the event name which is part of the signature
- const CEGUI::EventSet * GetEventSet () const
  returns the event set which is part of the signature
- virtual bool operator== (const CEGUIMemberSignatureBase &) const =0
  compares member-signatures

Protected Member Functions

- virtual ~CEGUIMemberSignatureBase ()

Protected Attributes

- CEGUI::EventSet * m_pEventSet
- std::string m_sEventName

Friends

- class CEGUIConnectionManager

5.5.1 Detailed Description

Common interface for a member-callback-connection-signatures.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 CEGUIMemberSignatureBase (CEGUI::EventSet * *pEventSet*, const std::string & *sEventName*) [inline]

constructs a signature for an object-member-function

5.5.2.2 virtual ~CEGUIMemberSignatureBase () [inline, protected, virtual]

5.5.3 Member Function Documentation

5.5.3.1 const std::string& GetEventName () const [inline]

returns the event name which is part of the signature

5.5.3.2 const CEGUI::EventSet* GetEventSet () const [inline]

returns the event set which is part of the signature

5.5.3.3 virtual bool operator== (const CEGUIMemberSignatureBase &) const [pure virtual]

compares member-signatures

Implemented in CEGUIMemberSignature< T >.

5.5.4 Friends And Related Function Documentation

5.5.4.1 friend class CEGUIConnectionManager [friend]

5.5.5 Member Data Documentation

5.5.5.1 CEGUI::EventSet* m_pEventSet [protected]

5.5.5.2 std::string m_sEventName [protected]

The documentation for this class was generated from the following file:

- ceguiconnectionsignatures.h

5.6 CEGUIMouseListener Class Reference

A mouse device listener to inject input to CEGUI.

`#include <inc/dtGUI/ceguimouselistener.h>`

Public Member Functions

- CEGUIMouseListener (HUD *pGui=NULL)
- bool HandleButtonClicked (const dtCore::Mouse *mouse, dtCore::Mouse::MouseButton button, int clickCount)
- bool HandleButtonPressed (const dtCore::Mouse *mouse, dtCore::Mouse::MouseButton button)
- bool HandleButtonReleased (const dtCore::Mouse *mouse, dtCore::Mouse::MouseButton button)
- bool HandleMouseDragged (const dtCore::Mouse *mouse, float x, float y)
- bool HandleMouseMoved (const dtCore::Mouse *mouse, float x, float y)
- bool HandleMouseScrolled (const dtCore::Mouse *mouse, int delta)
- void SetWindowSize (unsigned int width, unsigned int height)

Protected Member Functions

- ~CEGUIMouseListener ()

5.6.1 Detailed Description

A mouse device listener to inject input to CEGUI.

Todo

Why maintain mWidth AND mHalfWidth?

5.6.2 Constructor & Destructor Documentation

5.6.2.1 CEGUIMouseListener (HUD * *pGui* = `NULL`)

5.6.2.2 ~CEGUIMouseListener () `[protected]`

5.6.3 Member Function Documentation

5.6.3.1 bool HandleButtonClicked (const dtCore::Mouse * *mouse*, dtCore::Mouse::MouseButton *button*, int *clickCount*)

Todo

what is this supposed to do? inject both buttonpressed and buttonreleased to CEGUI::System? Haven't those already been injected with the other calls?

5.6.3.2 bool HandleButtonPressed (const dtCore::Mouse * *mouse*, dtCore::Mouse::MouseButton *button*)

Todo

test System for null, throw exception if null.

5.6.3.3 bool HandleButtonReleased (const dtCore::Mouse * *mouse*, dtCore::Mouse::MouseButton *button*)

Todo

test System for null, throw exception if null.

5.6.3.4 bool HandleMouseDragged (const dtCore::Mouse * *mouse*, float *x*, float *y*)

Todo

test System for null, throw exception if null.

5.6.3.5 bool HandleMouseMoved (const dtCore::Mouse * *mouse*, float *x*, float *y*)

Todo

test System for null, throw exception if null.

Todo

document these magic constants from the CEUIDrawable-days.

5.6.3.6 bool HandleMouseScrolled (const dtCore::Mouse * *mouse*, int *delta*)

Todo

test System for null, throw exception if null.

5.6.3.7 void SetWindowSize (unsigned int *width*, unsigned int *height*)

The documentation for this class was generated from the following files:

- ceguimouselistener.h
- ceguimouselistener.cpp

5.7 CEGUIRenderer Class Reference

`#include <inc/dtGUI/ceguirenderer.h>`

Classes

- struct **MyQuad**
- struct **QuadInfo**

Public Member Functions

- CEGUIRenderer (CEGUI::uint max_quads, int width, int height, CEGUI::ImageCodec *codec=0)
- CEGUIRenderer (CEGUI::uint max_quads, CEGUI::ImageCodec *codec=0)
- virtual ~CEGUIRenderer (void)
- virtual void addQuad (const CEGUI::Rect &dest_rect, float z, const CEGUI::Texture *tex, const CEGUI::Rect &texture_rect, const CEGUI::ColourRect &colours, CEGUI::QuadSplitMode quam_split_mode)
- virtual void clearRenderList (void)
- virtual CEGUI::Texture * createTexture (float size)
- virtual CEGUI::Texture * createTexture (const CEGUI::String &filename, const CEGUI::String &resourceGroup)
- virtual CEGUI::Texture * createTexture (void)
- virtual void destroyAllTextures (void)
- virtual void destroyTexture (CEGUI::Texture *texture)
- virtual void doRender (void)
- virtual float getHeight (void) const
- virtual CEGUI::uint getHorzScreenDPI (void) const
- CEGUI::ImageCodec & getImageCodec (void)
- virtual CEGUI::uint getMaxTextureSize (void) const
- virtual CEGUI::Rect getRect (void) const
- virtual CEGUI::Size getSize (void) const
- virtual CEGUI::uint getVertScreenDPI (void) const
- virtual float getWidth (void) const
- virtual bool isQueueingEnabled (void) const
- void setDisplaySize (const CEGUI::Size &sz)
- void SetGraphicsContext (osg::GraphicsContext *)
- void setImageCodec (CEGUI::ImageCodec *codec)
- void setImageCodec (const CEGUI::String &codecName)
- virtual void setQueueingEnabled (bool setting)

Static Public Member Functions

- static const CEGUI::String & getDefaultImageCodecName ()
- static void setDefaultImageCodecName (const CEGUI::String &codecName)

5.7.1 Constructor & Destructor Documentation

5.7.1.1 CEGUIRenderer (CEGUI::uint *max_quads*, CEGUI::ImageCodec * *codec* = 0)

5.7.1.2 CEGUIRenderer (CEGUI::uint *max_quads*, int *width*, int *height*, CEGUI::ImageCodec * *codec* = 0)

5.7.1.3 ~CEGUIRenderer (void) [virtual]

5.7.2 Member Function Documentation

5.7.2.1 void addQuad (const CEGUI::Rect & *dest_rect*, float *z*, const CEGUI::Texture * *tex*, const CEGUI::Rect & *texture_rect*, const CEGUI::ColourRect & *colours*, CEGUI::QuadSplitMode *quam_split_mode*) [virtual]

5.7.2.2 void clearRenderList (void) [virtual]

5.7.2.3 CEGUI::Texture * createTexture (float *size*) [virtual]

5.7.2.4 CEGUI::Texture * createTexture (const CEGUI::String & *filename*, const CEGUI::String & *resourceGroup*) [virtual]

5.7.2.5 CEGUI::Texture * createTexture (void) [virtual]

5.7.2.6 void destroyAllTextures (void) [virtual]

5.7.2.7 void destroyTexture (CEGUI::Texture * *texture*) [virtual]

5.7.2.8 void doRender (void) [virtual]

5.7.2.9 const CEGUI::String & getDefaultImageCodecName () [static]

5.7.2.10 virtual float getHeight (void) const [inline, virtual]

5.7.2.11 virtual CEGUI::uint getHorzScreenDPI (void) const [inline, virtual]

5.7.2.12 CEGUI::ImageCodec & getImageCodec (void)

5.7.2.13 virtual CEGUI::uint getMaxTextureSize (void) const [inline, virtual]

5.7.2.14 virtual CEGUI::Rect getRect (void) const [inline, virtual]

5.7.2.15 virtual CEGUI::Size getSize (void) const [inline, virtual]

5.7.2.16 virtual CEGUI::uint getVertScreenDPI (void) const [inline, virtual]

5.7.2.17 virtual float getWidth (void) const [inline, virtual]

5.7.2.18 virtual bool isQueueingEnabled (void) const [inline, virtual]

5.7.2.19 void setDefaultImageCodecName (const CEGUI::String & *codecName*) [static]

5.7.2.20 void setDisplaySize (const CEGUI::Size & *sz*)

5.7.2.21 void SetGraphicsContext (osg::GraphicsContext * *gc*)

5.7.2.22 void setImageCodec (CEGUI::ImageCodec * *codec*)

5.7.2.23 void setImageCodec (const CEGUI::String & *codecName*)

5.7.2.24 virtual void setQueueingEnabled (bool *setting*) [inline, virtual]

The documentation for this class was generated from the following files:

- ceguirenderer.h
- ceguirenderer.cpp

5.8 CEGUISignatureStatic Class Reference

objects instantiated from this class describes a static-method's signature

`#include <inc/dtGUI/ceguiconnectionsignatures.h>`

Public Types

- typedef bool(∗ CALLBACK_FPTR)(const CEGUI::EventArgs &)

Public Member Functions

- CEGUISignatureStatic (CEGUI::EventSet ∗pEventSet, const std::string &sEventName, const CALLBACK_FPTR pFunction)
- const std::string & GetEventName () const

  returns the event set which is part of the signature

- const CEGUI::EventSet ∗ GetEventSet () const

  returns the event set which is part of the signature

- const CALLBACK_FPTR GetFunctionPtr () const
- bool match (CEGUI::EventSet ∗pEventSet, const std::string &sEventName, const CALLBACK_FPTR pFunction) const

  check if one signature "matches" another (empty strings and NULL-Pointer's are Wildcards, means: "match any value")

- virtual bool operator== (const CEGUISignatureStatic &x) const

5.8.1 Detailed Description

objects instantiated from this class describes a static-method's signature

5.8.2 Member Typedef Documentation

5.8.2.1 typedef bool(∗ CALLBACK_FPTR)(const CEGUI::EventArgs &)

5.8.3 Constructor & Destructor Documentation

5.8.3.1 CEGUISignatureStatic (CEGUI::EventSet ∗ *pEventSet*, const std::string & *sEventName*, const CALLBACK_FPTR *pFunction*) `[inline]`

5.8.4 Member Function Documentation

5.8.4.1 const std::string& GetEventName () const `[inline]`

returns the event set which is part of the signature

5.8.4.2 const CEGUI::EventSet∗ GetEventSet () const `[inline]`

returns the event set which is part of the signature

5.8.4.3 const CALLBACK_FPTR GetFunctionPtr () const `[inline]`

5.8.4.4 bool match (CEGUI::EventSet ∗ *pEventSet*, const std::string & *sEventName*, const CALLBACK_FPTR *pFunction*) const `[inline]`

check if one signature "matches" another (empty strings and NULL-Pointer's are Wildcards, means: "match any value")

5.8.4.5 virtual bool operator== (const CEGUISignatureStatic & *x*) const `[inline, virtual]`

The documentation for this class was generated from the following file:

- ceguiconnectionsignatures.h

5.9 CEGUITexture Class Reference

implementation of the CEGUI::Texture for OpenSceneGraph

`#include <inc/dtGUI/ceguitexture.h>`

Public Member Functions

- CEGUITexture (CEGUI::Renderer *)
- ~CEGUITexture ()
- virtual CEGUI::ushort getHeight () const
- virtual CEGUI::ushort getOriginalHeight (void) const

  get the original height of the texture (cegui-textures are automatically resized to pot)

- virtual CEGUI::ushort getOriginalWidth (void) const

  get the original with of the texture (cegui-textures are automatically resized to pot)

- const osg::Texture2D * GetOSGTexture () const

  get current osg-texture

- osg::Texture2D * GetOSGTexture ()

  get current osg-texture

- unsigned int GetTextureID (osg::GraphicsContext *)

  get osg-texture-handle for a osg::GraphicsContext

- virtual CEGUI::ushort getWidth () const
- virtual float getXScale (void) const
- virtual float getYScale (void) const
- bool IsFlippedHorizontal () const

  if true the CEGUIRenderer will flip y-coordinates

- virtual void loadFromFile (const CEGUI::String &filename, const CEGUI::String &resourceGroup)
- virtual void loadFromMemory (const void *buffPtr, CEGUI::uint buffWidth, CEGUI::uint buffHeight, CEGUI::Texture::PixelFormat pixelFormat)
- void ResizeToMinPOT (unsigned short w, unsigned short h)
- void SetFlipHorizontal (bool bX)

  if set to true the CEGUIRenderer will flip y-coordinates

- void SetOSGTexture (osg::Texture2D *pTexture)

  set the osg-texture that'll be used

5.9.1 Detailed Description

implementation of the CEGUI::Texture for OpenSceneGraph

5.9.2 Constructor & Destructor Documentation

5.9.2.1 CEGUITexture (CEGUI::Renderer * *pRenderer*)

5.9.2.2 ~CEGUITexture ()

5.9.3 Member Function Documentation

5.9.3.1 CEGUI::ushort getHeight () const `[virtual]`

5.9.3.2 CEGUI::ushort getOriginalHeight (void) const `[virtual]`

get the original height of the texture (cegui-textures are automatically resized to pot)

5.9.3.3 CEGUI::ushort getOriginalWidth (void) const [virtual]

get the original with of the texture (cegui-textures are automatically resized to pot)

5.9.3.4 const osg::Texture2D* GetOSGTexture () const [inline]

get current osg-texture

5.9.3.5 osg::Texture2D* GetOSGTexture () [inline]

get current osg-texture

5.9.3.6 unsigned int GetTextureID (osg::GraphicsContext * *pGraphicsContext*)

get osg-texture-handle for a osg::GraphicsContext

5.9.3.7 CEGUI::ushort getWidth () const [virtual]

5.9.3.8 virtual float getXScale (void) const [inline, virtual]

5.9.3.9 virtual float getYScale (void) const [inline, virtual]

5.9.3.10 bool IsFlippedHorizontal () const [inline]

if true the CEGUIREnderer will flip y-coordinates

5.9.3.11 void loadFromFile (const CEGUI::String & *filename*, const CEGUI::String & *resourceGroup*) [virtual]

5.9.3.12 void loadFromMemory (const void * *buffPtr*, CEGUI::uint *buffWidth*, CEGUI::uint *buffHeight*, CEGUI::Texture::PixelFormat *pixelFormat*) [virtual]

5.9.3.13 void ResizeToMinPOT (unsigned short *w*, unsigned short *h*)

5.9.3.14 void SetFlipHorizontal (bool *bX*) [inline]

if set to true the CEGUIREnderer will flip y-coordinates

5.9.3.15 void SetOSGTexture (osg::Texture2D * *pTexture*) [inline]

set the osg-texture that'll be used

The documentation for this class was generated from the following files:

- ceguitexture.h
- ceguitexture.cpp

5.10 CEUIDrawable Class Reference

A DeltaDrawable used to render CEGUI.

`#include <inc/dtGUI/ceuidrawable.h>`

Classes

- class **osgCEUIDrawable**

  private class that ties the GUI rendering to an OSG node This is a private class that is used by the UIDrawable class.

Public Member Functions

- CEUIDrawable (dtCore::DeltaWin *win, dtCore::Keyboard *keyboard, dtCore::Mouse *mouse, dtGUI::BaseScriptModule *sm=0)

  Overloaded constructor, will automatically update CEGUI when the supplied Window is resized.

- bool AddChild (dtCore::DeltaDrawable *child)

  Attaches the Delta3D child's OSG graphics Node.

- bool GetAutoResizing () const
- const CEGUIKeyboardListener * GetKeyboardListener () const
- const CEGUIMouseListener * GetMouseListener () const
- const osg::Node * GetOSGNode () const
- osg::Node * GetOSGNode ()

  required by DeltaDrawable

- osg::Projection * GetProjectionNode ()

  Not usually needed, but this getter is provided for unusual scenarios.

- const dtGUI::CEGUIRenderer * GetRenderer () const
- dtGUI::CEGUIRenderer * GetRenderer ()

  Get a pointer to the underlying Renderer.

- osg::MatrixTransform * GetTransformNode ()

  Not usually needed, but this getter is provided for unusual scenarios.

- CEGUI::System * GetUI ()

  Get a pointer to the underlying CEGUI::System.

- void SetAutoResizing (bool enable=true)

  Automatically notify CEGUI of DeltaWin resizes (enabled by default).

- void SetOSGNode (osg::Node *pNode)
- void SetRenderBinDetails (int binNumber, const std::string &binName)

  Set the render bin details on the contained geode's state set.

- void SetRenderingSize (int width, int height)

  Manually set the size of the rendering area (in pixels).

- void ShutdownGUI ()

  Turns off the GUI System, no longer yielding GUI support.

Static Public Member Functions

- static void DisplayProperties (CEGUI::Window *window, bool onlyNonDefault=true)

  Display all the properties of the supplied CEGUI::Window.

Protected Member Functions

- virtual ~CEUIDrawable ()
- void OnMessage (dtCore::Base::MessageData *data)

Protected Attributes

- dtCore::RefPtr< osg::Geode > mGeode
- int mHeight
- dtCore::RefPtr< dtCore::Keyboard > mKeyboard
- dtCore::RefPtr< dtCore::Mouse > mMouse
- dtCore::RefPtr< osg::Node > mNode
- dtCore::RefPtr< osg::Projection > mProjection
- CEGUIRenderer * mRenderer

  The opengl renderer we're using.

- dtGUI::BaseScriptModule * mScriptModule
- dtCore::RefPtr< osg::MatrixTransform > mTransform
- CEGUI::System * mUI

  Pointer to the CUI_UI.

- int mWidth
- dtCore::RefPtr< dtCore::DeltaWin > mWindow

  The window this UI is being rendered in.

5.10.1 Detailed Description

A DeltaDrawable used to render CEGUI. This class is a derivative of DeltaDrawable and is used to render and manage the CEGUI system. The CEUIDrawable is responsible for setting up the CEGUI system and supplying mouse and keyboard events to the UI.

To create a new GUI, instantiate a CEGUIDrawable and add it to the Scene using Scene::AddDrawable(). You can then use the CEGUI API to create CEGUI::Windows and adjust their properties.

NOTE: The CEGUIDrawable class must be instantiated *after* the application has created a valid OpenGL context (i.e., during dtABC::Application::Config()).

5.10.2 Constructor & Destructor Documentation

5.10.2.1 CEUIDrawable (dtCore::DeltaWin * *win*, dtCore::Keyboard * *keyboard*, dtCore::Mouse * *mouse*, dtGUI::BaseScriptModule * *sm* = 0)

Overloaded constructor, will automatically update CEGUI when the supplied Window is resized. The supplied DeltaWin will automatically be monitored for size change and pass the new size onto the CEGUI Renderer.

Parameters

win : The DeltaWin to monitor for size change

sm : The ScriptModule to use for CEGUI script processing

Exceptions

dtUtil::Exception Gets thrown if CEGUI cannot be initialized

5.10.2.2 ~CEUIDrawable () [protected, virtual]

5.10.3 Member Function Documentation

5.10.3.1 bool AddChild (dtCore::DeltaDrawable * *child*)

Attaches the Delta3D child's OSG graphics Node.

5.10.3.2 void DisplayProperties (CEGUI::Window * *window*, bool *onlyNonDefault* = true) [static]

Display all the properties of the supplied CEGUI::Window. Display the properties associated with the supplied CEGUI::Window to the console.

Useful to find all the text names of the properties and see what the current values of the properties are.

Parameters

window : The window to query the properties of

onlyNonDefault : Display only properties that are not default values (default=true)

5.10.3.3 bool GetAutoResizing () const [inline]

5.10.3.4 const CEGUIKeyboardListener* GetKeyboardListener () const [inline]

5.10.3.5 const CEGUIMouseListener* GetMouseListener () const [inline]

5.10.3.6 const osg::Node* GetOSGNode () const [inline]

5.10.3.7 osg::Node* GetOSGNode () [inline]

required by DeltaDrawable

5.10.3.8 osg::Projection* GetProjectionNode () [inline]

Not usually needed, but this getter is provided for unusual scenarios.

5.10.3.9 const dtGUI::CEGUIRenderer* GetRenderer () const [inline]

5.10.3.10 dtGUI::CEGUIRenderer* GetRenderer () [inline]

Get a pointer to the underlying Renderer.

5.10.3.11 osg::MatrixTransform* GetTransformNode () [inline]

Not usually needed, but this getter is provided for unusual scenarios.

5.10.3.12 CEGUI::System* GetUI () [inline]

Get a pointer to the underlying CEGUI::System.

5.10.3.13 void OnMessage (dtCore::Base::MessageData * *data*) [protected]

Todo

must we really cast here?

5.10.3.14 void SetAutoResizing (bool *enable* = true) [inline]

Automatically notify CEGUI of DeltaWin resizes (enabled by default).

5.10.3.15 void SetOSGNode (osg::Node * *pNode*) [inline]

5.10.3.16 void SetRenderBinDetails (int *binNumber*, const std::string & *binName*)

Set the render bin details on the contained geode's state set. Parameters

binNumber Order of the bin in relation to other bins.

binName Name of the bin.

5.10.3.17 void SetRenderingSize (int *width*, int *height*)

Manually set the size of the rendering area (in pixels). Set the width and height of the rendering area. Typically this is just the size of the DeltaWin the GUI is being rendered in. If AutoResizing is enabled, these values will be overwritten. Disable AutoResizing to manually control the rendered area. See also SetAutoResizing()

Parameters

width : the width of the rendered area (pixels)

height : the heigh tof hte rendered area (pixels)

5.10.3.18 void ShutdownGUI ()

Turns off the GUI System, no longer yielding GUI support. The System will do its clean up at this time.

5.10.4 Member Data Documentation

5.10.4.1 dtCore::RefPtr<osg::Geode> mGeode [protected]

5.10.4.2 int mHeight [protected]

5.10.4.3 dtCore::RefPtr<dtCore::Keyboard> mKeyboard [protected]

5.10.4.4 dtCore::RefPtr<dtCore::Mouse> mMouse [protected]

5.10.4.5 dtCore::RefPtr<osg::Node> mNode [protected]

5.10.4.6 dtCore::RefPtr<osg::Projection> mProjection [protected]

5.10.4.7 CEGUIRenderer* mRenderer [protected]

The opengl renderer we're using.

5.10.4.8 dtGUI::BaseScriptModule* mScriptModule [protected]

5.10.4.9 dtCore::RefPtr<osg::MatrixTransform> mTransform [protected]

5.10.4.10 CEGUI::System* mUI [protected]

Pointer to the CUI_UI.

5.10.4.11 int mWidth [protected]

5.10.4.12 dtCore::RefPtr<dtCore::DeltaWin> mWindow [protected]

The window this UI is being rendered in.

The documentation for this class was generated from the following files:

- ceuidrawable.h
- ceuidrawable.cpp

5.11 ExceptionEnum Class Reference

`#include <inc/dtGUI/guiexceptionenum.h>`

Static Public Attributes

- static ExceptionEnum GenericCEGUIException

Protected Member Functions

- ExceptionEnum (const std::string &name)

5.11.1 Constructor & Destructor Documentation

5.11.1.1 ExceptionEnum (const std::string & *name*) `[inline, protected]`

5.11.2 Member Data Documentation

5.11.2.1 ExceptionEnum GenericCEGUIException `[static]`

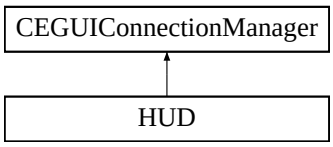
The documentation for this class was generated from the following files:

- guiexceptionenum.h
- guiexceptionenum.cpp

5.12 HUD Class Reference

gui-class for rendering and managing widgets

`#include <inc/dtGUI/hud.h>`Inheritance diagram for HUD::



Classes

- struct **GUIViewport**
  internal viewport-representation
- class **HUDDrawable**
  an osg::Drawable which will render a provided(defined via the constructor) gui

Public Member Functions

- HUD (dtCore::Camera *pTargetCamera, dtCore::Keyboard *pObservedKeyboard=0, dtCore::Mouse *pObservedMouse=0, const std::string &sName="HUD")
  constructs a new gui object
- HUD (osg::Camera *pTargetCamera, dtCore::Keyboard *pObservedKeyboard=0, dtCore::Mouse *pObservedMouse=0, const std::string &sName="HUD")
  constructs a new gui object
- void Connect (const std::string &sWidgetName, const std::string &sEventName, bool(*pFunction)(const CEGUI::EventArgs &))
- template<class T >
  void Connect (const std::string &sWidgetName, const std::string &sEventName, bool(T::*pObjectMember)(const CEGUI::EventArgs &), T *pObject)
  wraps CEGUIConnectionManager::Connect(this->GetWidget(widgetName), eventName, pObjectMember, pObject)
- Widget * CreateWidget (const std::string &sWidgetTypeName, const std::string &sWidgetName="", const std::string &sPrefix="")
  creates/adds a widget to root-sheet
- Widget * CreateWidget (Widget *pParentWidget, const std::string &sWidgetTypeName, const std::string &sWidgetName="", const std::string &sPrefix="")
  creates/adds a widget to a given parent
- void Disconnect (const std::string &sWidgetName, const std::string &sEventName, bool(*pFunction)(const CEGUI::EventArgs &))
- void Disconnect (const std::string &sWidgetName, const std::string &sEventName, int, int)
  wraps this->Disconnect(this->GetWidget(widgetName), eventName, 'wildcard', 'wildcard')
- template<class T >
  void Disconnect (const std::string &sWidgetName, const std::string &sEventName, bool(T::*pObjectMember)(const CEGUI::EventArgs &), int)
  wraps this->Disconnect(this->GetWidget(widgetName), eventName, pObjectMember, 'widlcard')
- template<class T >
  void Disconnect (const std::string &sWidgetName, const std::string &sEventName, int, T *pObject)

wraps this->Disconnect(this->GetWidget(widgetName), eventName, 'wildcard', pObject)

- template<class T >
  void Disconnect (const std::string &sWidgetName, const std::string &sEventName, bool(T::*pObjectMember)(const CEGUI::EventArgs &), T *pObject)
  wraps this->Disconnect(this->GetWidget(widgetName), eventName, pObjectMember, pObject)

- osg::Geode * GetInternalGraph () const
- bool GetIsMouseCursorVisible () const
  returns whether the CEGUI mouse cursor should be visible or not

- const dtCore::Keyboard * GetKeyboard () const
  returns keyboard-object used by the gui for input

- dtCore::Keyboard * GetKeyboard ()
  returns keyboard-object used by the gui for input

- const dtCore::Mouse * GetMouse () const
  returns mouse-object from which the gui receives events

- dtCore::Mouse * GetMouse ()
  returns mouse-object from which the gui receives events

- osg::Texture2D * GetOrCreateOSGTexture (const std::string &sWidgetName)
  returns associated osg-texture for any widget which has the "Image" property

- const std::string GetPrefix () const
  returns auto-generated prefix for the gui

- const Widget * GetRootsheet () const
  returns default sheet (autogenerated root-window)

- Widget * GetRootSheet ()
  returns default sheet (autogenerated root-window)

- Widget * GetWidget (const std::string &sWidgetName, const std::string &sPrefix="")
  searches widget by it's name

- Widget * LoadLayout (Widget *pWidgetParent, const std::string &sFileName, const std::string &sPrefix="")
  loads a layout-file and add it to the given Widget(which should a child of the gui)

- Widget * LoadLayout (const std::string &sFileName, const std::string &sPrefix="")
  loads a layout-file

- void MakeCurrent () const
- void SetCamera (osg::Camera *pTargetCamera)
  sets the rendertarget (camera) for the gui-object

- void SetIsMouseCursorVisible (bool val)
  sets whether the CEGUI mouse cursor should be visible or not

- void SetKeyboard (dtCore::Keyboard *pObservedKeyboard)
  sets keyboard-object which is used for input

- void SetMouse (dtCore::Mouse *pObservedMouse)
  sets the mouse-object from which the gui receives events

Static Public Member Functions

- static void CreateFont (const std::string &)
  shortcut/wrapper to CEGUI::FontManager::createFont
- static void DestroyAllFonts ()
  shortcut/wrapper to CEGUI::FontManager::destroryAllFonts
- static void DestroyFont (const std::string &)
  shortcut/wrapper to CEGUI::FontManager::destroryFont
- static const std::string & GetFilePath ()
  returns the file path for gui-xml-data/images
- static void LoadScheme (const std::string &)
  shortcut/wrapper to CEGUI::SchemeManager::loadScheme
- static void SetFilePath (const std::string &sPath)
  sets the file path for all gui-objects
- static void SetMouseCursor (const std::string &sImagesetName, const std::string &sImageName)
  shortcut/wrapper to CEGUI::System::getSingleton().setMouseCursor
- static void UnloadAllSchemes ()
  shortcut/wrapper to CEGUI::SchemeManager::undloadAllSchemes
- static void UnloadScheme (const std::string &)
  shortcut/wrapper to CEGUI::SchemeManager::unloadScheme

Protected Member Functions

- virtual ~HUD ()
  destructs an existing gui object

5.12.1 Detailed Description

gui-class for rendering and managing widgets any HUD can use any camera as render-target each HUD owns it's own root-sheet (See also GetRootSheet()) which always fits the the whole viewport of the targeted-camera internally each HUD will use an own name-prefix for widgets(if they were created via LoadLayout/CreateWidget). so if one intend to use cegui-standard-api (example: CEGUI::WindowManager::getWindow()) window names must be prefixed by the HUD's prefix See also GetPrefix() if creating new connections via Connect/Disconnect one keep in mind that these connections are handled by the gui-object and thus they will be lost on gui-object-destruction the Connect/Disconnect are inteded to handle the connection creation/destruction more easyly for widgets. because of this fact it's wise to use them only for this

5.12.2 Constructor & Destructor Documentation

5.12.2.1 HUD (osg::Camera * *pTargetCamera*, dtCore::Keyboard * *pObservedKeyboard* = 0, dtCore::Mouse * *pObservedMouse* = 0, const std::string & *sName* = "HUD")

constructs a new gui object Parameters

pTargetCamera camera where the gui is drawn as hud

pObservedKeyboard observed keyboard-object for input

pObservedMouse observed mouse-object for input

sName delta3D-name of the gui-object

5.12.2.2 HUD (dtCore::Camera * *pTargetCamera*, dtCore::Keyboard * *pObservedKeyboard* = 0, dtCore::Mouse * *pObservedMouse* = 0, const std::string & *sName* = "HUD")

constructs a new gui object Parameters

pTargetCamera camera where the gui will show up (as hud)

pObservedKeyboard observed keyboard-object for input

pObservedMouse observed mouse-object for input

sName delta3D-name of the gui-object

5.12.2.3 ~HUD () [protected, virtual]

destructs an existing gui object

5.12.3 Member Function Documentation

5.12.3.1 void Connect (const std::string & *sWidgetName*, const std::string & *sEventName*, bool(*)(const CEGUI::EventArgs &) *pFunction*) [inline]

5.12.3.2 void Connect (const std::string & *sWidgetName*, const std::string & *sEventName*, bool(T::*)(const CEGUI::EventArgs &) *pObjectMember*, T * *pObject*) [inline]

wraps CEGUIConnectionManager::Connect(this->GetWidget(widgetName), eventName, pObjectMember, pObject)

5.12.3.3 void CreateFont (const std::string & *sFileName*) [static]

shortcut/wrapper to CEGUI::FontManager::createFont

5.12.3.4 Widget * CreateWidget (const std::string & *sWidgetTypeName*, const std::string & *sWidgetName* = "", const std::string & *sPrefix* = "")

creates/adds a widget to root-sheet

5.12.3.5 Widget * CreateWidget (Widget * *pParentWidget*, const std::string & *sWidgetTypeName*, const std::string & *sWidgetName* = "", const std::string & *sPrefix* = "")

creates/adds a widget to a given parent

5.12.3.6 void DestroyAllFonts () [static]

shortcut/wrapper to CEGUI::FontManager::destroryAllFonts

5.12.3.7 void DestroyFont (const std::string & *sFileName*) [static]

shortcut/wrapper to CEGUI::FontManager::destroryFont

5.12.3.8 void Disconnect (const std::string & *sWidgetName*, const std::string & *sEventName*, bool(*)(const CEGUI::EventArgs &) *pFunction*) [inline]

5.12.3.9 void Disconnect (const std::string & *sWidgetName*, const std::string & *sEventName*, int, int) [inline]

wraps this->Disconnect(this->GetWidget(widgetName), eventName, 'wildcard', 'wildcard')

5.12.3.10 void Disconnect (const std::string & *sWidgetName*, const std::string & *sEventName*, bool(T::*)(const CEGUI::EventArgs &) *pObjectMember*, int) [inline]

wraps this->Disconnect(this->GetWidget(widgetName), eventName, pObjectMember, 'widlcard')

5.12.3.11 void Disconnect (const std::string & *sWidgetName*, const std::string & *sEventName*, int, T * *pObject*) [inline]

wraps this->Disconnect(this->GetWidget(widgetName), eventName, 'wildcard', pObject)

5.12.3.12 void Disconnect (const std::string & *sWidgetName*, const std::string & *sEventName*, bool(T::*)(const CEGUI::EventArgs &) *pObjectMember*, T * *pObject*) [inline]

wraps this->Disconnect(this->GetWidget(widgetName), eventName, pObjectMember, pObject)

5.12.3.13 const std::string & GetFilePath () [static]

returns the file path for gui-xml-data/images See also SetFilePath

5.12.3.14 osg::Geode* GetInternalGraph () const [inline]

5.12.3.15 bool GetIsMouseCursorVisible () const [inline]

returns whether the CEGUI mouse cursor should be visible or not

5.12.3.16 const dtCore::Keyboard* GetKeyboard () const [inline]

returns keyboard-object used by the gui for input

5.12.3.17 dtCore::Keyboard* GetKeyboard () [inline]

returns keyboard-object used by the gui for input

5.12.3.18 const dtCore::Mouse* GetMouse () const [inline]

returns mouse-object from which the gui receives events

5.12.3.19 dtCore::Mouse* GetMouse () [inline]

returns mouse-object from which the gui receives events

5.12.3.20 osg::Texture2D * GetOrCreateOSGTexture (const std::string & *sWidgetName*)

returns associated osg-texture for any widget which has the "Image" property

5.12.3.21 const std::string GetPrefix () const

returns auto-generated prefix for the gui

5.12.3.22 const Widget* GetRootsheet () const [inline]

returns default sheet (autogenerated root-window)

5.12.3.23 Widget* GetRootSheet () [inline]

returns default sheet (autogenerated root-window)

5.12.3.24 Widget * GetWidget (const std::string & *sWidgetName*, const std::string & *sPrefix* = "")

searches widget by it's name

5.12.3.25 Widget * LoadLayout (Widget * *pWidgetParent*, const std::string & *sFileName*, const std::string & *sPrefix* = "")

loads a layout-file and add it to the given Widget(which should a child of the gui)

5.12.3.26 Widget * LoadLayout (const std::string & *sFileName*, const std::string & *sPrefix* = "")

loads a layout-file

5.12.3.27 void LoadScheme (const std::string & *sFileName*) [static]

shortcut/wrapper to CEGUI::SchemeManager::loadScheme

5.12.3.28 void MakeCurrent () const

5.12.3.29 void SetCamera (osg::Camera * *pTargetCamera*)

sets the rendertarget (camera) for the gui-object

5.12.3.30 void SetFilePath (const std::string & *sPath*) [static]

sets the file path for all gui-objects

5.12.3.31 void SetIsMouseCursorVisible (bool *val*) [inline]

sets whether the CEGUI mouse cursor should be visible or not

5.12.3.32 void SetKeyboard (dtCore::Keyboard * *pObservedKeyboard*)

sets keyboard-object which is used for input

5.12.3.33 void SetMouse (dtCore::Mouse * *pObservedMouse*)

sets the mouse-object from which the gui receives events

5.12.3.34 void SetMouseCursor (const std::string & *sImagesetName*, const std::string & *sImageName*) [static]

shortcut/wrapper to CEGUI::System::getSingleton().setMouseCursor

5.12.3.35 void UnloadAllSchemes () [static]

shortcut/wrapper to CEGUI::SchemeManager::undloadAllSchemes

5.12.3.36 void UnloadScheme (const std::string & *sFileName*) [static]

shortcut/wrapper to CEGUI::SchemeManager::unloadScheme

The documentation for this class was generated from the following files:

- hud.h
- hud.cpp

5.13 Renderer Class Reference

Simple CEGui rendering class based on the CEGUI::OpenGLRender class.

`#include <inc/dtGUI/renderer.h>`

Public Member Functions

- Renderer (unsigned int max_quads=0, int width=0, int height=0)
- ~Renderer ()
- virtual CEGUI::ResourceProvider * createResourceProvider ()

  Create the ResourceProvider (a dtGUI::ResourceProvider).

5.13.1 Detailed Description

Simple CEGui rendering class based on the CEGUI::OpenGLRender class.

5.13.2 Constructor & Destructor Documentation

5.13.2.1 Renderer (unsigned int *max_quads* = `0`, int *width* = `0`, int *height* = `0`)

Todo

deprecate this unnecessary ctor

5.13.2.2 ~Renderer () `[inline]`

5.13.3 Member Function Documentation

5.13.3.1 CEGUI::ResourceProvider * createResourceProvider (void) `[virtual]`

Create the ResourceProvider (a dtGUI::ResourceProvider).

The documentation for this class was generated from the following files:

- renderer.h
- renderer.cpp

5.14 ResourceProvider Class Reference

A simple CEGUI ResourceProvider based on the CEGUI::DefaultResourceProvider.

`#include <inc/dtGUI/resourceprovider.h>`

Public Member Functions

- ResourceProvider ()
- ~ResourceProvider ()
- void loadRawDataContainer (const CEGUI::String &filename, CEGUI::RawDataContainer &output, const CEGUI::String &resourceGroup)

  Load the data of the supplied filename.

5.14.1 Detailed Description

A simple CEGUI ResourceProvider based on the CEGUI::DefaultResourceProvider. Add the functionality of using search paths to find the data files used by CEGUI.

See also SetDataFilePathList()

5.14.2 Constructor & Destructor Documentation

5.14.2.1 ResourceProvider ()

5.14.2.2 ~ResourceProvider () `[inline]`

5.14.3 Member Function Documentation

5.14.3.1 void loadRawDataContainer (const CEGUI::String & *filename*, CEGUI::RawDataContainer & *output*, const CEGUI::String & *resourceGroup*)

Load the data of the supplied filename. This will load the file with the supplied filename using the search paths find the files.

See also SetDataFilePathList()

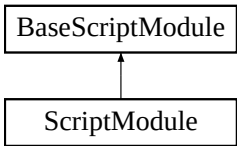
The documentation for this class was generated from the following files:

- resourceprovider.h
- resourceprovider.cpp

5.15 ScriptModule Class Reference

ScriptModule is the binding from CEGUI::Events to specific application callbacks.

`#include <inc/dtGUI/scriptmodule.h>`Inheritance diagram for ScriptModule::



Public Types

- typedef std::map< std::string, HandlerFunctor > CallbackRegistry
- typedef dtUtil::Functor< bool, TYPELIST_1(const CEGUI::EventArgs &)> HandlerFunctor
- typedef bool(* STATIC_FUNCTION)(const CEGUI::EventArgs &e)

Public Member Functions

- ScriptModule ()
- virtual ~ScriptModule ()
- bool AddCallback (const std::string &name, const HandlerFunctor &callback)

  Add a non-static callback handler.

- bool AddCallback (const std::string &name, STATIC_FUNCTION func)

  Add a static callback handler.

- virtual bool executeScriptedEventHandler (const CEGUI::String &handler_name, const CEGUI::EventArgs &ea)

  Uses the map of strings to functions to execute a function to handle the EventArgs.

- virtual void executeScriptFile (const CEGUI::String &filename, const CEGUI::String &resourceGroup="")
- virtual int executeScriptGlobal (const CEGUI::String &function_name)
- virtual void executeString (const CEGUI::String &str)
- const CallbackRegistry & GetRegistry () const

  Returns the StaticRegistry.

5.15.1 Detailed Description

ScriptModule is the binding from CEGUI::Events to specific application callbacks. Create an instance of this class, and provide it as a parameter during construction of a CEUIDrawable instance. Add new handlers with the AddCallback function.

5.15.2 Member Typedef Documentation

5.15.2.1 typedef std::map<std::string,HandlerFunctor> CallbackRegistry

5.15.2.2 typedef dtUtil::Functor<bool,TYPELIST_1(const CEGUI::EventArgs&)> HandlerFunctor

5.15.2.3 typedef bool(* STATIC_FUNCTION)(const CEGUI::EventArgs &e)

Todo

test to know if this typedef can support just NonMemberFunction, not requiring them to be static.

5.15.3 Constructor & Destructor Documentation

5.15.3.1 ScriptModule ()

5.15.3.2 ~ScriptModule () [virtual]

5.15.4 Member Function Documentation

5.15.4.1 bool AddCallback (const std::string & *name*, const HandlerFunctor & *callback*)

Add a non-static callback handler. Example:

```
class App
{
  ...
 public:
  bool OnClick( const CEGUI::EventArgs &e );
  ...
}

 ...
 App *mApp = new App();

 dtGUI::ScriptModule::HandlerFunctor handler( dtUtil::MakeFunctor( &App::OnClick
     , mApp ) );
 mScriptModule->AddCallback("OnDoSomething", handler );
...
```

Parameters

name is the string representation of the handler function to be executed for the CEGUI::Event.

func is an instance of a function object to be called when the CEGUI::Event is activated.

Attention An attempt was make an implemenation with the signature template<typename InstT> bool AddCallback(const std::string& name, bool (InstT::*MemFun)(const CEGUI::EventArgs&), InstT* instance) but somehow creating the HandleFunctor intenal to the function caused some problems.

5.15.4.2 bool AddCallback (const std::string & *name*, STATIC_FUNCTION *func*)

Add a static callback handler. Example:

```
class MyClass
{
  ...
  static bool OnClick( const CEGUI::EventArgs &e );
  ...
}

 ...
 mScriptModule->AddCallback("OnDoSomething", &OnClick);
 ...
```

Parameters

name is the string representation of the handler function to be executed for the CEGUI::Event.

func is the pointer to the function to be called when the CEGUI::Event is activated.

5.15.4.3 bool executeScriptedEventHandler (const CEGUI::String & *handler_name*, const CEGUI::EventArgs & *ea*) [virtual]

Uses the map of strings to functions to execute a function to handle the EventArgs. This function is called by CEGUI::System when a CEGUI::Window throws an CEGUI::Event

Implements BaseScriptModule.

5.15.4.4 void executeScriptFile (const CEGUI::String & *filename*, const CEGUI::String & *resourceGroup* = "") [virtual]

Implements BaseScriptModule.

5.15.4.5 int executeScriptGlobal (const CEGUI::String & *function_name*) [virtual]

Implements BaseScriptModule.

5.15.4.6 void executeString (const CEGUI::String & *str*) [virtual]

Implements BaseScriptModule.

5.15.4.7 const CallbackRegistry& GetRegistry () const [inline]

Returns the StaticRegistry.

The documentation for this class was generated from the following files:

- scriptmodule.h
- scriptmodule.cpp

Chapter 6

File Documentation

6.1 basescriptmodule.h File Reference

```
#include <CEGUI/CEGUIScriptModule.h>
#include <dtGUI/export.h>
```

Classes

- class BaseScriptModule

  An abstract interface for classes used by CEUIDrawable.

Namespaces

- namespace dtGUI

  The dtGUI namespace contains the functionality to render and control OpenGL graphical user interfaces.

6.2 ceguiconnectionmanager.cpp File Reference

```
#include <dtGUI/ceguiconnectionmanager.h>
```

6.3 ceguiconnectionmanager.h File Reference

```
#include <string>
#include <map>
#include <dtGUI/ceguiconnectionsignatures.h>
#include <CEGUI/CEGUIWindow.h>
#include <CEGUI/CEGUIEvent.h>
#include <CEGUI/CEGUIInputEvent.h>
#include <CEGUI/CEGUIEventSet.h>
#include <CEGUI/CEGUIEventArgs.h>
#include <CEGUI/CEGUISubscriberSlot.h>
#include <dtUtil/log.h>
```

Classes

- class CEGUIConnectionManager

  this class is used to keep track of connections (event-callbacks) for any widget/EventSet created for/by a gui-object.

Namespaces

- namespace dtGUI

  The dtGUI namespace contains the functionality to render and control OpenGL graphical user interfaces.

6.4 ceguiconnectionsignatures.h File Reference

```
#include <string>
```

Classes

- class CEGUIMemberSignature< T >
  objects instantiated from this class describes an object-member-"signature"
- class CEGUIMemberSignatureBase
  Common interface for a member-callback-connection-signatures.
- class CEGUISignatureStatic
  objects instantiated from this class describes a static-method's signature

Namespaces

- namespace dtGUI
  The dtGUI namespace contains the functionality to render and control OpenGL graphical user interfaces.

6.5 ceguikeyboardlistener.cpp File Reference

```
#include <dtGUI/ceguikeyboardlistener.h>
#include <dtGUI/hud.h>
#include <CEGUI/CEGUISystem.h>
#include <osgGA/GUIEventAdapter>
```

6.6 ceguikeyboardlistener.h File Reference

```
#include <dtCore/keyboard.h>
#include <CEGUI/CEGUIInputEvent.h>
#include <dtGUI/export.h>
```

Classes

- class CEGUIKeyboardListener

Namespaces

- namespace dtGUI

  The dtGUI namespace contains the functionality to render and control OpenGL graphical user interfaces.

6.7 ceguimouselistener.cpp File Reference

```
#include <dtGUI/ceguimouselistener.h>
#include <dtGUI/hud.h>
#include <CEGUI/CEGUIInputEvent.h>
#include <CEGUI/CEGUISystem.h>
#include <dtUtil/mathdefines.h>
```

6.8 ceguimouselistener.h File Reference

```
#include <dtCore/mouse.h>
#include <dtGUI/export.h>
```

Classes

- class CEGUIMouseListener

  A mouse device listener to inject input to CEGUI.

Namespaces

- namespace dtGUI

  The dtGUI namespace contains the functionality to render and control OpenGL graphical user interfaces.

6.9 ceguirenderer.cpp File Reference

```
#include <dtGUI/ceguirenderer.h>
#include <dtGUI/ceguitexture.h>
#include <dtUtil/macros.h>
#include <osg/GL>
#include <osg/GLU>
#include <iostream>
#include <CEGUI/CEGUIExceptions.h>
#include <CEGUI/CEGUIEventArgs.h>
#include <CEGUI/CEGUIImageCodec.h>
#include <CEGUI/CEGUIDynamicModule.h>
#include <CEGUI/CEGUIBase.h>
#include <CEGUI/CEGUITexture.h>
#include <CEGUI/CEGUILogger.h>
```

Defines

- #define S_(X) #X
- #define STRINGIZE(X) S_(X)

6.9.1 Define Documentation

6.9.1.1 #define S_(X) #X

6.9.1.2 #define STRINGIZE(X) S_(X)

6.10 ceguirenderer.h File Reference

```
#include <list>
#include <set>
#include <dtGUI/export.h>
#include <osg/GraphicsContext>
#include <CEGUI/CEGUIRenderer.h>
```

Classes

- class CEGUIRenderer
- struct **MyQuad**
- struct **QuadInfo**

Namespaces

- namespace dtGUI

  The dtGUI namespace contains the functionality to render and control OpenGL graphical user interfaces.

Defines

- #define OGLRENDERER_VBUFF_CAPACITY 4096

6.10.1 Define Documentation

6.10.1.1 #define OGLRENDERER_VBUFF_CAPACITY 4096

6.11 ceguitexture.cpp File Reference

```
#include <dtGUI/ceguitexture.h>
#include <dtGUI/ceguirenderer.h>
#include <osg/GraphicsContext>
#include <osg/Image>
#include <osg/Texture2D>
#include <CEGUI/CEGUIExceptions.h>
#include <CEGUI/CEGUISystem.h>
#include <CEGUI/CEGUIImageCodec.h>
#include <iostream>
```

6.12 ceguitexture.h File Reference

```
#include <dtGUI/export.h>
#include <osg/Texture2D>
#include <CEGUI/CEGUITexture.h>
```

Classes

- class CEGUITexture

  implementation of the CEGUI::Texture for OpenSceneGraph

Namespaces

- namespace dtGUI

  The dtGUI namespace contains the functionality to render and control OpenGL graphical user interfaces.

6.13 ceuidrawable.cpp File Reference

```
#include <CEGUI/CEGUIPropertySet.h>
#include <CEGUI/CEGUISystem.h>
#include <CEGUI/CEGUIWindow.h>
#include <CEGUI/CEGUIExceptions.h>
#include <CEGUI/CEGUIVersion.h>
#include <dtGUI/ceuidrawable.h>
#include <dtGUI/ceguimouselistener.h>
#include <dtGUI/ceguikeyboardlistener.h>
#include <dtGUI/basescriptmodule.h>
#include <dtGUI/guiexceptionenum.h>
#include <dtGUI/resourceprovider.h>
#include <dtGUI/ceguirenderer.h>
#include <dtCore/exceptionenum.h>
#include <dtUtil/exception.h>
#include <dtCore/deltawin.h>
#include <dtCore/system.h>
#include <dtUtil/log.h>
#include <dtGUI/renderer.h>
#include <osg/Geode>
#include <osg/Projection>
#include <osg/MatrixTransform>
#include <osg/State>
```

6.14 ceuidrawable.h File Reference

```
#include <dtCore/deltadrawable.h>
#include <dtCore/refptr.h>
#include <dtGUI/export.h>
#include <osg/Drawable>
#include <osg/CopyOp>
#include <osg/Node>
```

Classes

- class CEUIDrawable

  A DeltaDrawable used to render CEGUI.

- class **osgCEUIDrawable**

  private class that ties the GUI rendering to an OSG node This is a private class that is used by the UIDrawable class.

Namespaces

- namespace dtCore
- namespace dtGUI

  The dtGUI namespace contains the functionality to render and control OpenGL graphical user interfaces.

6.15 dtgui.h File Reference

```
#include <dtGUI/ceuidrawable.h>
#include <dtGUI/scriptmodule.h>
```

Namespaces

- namespace dtGUI

  The dtGUI namespace contains the functionality to render and control OpenGL graphical user interfaces.

6.16 export.h File Reference

Defines

- #define DT_GUI_EXPORT

6.16.1 Define Documentation

6.16.1.1 #define DT_GUI_EXPORT

6.17 guiexceptionenum.cpp File Reference

`#include <dtGUI/guiexceptionenum.h>`

Namespaces

- namespace dtGUI

  The dtGUI namespace contains the functionality to render and control OpenGL graphical user interfaces.

Functions

- IMPLEMENT_ENUM (ExceptionEnum)

6.18 guiexceptionenum.h File Reference

```
#include <dtUtil/enumeration.h>
#include <dtUtil/exception.h>
#include <dtGUI/export.h>
```

Classes

- class ExceptionEnum

Namespaces

- namespace dtGUI

  The dtGUI namespace contains the functionality to render and control OpenGL graphical user interfaces.

6.19 hud.cpp File Reference

```
#include <dtGUI/hud.h>
#include <dtGUI/ceguirenderer.h>
#include <dtGUI/ceguitexture.h>
#include <CEGUI/CEGUI.h>
#include <CEGUI/CEGUIDefaultResourceProvider.h>
#include <CEGUI/CEGUIPropertySet.h>
#include <CEGUI/CEGUISystem.h>
#include <CEGUI/CEGUIWindow.h>
#include <CEGUI/CEGUIWindowManager.h>
#include <CEGUI/CEGUIImageset.h>
#include <CEGUI/CEGUIImagesetManager.h>
#include <CEGUI/CEGUIExceptions.h>
#include <CEGUI/CEGUIVersion.h>
#include <CEGUI/CEGUIXMLParser.h>
#include <CEGUI/CEGUITexture.h>
#include <CEGUI/CEGUIPropertyHelper.h>
#include <CEGUI/CEGUISchemeManager.h>
#include <dtCore/system.h>
#include <dtCore/globals.h>
#include <dtCore/mouse.h>
#include <dtCore/camera.h>
#include <dtCore/keyboard.h>
#include <dtCore/deltadrawable.h>
#include <dtUtil/log.h>
#include <dtUtil/macros.h>
#include <osg/GraphicsContext>
#include <osg/State>
#include <osg/Camera>
#include <osg/Texture2D>
#include <osg/Geode>
#include <sstream>
```

Functions

- std::string GetCEGUIPrefix (CEGUI::Window *win)

6.19.1 Function Documentation

6.19.1.1 std::string GetCEGUIPrefix (CEGUI::Window * *win*)

6.20 hud.h File Reference

```
#include <map>
#include <string>
#include <dtGUI/export.h>
#include <dtGUI/ceguiconnectionmanager.h>
#include <dtGUI/ceguimouselistener.h>
#include <dtGUI/ceguikeyboardlistener.h>
#include <dtGUI/widget.h>
#include <dtCore/refptr.h>
#include <dtCore/base.h>
#include <osg/Drawable>
```

Classes

- struct **GUIViewport**
  internal viewport-representation
- class HUD
  gui-class for rendering and managing widgets
- class **HUDDrawable**
  an osg::Drawable which will render a provided(defined via the constructor) gui

Namespaces

- namespace dtCore
- namespace dtGUI
  The dtGUI namespace contains the functionality to render and control OpenGL graphical user interfaces.

6.21 mainpage.h File Reference

6.21.1 Detailed Description

This file contains Doxygen special commands and text for the Main Page and some other minor aspects of this documentation. It is not part of Delta3D.

6.22 renderer.cpp File Reference

```
#include <dtGUI/renderer.h>
#include <dtGUI/resourceprovider.h>
```

6.23 renderer.h File Reference

```
#include <dtGUI/export.h>
#include <CEGUI/CEGUIVersion.h>
#include <CEGUI/renderers/OpenGLGUIRenderer/openglrenderer.h>
```

Classes

- class Renderer

  Simple CEGui rendering class based on the CEGUI::OpenGLRender class.

Namespaces

- namespace dtGUI

  The dtGUI namespace contains the functionality to render and control OpenGL graphical user interfaces.

6.24 resourceprovider.cpp File Reference

```
#include <CEGUI/CEGUIExceptions.h>
#include <dtGUI/resourceprovider.h>
#include <dtCore/globals.h>
#include <dtUtil/log.h>
```

6.25 resourceprovider.h File Reference

```
#include <dtGUI/export.h>
#include <CEGUI/CEGUIDefaultResourceProvider.h>
```

Classes

- class ResourceProvider

  A simple CEGUI ResourceProvider based on the CEGUI::DefaultResourceProvider.

Namespaces

- namespace dtGUI

  The dtGUI namespace contains the functionality to render and control OpenGL graphical user interfaces.

6.26 scriptmodule.cpp File Reference

```
#include <dtGUI/scriptmodule.h>
#include <dtUtil/log.h>
#include <CEGUI/CEGUIEventSet.h>
```

6.27 scriptmodule.h File Reference

```
#include <dtGUI/basescriptmodule.h>
#include <map>
#include <queue>
#include <string>
#include <dtUtil/functor.h>
```

Classes

- class ScriptModule

  ScriptModule is the binding from CEGUI::Events to specific application callbacks.

Namespaces

- namespace dtGUI

  The dtGUI namespace contains the functionality to render and control OpenGL graphical user interfaces.

6.28 widget.h File Reference

`#include <CEGUI/CEGUIWindow.h>`

Namespaces

- namespace dtGUI

  The dtGUI namespace contains the functionality to render and control OpenGL graphical user interfaces.

Typedefs

- typedef CEGUI::Window Widget

Index

- Symbols -

- A -

- B -

- C -

- D -

- E -

- G -

- H -

- I -

- K -

- L -

- M -

- O -

- R -

- S -

- U -

- W -