

Reference Manual

Generated by Doxygen 1.5.6

Wed Dec 10 03:49:03 2008

Contents

1	Class Index	1
1.1	Class List	1
2	Class Documentation	3
2.1	GLWidget Class Reference	3
2.1.1	Detailed Description	4
2.1.2	Constructor & Destructor Documentation	4
2.1.2.1	GLWidget	4
2.1.3	Member Function Documentation	4
2.1.3.1	minimumSizeHint	4
2.1.3.2	sizeHint	4
2.1.3.3	setVolume	5
2.1.3.4	setColorsPtr	5
2.1.3.5	setXRotation	5
2.1.3.6	setYRotation	5
2.1.3.7	setZRotation	5
2.1.3.8	setLength	5
2.1.3.9	setWidth	6
2.1.3.10	setDepth	6
2.1.3.11	setColors	6
2.1.3.12	setRenderMode	6
2.1.3.13	xRotationChanged	6
2.1.3.14	yRotationChanged	6
2.1.3.15	zRotationChanged	6
2.1.3.16	initializeGL	7
2.1.3.17	paintGL	7
2.1.3.18	resizeGL	7
2.1.3.19	mousePressEvent	7

2.1.3.20	mouseReleaseEvent	7
2.1.3.21	mouseMoveEvent	7
2.1.3.22	wheelEvent	7
2.1.3.23	drawSlice	8
2.1.3.24	normalizeAngle	8
2.1.3.25	file2string	8
2.1.3.26	loadShader	8
2.1.3.27	printLog	8
2.1.3.28	setVertexProp	9
2.1.3.29	renderBoundingBox	9
2.1.3.30	genTransferTexture	9
2.1.3.31	initBackfaceBuffer	9
2.1.3.32	updateRelVolSize	9
2.2	GradientEditor Class Reference	10
2.2.1	Detailed Description	10
2.2.2	Constructor & Destructor Documentation	10
2.2.2.1	GradientEditor	10
2.2.3	Member Function Documentation	11
2.2.3.1	setGradientStops	11
2.2.3.2	getColorsPtr	11
2.2.3.3	updateColorTexture	11
2.2.3.4	forwardHistogramPtr	11
2.2.3.5	pointsUpdated	11
2.2.3.6	colorDialog	11
2.2.3.7	moveColorsToAlpha	11
2.2.3.8	moveColorsWithAlpha	12
2.2.3.9	newColorPoints	12
2.2.3.10	deleteColorPoints	12
2.2.3.11	gradientStopsChanged	12
2.2.3.12	resizeEvent	12
2.3	HoverPoints Class Reference	13
2.3.1	Detailed Description	14
2.3.2	Member Enumeration Documentation	14
2.3.2.1	PointShape	14
2.3.2.2	LockType	14
2.3.2.3	SortType	14

2.3.2.4	ConnectionType	14
2.3.3	Constructor & Destructor Documentation	14
2.3.3.1	HoverPoints	14
2.3.4	Member Function Documentation	14
2.3.4.1	eventFilter	14
2.3.4.2	paintPoints	15
2.3.4.3	boundingRect	15
2.3.4.4	setBoundingRect	15
2.3.4.5	points	15
2.3.4.6	setPoints	15
2.3.4.7	pointSize	15
2.3.4.8	setPointSize	15
2.3.4.9	sortType	16
2.3.4.10	setSortType	16
2.3.4.11	connectionType	16
2.3.4.12	setConnectionType	16
2.3.4.13	setConnectionPen	16
2.3.4.14	setShapePen	16
2.3.4.15	setShapeBrush	16
2.3.4.16	setPointLock	17
2.3.4.17	setEditable	17
2.3.4.18	editable	17
2.3.4.19	setEnabled	17
2.3.4.20	setDisabled	17
2.3.4.21	pointsChanged	17
2.3.4.22	chooseColor	18
2.3.4.23	alphaChanged	18
2.3.4.24	alphaPointCreated	18
2.3.4.25	alphaPointDeleted	18
2.3.4.26	firePointChange	18
2.3.4.27	movePoint	18
2.3.4.28	newPoint	19
2.3.4.29	deletePoint	19
2.3.4.30	fireChooseColor	19
2.4	ShadeWidget Class Reference	20
2.4.1	Detailed Description	20

2.4.2	Member Enumeration Documentation	20
2.4.2.1	ShadeType	20
2.4.3	Constructor & Destructor Documentation	20
2.4.3.1	ShadeWidget	20
2.4.4	Member Function Documentation	21
2.4.4.1	setHistogramPtr	21
2.4.4.2	setGradientStops	21
2.4.4.3	paintEvent	21
2.4.4.4	sizeHint	21
2.4.4.5	points	21
2.4.4.6	hoverPoints	21
2.4.4.7	colorAt	21
2.4.4.8	colorsChanged	22
2.4.4.9	generateShade	22
2.5	Volume Class Reference	23
2.5.1	Detailed Description	23
2.5.2	Constructor & Destructor Documentation	23
2.5.2.1	Volume	23
2.5.3	Member Function Documentation	23
2.5.3.1	getSize	23
2.5.3.2	getMinDensity	23
2.5.3.3	getMaxDensity	24
2.5.3.4	getTexture	24
2.5.3.5	getHistogramPtr	24
2.5.3.6	load	24
2.5.3.7	createTexture	24
2.5.3.8	createHistogram	24
2.6	Window Class Reference	25
2.6.1	Detailed Description	25
2.6.2	Constructor & Destructor Documentation	25
2.6.2.1	Window	25
2.6.3	Member Function Documentation	25
2.6.3.1	setDefaultGradientStops	25
2.6.3.2	setOpenFileName	25
2.6.3.3	saveGradientStops	26
2.6.3.4	colorTextureChanged	26

2.6.3.5	createSlider	26
---------	------------------------------	----

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

GLWidget	3
GradientEditor	10
HoverPoints	13
ShadeWidget	20
Volume	23
Window	25

Chapter 2

Class Documentation

2.1 GLWidget Class Reference

```
#include <glwidget.h>
```

Public Slots

- void [setXRotation](#) (int angle)
- void [setYRotation](#) (int angle)
- void [setZRotation](#) (int angle)
- void [setLength](#) (int length)
- void [setWidth](#) (int width)
- void [setDepth](#) (int depth)
- void [setColors](#) ()
- void [setRenderMode](#) (bool mode)

Signals

- void [xRotationChanged](#) (int angle)
- void [yRotationChanged](#) (int angle)
- void [zRotationChanged](#) (int angle)

Public Member Functions

- [GLWidget](#) (QWidget *parent=0)
- QSize [minimumSizeHint](#) () const
- QSize [sizeHint](#) () const
- void [setVolume](#) (Volume *newVolume)
- void [setColorsPtr](#) (unsigned int *colorsPtr)

Protected Member Functions

- void [initializeGL](#) ()
- void [paintGL](#) ()
- void [resizeGL](#) (int width, int height)
- void [mousePressEvent](#) (QMouseEvent *event)
- void [mouseReleaseEvent](#) (QMouseEvent *event)
- void [mouseMoveEvent](#) (QMouseEvent *event)
- void [wheelEvent](#) (QWheelEvent *event)

Private Member Functions

- void [drawSlice](#) ()
- void [normalizeAngle](#) (int *angle)
- char * [file2string](#) (const std::string &strFilename)
- GLuint [loadShader](#) (const std::string &strFilename)
- void [printLog](#) (GLuint obj)
- void [setVertexProp](#) (float s, float t, float p)
- void [renderBoundingBox](#) ()
- void [genTransferTexture](#) ()
- void [initBackfaceBuffer](#) ()
- void [updateRelVolSize](#) ()

2.1.1 Detailed Description

Sets up a Qt OpenGL Canvas Widget.

2.1.2 Constructor & Destructor Documentation

2.1.2.1 GLWidget::GLWidget (QWidget * *parent* = 0)

Sets up a Qt OpenGL Canvas.

Parameters:

parent is the parent window of the canvas.

2.1.3 Member Function Documentation

2.1.3.1 QSize GLWidget::minimumSizeHint () const

Returns:

The minimum size for the Widget.

2.1.3.2 QSize GLWidget::sizeHint () const

Returns:

The size hint for the Widget.

2.1.3.3 void GLWidget::setVolume (Volume * *newVolume*)

Sets up the Canvas for a new [Volume](#) object.

Parameters:

newVolume is the new [Volume](#).

2.1.3.4 void GLWidget::setColorPtr (unsigned int * *colorsPtr*)

Initializes the pointer to the transferfunction array.

Parameters:

colorsPtr is the pointer to the new array.

2.1.3.5 void GLWidget::setXRotation (int *angle*) [slot]

Handles rotation of the volume around the X-axis.

Parameters:

angle is the new X-angle.

2.1.3.6 void GLWidget::setYRotation (int *angle*) [slot]

Handles rotation of the volume around the Y-axis.

Parameters:

angle is the new Y-angle.

2.1.3.7 void GLWidget::setZRotation (int *angle*) [slot]

Handles rotation of the volume around the Z-axis.

Parameters:

angle is the new Z-angle.

2.1.3.8 void GLWidget::setLength (int *length*) [slot]

Sets the position and the slice shown by the YZ-Quad in slicemode.

Parameters:

length is the position of the new slice ([0 .. 1] * 5760).

2.1.3.9 void GLWidget::setWidth (int *width*) [slot]

Sets the position and the slice shown by the XZ-Quad in slicemode.

Parameters:

width is the position of the new slice ([0 .. 1] * 5760).

2.1.3.10 void GLWidget::setDepth (int *depth*) [slot]

Sets the position and the slice shown by the XY-Quad in slicemode.

Parameters:

depth is the position of the new slice ([0 .. 1] * 5760).

2.1.3.11 void GLWidget::setColors () [slot]

Updatefunction, called whenever the transferfunction is changed. Sends the new transfertexture to the graphics card.

2.1.3.12 void GLWidget::setRenderMode (bool *mode*) [slot]

Sets the current Rendermode.

Parameters:

mode is the new renderMode (0 - raycasting, 1 - sliceview)

2.1.3.13 void GLWidget::xRotationChanged (int *angle*) [signal]

A signal that is send on a change in the x rotation.

Parameters:

angle is the angle of the new rotation.

2.1.3.14 void GLWidget::yRotationChanged (int *angle*) [signal]

A signal that is send on a change in the y rotation.

Parameters:

angle is the angle of the new rotation.

2.1.3.15 void GLWidget::zRotationChanged (int *angle*) [signal]

A signal that is send on a change in the z rotation.

Parameters:

angle is the angle of the new rotation.

2.1.3.16 void GLWidget::initializeGL () [protected]

Initializes the OpenGL environment including lighting, textures and shaders.

2.1.3.17 void GLWidget::paintGL () [protected]

Applies affine transformations and starts the drawing of the frame.

2.1.3.18 void GLWidget::resizeGL (int *width*, int *height*) [protected]

Handles resizing of the OpenGL canvas.

Parameters:

width is the new width of the canvas.

height is the new height of the canvas.

2.1.3.19 void GLWidget::mousePressEvent (QMouseEvent * *event*) [protected]

Reduces the stepsize of the raycastingshader and initialises volume rotations.

Parameters:

event is the mousePressEvent.

2.1.3.20 void GLWidget::mouseReleaseEvent (QMouseEvent * *event*) [protected]

Increases the stepsize of the raycastingshader after finished volume rotations.

Parameters:

event is the mouseReleaseEvent.

2.1.3.21 void GLWidget::mouseMoveEvent (QMouseEvent * *event*) [protected]

Handles mouse dragging and rotation of the volume.

Parameters:

event is the mouseMoveEvent.

2.1.3.22 void GLWidget::wheelEvent (QWheelEvent * *event*) [protected]

Handles mouseWheelEvents.

Parameters:

event is the mouseWheelEvent.

2.1.3.23 void GLWidget::drawSlice () [private]

Renders the 3d-Objects if the current rendermode. Rendermode 0: Raycastingview, renders boundingbox backside into the FBO and frontside into the normal buffer. Rendermode 1: Sliceview, draws 3 Quads depending on current lengthf, widthf and heightf.

2.1.3.24 void GLWidget::normalizeAngle (int * *angle*) [private]

Keeps angle within [0 360] interval.

Parameters:

angle is the pointer to the angle.

2.1.3.25 char * GLWidget::file2string (const std::string & *strFilename*) [private]

Loads a binary file into a Charpointer.

Parameters:

strFilename ist the filename of the file to load.

Returns:

The Charpointer with the content of the file.

2.1.3.26 GLuint GLWidget::loadShader (const std::string & *strFilename*) [private]

Handels loading, compiling and attaching of a vertex-, fragmentshader pair to a shader program.

Parameters:

strFilename is the filepath of the shaders without extension. Loading requires 2 glsl shader files with "<filepath>.vert" and "<filepath>.frag" extension to be successful.

Returns:

The id of the new shaderprogram.

2.1.3.27 void GLWidget::printLog (GLuint *obj*) [private]

Prints the shaderobject debug log.

Parameters:

obj specifies the shaderobject.

2.1.3.28 void GLWidget::setVertexProp (float *s*, float *t*, float *p*) [private]

Sets glColor and glTexCoords with the given parameters.

Parameters:

s

t

p

2.1.3.29 void GLWidget::renderBoundingBox () [private]

Renders the boundingbox of the volume with the longest side normalized to 1.

2.1.3.30 void GLWidget::genTransferTexture () [private]

Generates the Transferfunction-texture and sends it to the graphics card.

2.1.3.31 void GLWidget::initBackfaceBuffer () [private]

Generates framebufferobject and texture for the backface rendering.

2.1.3.32 void GLWidget::updateRelVolSize () [private]

Calculates the relative volumesizes the the longest side normalized to 1.

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

- glwidget.h
- glwidget.cpp

2.2 GradientEditor Class Reference

```
#include <gradienteditor.h>
```

Public Slots

- void [pointsUpdated](#) ()
- void [colorDialog](#) (int index, QWidget *selectedWidget)
- void [moveColorsToAlpha](#) (bool state)
- void [moveColorsWithAlpha](#) (int index, qreal newXPos)
- void [newColorPoints](#) (int index, qreal xPos)
- void [deleteColorPoints](#) (int index)

Signals

- void [gradientStopsChanged](#) (const QGradientStops &stops)

Public Member Functions

- [GradientEditor](#) (QWidget *parent)
- void [setGradientStops](#) (const QGradientStops &stops)
- unsigned int * [getColorsPtr](#) ()
- void [updateColorTexture](#) (int minDensity=0, int maxDensity=4095)
- void [forwardHistogramPtr](#) (QPixmap *histogramPtr)

Protected Member Functions

- void [resizeEvent](#) (QResizeEvent *event)

2.2.1 Detailed Description

The gradienteditor class is a collection of a red, green, blue and alpha [ShadeWidget](#).

2.2.2 Constructor & Destructor Documentation

2.2.2.1 GradientEditor::GradientEditor (QWidget *parent)

Constructor that creates the transfer function editor.

Returns:

The pointer to the created transfer function editor.

2.2.3 Member Function Documentation

2.2.3.1 void GradientEditor::setGradientStops (const QGradientStops & stops)

Sets the gradient stops position and color from the given stops.

Parameters:

stops are the gradient stops which should be set.

2.2.3.2 unsigned int * GradientEditor::getColorPtr ()

Returns:

The pointer to the transfer texture.

2.2.3.3 void GradientEditor::updateColorTexture (int minDensity = 0, int maxDensity = 4095)

Updates the transfer texture with the color currently set in ShadeWidgets.

2.2.3.4 void GradientEditor::forwardHistogramPtr (QPixmap * histogramPtr)

Forwards the QPixmap pointer to the alpha [ShadeWidget](#).

Parameters:

histogramPtr is the pointer which should be forwarded.

2.2.3.5 void GradientEditor::pointsUpdated () [slot]

Updates the gradient stops position and color from the currently set points. Sets this gradient points at the alpha [ShadeWidget](#) and emits a gradientStopsChanged signal.

2.2.3.6 void GradientEditor::colorDialog (int index, QWidget * selectedWidget) [slot]

Creates a color dialog with which a new color can be set.

Parameters:

index is the index of the point for which the color should be changed.

selectedWidget is a pointer to the [ShadeWidget](#) from whom the x pos for the other 3 points, which should be changed, is taken.

2.2.3.7 void GradientEditor::moveColorsToAlpha (bool state) [slot]

Sets the state of colorsFixedToAlpha. Creates the color points over each alpha point with the current color of the alpha point or not (dependent on the state). Also sets the 3 color ShadeWidgets as editable or not (dependent on the state).

Parameters:

state is the state of the checkbox which calls this function.

2.2.3.8 void GradientEditor::moveColorsWithAlpha (int *index*, qreal *newXPos*) [slot]

Moves the 3 color points with the corresponding alpha point if colorsFixatedToAlpha is true.

Parameters:

index is the index of the points which should be moved.

newXPos is the new x position to which the corresponding points should be moved.

2.2.3.9 void GradientEditor::newColorPoints (int *index*, qreal *xPos*) [slot]

Creates the 3 color points for the corresponding alpha point if colorsFixatedToAlpha is true.

Parameters:

index is the index where the points should be saved.

xPos is the x position where the points should be created.

2.2.3.10 void GradientEditor::deleteColorPoints (int *index*) [slot]

Deletes the 3 color points of the corresponding alpha point if colorsFixatedToAlpha is true.

Parameters:

index is the index where the points should be deleted.

2.2.3.11 void GradientEditor::gradientStopsChanged (const QGradientStops & *stops*) [signal]

A signal that is send on change of the gradient stops.

Parameters:

stops are the gradient stops which changed.

2.2.3.12 void GradientEditor::resizeEvent (QResizeEvent * *event*) [protected]

Updates the gradient points at each [GradientEditor](#) resizing.

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

- `gradienteditor.h`
- `gradienteditor.cpp`

2.3 HoverPoints Class Reference

```
#include <hoverpoints.h>
```

Public Types

- enum [PointShape](#)
- enum [LockType](#)
- enum [SortType](#)
- enum [ConnectionType](#)

Public Slots

- void [setEnabled](#) (bool enabled)
- void [setDisabled](#) (bool disabled)

Signals

- void [pointsChanged](#) (const QPolygonF &points)
- void [chooseColor](#) (int index, QWidget *selectedWidget)
- void [alphaChanged](#) (int index, qreal newXPos)
- void [alphaPointCreated](#) (int index, qreal xPos)
- void [alphaPointDeleted](#) (int index)

Public Member Functions

- [HoverPoints](#) (QWidget *widget, [PointShape](#) shape)
- bool [eventFilter](#) (QObject *object, QEvent *event)
- void [paintPoints](#) ()
- QRectF [boundingRect](#) () const
- void [setBoundingRect](#) (const QRectF &boundingRect)
- QPolygonF [points](#) () const
- void [setPoints](#) (const QPolygonF &points)
- QSizeF [pointSize](#) () const
- void [setPointSize](#) (const QSizeF &size)
- [SortType](#) [sortType](#) () const
- void [setSortType](#) ([SortType](#) sortType)
- [ConnectionType](#) [connectionType](#) () const
- void [setConnectionType](#) ([ConnectionType](#) connectionType)
- void [setConnectionPen](#) (const QPen &pen)
- void [setShapePen](#) (const QPen &pen)
- void [setShapeBrush](#) (const QBrush &brush)
- void [setPointLock](#) (int pos, [LockType](#) lock)
- void [setEditable](#) (bool editable)
- bool [editable](#) () const
- void [firePointChange](#) ()
- void [movePoint](#) (int i, const QPointF &newPos, bool emitChange=true)

Private Slots

- void [newPoint](#) ()
- void [deletePoint](#) ()
- void [fireChooseColor](#) ()

2.3.1 Detailed Description

The `hoverpoints` class represents the points which are used in the [ShadeWidget](#).

2.3.2 Member Enumeration Documentation

2.3.2.1 enum `HoverPoints::PointShape`

Enumerates the `PointShape` options.

2.3.2.2 enum `HoverPoints::LockType`

Enumerates the `LockType` options.

2.3.2.3 enum `HoverPoints::SortType`

Enumerates the `SortType` options.

2.3.2.4 enum `HoverPoints::ConnectionType`

Enumerates the `ConnectionType` options.

2.3.3 Constructor & Destructor Documentation

2.3.3.1 `HoverPoints::HoverPoints (QWidget * widget, PointShape shape)`

Constructor that creates the [HoverPoints](#).

Returns:

The pointer to the created [HoverPoints](#).

2.3.4 Member Function Documentation

2.3.4.1 `bool HoverPoints::eventFilter (QObject * object, QEvent * event)`

Filters the events and takes appropriate actions.

Parameters:

object is the `QObject` which caused the `QEvent`.

event is the `QEvent` which happend.

Returns:

True if it was a valid object and a valid event, false otherwise.

2.3.4.2 void HoverPoints::paintPoints ()

Paints the points and connections.

2.3.4.3 QRectF HoverPoints::boundingRect () const [inline]**Returns:**

The bounding rectangle for this hover points.

2.3.4.4 void HoverPoints::setBoundingRect (const QRectF & boundingRect) [inline]

Sets the new bounding rectangle for this hover points.

Parameters:

boundingRect is the new bounding rectangle which should be set.

2.3.4.5 QPolygonF HoverPoints::points () const [inline]**Returns:**

The hover points.

2.3.4.6 void HoverPoints::setPoints (const QPolygonF & points)

Sets points position from the given points.

Parameters:

points are the points which should be set.

2.3.4.7 QSizeF HoverPoints::pointSize () const [inline]**Returns:**

The pixel size of one hover point (for drawing).

2.3.4.8 void HoverPoints::setPointSize (const QSizeF & size) [inline]

Sets the new pixel size for this hover points.

Parameters:

size is the new pixel size which should be set.

2.3.4.9 `SortType HoverPoints::sortType () const` [inline]**Returns:**

The sort type, in which the hover points should be sorted.

2.3.4.10 `void HoverPoints::setSortType (SortType sortType)` [inline]

Sets the way, in which the hover points should be sorted.

Parameters:

sortType is the new sortType which should be set.

2.3.4.11 `ConnectionType HoverPoints::connectionType () const` [inline]**Returns:**

The way in which the point connections are drawn.

2.3.4.12 `void HoverPoints::setConnectionType (ConnectionType connectionType)` [inline]

Sets the way, in which the point connections should be drawn.

Parameters:

connectionType is the new connectionType which should be set.

2.3.4.13 `void HoverPoints::setConnectionPen (const QPen & pen)` [inline]

Sets the pen, which draws the point connections.

Parameters:

pen is the new pen which should be set.

2.3.4.14 `void HoverPoints::setShapePen (const QPen & pen)` [inline]

Sets the pen, which draws the shape of the points.

Parameters:

pen is the new pen which should be set.

2.3.4.15 `void HoverPoints::setShapeBrush (const QBrush & brush)` [inline]

Sets the brush, which fills the shape of the points.

Parameters:

brush is the new brush which should be set.

2.3.4.16 void HoverPoints::setPointLock (int *pos*, LockType *lock*) [inline]

Sets the point lock for a point.

Parameters:

pos is the index of the point from which the lock should be set.

lock is the typ of lock which should be set.

2.3.4.17 void HoverPoints::setEditable (bool *editable*) [inline]

Sets this points editable state.

Parameters:

editable is true if the points are editable and flase otherwise.

2.3.4.18 bool HoverPoints::editable () const [inline]**Returns:**

true if the points are editable and flase otherwise.

2.3.4.19 void HoverPoints::setEnabled (bool *enabled*) [slot]

Enables or disables these HoverPints.

Parameters:

enabled is true if the points are enabled and flase otherwise.

2.3.4.20 void HoverPoints::setDisabled (bool *disabled*) [inline, slot]

Enables or disables these HoverPints.

Parameters:

disabled is true if the points are disabled and flase otherwise.

2.3.4.21 void HoverPoints::pointsChanged (const QPolygonF & *points*) [signal]

A signal that is send on change of the points.

Parameters:

points are the new points.

2.3.4.22 void HoverPoints::chooseColor (int *index*, QWidget * *selectedWidget*) [signal]

A signal that is send to initiate a color dialog.

Parameters:

index is the index of the point for which the color should be changed.

selectedWidget is a pointer to the [ShadeWidget](#) from whom the x pos for the other 3 points, which should be changed, is taken.

2.3.4.23 void HoverPoints::alphaChanged (int *index*, qreal *newXPos*) [signal]

A signal that is send if an alpha point got moved.

Parameters:

index is the index of the points which should be moved.

newXPos is the new x position to which the corresponding points should be moved.

2.3.4.24 void HoverPoints::alphaPointCreated (int *index*, qreal *xPos*) [signal]

A signal that is send if an alpha point got created.

Parameters:

index is the index where the points should be saved.

xPos is the x position where the points should be created.

2.3.4.25 void HoverPoints::alphaPointDeleted (int *index*) [signal]

A signal that is send if an alpha point got deleted.

Parameters:

index is the index where the points should be deleted.

2.3.4.26 void HoverPoints::firePointChange ()

Sorts the points and emits pointsChanged.

2.3.4.27 void HoverPoints::movePoint (int *index*, const QPointF & *point*, bool *emitUpdate* = true)

Moves the given point.

Parameters:

index is the index of the point which should be moved.

point is the point which contains the new position.

emitUpdate specifies if alphaChanged and pointsChanged should be emitted.

2.3.4.28 void HoverPoints::newPoint () [private, slot]

Creates a new point. Emits a pointsChanged and an alphaPointCreated.

2.3.4.29 void HoverPoints::deletePoint () [private, slot]

Deletes a point. Emits a pointsChanged and an alphaPointCreated.

2.3.4.30 void HoverPoints::fireChooseColor () [private, slot]

Emits a chooseColor.

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

- hoverpoints.h
- hoverpoints.cpp

2.4 ShadeWidget Class Reference

```
#include <shadewidget.h>
```

Public Types

- enum [ShadeType](#)

Signals

- void [colorsChanged](#) ()

Public Member Functions

- [ShadeWidget](#) ([ShadeType](#) type, QWidget *parent)
- void [setHistogramPtr](#) (QPixmap *histogramPtr)
- void [setGradientStops](#) (const QGradientStops &stops)
- void [paintEvent](#) (QPaintEvent *e)
- QSize [sizeHint](#) () const
- QPolygonF [points](#) () const
- [HoverPoints](#) * [hoverPoints](#) () const
- uint [colorAt](#) (int x)

Private Member Functions

- void [generateShade](#) ()

2.4.1 Detailed Description

The shadewidget class is a drawable rectangle in which points and a color or alpha gradient can be drawn.

2.4.2 Member Enumeration Documentation

2.4.2.1 enum ShadeWidget::ShadeType

Enumerates the ShateType options.

2.4.3 Constructor & Destructor Documentation

2.4.3.1 ShadeWidget::ShadeWidget (ShadeType type, QWidget * parent)

Constructor that creates a color or alpha [ShadeWidget](#).

Returns:

The pointer to the created color or alpha [ShadeWidget](#).

2.4.4 Member Function Documentation

2.4.4.1 void ShadeWidget::setHistogramPtr (QPixmap * *histogramPtr*)

Sets the histogram pointer if the widget is a ARGBShade widget.

Parameters:

histogramPtr is the pointer which should be set.

2.4.4.2 void ShadeWidget::setGradientStops (const QGradientStops & *stops*)

Sets the gradient stops position and color from the given stops.

Parameters:

stops are the gradient stops which should be set.

2.4.4.3 void ShadeWidget::paintEvent (QPaintEvent * *e*)

Paints the background, points and the connection lines.

2.4.4.4 QSize ShadeWidget::sizeHint () const

Returns:

The size hint for the Widget.

2.4.4.5 QPolygonF ShadeWidget::points () const

Returns:

The points of this [ShadeWidget](#).

2.4.4.6 HoverPoints* ShadeWidget::hoverPoints () const [inline]

Returns:

The pointer to the hover points of this [ShadeWidget](#).

2.4.4.7 uint ShadeWidget::colorAt (int *x*)

Gets the current color of the ShadeWidget at the specific position.

Parameters:

x is the position from which the color should be taken.

Returns:

The color at the position.

2.4.4.8 void ShadeWidget::colorsChanged () [signal]

A signal that is send on change of the colors.

2.4.4.9 void ShadeWidget::generateShade () [private]

Generates the backround.

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

- shadewidget.h
- shadewidget.cpp

2.5 Volume Class Reference

```
#include <Volume.h>
```

Public Member Functions

- [Volume](#) (const std::string &strFilename)
- int * [getSize](#) ()
- int [getMinDensity](#) () const
- int [getMaxDensity](#) () const
- GLuint [getTexture](#) () const
- QPixmap * [getHistogramPtr](#) () const

Private Member Functions

- void [load](#) (const std::string &strFilename)
- void [createTexture](#) ()
- void [createHistogram](#) ()

2.5.1 Detailed Description

Represents a new [Volume](#) object.

2.5.2 Constructor & Destructor Documentation

2.5.2.1 [Volume::Volume](#) (const std::string & *strFilename*)

Creates a new [Volume](#) object and starts loading a file.

Parameters:

strFilename is the filename of the .dat volume data.

2.5.3 Member Function Documentation

2.5.3.1 int * [Volume::getSize](#) ()

Get funtion for the Volumedimensions.

Returns:

The size of the volume.

2.5.3.2 int [Volume::getMinDensity](#) () const [inline]

Returns:

The minimum density.

2.5.3.3 `int Volume::getMaxDensity () const` [inline]**Returns:**

The maximum density.

2.5.3.4 `GLuint Volume::getTexture () const` [inline]**Returns:**

The id of the texture.

2.5.3.5 `QPixmap* Volume::getHistogramPtr () const` [inline]**Returns:**

The pointer to the histogram pixmap.

2.5.3.6 `void Volume::load (const std::string & strFilename)` [private]

Loads the volume data from a binary file and precalculates the gradient at each voxel.

Parameters:

strFilename is the filename of the .dat volume data.

2.5.3.7 `void Volume::createTexture ()` [private]

Generates the Volumetexture and sends it to the graphics card.

2.5.3.8 `void Volume::createHistogram ()` [private]

Generates the histogram for the gradienteditor.

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

- Volume.h
- Volume.cpp

2.6 Window Class Reference

```
#include <window.h>
```

Signals

- void [colorTextureChanged](#) ()

Public Member Functions

- [Window](#) ()

Private Slots

- void [setDefaultGradientStops](#) ()
- void [setOpenFileName](#) ()
- void [saveGradientStops](#) (const QGradientStops &stops)

Private Member Functions

- QSlider * [createSlider](#) (QWidget *parent=0)

2.6.1 Detailed Description

The window class represents the whole application window.

2.6.2 Constructor & Destructor Documentation

2.6.2.1 Window::Window ()

Constructor that creates the GUI.

2.6.3 Member Function Documentation

2.6.3.1 void Window::setDefaultGradientStops () [private, slot]

Loads and sets the gradient stops for the current volume.

2.6.3.2 void Window::setOpenFileName () [private, slot]

Creates a file dialog with which a [Volume](#) can be loaded.

2.6.3.3 void Window::saveGradientStops (const QGradientStops & stops) [private, slot]

Saves the given gradient stops into a file, calls for a update of the transfer texture and emits a colorTextureChanged signal.

Parameters:

stops are the gradient stops which should be saved.

2.6.3.4 void Window::colorTextureChanged () [signal]

A signal that is send on change of the transfer texture.

2.6.3.5 QSlider * Window::createSlider (QWidget * parent = 0) [private]

Creates a QSlider.

Parameters:

parent is the parent of the slider.

Returns:

The pointer to the created QSlider.

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

- window.h
- window.cpp

Index

- alphaChanged
 - HoverPoints, 18
- alphaPointCreated
 - HoverPoints, 18
- alphaPointDeleted
 - HoverPoints, 18
- boundingRect
 - HoverPoints, 15
- chooseColor
 - HoverPoints, 17
- colorAt
 - ShadeWidget, 21
- colorDialog
 - GradientEditor, 11
- colorsChanged
 - ShadeWidget, 21
- colorTextureChanged
 - Window, 26
- ConnectionType
 - HoverPoints, 14
- connectionType
 - HoverPoints, 16
- createHistogram
 - Volume, 24
- createSlider
 - Window, 26
- createTexture
 - Volume, 24
- deleteColorPoints
 - GradientEditor, 12
- deletePoint
 - HoverPoints, 19
- drawSlice
 - GLWidget, 7
- editable
 - HoverPoints, 17
- eventFilter
 - HoverPoints, 14
- file2string
 - GLWidget, 8
- fireChooseColor
 - HoverPoints, 19
- firePointChange
 - HoverPoints, 18
- forwardHistogramPtr
 - GradientEditor, 11
- generateShade
 - ShadeWidget, 22
- genTransferTexture
 - GLWidget, 9
- getColorsPtr
 - GradientEditor, 11
- getHistogramPtr
 - Volume, 24
- getMaxDensity
 - Volume, 23
- getMinDensity
 - Volume, 23
- getSize
 - Volume, 23
- getTexture
 - Volume, 24
- GLWidget, 3
 - drawSlice, 7
 - file2string, 8
 - genTransferTexture, 9
 - GLWidget, 4
 - initBackfaceBuffer, 9
 - initializeGL, 6
 - loadShader, 8
 - minimumSizeHint, 4
 - mouseMoveEvent, 7
 - mousePressEvent, 7
 - mouseReleaseEvent, 7
 - normalizeAngle, 8
 - paintGL, 7
 - printLog, 8
 - renderBoundingBox, 9
 - resizeGL, 7
 - setColors, 6
 - setColorsPtr, 5
 - setDepth, 6
 - setLength, 5
 - setRenderMode, 6
 - setVertexProp, 8

- setVolume, 4
- setWidth, 5
- setXRotation, 5
- setYRotation, 5
- setZRotation, 5
- sizeHint, 4
- updateRelVolSize, 9
- wheelEvent, 7
- xRotationChanged, 6
- yRotationChanged, 6
- zRotationChanged, 6
- GradientEditor, 10
 - colorDialog, 11
 - deleteColorPoints, 12
 - forwardHistogramPtr, 11
 - getColorPtr, 11
 - GradientEditor, 10
 - gradientStopsChanged, 12
 - moveColorsToAlpha, 11
 - moveColorsWithAlpha, 11
 - newColorPoints, 12
 - pointsUpdated, 11
 - resizeEvent, 12
 - setGradientStops, 11
 - updateColorTexture, 11
- gradientStopsChanged
 - GradientEditor, 12
- HoverPoints, 13
 - alphaChanged, 18
 - alphaPointCreated, 18
 - alphaPointDeleted, 18
 - boundingRect, 15
 - chooseColor, 17
 - ConnectionType, 14
 - connectionType, 16
 - deletePoint, 19
 - editable, 17
 - eventFilter, 14
 - fireChooseColor, 19
 - firePointChange, 18
 - HoverPoints, 14
 - LockType, 14
 - movePoint, 18
 - newPoint, 18
 - paintPoints, 15
 - points, 15
 - pointsChanged, 17
 - PointShape, 14
 - pointSize, 15
 - setBoundingRect, 15
 - setConnectionPen, 16
 - setConnectionType, 16
 - setDisabled, 17
 - setEditable, 17
 - setEnabled, 17
 - setPointLock, 16
 - setPoints, 15
 - setPointSize, 15
 - setShapeBrush, 16
 - setShapePen, 16
 - setSortType, 16
 - SortType, 14
 - sortType, 15
- hoverPoints
 - ShadeWidget, 21
- initBackfaceBuffer
 - GLWidget, 9
- initializeGL
 - GLWidget, 6
- load
 - Volume, 24
- loadShader
 - GLWidget, 8
- LockType
 - HoverPoints, 14
- minimumSizeHint
 - GLWidget, 4
- mouseMoveEvent
 - GLWidget, 7
- mousePressEvent
 - GLWidget, 7
- mouseReleaseEvent
 - GLWidget, 7
- moveColorsToAlpha
 - GradientEditor, 11
- moveColorsWithAlpha
 - GradientEditor, 11
- movePoint
 - HoverPoints, 18
- newColorPoints
 - GradientEditor, 12
- newPoint
 - HoverPoints, 18
- normalizeAngle
 - GLWidget, 8
- paintEvent
 - ShadeWidget, 21
- paintGL
 - GLWidget, 7
- paintPoints
 - HoverPoints, 15
- points
 - HoverPoints, 15

- ShadeWidget, 21
- pointsChanged
 - HoverPoints, 17
- PointShape
 - HoverPoints, 14
- pointSize
 - HoverPoints, 15
- pointsUpdated
 - GradientEditor, 11
- printLog
 - GLWidget, 8
- renderBoundingBox
 - GLWidget, 9
- resizeEvent
 - GradientEditor, 12
- resizeGL
 - GLWidget, 7
- saveGradientStops
 - Window, 25
- setBoundingRect
 - HoverPoints, 15
- setColors
 - GLWidget, 6
- setColorsPtr
 - GLWidget, 5
- setConnectionPen
 - HoverPoints, 16
- setConnectionType
 - HoverPoints, 16
- setDefaultGradientStops
 - Window, 25
- setDepth
 - GLWidget, 6
- setDisabled
 - HoverPoints, 17
- setEditable
 - HoverPoints, 17
- setEnabled
 - HoverPoints, 17
- setGradientStops
 - GradientEditor, 11
 - ShadeWidget, 21
- setHistogramPtr
 - ShadeWidget, 21
- setLength
 - GLWidget, 5
- setOpenFileName
 - Window, 25
- setPointLock
 - HoverPoints, 16
- setPoints
 - HoverPoints, 15
- setPointSize
 - HoverPoints, 15
- setRenderMode
 - GLWidget, 6
- setShapeBrush
 - HoverPoints, 16
- setShapePen
 - HoverPoints, 16
- setSortType
 - HoverPoints, 16
- setVertexProp
 - GLWidget, 8
- setVolume
 - GLWidget, 4
- setWidth
 - GLWidget, 5
- setXRotation
 - GLWidget, 5
- setYRotation
 - GLWidget, 5
- setZRotation
 - GLWidget, 5
- ShadeType
 - ShadeWidget, 20
- ShadeWidget, 20
 - colorAt, 21
 - colorsChanged, 21
 - generateShade, 22
 - hoverPoints, 21
 - paintEvent, 21
 - points, 21
 - setGradientStops, 21
 - setHistogramPtr, 21
 - ShadeType, 20
 - ShadeWidget, 20
 - sizeHint, 21
- sizeHint
 - GLWidget, 4
 - ShadeWidget, 21
- SortType
 - HoverPoints, 14
- sortType
 - HoverPoints, 15
- updateColorTexture
 - GradientEditor, 11
- updateRelVolSize
 - GLWidget, 9
- Volume, 23
 - createHistogram, 24
 - createTexture, 24
 - getHistogramPtr, 24
 - getMaxDensity, 23

- getMinDensity, [23](#)
- getSize, [23](#)
- getTexture, [24](#)
- load, [24](#)
- Volume, [23](#)

- wheelEvent
 - GLWidget, [7](#)
- Window, [25](#)
 - colorTextureChanged, [26](#)
 - createSlider, [26](#)
 - saveGradientStops, [25](#)
 - setDefaultGradientStops, [25](#)
 - setOpenFileName, [25](#)
 - Window, [25](#)

- xRotationChanged
 - GLWidget, [6](#)

- yRotationChanged
 - GLWidget, [6](#)

- zRotationChanged
 - GLWidget, [6](#)