

DOCUMENTATION

Carina Weber, Jan Benedikt Schwarz, Johannes Werner, Noel Schuhmacher, Sascha Rapp, Simon Grafenhorst

Contents

1	Introduction	3
2	Diagrams	5
3	Namespace Documentation	13
4	Data Structure Documentation	19
Index		261
Appendix		261

Chapter 1

Introduction

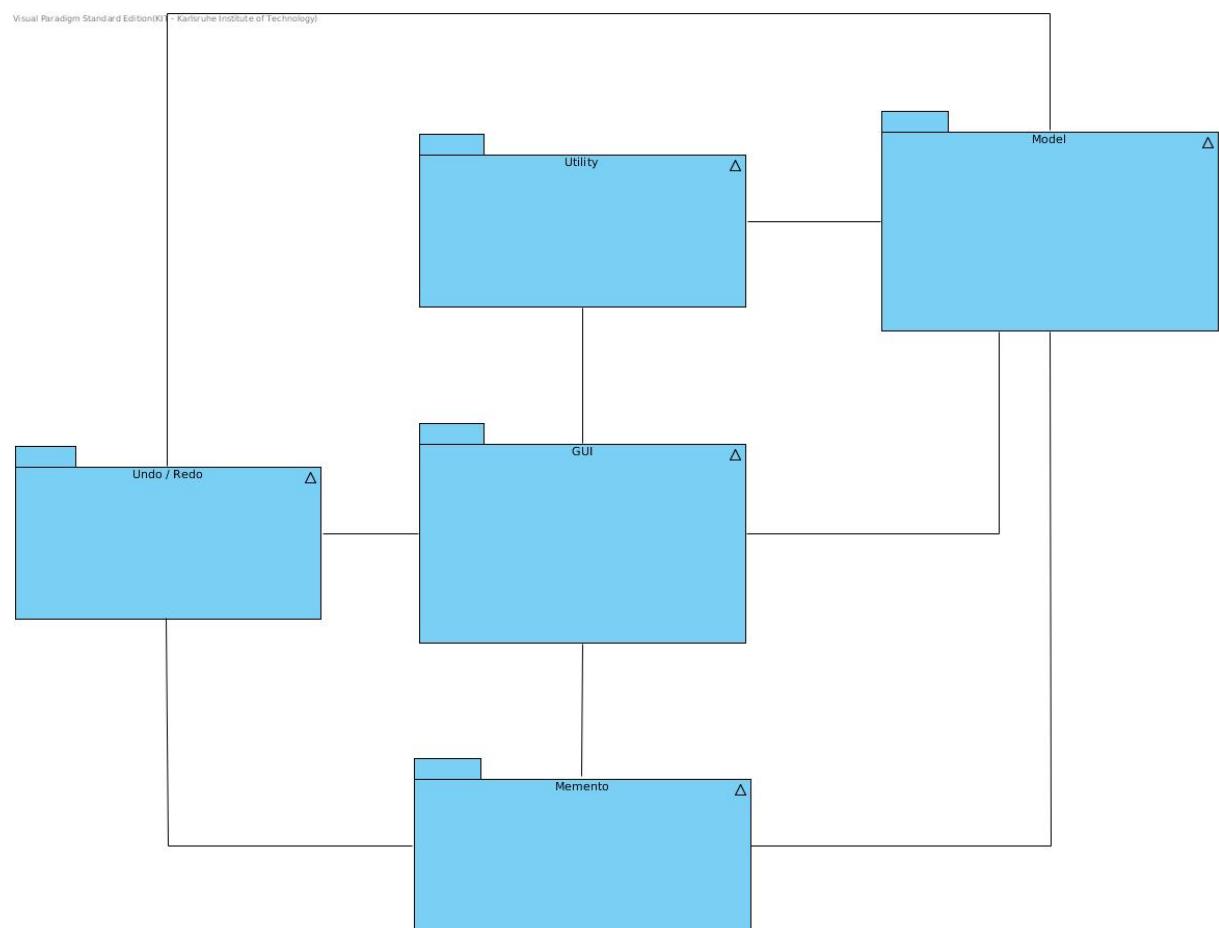
Vive (short for: Video veritatem) is a programm for testing different videoencoders. This is the documentation in which diagrams and specifications of the classes are included. It is intended as a guideline for the implementation. This document is separated into different parts: in the first part are diagrams to give a rough overview of the structure of the programm. In the second part are detailed descriptions of every class and methode. An overview of the implementation phase is given in the form of a gantt-diagram in the appendix.

Chapter 2

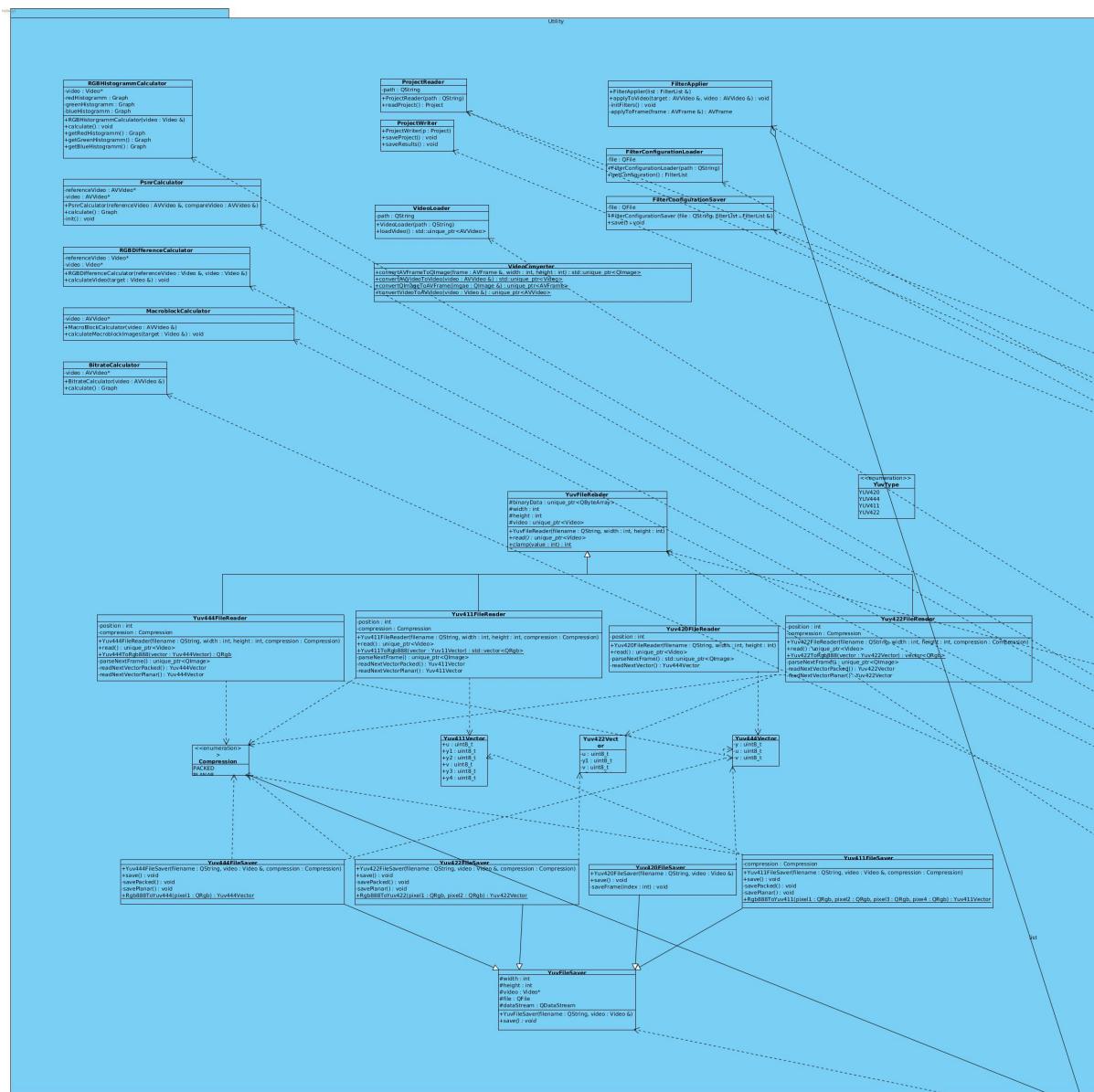
Diagrams

2.1 Class Diagrams

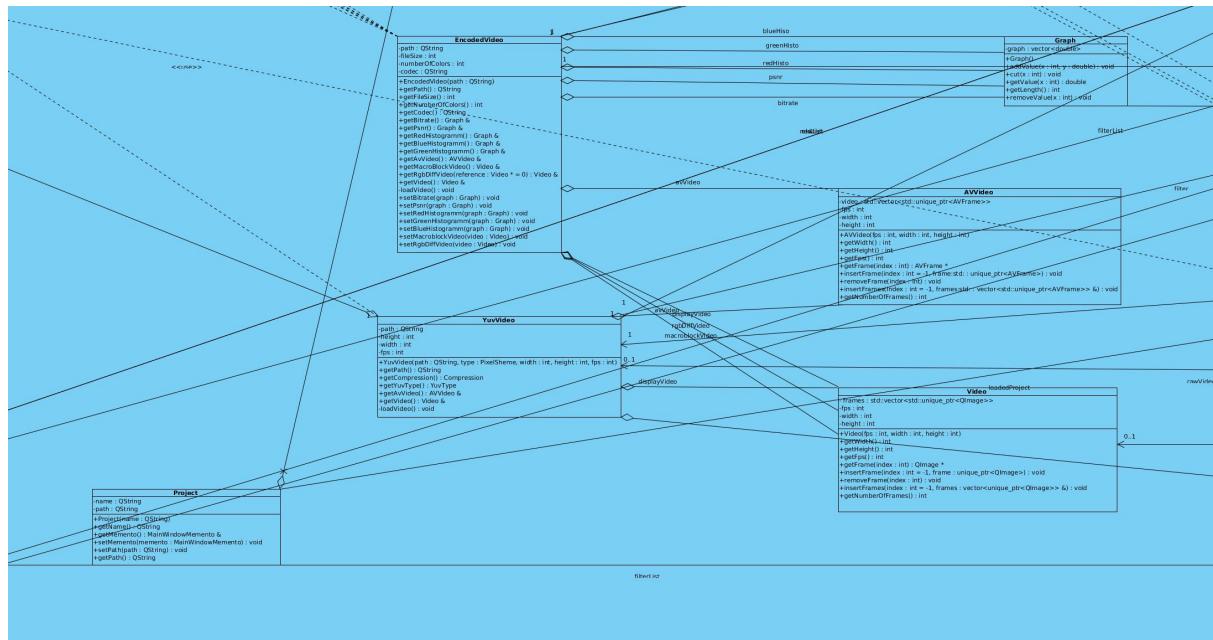
Program structure



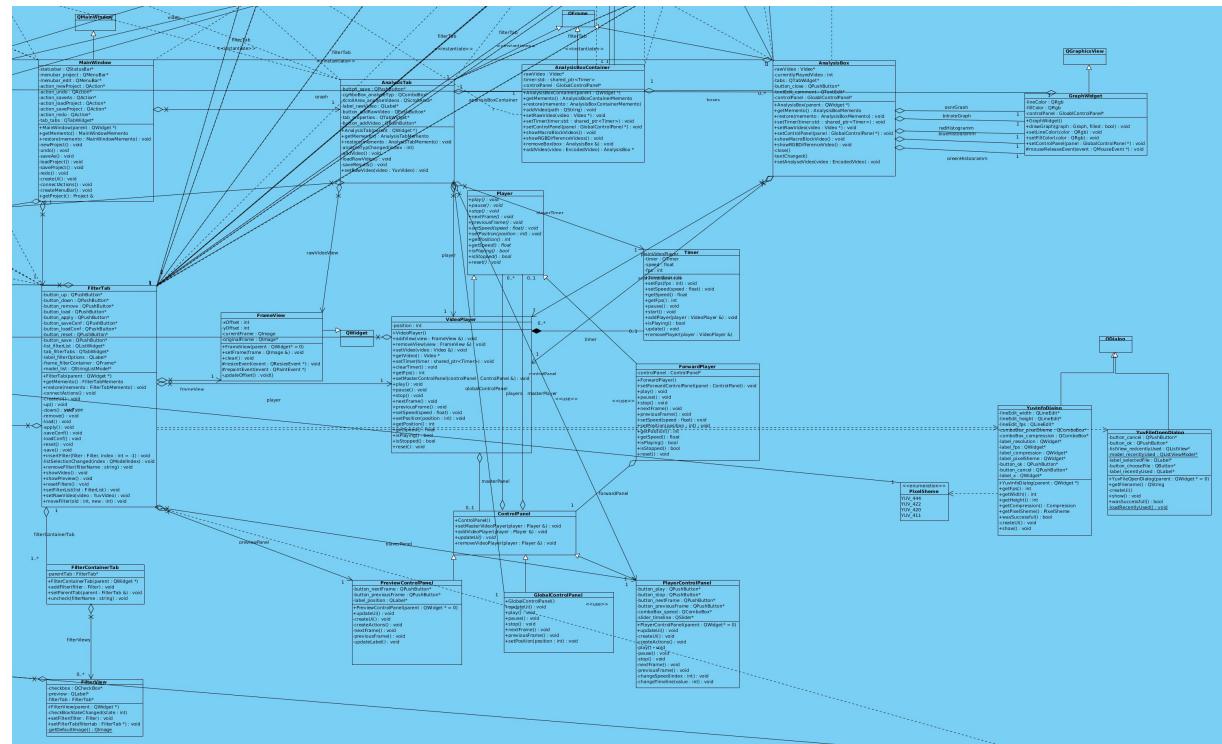
Utility Package



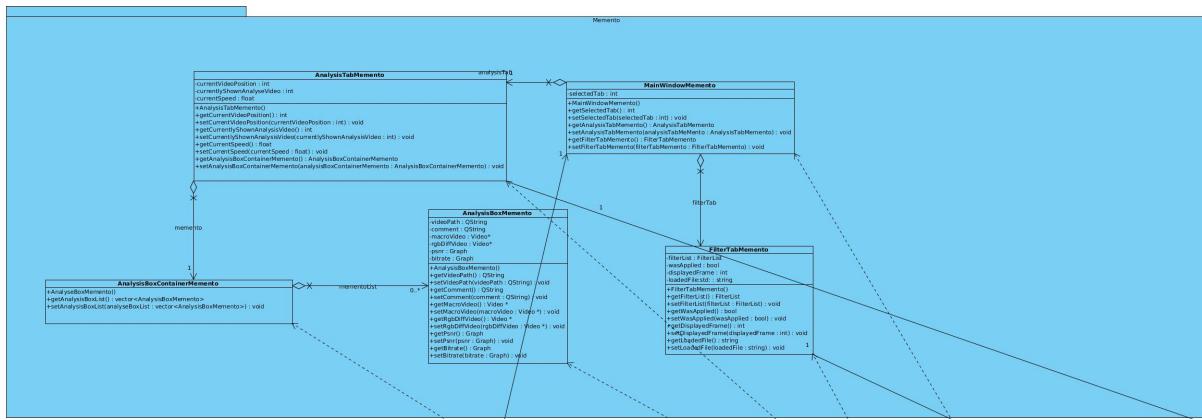
Model without the filters



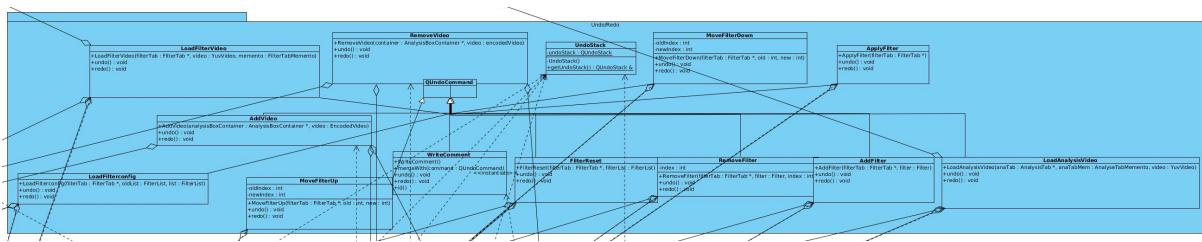
GUI without filterboxes



Memento Package

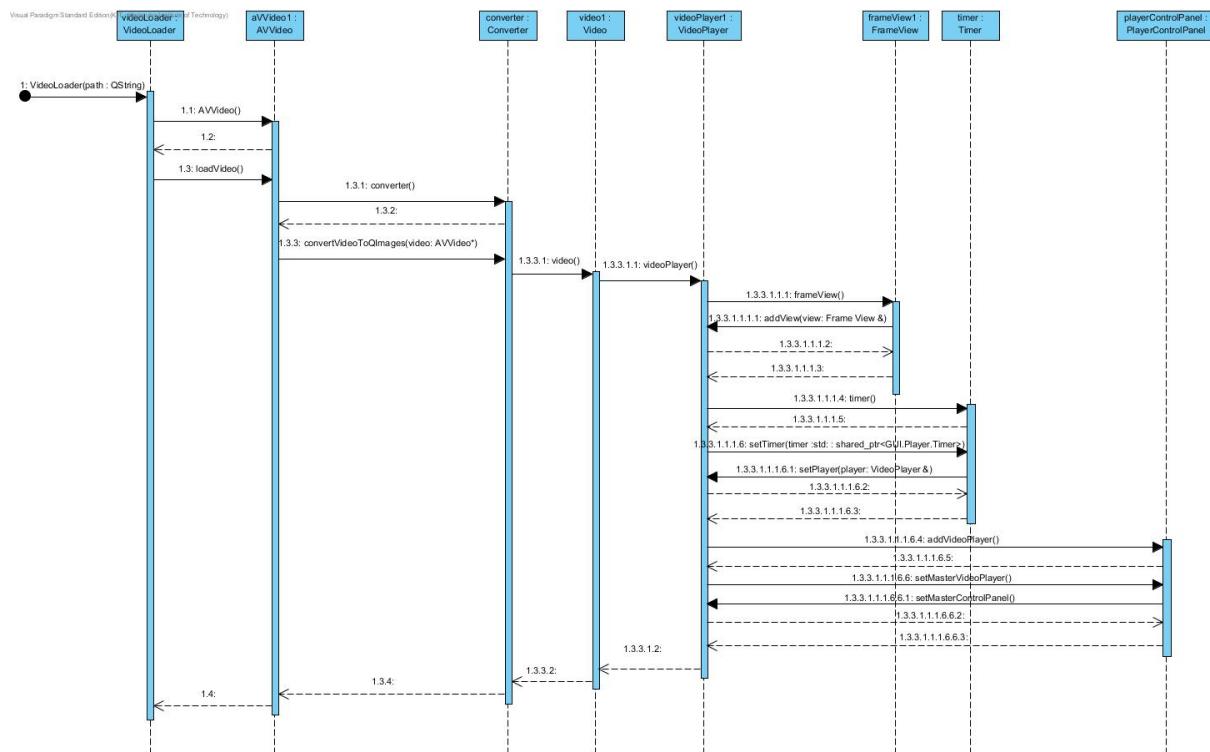


Undo/Redo Package



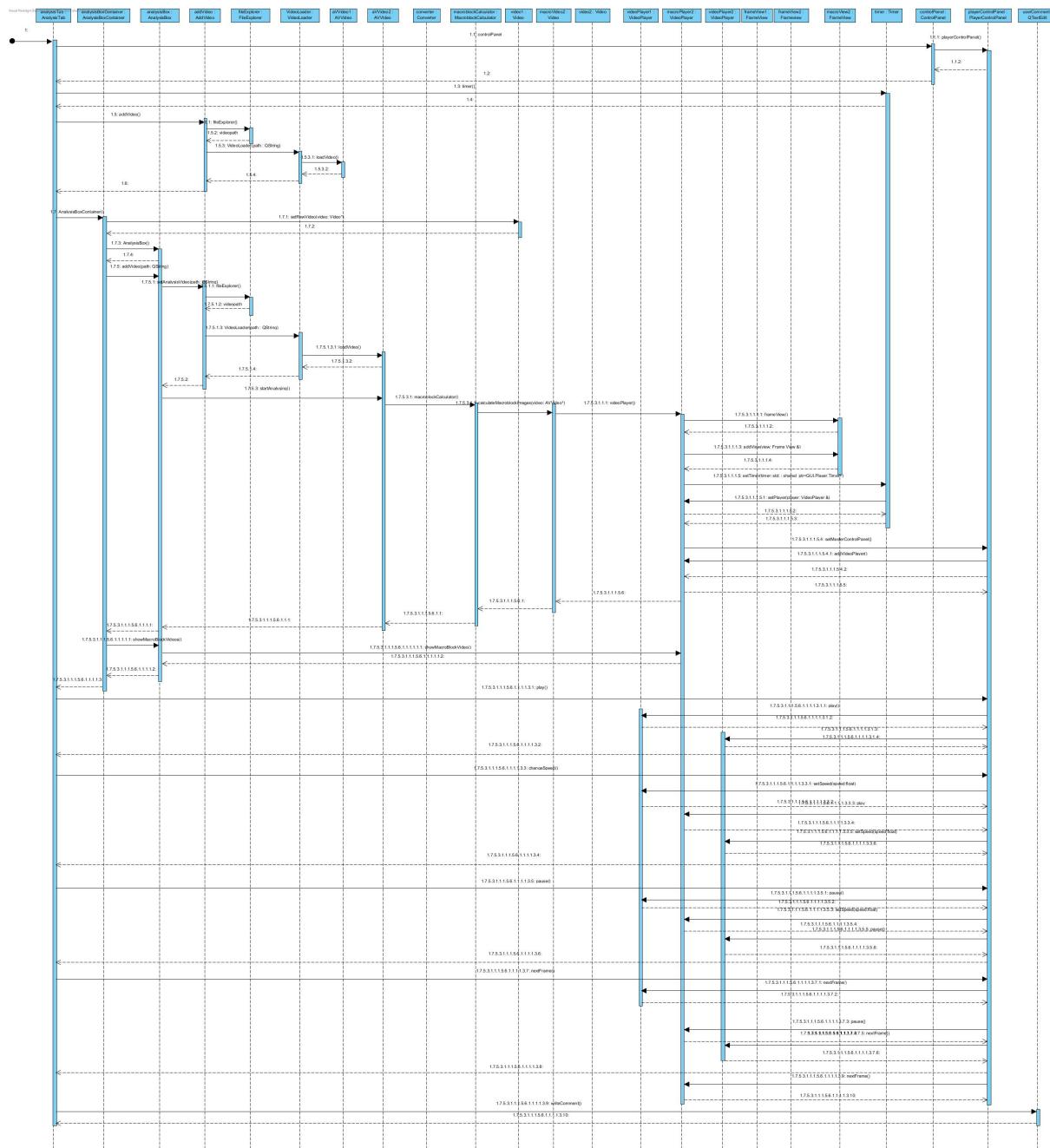
2.2 Sequence Diagrams

LoadVideo sequence diagram



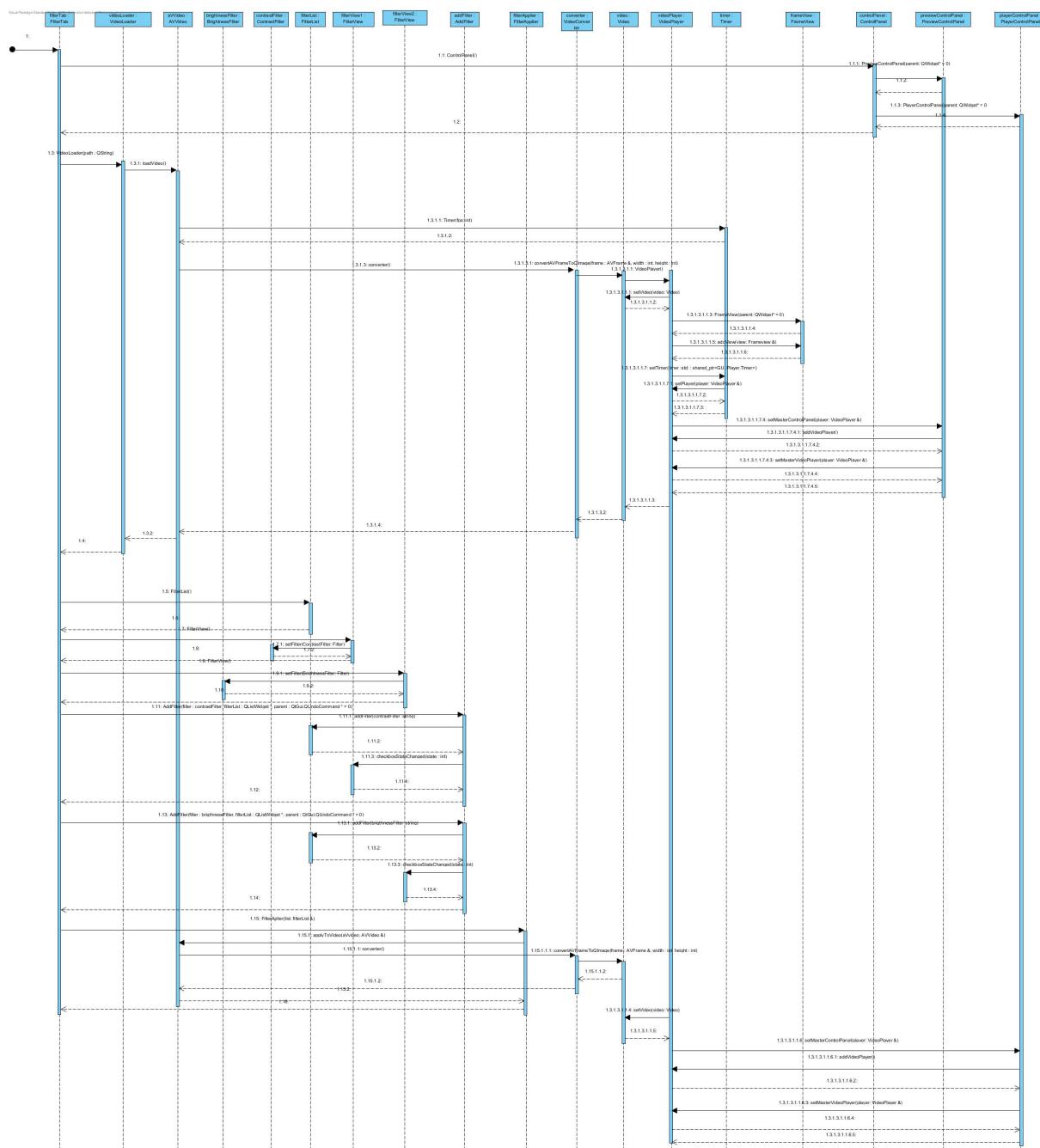
When you load a new Video a LoadVideo object is created with the path to the Video as a parameter. Then a AVVideo is created and with the Converter a Video which is then given to a VideoPlayer. This VideoPlayer has a FrameView a Timer and is then given to a ControlPanel

Analysis sequence diagram



First a ControlPanel and a PlayerControlPanel are created. When a Video is added the LoadVideo Sequenz which is described above is initiated. If an additional Video is added an AnalysisBox is created in which the added Video is loaded and the Analysis starts in which he MacroblockVideo is calculated and Displayed in a second VideoPlayer. When play is pressed in the ControlPanel all Videos start to play.

Filter sequence diagram



First the ControlPanel, PlayerControlPanel and PreviewControlPanel are created. Then a Video is loaded with the LoadVideo sequenz. Then a Filter are added to the FilterList. This FilterList is then applied to the AVVideo which is the converted to an Video and Displayed in the VideoPlayer.

Chapter 3

Namespace Documentation

3.1 GUI Namespace Reference

Data Structures

- class [AnalysisBox](#)
- class [AnalysisBoxContainer](#)
- class [AnalysisTab](#)
- class [BlendingFilterBox](#)
- class [BlurFilterBox](#)
- class [BorderFilterBox](#)
- class [BrightnessFilterBox](#)
- class [ColorbalanceFilterBox](#)
- class [ContrastFilterBox](#)
- class [ControlPanel](#)
- class [FilterConfigurationBox](#)
- class [FilterContainerTab](#)
- class [FilterTab](#)
- class [FilterView](#)
- class [ForwardPlayer](#)
- class [FrameView](#)
- class [GlobalControlPanel](#)
- class [GraphWidget](#)
- class [GridFilterBox](#)
- class [MainWindow](#)
- class [MirrorFilterBox](#)
- class [NoiseFilterBox](#)
- class [PlainFilterBox](#)
- class [Player](#)
- class [PlayerControlPanel](#)
- class [PosterFilterBox](#)
- class [PreviewControlPanel](#)
- class [QCheckBox](#)
- class [QComboBox](#)
- class [QDialog](#)
- class [QFrame](#)
- class [QGraphicsView](#)
- class [QMainWindow](#)
- class [QWidget](#)
- class [RectangleFilterBox](#)

- class [RGBFilterBox](#)
- class [RotationFilterBox](#)
- class [SaturationFilterBox](#)
- class [ScaleFilterBox](#)
- class [SharpnessFilterBox](#)
- class [Timer](#)
- class [Video](#)
- class [VideoPlayer](#)
- class [YuvFileDialog](#)
- class [YuvInfoDialog](#)
- class [ZoomFilterBox](#)

Enumerations

- enum [PixelScheme](#) { [YUV_444](#), [YUV_422](#), [YUV_420](#), [YUV_411](#) }

3.1.1 Enumeration Type Documentation

3.1.1.1 enum PixelScheme

Enumerator

[YUV_444](#)
[YUV_422](#)
[YUV_420](#)
[YUV_411](#)

3.2 Memento Namespace Reference

Data Structures

- class [AnalysisBoxContainerMemento](#)
- class [AnalysisBoxMemento](#)
- class [AnalysisTabMemento](#)
- class [FilterTabMemento](#)
- class [MainWindowMemento](#)

3.3 Model Namespace Reference

Data Structures

- class [AVVideo](#)
- class [BlackWhiteFilter](#)
- class [BlendingFilter](#)
- class [BlurFilter](#)
- class [BorderFilter](#)
- class [BrightnessFilter](#)
- class [ColorbalanceFilter](#)
- class [ContrastFilter](#)
- class [EdgeFilter](#)
- class [EncodedVideo](#)

- class [Filter](#)
- class [FilterApplier](#)
- class [FilterList](#)
- class [Graph](#)
- class [GridFilter](#)
- class [MirrorFilter](#)
- class [NegativeFilter](#)
- class [NoiseFilter](#)
- class [PosterFilter](#)
- class [Project](#)
- class [RectangleFilter](#)
- class [RGBFilter](#)
- class [RotationFilter](#)
- class [SaturationFilter](#)
- class [ScaleFilter](#)
- class [SepiaFilter](#)
- class [SharpnessFilter](#)
- class [VintageFilter](#)
- class [YuvVideo](#)
- class [ZoomFilter](#)

Enumerations

- enum [BasicColor](#) { [RED](#), [GREEN](#), [BLUE](#) }
- enum [MirrorMode](#) { [HORIZONTAL](#), [VERTICAL](#) }
- enum [NoiseMode](#) { [STATIC](#), [RANDOM](#), [MUSTER](#) }

3.3.1 Enumeration Type Documentation

3.3.1.1 enum BasicColor

Enumerator

RED

GREEN

BLUE

3.3.1.2 enum MirrorMode

Enumerator

HORIZONTAL

VERTICAL

3.3.1.3 enum NoiseMode

Enumerator

STATIC

RANDOM

MUSTER

3.4 UndoRedo Namespace Reference

Data Structures

- class [AddFilter](#)
- class [AddVideo](#)
- class [ApplyFilter](#)
- class [FilterReset](#)
- class [LoadAnalysisVideo](#)
- class [LoadFilterconfig](#)
- class [LoadFilterVideo](#)
- class [MoveFilterDown](#)
- class [MoveFilterUp](#)
- class [QUndoCommand](#)
- class [RemoveFilter](#)
- class [RemoveVideo](#)
- class [UndoStack](#)
- class [WriteComment](#)

3.5 Utility Namespace Reference

Data Structures

- class [BitrateCalculator](#)
- class [FilterConfigurationLoader](#)
- class [FilterConfigurationSaver](#)
- class [MacroblockCalculator](#)
- class [ProjectReader](#)
- class [ProjectWriter](#)
- class [PsnrCalculator](#)
- class [RGBDifferenceCalculator](#)
- class [RGBHistogrammCalculator](#)
- class [VideoConverter](#)
- class [VideoLoader](#)
- class [Yuv411FileReader](#)
- class [Yuv411FileSaver](#)
- class [Yuv411Vector](#)
- class [Yuv420FileReader](#)
- class [Yuv420FileSaver](#)
- class [Yuv422FileReader](#)
- class [Yuv422FileSaver](#)
- class [Yuv422Vector](#)
- class [Yuv444FileReader](#)
- class [Yuv444FileSaver](#)
- class [Yuv444Vector](#)
- class [YuvFileReader](#)
- class [YuvFileSaver](#)

Enumerations

- enum [Compression](#) { [PACKED](#), [PLANAR](#) }
- enum [YuvType](#) { [YUV420](#), [YUV444](#), [YUV411](#), [YUV422](#) }

3.5.1 Enumeration Type Documentation

3.5.1.1 enum Compression

Enumerator

PACKED

PLANAR

3.5.1.2 enum YuvType

Enumerator

YUV420

YUV444

YUV411

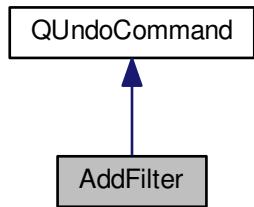
YUV422

Chapter 4

Data Structure Documentation

4.1 AddFilter Class Reference

Inheritance diagram for AddFilter:



Public Member Functions

- `AddFilter (Model::Filter::FilterTab *filterTab, Model::Filter filter)`
- `void undo ()`
- `void redo ()`

Private Attributes

- `GUI::FilterTab * filterTab`
- `Model::Filter * filter`

4.1.1 Detailed Description

This is the undo command for adding a filter to the filterlist on the filtertab.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 `AddFilter (Model::Filter::FilterTab * filterTab, Model::Filter filter)`

Constructor.

Parameters

<code>filterTab</code>	The filtertab to operate on.
<code>filter</code>	The filter on which the operation is performed on.

4.1.3 Member Function Documentation

4.1.3.1 `void redo ()`

Adds a filter to the filterlist.

4.1.3.2 `void undo ()`

Removes the added filter from the filterlist.

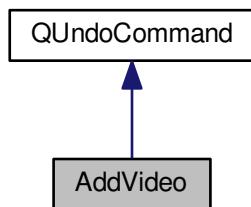
4.1.4 Field Documentation

4.1.4.1 **Model::Filter*** filter [private]

4.1.4.2 **GUI::FilterTab*** filterTab [private]

4.2 AddVideo Class Reference

Inheritance diagram for AddVideo:



Public Member Functions

- [AddVideo \(GUI::AnalysisBoxContainer *analysisBoxContainer, Model::EncodedVideo video\)](#)
- void [undo \(\)](#)
- void [redo \(\)](#)

Private Attributes

- GUI::AnalysisBox * [anaBox](#)
- GUI::AnalysisBoxContainer * [anaBoxContainer](#)
- Model::EncodedVideo * [video](#)

4.2.1 Detailed Description

This class is the undo command for adding a encoded video on the analysis tab.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 AddVideo (GUI::AnalysisBoxContainer * *analysisBoxContainer*, Model::EncodedVideo *video*)

Constructor.

Parameters

<i>analysisBoxContainer</i>	The AnalysisBoxContainer to operate on.
<i>video</i>	The video on which the action is performed.

4.2.3 Member Function Documentation

4.2.3.1 void redo ()

Adds a video to the Analysis tab.

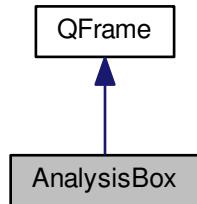
4.2.3.2 void undo ()

Removes the added video from the analysis tab.

4.2.4 Field Documentation**4.2.4.1 GUI::AnalysisBox* anaBox [private]****4.2.4.2 GUI::AnalysisBoxContainer* anaBoxContainer [private]****4.2.4.3 Model::EncodedVideo* video [private]**

4.3 AnalysisBox Class Reference

Inheritance diagram for AnalysisBox:



Public Member Functions

- `AnalysisBox (GUI::QWidget *parent)`
- `Memento::AnalysisBoxMemento getMemento ()`
- `void restore (Memento::AnalysisBoxMemento memento)`
- `void setTimer (shared_ptr< GUI::Timer > timer:std::)`
- `void setRawVideo (GUI::Video *video)`
- `void setControlPanel (Player::GlobalControlPanel *panel)`
- `void showMacroBlockVideo ()`
- `void showRGBDifferenceVideo ()`
- `void setAnalyseVideo (Model::EncodedVideo video)`

Data Fields

- `UndoRedo::RemoveVideo * anaBox`

Private Member Functions

- `void close ()`
- `void textChanged ()`

Private Attributes

- `GUI::Video * rawVideo`
- `int currentlyPlayedVideo`
- `QTabWidget * tabs`
- `QPushButton * button_close`
- `QTextEdit * textEdit_comment`
- `GloablControlPanel * controlPanel`
- `GUI::AnalysisBoxContainer * boxes`
- `GUI::VideoPlayer * plainVideoPlayer`
- `GUI::VideoPlayer * analysisVideoPlayer`
- `GUI::GraphWidget * psnrGraph`
- `GUI::GraphWidget * bitrateGraph`

- `GUI::GraphWidget * redHistogramm`
- `GUI::GraphWidget * blueHistogramm`
- `GUI::GraphWidget * greenHistogramm`
- `Model::EncodedVideo * video`

4.3.1 Detailed Description

Shows the Analysis of a single encoded video.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 `AnalysisBox (GUI::QWidget * parent)`

4.3.3 Member Function Documentation

4.3.3.1 `void close () [private]`

Slot: connected to `button_close.pressed()` removes this box from `AnalysisBoxContainer`

4.3.3.2 `Memento::AnalysisBoxMemento getMemento ()`

Creates a memento which contains the state of the box.

Returns

The created memento.

4.3.3.3 `void restore (Memento::AnalysisBoxMemento memento)`

Restores the box based on the memento.

Parameters

<code>memento</code>	The memento which contains the state of the box.
----------------------	--

4.3.3.4 `void setAnalyseVideo (Model::EncodedVideo video)`

Sets the video this box shall present.

Parameters

<code>video</code>	The video to present.
--------------------	-----------------------

4.3.3.5 `void setControlPanel (Player::GlobalControlPanel * panel)`

Sets the `GlobalControlPanel`.

Parameters

<code>panel</code>	The <code>GlobalControlPanel</code> .
--------------------	---------------------------------------

4.3.3.6 `void setRawVideo (GUI::Video * video)`

Sets the rawvideo the encoded video is compared to.

Parameters

<i>video</i>	The rawvideo.
--------------	---------------

4.3.3.7 void setTimer (shared_ptr< GUI::Timer > timer:std:)

Sets the timer for the videoplayer.

Parameters

<i>timer:std,::</i>	The timer for the videoplayer.
---------------------	--------------------------------

4.3.3.8 void showMacroBlockVideo ()

Shows the macroblock video. The rgb difference video is no longer shown.

4.3.3.9 void showRGBDifferenceVideo ()

Shows the rgb difference video. The macroblock video is no longer shown.

4.3.3.10 void textChanged () [private]

Slot: connected with `textEdit_comment` updates `QUndoStack`

4.3.4 Field Documentation

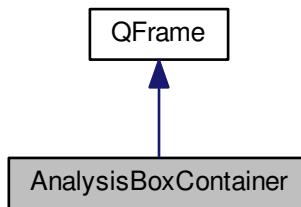
4.3.4.1 UndoRedo::RemoveVideo* anaBox**4.3.4.2 GUI::VideoPlayer* analysisVideoPlayer [private]****4.3.4.3 GUI::GraphWidget* bitrateGraph [private]****4.3.4.4 GUI::GraphWidget* blueHistogramm [private]****4.3.4.5 GUI::AnalysisBoxContainer* boxes [private]****4.3.4.6 QPushButton* button_close [private]****4.3.4.7 GlobalControlPanel* controlPanel [private]****4.3.4.8 int currentlyPlayedVideo [private]****4.3.4.9 GUI::GraphWidget* greenHistogramm [private]****4.3.4.10 GUI::VideoPlayer* plainVideoPlayer [private]****4.3.4.11 GUI::GraphWidget* psnrGraph [private]****4.3.4.12 GUI::Video* rawVideo [private]****4.3.4.13 GUI::GraphWidget* redHistogramm [private]****4.3.4.14 QTabWidget* tabs [private]**

4.3.4.15 `QTextEdit* textEdit_comment` [private]

4.3.4.16 `Model::EncodedVideo* video` [private]

4.4 AnalysisBoxContainer Class Reference

Inheritance diagram for AnalysisBoxContainer:



Public Member Functions

- `AnalysisBoxContainer (GUI::QWidget *parent)`
- `Memento::AnalysisBoxContainerMemento getMemento ()`
- `void restore (Memento::AnalysisBoxContainerMemento memento)`
- `void addVideo (QString path)`
- `void setRawVideo (GUI::Video *video)`
- `void setTimer (shared_ptr< GUI::Timer > timer:std::shared_ptr< GUI::Timer >)`
- `void setControlPanel (Player::GlobalControlPanel *panel)`
- `void showMacroBlockVideos ()`
- `void showRGBDifferenceVideos ()`
- `void removeBox (GUI::AnalysisBox &box)`
- `GUI::AnalysisBox * addVideo (Model::EncodedVideo video)`

Data Fields

- `UndoRedo::RemoveVideo * anaBoxContainer`

Private Attributes

- `GUI::Video * rawVideo`
- `shared_ptr< GUI::Timer > timer:std::shared_ptr< GUI::Timer >`
- `Player::GlobalControlPanel * controlPanel`
- `GUI::AnalysisTab * analysisBoxContainer`
- `std::vector< GUI::AnalysisBox * > boxes`

4.4.1 Detailed Description

Contains and manages the AnalysisBoxes.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 AnalysisBoxContainer (`GUI::QWidget * parent`)

Constructor.

4.4.3 Member Function Documentation

4.4.3.1 void addVideo (QString *path*)

Creates a Analysis box and shows it.

Parameters

<i>path</i>	The path of the video to analyse.
-------------	-----------------------------------

4.4.3.2 GUI::AnalysisBox* addVideo (Model::EncodedVideo *video*)

Adds the given video to the container.

Parameters

<i>video</i>	The video to add.
--------------	-------------------

Returns

The box in which the video is presented.

4.4.3.3 Memento::AnalysisBoxContainerMemento getMemento ()

Creates a memento which contains the state of the container.

Returns

The created memento.

4.4.3.4 void removeBox (GUI::AnalysisBox & *box*)

Removes a box from the list.

Parameters

<i>box</i>	The box to remove.
------------	--------------------

4.4.3.5 void restore (Memento::AnalysisBoxContainerMemento *memento*)

Restores the container based on the memento.

Parameters

<i>memento</i>	The memento which contains the state to restore.
----------------	--

4.4.3.6 void setControlPanel (Player::GlobalControlPanel * *panel*)

Sets the [GlobalControlPanel](#).

Parameters

<i>panel</i>	The panel.
--------------	------------

4.4.3.7 void setRawVideo (**GUI::Video** * *video*)

Sets the rawVideo the encoded videos are compared to.

Parameters

<i>video</i>	The raw video.
--------------	----------------

4.4.3.8 void setTimer (**shared_ptr<GUI::Timer>** *timer:std::*)

Sets the timer for the videoplayers.

Parameters

<i>timer:std::</i>	The timer for the videoplayers.
--------------------	---------------------------------

4.4.3.9 void showMacroBlockVideos ()

Tells all AnalysisBoxes to show the macro block video.

4.4.3.10 void showRGBDifferenceVideos ()

Tells all AnalysisBoxes to show the RGBDiff video.

4.4.4 Field Documentation

4.4.4.1 **UndoRedo::RemoveVideo*** *anaBoxContainer*

4.4.4.2 **GUI::AnalysisTab*** *analysisBoxContainer* [private]

4.4.4.3 **std::vector<GUI::AnalysisBox*>** *boxes* [private]

4.4.4.4 **Player::GlobalControlPanel*** *controlPanel* [private]

4.4.4.5 **GUI::Video*** *rawVideo* [private]

4.4.4.6 **shared_ptr<GUI::Timer>** *timer* [private]

4.5 AnalysisBoxContainerMemento Class Reference

Public Member Functions

- void [analyseBoxMemento \(\)](#)
- [vector< Memento::AnalysisBoxMemento > getAnalysisBoxList \(\)](#)
- void [setAnalysisBoxList \(vector< Memento::AnalysisBoxMemento > analyseBoxList\)](#)

Private Attributes

- std::vector< [Memento::AnalysisBoxMemento](#) * > [mementoList](#)

4.5.1 Detailed Description

This class is the memento for the AnalysisBoxContainer.

4.5.2 Member Function Documentation

4.5.2.1 void analyseBoxMemento ()

Constructor.

4.5.2.2 [vector< Memento::AnalysisBoxMemento > getAnalysisBoxList \(\)](#)

Returns a list of AnalysisBox mementos.

Returns

The list of [AnalysisBoxMemento](#).

4.5.2.3 void setAnalysisBoxList ([vector< Memento::AnalysisBoxMemento > analyseBoxList](#))

Sets the list of [AnalysisBoxMemento](#)

Parameters

analyseBoxList	The list of the mementos.
--------------------------------	---------------------------

4.5.3 Field Documentation

4.5.3.1 std::vector<[Memento::AnalysisBoxMemento](#)*> [mementoList](#) [private]

4.6 AnalysisBoxMemento Class Reference

Public Member Functions

- `AnalysisBoxMemento ()`
- `QString getVideoPath ()`
- `void setVideoPath (QString videoPath)`
- `QString getComment ()`
- `void setComment (QString comment)`
- `GUI::Video * getMacroVideo ()`
- `void setMacroVideo (GUI::Video *macroVideo)`
- `GUI::Video * getRgbDiffVideo ()`
- `void setRgbDiffVideo (GUI::Video *rgbDiffVideo)`
- `Model::Graph getPsnr ()`
- `void setPsnr (Model::Graph psnr)`
- `Model::Graph getBitrate ()`
- `void setBitrate (Model::Graph bitrate)`

Private Attributes

- `QString videoPath`
- `QString comment`
- `GUI::Video * macroVideo`
- `GUI::Video * rgbDiffVideo`
- `Model::Graph psnr`
- `Model::Graph bitrate`

4.6.1 Detailed Description

This class is the memento for the AnalysisBox.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 AnalysisBoxMemento ()

Constructor.

4.6.3 Member Function Documentation

4.6.3.1 Model::Graph getBitrate ()

Returns the bitrate graph.

Returns

The bitrate graph.

4.6.3.2 QString getComment ()

Returns the user comment.

Returns

The user comment.

4.6.3.3 GUI::Video * getMacroVideo ()

Returns the macroblock video.

Returns

The macroblock video.

4.6.3.4 Model::Graph getPsnr ()

Returns the psnr graph.

Returns

The psnr graph.

4.6.3.5 GUI::Video * getRgbDiffVideo ()

Returns the rgb difference video.

Returns

The rgb difference video.

4.6.3.6 QString getVideoPath ()

Returns the path to the video.

Returns

Absolute path to the video.

4.6.3.7 void setBitrate (Model::Graph *bitrate*)

Sets the bitrate graph.

Parameters

<i>bitrate</i>	The bitrate gaph.
----------------	-------------------

4.6.3.8 void setComment (QString *comment*)

Sets the user comment.

Parameters

<i>comment</i>	The user comment.
----------------	-------------------

4.6.3.9 void setMacroVideo (GUI::Video * *macroVideo*)

Sets the macroblock video.

Parameters

<i>macroVideo</i>	The macroblock video.
-------------------	-----------------------

4.6.3.10 void setPsnr (Model::Graph *psnr*)

Sets the the psnr graph.

Parameters

<i>psnr</i>	The psnr graph.
-------------	-----------------

4.6.3.11 void setRgbDiffVideo (GUI::Video * *rgbDiffVideo*)

Sets the rgb difference video.

Parameters

<i>rgbDiffVideo</i>	The rgb difference video.
---------------------	---------------------------

4.6.3.12 void setVideoPath (QString *videoPath*)

Sets the path to the video.

Parameters

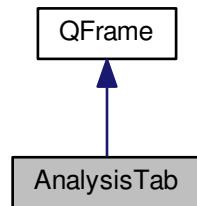
<i>videoPath</i>	Absolute path to the video.
------------------	-----------------------------

4.6.4 Field Documentation

4.6.4.1 Model::Graph *bitrate* [private]**4.6.4.2 QString *comment* [private]****4.6.4.3 GUI::Video* *macroVideo* [private]****4.6.4.4 Model::Graph *psnr* [private]****4.6.4.5 GUI::Video* *rgbDiffVideo* [private]****4.6.4.6 QString *videoPath* [private]**

4.7 AnalysisTab Class Reference

Inheritance diagram for AnalysisTab:



Public Member Functions

- `AnalysisTab (GUI::QWidget *parent)`
- `Memento::AnalysisTabMemento getMemento ()`
- `void restore (Memento::AnalysisTabMemento memento)`
- `void setRawVideo (Model::YuvVideo video)`

Data Fields

- `UndoRedo::LoadAnalysisVideo * anaTab`
- `GUI::FrameView * rawVideoView`

Private Member Functions

- `void analyseTypChanged (int index)`
- `void addVideo ()`
- `void loadRawVideo ()`
- `void saveResults ()`

Private Attributes

- `QPushButton * button_save`
- `GUI::QComboBox * comboBox_analyseTyp`
- `QScrollArea * scrollArea_analyseVideos`
- `QLabel * label_rawVideo`
- `QPushButton * button_addRawVideo`
- `QTabWidget * tab_properties`
- `QPushButton * button_addVideo`
- `GUI::VideoPlayer * player`
- `GUI::PlayerControlPanel * controlPanel`
- `GUI::Timer * playerTimer`
- `GUI::AnalysisBoxContainer * analysisBoxContainer`
- `Model::YuvVideo * rawVideo`
- `GUI::GlobalControlPanel * globalControlPanel`

4.7.1 Detailed Description

The tab that shows videos and analyses them.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 AnalysisTab (*GUI::QWidget * parent*)

Constructor.

4.7.3 Member Function Documentation

4.7.3.1 void addVideo () [private]

Slot: connected to button_addVideo.pressed() opens QFileDialog, and gives string to [AnalysisBoxContainer](#)

4.7.3.2 void analyseTypChanged (int *index*) [private]

Slot: connected to comboBox_analyseTyp.currentIndexChanged(index : int) tells [AnalysisBox](#) to showMacroBlocks() or showRGBDiff()

Parameters

<i>index</i>	The new type.
--------------	---------------

4.7.3.3 Memento::AnalysisTabMemento getMemento ()

Creates a memento which contains the state of this tab.

Returns

The created memento.

4.7.3.4 void loadRawVideo () [private]

Slot: connected to button_addRawVideo.pressed() opens [YuvFileDialog](#) and gives the video to [AnalysisBox](#) Container

4.7.3.5 void restore (Memento::AnalysisTabMemento *memento*)

Restores the tab based on the memento.

Parameters

<i>memento</i>	The memento which contains the state of the tab.
----------------	--

4.7.3.6 void saveResults () [private]

Opens QFileDialog and saves relevant data in the path.

4.7.3.7 void setRawVideo (Model::YuvVideo *video*)

Sets the raw video for the analysis. This operation resets the tab completely.

Parameters

<i>video</i>	The new raw video.
--------------	--------------------

4.7.4 Field Documentation**4.7.4.1 GUI::AnalysisBoxContainer* analysisBoxContainer [private]****4.7.4.2 UndoRedo::LoadAnalysisVideo* anaTab****4.7.4.3 QPushButton* button_addRawVideo [private]****4.7.4.4 QPushButton* button_addVideo [private]****4.7.4.5 QPushButton* button_save [private]****4.7.4.6 GUI::QComboBox* comboBox_analyseTyp [private]****4.7.4.7 GUI::PlayerControlPanel* controlPanel [private]****4.7.4.8 GUI::GlobalControlPanel* globalControlPanel [private]****4.7.4.9 QLabel* label_rawVideo [private]****4.7.4.10 GUI::VideoPlayer* player [private]****4.7.4.11 GUI::Timer* playerTimer [private]****4.7.4.12 Model::YuvVideo* rawVideo [private]****4.7.4.13 GUI::FrameView* rawVideoView****4.7.4.14 QScrollArea* scrollArea_analyseVideos [private]****4.7.4.15 QTabWidget* tab_properties [private]**

4.8 AnalysisTabMemento Class Reference

Public Member Functions

- `AnalysisTabMemento ()`
- `int getCurrentVideoPosition ()`
- `void setCurrentVideoPosition (int currentVideoPosition)`
- `int getCurrentlyShownAnalysisVideo ()`
- `void setCurrentlyShownAnalysisVideo (int currentlyShownAnalysisVideo)`
- `float getCurrentSpeed ()`
- `void setCurrentSpeed (float currentSpeed)`
- `Memento::AnalysisBoxContainerMemento getAnalysisBoxContainerMemento ()`
- `void setAnalysisBoxContainerMemento (Memento::AnalysisBoxContainerMemento analysisBoxContainerMemento)`

Data Fields

- `UndoRedo::LoadAnalysisVideo * memento`

Private Attributes

- `int currentVideoPosition`
- `int currentlyShownAnalyseVideo`
- `float currentSpeed`

4.8.1 Detailed Description

This class is the memento for the analysis tab.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 AnalysisTabMemento ()

Constructor.

4.8.3 Member Function Documentation

4.8.3.1 Memento::AnalysisBoxContainerMemento getAnalysisBoxContainerMemento ()

Returns the memento of the AnalysisBoxContainer.

Returns

The memento of the AnalysisBoxContainer.

4.8.3.2 int getCurrentlyShownAnalysisVideo ()

Returns what analysis video is currently shown. 0 means rgb difference. non zero means macroblocks.

Returns

The currently shown analysis video.

4.8.3.3 float getCurrentSpeed ()

Returns the current speed of the player.

Returns

The current speed of the player.

4.8.3.4 int getCurrentVideoPosition ()

Returns the current position the player is at.

Returns

The current position of the player.

4.8.3.5 void setAnalysisBoxContainerMemento (Memento::AnalysisBoxContainerMemento analysisBoxContainerMemento)

Sets the memento of the AnalysisBoxContainer.

Parameters

<i>analysisBox</i> <i>Container</i> <i>Memento</i>	The memento of the AnalysisBoxContainer.
--	--

4.8.3.6 void setCurrentlyShownAnalysisVideo (int currentlyShownAnalysisVideo)

Sets the currently shown analysis video. 0 means rgb difference. non 0 means macroblocks.

Parameters

<i>currently</i> <i>Shown</i> <i>AnalysisVideo</i>	The currrently shown analysis video.
--	--------------------------------------

4.8.3.7 void setCurrentSpeed (float currentSpeed)

Sets the current speed of the player.

Parameters

<i>currentSpeed</i>	The current speed of the player.
---------------------	----------------------------------

4.8.3.8 void setCurrentVideoPosition (int currentVideoPosition)

Sets the current position of the player.

Parameters

<i>currentVideoPosition</i>	The current position of the player.
-----------------------------	-------------------------------------

4.8.4 Field Documentation

4.8.4.1 int currentlyShownAnalyseVideo [private]

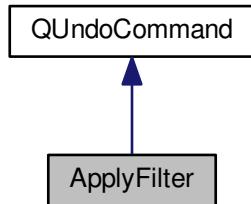
4.8.4.2 float currentSpeed [private]

4.8.4.3 int currentVideoPosition [private]

4.8.4.4 UndoRedo::LoadAnalysisVideo* memento

4.9 ApplyFilter Class Reference

Inheritance diagram for ApplyFilter:



Public Member Functions

- [ApplyFilter \(GUI::FilterTab *filterTab\)](#)
- void [undo \(\)](#)
- void [redo \(\)](#)

Private Attributes

- [GUI::FilterTab * filterTab](#)

4.9.1 Detailed Description

This class is the undo command for applying filters on the filtertab.

4.9.2 Constructor & Destructor Documentation

4.9.2.1 [ApplyFilter \(GUI::FilterTab * filterTab \)](#)

Constructor

4.9.3 Member Function Documentation

4.9.3.1 [void redo \(\)](#)

Shows the complete video with applied filters.

4.9.3.2 [void undo \(\)](#)

Shows the 5 frame preview.

4.9.4 Field Documentation

4.9.4.1 **GUI::FilterTab*** filterTab [private]

4.10 AVVideo Class Reference

Public Member Functions

- `AVVideo (int fps, int width, int height)`
- `int getWidth ()`
- `int getHeight ()`
- `int getFps ()`
- `AVFrame * getFrame (int index)`
- `void insertFrame (int index=-1, unique_ptr< AVFrame > frame:std::unique_ptr< AVFrame > frame)`
- `void removeFrame (int index)`
- `void insertFrames (int index=-1, vector< std::unique_ptr< AVFrame > > &frames:std::vector< std::unique_ptr< AVFrame > > frames)`
- `int getNumberOfFrames ()`

Data Fields

- `Model::EncodedVideo * avVideo`

Private Attributes

- `std::vector< std::unique_ptr< AVFrame > > video`
- `int fps`
- `int width`
- `int height`

4.10.1 Detailed Description

This class contains AVFrames from the ffmpeg library and manages them.

4.10.2 Constructor & Destructor Documentation

4.10.2.1 `AVVideo (int fps, int width, int height)`

Constructor.

Parameters

<code>fps</code>	The fps the video should be played at.
<code>width</code>	The width of the video.
<code>height</code>	The height of the video.

4.10.3 Member Function Documentation

4.10.3.1 `int getFps ()`

Returns the fps of the video.

Returns

Fps of the video.

4.10.3.2 `AVFrame * getFrame (int index)`

Returns the frame at the given index. If the index is invalid nullptr is returned.

Parameters

<i>index</i>	the index of the frame to return
--------------	----------------------------------

Returns

The frame at the given index.

4.10.3.3 int getHeight()

Returns the height of the video.

Returns

The height of the video.

4.10.3.4 int getNumberOfFrames()

Returns the number of frames in the video.

Returns

The number of frames in the video.

4.10.3.5 int getWidth()

Returns the width of the video.

Returns

The width of the video.

4.10.3.6 void insertFrame(int *index* = -1, unique_ptr<AVFrame> *frame*:std::)

Inserts a frame at the given index. If *index* < 0 then the frame gets pushed to the back. If the index is greater than [getNumberOfFrames\(\)](#) the frames gets pushed to the back.

Parameters

<i>index</i>	The index to insert the frame at.
<i>frame</i> :std:::	The frame to insert.

4.10.3.7 void insertFrames(int *index* = -1, vector<unique_ptr<AVFrame>> &*frames*:std::)

Inserts a vector of frames at the given index. If the *index*<0 or *index* is greater than [getNumberOfFrames\(\)](#) then the frames are pushed to the back.

Parameters

<i>index</i>	The index to insert the frames at.
--------------	------------------------------------

<i>frames:std::vector<AVFrame*></i>	The frames to insert.
---	-----------------------

4.10.3.8 void removeFrame (int *index*)

Removes the frame at the given index. If the index is invalid nothing happens.

Parameters

<i>index</i>	The index of the frame to remove.
--------------	-----------------------------------

4.10.4 Field Documentation

4.10.4.1 Model::EncodedVideo* avVideo

4.10.4.2 int fps [private]

4.10.4.3 int height [private]

4.10.4.4 std::vector<std::unique_ptr<AVFrame>> video [private]

4.10.4.5 int width [private]

4.11 BitrateCalculator Class Reference

Public Member Functions

- BitrateCalculator (Model::AVVideo &video)
- Model::Graph calculate ()

Private Attributes

- Model::AVVideo * video

4.11.1 Detailed Description

This class calculates the bitrate of a video.

4.11.2 Constructor & Destructor Documentation

4.11.2.1 BitrateCalculator (Model::AVVideo & video)

Constructor.

Parameters

video	The video of which the bitrate is calculated.
-------	---

4.11.3 Member Function Documentation

4.11.3.1 Model::Graph calculate ()

Calculates the bitrate graph.

Returns

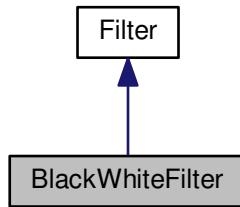
The calculated bitrate graph.

4.11.4 Field Documentation

4.11.4.1 Model::AVVideo* video [private]

4.12 BlackWhiteFilter Class Reference

Inheritance diagram for BlackWhiteFilter:



Public Member Functions

- [BlackWhiteFilter \(\)](#)
- string [getName \(\)](#)
- string [getFilterDescription \(\)](#)

Additional Inherited Members

4.12.1 Detailed Description

Converts the video to a black and white video.

4.12.2 Constructor & Destructor Documentation

4.12.2.1 [BlackWhiteFilter\(\)](#)

Constructor.

4.12.3 Member Function Documentation

4.12.3.1 [string getFilterDescription\(\) \[virtual\]](#)

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implements [Filter](#).

4.12.3.2 [string getName\(\) \[virtual\]](#)

Returns the name of the filter.

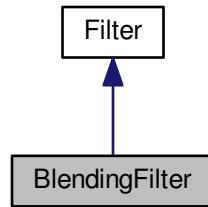
Returns

The filtername.

Implements [Filter](#).

4.13 BlendingFilter Class Reference

Inheritance diagram for BlendingFilter:



Public Member Functions

- `BlendingFilter ()`
- `bool getInBlend ()`
- `void setInBlend (bool inBlend)`
- `int getStartFrame ()`
- `void setStartFrame (int startFrame)`
- `int getEndFrame ()`
- `void setEndFrame (int endFrame)`
- `string getName ()`
- `string getFilterDescription ()`

Private Attributes

- `bool inBlend`
- `int startFrame`
- `int endFrame`

Additional Inherited Members

4.13.1 Detailed Description

Inserts black blending into the video

4.13.2 Constructor & Destructor Documentation

4.13.2.1 `BlendingFilter()`

Constructor.

4.13.3 Member Function Documentation

4.13.3.1 int getEndFrame()

Returns the end frame of the blending.

Returns

The end frame.

4.13.3.2 string getFilterDescription() [virtual]

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implements [Filter](#).

4.13.3.3 bool getInBlend()

Whether it is an in blending.

Returns

true if it is an in blending.

4.13.3.4 string getName() [virtual]

Returns the name of the filter.

Returns

The filtername.

Implements [Filter](#).

4.13.3.5 int getStartFrame()

Returns the start frame of the blending.

Returns

The start frame of the blending.

4.13.3.6 void setEndFrame(int *endFrame*)

Sets the end frame of the blending.

Parameters

<i>endFrame</i>	The end frame.
-----------------	----------------

4.13.3.7 void setInBlend (bool *inBlend*)

Sets whether it is an in blending.

Parameters

<i>inBlend</i>	True if it is an in blending.
----------------	-------------------------------

4.13.3.8 void setStartFrame (int *startFrame*)

Sets the start frame of the blending.

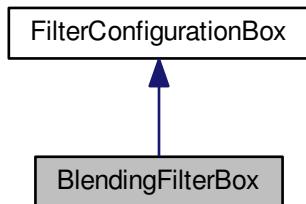
Parameters

<i>startFrame</i>	The start frame.
-------------------	------------------

4.13.4 Field Documentation**4.13.4.1 int endFrame [private]****4.13.4.2 bool inBlend [private]****4.13.4.3 int startFrame [private]**

4.14 BlendingFilterBox Class Reference

Inheritance diagram for BlendingFilterBox:



Public Member Functions

- [BlendingFilterBox \(GUI::QWidget *parent \)](#)

Additional Inherited Members

4.14.1 Detailed Description

This class contains the gui elements for changing the options of a blending filter.

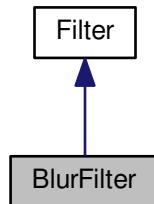
4.14.2 Constructor & Destructor Documentation

4.14.2.1 BlendingFilterBox (GUI::QWidget * *parent*)

Constructor.

4.15 BlurFilter Class Reference

Inheritance diagram for BlurFilter:



Public Member Functions

- `BlurFilter ()`
- `bool getPreserveEdges ()`
- `void setPreserveEdges (bool preserveEdges)`
- `int getIntensity ()`
- `void setIntensity (int intensity)`
- `string getFilterDescription ()`
- `string getName ()`

Private Attributes

- `bool preserveEdges`
- `int intensity`

Additional Inherited Members

4.15.1 Detailed Description

Blurs the video.

4.15.2 Constructor & Destructor Documentation

4.15.2.1 `BlurFilter()`

Constructor.

4.15.3 Member Function Documentation

4.15.3.1 `string getFilterDescription() [virtual]`

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implements [Filter](#).

4.15.3.2 int getIntensity ()

Returns the intensity of the blurring.

Returns

The intensity of the blurring.

4.15.3.3 string getName () [virtual]

Returns the name of the filter.

Returns

The filtername.

Implements [Filter](#).

4.15.3.4 bool getPreserveEdges ()

Whether edges shall be preserved when blurring.

Returns

true if the edges are preserved.

4.15.3.5 void setIntensity (int *intensity*)

Sets the intensity of the blurring.

Parameters

<i>intensity</i>	The intensity of the blurring.
------------------	--------------------------------

4.15.3.6 void setPreserveEdges (bool *preserveEdges*)

Sets whether the edges shall be preserved when blurring.

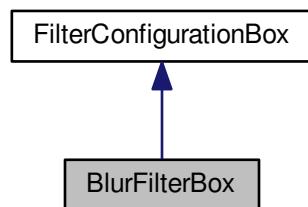
Parameters

<i>preserveEdges</i>	True if the edges shall be preserved.
----------------------	---------------------------------------

4.15.4 Field Documentation**4.15.4.1 int intensity [private]****4.15.4.2 bool preserveEdges [private]**

4.16 BlurFilterBox Class Reference

Inheritance diagram for BlurFilterBox:



Public Member Functions

- [BlurFilterBox \(GUI::QWidget *parent\)](#)

Additional Inherited Members

4.16.1 Detailed Description

This class contains the gui elements for changing the options of a blurring filter.

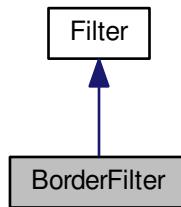
4.16.2 Constructor & Destructor Documentation

4.16.2.1 `BlurFilterBox (GUI::QWidget * parent)`

Constructor.

4.17 BorderFilter Class Reference

Inheritance diagram for BorderFilter:



Public Member Functions

- `BorderFilter ()`
- `bool getTop ()`
- `void setTop (bool top)`
- `bool getBottom ()`
- `void setBottom (bool bottom)`
- `bool getRight ()`
- `void setRight (bool right)`
- `bool getLeft ()`
- `void setLeft (bool left)`
- `int getThickness ()`
- `void setThickness (int thickness)`
- `QRgb getColor ()`
- `void setColor (QRgb color)`
- `string getName ()`
- `string getFilterDescription ()`

Private Attributes

- `bool top`
- `bool bottom`
- `bool right`
- `bool left`
- `int thickness`
- `QRgb color`

Additional Inherited Members

4.17.1 Detailed Description

Inserts border into the video

4.17.2 Constructor & Destructor Documentation

4.17.2.1 `BorderFilter()`

Constructor.

4.17.3 Member Function Documentation

4.17.3.1 `bool getBottom()`

Whether a border is inserted at the bottom.

Returns

True if a border is inserted at the bottom.

4.17.3.2 `QRgb getColor()`

Returns the color of the border,

Returns

The border color.

4.17.3.3 `string getFilterDescription() [virtual]`

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implements [Filter](#).

4.17.3.4 `bool getLeft()`

Whether a border is inserted at the left.

Returns

True if a border is inserted at the left.

4.17.3.5 `string getName() [virtual]`

Returns the name of the filter.

Returns

The filtername.

Implements [Filter](#).

4.17.3.6 bool getRight()

Whether a border is inserted at the right.

Returns

True if a border is inserted at the right.

4.17.3.7 int getThickness()

Returns the thickness of the border.

Returns

The thickness of the border.

4.17.3.8 bool getTop()

Whether a border is inserted at the top.

Returns

True if a border is inserted at the top.

4.17.3.9 void setBottom(bool *bottom*)

Sets whether a border is inserted at the bottom.

Parameters

<i>bottom</i>	True if a border is inserted at the bottom.
---------------	---

4.17.3.10 void setColor(QRgb *color*)

Sets the color of the border,

Parameters

<i>color</i>	The new border color.
--------------	-----------------------

4.17.3.11 void setLeft(bool *left*)

Sets whether a border is inserted at the left.

Parameters

<i>left</i>	True if a border is inserted at the left.
-------------	---

4.17.3.12 void setRight(bool *right*)

Sets whether a border is inserted at the right.

Parameters

<i>right</i>	True if a border is inserted at the right.
--------------	--

4.17.3.13 void setThickness (int *thickness*)

Sets the thickness of the border.

Parameters

<i>thickness</i>	The thickness.
------------------	----------------

4.17.3.14 void setTop (bool *top*)

Sets whether a border is inserted at the top.

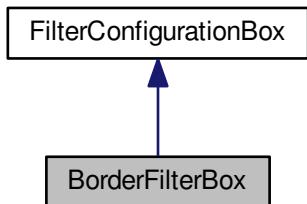
Parameters

<i>top</i>	True if a border is inserted at the top.
------------	--

4.17.4 Field Documentation**4.17.4.1 bool bottom [private]****4.17.4.2 QRgb color [private]****4.17.4.3 bool left [private]****4.17.4.4 bool right [private]****4.17.4.5 int thickness [private]****4.17.4.6 bool top [private]**

4.18 BorderFilterBox Class Reference

Inheritance diagram for BorderFilterBox:



Public Member Functions

- [BorderFilterBox \(GUI::QWidget *parent\)](#)

Additional Inherited Members

4.18.1 Detailed Description

This class contains the gui elements for changing the options of a border filter.

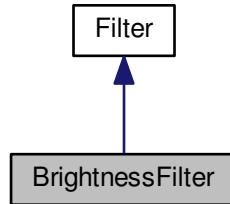
4.18.2 Constructor & Destructor Documentation

4.18.2.1 BorderFilterBox (GUI::QWidget * *parent*)

Constructor.

4.19 BrightnessFilter Class Reference

Inheritance diagram for BrightnessFilter:



Public Member Functions

- `BrightnessFilter()`
- `int getIntensity()`
- `void setIntensity(int intensity)`
- `string getName()`
- `string getFilterDescription()`

Private Attributes

- `int intensity`

Additional Inherited Members

4.19.1 Detailed Description

Adjusts the video brightness.

4.19.2 Constructor & Destructor Documentation

4.19.2.1 `BrightnessFilter()`

Constructor.

4.19.3 Member Function Documentation

4.19.3.1 `string getFilterDescription() [virtual]`

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implements [Filter](#).

4.19.3.2 int getIntensity ()

Returns the intensity of the brightness.

Returns

The intensity.

4.19.3.3 string getName () [virtual]

Returns the name of the filter.

Returns

The filtername.

Implements [Filter](#).

4.19.3.4 void setIntensity (int *intensity*)

Sets the intensity of the brightness.

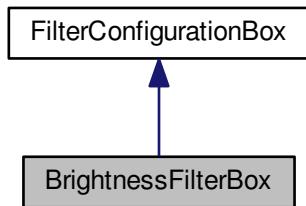
Parameters

<i>intensity</i>	The new intensity.
------------------	--------------------

4.19.4 Field Documentation**4.19.4.1 int intensity [private]**

4.20 BrightnessFilterBox Class Reference

Inheritance diagram for BrightnessFilterBox:



Public Member Functions

- `BrightnessFilterBox (GUI::QWidget *parent)`

Additional Inherited Members

4.20.1 Detailed Description

This class contains the gui elements for changing the options of a brightness filter.

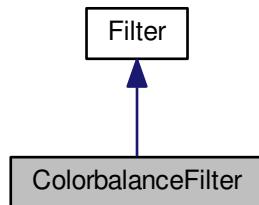
4.20.2 Constructor & Destructor Documentation

4.20.2.1 BrightnessFilterBox (GUI::QWidget * *parent*)

Constructor.

4.21 ColorbalanceFilter Class Reference

Inheritance diagram for ColorbalanceFilter:



Public Member Functions

- `ColorbalanceFilter ()`
- `Model::BasicColor getColor ()`
- `void setColor (Model::BasicColor color)`
- `int getIntensity ()`
- `void setIntensity (int intensity)`
- `bool getBrightPixels ()`
- `void setBrightPixels (bool brightPixels)`
- `bool getMediumPixels ()`
- `void setMediumPixels (bool mediumPixels)`
- `string getFilterDescription ()`
- `string getName ()`
- `bool getDarkPixels ()`
- `void setDarkPixels (bool darkPixels)`

Private Attributes

- `int intensity`
- `bool brightPixels`
- `bool mediumPixels`
- `bool darkPixels`
- `Model::BasicColor * color`

Additional Inherited Members

4.21.1 Detailed Description

Adjusts the colorbalance of the video for the 3 basic colors.

4.21.2 Constructor & Destructor Documentation

4.21.2.1 ColorbalanceFilter()

Constructor.

4.21.3 Member Function Documentation

4.21.3.1 bool getBrightPixels ()

Whether the bright pixels shall be changed.

Returns

True if the bright pixels are changed.

4.21.3.2 Model::BasicColor getColor ()

Returns the color whose balance is to be changed.

Returns

The color to change.

4.21.3.3 bool getDarkPixels ()

Whether the dark pixels shall be changed.

Returns

True if the dark pixels are changed.

4.21.3.4 string getFilterDescription () [virtual]

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implements [Filter](#).

4.21.3.5 int getIntensity ()

Returns the intensity of the change,

Returns

The intensity.

4.21.3.6 bool getMediumPixels ()

Whether the medium pixels shall be changed.

Returns

True if the medium pixels are changed.

4.21.3.7 string getName () [virtual]

Returns the name of the filter.

Returns

The filtername.

Implements [Filter](#).

4.21.3.8 void setBrightPixels (bool *brightPixels*)

Sets whether the bright pixels shall be changed.

Parameters

<i>brightPixels</i>	True if the bright pixels shall be changed.
---------------------	---

4.21.3.9 void setColor (Model::BasicColor *color*)

Sets the color whose balance shall be changed.

Parameters

<i>color</i>	The color to change.
--------------	----------------------

4.21.3.10 void setDarkPixels (bool *darkPixels*)

Sets whether the dark pixels shall be changed.

Parameters

<i>darkPixels</i>	True if the dark pixels are changed.
-------------------	--------------------------------------

4.21.3.11 void setIntensity (int *intensity*)

Sets the intensity of the change.

Parameters

<i>intensity</i>	The intensity.
------------------	----------------

4.21.3.12 void setMediumPixels (bool *mediumPixels*)

Sets whether the medium pixels shall be changed.

Parameters

<i>mediumPixels</i>	True if the medium pixels are changed.
---------------------	--

4.21.4 Field Documentation**4.21.4.1 bool brightPixels [private]****4.21.4.2 Model::BasicColor* color [private]**

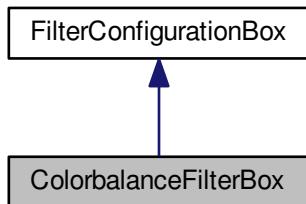
4.21.4.3 **bool darkPixels** [private]

4.21.4.4 **int intensity** [private]

4.21.4.5 **bool mediumPixels** [private]

4.22 ColorbalanceFilterBox Class Reference

Inheritance diagram for ColorbalanceFilterBox:



Public Member Functions

- [ColorbalanceFilterBox \(GUI::QWidget *parent\)](#)

Additional Inherited Members

4.22.1 Detailed Description

This class contains the gui elements for changing the options of a color balance filter.

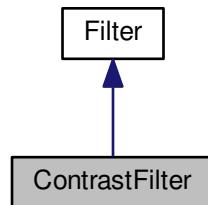
4.22.2 Constructor & Destructor Documentation

4.22.2.1 ColorbalanceFilterBox (GUI::QWidget * *parent*)

Constructor.

4.23 ContrastFilter Class Reference

Inheritance diagram for ContrastFilter:



Public Member Functions

- `ContrastFilter ()`
- `void setIntensity (int intensity)`
- `int getIntensity ()`
- `string getName ()`
- `string getFilterDescription ()`

Private Attributes

- `int intensity`

Additional Inherited Members

4.23.1 Detailed Description

Adjusts the contrast of the video.

4.23.2 Constructor & Destructor Documentation

4.23.2.1 `ContrastFilter ()`

Constructor.

4.23.3 Member Function Documentation

4.23.3.1 `string getFilterDescription () [virtual]`

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implements [Filter](#).

4.23.3.2 int getIntensity ()

Returns the intensity of the contrast.

Returns

The intensity.

4.23.3.3 string getName () [virtual]

Returns the name of the filter.

Returns

The filtername.

Implements [Filter](#).

4.23.3.4 void setIntensity (int *intensity*)

Sets the intensity of the contrast.

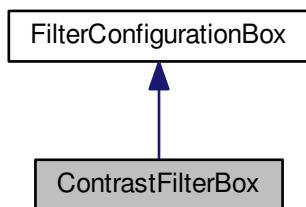
Parameters

<i>intensity</i>	The new intensity.
------------------	--------------------

4.23.4 Field Documentation**4.23.4.1 int intensity [private]**

4.24 ContrastFilterBox Class Reference

Inheritance diagram for ContrastFilterBox:



Public Member Functions

- [ContrastFilterBox \(GUI::QWidget *parent\)](#)

Additional Inherited Members

4.24.1 Detailed Description

This class contains the gui elements for changing the options of a contrast filter.

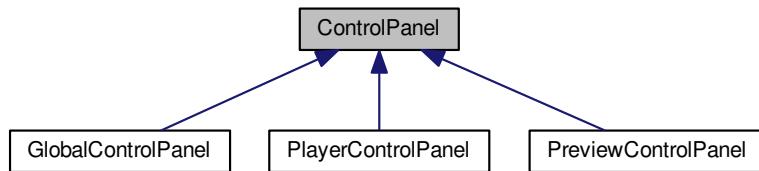
4.24.2 Constructor & Destructor Documentation

4.24.2.1 ContrastFilterBox (GUI::QWidget * *parent*)

Constructor.

4.25 ControlPanel Class Reference

Inheritance diagram for ControlPanel:



Public Member Functions

- `ControlPanel ()`
- `void setMasterVideoPlayer (Player::Player &player)`
- `void addVideoPlayer (GUI::Player &player)`
- `virtual void updateUi ()=0`
- `void removeVideoPlayer (Player::Player &player)`

Data Fields

- `GUI::ForwardPlayer * forwardPanel`
- `GUI::VideoPlayer * masterPanel`

Protected Attributes

- `std::vector< GUI::VideoPlayer * > players`

Private Attributes

- `GUI::Player * masterPlayer`

4.25.1 Detailed Description

This class is the base class for control panels. Control panels control videoplayers,

4.25.2 Constructor & Destructor Documentation

4.25.2.1 ControlPanel ()

Constructor.

4.25.3 Member Function Documentation

4.25.3.1 void addVideoPlayer (GUI::Player & player)

Adds the video player the list of players to notify.

Parameters

<i>player</i>	The player to add to the list.
---------------	--------------------------------

4.25.3.2 void removeVideoPlayer (Player::Player & *player*)

Removes the video player from the list of the players to notify.

Parameters

<i>player</i>	The player to remove.
---------------	-----------------------

4.25.3.3 void setMasterVideoPlayer (Player::Player & *player*)

Sets the master video player. The master video player is the reference to where to set the position of the slider, if the video is played paused or stopped.

Parameters

<i>player</i>	The master video player.
---------------	--------------------------

4.25.3.4 virtual void updateUi () [pure virtual]

Updates the ui of the control panel.

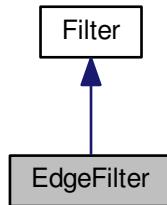
Implemented in [PlayerControlPanel](#), [PreviewControlPanel](#), and [GlobalControlPanel](#).

4.25.4 Field Documentation

4.25.4.1 GUI::ForwardPlayer* forwardPanel**4.25.4.2 GUI::VideoPlayer* masterPanel****4.25.4.3 GUI::Player* masterPlayer [private]****4.25.4.4 std::vector<GUI::VideoPlayer*> players [protected]**

4.26 EdgeFilter Class Reference

Inheritance diagram for EdgeFilter:



Public Member Functions

- [EdgeFilter \(\)](#)
- string [getFilterDescription \(\)](#)
- string [getName \(\)](#)

Additional Inherited Members

4.26.1 Detailed Description

Filters everything but the edges out of the video.

4.26.2 Constructor & Destructor Documentation

4.26.2.1 [EdgeFilter \(\)](#)

4.26.3 Member Function Documentation

4.26.3.1 [string getFilterDescription \(\) \[virtual\]](#)

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implements [Filter](#).

4.26.3.2 [string getName \(\) \[virtual\]](#)

Returns the name of the filter.

Returns

The filtername.

Implements [Filter](#).

4.27 EncodedVideo Class Reference

Public Member Functions

- `EncodedVideo (QString path)`
- `QString getPath ()`
- `int getFileSize ()`
- `int getNumberOfColors ()`
- `QString getCodec ()`
- `Model::Graph & getBitrate ()`
- `Model::Graph & getPsnr ()`
- `Model::Graph & getRedHistogramm ()`
- `Model::Graph & getBlueHistogramm ()`
- `Model::Graph & getGreenHistogramm ()`
- `Model::AVVideo & getAvVideo ()`
- `GUI::Video & getMacroBlockVideo ()`
- `GUI::Video & getRgbDiffVideo (GUI::Video *reference=0)`
- `GUI::Video & getVideo ()`
- `void setBitrate (Model::Graph graph)`
- `void setPsnr (Model::Graph graph)`
- `void setRedHistogramm (Model::Graph graph)`
- `void setGreenHistogramm (Model::Graph graph)`
- `void setBlueHistogramm (Model::Graph graph)`
- `void setMacroblockVideo (GUI::Video video)`
- `void setRgbDiffVideo (GUI::Video video)`

Data Fields

- `GUI::AnalysisBox * video`
- `Model::AVVideo * avVideo`
- `GUI::Video * displayVideo`
- `GUI::Video * macroblockVideo`
- `GUI::Video * rgbDiffVideo`
- `Model::Graph * bitrate`
- `Model::Graph * psnr`
- `Model::Graph * redHisto`
- `Model::Graph * greenHisto`
- `Model::Graph * blueHiso`

Private Member Functions

- `void loadVideo ()`

Private Attributes

- `QString path`
- `int fileSize`
- `int numberOfColors`
- `QString codec`

4.27.1 Detailed Description

This class contains all analysis info of an encoded video.

4.27.2 Constructor & Destructor Documentation

4.27.2.1 EncodedVideo (*QString path*)

Constructor.

Parameters

<i>path</i>	Path to the video.
-------------	--------------------

4.27.3 Member Function Documentation

4.27.3.1 Model::AVVideo & getAvVideo ()

Returns the [AVVideo](#).

Returns

The [AVVideo](#).

4.27.3.2 Model::Graph & getBitrate ()

Returns the bitrate graph.

Returns

The bitrate graph.

4.27.3.3 Model::Graph & getBlueHistogramm ()

Returns the blue histogramm graph.

Returns

The blue histogramm.

4.27.3.4 QString getCodec ()

Returns the codec used in the video file.

Returns

The used codec.

4.27.3.5 int getFileSize ()

Returns the size of the video file.

Returns

the file size.

4.27.3.6 Model::Graph & getGreenHistogramm()

Returns the green histogramm graph.

Returns

The green histogramm.

4.27.3.7 GUI::Video & getMacroBlockVideo()

Returns the video which shows the macroblocks.

Returns

The macroblock video.

4.27.3.8 int getNumberOfColors()

Returns the number of colors that appear in the whole video.

Returns

The number of colors in the video.

4.27.3.9 QString getPath()

Returns the path to the video.

Returns

The path to the video.

4.27.3.10 Model::Graph & getPsnr()

Returns the psnr graph.

Returns

The psnr graph.

4.27.3.11 Model::Graph & getRedHistogramm()

Returns the red histogramm graph.

Returns

The red histogramm.

4.27.3.12 GUI::Video & getRgbDiffVideo(GUI::Video * reference = 0)

Returns the video which shows the rgb difference to another video.

Parameters

<i>reference</i>	The video to compare to.
------------------	--------------------------

Returns

The rgb diff video.

4.27.3.13 GUI::Video & getVideo()

Returns the Video.

Returns

The Video.

4.27.3.14 void loadVideo() [private]

Loads the video from the file.

4.27.3.15 void setBitrate(Model::Graph *graph*)

Sets the bitrate graph.

Parameters

<i>graph</i>	The bitrate graph.
--------------	--------------------

4.27.3.16 void setBlueHistogramm(Model::Graph *graph*)

Sets the blue histogramm graph.

Parameters

<i>graph</i>	The blue histogramm.
--------------	----------------------

4.27.3.17 void setGreenHistogramm(Model::Graph *graph*)

Sets the green histogramm graph.

Parameters

<i>graph</i>	The green histogramm.
--------------	-----------------------

4.27.3.18 void setMacroblockVideo(GUI::Video *video*)

Sets the video that shows the macroblocks.

Parameters

<i>video</i>	The macroblcok video.
--------------	-----------------------

4.27.3.19 void setPsnr (Model::Graph *graph*)

Sets the psnr graph.

Parameters

<i>graph</i>	The psnr graph.
--------------	-----------------

4.27.3.20 void setRedHistogramm (Model::Graph *graph*)

Sets the red histogramm graph.

Parameters

<i>graph</i>	The red histogramm.
--------------	---------------------

4.27.3.21 void setRgbDiffVideo (GUI::Video *video*)

Sets the video that shows a rgb difference to another video.

Parameters

<i>video</i>	The rgb diff video.
--------------	---------------------

4.27.4 Field Documentation

4.27.4.1 Model::AVVideo* avVideo

4.27.4.2 Model::Graph* bitrate

4.27.4.3 Model::Graph* blueHiso

4.27.4.4 QString codec [private]

4.27.4.5 GUI::Video* displayVideo

4.27.4.6 int fileSize [private]

4.27.4.7 Model::Graph* greenHisto

4.27.4.8 GUI::Video* macroblockVideo

4.27.4.9 int numberOfColors [private]

4.27.4.10 QString path [private]

4.27.4.11 Model::Graph* psnr

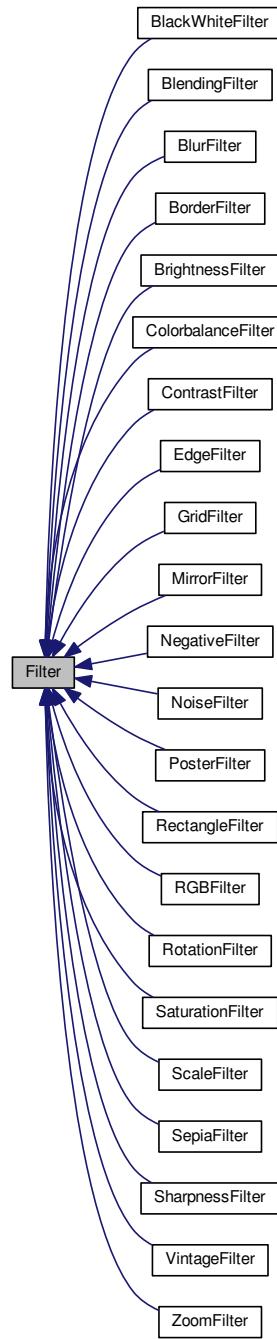
4.27.4.12 Model::Graph* redHisto

4.27.4.13 GUI::Video* rgbDiffVideo

4.27.4.14 GUI::AnalysisBox* video

4.28 Filter Class Reference

Inheritance diagram for Filter:



Public Member Functions

- virtual string `getFilterDescription ()=0`
- virtual string `getName ()=0`

Data Fields

- `UndoRedo::RemoveFilter * filter`

4.28.1 Detailed Description

Baseclass for Filters.

4.28.2 Member Function Documentation

4.28.2.1 `virtual string getFilterDescription() [pure virtual]`

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implemented in `BorderFilter`, `ColorbalanceFilter`, `BlendingFilter`, `BlurFilter`, `BrightnessFilter`, `ContrastFilter`, `RectangleFilter`, `GridFilter`, `NoiseFilter`, `ScaleFilter`, `MirrorFilter`, `RGBFilter`, `ZoomFilter`, `BlackWhiteFilter`, `PosterFilter`, `RotationFilter`, `SaturationFilter`, `SharpnessFilter`, `NegativeFilter`, `SepiaFilter`, `VintageFilter`, and `EdgeFilter`.

4.28.2.2 `string getName() [pure virtual]`

Returns the name of the filter.

Returns

The filtername.

Implemented in `BorderFilter`, `GridFilter`, `ColorbalanceFilter`, `RectangleFilter`, `BlendingFilter`, `BlurFilter`, `NoiseFilter`, `ScaleFilter`, `RGBFilter`, `BrightnessFilter`, `ContrastFilter`, `PosterFilter`, `RotationFilter`, `SaturationFilter`, `SharpnessFilter`, `MirrorFilter`, `NegativeFilter`, `ZoomFilter`, `BlackWhiteFilter`, `EdgeFilter`, `SepiaFilter`, and `VintageFilter`.

4.28.3 Field Documentation

4.28.3.1 `UndoRedo::RemoveFilter* filter`

4.29 FilterApplier Class Reference

Public Member Functions

- `FilterApplier (Model::FilterList &list)`
- `void applyToVideo (Model::AVVideo &target, Model::AVVideo &video)`

Private Member Functions

- `void initFilters ()`
- `AVFrame applyToFrame (AVFrame &frame)`

Private Attributes

- `Model::FilterList * list`

4.29.1 Detailed Description

Applies filters of a given `FilterList` to a video.

4.29.2 Constructor & Destructor Documentation

4.29.2.1 `FilterApplier (Model::FilterList & list)`

Constructor.

Parameters

<code>list</code>	The list with the filters to apply.
-------------------	-------------------------------------

4.29.3 Member Function Documentation

4.29.3.1 `AVFrame applyToFrame (AVFrame & frame) [private]`

Applies the filters to one frame.

Parameters

<code>frame</code>	The frame to apply the filters on.
--------------------	------------------------------------

Returns

The filtered frame.

4.29.3.2 `void applyToVideo (Model::AVVideo & target, Model::AVVideo & video)`

Applies the given filters to the video.

Parameters

<i>target</i>	The video to which the new frames are added to.
<i>video</i>	The video to apply the filters on.

4.29.3.3 void initFilters() [private]

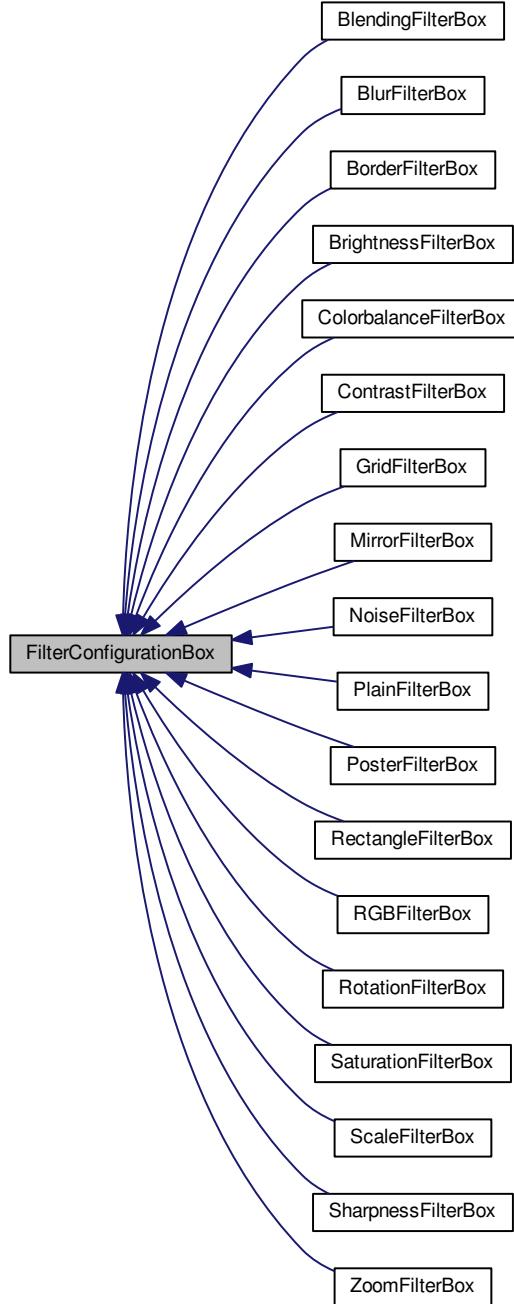
Initializes the filters.

4.29.4 Field Documentation

4.29.4.1 Model::FilterList* list [private]

4.30 FilterConfigurationBox Class Reference

Inheritance diagram for FilterConfigurationBox:



Public Member Functions

- `FilterConfigurationBox (GUI::QWidget *parent)`
- `void setFilter (Model::Filter &filter)`
- `Model::Filter * getFilter ()`

Protected Attributes

- Model::Filter * *filter*

4.30.1 Detailed Description

This class is the base class for the configuration boxes for the filters.

4.30.2 Constructor & Destructor Documentation

4.30.2.1 FilterConfigurationBox (GUI::QWidget * *parent*)

Constructor.

4.30.3 Member Function Documentation

4.30.3.1 Model::Filter * *getFilter* ()

Returns the filter the filterbox is responsible for.

Returns

The filter the filterbox shows the options for.

4.30.3.2 void *setFilter* (Model::Filter & *filter*)

Sets the filter the filterbox is responsible for.

Parameters

<i>filter</i>	The filter to show the options for.
---------------	-------------------------------------

4.30.4 Field Documentation

4.30.4.1 Model::Filter* *filter* [protected]

4.31 FilterConfigurationLoader Class Reference

Public Member Functions

- [FilterConfigurationLoader \(QString path\)](#)
- [Model::FilterList getConfiguration \(\)](#)

Private Attributes

- QFile [file](#)

4.31.1 Detailed Description

This class can load a Filterlist from a file.

4.31.2 Constructor & Destructor Documentation

4.31.2.1 FilterConfigurationLoader (QString *path*)

Constructor.

Parameters

<i>path</i>	The path to the filerlist to load.
-------------	------------------------------------

4.31.3 Member Function Documentation

4.31.3.1 Model::FilterList getConfiguration ()

Loads the filterlist.

Returns

The loaded filterlist.

4.31.4 Field Documentation

4.31.4.1 QFile *file* [private]

4.32 FilterConfigurationSaver Class Reference

Public Member Functions

- `FilterConfigurationSaver` (QString `file`, Model::FilterList &`filterList`)
- void `save` ()

Private Attributes

- QFile `file`

4.32.1 Detailed Description

This class can save a filterlist to a file.

4.32.2 Constructor & Destructor Documentation

4.32.2.1 FilterConfigurationSaver (QString `file`, Model::FilterList & `filterList`)

Constructor.

Parameters

<code>file</code>	Absolute path to the file to save to.
<code>filterList</code>	The filterlist to save.

4.32.3 Member Function Documentation

4.32.3.1 void `save` ()

Saves the filterlist.

4.32.4 Field Documentation

4.32.4.1 QFile `file` [private]

4.33 FilterContainerTab Class Reference

Public Member Functions

- `FilterContainerTab (GUI::QWidget *parent)`
- `void addFilter (Model::Filter filter)`
- `void setParentTab (GUI::FilterTab &parent)`
- `void uncheck (string filterName)`

Private Attributes

- `GUI::FilterTab * parentTab`
- `GUI::FilterTab * filterContainerTab`
- `std::vector< GUI::FilterView * > filterViews`

4.33.1 Detailed Description

This class shows all the selectable filters.

4.33.2 Constructor & Destructor Documentation

4.33.2.1 `FilterContainerTab (GUI::QWidget * parent)`

Constructor.

4.33.3 Member Function Documentation

4.33.3.1 `void addFilter (Model::Filter filter)`

Adds a selectable filter.

Parameters

<code>filter</code>	The new filter.
---------------------	-----------------

4.33.3.2 `void setParentTab (GUI::FilterTab & parent)`

Sets the parent tab.

Parameters

<code>parent</code>	The parent tab.
---------------------	-----------------

4.33.3.3 `void uncheck (string filterName)`

Searches for the filterView with the filter filterName and unchecks it.

Parameters

<i>filterName</i>	The filter to uncheck.
-------------------	------------------------

4.33.4 Field Documentation

- 4.33.4.1 **GUI::FilterTab*** `filterContainerTab` [private]
- 4.33.4.2 **std::vector<GUI::FilterView*>** `filterViews` [private]
- 4.33.4.3 **GUI::FilterTab*** `parentTab` [private]

4.34 FilterList Class Reference

Public Member Functions

- `FilterList ()`
- `Model::Filter * getFilterByName (string name)`
- `void removeFilter (string name)`
- `void moveFilter (int oldPosition, int newPosition)`
- `void removeFilter (int position)`
- `void addFilter (string name, int index=-1)`
- `Model::Filter * getFilterByIndex (int index)`
- `int getIndex (string name)`

Data Fields

- `Model::FilterApplier * list`
- `UndoRedo::FilterReset * filterList`
- `UndoRedo::LoadFilterconfig * oldList`
- `UndoRedo::LoadFilterconfig * newList`

Private Attributes

- `vector filters:std:`
- `std::vector< Model::Filter * > filters`

4.34.1 Detailed Description

This class contains a filter configuration. Every filter can only be once in the list.

4.34.2 Constructor & Destructor Documentation

4.34.2.1 FilterList()

Constructor.

4.34.3 Member Function Documentation

4.34.3.1 void addFilter(string name, int index = -1)

Inserts a filter at the given index. If the index is -1 then the filter is added to the end.

Parameters

<code>name</code>	Name of the filter to add.
<code>index</code>	Index to insert the filter at.

4.34.3.2 Model::Filter * getFilterByIndex(int index)

Returns the filter at the given index.

Parameters

<i>index</i>	Index of the filter.
--------------	----------------------

Returns

The filter at the given index.

4.34.3.3 Model::Filter * getFilterByName (string *name*)

Returns a filter by its name.

Parameters

<i>name</i>	The name of the filter.
-------------	-------------------------

Returns

The filter.

4.34.3.4 int getIndex (string *name*)

Returns the index of a filter.

Parameters

<i>name</i>	The name of the filter.
-------------	-------------------------

Returns

The index.

4.34.3.5 void moveFilter (int *oldPosition*, int *newPosition*)

Moves a filter to another position.

Parameters

<i>oldPosition</i>	The old position.
<i>newPosition</i>	The new position.

4.34.3.6 void removeFilter (string *name*)

Removes a filter.

Parameters

<i>name</i>	Name of the filter to remove.
-------------	-------------------------------

4.34.3.7 void removeFilter (int *position*)

Removes a filter.

Parameters

<i>position</i>	Position of the filter to remove.
-----------------	-----------------------------------

4.34.4 Field Documentation

4.34.4.1 **UndoRedo::FilterReset*** filterList

4.34.4.2 **vector filters** [private]

4.34.4.3 **std::vector<Model::Filter*> filters** [private]

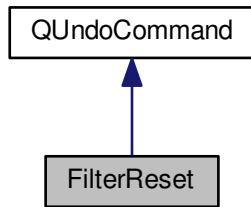
4.34.4.4 **Model::FilterApplier*** list

4.34.4.5 **UndoRedo::LoadFilterconfig*** newList

4.34.4.6 **UndoRedo::LoadFilterconfig*** oldList

4.35 FilterReset Class Reference

Inheritance diagram for FilterReset:



Public Member Functions

- `FilterReset (GUI::FilterTab *filterTab, Model::FilterList filterList)`
- `void undo ()`
- `void redo ()`

Private Attributes

- `GUI::FilterTab * filterTab`
- `Model::FilterList * filterList`

4.35.1 Detailed Description

This class is the undo command for resting the filterlist in the filter tab.

4.35.2 Constructor & Destructor Documentation

4.35.2.1 `FilterReset (GUI::FilterTab * filterTab, Model::FilterList filterList)`

Constructor.

Parameters

<code>filterTab</code>	The filtertab to operate on.
<code>filterList</code>	The filterlist the action to perform on.

4.35.3 Member Function Documentation

4.35.3.1 `void redo ()`

Clears the filter configurations and the filter list.

4.35.3.2 `void undo ()`

Loads the filterlist and filter configuration to the state it was before the reset.

4.35.4 Field Documentation

4.35.4.1 **Model::FilterList* filterList** [private]

4.35.4.2 **GUI::FilterTab* filterTab** [private]

4.36 FilterTab Class Reference

Public Member Functions

- `FilterTab (GUI::QWidget *parent)`
- `Memento::FilterTabMemento getMemento ()`
- `void restore (Memento::FilterTabMemento memento)`
- `void insertFilter (Model::Filter filter, int index=-1)`
- `void removeFilter (string filterName)`
- `void showVideo ()`
- `void showPreview ()`
- `void resetFilters ()`
- `void setFilterList (Model::FilterList list)`
- `void setRawVideo (Model::YuvVideo video)`
- `void moveFilter (int old, int new_3)`

Data Fields

- `UndoRedo::LoadFilterVideo * filterTab`

Private Member Functions

- `void connectActions ()`
- `void createUi ()`
- `void up ()`
- `void down ()`
- `void remove ()`
- `void load ()`
- `void apply ()`
- `void saveConf ()`
- `void loadConf ()`
- `void reset ()`
- `void save ()`
- `void listSelectionChanged (QModelIndex index)`

Private Attributes

- `QPushButton * button_up`
- `QPushButton * button_down`
- `QPushButton * button_remove`
- `QPushButton * button_load`
- `QPushButton * button_apply`
- `QPushButton * button_saveConf`
- `QPushButton * button_loadConf`
- `QPushButton * button_reset`
- `QPushButton * button_save`
- `QListWidget * list_filterList`
- `QTabWidget * tab_filterTabs`
- `QLabel * label_filterOptions`
- `GUI::QFrame * frame_filterContainer`
- `QStringListModel * model_list`
- `GUI::VideoPlayer * player`
- `GUI::PreviewControlPanel * previewPanel`

- `GUI::FrameView * frameView`
- `GUI::PlayerControlPanel * playerPanel`
- `Model::FilterList * filterList`
- `Model::YuvVideo * rawVideo`
- `std::vector< GUI::FilterContainerTab * > filterContainerTab`

4.36.1 Detailed Description

This class is the tab to filter videos.

4.36.2 Constructor & Destructor Documentation

4.36.2.1 `FilterTab (QWidget * parent)`

Constructor.

4.36.3 Member Function Documentation

4.36.3.1 `void apply () [private]`

Slot: connected with button_apply.pressed() applies the filter list on the video and shows it

4.36.3.2 `void connectActions () [private]`

Connect the buttons to their slots in this class.

4.36.3.3 `void createUi () [private]`

Creates the user interface.

4.36.3.4 `void down () [private]`

Slot: connected with button_down.pressed() increases the index of a filter in the list

4.36.3.5 `Memento::FilterTabMemento getMemento ()`

Creates a memento which contains the state of this tab.

Returns

The created memento.

4.36.3.6 `void insertFilter (Model::Filter filter, int index = -1)`

Inserts a filter to the filterList. If index is -1 the the filter is added to the end.

Parameters

<i>filter</i>	The filter to add.
<i>index</i>	The index to insert the filter at.

4.36.3.7 void listSelectionChanged (QModelIndex *index*) [private]

Slot: connected with list_filterList.clicked(index : QModelIndex); updates the filterConfigurationBox that is shown

Parameters

<i>index</i>	Index of the selected item.
--------------	-----------------------------

4.36.3.8 void load () [private]

Slot: connected with button_load.pressed() opens QFileDialog and opens the video in the selected path

4.36.3.9 void loadConf () [private]

Slot: connected with button_loadConf.pressed() opens QFileDialog and loads the filter configuration in the selected path

4.36.3.10 void moveFilter (int *old*, int *new_3*)

Moves a filter in the filterlist.

Parameters

<i>old</i>	Old list index.
<i>new</i>	New list index.

4.36.3.11 void remove () [private]

Slot: connected with button_remove.pressed() removes a filter from the list

4.36.3.12 void removeFilter (string *filterName*)

Removes the filter with the name *filterName* in the filterlist.

Parameters

<i>filterName</i>	Name of the filter to remove.
-------------------	-------------------------------

4.36.3.13 void reset () [private]

Slot: connected with button_reset.pressed() resets the filter list

4.36.3.14 void resetFilters ()

Resets the filterlist.

4.36.3.15 void restore (Memento::FilterTabMemento *memento*)

Restores the tab based on the memento.

Parameters

<i>memento</i>	The memento which contains the state of the tab.
----------------	--

4.36.3.16 void save() [private]

Slot: connected with button_save.pressed() opens QFileDialog and saves the video with the filters used on it.

4.36.3.17 void saveConf() [private]

Slot: connected with button_saveConf.pressed() opens QFileDialog and saves the filter configuration in the selected path

4.36.3.18 void setFilterList (Model::FilterList *list*)

Sets the filterlist.

Parameters

<i>list</i>	The filterlist to use.
-------------	------------------------

4.36.3.19 void setRawVideo (Model::YuvVideo *video*)

Sets the video the filters are applied to. This operation resets the whole filtertab.

Parameters

<i>video</i>	The video to apply the filters on.
--------------	------------------------------------

4.36.3.20 void showPreview ()

Shows the 5 frame preview.

4.36.3.21 void showVideo ()

Shows the video with the applied filters.

4.36.3.22 void up() [private]

Slot: connected with button_up.pressed() decreases the index of a filter in the list

4.36.4 Field Documentation

4.36.4.1 QPushButton* button_apply [private]**4.36.4.2 QPushButton* button_down [private]****4.36.4.3 QPushButton* button_load [private]****4.36.4.4 QPushButton* button_loadConf [private]**

- 4.36.4.5 **QPushButton*** button_remove [private]
- 4.36.4.6 **QPushButton*** button_reset [private]
- 4.36.4.7 **QPushButton*** button_save [private]
- 4.36.4.8 **QPushButton*** button_saveConf [private]
- 4.36.4.9 **QPushButton*** button_up [private]
- 4.36.4.10 **std::vector<GUI::FilterContainerTab*>** filterContainerTab [private]
- 4.36.4.11 **Model::FilterList*** filterList [private]
- 4.36.4.12 **UndoRedo::LoadFilterVideo*** filterTab
- 4.36.4.13 **GUI::QFrame*** frame_filterContainer [private]
- 4.36.4.14 **GUI::FrameView*** frameView [private]
- 4.36.4.15 **QLabel*** label_filterOptions [private]
- 4.36.4.16 **QWidget*** list_filterList [private]
- 4.36.4.17 **QStringListModel*** model_list [private]
- 4.36.4.18 **GUI::VideoPlayer*** player [private]
- 4.36.4.19 **GUI::PlayerControlPanel*** playerPanel [private]
- 4.36.4.20 **GUI::PreviewControlPanel*** previewPanel [private]
- 4.36.4.21 **Model::YuvVideo*** rawVideo [private]
- 4.36.4.22 **QTabWidget*** tab_filterTabs [private]

4.37 FilterTabMemento Class Reference

Public Member Functions

- `FilterTabMemento ()`
- `Model::FilterList getFilterList ()`
- `void setFilterList (Model::FilterList filterList)`
- `bool getWasApplied ()`
- `void setWasApplied (bool wasApplied)`
- `int getDisplayedFrame ()`
- `void setDisplayedFrame (int displayedFrame)`
- `string getLoadedFile ()`
- `void setLoadedFile (string loadedFile)`

Data Fields

- `UndoRedo::LoadFilterVideo * memento`

Private Attributes

- `Model::FilterList filterList`
- `bool wasApplied`
- `int displayedFrame`
- `string loadedFile:std:`

4.37.1 Detailed Description

This class is the memento for the FilterTab.

4.37.2 Constructor & Destructor Documentation

4.37.2.1 FilterTabMemento ()

Constructor.

4.37.3 Member Function Documentation

4.37.3.1 int getDisplayedFrame ()

Returns the currently displayed frame in the frame preview.

Returns

The currently displayed frame.

4.37.3.2 Model::FilterList getFilterList ()

Returns the list of the currently selected filters.

Returns

List of the selected filters.

4.37.3.3 string getLoadedFile()

Returns the path to the currently loaded yuv file.

Returns

Absolute path to the currently loaded yuv file.

4.37.3.4 bool getWasApplied()

Whether the filter were already applied.

Returns

True if the filter were already applied.

4.37.3.5 void setDisplayedFrame(int *displayedFrame*)

Sets the currently displayed frame in the frame preview.

Parameters

<i>displayedFrame</i>	The currently displayed frame.
-----------------------	--------------------------------

4.37.3.6 void setFilterList(Model::FilterList *filterList*)

Sets the list of the currently selected filters.

Parameters

<i>filterList</i>	List of the selected filters.
-------------------	-------------------------------

4.37.3.7 void setLoadedFile(string *loadedFile*)

Sets the path to the currently loaded yuv file.

Parameters

<i>loadedFile</i>	Absolute path to the loaded yuv file.
-------------------	---------------------------------------

4.37.3.8 void setWasApplied(bool *wasApplied*)

Sets whether the filters were already aplied.

Parameters

<i>wasApplied</i>	True if the filter were already applied.
-------------------	--

4.37.4 Field Documentation**4.37.4.1 int displayedFrame [private]****4.37.4.2 Model::FilterList filterList [private]**

4.37.4.3 **string loadedFile** [private]

4.37.4.4 **UndoRedo::LoadFilterVideo* memento**

4.37.4.5 **bool wasApplied** [private]

4.38 FilterView Class Reference

Public Member Functions

- `FilterView (GUI::QWidget *parent)`
- `void setFilter (Model::Filter filter)`
- `void setFilterTab (GUI::FilterTab *filtertab)`

Private Member Functions

- `void checkBoxStateChanged (int state)`

Static Private Member Functions

- `static QImage getDefaultImage ()`

Private Attributes

- `GUI::QCheckBox * checkbox`
- `QLabel * preview`
- `GUI::FilterTab * filterTab`
- `Model::Filter * filter`

4.38.1 Detailed Description

Represents a selectable filter in the gui. Shows a example of the filter and a checkbox as well as its name.

4.38.2 Constructor & Destructor Documentation

4.38.2.1 `FilterView (GUI::QWidget * parent)`

Constructor.

4.38.3 Member Function Documentation

4.38.3.1 `void checkBoxStateChanged (int state) [private]`

Slot: connected with `checkbox.stateChanged(state : int)` does `filterTab.addFilter(filter)` or `filterTab.removeFilter(filter.getName())`

4.38.3.2 `static QImage getDefaultImage () [inline], [static], [private]`

Returns the default image on which the filter is applied as a preview.

Returns

The default image.

4.38.3.3 `void setFilter (Model::Filter filter)`

Sets the filter this view represents.

Parameters

<i>filter</i>	The filter for this view.
---------------	---------------------------

4.38.3.4 void setFilterTab (GUI::FilterTab * *filtertab*)

Sets the tab this view is contained in.

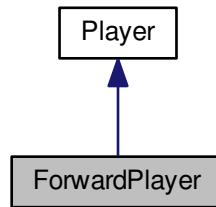
Parameters

<i>filtertab</i>	The parent filtertab.
------------------	-----------------------

4.38.4 Field Documentation**4.38.4.1 GUI::QCheckBox* checkbox [private]****4.38.4.2 Model::Filter* filter [private]****4.38.4.3 GUI::FilterTab* filterTab [private]****4.38.4.4 QLabel* preview [private]**

4.39 ForwardPlayer Class Reference

Inheritance diagram for ForwardPlayer:



Public Member Functions

- [ForwardPlayer \(\)](#)
- void [setForwardControlPanel \(GUI::ControlPanel panel\)](#)
- void [play \(\)](#)
- void [pause \(\)](#)
- void [stop \(\)](#)
- void [nextFrame \(\)](#)
- void [previousFrame \(\)](#)
- void [setSpeed \(float speed\)](#)
- void [setPosition \(int position\)](#)
- int [getPosition \(\)](#)
- float [getSpeed \(\)](#)
- bool [isPlaying \(\)](#)
- bool [isStopped \(\)](#)
- void [reset \(\)](#)

Private Attributes

- [GUI::ControlPanel * controlPanel](#)
- [GUI::ControlPanel * forwardPanel](#)

Additional Inherited Members

4.39.1 Detailed Description

This player forwards its input to a [ControlPanel](#).

4.39.2 Constructor & Destructor Documentation

4.39.2.1 ForwardPlayer()

Constructor.

4.39.3 Member Function Documentation

4.39.3.1 `int getPosition() [virtual]`

Returns the position in the video.

Returns

The current position.

Implements [Player](#).

4.39.3.2 `float getSpeed() [virtual]`

Returns the speed.

Returns

The current speed.

Implements [Player](#).

4.39.3.3 `bool isPlaying() [virtual]`

Whether the player is currently playing.

Returns

True if the player is playing.

Implements [Player](#).

4.39.3.4 `bool isStopped() [virtual]`

Whether the player is stopped.

Returns

True if the player is stopped.

Implements [Player](#).

4.39.3.5 `void nextFrame() [virtual]`

Shows the next frame.

Implements [Player](#).

4.39.3.6 `void pause() [virtual]`

Pauses the video.

Implements [Player](#).

4.39.3.7 void play() [virtual]

Plays the video.

Implements [Player](#).

4.39.3.8 void previousFrame() [virtual]

Shows the previous frame.

Implements [Player](#).

4.39.3.9 void reset() [virtual]

Resets the player.

Implements [Player](#).

4.39.3.10 void setForwardControlPanel(GUI::ControlPanel panel)

Sets the control panel that the player forwards its input to.

Parameters

<i>panel</i>	The player to forward to.
--------------	---------------------------

4.39.3.11 void setPosition(int position) [virtual]

Sets the position in the video.

Parameters

<i>position</i>	The new position.
-----------------	-------------------

Implements [Player](#).

4.39.3.12 void setSpeed(float speed) [virtual]

Sets the speed.

Parameters

<i>speed</i>	The new speed.
--------------	----------------

Implements [Player](#).

4.39.3.13 void stop() [virtual]

Stops the video.

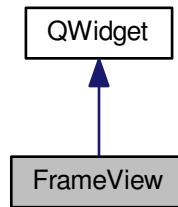
Implements [Player](#).

4.39.4 Field Documentation

4.39.4.1 GUI::ControlPanel* controlPanel [private]**4.39.4.2 GUI::ControlPanel* forwardPanel [private]**

4.40 FrameView Class Reference

Inheritance diagram for FrameView:



Public Member Functions

- `FrameView (QWidget *parent=0)`
- `void setFrame (QImage &frame)`
- `void clear ()`

Protected Member Functions

- `void resizeEvent (QResizeEvent *event)`
- `void repaintEvent (QPaintEvent *event)`

Private Member Functions

- `void () updateOffset()`

Private Attributes

- `int xOffset`
- `int yOffset`
- `QImage currentFrame`
- `QImage * originalFrame`

4.40.1 Detailed Description

This class is the view used by the video player. It automatically scales the frames passed to it.

4.40.2 Constructor & Destructor Documentation

4.40.2.1 `FrameView (QWidget * parent = 0)`

Constructor.

4.40.3 Member Function Documentation

4.40.3.1 void clear()

Clears the current frame so nothing is shown.

4.40.3.2 void repaintEvent(QPaintEvent * event) [protected]

This method is called when the widget has to be repainted.

4.40.3.3 void resizeEvent(QResizeEvent * event) [protected]

This method is called when the widget got resized.

4.40.3.4 void setFrame(QImage & frame)

Sets the frame to show.

Parameters

<i>frame</i>	The frame to show.
--------------	--------------------

4.40.3.5 void() [private]

Updates the offset at which the image is drawn.

4.40.4 Field Documentation

4.40.4.1 QImage currentFrame [private]

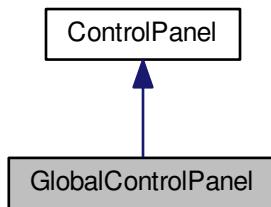
4.40.4.2 QImage* originalFrame [private]

4.40.4.3 int xOffset [private]

4.40.4.4 int yOffset [private]

4.41 GlobalControlPanel Class Reference

Inheritance diagram for GlobalControlPanel:



Public Member Functions

- [GlobalControlPanel \(\)](#)
- void [updateUi \(\)](#)
- void [play \(\)](#)
- void [pause \(\)](#)
- void [stop \(\)](#)
- void [nextFrame \(\)](#)
- void [previousFrame \(\)](#)
- void [setPosition \(int position\)](#)

Data Fields

- [GUI::AnalysisTab * globalControlPanel](#)

Additional Inherited Members

4.41.1 Detailed Description

This control panel has no gui. Instead it has functions to control the video.

4.41.2 Constructor & Destructor Documentation

4.41.2.1 GlobalControlPanel ()

Constructor.

4.41.3 Member Function Documentation

4.41.3.1 void nextFrame ()

Sends a nextFrame signal to the players.

4.41.3.2 void pause()

Sends a pause signal to the players.

4.41.3.3 void play()

Sends a play signal to the players.

4.41.3.4 void previousFrame()

Sends a previousFrame signal to the players.

4.41.3.5 void setPosition(int *position*)

Sends a setPosition signal to the players.

Parameters

<i>position</i>	The position to show.
-----------------	-----------------------

4.41.3.6 void stop()

Sends a stop signal to the players.

4.41.3.7 void updateUi() [virtual]

Updates the ui of the control panel.

Implements [ControlPanel](#).

4.41.4 Field Documentation

4.41.4.1 GUI::AnalysisTab* globalControlPanel

4.42 Graph Class Reference

Public Member Functions

- `Graph()`
- `void.addValue(int x, double y)`
- `void.cut(int x)`
- `double.getValue(int x)`
- `int.getLength()`
- `void.removeValue(int x)`

Data Fields

- `Model::EncodedVideo * bitrate`
- `Model::EncodedVideo * psnr`
- `Model::EncodedVideo * redHisto`
- `Model::EncodedVideo * greenHisto`
- `Model::EncodedVideo * blueHisto`

Private Attributes

- `vector<double> graph`

4.42.1 Detailed Description

This class is a graph.

4.42.2 Constructor & Destructor Documentation

4.42.2.1 `Graph()`

Constructor.

4.42.3 Member Function Documentation

4.42.3.1 `void.addValue(int x, double y)`

Adds a value pair.

Parameters

<code>x</code>	Value on the x-axes.
<code>y</code>	Value on the y-axes.

4.42.3.2 `void.cut(int x)`

Cuts the number of vectors down up to a certain value x.

Parameters

x	The last x-value in the cut down vectors.
---	---

4.42.3.3 int getLength ()

Returns the biggest x value.

Returns

The biggest x value.

4.42.3.4 double getValue (int x)

Returns the y-value to a specific x-value.

Parameters

x	The x value.
---	--------------

4.42.3.5 void removeValue (int x)

Removes the corresponding y value.

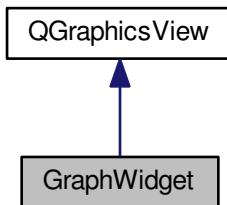
Parameters

x	The x value whose y value shall be removed.
---	---

4.42.4 Field Documentation**4.42.4.1 Model::EncodedVideo* bitrate****4.42.4.2 Model::EncodedVideo* blueHiso****4.42.4.3 vector<double> graph [private]****4.42.4.4 Model::EncodedVideo* greenHisto****4.42.4.5 Model::EncodedVideo* psnr****4.42.4.6 Model::EncodedVideo* redHisto**

4.43 GraphWidget Class Reference

Inheritance diagram for GraphWidget:



Public Member Functions

- `GraphWidget ()`
- `void drawGraph (Model::Graph graph, bool filled)`
- `void setLineColor (QRgb color)`
- `void setFillColor (QRgb color)`
- `void setControlPanel (GUI::GlobalControlPanel *panel)`

Data Fields

- `GUI::AnalysisBox * psnrGraph`
- `GUI::AnalysisBox * bitrateGraph`
- `GUI::AnalysisBox * redHistogramm`
- `GUI::AnalysisBox * blueHistogramm`
- `GUI::AnalysisBox * greenHistogramm`
- `Model::Graph * graph`

Protected Member Functions

- `void mouseReleaseEvent (QMouseEvent *event)`

Private Attributes

- `QRgb lineColor`
- `QRgb fillColor`
- `GloablControlPanel * controlPanel`

4.43.1 Detailed Description

This class is a widget to draw graphs.

4.43.2 Constructor & Destructor Documentation

4.43.2.1 GraphWidget ()

Constructor.

4.43.3 Member Function Documentation

4.43.3.1 void drawGraph (Model::Graph *graph*, bool *filled*)

Draws a graph to the widget.

Parameters

<i>graph</i>	The graph to draw.
<i>filled</i>	Whether the area under the graph is filled.

4.43.3.2 void mouseReleaseEvent (QMouseEvent * *event*) [protected]

This method is called if there was a click on the widget.

4.43.3.3 void setControlPanel (GUI::GlobalControlPanel * *panel*)

Sets the [GlobalControlPanel](#) to notify if a click on the graph was performed.

Parameters

<i>panel</i>	The panel to notify.
--------------	----------------------

4.43.3.4 void setFillColor (QRgb *color*)

Determines the color of the area beneath the graph line.

Parameters

<i>color</i>	The color in which the area beneath the graph line is filled.
--------------	---

4.43.3.5 void setLineColor (QRgb *color*)

Determines the color of the graph line.

Parameters

<i>color</i>	The color in which the line is shown.
--------------	---------------------------------------

4.43.4 Field Documentation

4.43.4.1 GUI::AnalysisBox* bitrateGraph

4.43.4.2 GUI::AnalysisBox* blueHistogramm

4.43.4.3 GlobalControlPanel* controlPanel [private]

4.43.4.4 QRgb fillColor [private]

4.43.4.5 Model::Graph* graph

4.43.4.6 GUI::AnalysisBox* greenHistogramm

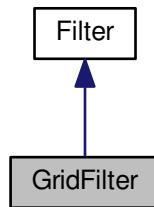
4.43.4.7 QRgb lineColor [private]

4.43.4.8 GUI::AnalysisBox* psnrGraph

4.43.4.9 GUI::AnalysisBox* redHistogramm

4.44 GridFilter Class Reference

Inheritance diagram for GridFilter:



Public Member Functions

- `GridFilter ()`
- string `getFilterDescription ()`
- int `getHorizontalLines ()`
- void `setHorizontalLines (int horizontalLines)`
- int `getVerticalLines ()`
- void `setVerticalLines (int verticalLines)`
- QRgb `getColor ()`
- void `setColor (QRgb color)`
- int `getThickness ()`
- void `setThickness (int thickness)`
- int `getOpacity ()`
- void `setOpacity (int opacity)`
- string `getName ()`

Private Attributes

- int `horizontalLines`
- int `verticalLines`
- QRgb `color`
- int `thickness`
- int `opacity`

Additional Inherited Members

4.44.1 Detailed Description

Inserts a grid into the video as an overlay.

4.44.2 Constructor & Destructor Documentation

4.44.2.1 GridFilter()

Constructor.

4.44.3 Member Function Documentation

4.44.3.1 QRgb getColor ()

Returns the color of the grid.

Returns

The gridcolor.

4.44.3.2 string getFilterDescription () [virtual]

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implements [Filter](#).

4.44.3.3 int getHorizontalLines ()

Returns the number of horizontal drawn lines.

Returns

Number of horizontal lines.

4.44.3.4 string getName () [virtual]

Returns the name of the filter.

Returns

The filtername.

Implements [Filter](#).

4.44.3.5 int getOpacity ()

Returns the opacity of the grid.

Returns

The grids opacity.

4.44.3.6 int getThickness ()

Returns the thickness of the drawn lines.

Returns

The line thickness.

4.44.3.7 int getVerticalLines()

Returns the number of vertical drawn lines.

Returns

Number of vertical lines.

4.44.3.8 void setColor (QRgb color)

Sets the color of the grid.

Parameters

<i>color</i>	The gridcolor.
--------------	----------------

4.44.3.9 void setHorizontalLines (int horizontalLines)

Sets the number of horizontal drawn lines.

Parameters

<i>horizontalLines</i>	Number of horizontal lines.
------------------------	-----------------------------

4.44.3.10 void setOpacity (int opacity)

Sets the opacity of the grid.

Parameters

<i>opacity</i>	The grids opacity.
----------------	--------------------

4.44.3.11 void setThickness (int thickness)

Sets the thickness of the drawn lines.

Parameters

<i>thickness</i>	The thickness of the drawn lines.
------------------	-----------------------------------

4.44.3.12 void setVerticalLines (int verticalLines)

Sets the number of vertical drawn lines.

Parameters

<i>verticalLines</i>	Number of vertical drawn lines.
----------------------	---------------------------------

4.44.4 Field Documentation

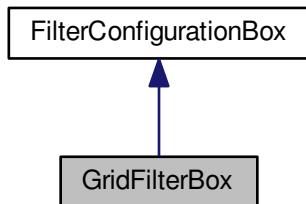
4.44.4.1 QRgb color [private]**4.44.4.2 int horizontalLines [private]****4.44.4.3 int opacity [private]**

4.44.4.4 int thickness [private]

4.44.4.5 int verticalLines [private]

4.45 GridFilterBox Class Reference

Inheritance diagram for GridFilterBox:



Public Member Functions

- [GridFilterBox \(GUI::QWidget *parent\)](#)

Additional Inherited Members

4.45.1 Detailed Description

This class contains the gui elements for changing the options of a grid filter.

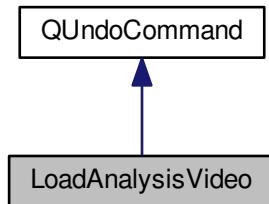
4.45.2 Constructor & Destructor Documentation

4.45.2.1 GridFilterBox (GUI::QWidget * *parent*)

Constructor.

4.46 LoadAnalysisVideo Class Reference

Inheritance diagram for LoadAnalysisVideo:



Public Member Functions

- [LoadAnalysisVideo \(GUI::AnalysisTab *anaTab, AnalyseTabMemento anaTabMem, Model::YuvVideo video\)](#)
- void [undo \(\)](#)
- void [redo \(\)](#)

Data Fields

- [Memento::AnalysisTabMemento * memento](#)

Private Attributes

- [GUI::AnalysisTab * anaTab](#)
- [Model::YuvVideo * video](#)

4.46.1 Detailed Description

This class is the undo command for loading the raw video on the analysis tab.

4.46.2 Constructor & Destructor Documentation

4.46.2.1 LoadAnalysisVideo (GUI::AnalysisTab * *anaTab*, AnalyseTabMemento *anaTabMem*, Model::YuvVideo *video*)

Constructor..

Parameters

<i>anaTab</i>	The AnalyseTab to operate on.
<i>anaTabMem</i>	The memento of the analyse tab before the raw video is loaded.
<i>video</i>	The new raw video.

4.46.3 Member Function Documentation

4.46.3.1 void redo ()

Loads anew raw video in the analysis tab.

4.46.3.2 void undo ()

Restores the analysis tab to the state before the new video was loaded.

4.46.4 Field Documentation

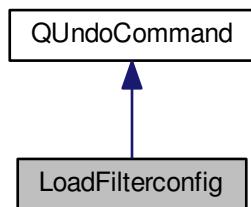
4.46.4.1 GUI::AnalysisTab* anaTab [private]

4.46.4.2 Memento::AnalysisTabMemento* memento

4.46.4.3 Model::YuvVideo* video [private]

4.47 LoadFilterconfig Class Reference

Inheritance diagram for LoadFilterconfig:



Public Member Functions

- `LoadFilterconfig (GUI::FilterTab *filterTab, Model::FilterList oldList, Model::FilterList list)`
- `void undo ()`
- `void redo ()`

Private Attributes

- `GUI::FilterTab * filterTab`
- `Model::FilterList * oldList`
- `Model::FilterList * newList`

4.47.1 Detailed Description

This class is the undo command for loading a filter config on the filter tab.

4.47.2 Constructor & Destructor Documentation

4.47.2.1 LoadFilterconfig (`GUI::FilterTab * filterTab, Model::FilterList oldList, Model::FilterList list`)

Constructor.

Parameters

<code>filterTab</code>	The filtertab to operate on.
<code>oldList</code>	The filterlist before the config is loaded.
<code>list</code>	The new filter configuration.

4.47.3 Member Function Documentation

4.47.3.1 void redo ()

Loads a filter configuration from a external file.

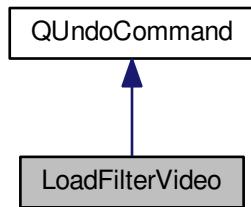
4.47.3.2 void undo()

Loads the filter configuration present before external configuration was loaded.

4.47.4 Field Documentation**4.47.4.1 GUI::FilterTab* filterTab [private]****4.47.4.2 Model::FilterList* newList [private]****4.47.4.3 Model::FilterList* oldList [private]**

4.48 LoadFilterVideo Class Reference

Inheritance diagram for LoadFilterVideo:



Public Member Functions

- [LoadFilterVideo \(GUI::FilterTab *filterTab, Model::YuvVideo video, Memento::FilterTabMemento memento\)](#)
- void [undo \(\)](#)
- void [redo \(\)](#)

Private Attributes

- Memento::FilterTabMemento * [memento](#)
- GUI::FilterTab * [filterTab](#)
- Model::YuvVideo * [video](#)

4.48.1 Detailed Description

This class is the undo command for loading a raw video in the filtertab.

4.48.2 Constructor & Destructor Documentation

4.48.2.1 LoadFilterVideo (GUI::FilterTab * *filterTab*, Model::YuvVideo *video*, Memento::FilterTabMemento *memento*)

Constructor.

Parameters

<i>filterTab</i>	The filtertab to operate on.
<i>video</i>	The video to use.
<i>memento</i>	The memento before the new video is loaded.

4.48.3 Member Function Documentation

4.48.3.1 void redo ()

Loads video to which filter can be applied.

4.48.3.2 void undo ()

Removes current video to which filters can be applied and loads previous video.

4.48.4 Field Documentation**4.48.4.1 GUI::FilterTab* filterTab [private]****4.48.4.2 Memento::FilterTabMemento* memento [private]****4.48.4.3 Model::YuvVideo* video [private]**

4.49 MacroblockCalculator Class Reference

Public Member Functions

- void `macroBlockCalculator (Model::AVVideo &video)`
- void `calculateMacroblockImages (GUI::Video &target)`

Private Attributes

- `Model::AVVideo * video`

4.49.1 Detailed Description

This class calculates the macroblocks of a video.

4.49.2 Member Function Documentation

4.49.2.1 void calculateMacroblockImages (GUI::Video & target)

Calculates the Video with macroblock overlay.

Parameters

<code>target</code>	The video the frames with the calculated macroblocks are added to.
---------------------	--

4.49.2.2 void macroBlockCalculator (Model::AVVideo & video)

Constructor.

Parameters

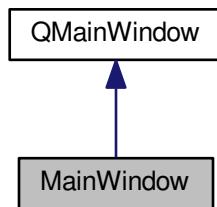
<code>video</code>	The video to calculate the macroblocks for.
--------------------	---

4.49.3 Field Documentation

4.49.3.1 Model::AVVideo* video [private]

4.50 MainWindow Class Reference

Inheritance diagram for MainWindow:



Public Member Functions

- `MainWindow (GUI::QWidget *parent)`
- `Memento::MainWindowMemento getMemento ()`
- `void restore (Memento::MainWindowMemento memento)`
- `Model::Project & getProject ()`

Data Fields

- `Model::Project * loadedProject`

Private Member Functions

- `void newProject ()`
- `void undo ()`
- `void saveAs ()`
- `void loadProject ()`
- `void saveProject ()`
- `void redo ()`
- `void createUi ()`
- `void connectActions ()`
- `void createMenuBar ()`

Private Attributes

- `QStatusBar * statusbar`
- `QMenuBar * menubar_project`
- `QMenuBar * menubar_edit`
- `QAction * action_newProject`
- `QAction * action_undo`
- `QAction * action_saveAs`
- `QAction * action_loadProject`
- `QAction * action_saveProject`
- `QAction * action_redo`
- `QTabWidget * tab_tabs`

4.50.1 Detailed Description

This class is the main window that is shown.

4.50.2 Constructor & Destructor Documentation

4.50.2.1 MainWindow (*GUI::QWidget * parent*)

Constructor.

4.50.3 Member Function Documentation

4.50.3.1 void connectActions () [private]

connect actions in the menuBars to the slots in this class

4.50.3.2 void createMenuBar () [private]

4.50.3.3 void createUi () [private]

creates the UI

4.50.3.4 Memento::MainWindowMemento getMemento ()

Creates a memento which contains the state of the window.

Returns

The created memento.

4.50.3.5 Model::Project & getProject ()

Returns the project that is currently loaded.

Returns

The currently loaded project.

4.50.3.6 void loadProject () [private]

Slot:connected with action_loadProject.triggered() opens QFileDialog and opens project in the selected file if possible

4.50.3.7 void newProject () [private]

slot creates new project, removes existing one

4.50.3.8 void redo () [private]

Slot:connected with action_redo.triggered() redo action if undo has been used

4.50.3.9 void restore (**Memento::MainWindowMemento** *memento*)

Restores the window based on the memento.

Parameters

<i>memento</i>	The memento which contains the state of the window.
----------------	---

4.50.3.10 void saveAs() [private]

Slot:connected with action_saveAs.triggered() Opens QFileDialog and saves project in selected file

4.50.3.11 void saveProject() [private]

Slot:connected with action_saveProject.triggered() opens QFileDialog and saves project in selected file

4.50.3.12 void undo() [private]

Slot:connected with action_undo.triggered() undo last action if possible

4.50.4 Field Documentation**4.50.4.1 QAction* action_loadProject [private]****4.50.4.2 QAction* action_newProject [private]****4.50.4.3 QAction* action_redo [private]****4.50.4.4 QAction* action_saveAs [private]****4.50.4.5 QAction* action_saveProject [private]****4.50.4.6 QAction* action_undo [private]****4.50.4.7 Model::Project* loadedProject****4.50.4.8 QMenuBar* menubar_edit [private]****4.50.4.9 QMenuBar* menubar_project [private]****4.50.4.10 QStatusBar* statusbar [private]****4.50.4.11 QTabWidget* tab_tabs [private]**

4.51 MainWindowMemento Class Reference

Public Member Functions

- `MainWindowMemento ()`
- `int getSelectedTab ()`
- `void setSelectedTab (int selectedTab)`
- `Memento::AnalysisTabMemento getAnalysisTabMemento ()`
- `void setAnalysisTabMemento (Memento::AnalysisTabMemento analysisTabMeMento)`
- `Memento::FilterTabMemento getFilterTabMemento ()`
- `void setFilterTabMemento (Memento::FilterTabMemento filterTabMemento)`

Private Attributes

- `int selectedTab`
- `Memento::AnalysisTabMemento * analysisTab`
- `Memento::FilterTabMemento * filterTab`

4.51.1 Detailed Description

This class is the memento for the MainWindow.

4.51.2 Constructor & Destructor Documentation

4.51.2.1 MainWindowMemento ()

Constructor.

4.51.3 Member Function Documentation

4.51.3.1 Memento::AnalysisTabMemento getAnalysisTabMemento ()

Returns the [AnalysisTabMemento](#).

Returns

The [AnalysisTabMemento](#).

4.51.3.2 Memento::FilterTabMemento getFilterTabMemento ()

Returns the [FilterTabMemento](#).

Returns

The [FilterTabMemento](#).

4.51.3.3 int getSelectedTab ()

Returns the currently selected tab.

Returns

The currently selected tab.

4.51.3.4 void setAnalysisTabMemento (Memento::AnalysisTabMemento *analysisTabMeMento*)

Sets the [AnalysisTabMemento](#).

Parameters

<i>analysisTab</i> ↪ MeMento	The AnalysisTabMemento .
---------------------------------	--

4.51.3.5 void setFilterTabMemento (Memento::FilterTabMemento *filterTabMemento*)

Sets the [FilterTabMemento](#).

Parameters

<i>filterTab</i> ↪ Memento	The FilterTabMemento .
-------------------------------	--

4.51.3.6 void setSelectedTab (int *selectedTab*)

Sets the currently selected tab.

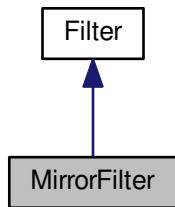
Parameters

<i>selectedTab</i>	The currently selected tab.
--------------------	-----------------------------

4.51.4 Field Documentation**4.51.4.1 Memento::AnalysisTabMemento* analysisTab [private]****4.51.4.2 Memento::FilterTabMemento* filterTab [private]****4.51.4.3 int selectedTab [private]**

4.52 MirrorFilter Class Reference

Inheritance diagram for MirrorFilter:



Public Member Functions

- `MirrorFilter ()`
- string `getFilterDescription ()`
- string `getName ()`
- `Model::MirrorMode getMode ()`
- void `setMode (Model::MirrorMode mode)`

Private Attributes

- `Model::MirrorMode * mode`

Additional Inherited Members

4.52.1 Detailed Description

Mirrors the video horizontally or vertically.

4.52.2 Constructor & Destructor Documentation

4.52.2.1 `MirrorFilter ()`

Constructor.

4.52.3 Member Function Documentation

4.52.3.1 `string getFilterDescription () [virtual]`

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implements [Filter](#).

4.52.3.2 Model::MirrorMode getMode()

Returns the MirrorMode.

Returns

The MirrorMode.

4.52.3.3 string getName() [virtual]

Returns the name of the filter.

Returns

The filtername.

Implements [Filter](#).

4.52.3.4 void setMode(Model::MirrorMode mode)

Sets the MirrorMode.

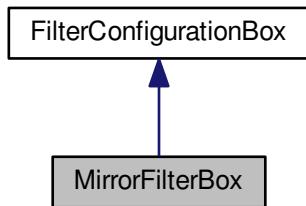
Parameters

<i>mode</i>	The MirrorMode.
-------------	-----------------

4.52.4 Field Documentation**4.52.4.1 Model::MirrorMode* mode [private]**

4.53 MirrorFilterBox Class Reference

Inheritance diagram for MirrorFilterBox:



Public Member Functions

- [MirrorFilterBox \(GUI::QWidget *parent \)](#)

Additional Inherited Members

4.53.1 Detailed Description

This class contains the gui elements for changing the options of a mirror filter.

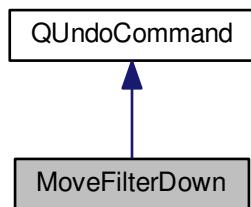
4.53.2 Constructor & Destructor Documentation

4.53.2.1 [MirrorFilterBox \(GUI::QWidget * parent \)](#)

Constructor.

4.54 MoveFilterDown Class Reference

Inheritance diagram for MoveFilterDown:



Public Member Functions

- [MoveFilterDown \(GUI::FilterTab *filterTab, int old, int new_2\)](#)
- void [undo \(\)](#)
- void [redo \(\)](#)

Private Attributes

- int [oldIndex](#)
- int [newIndex](#)
- GUI::FilterTab * [filterTab](#)

4.54.1 Detailed Description

This class is the undo command for moving a filter down in the filterlist.

4.54.2 Constructor & Destructor Documentation

4.54.2.1 MoveFilterDown (GUI::FilterTab * *filterTab*, int *old*, int *new_2*)

Constructor.

Parameters

<i>filterTab</i>	The filtertab to operate on.
<i>old</i>	The old index of the filter.
<i>new</i>	The new index of the filter.

4.54.3 Member Function Documentation

4.54.3.1 void redo ()

Moves selected filter one position down in the filterlist.

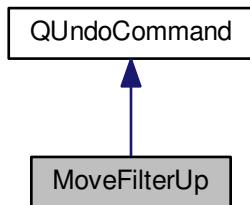
4.54.3.2 void undo ()

Moves the filter one position back up in the filterlist.

4.54.4 Field Documentation**4.54.4.1 GUI::FilterTab* filterTab [private]****4.54.4.2 int newIndex [private]****4.54.4.3 int oldIndex [private]**

4.55 MoveFilterUp Class Reference

Inheritance diagram for MoveFilterUp:



Public Member Functions

- [MoveFilterUp \(GUI::FilterTab *filterTab, int old, int new_1\)](#)
- void [undo \(\)](#)
- void [redo \(\)](#)

Private Attributes

- int [oldIndex](#)
- int [newIndex](#)
- GUI::FilterTab * [filterTab](#)

4.55.1 Detailed Description

This class is the undo command for moving a filter up in the filterlist.

4.55.2 Constructor & Destructor Documentation

4.55.2.1 MoveFilterUp (GUI::FilterTab * *filterTab*, int *old*, int *new_1*)

Constructor.

Parameters

<i>filterTab</i>	The filtertab to operate on.
<i>old</i>	Old index of the filter.
<i>new</i>	New index of the filter.

4.55.3 Member Function Documentation

4.55.3.1 void redo ()

Moves selected filter one position up in the filterlist.

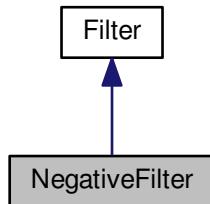
4.55.3.2 void undo ()

Moves the filter one position back down in the filterlist.

4.55.4 Field Documentation**4.55.4.1 GUI::FilterTab* filterTab [private]****4.55.4.2 int newIndex [private]****4.55.4.3 int oldIndex [private]**

4.56 NegativeFilter Class Reference

Inheritance diagram for NegativeFilter:



Public Member Functions

- [NegativeFilter \(\)](#)
- string [getFilterDescription \(\)](#)
- string [getName \(\)](#)

Additional Inherited Members

4.56.1 Detailed Description

Converts the video into it's negative.

4.56.2 Constructor & Destructor Documentation

4.56.2.1 NegativeFilter()

Constructor.

4.56.3 Member Function Documentation

4.56.3.1 string getFilterDescription() [virtual]

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implements [Filter](#).

4.56.3.2 string getName() [virtual]

Returns the name of the filter.

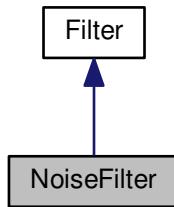
Returns

The filtername.

Implements [Filter](#).

4.57 NoiseFilter Class Reference

Inheritance diagram for NoiseFilter:



Public Member Functions

- `NoiseFilter ()`
- string `getFilterDescription ()`
- `Model::NoiseMode getMode ()`
- void `setMode (Model::NoiseMode mode)`
- int `getIntensity ()`
- string `getName ()`
- void `setIntensity (int intensity)`

Private Attributes

- int `intensity`
- `Model::NoiseMode * mode`

Additional Inherited Members

4.57.1 Detailed Description

Inserts noise into the video.

4.57.2 Constructor & Destructor Documentation

4.57.2.1 `NoiseFilter ()`

Constructor.

4.57.3 Member Function Documentation

4.57.3.1 `string getFilterDescription () [virtual]`

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implements [Filter](#).

4.57.3.2 int getIntensity ()

Returns the intensity of the noise.

Returns

The noise intensity.

4.57.3.3 Model::NoiseMode getMode ()

Returns the NoiseMode.

Returns

The NoiseMode.

4.57.3.4 string getName () [virtual]

Returns the name of the filter.

Returns

The filtername.

Implements [Filter](#).

4.57.3.5 void setIntensity (int *intensity*)

Sets the intensity of the noise.

Parameters

<i>intensity</i>	The new intensity of the noise.
------------------	---------------------------------

4.57.3.6 void setMode (Model::NoiseMode *mode*)

Sets the NoiseMode.

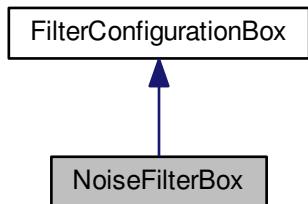
Parameters

<i>mode</i>	The new NoiseMode.
-------------	--------------------

4.57.4 Field Documentation**4.57.4.1 int intensity [private]****4.57.4.2 Model::NoiseMode* mode [private]**

4.58 NoiseFilterBox Class Reference

Inheritance diagram for NoiseFilterBox:



Public Member Functions

- [NoiseFilterBox \(GUI::QWidget *parent \)](#)

Additional Inherited Members

4.58.1 Detailed Description

This class contains the gui elements for changing the options of a noise filter.

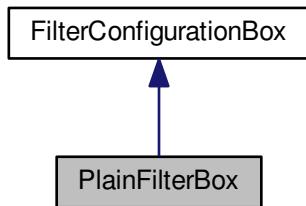
4.58.2 Constructor & Destructor Documentation

4.58.2.1 [NoiseFilterBox \(GUI::QWidget * parent \)](#)

Constructor.

4.59 PlainFilterBox Class Reference

Inheritance diagram for PlainFilterBox:



Public Member Functions

- [PlainFilterBox \(GUI::QWidget *parent\)](#)

Additional Inherited Members

4.59.1 Detailed Description

This class contains no sliders to adjust filter options.

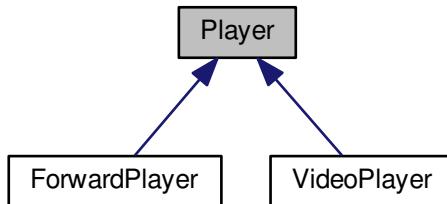
4.59.2 Constructor & Destructor Documentation

4.59.2.1 PlainFilterBox (GUI::QWidget * *parent*)

Constructor.

4.60 Player Class Reference

Inheritance diagram for Player:



Public Member Functions

- virtual void [play \(\)=0](#)
- virtual void [pause \(\)=0](#)
- virtual void [stop \(\)=0](#)
- virtual void [nextFrame \(\)=0](#)
- virtual void [previousFrame \(\)=0](#)
- virtual void [setSpeed \(float speed\)=0](#)
- virtual void [setPosition \(int position\)=0](#)
- virtual int [getPosition \(\)=0](#)
- virtual float [getSpeed \(\)=0](#)
- virtual bool [isPlaying \(\)=0](#)
- virtual bool [isStopped \(\)=0](#)
- virtual void [reset \(\)=0](#)

Data Fields

- [GUI::ControlPanel * players](#)
- [GUI::ControlPanel * masterPlayer](#)

4.60.1 Detailed Description

This class is the base class for players.

4.60.2 Member Function Documentation

4.60.2.1 `int getPosition () [pure virtual]`

Returns the position in the video.

Returns

The current position.

Implemented in [VideoPlayer](#), and [ForwardPlayer](#).

4.60.2.2 `float getSpeed() [pure virtual]`

Returns the speed.

Returns

The current speed.

Implemented in [VideoPlayer](#), and [ForwardPlayer](#).

4.60.2.3 `bool isPlaying() [pure virtual]`

Whether the player is currently playing.

Returns

True if the player is playing.

Implemented in [VideoPlayer](#), and [ForwardPlayer](#).

4.60.2.4 `bool isStopped() [pure virtual]`

Whether the player is stopped.

Returns

True if the player is stopped.

Implemented in [VideoPlayer](#), and [ForwardPlayer](#).

4.60.2.5 `void nextFrame() [pure virtual]`

Shows the next frame.

Implemented in [VideoPlayer](#), and [ForwardPlayer](#).

4.60.2.6 `void pause() [pure virtual]`

Pauses the video.

Implemented in [VideoPlayer](#), and [ForwardPlayer](#).

4.60.2.7 `void play() [pure virtual]`

Plays the video.

Implemented in [VideoPlayer](#), and [ForwardPlayer](#).

4.60.2.8 `void previousFrame() [pure virtual]`

Shows the previous frame.

Implemented in [VideoPlayer](#), and [ForwardPlayer](#).

4.60.2.9 `void reset() [pure virtual]`

Resets the player.

Implemented in [VideoPlayer](#), and [ForwardPlayer](#).

4.60.2.10 `void setPosition(int position) [pure virtual]`

Sets the position in the video.

Parameters

<code>position</code>	The new position.
-----------------------	-------------------

Implemented in [VideoPlayer](#), and [ForwardPlayer](#).

4.60.2.11 `void setSpeed(float speed) [pure virtual]`

Sets the speed.

Parameters

<code>speed</code>	The new speed.
--------------------	----------------

Implemented in [VideoPlayer](#), and [ForwardPlayer](#).

4.60.2.12 `void stop() [pure virtual]`

Stops the video.

Implemented in [VideoPlayer](#), and [ForwardPlayer](#).

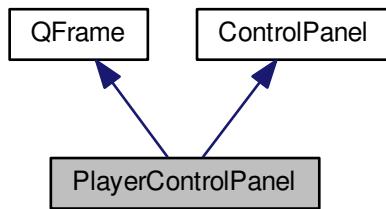
4.60.3 Field Documentation

4.60.3.1 `GUI::ControlPanel* masterPlayer`

4.60.3.2 `GUI::ControlPanel* players`

4.61 PlayerControlPanel Class Reference

Inheritance diagram for PlayerControlPanel:



Public Member Functions

- [PlayerControlPanel \(QWidget *parent=0\)](#)
- [void updateUi \(\)](#)

Private Member Functions

- [void createUi \(\)](#)
- [void createActions \(\)](#)
- [void play \(\)](#)
- [void pause \(\)](#)
- [void stop \(\)](#)
- [void nextFrame \(\)](#)
- [void previousFrame \(\)](#)
- [void changeSpeed \(int index\)](#)
- [void changeTimeline \(int value\)](#)

Private Attributes

- [QPushButton * button_play](#)
- [QPushButton * button_stop](#)
- [QPushButton * button_nextFrame](#)
- [QPushButton * button_previousFrame](#)
- [GUI::QComboBox * comboBox_speed](#)
- [QSlider * slider_timeline](#)

Additional Inherited Members

4.61.1 Detailed Description

This class is the control panel to play videos.

4.61.2 Constructor & Destructor Documentation

4.61.2.1 `PlayerControlPanel (QWidget * parent = 0)`

Constructor.

4.61.3 Member Function Documentation

4.61.3.1 `void changeSpeed (int index) [private]`

Slot for comboBox_speed.currentIndexChanged(int) signal.

4.61.3.2 `void changeTimeline (int value) [private]`

Slot for slider_timeLine.valueChanged(int) signal.

4.61.3.3 `void createActions () [private]`

Creates the actions.

4.61.3.4 `void createUi () [private]`

Creates the ui.

4.61.3.5 `void nextFrame () [private]`

Slot for button_nextFrame.clicked() signal.

4.61.3.6 `void pause () [private]`

Slot for button_pause.clicked() signal.

4.61.3.7 `void play () [private]`

Slot for the button_play.clicked() signal.

4.61.3.8 `void previousFrame () [private]`

Slot for button_previousFrame.clicked() signal.

4.61.3.9 `void stop () [private]`

Slot for button_stop.clicked() signal.

4.61.3.10 `void updateUi () [virtual]`

Updates the ui of the control panel.

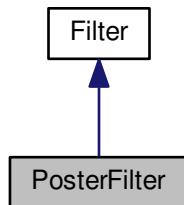
Implements [ControlPanel](#).

4.61.4 Field Documentation

- 4.61.4.1 `QPushButton* button_nextFrame [private]`
- 4.61.4.2 `QPushButton* button_play [private]`
- 4.61.4.3 `QPushButton* button_previousFrame [private]`
- 4.61.4.4 `QPushButton* button_stop [private]`
- 4.61.4.5 `GUI::QComboBox* comboBox_speed [private]`
- 4.61.4.6 `QSlider* slider_timeline [private]`

4.62 PosterFilter Class Reference

Inheritance diagram for PosterFilter:



Public Member Functions

- `PosterFilter ()`
- string `getFilterDescription ()`
- int `getNumberOfColors ()`
- string `getName ()`
- void `setNumberOfColors (int numberOfColors)`

Private Attributes

- int `numberOfColors`

Additional Inherited Members

4.62.1 Detailed Description

Reduces the maximum number of colors in the video.

4.62.2 Constructor & Destructor Documentation

4.62.2.1 `PosterFilter ()`

Constructor.

4.62.3 Member Function Documentation

4.62.3.1 `string getFilterDescription () [virtual]`

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implements [Filter](#).

4.62.3.2 string getName() [virtual]

Returns the name of the filter.

Returns

The filtername.

Implements [Filter](#).

4.62.3.3 int getNumberOfColors()

Returns the maximum number of colors.

Returns

Maximum number of colors.

4.62.3.4 void setNumberOfColors(int *numberOfColors*)

Sets the maximum number of colors.

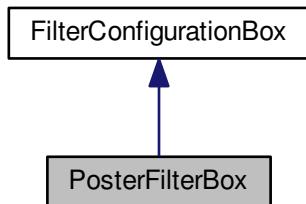
Parameters

<i>numberOfColors</i>	Maximum number of colors.
-----------------------	---------------------------

4.62.4 Field Documentation**4.62.4.1 int *numberOfColors* [private]**

4.63 PosterFilterBox Class Reference

Inheritance diagram for PosterFilterBox:



Public Member Functions

- [PosterFilterBox \(GUI::QWidget *parent\)](#)

Additional Inherited Members

4.63.1 Detailed Description

This class contains the gui elements for changing the options of a poster filter.

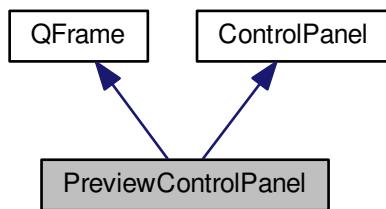
4.63.2 Constructor & Destructor Documentation

4.63.2.1 [PosterFilterBox \(GUI::QWidget * parent \)](#)

Constructor.

4.64 PreviewControlPanel Class Reference

Inheritance diagram for PreviewControlPanel:



Public Member Functions

- `PreviewControlPanel (GUI::QWidget *parent=0)`
- `void updateUi ()`

Private Member Functions

- `void createUi ()`
- `void createActions ()`
- `void nextFrame ()`
- `void previousFrame ()`
- `void updateLabel ()`

Private Attributes

- `QPushButton *button_nextFrame`
- `QPushButton *button_previousFrame`
- `QLabel *label_position`

Additional Inherited Members

4.64.1 Detailed Description

This class is the control panel for the frame preview.

4.64.2 Constructor & Destructor Documentation

4.64.2.1 `PreviewControlPanel (GUI::QWidget * parent = 0)`

Constructor.

4.64.3 Member Function Documentation

4.64.3.1 `void createActions() [private]`

Creates the actions.

4.64.3.2 `void createUi() [private]`

Creates the ui.

4.64.3.3 `void nextFrame() [private]`

Slot for `button_nextFrame.clicked()` signal.

4.64.3.4 `void previousFrame() [private]`

Slot for `button_previousFrame.clicked()` signal.

4.64.3.5 `void updateLabel() [private]`

Updates the label that shows the current position.

4.64.3.6 `void updateUi() [virtual]`

Updates the ui of the control panel.

Implements [ControlPanel](#).

4.64.4 Field Documentation

4.64.4.1 `QPushButton* button_nextFrame [private]`

4.64.4.2 `QPushButton* button_previousFrame [private]`

4.64.4.3 `QLabel* label_position [private]`

4.65 Project Class Reference

Public Member Functions

- `Project (QString name)`
- `QString getName ()`
- `Memento::MainWindowMemento & getMemento ()`
- `void setMemento (Memento::MainWindowMemento memento)`
- `void setPath (QString path)`
- `QString getPath ()`

Private Attributes

- `QString name`
- `QString path`
- `GUI::MainWindow * loadedProject`
- `Memento::MainWindowMemento * mainWindowMemento`

4.65.1 Detailed Description

This class contains the different mementos.

4.65.2 Constructor & Destructor Documentation

4.65.2.1 `Project (QString name)`

Constructor.

Parameters

<code>name</code>	Name of the project.
-------------------	----------------------

4.65.3 Member Function Documentation

4.65.3.1 `Memento::MainWindowMemento & getMemento ()`

Returns the MainWindowMemento.

Returns

The MainWindowMemento.

4.65.3.2 `QString getName ()`

Returns the name of the project.

Returns

Name of the project.

4.65.3.3 `QString getPath()`

Returns the project save path.

Returns

The project save path.

4.65.3.4 `void setMemento (Memento::MainWindowMemento memento)`

Sets the MainWindowMemento.

Parameters

<i>memento</i>	The MainWindowMemento.
----------------	------------------------

4.65.3.5 `void setPath (QString path)`

Sets the path at which the project is saved.

Parameters

<i>path</i>	The project save path.
-------------	------------------------

4.65.4 Field Documentation**4.65.4.1 `GUI::MainWindow* loadedProject` [private]****4.65.4.2 `Memento::MainWindowMemento* mainWindowMemento` [private]****4.65.4.3 `QString name` [private]****4.65.4.4 `QString path` [private]**

4.66 ProjectReader Class Reference

Public Member Functions

- [ProjectReader \(QString path\)](#)
- [Model::Project readProject \(\)](#)

Private Attributes

- [QString path](#)

4.66.1 Detailed Description

This class can read a project from a file.

4.66.2 Constructor & Destructor Documentation

4.66.2.1 ProjectReader (QString path)

Constructor.

Parameters

<i>path</i>	The absolute path to the project file.
-------------	--

4.66.3 Member Function Documentation

4.66.3.1 Model::Project readProject ()

Reads a project from a file.

Returns

The loaded project.

4.66.4 Field Documentation

4.66.4.1 QString path [private]

4.67 ProjectWriter Class Reference

Public Member Functions

- `ProjectWriter (Model::Project p)`
- `void saveProject ()`
- `void saveResults ()`

4.67.1 Detailed Description

This class can write the project files and the results.

4.67.2 Constructor & Destructor Documentation

4.67.2.1 `ProjectWriter (Model::Project p)`

Constructor.

Parameters

<code>p</code>	The project to save.
----------------	----------------------

4.67.3 Member Function Documentation

4.67.3.1 `void saveProject ()`

Saves the whole project.

4.67.3.2 `void saveResults ()`

Saves on the analysis results.

4.68 PsnrCalculator Class Reference

Public Member Functions

- [PsnrCalculator \(Model::AVVideo &referenceVideo, Model::AVVideo &compareVideo\)](#)
- [Model::Graph calculate \(\)](#)

Private Member Functions

- [void init \(\)](#)

Private Attributes

- [Model::AVVideo * referenceVideo](#)
- [Model::AVVideo * video](#)

4.68.1 Detailed Description

This class calculates the psnr-graph of a video.

4.68.2 Constructor & Destructor Documentation

4.68.2.1 [PsnrCalculator \(Model::AVVideo & referenceVideo, Model::AVVideo & compareVideo \)](#)

Constructor.

Parameters

referenceVideo	The reference video for the psnr calculation.
compareVideo	The video that is compared to the reference video.

4.68.3 Member Function Documentation

4.68.3.1 [Model::Graph calculate \(\)](#)

Calculates the psnr graph.

Returns

The calculated psnr graph.

4.68.3.2 [void init \(\) \[private\]](#)

Initializes the ffmpeg psnr filter.

4.68.4 Field Documentation

4.68.4.1 [Model::AVVideo* referenceVideo \[private\]](#)

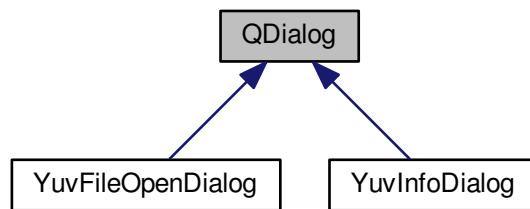
4.68.4.2 [Model::AVVideo* video \[private\]](#)

4.69 QCheckBox Class Reference

4.70 QComboBox Class Reference

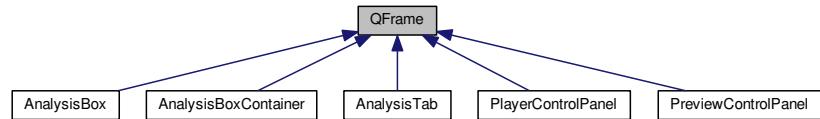
4.71 QDialog Class Reference

Inheritance diagram for QDialog:



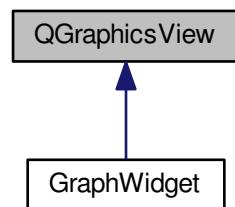
4.72 QFrame Class Reference

Inheritance diagram for QFrame:



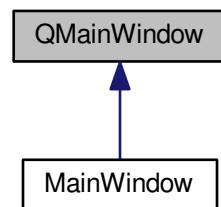
4.73 QGraphicsView Class Reference

Inheritance diagram for QGraphicsView:



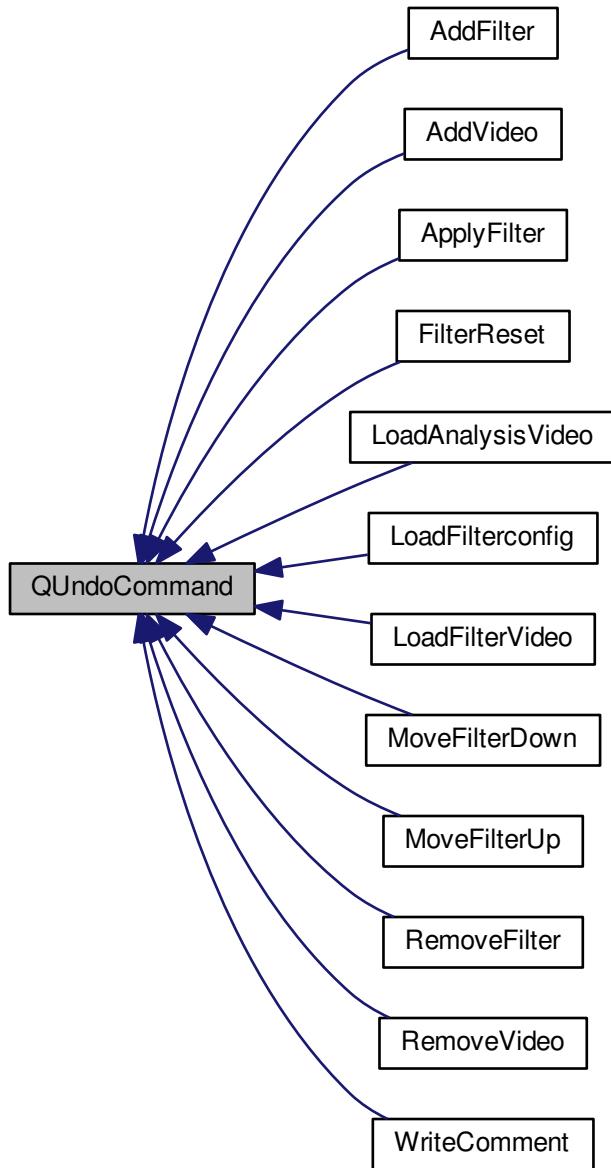
4.74 QMainWindow Class Reference

Inheritance diagram for QMainWindow:



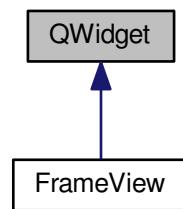
4.75 QUndoCommand Class Reference

Inheritance diagram for QUndoCommand:



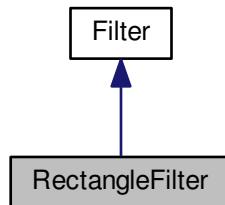
4.76 QWidget Class Reference

Inheritance diagram for QWidget:



4.77 RectangleFilter Class Reference

Inheritance diagram for RectangleFilter:



Public Member Functions

- `RectangleFilter ()`
- string `getFilterDescription ()`
- QRgb `getColor ()`
- void `setColor (QRgb color)`
- int `getWidth ()`
- void `setWidth (int width)`
- int `getHeight ()`
- void `setHeight (int height)`
- int `getX ()`
- string `getName ()`
- void `setX (int x)`
- int `getY ()`
- void `setY (int y)`
- int `getOpacity ()`
- void `setOpacity (int opacity)`

Private Attributes

- QRgb `color`
- int `width`
- int `height`
- int `x`
- int `y`
- int `opacity`

Additional Inherited Members

4.77.1 Detailed Description

Inserts a filled rectangle with a given color into the video

4.77.2 Constructor & Destructor Documentation

4.77.2.1 RectangleFilter()

Constructor.

4.77.3 Member Function Documentation

4.77.3.1 QRgb getColor()

Returns the color of the rectangle.

Returns

The color of the rectangle.

4.77.3.2 string getFilterDescription() [virtual]

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implements [Filter](#).

4.77.3.3 int getHeight()

Returns the height of the rectangle.

Returns

The height of the rectangle.

4.77.3.4 string getName() [virtual]

Returns the name of the filter.

Returns

The filtername.

Implements [Filter](#).

4.77.3.5 int getOpacity()

Returns the opacity of the rectangle.

Returns

The opacity of the rectangle.

4.77.3.6 int getWidth ()

Returns the width of the rectangle.

Returns

The width of the rectangle.

4.77.3.7 int getX ()

Returns the start position on the x axis.

Returns

The start position on the x axis.

4.77.3.8 int getY ()

Returns the start position on the y axis.

Returns

The start position on the y axis.

4.77.3.9 void setColor (QRgb color)

Sets the color of the rectangle.

Parameters

<i>color</i>	The new color of the rectangle.
--------------	---------------------------------

4.77.3.10 void setHeight (int height)

Sets the height of the rectangle.

Parameters

<i>height</i>	The new height of the rectangle.
---------------	----------------------------------

4.77.3.11 void setOpacity (int opacity)

Sets the opacity of the rectangle.

Parameters

<i>opacity</i>	The new opacity of the rectangle.
----------------	-----------------------------------

4.77.3.12 void setWidth (int width)

Sets the width of the rectangle.

Parameters

<i>width</i>	The new width of the rectangle.
--------------	---------------------------------

4.77.3.13 void setX(int x)

Sets the start position on the x axis.

Parameters

<i>x</i>	The new start position on the x axis.
----------	---------------------------------------

4.77.3.14 void setY(int y)

Sets the start position on the y axis.

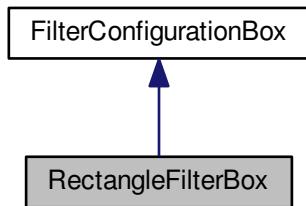
Parameters

<i>y</i>	The new start position on the y axis.
----------	---------------------------------------

4.77.4 Field Documentation**4.77.4.1 QRgb color [private]****4.77.4.2 int height [private]****4.77.4.3 int opacity [private]****4.77.4.4 int width [private]****4.77.4.5 int x [private]****4.77.4.6 int y [private]**

4.78 RectangleFilterBox Class Reference

Inheritance diagram for RectangleFilterBox:



Public Member Functions

- [RectangleFilterBox \(GUI::QWidget *parent\)](#)

Additional Inherited Members

4.78.1 Detailed Description

This class contains the gui elements for changing the options of a rectangle filter.

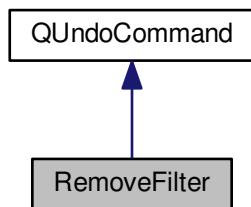
4.78.2 Constructor & Destructor Documentation

4.78.2.1 [RectangleFilterBox \(GUI::QWidget * parent \)](#)

Constructor.

4.79 RemoveFilter Class Reference

Inheritance diagram for RemoveFilter:



Public Member Functions

- `RemoveFilter (GUI::FilterTab *filterTab, Model::Filter filter, int index)`
- `void undo ()`
- `void redo ()`

Private Attributes

- `int index`
- `GUI::FilterTab * filterTab`
- `Model::Filter * filter`

4.79.1 Detailed Description

This class is the undo command for removing a filter in the filterlist on the filtertab.

4.79.2 Constructor & Destructor Documentation

4.79.2.1 RemoveFilter (`GUI::FilterTab * filterTab, Model::Filter filter, int index`)

Constructor.

Parameters

<code>filterTab</code>	The filtertab to operate on.
<code>filter</code>	The filter to remove.
<code>index</code>	The current index of the filter to remove.

4.79.3 Member Function Documentation

4.79.3.1 void redo ()

Removes a filter from the filterlist.

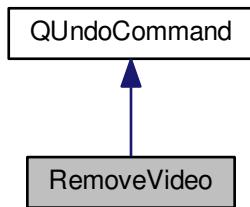
4.79.3.2 void undo()

Adds the removed filter back into the filterlist.

4.79.4 Field Documentation**4.79.4.1 Model::Filter* filter [private]****4.79.4.2 GUI::FilterTab* filterTab [private]****4.79.4.3 int index [private]**

4.80 RemoveVideo Class Reference

Inheritance diagram for RemoveVideo:



Public Member Functions

- `RemoveVideo (GUI::AnalysisBoxContainer *container, encodedVideo video)`
- `void undo ()`
- `void redo ()`

Private Attributes

- `GUI::AnalysisBox * anaBox`
- `GUI::AnalysisBoxContainer * anaBoxContainer`
- `Model::EncodedVideo * video`

4.80.1 Detailed Description

This class is the undo command for removing a encoded video in the analysis tab.

4.80.2 Constructor & Destructor Documentation

4.80.2.1 RemoveVideo (`GUI::AnalysisBoxContainer * container, encodedVideo video`)

Constructor.

Parameters

<code>container</code>	The AnalysisBoxContainer to operate on.
<code>video</code>	The video to remove.

4.80.3 Member Function Documentation

4.80.3.1 void redo ()

Removes a video from the analysis tab.

4.80.3.2 void undo()

Re adds the removed video to the analysis tab.

4.80.4 Field Documentation**4.80.4.1 GUI::AnalysisBox* anaBox [private]****4.80.4.2 GUI::AnalysisBoxContainer* anaBoxContainer [private]****4.80.4.3 Model::EncodedVideo* video [private]**

4.81 RGBDifferenceCalculator Class Reference

Public Member Functions

- `RGBDifferenceCalculator (GUI::Video &referenceVideo, GUI::Video &video)`
- `void calculateVideo (GUI::Video &target)`

Private Attributes

- `GUI::Video * referenceVideo`
- `GUI::Video * video`

4.81.1 Detailed Description

This class calculates the RGB-difference video of a video.

4.81.2 Constructor & Destructor Documentation

4.81.2.1 `RGBDifferenceCalculator (GUI::Video &referenceVideo, GUI::Video & video)`

Constructor.

Parameters

<code>referenceVideo</code>	The reference video.
<code>video</code>	The video that is compared to the reference video.

4.81.3 Member Function Documentation

4.81.3.1 `void calculateVideo (GUI::Video & target)`

Calculates the RGB difference between two videos.

Parameters

<code>target</code>	The video the calculated frames are added to.
---------------------	---

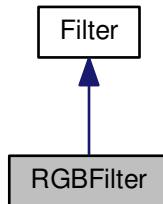
4.81.4 Field Documentation

4.81.4.1 `GUI::Video* referenceVideo [private]`

4.81.4.2 `GUI::Video* video [private]`

4.82 RGBFilter Class Reference

Inheritance diagram for RGBFilter:



Public Member Functions

- [RGBFilter \(\)](#)
- [string getFilterDescription \(\)](#)
- [Model::BasicColor getColor \(\)](#)
- [void setColor \(Model::BasicColor color\)](#)
- [string getName \(\)](#)

Private Attributes

- [Model::BasicColor * color](#)

Additional Inherited Members

4.82.1 Detailed Description

Filters the video by a given channel (red, green or blue).

4.82.2 Constructor & Destructor Documentation

4.82.2.1 RGBFilter()

Constructor.

4.82.3 Member Function Documentation

4.82.3.1 Model::BasicColor getColor()

Returns the color that is not filtered out.

Returns

The preserved color.

4.82.3.2 string getFilterDescription() [virtual]

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implements [Filter](#).

4.82.3.3 string getName() [virtual]

Returns the name of the filter.

Returns

The filtername.

Implements [Filter](#).

4.82.3.4 void setColor(Model::BasicColor *color*)

Sets the preserved color.

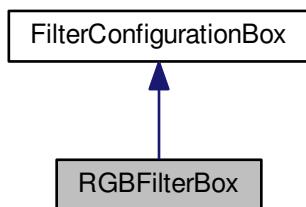
Parameters

<i>color</i>	The preserved color.
--------------	----------------------

4.82.4 Field Documentation**4.82.4.1 Model::BasicColor* *color* [private]**

4.83 RGBFilterBox Class Reference

Inheritance diagram for RGBFilterBox:



Public Member Functions

- [RGBFilterBox \(GUI::QWidget *parent\)](#)

Additional Inherited Members

4.83.1 Detailed Description

This class contains the gui elements for changing the options of a rgb filter.

4.83.2 Constructor & Destructor Documentation

4.83.2.1 RGBFilterBox (GUI::QWidget * *parent*)

Constructor.

4.84 RGBHistogrammCalculator Class Reference

Public Member Functions

- void [RGBHistogrammCalculator \(GUI::Video &video\)](#)
- void [calculate \(\)](#)
- [Model::Graph getRedHistogramm \(\)](#)
- [Model::Graph getGreenHistogramm \(\)](#)
- [Model::Graph getBlueHistogramm \(\)](#)

Private Attributes

- [GUI::Video * video](#)
- [Model::Graph redHistogramm](#)
- [Model::Graph greenHistogramm](#)
- [Model::Graph blueHistogramm](#)

4.84.1 Detailed Description

This class calculates the RGB histogramm for a video.

4.84.2 Member Function Documentation

4.84.2.1 void calculate ()

Calculates the red, green and blue components of a video.

4.84.2.2 Model::Graph getBlueHistogramm ()

Returns the blue components of a video.

4.84.2.3 Model::Graph getGreenHistogramm ()

Returns the green components of a video.

4.84.2.4 Model::Graph getRedHistogramm ()

Returns the red components of a video.

4.84.2.5 void RGBHistogrammCalculator (GUI::Video & video)

Constructor.

Parameters

<code>video</code>	the video that is analyzed.
--------------------	-----------------------------

4.84.3 Field Documentation

4.84.3.1 **Model::Graph blueHistogramm** [private]

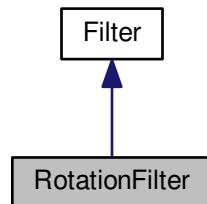
4.84.3.2 **Model::Graph greenHistogramm** [private]

4.84.3.3 **Model::Graph redHistogramm** [private]

4.84.3.4 **GUI::Video* video** [private]

4.85 RotationFilter Class Reference

Inheritance diagram for RotationFilter:



Public Member Functions

- [RotationFilter \(\)](#)
- string [getFilterDescription \(\)](#)
- int [getAngle \(\)](#)
- string [getName \(\)](#)
- void [setAngle \(int angle\)](#)

Private Attributes

- int [angle](#)

Additional Inherited Members

4.85.1 Detailed Description

Rotates the video.

4.85.2 Constructor & Destructor Documentation

4.85.2.1 [RotationFilter\(\)](#)

Constructor.

4.85.3 Member Function Documentation

4.85.3.1 [int getAngle\(\)](#)

Returns the angle of the rotation.

Returns

The rotation angle.

4.85.3.2 string getFilterDescription() [virtual]

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implements [Filter](#).

4.85.3.3 string getName() [virtual]

Returns the name of the filter.

Returns

The filtername.

Implements [Filter](#).

4.85.3.4 void setAngle(int *angle*)

Sets the angle of the rotation.

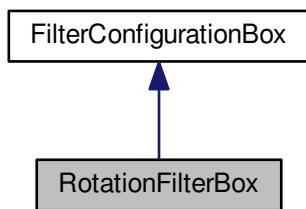
Parameters

<i>angle</i>	The new rotation angle.
--------------	-------------------------

4.85.4 Field Documentation**4.85.4.1 int angle [private]**

4.86 RotationFilterBox Class Reference

Inheritance diagram for RotationFilterBox:



Public Member Functions

- [RotationFilterBox \(GUI::QWidget *parent\)](#)

Additional Inherited Members

4.86.1 Detailed Description

This class contains the gui elements for changing the options of a rotation filter.

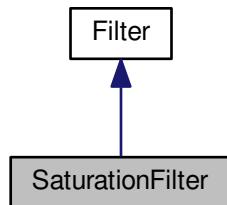
4.86.2 Constructor & Destructor Documentation

4.86.2.1 `RotationFilterBox (GUI::QWidget * parent)`

Constructor.

4.87 SaturationFilter Class Reference

Inheritance diagram for SaturationFilter:



Public Member Functions

- `SaturationFilter ()`
- string `getFilterDescription ()`
- int `getIntensity ()`
- string `getName ()`
- void `setIntensity (int intensity)`

Private Attributes

- int `intensity`

Additional Inherited Members

4.87.1 Detailed Description

Adjusts the saturation of the video.

4.87.2 Constructor & Destructor Documentation

4.87.2.1 `SaturationFilter ()`

Constructor.

4.87.3 Member Function Documentation

4.87.3.1 `string getFilterDescription () [virtual]`

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implements [Filter](#).

4.87.3.2 int getIntensity ()

Returns the intensity of the saturation.

Returns

The intensity.

4.87.3.3 string getName () [virtual]

Returns the name of the filter.

Returns

The filtername.

Implements [Filter](#).

4.87.3.4 void setIntensity (int *intensity*)

Sets the intensity of the saturation.

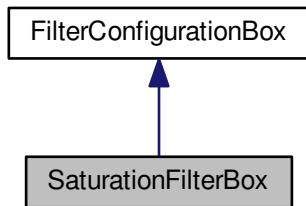
Parameters

<i>intensity</i>	The new intensity,
------------------	--------------------

4.87.4 Field Documentation**4.87.4.1 int intensity [private]**

4.88 SaturationFilterBox Class Reference

Inheritance diagram for SaturationFilterBox:



Public Member Functions

- [SaturationFilterBox \(GUI::QWidget *parent\)](#)

Additional Inherited Members

4.88.1 Detailed Description

This class contains the gui elements for changing the options of a saturation filter.

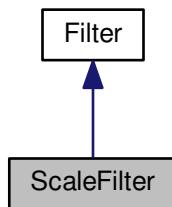
4.88.2 Constructor & Destructor Documentation

4.88.2.1 SaturationFilterBox (GUI::QWidget * *parent*)

Constructor.

4.89 ScaleFilter Class Reference

Inheritance diagram for ScaleFilter:



Public Member Functions

- `ScaleFilter ()`
- string `getFilterDescription ()`
- bool `getKeepRatio ()`
- void `setKeepRatio (bool keepRatio)`
- string `getName ()`
- int `getWidth ()`
- void `setWidth (int width)`
- int `getHeight ()`
- void `setHeight (int height)`
- int `getRatio ()`
- void `setRatio (int ratio)`

Private Attributes

- bool `keepRatio`
- int `width`
- int `height`
- int `ratio`

Additional Inherited Members

4.89.1 Detailed Description

Scales the video.

4.89.2 Constructor & Destructor Documentation

4.89.2.1 `ScaleFilter()`

Constructor.

4.89.3 Member Function Documentation

4.89.3.1 string getFilterDescription() [virtual]

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implements [Filter](#).

4.89.3.2 int getHeight()

Returns the new height.

Returns

The new height.

4.89.3.3 bool getKeepRatio()

Whether the ration is preserved.

Returns

True if the ration is preserved.

4.89.3.4 string getName() [virtual]

Returns the name of the filter.

Returns

The filtername.

Implements [Filter](#).

4.89.3.5 int getRatio()

Returns the ratio of the scaling.

Returns

The ration.

4.89.3.6 int getWidth()

Returns the new width.

Returns

The new width.

4.89.3.7 void setHeight(int *height*)

Sets the new height.

Parameters

<i>height</i>	The new height.
---------------	-----------------

4.89.3.8 void setKeepRatio (bool *keepRatio*)

Sets whether the ration is preserved.

Parameters

<i>keepRatio</i>	True if the ration is preserved.
------------------	----------------------------------

4.89.3.9 void setRatio (int *ratio*)

Sets the ration of the scaling.

Parameters

<i>ratio</i>	The ration.
--------------	-------------

4.89.3.10 void setWidth (int *width*)

Sets the new width,

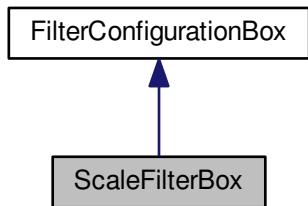
Parameters

<i>width</i>	The new width.
--------------	----------------

4.89.4 Field Documentation**4.89.4.1 int height [private]****4.89.4.2 bool keepRatio [private]****4.89.4.3 int ratio [private]****4.89.4.4 int width [private]**

4.90 ScaleFilterBox Class Reference

Inheritance diagram for ScaleFilterBox:



Public Member Functions

- [ScaleFilterBox \(GUI::QWidget *parent \)](#)

Additional Inherited Members

4.90.1 Detailed Description

This class contains the gui elements for changing the options of a scale filter.

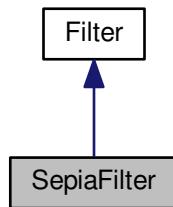
4.90.2 Constructor & Destructor Documentation

4.90.2.1 ScaleFilterBox (GUI::QWidget * *parent*)

Constructor.

4.91 SepiaFilter Class Reference

Inheritance diagram for SepiaFilter:



Public Member Functions

- [SepiaFilter \(\)](#)
- string [getName \(\)](#)
- string [getFilterDescription \(\)](#)

Additional Inherited Members

4.91.1 Detailed Description

Converts the video into sepia.

4.91.2 Constructor & Destructor Documentation

4.91.2.1 [SepiaFilter \(\)](#)

4.91.3 Member Function Documentation

4.91.3.1 [string getFilterDescription \(\) \[virtual\]](#)

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implements [Filter](#).

4.91.3.2 [string getName \(\) \[virtual\]](#)

Returns the name of the filter.

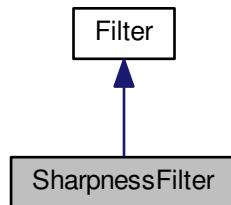
Returns

The filtername.

Implements [Filter](#).

4.92 SharpnessFilter Class Reference

Inheritance diagram for SharpnessFilter:



Public Member Functions

- `SharpnessFilter ()`
- string `getFilterDescription ()`
- int `getIntensity ()`
- string `getName ()`
- void `setIntensity (int intensity)`

Private Attributes

- int `intensity`

Additional Inherited Members

4.92.1 Detailed Description

Sharpens the video.

4.92.2 Constructor & Destructor Documentation

4.92.2.1 `SharpnessFilter()`

Constructor.

4.92.3 Member Function Documentation

4.92.3.1 `string getFilterDescription() [virtual]`

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implements [Filter](#).

4.92.3.2 int getIntensity ()

Returns the intensity of the sharpness.

Returns

The intensity.

4.92.3.3 string getName () [virtual]

Returns the name of the filter.

Returns

The filtername.

Implements [Filter](#).

4.92.3.4 void setIntensity (int *intensity*)

Sets the intensity of the sharpness.

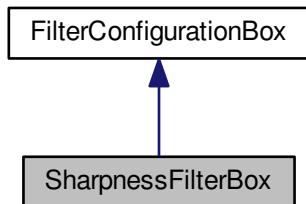
Parameters

<i>intensity</i>	The new intensity.
------------------	--------------------

4.92.4 Field Documentation**4.92.4.1 int intensity [private]**

4.93 SharpnessFilterBox Class Reference

Inheritance diagram for SharpnessFilterBox:



Public Member Functions

- [SharpnessFilterBox \(GUI::QWidget *parent\)](#)

Additional Inherited Members

4.93.1 Detailed Description

This class contains the gui elements for changing the options of a sharpness filter.

4.93.2 Constructor & Destructor Documentation

4.93.2.1 SharpnessFilterBox (GUI::QWidget * *parent*)

Constructor.

4.94 Timer Class Reference

Public Member Functions

- `Timer (int fps)`
- `void setFps (int fps)`
- `void setSpeed (float speed)`
- `float getSpeed ()`
- `int getFps ()`
- `void pause ()`
- `void start ()`
- `void addPlayer (GUI::VideoPlayer &player)`
- `bool isPlaying ()`
- `void removePlayer (GUI::VideoPlayer &player)`

Private Member Functions

- `void update ()`

Private Attributes

- `QTimer timer`
- `float speed`
- `int fps`

4.94.1 Detailed Description

This class is the timer for the video player. It handles the switching of the frames according to fps and speed.

4.94.2 Constructor & Destructor Documentation

4.94.2.1 `Timer (int fps)`

Constructor.

Parameters

<code>fps</code>	The fps to play at.
------------------	---------------------

4.94.3 Member Function Documentation

4.94.3.1 `void addPlayer (GUI::VideoPlayer & player)`

Adds a player.

Parameters

<code>player</code>	The player to add.
---------------------	--------------------

4.94.3.2 int getFps ()

Returns the current fps the timer plays at.

Returns

The current fps.

4.94.3.3 float getSpeed ()

Returns the current speed the timer plays at.

Returns

The current speed.

4.94.3.4 bool isPlaying ()

Whether the timer currently switches frames.

Returns

true if the timer currently switches frames.

4.94.3.5 void pause ()

Stops the timer from switching frames.

4.94.3.6 void removePlayer (GUI::VideoPlayer & player)

Removes a player from the list.

Parameters

<i>player</i>	The player to remove.
---------------	-----------------------

4.94.3.7 void setFps (int fps)

Sets the fps for the timer.

Parameters

<i>fps</i>	The new fps.
------------	--------------

4.94.3.8 void setSpeed (float speed)

Sets the speed to play at. The default value is 1.0.

Parameters

<i>speed</i>	The new speed.
--------------	----------------

4.94.3.9 void start()

Tells the timer to start switching frames.

4.94.3.10 void update() [private]

This method switched the frames in all players.

4.94.4 Field Documentation

4.94.4.1 int fps [private]

4.94.4.2 float speed [private]

4.94.4.3 QTimer timer [private]

4.95 UndoStack Class Reference

Static Public Member Functions

- static QUndoStack & [getUndoStack \(\)](#)

Private Member Functions

- [UndoStack \(\)](#)

Static Private Attributes

- static QUndoStack [undoStack](#)

4.95.1 Detailed Description

This class holds the stack that all undo commands are stacked on as a singleton.

4.95.2 Constructor & Destructor Documentation

4.95.2.1 [UndoStack \(\)](#) [private]

Constructor.

4.95.3 Member Function Documentation

4.95.3.1 static QUndoStack& [getUndoStack \(\)](#) [inline], [static]

Returns the undo stack to operate on.

Returns

The undo stack.

4.95.4 Field Documentation

4.95.4.1 QUndoStack [undoStack](#) [static], [private]

4.96 Video Class Reference

Public Member Functions

- `Video (int fps, int width, int height)`
- `int getWidth ()`
- `int getHeight ()`
- `int getFps ()`
- `QImage * getFrame (int index)`
- `void insertFrame (int index=-1, unique_ptr< QImage > frame)`
- `void removeFrame (int index)`
- `void insertFrames (int index=-1, vector< unique_ptr< QImage > > &frames)`
- `int getNumberOfFrames ()`

Data Fields

- `Model::EncodedVideo * displayVideo`
- `Model::EncodedVideo * macroblockVideo`
- `Model::EncodedVideo * rgbDiffVideo`

Private Attributes

- `std::vector< std::unique_ptr< QImage > > frames`
- `int fps`
- `int width`
- `int height`

4.96.1 Detailed Description

This class represents a video. It provides a basic interface to comfortably handle a vector of frames.

4.96.2 Constructor & Destructor Documentation

4.96.2.1 `Video (int fps, int width, int height)`

Constructor.

Parameters

<code>fps</code>	The fps the video should be played at.
<code>width</code>	The width of the video.
<code>height</code>	The height of the video.

4.96.3 Member Function Documentation

4.96.3.1 `int getFps ()`

Returns the fps of the video.

Returns

Fps of the video.

4.96.3.2 `QImage * getFrame(int index)`

Returns the frame at the given index. If the index is invalid nullptr is returned.

Parameters

<i>index</i>	The index of the frame to return.
--------------	-----------------------------------

4.96.3.3 int getHeight()

Returns the height of the video.

Returns

The height of the video.

4.96.3.4 int getNumberOfFrames()

Returns the number of frames in the video.

Returns

The number of frames in the video.

4.96.3.5 int getWidth()

Returns the width of the video.

Returns

The width of the video.

4.96.3.6 void insertFrame(int *index* = -1, unique_ptr< QImage > *frame*)

Inserts a frame at the given index. If *index* < 0 then the frame gets pushed to the back. If the index is greater than [getNumberOfFrames\(\)](#) the frames gets pushed to the back.

Parameters

<i>index</i>	The index to insert the frame at.
<i>frame</i>	The frame to insert.

4.96.3.7 void insertFrames(int *index* = -1, vector< unique_ptr< QImage > > & *frames*)

Inserts a vector of frames at the given index. If the *index* < 0 or *index* is greater than [getNumberOfFrames\(\)](#) then the frames are pushed to the back.

Parameters

<i>index</i>	The index to insert the frames at.
<i>frames</i>	The frames to insert.

4.96.3.8 void removeFrame(int *index*)

Removes the frame at the given index. If the index is invalid nothing happens.

Parameters

<i>index</i>	The index of the frame to remove.
--------------	-----------------------------------

4.96.4 Field Documentation**4.96.4.1 Model::EncodedVideo* displayVideo****4.96.4.2 int fps [private]****4.96.4.3 std::vector<std::unique_ptr<QImage>> frames [private]****4.96.4.4 int height [private]****4.96.4.5 Model::EncodedVideo* macroblockVideo****4.96.4.6 Model::EncodedVideo* rgbDiffVideo****4.96.4.7 int width [private]**

4.97 VideoConverter Class Reference

Static Public Member Functions

- static std::unique_ptr< QImage > [convertAVFrameToQImage](#) (AVFrame &frame, int width, int height)
- static std::unique_ptr< [GUI::Video](#) > [convertAVVideoToVideo](#) ([Model::AVVideo](#) &video)
- static unique_ptr< AVFrame > [convertQImageToAVFrame](#) (QImage &imgae)
- static unique_ptr< [Model::AVVideo](#) > [convertVideoToAVVideo](#) ([GUI::Video](#) &video)

4.97.1 Detailed Description

Converts AVFrames to QImages and vice versa.

4.97.2 Member Function Documentation

4.97.2.1 static std::unique_ptr<QImage> convertAVFrameToQImage (AVFrame & frame, int width, int height) [inline], [static]

Converts the given AVFrame to a QImage.

Parameters

<i>frame</i>	The avframe to convert.
<i>width</i>	The width of the frame.
<i>height</i>	The height of the frame.

Returns

The converted AVFrame.

4.97.2.2 static std::unique_ptr<[GUI::Video](#)> convertAVVideoToVideo ([Model::AVVideo](#) & video) [inline], [static]

Converts a AVVideo to a Video

Parameters

<i>video</i>	The video to convert.
--------------	-----------------------

Returns

The converted AVVideo.

4.97.2.3 static unique_ptr<AVFrame> convertQImageToAVFrame (QImage & imgae) [inline], [static]

Converts a qimage to a avframe.

Parameters

<i>imgae</i>	The qimage to convert.
--------------	------------------------

Returns

The converted qimage.

4.97.2.4 static unique_ptr<Model::AVVideo> convertVideoToAVVideo (GUI::Video & *video*) [inline],
[static]

Converts a Video to a AVVideo.

Parameters

<i>video</i>	The video to convert.
--------------	-----------------------

Returns

The converted video.

4.98 VideoLoader Class Reference

Public Member Functions

- `VideoLoader (QString path)`
- `std::unique_ptr<Model::AVVideo> loadVideo ()`

Private Attributes

- `QString path`

4.98.1 Detailed Description

This class can load a encoded video.

4.98.2 Constructor & Destructor Documentation

4.98.2.1 `VideoLoader (QString path)`

Constructor.

Parameters

<i>path</i>	Absolute path to the video to load.
-------------	-------------------------------------

4.98.3 Member Function Documentation

4.98.3.1 `std::unique_ptr<Model::AVVideo> loadVideo ()`

Loads the video and generates the AVVideo.

Returns

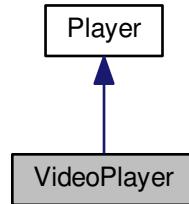
The loaded video.

4.98.4 Field Documentation

4.98.4.1 `QString path [private]`

4.99 VideoPlayer Class Reference

Inheritance diagram for VideoPlayer:



Public Member Functions

- `VideoPlayer ()`
- `void addView (GUI::FrameView &view)`
- `void removeView (GUI::FrameView &view)`
- `void setVideo (GUI::Video &video)`
- `GUI::Video * getVideo ()`
- `void setTimer (shared_ptr< GUI::Timer > timer)`
- `void clearTimer ()`
- `int getFps ()`
- `void setMasterControlPanel (GUI::ControlPanel &controlPanel)`
- `void play ()`
- `void pause ()`
- `void stop ()`
- `void nextFrame ()`
- `void previousFrame ()`
- `void setSpeed (float speed)`
- `void setPosition (int position)`
- `int getPosition ()`
- `float getSpeed ()`
- `bool isPlaying ()`
- `bool isStopped ()`
- `void reset ()`

Data Fields

- `std::vector< GUI::FrameView * > views`

Private Attributes

- `int position`
- `GUI::ControlPanel * players`
- `GUI::Video * video`
- `GUI::Timer * timer`
- `GUI::ControlPanel * masterPanel`

4.99.1 Detailed Description

This class is a video player. It provides a basic interface for handling playback of videos.

4.99.2 Constructor & Destructor Documentation

4.99.2.1 `VideoPlayer()`

Constructor.

4.99.3 Member Function Documentation

4.99.3.1 `void addView(GUI::FrameView & view)`

Adds a view. Multiple views can be added.

Parameters

<code>view</code>	The view to add.
-------------------	------------------

4.99.3.2 `void clearTimer()`

Clears the timer.

4.99.3.3 `int getFps()`

Returns the fps the player is currently playing at.

Returns

The current fps of the player.

4.99.3.4 `int getPosition() [virtual]`

Returns the position in the video.

Returns

The current position.

Implements [Player](#).

4.99.3.5 `float getSpeed() [virtual]`

Returns the speed.

Returns

The current speed.

Implements [Player](#).

4.99.3.6 GUI::Video * getVideo()

Returns a pointer to the currently played video. If no video is set nullptr is returned.

Returns

Pointer to the current video.

4.99.3.7 bool isPlaying() [virtual]

Whether the player is currently playing.

Returns

True if the player is playing.

Implements [Player](#).

4.99.3.8 bool isStopped() [virtual]

Whether the player is stopped.

Returns

True if the player is stopped.

Implements [Player](#).

4.99.3.9 void nextFrame() [virtual]

Shows the next frame.

Implements [Player](#).

4.99.3.10 void pause() [virtual]

Pauses the video.

Implements [Player](#).

4.99.3.11 void play() [virtual]

Plays the video.

Implements [Player](#).

4.99.3.12 void previousFrame() [virtual]

Shows the previous frame.

Implements [Player](#).

4.99.3.13 void removeView(GUI::FrameView & view)

Removes a view.

Parameters

<i>view</i>	The view to remove.
-------------	---------------------

4.99.3.14 void reset() [virtual]

Resets the player.

Implements [Player](#).

4.99.3.15 void setMasterControlPanel(GUI::ControlPanel & controlPanel)

Sets the MasterControlPanel. This panel is the reference for video position and speed.

4.99.3.16 void setPosition(int position) [virtual]

Sets the position in the video.

Parameters

<i>position</i>	The new position.
-----------------	-------------------

Implements [Player](#).

4.99.3.17 void setSpeed(float speed) [virtual]

Sets the speed.

Parameters

<i>speed</i>	The new speed.
--------------	----------------

Implements [Player](#).

4.99.3.18 void setTimer(shared_ptr< GUI::Timer > timer)

Sets the timer for the player. This method has to be called in order to be able to play the video.

Parameters

<i>timer</i>	The timer for the player.
--------------	---------------------------

4.99.3.19 void setVideo(GUI::Video & video)

Sets the video. If a video was previously set the old video gets deleted,

Parameters

<i>video</i>	The video to play.
--------------	--------------------

4.99.3.20 void stop() [virtual]

Stops the video.

Implements [Player](#).

4.99.4 Field Documentation

4.99.4.1 **GUI::ControlPanel*** masterPanel [private]

4.99.4.2 **GUI::ControlPanel*** players [private]

4.99.4.3 **int** position [private]

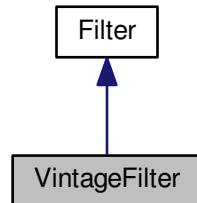
4.99.4.4 **GUI::Timer*** timer [private]

4.99.4.5 **GUI::Video*** video [private]

4.99.4.6 **std::vector<GUI::FrameView*>** views

4.100 VintageFilter Class Reference

Inheritance diagram for VintageFilter:



Public Member Functions

- [VintageFilter \(\)](#)
- string [getName \(\)](#)
- string [getFilterDescription \(\)](#)

Additional Inherited Members

4.100.1 Detailed Description

Adjusts the colors of the video to make it look vintage.

4.100.2 Constructor & Destructor Documentation

4.100.2.1 [VintageFilter \(\)](#)

4.100.3 Member Function Documentation

4.100.3.1 [string getFilterDescription \(\) \[virtual\]](#)

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implements [Filter](#).

4.100.3.2 [string getName \(\) \[virtual\]](#)

Returns the name of the filter.

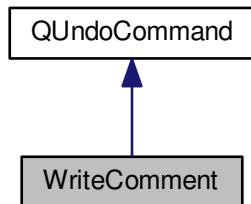
Returns

The filtername.

Implements [Filter](#).

4.101 WriteComment Class Reference

Inheritance diagram for WriteComment:



Public Member Functions

- [WriteComment \(\)](#)
- void [mergeWith \(UndoRedo::QUndoCommand command\)](#)
- void [undo \(\)](#)
- void [redo \(\)](#)
- void [id \(\)](#)

4.101.1 Constructor & Destructor Documentation

4.101.1.1 WriteComment ()

Constuctor

4.101.2 Member Function Documentation

4.101.2.1 void id ()

returns id of this command

4.101.2.2 void mergeWith (UndoRedo::QUndoCommand *command*)

attempts to merge this command with *command* if they have the same id, and the id is not -1

4.101.2.3 void redo ()

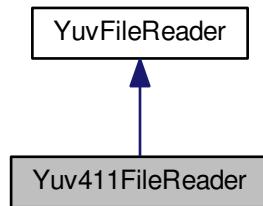
applies changes to the textbox

4.101.2.4 void undo ()

reverts changes to the textbox

4.102 Yuv411FileReader Class Reference

Inheritance diagram for Yuv411FileReader:



Public Member Functions

- [Yuv411FileReader \(QString filename, int width, int height, Utility::Compression compression\)](#)
- [unique_ptr< GUI::Video > read \(\)](#)

Static Public Member Functions

- static std::vector< QRgb > [yuv411ToRgb888 \(Yuv11Vector vector\)](#)

Private Member Functions

- unique_ptr< QImage > [parseNextFrame \(\)](#)
- [Utility::Yuv411Vector readNextVectorPacked \(\)](#)
- [Utility::Yuv411Vector readNextVectorPlanar \(\)](#)

Private Attributes

- int [position](#)
- [Utility::Compression compression](#)

Additional Inherited Members

4.102.1 Detailed Description

This class is able to read Yuv 411 files.

4.102.2 Constructor & Destructor Documentation

4.102.2.1 Yuv411FileReader (QString filename, int width, int height, Utility::Compression compression)

Constructor.

Parameters

<i>filename</i>	Absolute path to the file to load.
<i>width</i>	Width of the video.
<i>height</i>	Height of the video.
<i>compression</i>	The compression of the file.

4.102.3 Member Function Documentation**4.102.3.1 unique_ptr< QImage > parseNextFrame() [private]**

Parses the next frame.

Returns

The parsed frame.

4.102.3.2 unique_ptr< GUI::Video > read() [virtual]

Reads the file in.

Returns

The complete video.

Implements [YuvFileReader](#).

4.102.3.3 Utility::Yuv411Vector readNextVectorPacked() [private]

Reads the next vector from a packed file.

Returns

The new vector.

4.102.3.4 Utility::Yuv411Vector readNextVectorPlanar() [private]

Reads the next vector from a planar file.

Returns

The new vector.

4.102.3.5 static std::vector<QRgb> yuv411ToRgb888(Yuv11Vector vector) [inline], [static]

Converts a [Yuv11Vector](#) to the corresponding Rgb88 pixels.

Parameters

<i>vector</i>	The vector to convert.
---------------	------------------------

Returns

The computed rgb888 pixels.

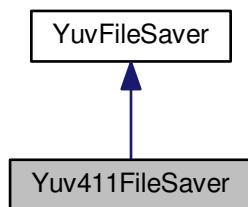
4.102.4 Field Documentation

4.102.4.1 **Utility::Compression compression** [private]

4.102.4.2 **int position** [private]

4.103 Yuv411FileSaver Class Reference

Inheritance diagram for Yuv411FileSaver:



Public Member Functions

- `Yuv411FileSaver (QString filename, GUI::Video &video, Utility::Compression compression)`
- `void save ()`

Static Public Member Functions

- static `Utility::Yuv411Vector rgb888ToYuv411 (QRgb pixel1, QRgb pixel2, QRgb pixel3, QRgb pixel4)`

Private Member Functions

- `void savePacked ()`
- `void savePlanar ()`

Private Attributes

- `Utility::Compression compression`

Additional Inherited Members

4.103.1 Detailed Description

This class can save videos in the yuv 411 format.

4.103.2 Constructor & Destructor Documentation

4.103.2.1 `Yuv411FileSaver (QString filename, GUI::Video & video, Utility::Compression compression)`

Constructor.

Parameters

<i>filename</i>	Absolute path to the file to save to.
<i>video</i>	The video to save.
<i>compression</i>	The compression mode.

4.103.3 Member Function Documentation

4.103.3.1 **static Utility::Yuv411Vector rgb888ToYuv411 (QRgb *pixel1*, QRgb *pixel2*, QRgb *pixel3*, QRgb *pixel4*)**
 [inline], [static]

Converts Rgb888 pixels to a [Yuv411Vector](#).

Parameters

<i>pixel1</i>	Pixel 1 to convert.
<i>pixel2</i>	Pixel 2 to convert.
<i>pixel3</i>	Pixel 3 to convert.
<i>pixel4</i>	Pixel 4 to convert.

Returns

The [Yuv411Vector](#)

4.103.3.2 **void save() [virtual]**

Saves the video to the file.

Implements [YuvFileSaver](#).

4.103.3.3 **void savePacked() [private]**

Saves the video in packed format.

4.103.3.4 **void savePlanar() [private]**

Saves the video in planar format.

4.103.4 Field Documentation

4.103.4.1 **Utility::Compression compression [private]**

4.104 Yuv411Vector Class Reference

Data Fields

- `uint8_t u`
- `uint8_t y1`
- `uint8_t y2`
- `uint8_t v`
- `uint8_t y3`
- `uint8_t y4`

4.104.1 Detailed Description

A [Yuv411Vector](#).

4.104.2 Field Documentation

4.104.2.1 `uint8_t u`

4.104.2.2 `uint8_t v`

4.104.2.3 `uint8_t y1`

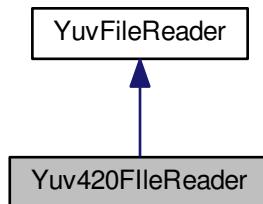
4.104.2.4 `uint8_t y2`

4.104.2.5 `uint8_t y3`

4.104.2.6 `uint8_t y4`

4.105 Yuv420FileReader Class Reference

Inheritance diagram for Yuv420FileReader:



Public Member Functions

- void `yuv420FileReader` (QString filename, int `width`, int `height`)
- unique_ptr< `GUI::Video` > `read` ()

Private Member Functions

- std::unique_ptr< `QImage` > `parseNextFrame` ()
- `Utility::Yuv444Vector` `readNextVector` ()

Private Attributes

- int `position`

Additional Inherited Members

4.105.1 Detailed Description

This class can read Yuv 420 files.

4.105.2 Member Function Documentation

4.105.2.1 std::unique_ptr< `QImage` > `parseNextFrame` () [private]

Parses the next frame.

Returns

The parsed frame.

4.105.2.2 unique_ptr< GUI::Video > read() [virtual]

Reads the file in.

Returns

The complete video.

Implements [YuvFileReader](#).

4.105.2.3 Utility::Yuv444Vector readNextVector() [private]

Reads the next Yuv 444 vector.

Returns

The new vector.

4.105.2.4 void yuv420FileReader(QString *filename*, int *width*, int *height*)

Constructor.

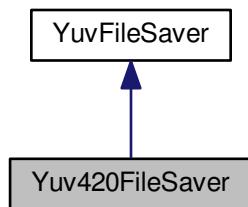
Parameters

<i>filename</i>	Absolute path to the file to load.
<i>width</i>	Width of the video.
<i>height</i>	Height of the video.

4.105.3 Field Documentation**4.105.3.1 int position [private]**

4.106 Yuv420FileSaver Class Reference

Inheritance diagram for Yuv420FileSaver:



Public Member Functions

- [Yuv420FileSaver \(QString filename, GUI::Video &video\)](#)
- [void save \(\)](#)

Private Member Functions

- [void saveFrame \(int index\)](#)

Additional Inherited Members

4.106.1 Detailed Description

This class can save videos in the yuv 420 format.

4.106.2 Constructor & Destructor Documentation

4.106.2.1 Yuv420FileSaver (QString *filename*, GUI::Video & *video*)

Constructor.

Parameters

<i>filename</i>	Absolute path to the file to save to.
<i>video</i>	The video to save.

4.106.3 Member Function Documentation

4.106.3.1 void save () [virtual]

Saves the video to the file.

Implements [YuvFileSaver](#).

4.106.3.2 void saveFrame(int *index*) [private]

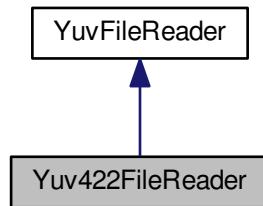
saves the *i*th frame. If the index is invalid nothing happens.

Parameters

<i>index</i>	index of the frame to safe
--------------	----------------------------

4.107 Yuv422FileReader Class Reference

Inheritance diagram for Yuv422FileReader:



Public Member Functions

- `Yuv422FileReader (QString filename, int width, int height, Utility::Compression compression)`
- `unique_ptr< GUI::Video > read ()`

Static Public Member Functions

- static vector< QRgb > `yuv422ToRgb888 (Utility::Yuv422Vector vector)`

Private Member Functions

- `unique_ptr< QImage > parseNextFrame ()`
- `Utility::Yuv422Vector readNextVectorPacked ()`
- `Utility::Yuv422Vector readNextVectorPlanar ()`

Private Attributes

- `int position`
- `Utility::Compression compression`

Additional Inherited Members

4.107.1 Detailed Description

This class can read yuv 422 files.

4.107.2 Constructor & Destructor Documentation

4.107.2.1 Yuv422FileReader (`QString filename, int width, int height, Utility::Compression compression`)

Constructor.

Parameters

<i>filename</i>	Absolute path to the file to load.
<i>width</i>	Width of the video.
<i>height</i>	Height of the video.
<i>compression</i>	Compression of the file.

4.107.3 Member Function Documentation**4.107.3.1 unique_ptr< QImage > parseNextFrame() [private]**

Parses the next frame.

Returns

The parsed frame.

4.107.3.2 unique_ptr< GUI::Video > read() [virtual]

Reads the file in.

Returns

The complete video.

Implements [YuvFileReader](#).

4.107.3.3 Utility::Yuv422Vector readNextVectorPacked() [private]

Reads the next vector from a packed file.

Returns

The new vector.

4.107.3.4 Utility::Yuv422Vector readNextVectorPlanar() [private]

Reads the next vector from a planar file.

Returns

The new vector.

4.107.3.5 static vector<QRgb> yuv422ToRgb888(Utility::Yuv422Vector vector) [inline], [static]

Converts a [Yuv422Vector](#) the Rgb888 pixels

Parameters

<i>vector</i>	The vector to convert.
---------------	------------------------

Returns

The computed rgb888 pixels.

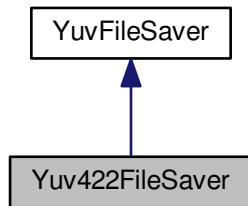
4.107.4 Field Documentation

4.107.4.1 **Utility::Compression compression** [private]

4.107.4.2 **int position** [private]

4.108 Yuv422FileSaver Class Reference

Inheritance diagram for Yuv422FileSaver:



Public Member Functions

- `Yuv422FileSaver (QString filename, GUI::Video &video, Utility::Compression compression)`
- `void save ()`

Static Public Member Functions

- static `Utility::Yuv422Vector rgb888ToYuv422 (QRgb pixel1, QRgb pixel2)`

Private Member Functions

- `void savePacked ()`
- `void savePlanar ()`

Additional Inherited Members

4.108.1 Detailed Description

This class can save a video in the yuv 422 format.

4.108.2 Constructor & Destructor Documentation

4.108.2.1 Yuv422FileSaver (QString *filename*, GUI::Video & *video*, Utility::Compression *compression*)

Constructor.

Parameters

<i>filename</i>	Absolute path to the file to save to.
<i>video</i>	The video to save.

<i>compression</i>	The compression mode.
--------------------	-----------------------

4.108.3 Member Function Documentation

4.108.3.1 static Utility::Yuv422Vector rgb888ToYuv422 (QRgb *pixel1*, QRgb *pixel2*) [inline], [static]

Converts Rgb888 pixel to a [Yuv422Vector](#).

Parameters

<i>pixel1</i>	Pixel 1 to convert.
<i>pixel2</i>	Pixel 2 to convert.

Returns

The [Yuv422Vector](#).

4.108.3.2 void save() [virtual]

Saves the video to the file.

Implements [YuvFileSaver](#).

4.108.3.3 void savePacked() [private]

Saves the video packed.

4.108.3.4 void savePlanar() [private]

Saves the video planar.

4.109 Yuv422Vector Class Reference

Private Attributes

- `uint8_t u`
- `uint8_t y1`
- `uint8_t v`
- `uint8_t y2`

4.109.1 Detailed Description

A [Yuv422Vector](#).

4.109.2 Field Documentation

4.109.2.1 `uint8_t u` [private]

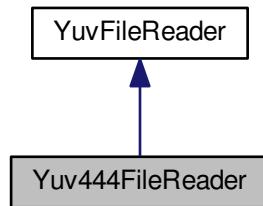
4.109.2.2 `uint8_t v` [private]

4.109.2.3 `uint8_t y1` [private]

4.109.2.4 `uint8_t y2` [private]

4.110 Yuv444FileReader Class Reference

Inheritance diagram for Yuv444FileReader:



Public Member Functions

- `Yuv444FileReader (QString filename, int width, int height, Utility::Compression compression)`
- `unique_ptr< GUI::Video > read ()`

Static Public Member Functions

- static QRgb `yuv444ToRgb888 (Utility::Yuv444Vector vector)`

Private Member Functions

- `unique_ptr< QImage > parseNextFrame ()`
- `Utility::Yuv444Vector readNextVectorPacked ()`
- `Utility::Yuv444Vector readNextVectorPlanar ()`

Private Attributes

- `int position`
- `Utility::Compression compression`

Additional Inherited Members

4.110.1 Detailed Description

This class can read Yuv 444 files.

4.110.2 Constructor & Destructor Documentation

4.110.2.1 Yuv444FileReader (`QString filename, int width, int height, Utility::Compression compression`)

Constructor.

Parameters

<i>filename</i>	Absolute path to the file to load.
<i>width</i>	Width of the video.
<i>height</i>	Height of the video.
<i>compression</i>	Compression of the file.

4.110.3 Member Function Documentation**4.110.3.1 unique_ptr< QImage > parseNextFrame() [private]**

Parses the next frame.

Returns

The parsed frame.

4.110.3.2 unique_ptr< GUI::Video > read() [virtual]

Reads the file in.

Returns

The complete video.

Implements [YuvFileReader](#).

4.110.3.3 Utility::Yuv444Vector readNextVectorPacked() [private]

Reads the next vector from a packed file.

Returns

The new vector.

4.110.3.4 Utility::Yuv444Vector readNextVectorPlanar() [private]

Reads the next vector from a planar file.

Returns

The new vector.

4.110.3.5 static QRgb yuv444ToRgb888(Utility::Yuv444Vector vector) [inline], [static]

Converts a [Yuv444Vector](#) to a Rgb888 pixel.

Parameters

<i>vector</i>	The vector to convert.
---------------	------------------------

Returns

The computed pixel.

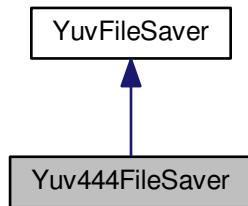
4.110.4 Field Documentation

4.110.4.1 **Utility::Compression compression** [private]

4.110.4.2 **int position** [private]

4.111 Yuv444FileSaver Class Reference

Inheritance diagram for Yuv444FileSaver:



Public Member Functions

- `Yuv444FileSaver (QString filename, GUI::Video &video, Utility::Compression compression)`
- `void save ()`

Static Public Member Functions

- static `Utility::Yuv444Vector rgb888ToYuv444 (QRgb pixel1)`

Private Member Functions

- `void savePacked ()`
- `void savePlanar ()`

Additional Inherited Members

4.111.1 Detailed Description

This class can save videos in the yuv 422 format.

4.111.2 Constructor & Destructor Documentation

4.111.2.1 Yuv444FileSaver (QString *filename*, GUI::Video & *video*, Utility::Compression *compression*)

Constructor.

Parameters

<i>filename</i>	Absolute path to the file to save to.
<i>video</i>	The video to save.

<i>compression</i>	The compression mode.
--------------------	-----------------------

4.111.3 Member Function Documentation

4.111.3.1 static Utility::Yuv444Vector rgb888ToYuv444(QRgb *pixel1*) [inline], [static]

Converts a Rgb888 pixel to a [Yuv444Vector](#).

Parameters

<i>pixel1</i>	The pixel to convert.
---------------	-----------------------

Returns

The [Yuv444Vector](#).

4.111.3.2 void save() [virtual]

Saves the video to the file.

Implements [YuvFileSaver](#).

4.111.3.3 void savePacked() [private]

Saves the video in packed format.

4.111.3.4 void savePlanar() [private]

Saves the video in planar format.

4.112 Yuv444Vector Class Reference

Private Attributes

- `uint8_t y`
- `uint8_t u`
- `uint8_t v`

4.112.1 Detailed Description

A [Yuv444Vector](#).

4.112.2 Field Documentation

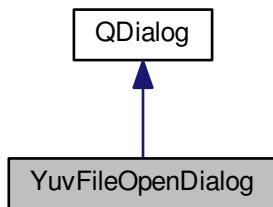
4.112.2.1 `uint8_t u` [private]

4.112.2.2 `uint8_t v` [private]

4.112.2.3 `uint8_t y` [private]

4.113 YuvFileDialog Class Reference

Inheritance diagram for YuvFileDialog:



Public Member Functions

- [YuvFileDialog \(GUI::QWidget *parent=0\)](#)
- [QString getFilename \(\)](#)
- [void show \(\)](#)
- [bool wasSuccessfull \(\)](#)

Private Member Functions

- [void createUi \(\)](#)

Static Private Member Functions

- [static void loadRecentlyUsed \(\)](#)

Private Attributes

- [QPushButton * button_cancel](#)
- [QPushButton * button_ok](#)
- [QListView * listView_recentlyUsed](#)
- [QLabel * label_selectedFile](#)
- [QPushButton * button_chooseFile](#)
- [QLabel * label_recentlyUsed](#)

Static Private Attributes

- [static QListViewModel * model_recentlyUsed](#)

4.113.1 Detailed Description

This class is the dialog that gets shown when the user wants to select a yuv file to load.

4.113.2 Constructor & Destructor Documentation

4.113.2.1 **YuvFileDialog (GUI::QWidget * *parent* = 0)**

Constructor.

4.113.3 Member Function Documentation

4.113.3.1 **void createUi () [private]**

Creates the ui.

4.113.3.2 **QString getFilename ()**

Returns the absolute path to the file the user wants to open.

Returns

Absolute path to the user chosen file.

4.113.3.3 **static void loadRecentlyUsed () [inline], [static], [private]**

Loads the recently opened yuv files.

4.113.3.4 **void show ()**

Shows the dialog.

4.113.3.5 **bool wasSuccessfull ()**

Whether the user clicked ok or cancel.

Returns

True if the user clicked ok. false otherwise.

4.113.4 Field Documentation

4.113.4.1 **QPushButton* button_cancel [private]**

4.113.4.2 **QPushButton* button_chooseFile [private]**

4.113.4.3 **QPushButton* button_ok [private]**

4.113.4.4 **QLabel* label_recentlyUsed [private]**

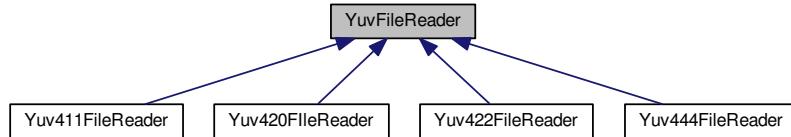
4.113.4.5 **QLabel* label_selectedFile [private]**

4.113.4.6 **QListView* listView_redcentlyUsed [private]**

4.113.4.7 **QListViewModel* model_recentlyUsed [static], [private]**

4.114 YuvFileReader Class Reference

Inheritance diagram for YuvFileReader:



Public Member Functions

- `YuvFileReader (QString filename, int width, int height)`
- `virtual unique_ptr< GUI::Video > read ()=0`

Static Public Member Functions

- `static int clamp (int value)`

Protected Attributes

- `unique_ptr< QByteArray > binaryData`
- `int width`
- `int height`
- `unique_ptr< GUI::Video > video`

4.114.1 Detailed Description

This is the base class for all different yuv file readers.

4.114.2 Constructor & Destructor Documentation

4.114.2.1 YuvFileReader (QString *filename*, int *width*, int *height*)

Constructor.

Parameters

<i>filename</i>	The absolute path to the file to load.
<i>width</i>	The width of the video.
<i>height</i>	The height of the video.

4.114.3 Member Function Documentation

4.114.3.1 static int clamp (int *value*) [inline], [static]

Clamps the given value to the range [0,255].

Parameters

<i>value</i>	The value to clamp.
--------------	---------------------

Returns

The clamped value.

4.114.3.2 virtual unique_ptr<GUI::Video> read() [pure virtual]

Reads the file in.

Returns

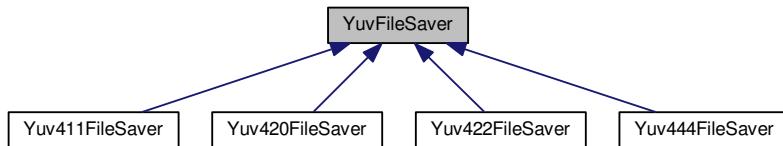
The complete video.

Implemented in [Yuv411FileReader](#), [Yuv422FileReader](#), [Yuv444FileReader](#), and [Yuv420FileReader](#).

4.114.4 Field Documentation**4.114.4.1 unique_ptr<QByteArray> binaryData [protected]****4.114.4.2 int height [protected]****4.114.4.3 unique_ptr<GUI::Video> video [protected]****4.114.4.4 int width [protected]**

4.115 YuvFileSaver Class Reference

Inheritance diagram for YuvFileSaver:



Public Member Functions

- [YuvFileSaver \(QString filename, GUI::Video &video\)](#)
- virtual void [save \(\)=0](#)

Protected Attributes

- int [width](#)
- int [height](#)
- GUI::Video * [video](#)
- QFile [file](#)
- QDataStream [dataStream](#)

4.115.1 Detailed Description

This is the base class for yuv savers.

4.115.2 Constructor & Destructor Documentation

4.115.2.1 YuvFileSaver (QString *filename*, GUI::Video & *video*)

Constructor.

Parameters

<i>filename</i>	Absolute path to the file to save to.
<i>video</i>	The video to save.

4.115.3 Member Function Documentation

4.115.3.1 virtual void save() [pure virtual]

Saves the video to the file.

Implemented in [Yuv411FileSaver](#), [Yuv422FileSaver](#), [Yuv444FileSaver](#), and [Yuv420FileSaver](#).

4.115.4 Field Documentation

4.115.4.1 **QDataStream** **dataStream** [protected]

4.115.4.2 **QFile** **file** [protected]

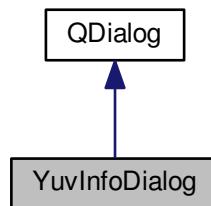
4.115.4.3 **int** **height** [protected]

4.115.4.4 **GUI::Video*** **video** [protected]

4.115.4.5 **int** **width** [protected]

4.116 YuvInfoDialog Class Reference

Inheritance diagram for YuvInfoDialog:



Public Member Functions

- `YuvInfoDialog (GUI::QWidget *parent)`
- `int getFps ()`
- `int getWidth ()`
- `int getHeight ()`
- `Utility::Compression getCompression ()`
- `GUI::PixelScheme getPixelScheme ()`
- `bool wasSuccessful ()`
- `void show ()`

Private Member Functions

- `void createUi ()`

Private Attributes

- `QLineEdit *lineEdit_width`
- `QLineEdit *lineEdit_height`
- `QLineEdit *lineEdit_fps`
- `GUI::QComboBox *comboBox_pixelScheme`
- `GUI::QComboBox *comboBox_compression`
- `GUI::QWidget *label_resolution`
- `GUI::QWidget *label_fps`
- `GUI::QWidget *label_compression`
- `GUI::QWidget *label_pixelScheme`
- `QPushButton *button_ok`
- `QPushButton *button_cancel`
- `GUI::QWidget *label_x`

4.116.1 Detailed Description

This class is the dialog that gets shown to ask the user for additional information about the yuv file he wants to load.

4.116.2 Constructor & Destructor Documentation

4.116.2.1 **YuvInfoDialog (GUI::QWidget * parent)**

Constructor.

4.116.3 Member Function Documentation

4.116.3.1 **void createUi () [private]**

Creates the gui.

4.116.3.2 **Utility::Compression getCompression ()**

Returns the compression the user entered.

Returns

The compression.

4.116.3.3 **int getFps ()**

Returns the fps the user entered.

Returns

The fps.

4.116.3.4 **int getHeight ()**

Returns the height the user entered.

Returns

The height.

4.116.3.5 **GUI::PixelSheme getPixelSheme ()**

Returns the pixelsheme the user entered.

Returns

The pixelsheme.

4.116.3.6 **int getWidth ()**

Returns the width the user entered.

Returns

The width.

4.116.3.7 void show()

Shows the dialog.

4.116.3.8 bool wasSuccessful()

Whether the user clicked ok or cancel.

Returns

True if the user clicked ok. false otherwise.

4.116.4 Field Documentation**4.116.4.1 QPushButton* button_cancel [private]****4.116.4.2 QPushButton* button_ok [private]****4.116.4.3 GUI::QComboBox* comboBox_compression [private]****4.116.4.4 GUI::QComboBox* comboBox_pixelSheme [private]****4.116.4.5 GUI::QWidget* label_compression [private]****4.116.4.6 GUI::QWidget* label_fps [private]****4.116.4.7 GUI::QWidget* label_pixelSheme [private]****4.116.4.8 GUI::QWidget* label_resolution [private]****4.116.4.9 GUI::QWidget* label_x [private]****4.116.4.10 QLineEdit* lineEdit_fps [private]****4.116.4.11 QLineEdit* lineEdit_height [private]****4.116.4.12 QLineEdit* lineEdit_width [private]**

4.117 YuvVideo Class Reference

Public Member Functions

- `YuvVideo (QString path, GUI::PixelSheme type, int width, int height, int fps)`
- `QString getPath ()`
- `Utility::Compression getCompression ()`
- `Utility::YuvType getYuvType ()`
- `Model::AVVideo & getAvVideo ()`
- `GUI::Video & getVideo ()`

Data Fields

- `UndoRedo::LoadFilterVideo * video`
- `Utility::Compression * compression`
- `GUI::Video * displayVideo`
- `Model::AVVideo * avVideo`

Private Member Functions

- `void loadVideo ()`

Private Attributes

- `QString path`
- `int height`
- `int width`
- `int fps`
- `GUI::PixelSheme * yuvType`

4.117.1 Detailed Description

This class holds a yuv video with all its properties.

4.117.2 Constructor & Destructor Documentation

4.117.2.1 `YuvVideo (QString path, GUI::PixelSheme type, int width, int height, int fps)`

Constructor.

Parameters

<code>path</code>	Path to the yuv file.
<code>type</code>	Pixelscheme of the yuv video.
<code>width</code>	Width of the video.
<code>height</code>	Height of the video.
<code>fps</code>	Fps of the video.

4.117.3 Member Function Documentation

4.117.3.1 Model::AVVideo & getAvVideo()

Returns the [AVVideo](#).

Returns

The [AVVideo](#).

4.117.3.2 Utility::Compression getCompression()

Returns the compression of the video.

Returns

The compression of the video.

4.117.3.3 QString getPath()

Returns the path to the video.

Returns

Path to the video.

4.117.3.4 GUI::Video & getVideo()

Returns the Video.

Returns

The Video.

4.117.3.5 Utility::YuvType getYuvType()

Returns the pixel sheme of the video.

Returns

The pixel scheme.

4.117.3.6 void loadVideo() [private]

Loads the video from the file.

4.117.4 Field Documentation

4.117.4.1 Model::AVVideo* avVideo

4.117.4.2 Utility::Compression* compression

4.117.4.3 **GUI::Video*** displayVideo

4.117.4.4 **int fps** [private]

4.117.4.5 **int height** [private]

4.117.4.6 **QString path** [private]

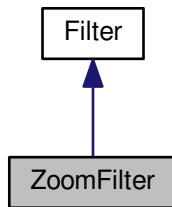
4.117.4.7 **UndoRedo::LoadFilterVideo* video**

4.117.4.8 **int width** [private]

4.117.4.9 **GUI::PixelScheme* yuvType** [private]

4.118 ZoomFilter Class Reference

Inheritance diagram for ZoomFilter:



Public Member Functions

- `ZoomFilter ()`
- string `getName ()`
- string `getFilterDescription ()`
- int `getIntensity ()`
- void `setIntensity (int intensity)`

Private Attributes

- int `intensity`

Additional Inherited Members

4.118.1 Detailed Description

Zooms into the center of the video.

4.118.2 Constructor & Destructor Documentation

4.118.2.1 `ZoomFilter()`

Constructor.

4.118.3 Member Function Documentation

4.118.3.1 `string getFilterDescription() [virtual]`

Returns the string that the ffmpeg library needs to apply the filter to a video.

Returns

The string for the ffmpeg library.

Implements [Filter](#).

4.118.3.2 int getIntensity()

Returns the intensity of the zoom.

Returns

The zoom intensity.

4.118.3.3 string getName() [virtual]

Returns the name of the filter.

Returns

The filtername.

Implements [Filter](#).

4.118.3.4 void setIntensity(int *intensity*)

Sets the intensity of the zoom.

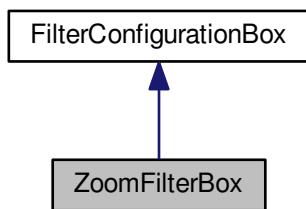
Parameters

<i>intensity</i>	The new intensity.
------------------	--------------------

4.118.4 Field Documentation**4.118.4.1 int intensity [private]**

4.119 ZoomFilterBox Class Reference

Inheritance diagram for ZoomFilterBox:



Public Member Functions

- [ZoomFilterBox \(GUI::QWidget *parent \)](#)

Additional Inherited Members

4.119.1 Detailed Description

This class contains the gui elements for changing the options of a zoom filter.

4.119.2 Constructor & Destructor Documentation

4.119.2.1 ZoomFilterBox (GUI::QWidget * *parent*)

Constructor.

Appendix

