

ENTWURFSHEFT

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

Inhaltsverzeichnis

1 Einleitung	3
2 Diagramme	5
3 Namensräume	13
4 Klassendokumentation	19
Index	145
Anhang	145

Kapitel 1

Einleitung

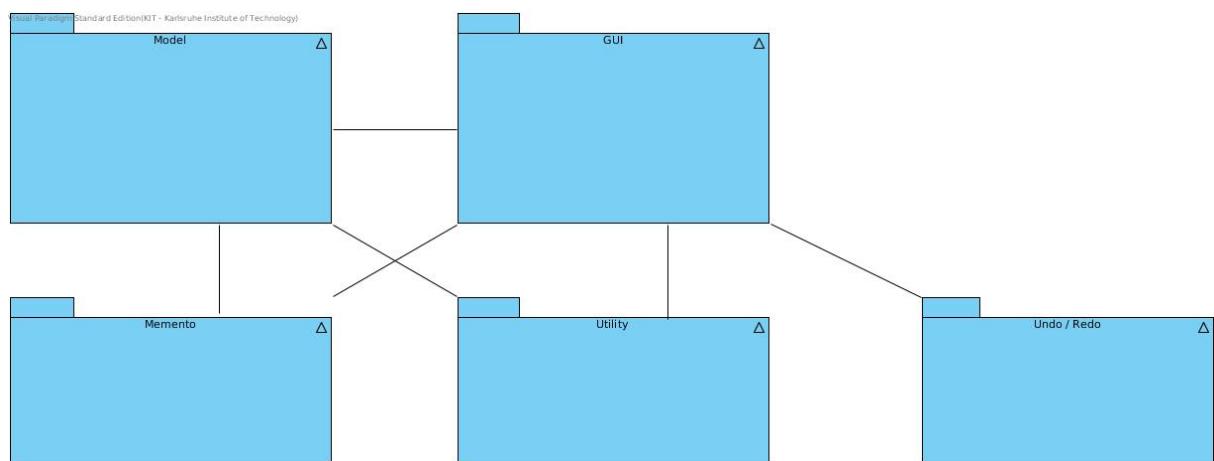
Vive (lang: Video veritatem) ist ein Programm zum Testen verschiedener Videoencoder. Dieses Dokument ist das Entwurfsheft, in welchem sich Diagramme und Spezifikationen zu jeder einzelnen Klasse befinden. Außerdem dient es als Richtlinie für die Implementierungsphase. Das Entwurfsheft ist in mehrere Abschnitte gegliedert: Im ersten Abschnitt sind Diagramme zu sehen, die einen groben Überblick über die Struktur des Programms liefern. Darauf folgt die detaillierte Beschreibung jeder einzelnen Klasse und Methode. Eine Übersicht über die Implementierungsphase in Form eines Gantt-Diagrams ist im Anhang enthalten.

Kapitel 2

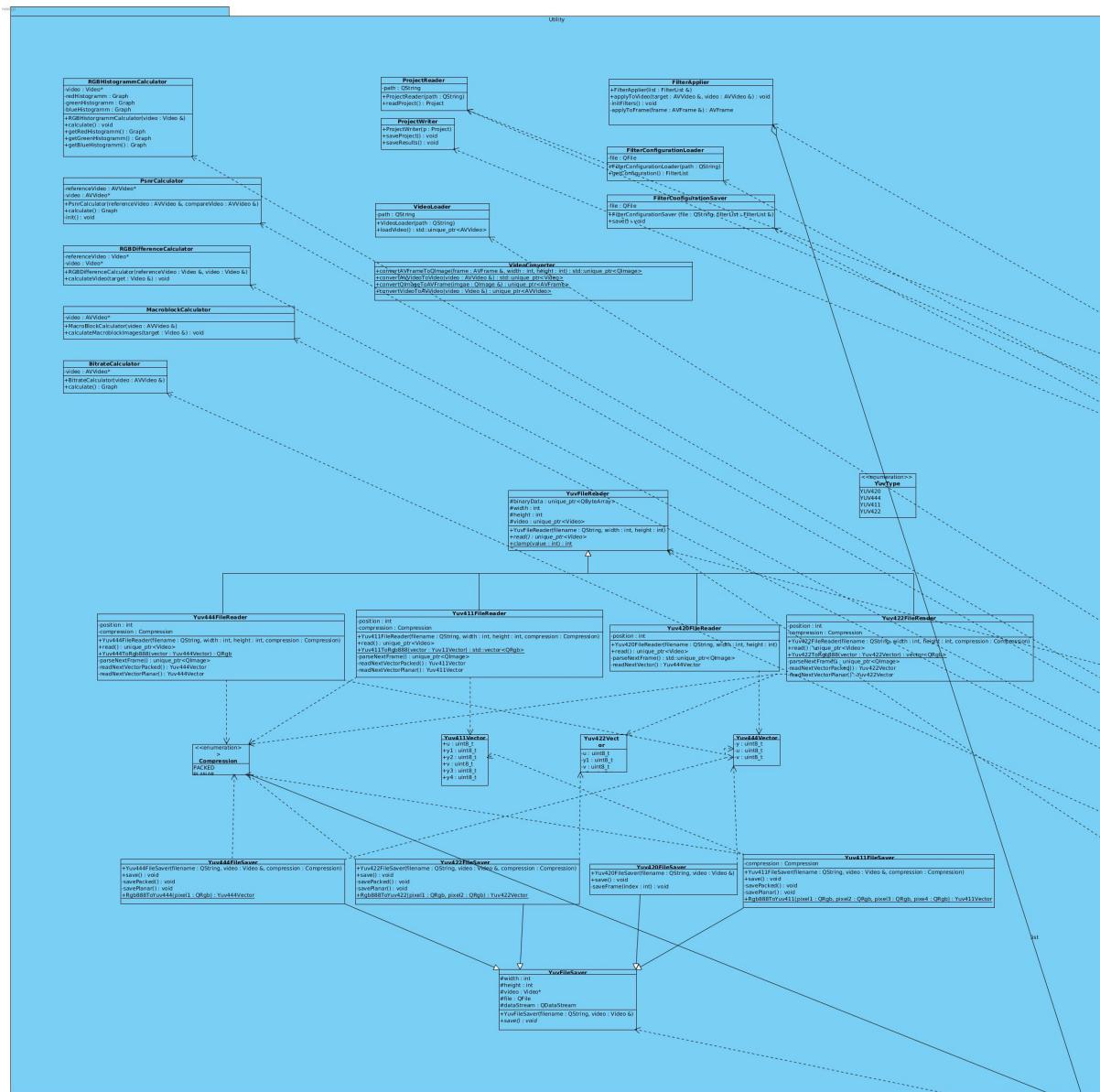
Diagramme

2.1 Klassendiagramme

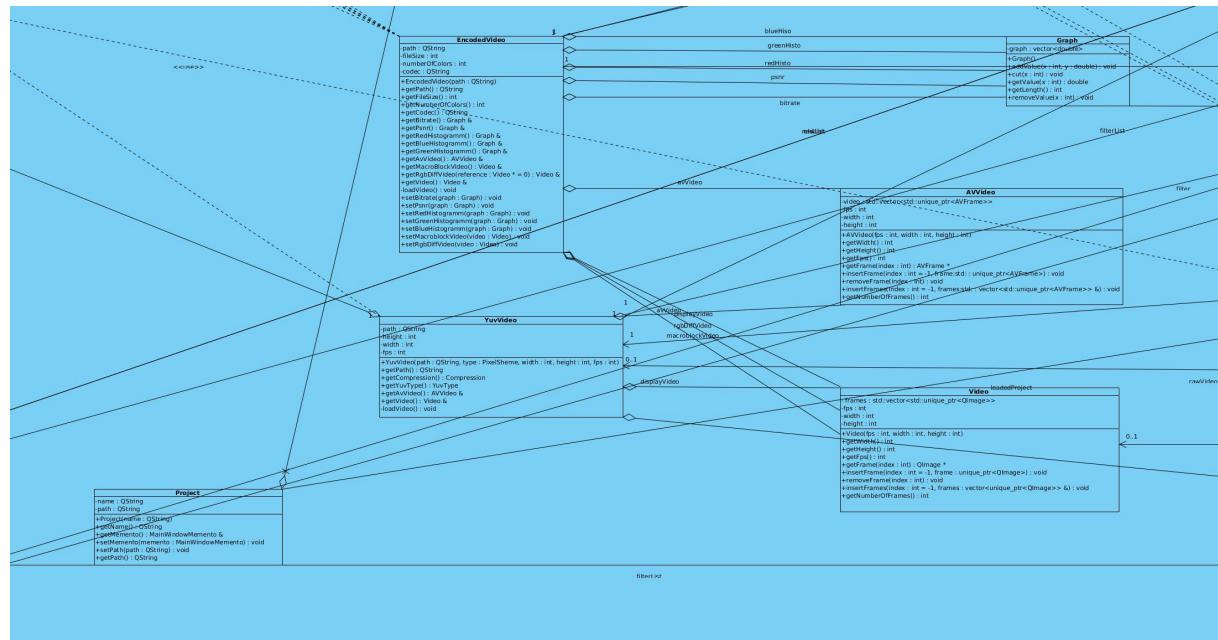
Der grobe Aufbau des Programms



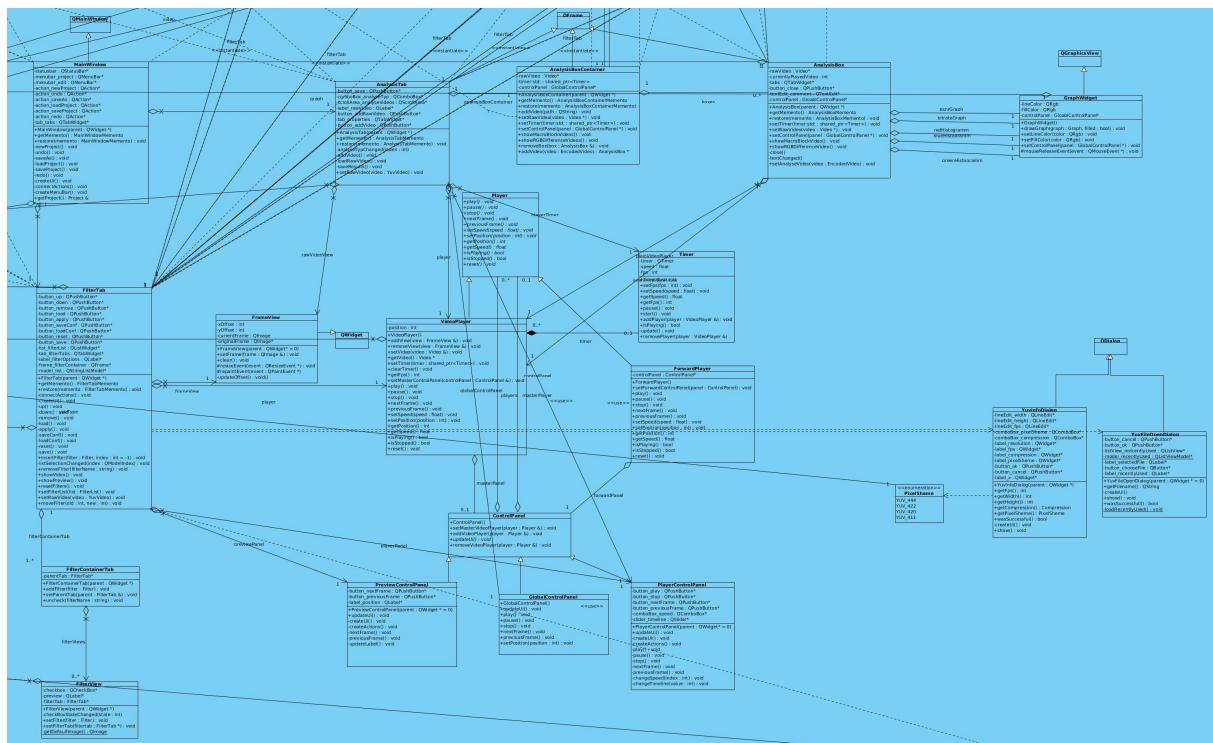
Das Utility Package



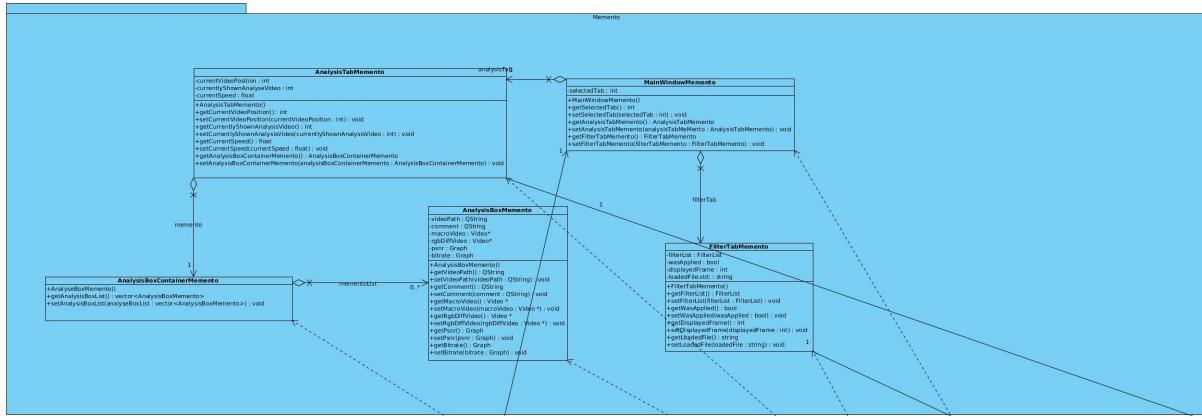
Das Model ohne die Filter



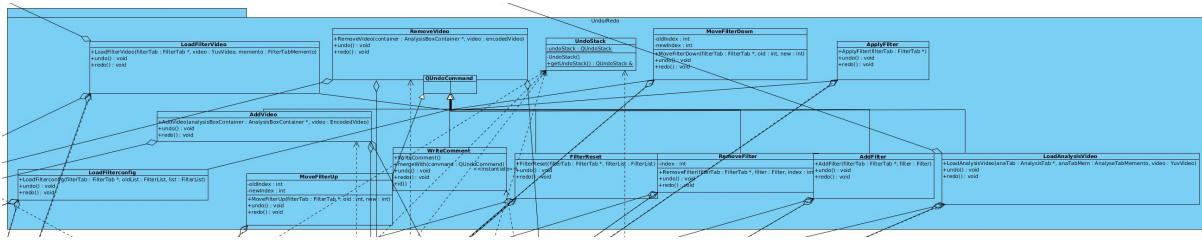
Die GUI ohne die FilterBox Klassen



Das Memento Package



Das Undo/Redo Package



2.2 Sequenzdiagramme

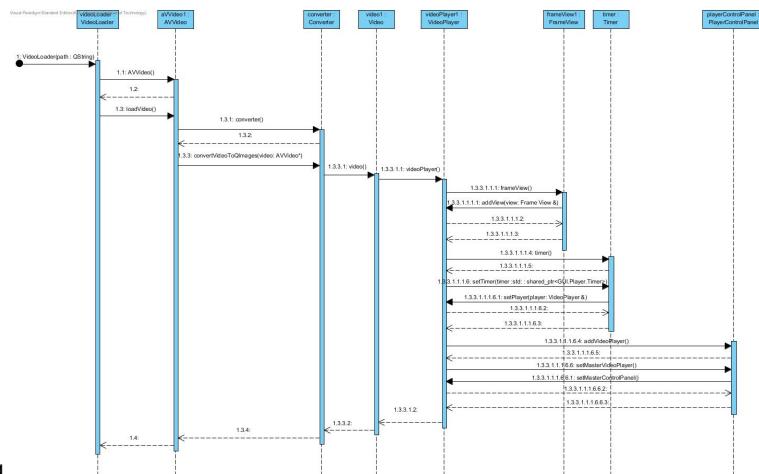


Diagram1.jpg

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

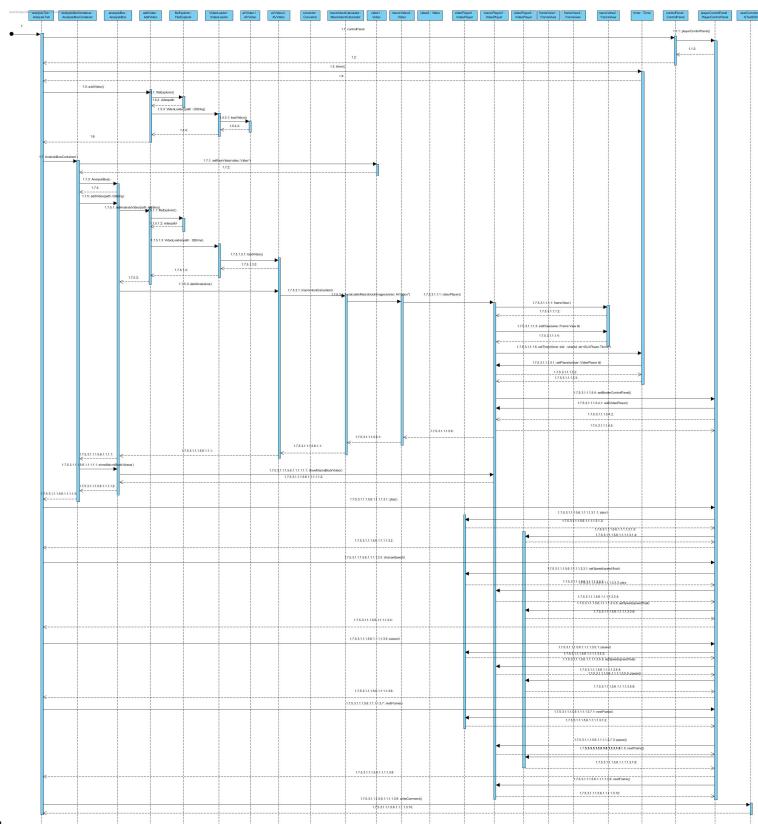


Diagram2.jpg

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.

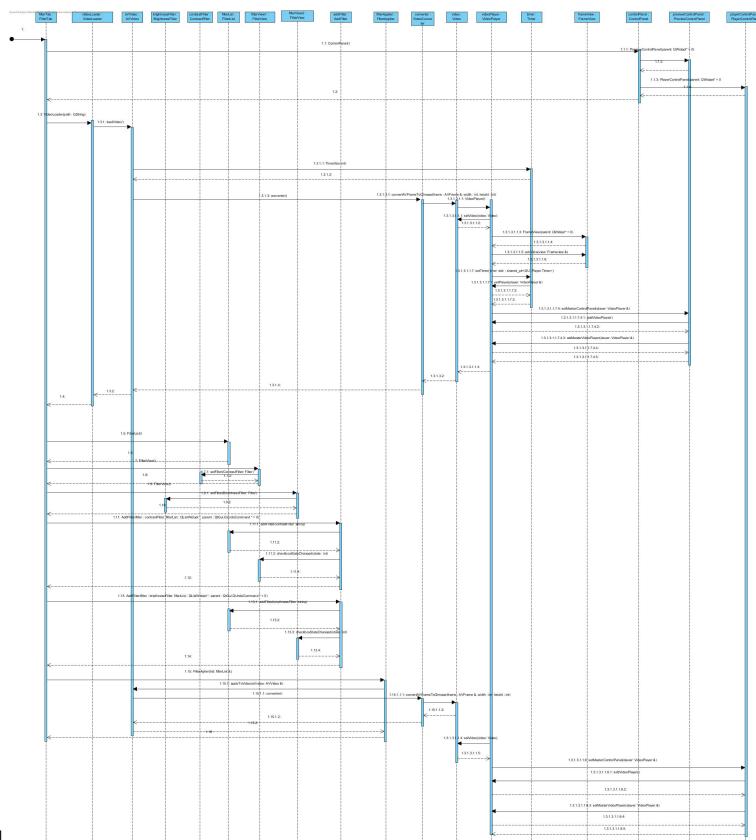


Diagram3.jpg

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.

Kapitel 3

Namensräume

3.1 GUI Namespace Reference

Namespaces

- [Player](#)
- [QtGui](#)

Data Structures

- class [AnalysisBox](#)
- class [AnalysisBoxContainer](#)
- class [AnalysisTab](#)
- class [BlendingFilterBox](#)
- class [BlurFilterBox](#)
- class [BorderFilterBox](#)
- class [BrightnessFilterBox](#)
- class [ColorbalanceFilterBox](#)
- class [ContrastFilterBox](#)
- class [FilterConfigurationBox](#)
- class [FilterContainerTab](#)
- class [FilterTab](#)
- class [FilterView](#)
- class [ForwardPlayer](#)
- class [GlobalControlPanel](#)
- class [GraphWidget](#)
- class [GridFilterBox](#)
- class [MainWindow](#)
- class [MirrorFilterBox](#)
- class [NoiseFilterBox](#)
- class [PlainFilterBox](#)
- class [PosterFilterBox](#)
- class [RectangleFilterBox](#)
- class [RGBFilterBox](#)
- class [RotationFilterBox](#)
- class [SaturationFilterBox](#)
- class [ScaleFilterBox](#)
- class [SharpnessFilterBox](#)
- class [ZoomFilterBox](#)

3.2 GUI::Player Namespace Reference

Data Structures

- class [ControlPanel](#)
- class [FrameView](#)
- class [Player](#)
- class [PlayerControlPanel](#)
- class [PreviewControlPanel](#)
- class [QDialog](#)
- class [QFrame](#)
- class [QWidget](#)
- class [Timer](#)
- class [Video](#)
- class [VideoPlayer](#)
- class [YuvFileDialog](#)
- class [YuvInfoDialog](#)

Enumerations

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

3.2.1 Enumeration Type Documentation

3.2.1.1 enum PixelSheme

Enumerator

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

3.3 GUI::QtGui Namespace Reference

Data Structures

- class [QCheckBox](#)
- class [QComboBox](#)
- class [QGraphicsView](#)
- class [QMainWindow](#)

3.4 Memento Namespace Reference

Data Structures

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

3.5 Model Namespace Reference

Namespaces

- Filter

Data Structures

- class AVVideo
- class EncodedVideo
- class Graph
- class Project
- class YuvVideo

3.6 Model::Filter Namespace Reference

Data Structures

- class BlackWhiteFilter
- class BlendingFilter
- class BlurFilter
- class BorderFilter
- class BrightnessFilter
- class ColorbalanceFilter
- class ContrastFilter
- class EdgeFilter
- class Filter
- class FilterApplier
- class FilterList
- class GridFilter
- class MirrorFilter
- class NegativeFilter
- class NoiseFilter
- class PosterFilter
- class RectangleFilter
- class RGBFilter
- class RotationFilter
- class SaturationFilter
- class ScaleFilter
- class SepiaFilter
- class SharpnessFilter
- class VintageFilter
- class ZoomFilter

Enumerations

- enum BasicColor { RED, GREEN, BLUE }
- enum MirrorMode { HORIZONTAL, VERTICAL }
- enum NoiseMode { STATIC, RANDOM, MUSTER }

3.6.1 Enumeration Type Documentation

3.6.1.1 enum BasicColor

Enumerator

RED

GREEN

BLUE

3.6.1.2 enum MirrorMode

Enumerator

HORIZONTAL

VERTICAL

3.6.1.3 enum NoiseMode

Enumerator

STATIC

RANDOM

MUSTER

3.7 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](#)

3.8 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 [YuvTupe](#) { **YUV420**, **YUV444**, **YUV411**, **YUV422** }

3.8.1 Enumeration Type Documentation

3.8.1.1 enum **Compression**

Enumerator

PACKED
PLANAR

3.8.1.2 enum **YuvTupe**

Enumerator

YUV420
YUV444
YUV411
YUV422

Kapitel 4

Klassendokumentation

4.1 AddFilter Class Reference

Public Member Functions

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

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 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.

4.2 AddVideo Class Reference

Public Member Functions

- `AddVideo (GUI::AnalysisBoxContainer *analysisBoxContainer, Model::EncodedVideo video)`
- `void undo ()`
- `void redo ()`

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

<code>analysisBoxContainer</code>	The AnalysisBoxContainer to operate on.
<code>video</code>	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.

4.3 AnalysisBox Class Reference

Public Member Functions

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

Data Fields

- `UndoRedo::RemoveVideo * anaBox`

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::Player::QWidget * *parent*)**

4.3.3 Member Function Documentation

4.3.3.1 **Memento::AnalysisBoxMemento getMemento ()**

Creates a memento which contains the state of the box.

Returns

The created memento.

4.3.3.2 **void restore (Memento::AnalysisBoxMemento *memento*)**

Restores the box based on the memento.

Parameters

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

4.3.3.3 **void setAnalyseVideo (Model::EncodedVideo *video*)**

Sets the video this box shall present.

Parameters

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

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

Sets the [GlobalControlPanel](#).

Parameters

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

4.3.3.5 **void setRawVideo (GUI::Player::Video * *video*)**

Sets the rawvideo the encoded video is compared to.

Parameters

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

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

Sets the timer for the videoplayer.

Parameters

<i>timer:std::shared_ptr<GUI::Player::Timer></i>	The timer for the videoplayer.
--	--------------------------------

4.3.3.7 void showMacroBlockVideo ()

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

4.3.3.8 void showRGBDifferenceVideo ()

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

4.3.4 Field Documentation**4.3.4.1 UndoRedo::RemoveVideo* anaBox****4.4 AnalysisBoxContainer Class Reference****Public Member Functions**

- [AnalysisBoxContainer \(GUI::Player::QWidget *parent\)](#)
- [Memento::AnalysisBoxContainerMemento getMemento \(\)](#)
- [void restore \(Memento::AnalysisBoxContainerMemento memento\)](#)
- [void addVideo \(QString path\)](#)
- [void setRawVideo \(GUI::Player::Video *video\)](#)
- [void setTimer \(shared_ptr<GUI::Player::Timer> timer:std::shared_ptr<GUI::Player::Timer>\)](#)
- [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](#)

4.4.1 Detailed Description

Contains and manages the AnalysisBoxes.

4.4.2 Constructor & Destructor Documentation**4.4.2.1 AnalysisBoxContainer (GUI::Player::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::Player::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::Player::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****4.5 AnalysisBoxContainerMemento Class Reference****Public Member Functions**

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

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

<code>analyseBoxList</code>	The list of the mementos.
-----------------------------	---------------------------

4.6 AnalysisBoxMemento Class Reference

Public Member Functions

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

4.7 AnalysisTab Class Reference

Public Member Functions

- [AnalysisTab \(GUI::Player::QWidget *parent\)](#)
- [Memento::AnalysisTabMemento getMemento \(\)](#)
- [void restore \(Memento::AnalysisTabMemento memento\)](#)
- [void setRawVideo \(Model::YuvVideo video\)](#)

Data Fields

- [UndoRedo::LoadAnalysisVideo * anaTab](#)
- [GUI::Player::FrameView * rawVideoView](#)

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::Player::QWidget * *parent*)

Constructor.

4.7.3 Member Function Documentation

4.7.3.1 Memento::AnalysisTabMemento getMemento ()

Creates a memento which contains the state of this tab.

Returns

The created memento.

4.7.3.2 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.
----------------	--

4.7.3.3 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 UndoRedo::LoadAnalysisVideo* anaTab

4.7.4.2 GUI::Player::FrameView* rawVideoView

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](#)

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← Container← 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← Shown← AnalysisVideo</i>	The currently 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>currentVideo← Position</i>	The current position of the player.
-----------------------------------	-------------------------------------

4.8.4 Field Documentation**4.8.4.1 UndoRedo::LoadAnalysisVideo* memento****4.9 ApplyFilter Class Reference****Public Member Functions**

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

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.

4.10 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 \(int index=-1, unique_ptr< AVFrame > frame:std::unique_ptr< AVFrame >\)](#)
- void [removeFrame \(int index\)](#)
- void [insertFrames \(int index=-1, vector< std::unique_ptr< AVFrame > > &frames:std::vector< std::unique_ptr< AVFrame > >\)](#)
- int [getNumberOfFrames \(\)](#)

Data Fields

- [Model::EncodedVideo * avVideo](#)

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

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

<code>frames:std::vector<Frame></code>	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

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

4.10.4 Field Documentation

4.10.4.1 Model::EncodedVideo* avVideo

4.11 BitrateCalculator Class Reference

Public Member Functions

- [BitrateCalculator \(Model::AVVideo &video\)](#)
- [Model::Graph calculate \(\)](#)

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

<code>video</code>	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.

4.12 BlackWhiteFilter Class Reference

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

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\(\)](#)

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

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

4.13.3.7 void setInBlend (bool *inBlend*)

Sets whether it is an in blending.

Parameters

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

4.13.3.8 void setStartFrame (int *startFrame*)

Sets the start frame of the blending.

Parameters

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

4.14 BlendingFilterBox Class Reference

Public Member Functions

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

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::Player::QWidget * *parent*)

Constructor.

4.15 BlurFilter Class Reference

Public Member Functions

- [BlurFilter \(\)](#)
- [bool getPreserveEdges \(\)](#)
- [void setPreserveEdges \(bool preserveEdges\)](#)
- [int getIntensity \(\)](#)
- [void setIntensity \(int intensity\)](#)
- [string getFilterDescription \(\)](#)
- [string getName \(\)](#)

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.
----------------------	---------------------------------------

4.16 BlurFilterBox Class Reference

Public Member Functions

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

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 \(GUI::Player::QWidget * parent \)](#)

Constructor.

4.17 BorderFilter Class Reference

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* \(\)](#)

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.
------------	--

4.18 BorderFilterBox Class Reference

Public Member Functions

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

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 \(GUI::Player::QWidget * parent \)](#)

Constructor.

4.19 BrightnessFilter Class Reference

Public Member Functions

- [BrightnessFilter \(\)](#)
- [int getIntensity \(\)](#)
- [void setIntensity \(int intensity\)](#)
- [string getName \(\)](#)
- [string getFilterDescription \(\)](#)

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.
------------------	--------------------

4.20 BrightnessFilterBox Class Reference

Public Member Functions

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

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::Player::QWidget * parent`)

Constructor.

4.21 ColorbalanceFilter Class Reference

Public Member Functions

- `ColorbalanceFilter ()`
- `Model::Filter::BasicColor getColor ()`
- `void setColor (Model::Filter::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)`

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::Filter::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

brightPixels	True if the bright pixels shall be changed.
--------------	---

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

Sets the color whose balance shall be changed.

Parameters

color	The color to change.
-------	----------------------

4.21.3.10 void setDarkPixels (bool *darkPixels*)

Sets whether the dark pixels shall be changed.

Parameters

darkPixels	True if the dark pixels are changed.
------------	--------------------------------------

4.21.3.11 void setIntensity (int *intensity*)

Sets the intensity of the change.

Parameters

intensity	The intensity.
-----------	----------------

4.21.3.12 void setMediumPixels (bool *mediumPixels*)

Sets whether the medium pixels shall be changed.

Parameters

mediumPixels	True if the medium pixels are changed.
--------------	--

4.22 ColorbalanceFilterBox Class Reference

Public Member Functions

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

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::Player::QWidget * parent`)

Constructor.

4.23 ContrastFilter Class Reference

Public Member Functions

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

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.
------------------	--------------------

4.24 ContrastFilterBox Class Reference

Public Member Functions

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

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::Player::QWidget * *parent*)**

Constructor.

4.25 ControlPanel Class Reference

Public Member Functions

- [ControlPanel \(\)](#)
- [void setMasterVideoPlayer \(Player::Player &player\)](#)
- [void addVideoPlayer \(GUI::Player::Player &player\)](#)
- [virtual void updateUi \(\)=0](#)
- [void removeVideoPlayer \(Player::Player &player\)](#)

Data Fields

- [GUI::ForwardPlayer * forwardPanel](#)
- [GUI::Player::VideoPlayer * masterPanel](#)

Protected Attributes

- [std::vector< GUI::Player::VideoPlayer * > players](#)

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 & 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::Player::VideoPlayer* masterPanel

4.25.4.3 std::vector<GUI::Player::VideoPlayer*> players [protected]

4.26 EdgeFilter Class Reference

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::Player::Video & getMacroBlockVideo ()`
- `GUI::Player::Video & getRgbDiffVideo (GUI::Player::Video *reference=0)`
- `GUI::Player::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::Player::Video video)`
- `void setRgbDiffVideo (GUI::Player::Video video)`

Data Fields

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

4.27.1 Detailed Description

This class contains all analysis info of a encoded video.

4.27.2 Constructor & Destructor Documentation

4.27.2.1 EncodedVideo (QString path)

Constructor.

Parameters

<code>path</code>	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::Player::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::Player::Video & getRgbDiffVideo(GUI::Player::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::Player::Video & getVideo()

Returns the Video.

Returns

The Video.

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

Sets the bitrate graph.

Parameters

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

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

Sets the blue histogramm graph.

Parameters

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

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

Sets the green histogramm graph.

Parameters

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

4.27.3.17 void setMacroblockVideo (GUI::Player::Video *video*)

Sets the video that shows the macroblocks.

Parameters

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

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

Sets the psnr graph.

Parameters

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

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

Sets the red histogramm graph.

Parameters

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

4.27.3.20 void setRgbDiffVideo (GUI::Player::Video *video*)

Sets the video that shows a rgb differenece 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 GUI::Player::Video* displayVideo

4.27.4.5 Model::Graph* greenHisto

4.27.4.6 **GUI::Player::Video*** macroblockVideo

4.27.4.7 **Model::Graph*** psnr

4.27.4.8 **Model::Graph*** redHisto

4.27.4.9 **GUI::Player::Video*** rgbdiffVideo

4.27.4.10 **GUI::AnalysisBox*** video

4.28 Filter Class Reference

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 \(Model::Filter::FilterList &list\)](#)
- [void applyToVideo \(Model::AVVideo &target, Model::AVVideo &video\)](#)

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::Filter::FilterList & *list*)

Constructor.

Parameters

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

4.29.3 Member Function Documentation

4.29.3.1 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.

4.30 FilterConfigurationBox Class Reference

Public Member Functions

- [FilterConfigurationBox \(GUI::Player::QWidget *parent\)](#)
- [void setFilter \(Model::Filter::Filter &filter\)](#)
- [Model::Filter::Filter * getFilter \(\)](#)

Protected Attributes

- [Model::Filter::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::Player::QWidget * parent`)

Constructor.

4.30.3 Member Function Documentation

4.30.3.1 Model::Filter::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 & filter)

Sets the filter the filterbox is responsible for.

Parameters

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

4.30.4 Field Documentation

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

4.31 FilterConfigurationLoader Class Reference

Public Member Functions

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

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

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

4.31.3 Member Function Documentation

4.31.3.1 Model::Filter::FilterList getConfigurations ()

Loads the filterlist.

Returns

The loaded filterlist.

4.32 FilterConfigurationSaver Class Reference

Public Member Functions

- [FilterConfigurationSaver](#) (QString *file*, Model::Filter::FilterList &*filterList*)
- void [save \(\)](#)

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::Filter::FilterList & *filterList*)

Constructor.

Parameters

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

4.32.3 Member Function Documentation

4.32.3.1 void [save \(\)](#)

Saves the filterlist.

4.33 FilterContainerTab Class Reference

Public Member Functions

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

4.33.1 Detailed Description

This class shows all the selectable filters.

4.33.2 Constructor & Destructor Documentation

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

Constructor.

4.33.3 Member Function Documentation

4.33.3.1 void addFilter (Model::Filter filter)

Adds a selectable filter.

Parameters

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

4.33.3.2 void setParentTab (GUI::FilterTab & parent)

Sets the parent tab.

Parameters

<i>parent</i>	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.
-------------------	------------------------

4.34 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::Filter::FilterApplier * list](#)
- [UndoRedo::FilterReset * filterList](#)
- [UndoRedo::LoadFilterconfig * oldList](#)
- [UndoRedo::LoadFilterconfig * newList](#)

4.34.1 Detailed Description

This class contains a filter configuration. Every filter can only be once in thi slist.

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

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

4.34.3.2 Model::Filter::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::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 Model::Filter::FilterApplier* list

4.34.4.3 UndoRedo::LoadFilterconfig* newList

4.34.4.4 UndoRedo::LoadFilterconfig* oldList

4.35 FilterReset Class Reference

Public Member Functions

- [FilterReset \(GUI::FilterTab *filterTab, Model::Filter::FilterList filterList\)](#)
- void [undo \(\)](#)
- void [redo \(\)](#)

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::Filter::FilterList *filterList*)

Constructor.

Parameters

<i>filterTab</i>	The filtertab to operate on.
<i>filterList</i>	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.

4.36 FilterTab Class Reference

Public Member Functions

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

Data Fields

- `UndoRedo::LoadFilterVideo * filterTab`

4.36.1 Detailed Description

This class is the tab to filter videos.

4.36.2 Constructor & Destructor Documentation

4.36.2.1 `FilterTab (GUI::Player::QWidget * parent)`

Constructor.

4.36.3 Member Function Documentation

4.36.3.1 `Memento::FilterTabMemento getMemento ()`

Creates a memento which contains the state of this tab.

Returns

The created memento.

4.36.3.2 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.

4.36.3.3 void moveFilter (int old, int new)

Moves a filter in the filterlist.

Parameters

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

4.36.3.4 void removeFilter (string filterName)

Removes the filter with the name filterName in the filterlist.

Parameters

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

4.36.3.5 void resetFilters ()

Resets the filterlist.

4.36.3.6 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.
----------------	--

4.36.3.7 void setFilterList (Model::Filter::FilterList list)

Sets the filterlist.

Parameters

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

4.36.3.8 void setRawVideo (Model::YuvVideo video)

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

Parameters

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

4.36.3.9 void showPreview ()

Shows the 5 frame preview.

4.36.3.10 void showVideo ()

Shows the video with the applied filters.

4.36.4 Field Documentation**4.36.4.1 UndoRedo::LoadFilterVideo* filterTab****4.37 FilterTabMemento Class Reference****Public Member Functions**

- [FilterTabMemento \(\)](#)
- [Model::Filter::FilterList getFilterList \(\)](#)
- [void setFilterList \(Model::Filter::FilterList filterList\)](#)
- [bool getWasApplied \(\)](#)
- [void setWasApplied \(bool wasApplied\)](#)
- [int getDisplayedFrame \(\)](#)
- [void setDisplayedFrame \(int displayedFrame\)](#)
- [string getLoadedFile \(\)](#)
- [void setLoadedFile \(string loadedFile\)](#)

Data Fields

- [UndoRedo::LoadFilterVideo * memento](#)

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::Filter::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::Filter::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 UndoRedo::LoadFilterVideo* *memento*

4.38 FilterView Class Reference

Public Member Functions

- [FilterView \(GUI::Player::QWidget *parent\)](#)
- [void setFilter \(Model::Filter::Filter filter\)](#)
- [void setFilterTab \(GUI::FilterTab *filtertab\)](#)

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::Player::QWidget * *parent*)

Constructor.

4.38.3 Member Function Documentation

4.38.3.1 void setFilter (Model::Filter::Filter *filter*)

Sets the filter this view represents.

Parameters

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

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

Sets the tab this view is contained in.

Parameters

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

4.39 ForwardPlayer Class Reference

Public Member Functions

- [ForwardPlayer \(\)](#)
- void [setForwardControlPanel \(GUI::Player::ControlPanel panel\)](#)
- 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 \(\)](#)

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::Player::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](#).

4.40 FrameView Class Reference

Public Member Functions

- [FrameView \(QWidget *parent=0\)](#)
- [void setFrame \(QImage &frame\)](#)
- [void clear \(\)](#)

Protected Member Functions

- [void resizeEvent \(QResizeEvent *event\)](#)
- [void repaintEvent \(QPaintEvent *event\)](#)

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.
--------------	--------------------

4.41 GlobalControlPanel Class Reference

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

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

x	Value on the x-axes.
y	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 Model::EncodedVideo* greenHisto****4.42.4.4 Model::EncodedVideo* psnr****4.42.4.5 Model::EncodedVideo* redHisto****4.43 GraphWidget Class Reference****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 \(QMouseEvent *event\)](#)

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 Model::Graph* graph

4.43.4.4 `GUI::AnalysisBox*` `greenHistogramm`

4.43.4.5 `GUI::AnalysisBox*` `psnrGraph`

4.43.4.6 `GUI::AnalysisBox*` `redHistogramm`

4.44 GridFilter Class Reference

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 ()`

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.
----------------------	---------------------------------

4.45 GridFilterBox Class Reference

Public Member Functions

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

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::Player::QWidget * *parent*)

Constructor.

4.46 LoadAnalysisVideo Class Reference

Public Member Functions

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

- void [undo \(\)](#)
- void [redo \(\)](#)

Data Fields

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

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 [Memento::AnalysisTabMemento* memento](#)

4.47 LoadFilterconfig Class Reference

Public Member Functions

- [LoadFilterconfig \(GUI::FilterTab *filterTab, Model::Filter::FilterList oldList, Model::Filter::FilterList list\)](#)
- void [undo \(\)](#)
- void [redo \(\)](#)

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::Filter::FilterList oldList, Model::Filter::FilterList list`)

Constuctor.

Parameters

<i>filterTab</i>	The filtertab to operate on.
<i>oldList</i>	The filterlist before the config is loaded.
<i>list</i>	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.

4.48 LoadFilterVideo Class Reference

Public Member Functions

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

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.

4.49 MacroblockCalculator Class Reference

Public Member Functions

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

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::Player::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.
--------------------	---

4.50 MainWindow Class Reference

Public Member Functions

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

Data Fields

- `Model::Project * loadedProject`

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::Player::QWidget * parent)

Constructor.

4.50.3 Member Function Documentation

4.50.3.1 Memento::MainWindowMemento getMemento ()

Creates a memento which contains the state of the window.

Returns

The created memento.

4.50.3.2 Model::Project & getProject ()

Returns the project that is currently loaded.

Returns

The currently loaded project.

4.50.3.3 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.
----------------	---

4.50.4 Field Documentation

4.50.4.1 Model::Project* loadedProject

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 \(Memento::FilterTabMemento filterTabMemento\)](#)

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← MeMento</i>	The AnalysisTabMemento .
---------------------------------	--

4.51.3.5 void setFilterTabMemento (Memento::FilterTabMemento filterTabMemento)

Sets the [FilterTabMemento](#).

Parameters

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

4.51.3.6 void setSelectedTab (int selectedTab)

Sets the currently selected tab.

Parameters

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

4.52 MirrorFilter Class Reference

Public Member Functions

- [MirrorFilter \(\)](#)
- string [getFilterDescription \(\)](#)
- string [getName \(\)](#)
- Model::Filter::MirrorMode [getMode \(\)](#)
- void [setMode \(Model::Filter::MirrorMode mode\)](#)

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::Filter::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::Filter::MirrorMode mode)

Sets the MirrorMode.

Parameters

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

4.53 MirrorFilterBox Class Reference

Public Member Functions

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

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::Player::QWidget * *parent*)

Constructor.

4.54 MoveFilterDown Class Reference

Public Member Functions

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

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.

4.55 MoveFilterUp Class Reference

Public Member Functions

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

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.

4.56 NegativeFilter Class Reference

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

Public Member Functions

- [NoiseFilter \(\)](#)
- [string getFilterDescription \(\)](#)
- [Model::Filter::NoiseMode getMode \(\)](#)
- [void setMode \(Model::Filter::NoiseMode mode\)](#)
- [int getIntensity \(\)](#)
- [string getName \(\)](#)
- [void setIntensity \(int intensity\)](#)

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::Filter::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::Filter::NoiseMode mode)

Sets the NoiseMode.

Parameters

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

4.58 NoiseFilterBox Class Reference

Public Member Functions

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

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::Player::QWidget * *parent*)

Constructor.

4.59 PlainFilterBox Class Reference

Public Member Functions

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

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::Player::QWidget * *parent*)

Constructor.

4.60 Player Class Reference

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::Player::ControlPanel * players`
- `GUI::Player::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

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

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

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

Sets the speed.

Parameters

<i>speed</i>	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::Player::ControlPanel* masterPlayer****4.60.3.2 GUI::Player::ControlPanel* players**

4.61 PlayerControlPanel Class Reference

Public Member Functions

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

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 updateUi() [virtual]

Updates the ui of the control panel.

Implements [ControlPanel](#).

4.62 PosterFilter Class Reference

Public Member Functions

- [PosterFilter \(\)](#)
- string [getFilterDescription \(\)](#)
- int [getNumberOfColors \(\)](#)

- string [getName \(\)](#)
- void [setNumberOfColors \(int numberOfColors\)](#)

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.
-----------------------	---------------------------

4.63 PosterFilterBox Class Reference

Public Member Functions

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

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 \(GUI::Player::QWidget * parent \)](#)

Constructor.

4.64 PreviewControlPanel Class Reference

Public Member Functions

- [PreviewControlPanel \(GUI::Player::QWidget *parent=0\)](#)
- [void updateUi \(\)](#)

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 \(GUI::Player::QWidget * parent = 0 \)](#)

Constructor.

4.64.3 Member Function Documentation

4.64.3.1 [void updateUi \(\) \[virtual\]](#)

Updates the ui of the control panel.

Implements [ControlPanel](#).

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 ()`

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.
-------------	------------------------

4.66 ProjectReader Class Reference

Public Member Functions

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

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.

4.67 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

<i>p</i>	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 \(\)](#)

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

<i>referenceVideo</i>	The reference video for the psnr calculation.
<i>compareVideo</i>	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.

4.69 QCheckBox Class Reference**4.70 QComboBox Class Reference****4.71 QDialog Class Reference****4.72 QFrame Class Reference****4.73 QGraphicsView Class Reference****4.74 QMainWindow Class Reference****4.75 QUndoCommand Class Reference****4.76 QWidget Class Reference****4.77 RectangleFilter Class Reference****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 \(int opacity\)](#)

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.
----------	---------------------------------------

4.78 RectangleFilterBox Class Reference

Public Member Functions

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

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::Player::QWidget * *parent*)

Constructor.

4.79 RemoveFilter Class Reference

Public Member Functions

- [RemoveFilter \(GUI::FilterTab *filterTab, Model::Filter::Filter filter, int index\)](#)
- [void undo \(\)](#)
- [void redo \(\)](#)

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.

4.80 RemoveVideo Class Reference

Public Member Functions

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

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.

4.81 RGBDifferenceCalculator Class Reference

Public Member Functions

- [RGBDifferenceCalculator \(GUI::Player::Video &referenceVideo, GUI::Player::Video &video\)](#)
- [void calculateVideo \(GUI::Player::Video &target\)](#)

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::Player::Video & *referenceVideo*, GUI::Player::Video & *video*)

Constructor.

Parameters

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

4.81.3 Member Function Documentation

4.81.3.1 void calculateVideo (GUI::Player::Video & *target*)

Calculates the RGB difference between two videos.

Parameters

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

4.82 RGBFilter Class Reference

Public Member Functions

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

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::Filter::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::Filter::BasicColor color)

Sets the preserved color.

Parameters

color	The preserved color.
-------	----------------------

4.83 RGBFilterBox Class Reference

Public Member Functions

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

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::Player::QWidget * parent`)

Constructor.

4.84 RGBHistogrammCalculator Class Reference

Public Member Functions

- void `RGBHistogrammCalculator (GUI::Player::Video &video)`
- void `calculate ()`
- `Model::Graph getRedHistogramm ()`
- `Model::Graph getGreenHistogramm ()`
- `Model::Graph getBlueHistogramm ()`

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::Player::Video & video`)

Constructor.

Parameters

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

4.85 RotationFilter Class Reference

Public Member Functions

- `RotationFilter ()`
- string `getFilterDescription ()`
- int `getAngle ()`
- string `getName ()`
- void `setAngle (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.
--------------	-------------------------

4.86 RotationFilterBox Class Reference

Public Member Functions

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

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 \(GUI::Player::QWidget * parent \)](#)

Constructor.

4.87 SaturationFilter Class Reference

Public Member Functions

- [SaturationFilter \(\)](#)
- string [getFilterDescription \(\)](#)
- int [getIntensity \(\)](#)
- string [getName \(\)](#)
- void [setIntensity \(int intensity\)](#)

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,
------------------	--------------------

4.88 SaturationFilterBox Class Reference

Public Member Functions

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

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::Player::QWidget * *parent*)

Constructor.

4.89 ScaleFilter Class Reference

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)`

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.
--------------	----------------

4.90 ScaleFilterBox Class Reference

Public Member Functions

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

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::Player::QWidget * *parent*)

Constructor.

4.91 SepiaFilter Class Reference

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

Public Member Functions

- [SharpnessFilter\(\)](#)
- string [getFilterDescription\(\)](#)
- int [getIntensity\(\)](#)
- string [getName\(\)](#)
- void [setIntensity\(int intensity\)](#)

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.
------------------	--------------------

4.93 SharpnessFilterBox Class Reference

Public Member Functions

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

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::Player::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 \(GUI::Player::VideoPlayer &player\)](#)
- [bool isPlaying \(\)](#)
- [void removePlayer \(GUI::Player::VideoPlayer &player\)](#)

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

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

4.94.3 Member Function Documentation

4.94.3.1 void addPlayer (GUI::Player::VideoPlayer & *player*)

Adds a player.

Parameters

<i>player</i>	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::Player::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.

4.95 UndoStack Class Reference

Static Public Member Functions

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

4.95.1 Detailed Description

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

4.95.2 Member Function Documentation

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

Returns the undo stack to operate on.

Returns

The undo stack.

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`

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

<code>index</code>	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 Model::EncodedVideo* macroblockVideo

4.96.4.3 Model::EncodedVideo* rgbDiffVideo

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::Player::Video > [convertAVVideoToVideo](#) (Model::AVVideo &video)
- static unique_ptr< AVFrame > [convertQImageToAVFrame](#) (QImage &imgae)
- static unique_ptr< Model::AVVideo > [convertVideoToAVVideo](#) (GUI::Player::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::Player::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::Player::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 \(\)](#)

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.

4.99 VideoPlayer Class Reference

Public Member Functions

- [VideoPlayer \(\)](#)
- void [addView \(GUI::Player::FrameView &view\)](#)
- void [removeView \(GUI::Player::FrameView &view\)](#)
- void [setVideo \(GUI::Player::Video &video\)](#)
- [GUI::Player::Video * getVideo \(\)](#)
- void [setTimer \(shared_ptr< GUI::Player::Timer > timer\)](#)
- void [clearTimer \(\)](#)
- int [getFps \(\)](#)
- void [setMasterControlPanel \(GUI::Player::ControlPanel &controlPanel\)](#)
- 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::Player::FrameView * > views`

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

4.100 VintageFilter Class Reference

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

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

Public Member Functions

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

Static Public Member Functions

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

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< GUI::Player::Video > read \(\) \[virtual\]](#)

Reads the file in.

Returns

The complete video.

Implements [YuvFileReader](#).

4.102.3.2 [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.

4.103 Yuv411FileSaver Class Reference

Public Member Functions

- [Yuv411FileSaver \(QString filename, GUI::Player::Video &video, Utility::Compression compression\)](#)
- void [save \(\)](#)

Static Public Member Functions

- static [Utility::Yuv411Vector rgb888ToYuv411](#) (QRgb pixel1, QRgb pixel2, QRgb pixel3, QRgb pixel4)

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 \(QString filename, GUI::Player::Video & video, Utility::Compression compression \)](#)

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](#).

4.104 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

Public Member Functions

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

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

Reads the file in.

Returns

The complete video.

Implements [YuvFileReader](#).

4.105.2.2 `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.

4.106 Yuv420FileSaver Class Reference

Public Member Functions

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

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::Player::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](#).

4.107 Yuv422FileReader Class Reference

Public Member Functions

- [Yuv422FileReader \(QString filename, int width, int height, Utility::Compression compression\)](#)
- unique_ptr< GUI::Player::Video > [read \(\)](#)

Static Public Member Functions

- static vector< QRgb > [yuv422ToRgb888 \(Utility::Yuv422Vector vector\)](#)

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

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

4.107.3 Member Function Documentation

4.107.3.1 `unique_ptr< GUI::Player::Video > read () [virtual]`

Reads the file in.

Returns

The complete video.

Implements [YuvFileReader](#).

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

Converts a [Yuv422Vector](#) the Rgb888 pixels

Parameters

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

Returns

The computed rgb888 pixels.

4.108 Yuv422FileSaver Class Reference

Public Member Functions

- [Yuv422FileSaver](#) (QString *filename*, [GUI::Player::Video](#) &*video*, Utility::Compression *compression*)
- void [save](#) ()

Static Public Member Functions

- static [Utility::Yuv422Vector](#) [rgb888ToYuv422](#) (QRgb *pixel1*, QRgb *pixel2*)

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::Player::Video & video, Utility::Compression compression)`

Constructor.

Parameters

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

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

Returns

The [Yuv422Vector](#).

4.108.3.2 `void save () [virtual]`

Saves the video to the file.

Implements [YuvFileSaver](#).

4.109 Yuv422Vector Class Reference

4.109.1 Detailed Description

A [Yuv422Vector](#).

4.110 Yuv444FileReader Class Reference

Public Member Functions

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

Static Public Member Functions

- static QRgb [yuv444ToRgb888](#) ([Utility::Yuv444Vector](#) vector)

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<GUI::Player::Video>](#) [read](#) () [virtual]

Reads the file in.

Returns

The complete video.

Implements [YuvFileReader](#).

4.110.3.2 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.

4.111 Yuv444FileSaver Class Reference

Public Member Functions

- [Yuv444FileSaver](#) ([QString](#) *filename*, [GUI::Player::Video](#) &*video*, [Utility::Compression](#) *compression*)
- void [save](#) ()

Static Public Member Functions

- static [Utility::Yuv444Vector rgb888ToYuv444 \(QRgb pixel1\)](#)

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 \(QString filename, GUI::Player::Video & video, Utility::Compression compression \)](#)

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](#).

4.112 Yuv444Vector Class Reference

4.112.1 Detailed Description

A [Yuv444Vector](#).

4.113 YuvFileDialog Class Reference

Public Member Functions

- [YuvFileDialog \(GUI::Player::QWidget *parent=0\)](#)

- `QString getFilename ()`
- `void show ()`
- `bool wasSuccessfull ()`

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::Player::QWidget * parent = 0`)

Constructor.

4.113.3 Member Function Documentation

4.113.3.1 `QString getFilename ()`

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

Returns

Absolute path to the user chosen file.

4.113.3.2 `void show ()`

Shows the dialog.

4.113.3.3 `bool wasSuccessfull ()`

Whether the user clicked ok or cancel.

Returns

True if the user clicked ok. false otherwise.

4.114 YuvFileReader Class Reference

Public Member Functions

- `YuvFileReader (QString filename, int width, int height)`
- `virtual unique_ptr< GUI::Player::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::Player::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

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

<code>value</code>	The value to clamp.
--------------------	---------------------

Returns

The clamped value.

4.114.3.2 virtual unique_ptr<GUI::Player::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::Player::Video> `video` [protected]

4.114.4.4 int `width` [protected]

4.115 YuvFileSaver Class Reference

Public Member Functions

- `YuvFileSaver` (QString *filename*, GUI::Player::Video &*video*)
- virtual void `save` ()=0

Protected Attributes

- int `width`
- int `height`
- GUI::Player::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::Player::Video & *video*)

Constructor.

Parameters

<code>filename</code>	Absolute path to the file to save to.
<code>video</code>	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::Player::Video* `video` [protected]

4.115.4.5 int `width` [protected]

4.116 YuvInfoDialog Class Reference

Public Member Functions

- `YuvInfoDialog (GUI::Player::QWidget *parent)`
- `int getFps ()`
- `int getWidth ()`
- `int getHeight ()`
- `Utility::Compression getCompression ()`
- `GUI::Player::PixelScheme getPixelScheme ()`
- `bool wasSuccessful ()`
- `void show ()`

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::Player::QWidget * parent)`

Constructor.

4.116.3 Member Function Documentation

4.116.3.1 `Utility::Compression getCompression ()`

Returns the compression the user entered.

Returns

The compression.

4.116.3.2 `int getFps ()`

Returns the fps the user entered.

Returns

The fps.

4.116.3.3 `int getHeight ()`

Returns the height the user entered.

Returns

The height.

4.116.3.4 GUI::Player::PixelSheme getPixelSheme()

Returns the pixelsheme the user entered.

Returns

The pixelsheme.

4.116.3.5 int getWidth()

Returns the width the user entered.

Returns

The width.

4.116.3.6 void show()

Shows the dialog.

4.116.3.7 bool wasSuccessful()

Whether the user clicked ok or cancel.

Returns

True if the user clicked ok. false otherwise.

4.117 YuvVideo Class Reference

Public Member Functions

- [YuvVideo \(QString path, GUI::Player::PixelSheme type, int width, int height, int fps\)](#)
- [QString getPath \(\)](#)
- [Utility::Compression getCompression \(\)](#)
- [YuvType getYuvType \(\)](#)
- [Model::AVVideo & getAvVideo \(\)](#)
- [GUI::Player::Video & getVideo \(\)](#)

Data Fields

- [UndoRedo::LoadFilterVideo * video](#)
- [Utility::Compression * compression](#)
- [GUI::Player::Video * displayVideo](#)
- [Model::AVVideo * avVideo](#)

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::Player::PixelScheme type, int width, int height, int fps`)

Constructor.

Parameters

<i>path</i>	Path to the yuv file.
<i>type</i>	Pixelscheme of the yuv video.
<i>width</i>	Width of the video.
<i>height</i>	Height of the video.
<i>fps</i>	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::Player::Video & getVideo()

Returns the Video.

Returns

The Video.

4.117.3.5 YuvType getYuvType()

Returns the pixel sheme of the video.

Returns

The pixel scheme.

4.117.4 Field Documentation

4.117.4.1 **Model::AVVideo*** avVideo

4.117.4.2 **Utility::Compression*** compression

4.117.4.3 **GUI::Player::Video*** displayVideo

4.117.4.4 **UndoRedo::LoadFilterVideo*** video

4.118 ZoomFilter Class Reference

Public Member Functions

- [ZoomFilter \(\)](#)
- string [getName \(\)](#)
- string [getFilterDescription \(\)](#)
- int [getIntensity \(\)](#)
- void [setIntensity \(int intensity\)](#)

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

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

4.119 ZoomFilterBox Class Reference

Public Member Functions

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

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::Player::QWidget * parent)`

Constructor.

Anhang

Implementierungsplan

Projektplan

