ENTWURFSHEFT

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

Contents

Chapter 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.

Chapter 2

Data Structure Documentation

2.1 AddFilter Class Reference

Public Member Functions

- AddFilter ()
- void undo ()
- void redo ()

2.1.1 Constructor & Destructor Documentation

2.1.1.1 AddFilter ( )

Constructor

2.1.2 Member Function Documentation

2.1.2.1 void redo ( )

Adds a filter to the filterlist

2.1.2.2 void undo ( )

Removes added filter from the filterlist

2.2 AddVideo Class Reference

Public Member Functions

- AddVideo ()
- void undo ()
- void redo ()

2.2.1 Constructor & Destructor Documentation

2.2.1.1 AddVideo ( )

Constuctor

2.2.2 Member Function Documentation

2.2.2.1 void redo ( )

adds a video to the Analysis tab

2.2.2.2 void undo ( )

removes the added video from the analysis tab

2.3 AnalysisBox Class Reference

Public Member Functions

- **AnalysisBox** (GUI::QtGui::QWidget__10 ∗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** (QString path)

2.3.1 Detailed Description

Shows the Analysis of a single video

2.3.2 Member Function Documentation

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

Shows the Analysis of a single video

2.4 AnalysisBoxContainer Class Reference

Public Member Functions

- **AnalysisBoxContainer** (GUI::QtGui::QWidget__10 ∗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:)
- void setControlPanel (Player::GlobalControlPanel ∗panel)
- void showMacroBlockVideos ()

- void showRGBDifferenceVideos ()
- void removeBox (GUI::AnalysisBox &box)

2.4.1 Detailed Description

contains and manages the AnalysisBoxes

2.4.2 Member Function Documentation

2.4.2.1 void addVideo (QString *path*)

creates a Analysis box for the video and adds it to boxes

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

removes box from the list

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

set panel of the raw video and tells analysisBox to do the same with their video

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

sets the rawVideo to video

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

set timer of the raw video and tells analysisBox to do the same with their video

2.4.2.6 void showMacroBlockVideos ( )

tells all AnalysisBoxes to show the macro block video

2.4.2.7 void showRGBDifferenceVideos ( )

tells all AnalysisBoxes to show the RGBDiff video

2.5 AnalysisBoxContainer2 Class Reference

Public Member Functions

- void **analysisBoxContainer** (Memento::AnalysisBoxContainerMemento anaBoxConMemento)
- void **changeAnalyseTyp** (int typ)
- QList< AnalysisBox2 > getBoxesList ()
- analysisBoxContainerMemento **getMemento** ()
- QPushButton **getAddVidButton** ()

2.5.1 Member Function Documentation

2.5.1.1 QList< AnalysisBox2 > getBoxesList ( )

returns all Analyis boxes known to this container

2.6 AnalysisBoxContainerMemento Class Reference

Public Member Functions

- void analyseBoxMemento ()
- vector< Memento::AnalysisBoxMemento > getAnalysisBoxList ()
- void setAnalysisBoxList (vector< Memento::AnalysisBoxMemento > analyseBoxList:std:)

2.6.1 Detailed Description

This class is the memento for the AnalysisBoxContainer.

2.6.2 Member Function Documentation

2.6.2.1 void analyseBoxMemento ( )

Constructor.

2.6.2.2 vector< Memento::AnalysisBoxMemento > getAnalysisBoxList ( )

Returns a list of AnalysisBox mementos.

Returns

a lists of AnalysisBoxMemento

2.6.2.3 void setAnalysisBoxList (vector< Memento::AnalysisBoxMemento > analyseBoxList:std:)

Sets the list of AnalysisBoxMemento

Parameters

analyseBox↩ List:std,:	the list of the mementos

2.7 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)
- QImage getPsnr ()
- void setPsnr (QImage psnr)
- QImage getBitrate ()
- void setBitrate (QImage bitrate)

2.7.1 Detailed Description

This class is the memento for the AnalysisBox.

2.7.2 Constructor & Destructor Documentation

2.7.2.1 AnalysisBoxMemento ( )

Constrictor.

2.7.3 Member Function Documentation

2.7.3.1 QImage getBitrate ( )

Returns an image of the bitrate graph.

Returns

an image of the bitrate graph

2.7.3.2 QString getComment ( )

Returns the user comment.

Returns

the user comment

2.7.3.3 GUI::Player::Video * getMacroVideo ( )

Returns the macroblock video.

Returns

the macroblock video

2.7.3.4 QImage getPsnr ( )

Returns an image of the psnr graph.

Returns

an image of the psnr graph

2.7.3.5 GUI::Player::Video * getRgbDiffVideo ( )

Returns the rgb difference video.

Returns

the rgb difference video

2.7.3.6 QString getVideoPath ( )

Returns the path to the video.

Returns

absolute path to the video

2.7.3.7 void setBitrate (QImage *bitrate*)

Sets the image for the bitrate graph.

Parameters

bitrate	the image of the bitrate gaph

2.7.3.8 void setComment (QString *comment*)

Sets the user comment.

Parameters

comment	the user comment

2.7.3.9 void setMacroVideo (GUI::Player::Video * *macroVideo*)

Sets the macroblock video.

Parameters

macroVideo	the macroblock video

2.7.3.10 void setPsnr (QImage *psnr*)

Sets the image of the psnr graph.

Parameters

psnr	the image of the psnr graph

2.7.3.11 void setRgbDiffVideo (GUI::Player::Video * *rgbDiffVideo*)

Sets the rgb difference video.

Parameters

rgbDiffVideo	the rgb difference video

2.7.3.12 void setVideoPath (QString *videoPath*)

Sets the path to the video.

Parameters

videoPath	absolute path to the video

2.8 AnalysisTab Class Reference

Public Member Functions

- **AnalysisTab** (GUI::QtGui::QWidget__10 ∗parent)
- Memento::AnalysisTabMemento **getMemento** ()
- void **restore** (Memento::AnalysisTabMemento memento)

Data Fields

- GUI::Player::FrameView ∗ **rawVideoView**

2.8.1 Detailed Description

the tab that shows videos and analyses them

2.9 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 analysisBoxContainer↩ Memento)

2.9.1 Detailed Description

This class is the memento for the analysis tab.

2.9.2 Constructor & Destructor Documentation

2.9.2.1 AnalysisTabMemento ( )

Constructor.

2.9.3 Member Function Documentation

2.9.3.1 Memento::AnalysisBoxContainerMemento getAnalysisBoxContainerMemento ( )

Returns the memento of the AnalysisBoxContainer.

Returns

the memento of the AnalysisBoxContainer

2.9.3.2 int getCurrentlyShownAnalysisVideo ( )

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

Returns

the currently shown analysis video

2.9.3.3 float getCurrentSpeed ( )

Returns the current speed of the player.

Returns

the current speed of the player

2.9.3.4 int getCurrentVideoPosition ( )

Returns the current position the player is at.

Returns

the current position of the player

2.9.3.5 void setAnalysisBoxContainerMemento (Memento::AnalysisBoxContainerMemento *analysisBoxContainerMemento*)

Sets the memento of the AnalysisBoxContainer.

Parameters

analysisBox↩ Container↩ Memento	the memento of the AnalysisBoxContainer

2.9.3.6 void setCurrentlyShownAnalysisVideo (int *currentlyShownAnalysisVideo*)

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

Parameters

currently↩ Shown↩ AnalysisVideo	the currrently shown analysis video

2.9.3.7 void setCurrentSpeed (float *currentSpeed*)

Sets the current speed of the player.

Parameters

currentSpeed	the current speed of the player

2.9.3.8 void setCurrentVideoPosition (int *currentVideoPosition*)

Sets the current position of the player.

Parameters

currentVideo↩ Position	the current position of the player

2.10 ApplyFilter Class Reference

Public Member Functions

- ApplyFilter ()
- void undo ()
- void redo ()

2.10.1 Constructor & Destructor Documentation

2.10.1.1 ApplyFilter ( )

Constuctor

2.10.2 Member Function Documentation

2.10.2.1 void redo ( )

applies filter to the video

2.10.2.2 void undo ( )

resets to the state before filters were applied

2.11 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:)
- void removeFrame (int index)
- void insertFrames (int index=-1, vector< std::unique_ptr< AVFrame > > &frames:std:)
- int getNumberOfFrames ()

Data Fields

- Model::EncodedVideo * **avVideo**

2.11.1 Constructor & Destructor Documentation

2.11.1.1 AVVideo (int *fps*, int *width*, int *height*)

Constructor.

Parameters

fps	the fps the video should be played at
width	the width of the video
height	the height of the video

2.11.2 Member Function Documentation

2.11.2.1 int getFps ( )

Returns the fps of the video.

Returns

fps of the video

2.11.2.2 AVFrame * getFrame (int *index*)

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

Parameters

index	the index of the frame to return

2.11.2.3 int getHeight ( )

Returns the height of the video.

Returns

the height of the video

2.11.2.4 int getNumberOfFrames ( )

Returns the number of frames in the video.

Returns

the number of frames in the video

2.11.2.5 int getWidth ( )

Returns the width of the video.

Returns

the width of the video

2.11.2.6 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

index	The index to insert the frame at
frame:std,:	The frame to insert

2.11.2.7 void insertFrames (int *index* = −1, vector< std::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

index	the index to insert the frames at
frames:std,:	the frames to insert

2.11.2.8 void removeFrame (int *index*)

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

Parameters

index	the index of the frame to remove

2.12 BitrateCalculator Class Reference

Public Member Functions

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

2.12.1 Detailed Description

This class calculates the bitrate of a video.

2.12.2 Constructor & Destructor Documentation

2.12.2.1 BitrateCalculator (Model::AVVideo & *video*)

Constructor.

Parameters

video	The video of which the bitrate is calculated.

2.12.3 Member Function Documentation

2.12.3.1 Model::Graph calculate ( )

Calculates the bitrate graph.

Returns

the bitrate graph

2.13 BlackWhiteFilter Class Reference

Public Member Functions

- std::string getName ()
- std::string getFilterDescription ()

2.13.1 Detailed Description

Converts the video to black and white

2.13.2 Member Function Documentation

2.13.2.1 std::string getFilterDescription ( ) [virtual]

Returns the description of the filter

Implements Filter.

2.13.2.2 std::string getName ( ) [virtual]

Returns the name of the filter

Implements Filter.

2.14 BlackWhiteFilterBox Class Reference

Public Member Functions

- **BlackWhiteFilterBox** (GUI::QtGui::QWidget__10 *parent)
- virtual void **setFilter** (Model::Filter::Filter &filter)
- virtual Model::Filter::Filter * **getFilter** ()

2.15 BlendingFilter Class Reference

Public Member Functions

- std::string getFilterDescription ()
- bool **getInBlend** ()
- void **setInBlend** (bool inBlend)
- int **getStartFrame** ()
- void **setStartFrame** (int startFrame)
- std::string getName ()
- int **getEndFrame** ()
- void **setEndFrame** (int endFrame)

2.15.1 Detailed Description

Inserts black blending into the video

2.15.2 Member Function Documentation

2.15.2.1 std::string getFilterDescription ( ) [virtual]

Returns the description of the filter

Implements Filter.

2.15.2.2 std::string getName ( ) [virtual]

Returns the name of the filter

Implements Filter.

2.16 BlendingFilterBox Class Reference

Public Member Functions

- **BlendingFilterBox** (GUI::QtGui::QWidget__10 *parent)
- virtual void **setFilter** (Model::Filter::Filter &filter)
- virtual Model::Filter::Filter * **getFilter** ()

2.17 BlurFilter Class Reference

Public Member Functions

- std::string getFilterDescription ()
- bool **getPreserveEdges** ()
- void **setPreserveEdges** (bool preserveEdges)
- std::string getName ()
- int **getIntensity** ()
- void **setIntensity** (int intensity)

2.17.1 Detailed Description

Blurs the video

2.17.2 Member Function Documentation

2.17.2.1 std::string getFilterDescription ( ) [virtual]

Returns the description of the filter

Implements Filter.

2.17.2.2 std::string getName ( ) [virtual]

Returns the name of the filter

Implements Filter.

2.18 BlurFilterBox Class Reference

Public Member Functions

- **BlurFilterBox** (GUI::QtGui::QWidget__10 *parent)
- virtual void **setFilter** (Model::Filter::Filter &filter)
- virtual Model::Filter::Filter * **getFilter** ()

2.19 BorderFilter Class Reference

Public Member Functions

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

2.19.1 Detailed Description

Inserts border into the video

2.19.2 Member Function Documentation

2.19.2.1 std::string getFilterDescription ( ) [virtual]

Returns the description of the filter

Implements Filter.

2.19.2.2 std::string getName ( ) [virtual]

Returns the name of the filter

Implements Filter.

2.20 BorderFilterBox Class Reference

Public Member Functions

- **BorderFilterBox** (GUI::QtGui::QWidget__10 *parent)
- virtual void **setFilter** (Model::Filter::Filter &filter)
- virtual Model::Filter::Filter * **getFilter** ()

2.21 BrightnessFilter Class Reference

Public Member Functions

- std::string getFilterDescription ()
- int **getIntensity** ()
- std::string getName ()
- void **setIntensity** (int intensity)

2.21.1 Detailed Description

Adjusts the video brightness

2.21.2 Member Function Documentation

2.21.2.1 std::string getFilterDescription ( ) [virtual]

Returns the description of the filter

Implements Filter.

2.21.2.2 std::string getName ( ) [virtual]

Returns the name of the filter

Implements Filter.

2.22 BrightnessFilterBox Class Reference

Public Member Functions

- **BrightnessFilterBox** (GUI::QtGui::QWidget__10 *parent)
- virtual void **setFilter** (Model::Filter::Filter &filter)
- virtual Model::Filter::Filter * **getFilter** ()

2.23 ColorbalanceFilter Class Reference

Public Member Functions

- std::string getFilterDescription ()
- 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)
- bool **getDarkPixels** ()
- void **setDarkPixels** (bool darkPixels)
- virtual std::string getName ()

2.23.1 Detailed Description

Adjusts the colorbalance of the video

2.23.2 Member Function Documentation

2.23.2.1 std::string getFilterDescription ( ) [virtual]

Returns the description of the filter

Implements Filter.

2.23.2.2 std::string getName ( ) [virtual]

Returns the name of the filter

Implements Filter.

2.24 ColorbalanceFilterBox Class Reference

Public Member Functions

- **ColorbalanceFilterBox** (GUI::QtGui::QWidget__10 *parent)
- virtual void **setFilter** (Model::Filter::Filter &filter)
- virtual Model::Filter::Filter * **getFilter** ()

2.25 ContrastFilter Class Reference

Public Member Functions

- std::string getFilterDescription ()
- void **setIntensity** (int intensity)
- int **getIntensity** ()
- std::string getName ()

2.25.1 Detailed Description

Adjusts the contrast of the video

2.25.2 Member Function Documentation

2.25.2.1 std::string getFilterDescription ( ) [virtual]

Returns the description of the filter

Implements Filter.

2.25.2.2 std::string getName ( ) [virtual]

Returns the name of the filter

Implements Filter.

2.26 ContrastFilterBox Class Reference

Public Member Functions

- **ContrastFilterBox** (GUI::QtGui::QWidget__10 *parent)
- virtual void **setFilter** (Model::Filter::Filter &filter)
- virtual Model::Filter::Filter * **getFilter** ()

2.27 ControlPanel Class Reference

Public Member Functions

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

Data Fields

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

Protected Attributes

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

2.27.1 Detailed Description

This class is the base class for control panels.

2.27.2 Constructor & Destructor Documentation

2.27.2.1 ControlPanel ( )

Constructor.

2.27.3 Member Function Documentation

2.27.3.1 void addVideoPlayer (GUI::Player & *player*)

Adds the video player the list of players to notify.

Parameters

player	The player to add to the list

2.27.3.2 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

player	the master video player

2.28 EdgeFilter Class Reference

Public Member Functions

- std::string getFilterDescription ()
- std::string getName ()

2.28.1 Detailed Description

Only shows the borders and places a black overlay over the rest

2.28.2 Member Function Documentation

2.28.2.1 std::string getFilterDescription ( ) [virtual]

Returns the description of the filter

Implements Filter.

2.28.2.2 std::string getName ( ) [virtual]

Returns the name of the filter

Implements Filter.

2.29 EdgeFilterBox Class Reference

Public Member Functions

- **EdgeFilterBox** (GUI::QtGui::QWidget__10 *parent)
- virtual void **setFilter** (Model::Filter::Filter &filter)
- virtual Model::Filter::Filter * **getFilter** ()

2.30 EncodedVideo Class Reference

Public Member Functions

- **EncodedVideo** (QString path)
- QString **getPath** ()
- int **getFileSize** ()
- int **getNumberOfColors** ()
- QString **getCodec** ()
- Model::Graph & **getBitrate** ()
- Model::Graph & **getPsnr** ()
- void **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** ()

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 * **blueHiso**

2.31 Filter Class Reference

Public Member Functions

- virtual std::string getFilterDescription ()=0
- virtual std::string getName ()=0

2.31.1 Detailed Description

Abstract class to provide superclass for filters.

2.31.2 Member Function Documentation

2.31.2.1 virtual std::string getFilterDescription ( ) [pure virtual]

Returns the description of the filter

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

2.31.2.2 std::string getName ( ) [pure virtual]

Returns the name of the filter

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

2.32 FilterApplier Class Reference

Public Member Functions

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

2.32.1 Detailed Description

Applies filters of a given FilterList to the video

2.32.2 Constructor & Destructor Documentation

2.32.2.1 FilterApplier (Model::Filter::FilterList & *list*)

Constructor.

Parameters

list	the list with the filters to apply

2.32.3 Member Function Documentation

2.32.3.1 void applyToVideo (Model::AVVideo & *target,* Model::AVVideo & *video*)

Applies the given filters to the video.

Parameters

target	the video to which the new frames are added to
video	the video to apply the filters on

2.33 FilterConfiguration Class Reference

Public Member Functions

- void **add** (Model::Filter::Filter filter)
- void **remove** (Model::Filter::Filter filter)
- void **remove** (int index)
- void **move** (int old_index, int new_index)
- void **get** (int index)
- void **move** (Model::Filter::Filter filter, int new_index)
- int **size** ()

2.34 FilterConfigurationBox Class Reference

Public Member Functions

- **FilterConfigurationBox** (GUI::QtGui::QWidget__10 *parent)
- virtual void **setFilter** (Model::Filter::Filter &filter)=0
- virtual Model::Filter::Filter * **getFilter** ()=0

2.35 FilterConfigurationLoader Class Reference

Public Member Functions

- FilterConfigurationLoader (QString path)
- Model::Filter::FilterList getConfiguration ()

2.35.1 Detailed Description

This class can load a Filterlist.

2.35.2 Constructor & Destructor Documentation

2.35.2.1 FilterConfigurationLoader (QString *path*)

Constructor.

Parameters

path	the path to the filerlist to load

2.35.3 Member Function Documentation

2.35.3.1 Model::Filter::FilterList getConfiguration ( )

Loads the filterlist.

Returns

the loaded filterlist

2.36 FilterConfigurationLoader2 Class Reference

2.37 FilterConfigurationSaver Class Reference

2.38 FilterConfigurationSaver_ Class Reference

Public Member Functions

- FilterConfigurationSaver_ (QString file, Model::Filter::FilterList &filterList)
- void save ()

2.38.1 Detailed Description

This class can save a filterlist.

2.38.2 Constructor & Destructor Documentation

2.38.2.1 FilterConfigurationSaver_ (QString *file,* Model::Filter::FilterList & *filterList*)

Constructor.

Parameters

file	absolute path to the file to save to
filterList	the filterlist to save

2.38.3 Member Function Documentation

2.38.3.1 void save ( )

Saves the filterlist.

2.39 FilterContainerTab Class Reference

Public Member Functions

- **FilterContainerTab** (GUI::QtGui::QWidget__10 *parent)
- void addFilter (Model::Filter::Filter filter)
- void **setParentTab** (GUI::FilterTab &parent)
- void uncheck (string filterName)

2.39.1 Detailed Description

shows all FilterViews

2.39.2 Member Function Documentation

2.39.2.1 void addFilter (Model::Filter::Filter *filter*)

Slot: connected to to every FilterView.checked(filter:Filter)

2.39.2.2 void uncheck (string *filterName*)

searches for the filterView with the filter filterName and unchecks it

2.40 FilterList Class Reference

Public Member Functions

- Model::Filter::Filter * **getFilterByName** (string name:std:)
- void **removeFilter** (string name:std:)
- void **moveFilter** (int oldPosition, int newPosition)
- void **removeFilter** (int position)
- void **addFilter** (string name:std:)
- void **getFIlterByIndex** (int index)

Data Fields

- Model::Filter::FilterApplier * **list**

2.41 FilterReset Class Reference

Public Member Functions

- FilterReset ()
- void undo ()
- void redo ()

2.41.1 Constructor & Destructor Documentation

2.41.1.1 FilterReset ( )

Constuctor

2.41.2 Member Function Documentation

2.41.2.1 void redo ( )

clears the filter configurations and the filter list

2.41.2.2 void undo ( )

loads the filterlist and filter configuration to the state it was before the reset

2.42 FilterTab Class Reference

Public Member Functions

- **FilterTab** (GUI::QtGui::QWidget__10 *parent)
- Memento::FilterTabMemento **getMemento** ()
- void **restore** (Memento::FilterTabMemento memento)
- void addFilter (Model::Filter::Filter filter)
- void removeFilter (string filterName)

2.42.1 Member Function Documentation

2.42.1.1 void addFilter (Model::Filter::Filter *filter*)

adds a filter to list_filterList

2.42.1.2 void removeFilter (string *filterName*)

removes the filter with the name filterName in list_filterList

2.43 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)

2.43.1 Detailed Description

This class is the memento for the FilterTab.

2.43.2 Constructor & Destructor Documentation

2.43.2.1 FilterTabMemento ( )

Constructor.

2.43.3 Member Function Documentation

2.43.3.1 int getDisplayedFrame ( )

Returns the currently displayed frame in the frame preview.

Returns

the currently displayed frame

2.43.3.2 Model::Filter::FilterList getFilterList ( )

Returns the list of the currently selected filters.

Returns

list of the selected filters

2.43.3.3 string getLoadedFile ( )

Returns the path to the currently loaded yuv file.

Returns

absolute path to the currently loaded yuv file

2.43.3.4 bool getWasApplied ( )

Whether the filter were already applied.

Returns

true if the filter wre already applied

2.43.3.5 void setDisplayedFrame (int *displayedFrame*)

Sets the currently displayed frame in the frame preview.

Parameters

displayedFrame	the currently displayed frame

2.43.3.6 void setFilterList (Model::Filter::FilterList *filterList*)

Sets the list of the currently selected filters.

Parameters

filterList	list of the selected filters

2.43.3.7 void setLoadedFile (string *loadedFile*)

Sets the path to the currently loaded yuv file.

Parameters

loadedFile	absolute path to the loaded yuv file

2.43.3.8 void setWasApplied (bool *wasApplied*)

Sets whether the filters were already aplied.

Parameters

wasApplied	true if the filter were already applied

2.44 FilterView Class Reference

Public Member Functions

- **FilterView** (GUI::QtGui::QWidget__10 ∗parent)
- void **setFilter** (Model::Filter::Filter filter)
- void **setFilterTab** (GUI::FilterTab ∗filtertab)

2.44.1 Detailed Description

represents a filter in the gui, shows a example of the filter and a checkbox as well as its name

2.45 ForwardPlayer Class Reference

Public Member Functions

- 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** ()

2.46 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)

2.46.1 Detailed Description

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

2.46.2 Constructor & Destructor Documentation

2.46.2.1 FrameView (QWidget ∗ *parent* = 0)

Constructor.

2.46.3 Member Function Documentation

2.46.3.1 void clear ( )

Clears the current frame so nothing get shown.

2.46.3.2 void repaintEvent (QPaintEvent ∗ *event*) [protected]

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

2.46.3.3 void resizeEvent (QResizeEvent ∗ *event*) [protected]

This method is called when the widget got resized.

2.46.3.4 void setFrame (QImage & *frame*)

Sets the frame to show.

Parameters

frame	the frame to show

2.47 GlobalControlPanel Class Reference

Public Member Functions

- void **play** ()
- void **pause** ()
- void **stop** ()
- void **nextFrame** ()
- void **previousFrame** ()
- void **setPosition** (int position)
- void **updateUi** ()

Data Fields

- GUI::AnalysisTab * **globalControlPanel**

Additional Inherited Members

2.48 Graph Class Reference

Public Member Functions

- Graph ()
- void setValue (int x, double y)
- void cut (int x)
- double getValue (int x)
- int getLength ()

Data Fields

- Model::EncodedVideo * **bitrate**
- Model::EncodedVideo * **psnr**
- Model::EncodedVideo * **redHisto**
- Model::EncodedVideo * **greenHisto**
- Model::EncodedVideo * **blueHiso**

2.48.1 Detailed Description

This class is a graph.

2.48.2 Constructor & Destructor Documentation

2.48.2.1 Graph ( )

Constructor.

2.48.3 Member Function Documentation

2.48.3.1 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.

2.48.3.2 int getLength ( )

Returns the length of the unbounded array. That many x/y values are saved.

2.48.3.3 double getValue (int *x*)

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

Parameters

x	The x value.

2.48.3.4 void setValue (int *x*, double *y*)

gives a y-value to a x-value. Later the point in the graph.

Parameters

x	value on the x-axes.
y	value on the y-axes.

2.49 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**

2.49.1 Detailed Description

This class is a widget to draw graphs.

2.49.2 Constructor & Destructor Documentation

2.49.2.1 GraphWidget ( )

Constructor.

2.49.3 Member Function Documentation

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

Draws a graph to the widget.

2.49.3.2 void setFillColor (QRgb *color*)

Determines the color of the area beneath the graph line.

Parameters

color	The color in which the area beneath the graph line is filled.

2.49.3.3 void setLineColor (QRgb *color*)

Determines the color of the graph line.

Parameters

color	The color in which the line is shown.

2.50 GridFilter Class Reference

Public Member Functions

- std::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)
- std::string getName ()

2.50.1 Detailed Description

Inserts a grid into the video as an overlay

2.50.2 Member Function Documentation

2.50.2.1 std::string getFilterDescription ( ) [virtual]

Returns the description of the filter

Implements Filter.

2.50.2.2 std::string getName ( ) [virtual]

Returns the name of the filter

Implements Filter.

2.51 GridFilterBox Class Reference

Public Member Functions

- **GridFilterBox** (GUI::QtGui::QWidget__10 *parent)

- virtual void **setFilter** (Model::Filter::Filter &filter)
- virtual Model::Filter::Filter * **getFilter** ()

2.52 LoadAnalysisVideo Class Reference

Public Member Functions

- void **undo** ()
- void **redo** ()

2.53 LoadFilterconfig Class Reference

Public Member Functions

- LoadFilterconfig ()
- void undo ()
- void redo ()

2.53.1 Constructor & Destructor Documentation

2.53.1.1 LoadFilterconfig ( )

Constuctor

2.53.2 Member Function Documentation

2.53.2.1 void redo ( )

loads a filter configuration from a external file

2.53.2.2 void undo ( )

loads the filter configuration present before external configuration was loaded

2.54 LoadFilterVideo Class Reference

Public Member Functions

- LoadFilterVideo ()
- void undo ()
- void redo ()

2.54.1 Constructor & Destructor Documentation

2.54.1.1 LoadFilterVideo ( )

Constuctor

2.54.2 Member Function Documentation

2.54.2.1 void redo ( )

loads video to which filter can be applied

2.54.2.2 void undo ( )

removes current video to which filters can be applied and loads previous video

2.55 LoadVideo Class Reference

Public Member Functions

- LoadVideo ()
- void undo ()
- void redo ()

2.55.1 Constructor & Destructor Documentation

2.55.1.1 LoadVideo ( )

Constuctor

2.55.2 Member Function Documentation

2.55.2.1 void redo ( )

loads video to which filter can be applied

2.55.2.2 void undo ( )

removes current video to which filters can be applied and loads previous video

2.56 LoadVideoControlPanel Class Reference

Public Member Functions

- void **updateUi** ()

Additional Inherited Members

2.57 MacroblockCalculator Class Reference

Public Member Functions

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

2.57.1 Detailed Description

This class calculates the macroblocks of a video.

2.57.2 Member Function Documentation

2.57.2.1 void calculateMacroblockImages (GUI::Player::Video & *target*)

Calculates QImage video with macroblock overlay.

Parameters

target	the video the frames with the calculated macroblocks are added to

2.57.2.2 void macroBlockCalculator (Model::AVVideo & *video*)

Constructor.

Parameters

video	the video to calculate the macroblocks for

2.58 MainWindow Class Reference

Public Member Functions

- **MainWindow** (GUI::QtGui::QWidget__10 *parent)
- Memento::MainWindowMemento **getMemento** ()
- void **restore** (Memento::MainWindowMemento memento)
- Model::Project & **getProject** ()

Data Fields

- Model::Project * **loadedProject**

2.58.1 Detailed Description

Contains the elements of the GUI

2.59 MainWindowMemento Class Reference

Public Member Functions

- MainWindowMemento ()
- int getSelectedTab ()
- void setSelectedTab (int selectedTab)
- Memento::AnalysisTabMemento getAnalyseTabMeMento ()
- void setAnalyseTabMeMento (Memento::AnalysisTabMemento analyseTabMeMento)
- Memento::FilterTabMemento getFilterTabMemento ()
- void setFilterTabMemento (Memento::FilterTabMemento filterTabMemento)

2.59.1 Detailed Description

This class is the memento for the MainWindow.

2.59.2 Constructor & Destructor Documentation

2.59.2.1 MainWindowMemento ( )

Constructor.

2.59.3 Member Function Documentation

2.59.3.1 Memento::AnalysisTabMemento getAnalyseTabMeMento ( )

Returns the AnalysebTaMemento.

Returns

the AnalyseTabMemento

2.59.3.2 Memento::FilterTabMemento getFilterTabMemento ( )

Returns the FilterTabMemento.

Returns

the FilterTabMemento

2.59.3.3 int getSelectedTab ( )

Returns the currently selected tab.

Returns

the currently selected tab

2.59.3.4 void setAnalyseTabMeMento (Memento::AnalysisTabMemento *analyseTabMeMento*)

Sets the AnalyseTabMemento.

Parameters

analyseTabMe↩ Mento	the AnalyseTabMemento

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

Sets the FilterTabMemento.

Parameters

filterTab↩ Memento	the FilterTabMemento

2.59.3.6 void setSelectedTab (int *selectedTab*)

Sets the currently selected tab.

Parameters

selectedTab	the currently selected tab

2.60 MirrorFilter Class Reference

Public Member Functions

- std::string getFilterDescription ()
- std::string getName ()
- Model::Filter::MirrorMode **getMode** ()
- void **setMode** (Model::Filter::MirrorMode mode)

2.60.1 Detailed Description

Mirrors the video horizontally or vertically

2.60.2 Member Function Documentation

2.60.2.1 std::string getFilterDescription ( ) [virtual]

Returns the description of the filter

Implements Filter.

2.60.2.2 std::string getName ( ) [virtual]

Returns the name of the filter

Implements Filter.

2.61 MirrorFilterBox Class Reference

Public Member Functions

- **MirrorFilterBox** (GUI::QtGui::QWidget__10 *parent)
- virtual void **setFilter** (Model::Filter::Filter &filter)
- virtual Model::Filter::Filter * **getFilter** ()

2.62 MoveFilterDown Class Reference

Public Member Functions

- MoveFilterDown ()
- void undo ()
- void redo ()

2.62.1 Constructor & Destructor Documentation

2.62.1.1 MoveFilterDown ( )

Constuctor

2.62.2 Member Function Documentation

2.62.2.1 void redo ( )

moves selected filter one position down in the filterlist

2.62.2.2 void undo ( )

moves the filter one position back up in the filterlist

2.63 MoveFilterUp Class Reference

Public Member Functions

- MoveFilterUp ()
- void undo ()
- void redo ()

2.63.1 Constructor & Destructor Documentation

2.63.1.1 MoveFilterUp ( )

Constuctor

2.63.2 Member Function Documentation

2.63.2.1 void redo ( )

moves selected filter one position up in the filterlist

2.63.2.2 void undo ( )

moves the filter one position back down in the filterlist

2.64 NegativeFilter Class Reference

Public Member Functions

- std::string getFilterDescription ()
- std::string getName ()

2.64.1 Detailed Description

Converts the video into it's negative

2.64.2 Member Function Documentation

2.64.2.1 std::string getFilterDescription ( ) [virtual]

Returns the description of the filter

Implements Filter.

2.64.2.2 std::string getName ( ) [virtual]

Returns the name of the filter

Implements Filter.

2.65 NegativeFilterBox Class Reference

Public Member Functions

- **NegativeFilterBox** (GUI::QtGui::QWidget__10 *parent)
- virtual void **setFilter** (Model::Filter::Filter &filter)
- virtual Model::Filter::Filter * **getFilter** ()

2.66 NoiseFilter Class Reference

Public Member Functions

- std::string getFilterDescription ()
- Model::Filter::NoiseMode **getMode** ()
- void **setMode** (Model::Filter::NoiseMode mode)
- int **getIntensity** ()
- std::string getName ()
- void **setIntensity** (int intensity)

2.66.1 Detailed Description

Inserts noise into the video

2.66.2 Member Function Documentation

2.66.2.1 std::string getFilterDescription ( ) [virtual]

Returns the description of the filter

Implements Filter.

2.66.2.2 std::string getName ( ) [virtual]

Returns the name of the filter

Implements Filter.

2.67 NoiseFilterBox Class Reference

Public Member Functions

- **NoiseFilterBox** (GUI::QtGui::QWidget__10 *parent)
- virtual void **setFilter** (Model::Filter::Filter &filter)
- virtual Model::Filter::Filter * **getFilter** ()

2.68 PlayerControlPanel Class Reference

Public Member Functions

- PlayerControlPanel (QWidget *parent=0)
- void **updateUi** ()

2.68.1 Detailed Description

This class is the control panel to play videos.

2.68.2 Constructor & Destructor Documentation

2.68.2.1 PlayerControlPanel (QWidget * *parent* = 0)

Constructor.

2.69 PlayerControlPanel__6 Class Reference

2.70 PlayerControlPanel__7 Class Reference

Public Member Functions

- void playerControlPanel (QWidget *parent=0)
- void **updateUi** ()

Additional Inherited Members

2.70.1 Detailed Description

This class is the control panel to play videos.

2.70.2 Member Function Documentation

2.70.2.1 void playerControlPanel (QWidget * *parent* = 0)

Constructor.

2.71 PosterFilter Class Reference

Public Member Functions

- std::string getFilterDescription ()
- int **getNumberOfColors** ()
- std::string getName ()
- void **setNumberOfColors** (int numberOfColors)

2.71.1 Detailed Description

Reduces the maximum number of colors in the video

2.71.2 Member Function Documentation

2.71.2.1 std::string getFilterDescription ( ) [virtual]

Returns the description of the filter

Implements Filter.

2.71.2.2 std::string getName ( ) [virtual]

Returns the name of the filter

Implements Filter.

2.72 PosterFilterBox Class Reference

Public Member Functions

- **PosterFilterBox** (GUI::QtGui::QWidget__10 *parent)
- virtual void **setFilter** (Model::Filter::Filter &filter)
- virtual Model::Filter::Filter * **getFilter** ()

2.73 PreviewControlPanel Class Reference

Public Member Functions

- PreviewControlPanel (Player::QWidget ∗parent=0)
- void **updateUi** ()

2.73.1 Detailed Description

This class is the control panel for the frame preview.

2.73.2 Constructor & Destructor Documentation

2.73.2.1 PreviewControlPanel (Player::QWidget ∗ *parent* = 0)

Constructor.

2.74 PreviewControlPanel__8 Class Reference

2.75 PreviewControlPanel__9 Class Reference

Public Member Functions

- void previewControlPanel (GUI::Player::QWidget__11 ∗parent=0)
- void **updateUi** ()

Additional Inherited Members

2.75.1 Detailed Description

This class is the control panel for the frame preview.

2.75.2 Member Function Documentation

2.75.2.1 void previewControlPanel (GUI::Player::QWidget__11 ∗ *parent* = 0)

Constructor.

2.76 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** ()

2.76.1 Detailed Description

This class contains the different mementos.

2.76.2 Constructor & Destructor Documentation

2.76.2.1 Project (QString *name*)

Constructor.

Parameters

name	name of the project

2.76.3 Member Function Documentation

2.76.3.1 Memento::MainWindowMemento & getMemento ( )

Returns the MainWindoeMemento.

Returns

the MainWindowMemento

2.76.3.2 QString getName ( )

Returns the name of the project.

Returns

name of the project

2.77 ProjectReader Class Reference

Public Member Functions

- ProjectReader (QString path)
- Model::Project readProject ()

2.77.1 Detailed Description

This class can read a project from a file.

2.77.2 Constructor & Destructor Documentation

2.77.2.1 ProjectReader (QString *path*)

Constructor.

Parameters

path	the absolute path to the project file

2.77.3 Member Function Documentation

2.77.3.1 Model::Project readProject ( )

Reads a project from a file.

Returns

the project

2.78 ProjectWriter Class Reference

Public Member Functions

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

2.78.1 Detailed Description

This class can write the project files and the results.

2.78.2 Constructor & Destructor Documentation

2.78.2.1 ProjectWriter (Model::Project *p*)

Constructor.

Parameters

p	the project to save

2.78.3 Member Function Documentation

2.78.3.1 void saveProject ( )

Saves the whole project.

2.78.3.2 void saveResults ( )

Saves on the analysis results.

2.79 PsnrCalculator Class Reference

Public Member Functions

- PsnrCalculator (Model::AVVideo &referenceVideo, Model::AVVideo &compareVideo)
- Model::Graph calculate ()

2.79.1 Detailed Description

This class calculates the psnr-graph of a video.

2.79.2 Constructor & Destructor Documentation

2.79.2.1 PsnrCalculator (Model::AVVideo & *referenceVideo,* Model::AVVideo & *compareVideo*)

Constructor.

Parameters

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

2.79.3 Member Function Documentation

2.79.3.1 Model::Graph calculate ( )

Calculates the psnr graph.

Returns

the psnr graph

2.80 QCheckBox Class Reference

2.81 QComboBox Class Reference

2.82 QDialog Class Reference

2.83 QDialog__12 Class Reference

2.84 QDialog__13 Class Reference

2.85 QFarbDialog Class Reference

2.86 QFrame__1 Class Reference

2.87 QFrame__1 Class Reference

2.88 QFrame__2 Class Reference

2.89 QFrame__2 Class Reference

2.90 QFrame__3 Class Reference

2.91 QGraphicsView Class Reference

2.92 QGroupBox Class Reference

2.93 QHBoxLayout Class Reference

2.94 QLabel Class Reference

2.95 QListWidget Class Reference

2.96 QMainWindow Class Reference

2.97 QMainWindow__4 Class Reference

2.98 QMainWindow__5 Class Reference

2.99 QMediaPlayer Class Reference

Public Member Functions

- void **setMedia** (QMediaContent media)
- void **setPlaylist** (QMediaPlaylist ∗playlist)
- void **setPlaybackRate** (qreal playbackRate)
- void **setVideoOutput** (GUI::QtGui::QVideoWidget output)

2.100 QMenuBar Class Reference

2.101 QRadioButton Class Reference

2.102 QSpinBox Class Reference

2.103 QUndoCommand Class Reference

2.104 QUndoStack Class Reference

2.105 QVBoxLayout Class Reference

2.106 QVideoFrame Class Reference

2.107 QVideoWidget Class Reference

Public Member Functions

- void **show** ()
- Qt::AspectRatioMode **getAspectRatioMode** ()
- void **setAspectRatioMode** (Qt::AspectRatioMode aspectRatioMode)

2.108 QWidget Class Reference

2.109 QWidget__10 Class Reference

2.110 QWidget__11 Class Reference

2.111 RectangleFilter Class Reference

Public Member Functions

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

2.111.1 Detailed Description

Inserts a filled rectangle with a given color into the video

2.111.2 Member Function Documentation

2.111.2.1 std::string getFilterDescription ( ) [virtual]

Returns the description of the filter

Implements Filter.

2.111.2.2 std::string getName ( ) [virtual]

Returns the name of the filter

Implements Filter.

2.112 RectangleFilterBox Class Reference

Public Member Functions

- **RectangleFilterBox** (GUI::QtGui::QWidget__10 *parent)
- virtual void **setFilter** (Model::Filter::Filter &filter)
- virtual Model::Filter::Filter * **getFilter** ()

2.113 RemoveFilter Class Reference

Public Member Functions

- RemoveFilter ()
- void undo ()
- void redo ()

2.113.1 Constructor & Destructor Documentation

2.113.1.1 RemoveFilter ( )

Constuctor

2.113.2 Member Function Documentation

2.113.2.1 void redo ( )

removes a filter from the filterlist

2.113.2.2 void undo ( )

adds the removed filter back into the filterlist

2.114 RemoveVideo Class Reference

Public Member Functions

- RemoveVideo ()
- void undo ()
- void redo ()

2.114.1 Constructor & Destructor Documentation

2.114.1.1 RemoveVideo ( )

Constuctor

2.114.2 Member Function Documentation

2.114.2.1 void redo ( )

removes a video from the analysis tab

2.114.2.2 void undo ( )

readds the removed video to the analysis tab

2.115 RGBDifferenceCalculator Class Reference

Public Member Functions

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

2.115.1 Detailed Description

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

2.115.2 Constructor & Destructor Documentation

2.115.2.1 RGBDifferenceCalculator (GUI::Player::Video & *referenceVideo,* GUI::Player::Video & *video*)

Constructor.

Parameters

referenceVideo	The reference video.
video	The video that is compared to the reference video.

2.115.3 Member Function Documentation

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

Calculates the RGB difference between two videos.

Parameters

target	the video the calculated frames are added to

2.116 RGBFilter Class Reference

Public Member Functions

- std::string getFilterDescription ()
- Model::Filter::BasicColor **getColor** ()
- void **setColor** (Model::Filter::BasicColor color)
- std::string getName ()

2.116.1 Detailed Description

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

2.116.2 Member Function Documentation

2.116.2.1 std::string getFilterDescription ( ) [virtual]

Returns the description of the filter

Implements Filter.

2.116.2.2 std::string getName ( ) [virtual]

Returns the name of the filter

Implements Filter.

2.117 RGBFilterBox Class Reference

Public Member Functions

- **RGBFilterBox** (GUI::QtGui::QWidget__10 *parent)
- virtual void **setFilter** (Model::Filter::Filter &filter)
- virtual Model::Filter::Filter * **getFilter** ()

2.118 RGBHistogrammCalculator Class Reference

Public Member Functions

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

2.118.1 Detailed Description

This class calculates the RGB histogramm for a video.

2.118.2 Member Function Documentation

2.118.2.1 void calculate ( )

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

2.118.2.2 Model::Graph getBlueHistogramm ( )

Returns the blue components of a video.

2.118.2.3 Model::Graph getGreenHistogramm ( )

Returns the green components of a video.

2.118.2.4 Model::Graph getRedHistogramm ( )

Returns the red components of a video.

2.118.2.5 void RGBHistorgrammCalculator (GUI::Player::Video & *video*)

Constructor.

Parameters

video	the video that is analyzed.

2.119 RotationFilter Class Reference

Public Member Functions

- std::string getFilterDescription ()
- int **getAngle** ()
- std::string getName ()
- void **setAngle** (int angle)

2.119.1 Detailed Description

Rotates the video

2.119.2 Member Function Documentation

2.119.2.1 std::string getFilterDescription ( ) [virtual]

Returns the description of the filter

Implements Filter.

2.119.2.2 std::string getName ( ) [virtual]

Returns the name of the filter

Implements Filter.

2.120 RotationFilterBox Class Reference

Public Member Functions

- **RotationFilterBox** (GUI::QtGui::QWidget__10 *parent)
- virtual void **setFilter** (Model::Filter::Filter &filter)
- virtual Model::Filter::Filter * **getFilter** ()

2.121 SaturationFilter Class Reference

Public Member Functions

- std::string getFilterDescription ()
- int **getIntensity** ()
- std::string getName ()
- void **setIntensity** (int intensity)

2.121.1 Detailed Description

Adjusts the saturation of the video

2.121.2 Member Function Documentation

2.121.2.1 std::string getFilterDescription ( ) [virtual]

Returns the description of the filter

Implements Filter.

2.121.2.2 std::string getName ( ) [virtual]

Returns the name of the filter

Implements Filter.

2.122 SaturationFilterBox Class Reference

Public Member Functions

- **SaturationFilterBox** (GUI::QtGui::QWidget__10 *parent)
- virtual void **setFilter** (Model::Filter::Filter &filter)
- virtual Model::Filter::Filter * **getFilter** ()

2.123 ScaleFilter Class Reference

Public Member Functions

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

2.123.1 Detailed Description

Scales the video

2.123.2 Member Function Documentation

2.123.2.1 std::string getFilterDescription ( ) [virtual]

Returns the description of the filter

Implements Filter.

2.123.2.2 std::string getName ( ) [virtual]

Returns the name of the filter

Implements Filter.

2.124 ScaleFilterBox Class Reference

Public Member Functions

- **ScaleFilterBox** (GUI::QtGui::QWidget__10 *parent)
- virtual void **setFilter** (Model::Filter::Filter &filter)
- virtual Model::Filter::Filter * **getFilter** ()

2.125 SepiaFilter Class Reference

Public Member Functions

- std::string getName ()
- std::string getFilterDescription ()

2.125.1 Detailed Description

Converts the video into sepia

2.125.2 Member Function Documentation

2.125.2.1 std::string getFilterDescription ( ) [virtual]

Returns the description of the filter

Implements Filter.

2.125.2.2 std::string getName ( ) [virtual]

Returns the name of the filter

Implements Filter.

2.126 SepiaFilterBox Class Reference

Public Member Functions

- **SepiaFilterBox** (GUI::QtGui::QWidget__10 *parent)
- virtual void **setFilter** (Model::Filter::Filter &filter)
- virtual Model::Filter::Filter * **getFilter** ()

2.127 SharpnessFilter Class Reference

Public Member Functions

- std::string getFilterDescription ()
- int **getIntensity** ()
- std::string getName ()
- void **setIntensity** (int intensity)

2.127.1 Detailed Description

Sharpens the video

2.127.2 Member Function Documentation

2.127.2.1 std::string getFilterDescription ( ) [virtual]

Returns the description of the filter

Implements Filter.

2.127.2.2 std::string getName ( ) [virtual]

Returns the name of the filter

Implements Filter.

2.128 SharpnessFilterBox Class Reference

Public Member Functions

- **SharpnessFilterBox** (GUI::QtGui::QWidget__10 *parent)
- virtual void **setFilter** (Model::Filter::Filter &filter)
- virtual Model::Filter::Filter * **getFilter** ()

2.129 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 ()

2.129.1 Detailed Description

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

2.129.2 Constructor & Destructor Documentation

2.129.2.1 Timer (int *fps*)

Constructor.

Parameters

fps	the fps to play at

2.129.3 Member Function Documentation

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

Adds a player.

Parameters

player	the player to add

2.129.3.2 int getFps ( )

Returns the current fps the timer plays at.

Returns

the current fps

2.129.3.3 float getSpeed ( )

Returns the current speed the timer plays at.

Returns

the current speed

2.129.3.4 bool isPlaying ( )

Whether the timer currently switches frames.

Returns

true if the timer currently switches frames.

2.129.3.5 void pause ( )

Stops the timer from switch frames.

2.129.3.6 void setFps (int *fps*)

Sets the fps for the timer.

Parameters

fps	the new fps

2.129.3.7 void setSpeed (float *speed*)

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

Parameters

speed	the new speed

2.129.3.8 void start ( )

Tells the timer to start switching frames.

2.130 UndoStack Class Reference

Static Public Member Functions

- static QUndoStack & **getUndoStack** ()

2.131 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:std:)
- void removeFrame (int index)
- void insertFrames (int index=-1, vector< std::unique_ptr< QImage > > &frames:std:)
- int getNumberOfFrames ()

Data Fields

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

2.131.1 Detailed Description

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

2.131.2 Constructor & Destructor Documentation

2.131.2.1 Video (int *fps,* int *width,* int *height*)

Constructor.

Parameters

fps	the fps the video should be played at
width	the width of the video
height	the height of the video

2.131.3 Member Function Documentation

2.131.3.1 int getFps ( )

Returns the fps of the video.

Returns

fps of the video

2.131.3.2 QImage * getFrame (int *index*)

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

Parameters

index	the index of the frame to return

2.131.3.3 int getHeight ( )

Returns the height of the video.

Returns

the height of the video

2.131.3.4 int getNumberOfFrames ( )

Returns the number of frames in the video.

Returns

the number of frames in the video

2.131.3.5 int getWidth ( )

Returns the width of the video.

Returns

the width of the video

2.131.3.6 void insertFrame (int *index* = −1, unique_ptr< QImage > 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

index	The index to insert the frame at
frame:std,:	The frame to insert

2.131.3.7 void insertFrames (int *index* = −1, vector< std::unique_ptr< QImage > > &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

index	the index to insert the frames at
frames:std,:	the frames to insert

2.131.3.8 void removeFrame (int *index*)

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

Parameters

index	the index of the frame to remove

2.132 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)

2.132.1 Detailed Description

Converts AVFrames to QImages and vice versa.

2.132.2 Member Function Documentation

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

Converts the given AVFrame to a QImage.

Parameters

frame	the avframe to convert
width	The width of the frame
height	The height of the frame

Returns

the converted AVFrame

2.132.2.2 static std::unique_ptr<GUI::Player::Video> convertAVVideoToVideo (Model::AVVideo & *video*) [inline], [static]

Converts a AVVideo to a Video

Parameters

video	The video to convert

Returns

the converted AVVideo

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

Converts a qimage to a avframe.

Parameters

imgae	the qimage to convert

Returns

the converted qimage

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

Converts a Video to a AVVideo.

Parameters

video	the video to convert

Returns

the converted video

2.133 VideoLoader Class Reference

Public Member Functions

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

2.133.1 Detailed Description

This class can load a encoded video.

2.133.2 Member Function Documentation

2.133.2.1 std::uinque_ptr< Model::AVVideo > loadVideo ( )

Loads the video and generates the AVVideo.

Returns

the loaded video

2.134 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_:std:)
- 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**

2.134.1 Detailed Description

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

2.134.2 Constructor & Destructor Documentation

2.134.2.1 VideoPlayer ( )

Constructor.

2.134.3 Member Function Documentation

2.134.3.1 void addView (GUI::Player::FrameView & *view*)

Adds a view. Multiple views can be added.

Parameters

view	the view to add

2.134.3.2 void clearTimer ( )

Clears the timer.

2.134.3.3 int getFps ( )

Returns the fps the player is currently playing at.

Returns

the current fps of the player

2.134.3.4 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

2.134.3.5 void removeView (GUI::Player::FrameView & *view*)

Removes a view.

Parameters

view	the view to remove

2.134.3.6 void setTimer (shared_ptr< GUI::Player::Timer > timer_:std:)

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

Parameters

timer :std,:	the timer for the player

2.134.3.7 void setVideo (GUI::Player::Video & *video*)

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

Parameters

video	the video to play

2.135 VintageFilter Class Reference

Public Member Functions

- std::string getName ()
- std::string getFilterDescription ()

2.135.1 Detailed Description

Adjusts the colors of the video to make it look vintage

2.135.2 Member Function Documentation

2.135.2.1 std::string getFilterDescription ( ) [virtual]

Returns the description of the filter

Implements Filter.

2.135.2.2 std::string getName ( ) [virtual]

Returns the name of the filter

Implements Filter.

2.136 VintageFilterbox Class Reference

Public Member Functions

- void **vintageFilterBox** (GUI::QtGui::QWidget__10 *parent)
- virtual void **setFilter** (Model::Filter::Filter &filter)
- virtual Model::Filter::Filter * **getFilter** ()

2.137 WriteComment Class Reference

Public Member Functions

- WriteComment ()
- void mergeWith (Undo_Redo::QUndoCommand command)
- void undo ()
- void redo ()
- void id ()

2.137.1 Constructor & Destructor Documentation

2.137.1.1 WriteComment ( )

Constuctor

2.137.2 Member Function Documentation

2.137.2.1 void id ( )

returns id of this command

2.137.2.2 void mergeWith (Undo_Redo::QUndoCommand *command*)

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

2.137.2.3 void redo ( )

applies changes to the textbox

2.137.2.4 void undo ( )

reverts changes to the textbox

2.138 Yuv411FileReader Class Reference

Public Member Functions

- Yuv411FileReader (QString filename, int width, int height, GUI::Player::Compression compression)
- std::unique_ptr< GUI::Player::Video > read ()

Static Public Member Functions

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

Additional Inherited Members

2.138.1 Detailed Description

This class is able to read Yuv 411 files.

2.138.2 Constructor & Destructor Documentation

2.138.2.1 Yuv411FileReader (QString *filename,* int *width,* int *height,* GUI::Player::Compression *compression*)

Constructor.

Parameters

filename	absolute path to the file to load
width	width of the video
height	height of the video
compression	the compression of the file

2.138.3 Member Function Documentation

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

Reads the file in.

Returns

the complete video

Implements YuvFileReader.

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

Converts a Yuv411Vector to the corresponding Rgb88 pixels.

Parameters

vector	the vector to convert

Returns

the computed rgb888 pixels

2.139 Yuv411FileSaver Class Reference

Public Member Functions

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

Static Public Member Functions

- static GUI::Player::Yuv411Vector rgb888ToYuv411 (QRgb pixel1, QRgb pixel2, QRgb pixel3, QRgb pixe4)

Additional Inherited Members

2.139.1 Detailed Description

This class can save videos in the yuv 411 format.

2.139.2 Constructor & Destructor Documentation

2.139.2.1 Yuv411FileSaver (QString *filename,* GUI::Player::Video & *video,* GUI::Player::Compression *compression*)

Constructor.

Parameters

filename	absolute path to the file to save to
video	the video to save
compression	the compression mode

2.139.3 Member Function Documentation

2.139.3.1 static GUI::Player::Yuv411Vector rgb888ToYuv411 (QRgb *pixel1*, QRgb *pixel2*, QRgb *pixel3*, QRgb *pixe4*) [inline], [static]

Converts Rgb888 pixels to a Yuv411Vector.

Returns

the Yuv411Vector

2.139.3.2 void save ( ) [virtual]

Saves the video to the file.

Implements YuvFileSaver.

2.140 Yuv411Vector Class Reference

Data Fields

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

2.140.1 Detailed Description

A Yuv411Vector.

2.141 Yuv420FIleReader Class Reference

Public Member Functions

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

Additional Inherited Members

2.141.1 Detailed Description

This class can read Yuv 420 files.

2.141.2 Member Function Documentation

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

Reads the file in.

Returns

the complete video

Implements YuvFileReader.

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

Constructor.

Parameters

filename	absolute path to the file to load
width	width of the video
height	height of the video

2.142 Yuv420FileSaver Class Reference

Public Member Functions

- Yuv420FileSaver (QString filename, GUI::Player::Video &video)
- void save ()

Additional Inherited Members

2.142.1 Detailed Description

This class can save videos in the yuv 420 format.

2.142.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

filename	absolute path to the file to save to
video	the video to save

2.142.3 Member Function Documentation

2.142.3.1 void save ( ) [virtual]

Saves the video to the file.

Implements YuvFileSaver.

2.143 Yuv422FileReader Class Reference

Public Member Functions

- Yuv422FileReader (QString filename, int width, int height, GUI::Player::Compression compression)
- std::unique_ptr< Videoi > read ()

Static Public Member Functions

- static std::vector< QRgb > yuv422ToRgb888 (GUI::Player::Yuv422Vector vector)

Additional Inherited Members

2.143.1 Detailed Description

This class can read yuv 422 files.

2.143.2 Constructor & Destructor Documentation

2.143.2.1 Yuv422FileReader (QString *filename,* int *width,* int *height,* GUI::Player::Compression *compression*)

Constructor.

Parameters

filename	absolute path to the file to load
width	width of the video
height	height of the video
compression	compression of the file

2.143.3 Member Function Documentation

2.143.3.1 std::unique_ptr< Videoi > read ( ) [virtual]

Reads the file in.

Returns

the complete video

Implements YuvFileReader.

2.143.3.2 static std::vector<QRgb> yuv422ToRgb888 (GUI::Player::Yuv422Vector *vector*) [inline], [static]

Converts a Yuv422Vector the Rgb888 pixels

Returns

the computed rgb888 pixels

2.144 Yuv422FileSaver Class Reference

Public Member Functions

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

Static Public Member Functions

- static GUI::Player::Yuv422Vector rgb888ToYuv422 (QRgb pixel1, QRgb pixel2)

Additional Inherited Members

2.144.1 Detailed Description

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

2.144.2 Constructor & Destructor Documentation

2.144.2.1 Yuv422FileSaver (QString *filename,* GUI::Player::Video & *video,* GUI::Player::Compression *compression*)

Constructor.

Parameters

filename	absolute path to the file to save to
video	the video to save
compression	the compression mode

2.144.3 Member Function Documentation

2.144.3.1 static GUI::Player::Yuv422Vector rgb888ToYuv422 (QRgb *pixel1,* QRgb *pixel2*) [inline], [static]

Converts Rgb888 pixel to a Yuv422Vector.

Returns

the Yuv422Vector

2.144.3.2 void save ( ) [virtual]

Saves the video to the file.

Implements YuvFileSaver.

2.145 Yuv422Vector Class Reference

2.145.1 Detailed Description

A Yuv422Vector.

2.146 Yuv444FileReader Class Reference

Public Member Functions

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

Static Public Member Functions

- static QRgb yuv444ToRgb888 (GUI::Player::Yuv444Vector vector)

Additional Inherited Members

2.146.1 Detailed Description

This class can read Yuv 444 files.

2.146.2 Constructor & Destructor Documentation

2.146.2.1 Yuv444FileReader (QString *filename,* int *width,* int *height,* GUI::Player::Compression *compression*)

Constructor.

Parameters

filename	absolute path to the file to load
width	width of the video
height	height of the video
compression	compression of the file

2.146.3 Member Function Documentation

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

Reads the file in.

Returns

the complete video

Implements YuvFileReader.

2.146.3.2 static QRgb yuv444ToRgb888 (GUI::Player::Yuv444Vector *vector*) [inline], [static]

Converts a Yuv444Vector to a Rgb888 pixel.

Parameters

vector	the vector to convert

Returns

the computed pixel

2.147 Yuv444FileSaver Class Reference

Public Member Functions

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

Static Public Member Functions

- static GUI::Player::Yuv444Vector rgb888ToYuv444 (QRgb pixel1)

Additional Inherited Members

2.147.1 Detailed Description

This class can save videos in the yuv 422 format.

2.147.2 Constructor & Destructor Documentation

2.147.2.1 Yuv444FileSaver (QString *filename,* GUI::Player::Video & *video,* GUI::Player::Compression *compression*)

Constructor.

Parameters

filename	absolute path to the file to save to
video	the video to save
compression	the compression mode

2.147.3 Member Function Documentation

2.147.3.1 static GUI::Player::Yuv444Vector rgb888ToYuv444 (QRgb *pixel1*) [inline],[static]

Converts a Rgb888 pixel to a Yuv444Vector.

Parameters

pixel1	the pixel to convert

Returns

the Yuv444Vector

2.147.3.2 void save ( ) [virtual]

Saves the video to the file.

Implements YuvFileSaver.

2.148 Yuv444Vector Class Reference

2.148.1 Detailed Description

A Yuv444Vector.

2.149 YuvFileOpenDialog Class Reference

Public Member Functions

- YuvFileOpenDialog (GUI::Player::QWidget__11 *parent=0)
- QString getFilename ()
- void show ()
- bool wasSuccessfull ()

2.149.1 Detailed Description

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

2.149.2 Constructor & Destructor Documentation

2.149.2.1 YuvFileOpenDialog (GUI::Player::QWidget__11 * *parent* = 0)

Constructor.

2.149.3 Member Function Documentation

2.149.3.1 QString getFilename ( )

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

Returns

absolute path to the user chosen file

2.149.3.2 void show ( )

Shows the dialog.

2.149.3.3 bool wasSuccessfull ( )

Whether the user clicked ok or cancel.

Returns

true if the user clicked ok. false otherwise.

2.150 YuvFileReader Class Reference

Public Member Functions

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

Static Public Member Functions

- static int clamp (int value)

Protected Attributes

- unique_ptr< QByteArray > **binaryData**:std:
- int **width**
- int **height**
- unique_ptr< GUI::Player::Video > **video**:std:

2.150.1 Detailed Description

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

2.150.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

filename	the absolute path to the file to load
width	the width of the video
height	the height of the video

2.150.3 Member Function Documentation

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

Clamps the given value to the range [0,255].

Parameters

value	the value to clamp

Returns

the clamped value

2.150.3.2 virtual std::unique_ptr<GUI::Player::Video> read ( ) [pure virtual]

Reads the file in.

Returns

the complete video

Implemented in Yuv411FileReader, Yuv444FileReader, Yuv422FileReader, and Yuv420FIleReader.

2.151 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**

2.151.1 Detailed Description

This is the base class for yuv savers.

2.151.2 Constructor & Destructor Documentation

2.151.2.1 YuvFileSaver (QString *filename,* GUI::Player::Video & *video*)

Constructor.

Parameters

filename	absolute path to the file to save to
video	the video to save

2.151.3 Member Function Documentation

2.151.3.1 virtual void save ( ) [pure virtual]

Saves the video to the file.

Implemented in Yuv411FileSaver, Yuv422FileSaver, Yuv444FileSaver, and Yuv420FileSaver.

2.152 YuvInfoDialog Class Reference

Public Member Functions

- YuvInfoDialog (QWidget *parent)
- int getFps ()
- int getWidth ()
- int getHeight ()
- GUI::Player::Compression getCompression ()
- GUI::Player::PixelSheme getPixelSheme ()
- bool wasSuccessful ()
- void show ()

2.152.1 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.

2.152.2 Constructor & Destructor Documentation

2.152.2.1 YuvInfoDialog (QWidget * *parent*)

Constructor.

2.152.3 Member Function Documentation

2.152.3.1 GUI::Player::Compression getCompression ( )

Returns the compression the user entered.

Returns

the compression

2.152.3.2 int getFps ( )

Returns the fps the user entered.

Returns

the fps

2.152.3.3 int getHeight ( )

Returns the height the user entered.

Returns

the height

2.152.3.4 GUI::Player::PixelSheme getPixelSheme ( )

Returns the pixelsheme the user entered.

Returns

the pixelsheme

2.152.3.5 int getWidth ( )

Returns the width the user entered.

Returns

the width

2.152.3.6 void show ( )

Shows the dialog.

2.152.3.7 bool wasSuccessful ( )

Whether the user clicked ok or cancel.

Returns

true if the user clicked ok. false otherwise

2.153 YuvVideo Class Reference

Public Member Functions

- **YuvVideo** (QString path, GUI::Player::PixelSheme type, int width, int height, int fpd)
- QString **getPath** ()
- GUI::Player::Compression **getCompression** ()
- YuvType **getYuvType** ()
- Model::AVVideo & **getAvVideo** ()
- GUI::Player::Video & **getVideo** ()
- void **loadVideo** ()

Data Fields

- GUI::Player::Compression * **compression**
- GUI::Player::Video * **displayVideo**
- Model::AVVideo * **avVideo**

2.154 ZoomFilter Class Reference

Public Member Functions

- std::string getName ()
- std::string getFilterDescription ()
- int **getIntensity** ()
- void **setIntensity** (int intensity)

2.154.1 Detailed Description

Zooms into the video

2.154.2 Member Function Documentation

2.154.2.1 std::string getFilterDescription ( ) [virtual]

Returns the description of the filter

Implements Filter.

2.154.2.2 std::string getName ( ) [virtual]

Returns the name of the filter

Implements Filter.

2.155 ZoomFilterBox Class Reference

Public Member Functions

- **ZoomFilterBox** (GUI::QtGui::QWidget__10 *parent)
- virtual void **setFilter** (Model::Filter::Filter &filter)
- virtual Model::Filter::Filter * **getFilter** ()

Index