Gazelle View

Generated by Doxygen 1.8.11

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

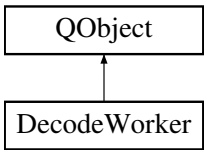
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 DecodeWorker Class Reference

Inheritance diagram for DecodeWorker:



Public Slots

- void decode (int frames)

  DecodeWorker::decode slot that decodes frames and converts them into a QGraphicsPixmapItem.

Signals

- void **resultReady** (QGraphicsPixmapItem ∗image, int frame)

Public Member Functions

- double **posMSec** () const
- double **posFrames** () const
- void setPosFrames (double frame)

  DecodeWorker::setPosFrames sets the Current Frame of the decoder to frame.
- double **posRelative** () const
- double **framerate** () const
- double **codec** () const
- double **frameCount** () const
- bool openStream (QString stream)

  DecodeWorker::openStream opens stream.
- void updateAllProperties ()

  DecodeWorker::updateAllProperties updated every property that can be accessed.

3.1.1 Member Function Documentation

3.1.1.1 void DecodeWorker::decode (int *frames*) [slot]

DecodeWorker::decode slot that decodes frames and converts them into a QGraphicsPixmapItem.

Parameters

frames	

3.1.1.2 bool DecodeWorker::openStream (QString *stream*)

DecodeWorker::openStream opens stream.

Parameters

stream	filename of the stream

Returns

true when successfull

3.1.1.3 void DecodeWorker::setPosFrames (double *frame*)

DecodeWorker::setPosFrames sets the Current Frame of the decoder to frame.

Parameters

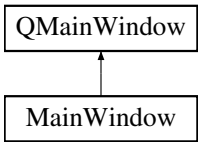
frame	

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

- decode_worker.h
- decode_worker.cpp

3.2 MainWindow Class Reference

Inheritance diagram for MainWindow:



Signals

- void freedImages ()

  freedImages The scene no longer holds ownership of any image

Public Member Functions

- **MainWindow** (QWidget *parent=0)

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

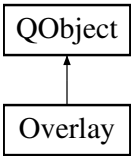
- mainwindow.h
- mainwindow.cpp

3.3 Overlay Class Reference

The Overlay class parses Overlays and Timestamps and gives access to them.

`#include <overlay.h>`

Inheritance diagram for Overlay:



Public Types

- enum ParserError { NoError, OpenFileError, ParseError }

  Overlay::ParseError Enum for storing euccess of parsing.

Public Member Functions

- QPair< QVector2D, qint64 > nextOverlay (quint32 timeStamp)

  Overlay::nextOverlay gets the next overlay after timeStamp.
- QPair< QVector2D, qint64 > nextOverlay ()

  Overlay::nextOverlay gets the next overlay after a timeStamp that is internaly stored.
- QPair< QVector2D, qint64 > previousOverlay (quint32 timeStamp)

  Overlay::previousOverlay gets the previous overlay before timeStamp.
- QPair< QVector2D, qint64 > previousOverlay ()

  Overlay::previousOverlay gets the previous overlay before a timeStamp that is internaly stored.
- QVector< QVector2D > overlaysToFrame (int frame)

  Overlay::overlaysToFrame.
- Overlay::ParserError parseOverlays (QString fileName)

  Overlay::parseOverlays parses the content of fileName as overlays. Each line that has more or equal to 3 tab separated values and doesn start with '#' is parsed.
- Overlay::ParserError parseFrames (QString fileName)

  Overlay::parseFrames parses the content of fileName as timestamp. Each line that has more or equal to 1 tab separated values and doesn start with '#' is parsed.
- qint64 timeStamp (int frame)

  Overlay::timeStamp returns the timestamp to frame in constant time.
- int frame (quint32 timestamp)

  Overlay::frame returns the next lower framenumber that comes before timestamp This is done in logarithmic time.

3.3.1 Detailed Description

The Overlay class parses Overlays and Timestamps and gives access to them.

3.3.2 Member Enumeration Documentation

3.3.2.1 enum Overlay::ParserError

Overlay::ParseError Enum for storing euccess of parsing.

Enumerator

NoError Parsing was successfull.

OpenFileError Could not open File to parse.

ParseError An Error occured while parsing.

3.3.3 Member Function Documentation

3.3.3.1 int Overlay::frame (quint32 *timestamp*)

Overlay::frame returns the next lower framenumber that comes before timestamp This is done in logarithmic time.

Parameters

timestamp	used as a threshold

Returns

framenumber, always between 0 and the amount of parsed Frames

3.3.3.2 QPair< QVector2D, qint64 > Overlay::nextOverlay (quint32 *timeStamp*)

Overlay::nextOverlay gets the next overlay after timeStamp.

Parameters

timeStamp	

Returns

position and timestamp of the next overlay after timeStamp If there is no next Overlay after timeStamp the function returns -1 as timeStamp

3.3.3.3 QPair< QVector2D, qint64 > Overlay::nextOverlay ( )

Overlay::nextOverlay gets the next overlay after a timeStamp that is internaly stored.

Returns

position and timestamp of the next overlay after timeStamp If there is no next Overlay after an internal stored timestamp the function returns -1 as timeStamp

3.3.3.4 QVector< QVector2D > Overlay::overlaysToFrame (int *frame*)

Overlay::overlaysToFrame.

Parameters

frame	framenumber

Returns

All overlays that have a timestamp between the timestamp of frame and the next frame as a QVector

3.3.3.5 Overlay::ParserError Overlay::parseFrames (QString *fileName*)

Overlay::parseFrames parses the content of fileName as timestamp. Each line that has more or equal to 1 tab separated values and doesn start with '#' is parsed.

Parameters

fileName	Path to the file that is to be parsed

Returns

ParserError OpenFileError when it isn't possible to open the file ParseError when the parser failed to parse any line or a line contained invalid characters Noerror when the parser could parse the file successfuly

3.3.3.6 Overlay::ParserError Overlay::parseOverlays (QString *fileName*)

Overlay::parseOverlays parses the content of fileName as overlays. Each line that has more or equal to 3 tab separated values and doesn start with '#' is parsed.

Parameters

fileName	Path to the file that is to be parsed

Returns

ParserError OpenFileError when it isn't possible to open the file ParseError when the parser failed to parse any line or a line contained invalid characters Noerror when the parser could parse the file successfuly

3.3.3.7 QPair< QVector2D, qint64 > Overlay::previousOverlay (quint32 *timeStamp*)

Overlay::previousOverlay gets the previous overlay before timeStamp.

Parameters

timeStamp	

Returns

position and timestamp of the previous overlay before timeStamp If there is no previous Overlay before time↩ Stamp the function returns -1 as timeStamp

3.3.3.8 QPair< QVector2D, qint64 > Overlay::previousOverlay ( )

Overlay::previousOverlay gets the previous overlay before a timeStamp that is internaly stored.

Returns

position and timestamp of the previous overlay before timeStamp If there is no previous Overlay before an internal stored timestamp the function returns -1 as timeStamp

3.3.3.9 qint64 Overlay::timeStamp (int *frame*)

Overlay::timeStamp returns the timestamp to frame in constant time.

Parameters

frame	

Returns

timestamp of frame with framenumber frame, -1 if frame is not available

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

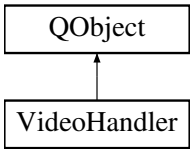
- overlay.h
- overlay.cpp

3.4 VideoHandler Class Reference

The VideoHandler class handles when each frame and overlay gets displayed on screen synchronized, handles the video buffer, requests frames as needed, allows controll of the playback.

`#include <video_handler.h>`

Inheritance diagram for VideoHandler:



Public Types

- enum PlayState { pause, playVideo, playOverlays }

  The PlayState enum holds if state of the video playback.

Public Slots

- void play ()

  VideoHandler::play starts video playback with normal speed If the video is already playing it pauses the playback.
- void playOverlay ()

  VideoHandler::playOverlay starts the playback of the overlays with the speed of the normal video. If the overlays are already playing it will pause the playback.
- void timeout ()

  VideoHandler::timeout slot that gets called if a new frame or overlay needs to be displayed. Reduces the buffer size to 50 Images if the buffer is bigger than 100 Images.
- void open ()

  VideoHandler::open opens a new video file and deletes all previous buffered images.
- void openOverlay ()

  VideoHandler::openOverlay opens a csv file and parses it.
- void openTimestamp ()

  VideoHandler::openTimestamp Opens timestamp file and parses the file.
- void imageFreed ()

  VideoHandler::imageFreed slots that gets calles when _currentImage and _previousImage are removed from the scene. And it deletes them.
- void nextImage ()

  VideoHandler::nextImage slot that sends the next image and updates the overlay.
- void nextOverlay ()

  VideoHandler::nextOverlay slot that sends the next Overlay and updates the Image if needed.
- void previousImage ()

  VideoHandler::previousImage slot that sends the previous Image and updates the overlay.
- void previousOverlay ()

  VideoHandler::previousOverlay slot that sends the previous Overlay and updates the Image if needed.

Signals

- void updateSlider (int totalFrames)

  updateSlider emitted when the slider needs to be updated
- void imageRefresh (QGraphicsPixmapItem *image)

  imageRefresh emitted when the image needs to be updated
- void overlayRefresh (QPoint pos)

  overlayRefresh emitted when the overlay needs to be adjusted
- void decode (int images)

  decode order new frame(s) to be decoded and converted
- void freeImage ()

  freeImage request ownership of images that are displayed

Public Member Functions

- VideoHandler ()
  VideoHandler::VideoHandler.
- double posMSec () const
  VideoHandler::posMSec.
- double posFrames () const
  VideoHandler::posFrames.
- void setPosFrames (double frame, bool updateCurrentFrame=true)
  VideoHandler::setPosFrames.
- double posRelative () const
  VideoHandler::posRelative.
- double framerate () const
  VideoHandler::framerate.
- double codec () const
  VideoHandler::codec.
- double frameCount () const
  VideoHandler::frameCount.
- PlayState playState () const
  VideoHandler::playState returns the current playState of the video playback.

3.4.1 Detailed Description

The VideoHandler class handles when each frame and overlay gets displayed on screen synchronized, handles the video buffer, requests frames as needed, allows controll of the playback.

3.4.2 Member Enumeration Documentation

3.4.2.1 enum VideoHandler::PlayState

The PlayState enum holds if state of the video playback.

Enumerator

pause Video and overlay is paused.

playVideo Video is playing.

playOverlays Overlay is playing.

3.4.3 Member Function Documentation

3.4.3.1 double VideoHandler::codec ( ) const

VideoHandler::codec.

Returns

Codec of the video that is currently decoded

3.4.3.2 void VideoHandler::decode (int *images*) [signal]

decode order new frame(s) to be decoded and converted

Parameters

images	count of frames to be decoded

3.4.3.3 double VideoHandler::frameCount ( ) const

VideoHandler::frameCount.

Returns

The total amount of frames in the current playback

3.4.3.4 double VideoHandler::framerate ( ) const

VideoHandler::framerate.

Returns

Framerate of the video that is currently decoded

3.4.3.5 void VideoHandler::imageRefresh (QGraphicsPixmapItem * *image*) [signal]

imageRefresh emitted when the image needs to be updated

Parameters

image	

3.4.3.6 void VideoHandler::overlayRefresh (QPoint *pos*) [signal]

overlayRefresh emitted when the overlay needs to be adjusted

Parameters

pos	

3.4.3.7 VideoHandler::PlayState VideoHandler::playState ( ) const

VideoHandler::playState returns the current playState of the video playback.

Returns

3.4.3.8 double VideoHandler::posFrames ( ) const

VideoHandler::posFrames.

Returns

Current positon of the playback in frameposition

3.4.3.9 double VideoHandler::posMSec ( ) const

VideoHandler::posMSec.

Returns

Current position of the playback in milliseconds

3.4.3.10 double VideoHandler::posRelative ( ) const

VideoHandler::posRelative.

Returns

Current positon of the playback relative to the entire playback

3.4.3.11 void VideoHandler::setPosFrames (double *frame*, bool *updateCurrentFrame* = `true`)

VideoHandler::setPosFrames.

Parameters

frame	Frame to set the DecodeWorker to.
updateCurrentFrame	Set

3.4.3.12 void VideoHandler::updateSlider (int *totalFrames*) `[signal]`

updateSlider emitted when the slider needs to be updated

Parameters

totalFrames	total count of frames

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

- video_handler.h
- video_handler.cpp

Index