

## My Project

Generated by Doxygen 1.8.11



# Contents

<b>1</b>	<b>Hierarchical Index</b>	<b>1</b>
1.1	Class Hierarchy . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	DecodeWorker Class Reference . . . . .	5
3.1.1	Member Function Documentation . . . . .	5
3.1.1.1	decode . . . . .	5
3.1.1.2	openStream(QString stream) . . . . .	6
3.1.1.3	setPosFrames(double frame) . . . . .	6
3.2	MainWindow Class Reference . . . . .	6
3.3	Overlay Class Reference . . . . .	7
3.3.1	Member Enumeration Documentation . . . . .	8
3.3.1.1	ParserError . . . . .	8
3.3.2	Member Function Documentation . . . . .	8
3.3.2.1	frame(quint32 timeStamp) . . . . .	8
3.3.2.2	nextOverlay(quint32 timeStamp) . . . . .	8
3.3.2.3	nextOverlay() . . . . .	8
3.3.2.4	overlaysToFrame(int frame) . . . . .	9
3.3.2.5	parseFrames(QString fileName) . . . . .	9
3.3.2.6	parseOverlays(QString fileName) . . . . .	9
3.3.2.7	previousOverlay(quint32 timeStamp) . . . . .	9

3.3.2.8	<code>previousOverlay()</code>	10
3.3.2.9	<code>timeStamp(int frame)</code>	10
3.4	VideoHandler Class Reference	10
3.4.1	Detailed Description	12
3.4.2	Member Function Documentation	12
3.4.2.1	<code>codec() const</code>	12
3.4.2.2	<code>decode</code>	12
3.4.2.3	<code>frameCount() const</code>	13
3.4.2.4	<code>framerate() const</code>	13
3.4.2.5	<code>imageRefresh</code>	13
3.4.2.6	<code>overlayRefresh</code>	13
3.4.2.7	<code>playState() const</code>	13
3.4.2.8	<code>posFrames() const</code>	13
3.4.2.9	<code>posMSec() const</code>	14
3.4.2.10	<code>posRelative() const</code>	14
3.4.2.11	<code>setPosFrames(double frame, bool updateCurrentFrame=true)</code>	14
3.4.2.12	<code>updateSlider</code>	14
<b>Index</b>		<b>15</b>

# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

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

QMainWindow	
MainWindow . . . . .	<a href="#">6</a>
QObject	
DecodeWorker . . . . .	<a href="#">5</a>
Overlay . . . . .	<a href="#">7</a>
VideoHandler . . . . .	<a href="#">10</a>



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">DecodeWorker</a>	5
<a href="#">MainWindow</a>	6
<a href="#">Overlay</a>	7
<a href="#">VideoHandler</a>	
Handles when each frame and overlay gets displayed on screen synchronized, handles the video buffer, requests frames as needed, allows control of the playback	10



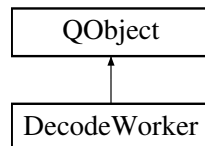


## Chapter 3

# 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

<i>frames</i>	
---------------	--

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

[DecodeWorker::openStream](#) opens stream.

## Parameters

<i>stream</i>	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

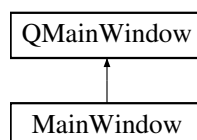
<i>frame</i>	
--------------	--

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

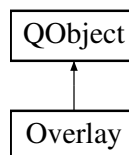
- **MainWindow** (QWidget \*parent=0)

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

- mainwindow.h
- mainwindow.cpp

## 3.3 Overlay Class Reference

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 framenummer that comes before timestamp This is done in logarithmic time.*

### 3.3.1 Member Enumeration Documentation

#### 3.3.1.1 enum Overlay::ParserError

Overlay::ParseError Enum for storing euccess of parsing.

Enumerator

**NoError** Parsing was successfull.

**OpenFileError** Could not open File to parse.

### 3.3.2 Member Function Documentation

#### 3.3.2.1 int Overlay::frame ( quint32 *timestamp* )

[Overlay::frame](#) returns the next lower framenumbrer that comes before timestamp This is done in logarithmic time.

Parameters

<i>timestamp</i>	used as a threshold
------------------	---------------------

Returns

framenumbrer, always between 0 and the amount of parsed Frames

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

[Overlay::nextOverlay](#) gets the next overlay after timeStamp.

Parameters

<i>timeStamp</i>	
------------------	--

Returns

position and timestamp of the next overlay after timeStamp If there is no next [Overlay](#) after timeStamp the function returns -1 as timeStamp

#### 3.3.2.3 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

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

[Overlay::overlaysToFrame](#).

##### Parameters

<i>frame</i>	framenummer
--------------	-------------

##### Returns

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

#### 3.3.2.5 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't start with '#' is parsed.

##### Parameters

<i>fileName</i>	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 successfully

#### 3.3.2.6 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't start with '#' is parsed.

##### Parameters

<i>fileName</i>	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 successfully

#### 3.3.2.7 QPair< QVector2D, quint64 > Overlay::previousOverlay ( quint32 *timeStamp* )

[Overlay::previousOverlay](#) gets the previous overlay before timeStamp.

## Parameters

<i>timeStamp</i>	
------------------	--

## Returns

position and timestamp of the previous overlay before timeStamp If there is no previous [Overlay](#) before timeStamp the function returns -1 as timeStamp

### 3.3.2.8 QPair< QVector2D, qint64 > Overlay::previousOverlay ( )

[Overlay::previousOverlay](#) gets the previous overlay before a timeStamp that is internally 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

### 3.3.2.9 qint64 Overlay::timeStamp ( int frame )

[Overlay::timeStamp](#) returns the timestamp to frame in constant time.

## Parameters

<i>frame</i>	
--------------	--

## Returns

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

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

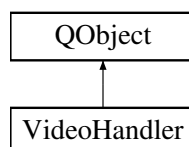
- overlay.h
- overlay.cpp

## 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** }

## 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 called 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.

### 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 Function Documentation

#### 3.4.2.1 double VideoHandler::codec ( ) const

[VideoHandler::codec.](#)

#### Returns

Codec of the video that is currently decoded

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

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

#### Parameters

<i>images</i>	count of frames to be decoded
---------------	-------------------------------



3.4.2.3 `double VideoHandler::frameCount ( ) const`

[VideoHandler::frameCount](#).

#### Returns

The total amount of frames in the current playback

3.4.2.4 `double VideoHandler::framerate ( ) const`

[VideoHandler::framerate](#).

#### Returns

Framerate of the video that is currently decoded

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

imageRefresh emitted when the image needs to be updated

#### Parameters

<i>image</i>	
--------------	--

3.4.2.6 `void VideoHandler::overlayRefresh ( QPoint pos )` [signal]

overlayRefresh emitted when the overlay needs to be adjusted

#### Parameters

<i>pos</i>	
------------	--

3.4.2.7 `VideoHandler::PlayState VideoHandler::playState ( ) const`

[VideoHandler::playState](#) returns the current playState of the video.

#### Returns

3.4.2.8 `double VideoHandler::posFrames ( ) const`

[VideoHandler::posFrames](#).

#### Returns

Current position of the playback in frame position

#### 3.4.2.9 double VideoHandler::posMSec ( ) const

[VideoHandler::posMSec.](#)

##### Returns

Current position of the playback in milliseconds

#### 3.4.2.10 double VideoHandler::posRelative ( ) const

[VideoHandler::posRelative.](#)

##### Returns

Current position of the playback relative to the entire playback

#### 3.4.2.11 void VideoHandler::setPosFrames ( double *frame*, bool *updateCurrentFrame* = true )

[VideoHandler::setPosFrames.](#)

##### Parameters

<i>frame</i>	Frame to set the <a href="#">DecodeWorker</a> to.
<i>updateCurrentFrame</i>	Set

#### 3.4.2.12 void VideoHandler::updateSlider ( int *totalFrames* ) [signal]

updateSlider emitted when the slider needs to be updated

##### Parameters

<i>totalFrames</i>	total count of frames
--------------------	-----------------------

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

- video\_handler.h
- video\_handler.cpp

# Index

- codec
  - VideoHandler, [12](#)
- decode
  - DecodeWorker, [5](#)
  - VideoHandler, [12](#)
- DecodeWorker, [5](#)
  - decode, [5](#)
  - openStream, [6](#)
  - setPosFrames, [6](#)
- frame
  - Overlay, [8](#)
- frameCount
  - VideoHandler, [12](#)
- framerate
  - VideoHandler, [13](#)
- imageRefresh
  - VideoHandler, [13](#)
- MainWindow, [6](#)
- nextOverlay
  - Overlay, [8](#)
- NoError
  - Overlay, [8](#)
- OpenFileError
  - Overlay, [8](#)
- openStream
  - DecodeWorker, [6](#)
- Overlay, [7](#)
  - frame, [8](#)
  - nextOverlay, [8](#)
  - NoError, [8](#)
  - OpenFileError, [8](#)
  - overlaysToFrame, [8](#)
  - parseFrames, [9](#)
  - parseOverlays, [9](#)
  - ParserError, [8](#)
  - previousOverlay, [9](#), [10](#)
  - timeStamp, [10](#)
- overlayRefresh
  - VideoHandler, [13](#)
- overlaysToFrame
  - Overlay, [8](#)
- parseFrames
  - Overlay, [9](#)
- parseOverlays
  - Overlay, [9](#)
- Overlay, [9](#)
- ParserError
  - Overlay, [8](#)
- playState
  - VideoHandler, [13](#)
- posFrames
  - VideoHandler, [13](#)
- posMSec
  - VideoHandler, [13](#)
- posRelative
  - VideoHandler, [14](#)
- previousOverlay
  - Overlay, [9](#), [10](#)
- setPosFrames
  - DecodeWorker, [6](#)
  - VideoHandler, [14](#)
- timeStamp
  - Overlay, [10](#)
- updateSlider
  - VideoHandler, [14](#)
- VideoHandler, [10](#)
  - codec, [12](#)
  - decode, [12](#)
  - frameCount, [12](#)
  - framerate, [13](#)
  - imageRefresh, [13](#)
  - overlayRefresh, [13](#)
  - playState, [13](#)
  - posFrames, [13](#)
  - posMSec, [13](#)
  - posRelative, [14](#)
  - setPosFrames, [14](#)
  - updateSlider, [14](#)