Appendix A: InstaDam Frontend API

Documentation for the frontend was done with QDoc, Qt's documentation engine. To generate the full documentation you must have a working qdoc binary (this is not built by default as it has external LLVM and CLANG dependencies). You also need to build the full Qt documentation so that this documentation can properly parse and link to the Qt supplied functions. (Note that this can take many hours). Once this is done run the following command from the root source directory:

```
qdoc --indexdir <PATH TO qtcore.index> --indexdir <PATH to
qtgui.index> --indexdir <PATH TO qtnetwork.index> --indexdir <PATH TO
qtwidgets.index> -I. -Iapp -ISelector -I<PATH TO QT include>/QtCore
-I<PATH TO QT include>/QtGui -I<PATH TO QT include>/QtNetwork -I<PATH
TO QT include>/QtWidgets instadam.qdocconf
```

AddCommand Class

The AddCommand class provides undo and redo actions for adding a SelectItem to a PhotoScene. More...

Header:	#include <AddCommand>
Inherits:	QUndoCommand

- List of all members, including inherited members

Public Functions

	AddCommand(SelectItem **item*, PhotoScene **scene*, InstaDam **idam*, QUndoCommand **parent* = nullptr)
virtual	**~AddCommand**() override

Reimplemented Public Functions

virtual void	**redo**() override
virtual void	**undo**() override

- 11 public functions inherited from QUndoCommand

Detailed Description

The AddCommand class provides undo and redo actions for adding a SelectItem to a PhotoScene.

AddCommand is used to hold a reference to a SelectItem and the PhotoScene to which it belongs. Using the undo() and redo() commands the item can be added and removed from the scene repeatedly.

See also DeleteCommand, MoveCommand, MoveVertexCommand, AddVertexCommand, DeleteVertexCommand, ErasePointsCommand, RotateCommand, and EditLabelCommand.

Member Function Documentation

AddCommand:: AddCommand**(SelectItem *_item_, PhotoScene *_scene_, InstaDam *_idam_, QUndoCommand *_parent_ = nullptr)**

Constructs an AddCommand using the provided *item* and *scene* so that the addition of this *item* to the *scene* can be undone and/or redone. *idam* is proveded for access to the calling InstaDam instance. *parent* refers to parent QUndoCommand, if any.

[override virtual]AddCommand::~AddCommand**()**

Destructor

[override virtual]void **AddCommand::**redo**()**

Reimplemented from QUndoCommand::redo().

[override virtual]void **AddCommand::**undo**()**

Reimplemented from QUndoCommand::undo().

AddUserToProject Class

The AddUserToProject class provides an interface for adding a user to a project. More...

Header:	#include <AddUserToProject>
Inherits:	QWidget

- List of all members, including inherited members
- Obsolete members

Public Functions

	AddUserToProject(QWidget **parent* = nullptr)
virtual	**~AddUserToProject**()

- 222 public functions inherited from QWidget
- 31 public functions inherited from QObject

- 15 public functions inherited from QPaintDevice

Public Variables

QString	**accessToken**
QString	**databaseURL**
int	**projectId**
QString	**userDetails**
UserPrivilege *	**userPrivilege**

Static Public Members

const QMetaObject	**staticMetaObject**

- 5 static public members inherited from QWidget
- 9 static public members inherited from QObject
- 1 static public member inherited from QPaintDevice

Additional Inherited Members

- 59 properties inherited from QWidget
- 1 property inherited from QObject
- 19 public slots inherited from QWidget
- 1 public slot inherited from QObject
- 4 signals inherited from QWidget
- 2 signals inherited from QObject
- 35 protected functions inherited from QWidget
- 9 protected functions inherited from QObject
- 1 protected function inherited from QPaintDevice
- 1 protected slot inherited from QWidget

Detailed Description

The AddUserToProject class provides an interface for adding a user to a project.

Member Function Documentation

AddUserToProject:: AddUserToProject(**QWidget** ***parent* = nullptr)**

Constructs an instance of AddUserToProject with parent *parent*, if any.

[virtual]AddUserToProject::~AddUserToProject**()**

Destructor

AddVertexCommand Class

The AddVertexCommand class provides undo and redo actions for adding a vertex to a SelectItem on a PhotoScene. More...

Header:	#include <AddVertexCommand>
Inherits:	QUndoCommand

- List of all members, including inherited members

Public Functions

	AddVertexCommand(SelectItem **item*, const QPointF *point*, QUndoCommand **parent* = nullptr)

Reimplemented Public Functions

virtual void	**redo**() override
virtual void	**undo**() override

- 11 public functions inherited from QUndoCommand

Detailed Description

The AddVertexCommand class provides undo and redo actions for adding a vertex to a SelectItem on a PhotoScene.

AddVertexCommand is used to hold a reference to a SelectItem and the position of the new vertex *point*. Using the undo() and redo() commands the vertex can removed from and to the SelectItem repeatedly.

See also AddCommand, MoveCommand, DeleteCommand, MoveVertexCommand, DeleteVertexCommand, ErasePointsCommand, RotateCommand, and EditLabelCommand.

Member Function Documentation

AddVertexCommand::AddVertexCommand(SelectItem **item*, const QPointF *point*, QUndoCommand **parent* = nullptr)

Constructs an AddVeretxCommand using the provided *item* and *point* so that the vertex can be added/removed from *item* repeatedly. *parent* refers to parent QUndoCommand, if any.

[override virtual]void **AddVertexCommand::**redo**()**

Reimplemented from QUndoCommand::redo().

[override virtual]void **AddVertexCommand::**undo**()**

Reimplemented from QUndoCommand::undo().

BoxBasedSelector Class

The BoxBasedSelector class provides a base class for SelectItem subclasses that can be described with a box. More...

Header:	#include <BoxBasedSelector>
Inherits:	SelectItem
Inherited By:	EllipseSelect and RectangleSelect

- List of all members, including inherited members
- Obsolete members

Public Functions

	BoxBasedSelector(const QJsonObject &*json*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)
	BoxBasedSelector(QPointF *point*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)

	BoxBasedSelector(QPointF *point*, qreal *vertSize*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)
virtual	**~BoxBasedSelector**() override
QRectF	**getRect**() const
qreal	**getRotationAngle**() const
virtual void	**setMirrorCorners**(QRectF *tlc*, QRectF *blc*, QRectF *trc*, QRectF *brc*) const = 0
virtual void	**setRectUnchecked**(QRectF *rect*) = 0
void	**setRotationAngle**(qreal *angle*)

Reimplemented Public Functions

virtual void	**clickPoint**(const QPointF &*point*) override
virtual void	**insertVertex**(const int *vertex*, const QPointF &*point*) override
virtual int	**numberOfVertices**() const override
virtual void	**read**(const QJsonObject &*json*) override
virtual void	**removeVertex**(int *vertex* = UNSELECTED) override
virtual void	**resetActiveVertex**() override
virtual void	**resizeItem**(const int *vertex*, QPointF &*oldP*, QPointF &*newP*) override

virtual void	**rotate**(const QPointF &*from*, const QPointF &*to*) override
virtual void	**write**(QJsonObject &*json*) const override

- 57 public functions inherited from SelectItem
- 183 public functions inherited from QGraphicsItem

Protected Functions

void	**calcCorners**(bool *mir* = true)

- 2 protected functions inherited from SelectItem
- 24 protected functions inherited from QGraphicsItem

Protected Variables

QRectF	**bl**
QRectF	**br**
qreal	**myRotation**
QRectF	**tl**
QRectF	**tr**

Additional Inherited Members

- 1 static public member inherited from SelectItem

Detailed Description

The BoxBasedSelector class provides a base class for SelectItem subclasses that can be described with a box.

The BoxBasedSelector class provides a base class for SelectItem subclasses that can be described with a box, such as RectangleSelect and EllipseSelect. It implements several of the virtual functions from SelectItem which will have identical implementations on all its subclasses:

- RectangleSelect
- EllipseSelect

Member Function Documentation

BoxBasedSelector:: BoxBasedSelector**(const QJsonObject &*json*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)**

Constructs a BoxBasedSelector object by reading a QJsonObject and setting the internal rectangle and rotation angle to the values given in *json*. *label* is the Label which owns this object and *item* is the parent QGraphicsItem, if any.

BoxBasedSelector:: BoxBasedSelector**(QPointF *point*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem**item* = nullptr)**

Constructs a BoxBasedSelector object by setting all vertices to be a *point*, *label* is the Label which owns this object and *item* is the parent QGraphicsItem, if any.

BoxBasedSelector:: BoxBasedSelector**(QPointF *point*, qreal *vertSize*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem *_item_ = nullptr)**

Constructs a BoxBasedSelector object by setting all vertices to be a *point*, *vertSize* indicates the size of the vertex highlight boxes, *label* is the Label which owns this object and *item* is the parent QGraphicsItem, if any.

[override virtual]BoxBasedSelector::~BoxBasedSelector**()**

Destructor

[protected]void **BoxBasedSelector::**calcCorners**(**bool ***mir* = true)**

Calculates the vertex boxes in the corners of the defining rectangle. The boxes are used to indicate the BoxBasedSelector is active and to act as points to click and drag to resize the item. If the mirror needs to be updated as well the *mir* should be true.

[override virtual]void **BoxBasedSelector::**clickPoint**(const QPointF &*point*)**

Reimplemented from SelectItem::clickPoint().

QRectF **BoxBasedSelector::**getRect**() const**

Returns a QRectF which denotes the bounding rectanlge of the BoxBasedSelector.

qreal **BoxBasedSelector::**getRotationAngle**() const**

Returns a qreal containing the rotation angle of the BoxBasedSelector in degrees.

See also setRotationAngle().

[override virtual]void **BoxBasedSelector::**insertVertex**(const** int ***vertex*, const QPointF &*point*)**

Reimplemented from SelectItem::insertVertex().

Empty function as this and any derived classes can have two, and only two vertices.

[override virtual]int **BoxBasedSelector::**numberOfVertices**() const**

Reimplemented from SelectItem::numberOfVertices().

Always returns 2, as this and any derived classes can have two, and only two vertices.

[override virtual]void **BoxBasedSelector::**read**(const QJsonObject &*json*)**

Reimplemented from SelectItem::read().

[override virtual]void **BoxBasedSelector::**removeVertex**(**int ***vertex* = UNSELECTED)**

[override virtual]void **BoxBasedSelector::**resetActiveVertex**()**

Reimplemented from SelectItem::resetActiveVertex().

[override virtual]void **BoxBasedSelector::**resizeItem**(const** int ***vertex*, QPointF &*oldP*, QPointF &*newP*)**

Reimplemented from SelectItem::resizeItem().

[override virtual]void **BoxBasedSelector::**rotate**(const QPointF &*from*, const QPointF &*to*)**

Reimplemented from SelectItem::rotate().

[pure virtual]void **BoxBasedSelector::**setMirrorCorners**(QRectF *tlc*, QRectF *blc*, QRectF *trc*, QRectF*brc*) const**

Pure virtual function to set the corner vertex boxes of the mirror object. *tlc* is the top-left corner, *blc* is the bottom-left corner, *trc* is the top-right corner, and *brc* is the bottom right corner.

[pure virtual]void **BoxBasedSelector::**setRectUnchecked**(QRectF *rect*)**

Pure virtual function to set the internal rectangle of the BoxBasedSelector to *rect* without checking to see if it completely fits in the QGraphicsScene. This is used when setting the internal rectangle of the mirror after checking has been done in this instance.

void **BoxBasedSelector::**setRotationAngle**(qreal *angle*)**

Sets the rotation angle of the BoxBasedSelector to *angle* degrees, measured from the horizontal going counterclockwise.

See also getRotationAngle().

[override virtual]void **BoxBasedSelector::**write(**QJsonObject &*json***) **const**

Reimplemented from SelectItem::write().

ColorDialog Class

Creates a QCOlorDialog to select label colors with an initial widget size. More...

Header:	#include <ColorDialog>
Inherits:	QColorDialog

- List of all members, including inherited members
- Obsolete members

Public Functions

ColorDialog(const QColor &*initial*, QWidget **parent* = nullptr)

- 9 public functions inherited from QColorDialog
- 8 public functions inherited from QDialog
- 222 public functions inherited from QWidget
- 31 public functions inherited from QObject
- 15 public functions inherited from QPaintDevice

Additional Inherited Members

- 2 properties inherited from QColorDialog
- 2 properties inherited from QDialog
- 59 properties inherited from QWidget
- 1 property inherited from QObject
- 5 public slots inherited from QDialog
- 19 public slots inherited from QWidget
- 1 public slot inherited from QObject
- 2 signals inherited from QColorDialog
- 3 signals inherited from QDialog
- 4 signals inherited from QWidget
- 2 signals inherited from QObject
- 6 static public members inherited from QColorDialog
- 5 static public members inherited from QWidget
- 9 static public members inherited from QObject
- 1 static public member inherited from QPaintDevice
- 2 protected functions inherited from QColorDialog
- 6 protected functions inherited from QDialog
- 35 protected functions inherited from QWidget
- 9 protected functions inherited from QObject
- 1 protected function inherited from QPaintDevice
- 1 protected slot inherited from QWidget

Detailed Description

Creates a QCOlorDialog to select label colors with an initial widget size.

Creates a QCOlorDialog to select label colors with an initial widget size. This is specifically used for the QtWebAssembly build as that infrastructure requires a specified size for all QDialogs.

Member Function Documentation

ColorDialog:: ColorDialog**(const QColor &*initial*, QWidget **parent* = nullptr)**

Constructs a color selection dialog with a hard coded size, initial QColor selection of *initial*, and parent QWidget *parent*, if any.

DeleteCommand Class

The DeleteCommand class provides undo and redo actions for adding a SelectItem to a PhotoScene. More...

Header:	#include <DeleteCommand>
Inherits:	QUndoCommand

- List of all members, including inherited members

Public Functions

DeleteCommand(SelectItem **item*, PhotoScene **scene*, InstaDam **idam*, QUndoCommand **parent* = nullptr)

Reimplemented Public Functions

virtual void	**redo**() override

virtual void	**undo**() override

- 11 public functions inherited from QUndoCommand

Detailed Description

The DeleteCommand class provides undo and redo actions for adding a SelectItem to a PhotoScene.

DeleteCommand is used to hold a reference to a SelectItem and the PhotoScene to which it belongs. Using the undo() and redo() commands the item can be added and removed from the scene repeatedly.

See also AddCommand, MoveCommand, MoveVertexCommand, AddVertexCommand, DeleteVertexCommand, ErasePointsCommand, RotateCommand, and EditLabelCommand.

Member Function Documentation

DeleteCommand:: DeleteCommand**(SelectItem ****item***, PhotoScene ****scene***, InstaDam ****idam***, QUndoCommand****parent*** **= nullptr)**

Constructs a DeleteCommand using the provided *item* and *scene* so that the deletion of this *item* from the *scene* can be undone and/or redone. *idam* is proveded for access to the calling InstaDam instance. *parent* refers to parent QUndoCommand, if any.

[override virtual]void **DeleteCommand::**redo**()**

Reimplemented from QUndoCommand::redo().

[override virtual]void **DeleteCommand::**undo**()**

Reimplemented from QUndoCommand::undo().

DeleteVertexCommand Class

The DeleteVertexCommand class provides undo and redo actions for deleting a vertex from a SelectItem on a PhotoScene. More...

Header:	#include <DeleteVertexCommand>
Inherits:	QUndoCommand

- List of all members, including inherited members

Public Functions

DeleteVertexCommand(SelectItem **item*, QUndoCommand **parent* = nullptr)

Reimplemented Public Functions

virtual void	**redo**() override
virtual void	**undo**() override

- 11 public functions inherited from QUndoCommand

Detailed Description

The DeleteVertexCommand class provides undo and redo actions for deleting a vertex from a SelectItem on a PhotoScene.

DeleteVertexCommand is used to hold a reference to a SelectItem and it's activeVertex. Using the undo() and redo() commands the vertex can be added to and removed from the SelectItem repeatedly.

See also AddCommand, MoveCommand, DeleteCommand, MoveVertexCommand, AddVertexCommand, ErasePointsCommand, RotateCommand, and EditLabelCommand.

Member Function Documentation

DeleteVertexCommand:: DeleteVertexCommand**(SelectItem ****item***, QUndoCommand ****parent*** **= nullptr)**

Constructs a DeleteVeretxCommand using the provided *item* the vertex can be added/removed from *item* repeatedly. *parent* refers to parent QUndoCommand, if any.

[override virtual]void **DeleteVertexCommand::**redo**()**

Reimplemented from QUndoCommand::redo().

[override virtual]void **DeleteVertexCommand::**undo**()**

Reimplemented from QUndoCommand::undo().

EditLabelCommand Class

The EditLabelCommand command is used to change the Label of the selected item. More...

Header:	#include <EditLabelCommand>
Inherits:	QUndoCommand

- List of all members, including inherited members

Public Functions

EditLabelCommand(SelectItem **item*, QSharedPointer<Label> *newLabel*, QSharedPointer<Label> *oldLabel*, PhotoScene **scene*, InstaDam **idam*, QUndoCommand **parent* = nullptr)

Reimplemented Public Functions

virtual void	**redo**() override
virtual void	**undo**() override

- 11 public functions inherited from QUndoCommand

Public Variables

SelectItem *	**myItem**
QSharedPointer<Label>	**myNewLabel**
QSharedPointer<Label>	**myOldLabel**

InstaDam *	**myParent**
PhotoScene *	**myScene**

Detailed Description

The EditLabelCommand command is used to change the Label of the selected item.

EditLabelCommand is used to change the Label of the selected item. Using the undo() and redo() commands the item can be unedited and edited repeatedly.

See also AddCommand, MoveCommand, AddVertexCommand, MoveVertexCommand, DeleteVertexCommand, ErasePointsCommand, and DeleteCommand.

Member Function Documentation

EditLabelCommand:: EditLabelCommand(SelectItem *item, QSharedPointer<Label> newLabel, QSharedPointer<Label> oldLabel, PhotoScene *scene, InstaDam *idam, QUndoCommand *parent = nullptr)

Constructs an EditLabel command using the provided *item*, *newLabel*, *oldLabel*, *scene*, *idam*, and *parent*, if any.

[override virtual]void **EditLabelCommand::**redo**()**

Reimplemented from QUndoCommand::redo().

[override virtual]void **EditLabelCommand::**undo**()**

Reimplemented from QUndoCommand::undo().

EllipseSelect Class

The EllipseSelect class provides a class for annotating elliptical regions. More...

Header:	#include <EllipseSelect>
Inherits:	QGraphicsEllipseItem and BoxBasedSelector

- List of all members, including inherited members
- Obsolete members

Public Functions

	EllipseSelect()
	EllipseSelect(const QJsonObject &*json*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)
	EllipseSelect(QPointF *point*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)
	EllipseSelect(QPointF *point*, qreal *vertexSize*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)
virtual	**~EllipseSelect**() override
bool	**isVisible**() const
QGraphicsScene *	**scene**()
void	**setOpacity**(qreal *val*)

Reimplemented Public Functions

virtual void	**addPoint**(QPointF &*point*, const int *vertex* = UNSELECTED) override
virtual QString	**baseInstructions**() const override
virtual QRectF	**boundingRect**() const override
virtual EllipseSelect *	**getMirror**() const override
virtual bool	**isInside**(const QPointF &*point*) const override
virtual void	**mirrorHide**() const override
virtual void	**mirrorShow**() const override
virtual void	**moveItem**(const QPointF &*oldPos*, QPointF &*newPos*) override
virtual void	**paint**(QPainter **painter*, const QStyleOptionGraphicsItem **option*, QWidget **widget* = nullptr) override
virtual void	**rotateMirror**() const override
virtual void	**setMirror**(SelectItem **item*) override
virtual void	**setMirrorActive**() const override
virtual void	**setMirrorAdded**() const override
virtual void	**setMirrorCorners**(QRectF *tlc*, QRectF *blc*, QRectF *trc*, QRectF *brc*) const override
virtual void	**setMirrorMoved**() const override

virtual void	**setMirrorResized**() const override
virtual void	**setMirrorVertex**(int *vertex*) const override
virtual void	**setRectUnchecked**(QRectF *rect*) override
virtual void	**toPixmap**(QPainter **painter*) override
virtual int	**type**() const override
virtual void	**updateMirrorScene**() const override
virtual void	**updatePen**(QPen *pen*) override

- 14 public functions inherited from QGraphicsEllipseItem
- 14 public functions inherited from BoxBasedSelector
- 6 public functions inherited from QAbstractGraphicsShapeItem
- 57 public functions inherited from SelectItem
- 366 public functions inherited from QGraphicsItem

Public Variables

const QString	**baseInstruction**

Additional Inherited Members

- 1 static public member inherited from SelectItem
- 1 protected function inherited from BoxBasedSelector
- 2 protected functions inherited from SelectItem
- 48 protected functions inherited from QGraphicsItem

Detailed Description

The EllipseSelect class provides a class for annotating elliptical regions.

Provides a class for annotating elliptical regions in InstaDam. The region is described by its top-left and lower-right vertices, and by its angle of rotation.

Member Function Documentation

EllipseSelect:: EllipseSelect**()**

Constructs an EllipseSelect object with all vertices at 0,0 in scene coordinates

EllipseSelect:: EllipseSelect**(const QJsonObject &*json*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem**item* = nullptr)**

Constructs an EllipseSelect object by reading a QJsonObject and setting the internal rectangle and rotation angle to the values given in *json*. *label* is the Label which owns this object and *item* is the parent QGraphicsItem, if any.

EllipseSelect:: EllipseSelect**(QPointF *point*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)**

Constructs an EllipseSelect object by setting all vertices to be a *point*, *label* is the Label which owns this object and *item* is the parent QGraphicsItem, if any.

EllipseSelect:: EllipseSelect**(QPointF *point*, qreal *vertexSize*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)**

Constructs an EllipseSelect object by setting all vertices to be a *point*, *vertexSize* indicates the size of the vertex highlight boxes, *label* is the Label which owns this object and *item* is the parent QGraphicsItem, if any.

[override virtual]EllipseSelect::~EllipseSelect**()**

Destructor

[override virtual]void **EllipseSelect::**addPoint**(QPointF &*point*, const** int ***vertex* = UNSELECTED)**

Reimplemented from SelectItem::addPoint().

[override virtual]QString EllipseSelect::baseInstructions**() const**

Reimplemented from SelectItem::baseInstructions().

[override virtual]QRectF EllipseSelect::boundingRect**() const**

Reimplemented from QGraphicsEllipseItem::boundingRect().

[override virtual]EllipseSelect *EllipseSelect::getMirror**() const**

Reimplemented from SelectItem::getMirror().

[override virtual]bool **EllipseSelect::**isInside**(const QPointF &*point*) const**

Reimplemented from SelectItem::isInside().

bool **EllipseSelect::**isVisible**() const**

Returns whether the EllipseSelect object is visible (true) on the QGraphicsScene or not (false).

[override virtual]void **EllipseSelect::**mirrorHide**() const**

Reimplemented from SelectItem::mirrorHide().

[override virtual]void **EllipseSelect::**mirrorShow**() const**

Reimplemented from SelectItem::mirrorShow().

[override virtual]void **EllipseSelect::**moveItem**(const QPointF &*oldPos*, QPointF &*newPos*)**

Reimplemented from SelectItem::moveItem().

[override virtual]void **EllipseSelect::**paint**(QPainter ****painter***, const QStyleOptionGraphicsItem ****option***, QWidget ****widget*** = nullptr)**

Reimplemented from QGraphicsEllipseItem::paint().

QGraphicsScene *EllipseSelect::scene**()**

This function overloads SelectItem::scene().

Returns the QGraphicsScene to which this item belongs.

[override virtual]void **EllipseSelect::**setMirror**(SelectItem ****item***)**

Reimplemented from SelectItem::setMirror().

[override virtual]void **EllipseSelect::**setMirrorActive**() const**

Reimplemented from SelectItem::setMirrorActive().

[override virtual]void **EllipseSelect::**setMirrorAdded**() const**

Reimplemented from SelectItem::setMirrorAdded().

[override virtual]void **EllipseSelect::**setMirrorCorners**(QRectF *tlc*, QRectF *blc*, QRectF *trc*, QRectF*brc*) const**

Reimplemented from BoxBasedSelector::setMirrorCorners().

[override virtual]void **EllipseSelect::**setMirrorMoved**() const**

Reimplemented from SelectItem::setMirrorMoved().

[override virtual]void **EllipseSelect::**setMirrorResized**() const**

Reimplemented from SelectItem::setMirrorResized().

[override virtual]void **EllipseSelect::**setMirrorVertex(int ***vertex*) const**

Reimplemented from SelectItem::setMirrorVertex().

void **EllipseSelect::**setOpacity**(qreal *val*)**

* Sets the opacity of the label to *val*

[override virtual]void **EllipseSelect::**setRectUnchecked**(QRectF *rect*)**

Reimplemented from BoxBasedSelector::setRectUnchecked().

[override virtual]void **EllipseSelect::**toPixmap**(QPainter **painter*)**

Reimplemented from SelectItem::toPixmap().

[override virtual]int **EllipseSelect::**type**() const**

Reimplemented from QGraphicsEllipseItem::type().

This function overloads SelectItem::type().

This function returns the type of this item.

See also QGraphicsItem::type() and SelectItem::type().

[override virtual]void **EllipseSelect::**updateMirrorScene**() const**

Reimplemented from SelectItem::updateMirrorScene().

[override virtual]void **EllipseSelect::**updatePen**(QPen *pen*)**

Reimplemented from SelectItem::updatePen().

ErasePointsCommand Class

The ErasePointsCommand class provides undo and redo actions for erasing points from a FreeDrawSelect object. More...

Header:	#include <ErasePointsCommand>
Inherits:	QUndoCommand

- List of all members, including inherited members

Public Functions

	ErasePointsCommand(FreeDrawErase **item*, PhotoScene **scene*, PhotoScene **maskScene*, QUndoCommand **parent* = nullptr)

Reimplemented Public Functions

virtual void	**redo**() override
virtual void	**undo**() override

- 11 public functions inherited from QUndoCommand

Detailed Description

The ErasePointsCommand class provides undo and redo actions for erasing points from a FreeDrawSelect object.

ErasePointsCommand is used to hold a reference to a FreeDrawSelect and the PhotoScenes to which it and its mirror belong. Using the undo() and redo() commands the erasure can be undone and redone repeatedly.

See also AddCommand, MoveCommand, AddVertexCommand, MoveVertexCommand, DeleteVertexCommand, DeleteCommand, RotateCommand, and EditLabelCommand.

Member Function Documentation

ErasePointsCommand::ErasePointsCommand(FreeDrawErase **item*, PhotoScene **scene*, PhotoScene **maskScene*, QUndoCommand **parent* = nullptr)

Constructs an ErasePointsCommand using the provided *item* and it's owning *scene* and the *maskScene* which owns it's mirror so that the erasure of points from this *item*can be undone/redone. *parent* refers to parent QUndoCommand, if any.

[override virtual]void **ErasePointsCommand::**redo**()**

Reimplemented from QUndoCommand::redo().

[override virtual]void **ErasePointsCommand::**undo**()**

Reimplemented from QUndoCommand::undo().

FreeDrawErase Class

The FreeDrawErase class provides a class for erasing regions breated by a FreeDrawSelect. More...

Header:	#include <FreeDrawErase>
Inherits:	FreeDrawSelect

- List of all members, including inherited members
- Obsolete members

Public Functions

	FreeDrawErase(QPointF *point*, int *brushSize*, Qt::PenCapStyle *brushMode*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)
virtual	**~FreeDrawErase**() override
QSharedPointer<EraseMap>	**getMap**() const

void	**setOpacity**(qreal *val*)

Reimplemented Public Functions

virtual bool	**isInside**(const QPointF &*point*) const override
virtual void	**moveItem**(const QPointF &*oldPos*, QPointF &*newPos*) override
virtual void	**paint**(QPainter **painter*, const QStyleOptionGraphicsItem **option*, QWidget **widget* = ...) override
virtual void	**toPixmap**(QPainter **painter*) override

- 38 public functions inherited from FreeDrawSelect
- 16 public functions inherited from QGraphicsPixmapItem
- 57 public functions inherited from SelectItem
- 366 public functions inherited from QGraphicsItem

Static Public Members

QString	**baseInstruction**

- 1 static public member inherited from SelectItem

Additional Inherited Members

- 2 protected functions inherited from SelectItem
- 48 protected functions inherited from QGraphicsItem

Detailed Description

The FreeDrawErase class provides a class for erasing regions breated by a FreeDrawSelect.

Provides a class for erasing from selection regions created by FreeDrawSelect objects. This is done by painting on the canvas with a brush. The region is defined by the pixels that are painted on. This class exists only to remove pixels from existing FreeDrawSelect objects and provide undo/redo capabilities to the deletion.

Member Function Documentation

FreeDrawErase:: FreeDrawErase**(QPointF *point*, int *brushSize*, Qt::PenCapStyle *brushMode*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)**

Constructs a FreeDrawErase instance withan initial selected point *point*, with a brush size of *brushSize* pixels, brush type of *brushMode*, *label* as the Label which owns this object and *item* is the parent QGraphicsItem, if any. Acceptable values for the *brushMode* are:

- Qt::SquareCap
- Qt::RoundCap

[override virtual]FreeDrawErase::~FreeDrawErase**()**

Destructor

QSharedPointer<EraseMap> FreeDrawErase::getMap**() const**

Returns a pointer to an EraseMap containing a map of the points deleted from each FreeDrawSelect object.

[override virtual]bool FreeDrawErase::isInside**(const QPointF &*point*) const**

Reimplemented from FreeDrawSelect::isInside().

Empty function as a point is not in an erased region.

[override virtual]void FreeDrawErase::moveItem**(const QPointF &*oldPos*, QPointF &*newPos*)**

Reimplemented from FreeDrawSelect::moveItem().

[override virtual]void FreeDrawErase::paint**(QPainter *_painter_, const QStyleOptionGraphicsItem *_option_, QWidget *_widget_ = ...)**

Reimplemented from FreeDrawSelect::paint().

Empty function as FreeDrawErase objects are not painted.

void FreeDrawErase::setOpacity**(qreal *val*)**

Empty function as FreeDrawErase objects are not painted. *val* is unused.

[override virtual]void FreeDrawErase::toPixmap**(QPainter *_painter_)**

Reimplemented from FreeDrawSelect::toPixmap().

Empty function as FreeDrawErase objects are not painted.

FreeDrawSelect Class

The FreeDrawSelect class provides a class for annotating regions by painting them. More...

Header:	#include <FreeDrawSelect>

Inherits:	QGraphicsPixmapItem and SelectItem
Inherited By:	FreeDrawErase

- List of all members, including inherited members
- Obsolete members

Public Functions

	FreeDrawSelect()
	FreeDrawSelect(const QPixmap *map*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)
	FreeDrawSelect(const QPixmap *map*, QPen *pen*)
	FreeDrawSelect(const QJsonObject &*json*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)
	FreeDrawSelect(const QList<FreeDrawSelect *> &*items*)
	FreeDrawSelect(QPointF *point*, QPen *pen*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)
	FreeDrawSelect(QPointF *point*, int *brushSize*, Qt::PenCapStyle *brushMode*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)
virtual	**~FreeDrawSelect**() override
void	**addPoints**(QSharedPointer<QPixmap> *points*)
void	**deletePoints**(const QPointF &*start*, const QPointF &*end*, QPen *pen*, QSharedPointer<QPixmap> *outmap*)

void	**deletePoints**(QPen &*pen*, QSharedPointer<QPixmap> *map*)
QPixmap	**getPixmap**() const
void	**importPixmap**(const QPixmap *map*)
bool	**isVisible**() const
QGraphicsScene *	**scene**() const
void	**setMirrorMap**()
void	**setOpacity**(qreal *val*)
void	**setPointsUnchecked**(QPixmap *map*)

Reimplemented Public Functions

virtual void	**addPoint**(QPointF &*point*, const int *vertex* = UNSELECTED) override
virtual QString	**baseInstructions**() const override
virtual QRectF	**boundingRect**() const override
virtual void	**clickPoint**(const QPointF &*point*) override
virtual FreeDrawSelect *	**getMirror**() const override
virtual void	**insertVertex**(const int *vertex*, const QPointF &*point*) override
virtual bool	**isInside**(const QPointF &*point*) const override

virtual void	**mirrorHide**() const override
virtual void	**mirrorShow**() const override
virtual void	**moveItem**(const QPointF &*oldPos*, QPointF &*newPos*) override
virtual int	**numberOfVertices**() const override
virtual void	**paint**(QPainter **painter*, const QStyleOptionGraphicsItem **option*, QWidget **widget* = nullptr) override
virtual void	**read**(const QJsonObject &*json*) override
virtual void	**removeVertex**(int *vertex* = ...) override
virtual void	**resetActiveVertex**() override
virtual void	**resizeItem**(const int *vertex*, QPointF &*oldP*, QPointF &*newP*) override
virtual void	**rotate**(const QPointF &*from*, const QPointF &*to*) override
virtual void	**rotateMirror**() const override
virtual void	**setMirror**(SelectItem **item*) override
virtual void	**setMirrorActive**() const override
virtual void	**setMirrorAdded**() const override
virtual void	**setMirrorMoved**() const override
virtual void	**setMirrorResized**() const override

virtual void	**setMirrorVertex**(int *vertex*) const override
virtual void	**toPixmap**(QPainter **painter*) override
virtual void	**updateMirrorScene**() const override
virtual void	**updatePen**(QPen *pen*) override
virtual void	**write**(QJsonObject &*json*) const override

- 16 public functions inherited from QGraphicsPixmapItem
- 57 public functions inherited from SelectItem
- 366 public functions inherited from QGraphicsItem

Static Public Members

QString	**baseInstruction**

- 1 static public member inherited from SelectItem

Protected Variables

QPoint	**lastPoint**
QPixmap	**myPixmap**

Additional Inherited Members

- 2 protected functions inherited from SelectItem
- 48 protected functions inherited from QGraphicsItem

Detailed Description

The FreeDrawSelect class provides a class for annotating regions by painting them.

Provides a class for creating selection regions defined by painting on the canvas with a brush. The region is defined by the pixels that are painted on.

Member Function Documentation

FreeDrawSelect:: FreeDrawSelect**()**

Constructs a FreeDrawSelect object with no selected points, and a square brush with a size of 2 pixels.

FreeDrawSelect:: FreeDrawSelect**(const QPixmap *map*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem *_item_ = nullptr)**

Constructs a FreeDwarSelect object based on the given *map*, Label *label* and parent *item*, if any.

FreeDrawSelect:: FreeDrawSelect**(const QPixmap *map*, QPen *pen*)**

Constructs a FreeDwarSelect object based on the given *map* and *pen*.

FreeDrawSelect:: FreeDrawSelect**(const QJsonObject &*json*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem *_item_ = nullptr)**

Constructs a FreeDrawSelect object by reading a QJsonObject and setting the internal pixel map to the values given in *json*. *label* is the Label which owns this object and *item* is the parent QGraphicsItem, if any.

FreeDrawSelect:: FreeDrawSelect**(const QList<FreeDrawSelect *> &*items*)**

Constructor used to combine several FreeDrawSelect items given as *items*, into a single instance. This is done by merging all of the internal pixel maps into a single entity, discarding duplicate points.

FreeDrawSelect:: FreeDrawSelect**(QPointF *point*, QPen *pen*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)**

Constructs a FreeDrawSelect object with an initial selected point *point*, with a QPen *pen*, *label* as the Label which owns this object and *item* is the parent QGraphicsItem, if any.

FreeDrawSelect:: FreeDrawSelect**(QPointF *point*,** int ***brushSize*, Qt::PenCapStyle *brushMode*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)**

Constructs a FreeDrawSelect object with an initial selected point *point*, with a brush size of *brushSize* pixels, brush type of *brushMode*, *label* as the Label which owns this object and *item* is the parent QGraphicsItem, if any. Acceptable values for the *brushMode* are:

- Qt::SquareCap
- Qt::RoundCap

[override virtual]FreeDrawSelect::~FreeDrawSelect**()**

Destructor

[override virtual]void **FreeDrawSelect::**addPoint**(QPointF &*point*, const** int ***vertex* = UNSELECTED)**

Reimplemented from SelectItem::addPoint().

void **FreeDrawSelect::**addPoints**(QSharedPointer<QPixmap> *points*)**

Adds *points* to the internal pixel map of selected points.

[override virtual]QString FreeDrawSelect::baseInstructions**() const**

Reimplemented from SelectItem::baseInstructions().

[override virtual]QRectF FreeDrawSelect::boundingRect**() const**

Reimplemented from QGraphicsPixmapItem::boundingRect().

[override virtual]void **FreeDrawSelect::**clickPoint**(const QPointF &*point*)**

Reimplemented from SelectItem::clickPoint().

void **FreeDrawSelect::**deletePoints**(const QPointF &*start*, const QPointF &*end*, QPen *pen*, QSharedPointer<QPixmap> *outmap*)**

Deletes points along the brush stroke defined by *start* and *end*, with the given *pen*, with the changes put in *outmap*.

void **FreeDrawSelect::**deletePoints**(QPen &*pen*, QSharedPointer<QPixmap> *map*)**

This function overloads FreeDrawSelect::deletePoints().

Deletes points based on the pixmap *map* with the given *pen*.

[override virtual]FreeDrawSelect *FreeDrawSelect::getMirror**() const**

Reimplemented from SelectItem::getMirror().

QPixmap FreeDrawSelect::getPixmap**() const**

Returns the internal pixmap.

void FreeDrawSelect::importPixmap(const QPixmap *map*)

Imports a pixmap given as *map*.

[override virtual]void FreeDrawSelect::insertVertex(const int *vertex*, const QPointF &*point*)

Reimplemented from SelectItem::insertVertex().

Empty function since FreeDrawSelect items have no vertices.

[override virtual]bool FreeDrawSelect::isInside(const QPointF &*point*) const

Reimplemented from SelectItem::isInside().

bool FreeDrawSelect::isVisible() const

Returns whether the FreeDrawSelect object is visible (true) on the QGraphicsScene or not (false).

[override virtual]void FreeDrawSelect::mirrorHide() const

Reimplemented from SelectItem::mirrorHide().

[override virtual]void FreeDrawSelect::mirrorShow() const

Reimplemented from SelectItem::mirrorShow().

[override virtual]void FreeDrawSelect::moveItem(const QPointF &*oldPos*, QPointF &*newPos*)

Reimplemented from SelectItem::moveItem().

[override virtual]int FreeDrawSelect::numberOfVertices() const

Reimplemented from SelectItem::numberOfVertices().

Returns 0 since FreeDrawSelect items have no vertices.

[override virtual]void **FreeDrawSelect::**paint**(QPainter ****painter***, const QStyleOptionGraphicsItem ****option***, QWidget ****widget*** **= nullptr)**

Reimplemented from QGraphicsPixmapItem::paint().

[override virtual]void **FreeDrawSelect::**read**(const QJsonObject &***json***)**

Reimplemented from SelectItem::read().

[override virtual]void **FreeDrawSelect::**removeVertex**(**int ***vertex*** **= ...)**

Empty function since FreeDrawSelect items have no vertices.

[override virtual]void **FreeDrawSelect::**resetActiveVertex**()**

Reimplemented from SelectItem::resetActiveVertex().

Empty function since FreeDrawSelect items have no vertices.

[override virtual]void **FreeDrawSelect::**resizeItem**(const** int ***vertex*****, QPointF &*****oldP*****, QPointF &*****newP*****)**

Reimplemented from SelectItem::resizeItem().

[override virtual]void **FreeDrawSelect::**rotate**(const QPointF &*****from*****, const QPointF &*****to*****)**

Reimplemented from SelectItem::rotate().

Empty function since FreeDrawSelect items cannot be rotated.

[override virtual]void **FreeDrawSelect::**rotateMirror**() const**

Reimplemented from SelectItem::rotateMirror().

Empty function since FreeDrawSelect items cannot be rotated.

QGraphicsScene *FreeDrawSelect::scene**() const**

Returns the QGraphicsScene to which this item belongs.

[override virtual]void **FreeDrawSelect::**setMirror**(SelectItem ****item***)**

Reimplemented from SelectItem::setMirror().

[override virtual]void **FreeDrawSelect::**setMirrorActive**() const**

Reimplemented from SelectItem::setMirrorActive().

[override virtual]void **FreeDrawSelect::**setMirrorAdded**() const**

Reimplemented from SelectItem::setMirrorAdded().

void **FreeDrawSelect::**setMirrorMap**()**

Sets the bounding rectangle and pixel map of the mirror, based on the values of this instance.

[override virtual]void **FreeDrawSelect::**setMirrorMoved**() const**

Reimplemented from SelectItem::setMirrorMoved().

[override virtual]void **FreeDrawSelect::**setMirrorResized**() const**

Reimplemented from SelectItem::setMirrorResized().

[override virtual]void **FreeDrawSelect::**setMirrorVertex**(**int ***vertex*****) const**

Reimplemented from SelectItem::setMirrorVertex().

void **FreeDrawSelect::**setOpacity(**qreal** ***val***)

* Sets the opacity of the label to *val*

void **FreeDrawSelect::**setPointsUnchecked(**QPixmap** ***map***)

Sets the objects internal map to *map* without boundary checking.

[override virtual]void **FreeDrawSelect::**toPixmap(**QPainter** ***painter***)

Reimplemented from SelectItem::toPixmap().

[override virtual]void **FreeDrawSelect::**updateMirrorScene**() const**

Reimplemented from SelectItem::updateMirrorScene().

[override virtual]void **FreeDrawSelect::**updatePen(**QPen** ***pen***)

Reimplemented from SelectItem::updatePen().

[override virtual]void **FreeDrawSelect::**write(**QJsonObject** **&*json*) const**

Reimplemented from SelectItem::write().

ImageList Class

Handles list of images form the server. More...

Header:	#include <ImageList>

Inherits:	QWidget

- List of all members, including inherited members
- Obsolete members

Public Functions

	ImageList(Project **project*, QWidget **parent* = nullptr, QString *databaseUrl* = "", QString *token* = "")
virtual	**~ImageList**()
void	**addItems**(QJsonObject *obj*)
void	**fileReplyFinished**()
QList<int>	**getIdList**()
QList<QString>	**getPathList**()
int	**getSelectedIdIndex**()
void	**openAnnotation**()
void	**setAnnotated**()
void	**setSelectedIdIndex**(int *id*)

- 222 public functions inherited from QWidget
- 31 public functions inherited from QObject
- 15 public functions inherited from QPaintDevice

Signals

void	**allAnnotationsLoaded**(QJsonObject *json*, fileTypes *type*)
void	**clearGUI**()
void	**fileDownloaded**(QString *path*)

- 4 signals inherited from QWidget
- 2 signals inherited from QObject

Public Variables

Project *	**currentProject**
QJsonObject	**json**
QJsonArray	**jsonLabelArray**
QList<QTableWidgetItem *>	**selectedRow**

Static Public Members

const QMetaObject	**staticMetaObject**

- 5 static public members inherited from QWidget
- 9 static public members inherited from QObject
- 1 static public member inherited from QPaintDevice

Additional Inherited Members

- 59 properties inherited from QWidget
- 1 property inherited from QObject
- 19 public slots inherited from QWidget
- 1 public slot inherited from QObject
- 35 protected functions inherited from QWidget

- 9 protected functions inherited from QObject
- 1 protected function inherited from QPaintDevice
- 1 protected slot inherited from QWidget

Detailed Description

Handles list of images form the server.

Member Function Documentation

ImageList:: ImageList**(Project ****project***, QWidget ****parent*** **= nullptr, QString** ***databaseUrl*** **= "", QString** ***token*** **= "")**

Creates an ImageList instance with *project*, *parent*, *databaseUrl*, and *token*

[virtual]ImageList::~ImageList**()**

Destructor

void **ImageList::**addItems**(QJsonObject** ***obj*****)**

* Adds Item QJsonObject *obj* to the ImageList.

[signal]void **ImageList::**allAnnotationsLoaded(**QJsonObject** ***json*****,** fileTypes ***type*****)**

Emitted when all annotations have been loaded from *json*, with *type*.

[signal]void **ImageList::**clearGUI**()**

Emitted when the GUI is cleared.

[signal]void **ImageList::**fileDownloaded**(**QString ***path*)

Emitted whne a file has been downloded, giving the *path*.

void **ImageList::**fileReplyFinished**()**

* Waits for the file to be received.

QList**<**int**> ImageList::**getIdList**()**

Returns the idList.

QList**<**QString**> ImageList::**getPathList**()**

Returns the pathList.

int **ImageList::**getSelectedIdIndex**()**

Returns selectedIdIndex.

void **ImageList::**openAnnotation**()**

Opens the annotation file.

void **ImageList::**setAnnotated**()**

Sets the selectedIdIndex annotated flag to true.

void **ImageList::**setSelectedIdIndex**(**int ***id*)

Set the selectedIdIndex ti *id*.

InstaDam Class

The InstaDam class defined the main window for the app. More...

Header:	#include <InstaDam>
Inherits:	QMainWindow

- List of all members, including inherited members
- Obsolete members

Public Functions

	InstaDam(QWidget **parent* = nullptr, QString *databaseURL* = "", QString *token* = "")
virtual	**~InstaDam**()
int	**annotationDraw**(PhotoScene::viewerTypes *type*, SelectItem **item*, QPointF *pos*, const Qt::MouseButton *button*, const Qt::KeyboardModifiers *modifiers*)
int	**annotationTransform**(PhotoScene::viewerTypes *type*, SelectItem **item*, QPointF *pos*, const Qt::MouseButton *button*, const Qt::KeyboardModifiers *modifiers*)
void	**assertError**(std::string *errorMessage*)
void	**clearLayout**(QLayout **layout*)
void	**connectFilters**()
void	**continueDrawingPolygon**(QPointF *pos*)
void	**deleteCurrentObject**(PhotoScene::viewerTypes *phototype*)

void	**exportImages**(bool *asBuffers* = false)
void	**generateLabelFileName**()
void	**inactivateSceneCancelSelection**()
void	**openFile_and_labels**()
void	**resetGUIclearLabels**()
void	**saveAndProgress**(int *num*)
void	**selectItemButton**(SelectItem::SelectType *type*)
void	**selectItemButton**(int *type*)
void	**setButtonsConfiguration**()
void	**setCurrentItem**(SelectItem **item*, bool *enable* = false)
void	**setCurrentProjectId**(int *id*)
void	**setLabels**()

- 48 public functions inherited from QMainWindow
- 222 public functions inherited from QWidget
- 31 public functions inherited from QObject
- 15 public functions inherited from QPaintDevice

Public Slots

void	**editLabel**(QSharedPointer<Label> *newLabel*)

void	**fileDownloaded**(QString *path*)
bool	**loadLabelJson**(QJsonObject *json*, fileTypes *fileType*)
void	**resetPixmapButtons**()

- 3 public slots inherited from QMainWindow
- 19 public slots inherited from QWidget
- 1 public slot inherited from QObject

Public Variables

QString	**annotationPath**
QFileInfo	**file**
int	**fileId**
QString	**filename**
filterControls *	**filterControl**
ImageList *	**il**
QStringList	**imagesList**
QString	**labelFile**
int	**labelIdsRecieved**
QVector<QString>	**labelPaths**

QList<PicPushButton *>	**maskButtonList**
QList<EnumConstants::maskTypes>	**maskTypeList**
QString	**oldAnnotationPath**
QFileInfo	**oldFile**
QString	**oldFilename**
QStringList	**oldImagesList**
QVector<QString>	**oldLabelPaths**
QDir	**oldPath**
QDir	**path**
bool	**photoLoaded**
bool	**runningLocally**
PhotoScene::viewerTypes	**selectedViewer**

Static Public Members

const QMetaObject	**staticMetaObject**

- 5 static public members inherited from QWidget
- 9 static public members inherited from QObject
- 1 static public member inherited from QPaintDevice

Additional Inherited Members

- 8 properties inherited from QMainWindow
- 59 properties inherited from QWidget
- 1 property inherited from QObject
- 3 signals inherited from QMainWindow
- 4 signals inherited from QWidget
- 2 signals inherited from QObject
- 2 protected functions inherited from QMainWindow
- 35 protected functions inherited from QWidget
- 9 protected functions inherited from QObject
- 1 protected function inherited from QPaintDevice
- 1 protected slot inherited from QWidget

Detailed Description

The InstaDam class defined the main window for the app.

InstaDam is the main window class for the InstaDam app. The class includes all the signals, initialization and slots of the various buttons and tools that can be selected from the main window.

Member Function Documentation

InstaDam:: InstaDam**(QWidget *****parent*** **= nullptr, QString** ***databaseURL*** **= "", QString** ***token*** **= "")**

Constructs an InstaDam window given QString *databaseURL*, Qstring *token* and QMainWindow *parent*, if any. The constructor connects all the buttons in the main window.

[virtual]InstaDam::~InstaDam**()**

Destructor

int **InstaDam::**annotationDraw**(PhotoScene::viewerTypes *type*, SelectItem ****item*, QPointF *pos*, const Qt::MouseButton *button*, const Qt::KeyboardModifiers *modifiers*)**

Draws annotations on the PhotoScene indicated by *type*, with *item*, position *pos*, with mouse button *button* depressed, and any keyboard modifiers given by *modifiers*.

int **InstaDam::**annotationTransform**(PhotoScene::viewerTypes *type*, SelectItem ****item*, QPointF *pos*, const Qt::MouseButton *button*, const Qt::KeyboardModifiers *modifiers*)**

Transforms an annotation, *item*, on *type*, at *pos*, with *button* depressed and keyboard modifiers *modifiers*.

void **InstaDam::**assertError**(**std::string ***errorMessage*)**

Displays the std::string *errorMessage* as a QMessageBox.

void **InstaDam::**clearLayout**(QLayout ****layout*)**

Removes all items from the QLayout *layout*.

void **InstaDam::**connectFilters**()**

Connects the filter buttons to the corresponding maskType.

void **InstaDam::**continueDrawingPolygon**(QPointF *pos*)**

Re-enable a PolygonSelect item for editing, at *pos*.

void **InstaDam::**deleteCurrentObject**(PhotoScene::viewerTypes *phototype*)**

Deletes the current SelectItem on *phototype*

[slot]void **InstaDam::**editLabel**(**QSharedPointer**<**Label**>** ***newLabel*****)**

Edit label given as *newLabel*.

void **InstaDam::**exportImages**(**bool ***asBuffers*** **= false)**

Saves the current annotations as rasterized PNGs or as a buffer depending on the value of *asBuffers*

[slot]void **InstaDam::**fileDownloaded**(**QString ***path*****)**

Opens a file stored at Qstring *path* for the web-assembly version.

void **InstaDam::**generateLabelFileName**()**

Generates the annotation and export file names to be stored locally based on the location and name of the image files.

void **InstaDam::**inactivateSceneCancelSelection**()**

Inactivates all SelectItems in the scene and calcels the current selection.

[slot]bool **InstaDam::**loadLabelJson**(**QJsonObject ***json*****,** fileTypes ***fileType*****)**

InstaDam::loadLabelJson loads the json object *json* of fileTypes *fileType* where *fileType* is either a PROJECT or ANNOTATION. Returns a bool indicating success (true) or failure (false) of the load.

void **InstaDam::**openFile_and_labels**()**

Uses defined QStringList of images in the path as well as the id of the current file and opens the file. If annotations exist, the annotations are opened.

void **InstaDam::**resetGUIclearLabels**()**

Clears the guis and resets the labels()

[slot]void **InstaDam::**resetPixmapButtons**()**

Resets the pixmaps of the filter buttons.

void **InstaDam::**saveAndProgress**(**int ***num*****)**

Save current annotations and continue. *num* is used to indicate the index of the current image.

void **InstaDam::**selectItemButton**(SelectItem::SelectType *****type*****)**

Slot called when a selectItemButton is called with *type*, enabling the appropriate functionality.

void **InstaDam::**selectItemButton**(**int ***type*****)**

* Selects the appropriate button based on *type*.

void **InstaDam::**setButtonsConfiguration**()**

Configures different buttons for local and server versions.

void **InstaDam::**setCurrentItem**(SelectItem *****item*****,** bool ***enable* = false)**

Set *item* to be the currently active SelectItem. *enable* indicates whenther (true) or not (false) the finishPolygon Button needs to be enabled.

void **InstaDam::**setCurrentProjectId**(**int ***id*****)**

Sets the id of the current project to *id*.

void **InstaDam::**setLabels**()**

Creates sets the label buttons to the labels of the current project.

Label Class

The Label class provides a class for holding SelectItems. More...

Header:	#include <Label>
Inherits:	QEnableSharedFromThis

- List of all members, including inherited members

Public Functions

	Label()
	Label(const QJsonObject &*json*, int *j*, bool *server* = false)
	~Label()
void	**addItem**(FreeDrawErase **item*)
void	**addItem**(FreeDrawSelect **item*)
void	**addItem**(RectangleSelect **item*)
void	**addItem**(EllipseSelect **item*)

void	**addItem**(PolygonSelect **item*)
void	**addItem**(SelectItem **item*)
void	**clear**()
QPixmap	**exportLabel**(const QSize &*rect*) const
QColor	**getColor**() const
int	**getId**()
QString	**getText**() const
void	**read**(const QJsonObject &*json*)
void	**readServer**(const QJsonObject &*json*)
void	**removeItem**(const int *id*)
void	**setColor**(QColor *col*)
void	**setId**(int *j*)
void	**setMaskState**(int *state*)
void	**setOpacity**(int *val*)
void	**setText**(const QString *tx*)
void	**write**(QJsonObject &*json*) const
void	**writeIdantn**(QJsonObject &*json*) const

- 2 public functions inherited from QEnableSharedFromThis

Public Variables

QHash<int, EllipseSelect *>	**ellipseObjects**
QHash<int, FreeDrawSelect *>	**freeDrawObjects**
QHash<int, PolygonSelect *>	**polygonObjects**
QHash<int, RectangleSelect *>	**rectangleObjects**

Detailed Description

The Label class provides a class for holding SelectItems.

Provides a class for holing SelectItems which annotate the same type of feature.

Member Function Documentation

Label:: Label**()**

Default constructor

Label:: Label**(const QJsonObject &*json*,** int ***j*,** bool ***server* = false)**

Constructs a Label object based on data read in from *json*. The Label color and text are read from the QJsonObject and any SelectItems in the object are constructed and added to this Label. *j* indicates the Label id, and *server* denotes whether to read the file from a server (true) or local file (false).

Label::~Label**()**

Destructor

void **Label::**addItem**(FreeDrawErase ****item***)**

Empty function as FreeDrawErase (*item*) objects are not saved.

void **Label::**addItem**(FreeDrawSelect ****item***)**

This function overloads addItem().

This convenience function adds a FreeDrawSelect object to the Label

void **Label::**addItem**(RectangleSelect ****item***)**

This function overloads addItem().

This convenience function adds a RectangleSelect object to the Label

void **Label::**addItem**(EllipseSelect ****item***)**

This function overloads addItem().

This convenience function adds an EllipseSelect object to the Label

void **Label::**addItem**(PolygonSelect ****item***)**

This function overloads addItem().

This convenience function adds a PolygonSelect object to the Label void **Label::**addItem**(SelectItem** ***item*)**

This function overloads addItem().

This convenience function adds a SelectItem object to the Label

void **Label::**clear**()**

Clears the label data in memory.

QPixmap Label::exportLabel**(const QSize &*rect*) const**

Creates a QPixmap bitmask based on all of the SelectItems that this Label holds. It paints each item based on their internal parameters. *rect* specifies the size of the output QPixmap.

QColor Label::getColor**() const**

Convenience function which returns the QColor of this Label.

int **Label::**getId**()**

Convenience funtion for getting the id of this Label.

QString Label::getText**() const**

Convenience funtion for getting the test of this Label as a QString

void **Label::**read**(const QJsonObject &*json*)**

Reads a QJsonObject *json* and sets the Label annotation's to the data it reads. Any SelectItems found in the QJsonObject are also constructed and added to this Label.

void **Label::**readServer**(const QJsonObject &*json*)**

Read a QJsonObject *json* into this object.

void Label::removeItem**(const int *id*)**

Removes a SelectItem from the Label based on it's unique ID given as *id*.

void Label::setColor**(QColor *col*)**

Convenience function for setting the color of this Label to *col*.

void Label::setId**(int *j*)**

Convenience funtion for setting the id of this Label as *j*.

void Label::setMaskState**(int *state*)**

Sets the state of the associated SelectItems to be hidden or shown depending on *state*. Values are

- Qt::Unchecked hidden
- Qt::Checked Shown

void Label::setOpacity**(int *val*)**

Set the opacity of the related SelectItems to *val* as an integer percentage.

void Label::setText**(const QString *tx*)**

Convenience funtion for setting the text of this laael to *tx*

void Label::write**(QJsonObject &*json*) const**

Writes the contents of the Label class to QJsonObject *json*.

void Label::writeIdantn(**QJsonObject &*json*) const**

Writes the contents of the Label data to QJsonObject *json*. Any SelectItems this Label holds are written.

LabelButton Class

A button for seleting the Label for annotation. More...

Header:	#include <LabelButton>
Inherits:	QPushButton

- List of all members, including inherited members
- Obsolete members

Public Functions

LabelButton(QSharedPointer<Label> *label*)

- 10 public functions inherited from QPushButton
- 21 public functions inherited from QAbstractButton
- 222 public functions inherited from QWidget
- 31 public functions inherited from QObject
- 15 public functions inherited from QPaintDevice

Public Slots

void	**reemitValueChanged**(int *value*)
void	**wasClicked**()

- 1 public slot inherited from QPushButton

- 5 public slots inherited from QAbstractButton
- 19 public slots inherited from QWidget
- 1 public slot inherited from QObject

Signals

void	**cclicked**(QSharedPointer<Label> *label*)
void	**opacity**(QSharedPointer<Label> *label*, int *op*)

- 4 signals inherited from QAbstractButton
- 4 signals inherited from QWidget
- 2 signals inherited from QObject

Public Variables

QSharedPointer<Label>	**myLabel**
QSlider *	**slider**

Static Public Members

const QMetaObject	**staticMetaObject**

- 5 static public members inherited from QWidget
- 9 static public members inherited from QObject
- 1 static public member inherited from QPaintDevice

Additional Inherited Members

- 3 properties inherited from QPushButton
- 11 properties inherited from QAbstractButton
- 59 properties inherited from QWidget
- 1 property inherited from QObject
- 6 protected functions inherited from QPushButton

- 14 protected functions inherited from QAbstractButton
- 35 protected functions inherited from QWidget
- 9 protected functions inherited from QObject
- 1 protected function inherited from QPaintDevice
- 1 protected slot inherited from QWidget

Detailed Description

A button for seleting the Label for annotation.

Member Function Documentation

LabelButton:: LabelButton**(**QSharedPointer**<Label>** ***label*)

Creates an instance, associated with *label*.

[signal]void **LabelButton::**cclicked**(**QSharedPointer**<Label>** ***label*)

Emitted when the button is clicked, giving *label*.

[signal]void **LabelButton::**opacity**(**QSharedPointer**<Label>** ***label*, int ***op*)

Emitted when the opacity slider changes, giving *label* and *op*.

[slot]void **LabelButton::**reemitValueChanged**(**int ***value*)

Signals the value changed to *value*.

[slot]void **LabelButton::**wasClicked**()**

Signals the button was clicked.

LabelButtonFilter Class

Button for filters. More...

Header:	#include <LabelButtonFilter>
Inherits:	QPushButton

- List of all members, including inherited members
- Obsolete members

Public Functions

	LabelButtonFilter(QSharedPointer<Label> *label*)

- 10 public functions inherited from QPushButton
- 21 public functions inherited from QAbstractButton
- 222 public functions inherited from QWidget
- 31 public functions inherited from QObject
- 15 public functions inherited from QPaintDevice

Public Slots

void	**wasClicked**()

- 1 public slot inherited from QPushButton
- 5 public slots inherited from QAbstractButton
- 19 public slots inherited from QWidget
- 1 public slot inherited from QObject

Signals

void	**cclicked**(QSharedPointer<Label> *label*)

- 4 signals inherited from QAbstractButton
- 4 signals inherited from QWidget
- 2 signals inherited from QObject

Public Variables

QSharedPointer<Label>	**myLabel**

Static Public Members

const QMetaObject	**staticMetaObject**

- 5 static public members inherited from QWidget
- 9 static public members inherited from QObject
- 1 static public member inherited from QPaintDevice

Additional Inherited Members

- 3 properties inherited from QPushButton
- 11 properties inherited from QAbstractButton
- 59 properties inherited from QWidget
- 1 property inherited from QObject
- 6 protected functions inherited from QPushButton
- 14 protected functions inherited from QAbstractButton
- 35 protected functions inherited from QWidget
- 9 protected functions inherited from QObject
- 1 protected function inherited from QPaintDevice
- 1 protected slot inherited from QWidget

Detailed Description

Button for filters.

Member Function Documentation

LabelButtonFilter:: LabelButtonFilter**(**QSharedPointer**<**Label**>** ***label*****)**

Creates an instance based on *label*.

[signal]void **LabelButtonFilter::**cclicked**(**QSharedPointer**<**Label**>** ***label*****)**

This signal is emitted when a filter button is clicked giving *label*.

[slot]void **LabelButtonFilter::**wasClicked**()**

Signals the button was clicked.

Login Class

Creates a login dialog. More...

Header:	#include <Login>
Inherits:	QWidget

- List of all members, including inherited members

- Obsolete members

Public Functions

	Login(QWidget **parent* = nullptr)
virtual	**~Login**()

- 222 public functions inherited from QWidget
- 31 public functions inherited from QObject
- 15 public functions inherited from QPaintDevice

Static Public Members

const QMetaObject	**staticMetaObject**

- 5 static public members inherited from QWidget
- 9 static public members inherited from QObject
- 1 static public member inherited from QPaintDevice

Additional Inherited Members

- 59 properties inherited from QWidget
- 1 property inherited from QObject
- 19 public slots inherited from QWidget
- 1 public slot inherited from QObject
- 4 signals inherited from QWidget
- 2 signals inherited from QObject
- 35 protected functions inherited from QWidget
- 9 protected functions inherited from QObject
- 1 protected function inherited from QPaintDevice
- 1 protected slot inherited from QWidget

Detailed Description

Creates a login dialog.

Member Function Documentation

Login:: Login**(QWidget *_parent_ = nullptr)**

Creates an instance with parent QWidget *parent*, if any.

[virtual]Login::~Login**()**

Destructor

MaskViewer Class

A masked view of the image in the associated PhotoViewer. More...

Header:	#include <MaskViewer>
Inherits:	PhotoViewer

- List of all members, including inherited members
- Obsolete members

Public Functions

	MaskViewer(QWidget **parent* = nullptr)
void	**LinkToPhotoViewer**(PhotoViewer **viewer*)

- 16 public functions inherited from PhotoViewer
- 78 public functions inherited from QGraphicsView
- 20 public functions inherited from QAbstractScrollArea
- 14 public functions inherited from QFrame
- 222 public functions inherited from QWidget
- 31 public functions inherited from QObject
- 15 public functions inherited from QPaintDevice

Public Slots

void	**setImMask**(EnumConstants::maskTypes *filterName*)

- 2 public slots inherited from PhotoViewer
- 3 public slots inherited from QGraphicsView
- 19 public slots inherited from QWidget
- 1 public slot inherited from QObject

Public Variables

PhotoViewer *	**photoViewer**
QPixmap	**qAlpha**
QPixmap	**qImg**

Static Public Members

const QMetaObject	**staticMetaObject**

- 5 static public members inherited from QWidget
- 9 static public members inherited from QObject

- 1 static public member inherited from QPaintDevice

Additional Inherited Members

- 13 properties inherited from QGraphicsView
- 3 properties inherited from QAbstractScrollArea
- 6 properties inherited from QFrame
- 59 properties inherited from QWidget
- 1 property inherited from QObject
- 3 signals inherited from PhotoViewer
- 1 signal inherited from QGraphicsView
- 4 signals inherited from QWidget
- 2 signals inherited from QObject
- 24 protected functions inherited from QGraphicsView
- 20 protected functions inherited from QAbstractScrollArea
- 4 protected functions inherited from QFrame
- 35 protected functions inherited from QWidget
- 9 protected functions inherited from QObject
- 1 protected function inherited from QPaintDevice
- 1 protected slot inherited from QGraphicsView
- 1 protected slot inherited from QWidget

Detailed Description

A masked view of the image in the associated PhotoViewer.

Member Function Documentation

MaskViewer:: MaskViewer(**QWidget** ****parent*** **= nullptr)**

Creates an instance with parent QWidget *parent*, if any.

void **MaskViewer::**LinkToPhotoViewer**(PhotoViewer *****viewer*****)**

Links this object to *viewer*.

[slot]void **MaskViewer::**setImMask(**EnumConstants::maskTypes *filterName***)

Sets the image mask to *filterName*.

MoveCommand Class

The MoveCommand class provides undo and redo actions for moving a SelectItem on a PhotoScene. More...

Header:	#include <MoveCommand>
Inherits:	QUndoCommand

- List of all members, including inherited members

Public Functions

MoveCommand(SelectItem **item*, const QPointF *oldPos*, const QPointF *newPos*, QUndoCommand **parent* = nullptr)

Reimplemented Public Functions

virtual void	**redo**() override
virtual void	**undo**() override

- 11 public functions inherited from QUndoCommand

Detailed Description

The MoveCommand class provides undo and redo actions for moving a SelectItem on a PhotoScene.

DeleteCommand is used to hold a reference to a SelectItem and the starting and ending positions of the move. Using the undo() and redo() commands the item can be moved and returned to it's original position in the scene repeatedly.

See also AddCommand, DeleteCommand, MoveVertexCommand, AddVertexCommand, DeleteVertexCommand, ErasePointsCommand, RotateCommand, and EditLabelCommand.

Member Function Documentation

MoveCommand:: MoveCommand**(SelectItem *_item_, const QPointF _oldPos_, const QPointF _newPos_, QUndoCommand *_parent_ = nullptr)**

Constructs a MoveCommand using the provided *item* so that the it's movement from *oldPos* to *newPos* on the scene can be undone and/or redone. *parent* refers to parent QUndoCommand, if any.

[override virtual]void **MoveCommand::**redo**()**

Reimplemented from QUndoCommand::redo().

[override virtual]void **MoveCommand::**undo**()**

Reimplemented from QUndoCommand::undo().

MoveVertexCommand Class

The MoveVertexCommand class provides undo and redo actions for moving a vertex of a SelectItem on a PhotoScene. More...

Header:	#include <MoveVertexCommand>
Inherits:	QUndoCommand

- List of all members, including inherited members

Public Functions

MoveVertexCommand(SelectItem **item*, const QPointF *oldPos*, const QPointF *newPos*, const int *vertex*, QUndoCommand **parent* = nullptr)

Reimplemented Public Functions

virtual void	**redo**() override
virtual void	**undo**() override

- 11 public functions inherited from QUndoCommand

Detailed Description

The MoveVertexCommand class provides undo and redo actions for moving a vertex of a SelectItem on a PhotoScene.

MoveVertexCommand is used to hold a reference to a SelectItem, the starting and ending positions, and which vertex is moving. Using the undo() and redo() commands the vertex can be moved from and to its original position in the scene repeatedly.

See also AddCommand, MoveCommand, DeleteCommand, AddVertexCommand, DeleteVertexCommand, ErasePointsCommand, RotateCommand, and EditLabelCommand.

Member Function Documentation

MoveVertexCommand:: MoveVertexCommand**(SelectItem ****item***, const QPointF** ***oldPos*****, const QPointF** ***newPos*****, const** int ***vertex*****, QUndoCommand *****parent*** **= nullptr)**

Constructs a MoveVeretxCommand using the provided *item* and *vertex* so that it's movement from *oldPos* to *newPos* on the scene can be undone and/or redone. *parent*refers to parent QUndoCommand, if any.

[override virtual]void **MoveVertexCommand::**redo**()**

Reimplemented from QUndoCommand::redo().

[override virtual]void **MoveVertexCommand::**undo**()**

Reimplemented from QUndoCommand::undo().

PhotoScene Class

The PhotoScene class provides a scene for rendering SelectItems. More...

Header:	#include <PhotoScene>
Inherits:	QGraphicsScene

- List of all members, including inherited members
- Obsolete members

Public Types

enum	**viewerTypes** { PHOTO_VIEWER_TYPE, MASK_VIEWER_TYPE }

Public Functions

	PhotoScene(PhotoScene::viewerTypes _type_, QObject *_parent_ = nullptr)
void	**addItem**(SelectItem *_item_)
void	**addItem**(QGraphicsItem *_item_)
void	**addLabel**(std::string _label_)
void	**addLabelItem**(SelectItem *_item_, std::string _label_)
void	**clearItems**()
void	**inactiveAll**() const
SelectItem *	**itemAt**(QPointF _point_) const
SelectItem *	**itemAt**(QPointF _point_, std::string _label_)
void	**removeItem**(SelectItem *_item_)

void	**setCurrentLabel**(const std::string &*label*)

- 78 public functions inherited from QGraphicsScene
- 31 public functions inherited from QObject

Signals

void	**keyPressed**(PhotoScene::viewerTypes *type*, const int *key*)
void	**mouseMoved**(const QPointF *fromPos*, const QPointF *toPos*, const Qt::KeyboardModifiers *modifiers*)
void	**mouseReleased**(const PhotoScene::viewerTypes *type*, const QPointF *oldPos*, const QPointF *newPos*, const Qt::MouseButton *button*, const Qt::KeyboardModifiers *modifiers*)
void	**pointClicked**(const PhotoScene::viewerTypes *type*, SelectItem **item*, const QPointF *point*, const Qt::MouseButton *button*, const Qt::KeyboardModifiers *modifiers*)

- 4 signals inherited from QGraphicsScene
- 2 signals inherited from QObject

Public Variables

PhotoScene::viewerTypes	**myViewerType**

Static Public Members

const QMetaObject	**staticMetaObject**

- 9 static public members inherited from QObject

Reimplemented Protected Functions

virtual void	**keyPressEvent**(QKeyEvent *_event_) override
virtual void	**mouseMoveEvent**(QGraphicsSceneMouseEvent *_event_) override
virtual void	**mousePressEvent**(QGraphicsSceneMouseEvent *_event_) override
virtual void	**mouseReleaseEvent**(QGraphicsSceneMouseEvent *_event_) override

- 20 protected functions inherited from QGraphicsScene
- 9 protected functions inherited from QObject

Additional Inherited Members

- 10 properties inherited from QGraphicsScene
- 1 property inherited from QObject
- 5 public slots inherited from QGraphicsScene
- 1 public slot inherited from QObject
- 20 protected functions inherited from QGraphicsScene
- 9 protected functions inherited from QObject
- 1 protected slot inherited from QGraphicsScene

Detailed Description

The PhotoScene class provides a scene for rendering SelectItems.

Provides a class for rendering SelectItems on top of a background image. This class emits signals when the mouse moves and/or clicks and when keystrokes are made. It holds an internal list of SelectItems that have been drawn in the scene.

Member Type Documentation

enum PhotoScene::viewerTypes

This enum is used to indicate which panel this PhotoScene belongs to.

Constant	Value	Description
PhotoScene::PHOTO_VIEWER_TYPE	0	The scene containing the opened image.
PhotoScene::MASK_VIEWER_TYPE	1	The scene containing the masked image.

Member Function Documentation

PhotoScene:: PhotoScene(**PhotoScene::viewerTypes *type*, QObject ****parent* = nullptr)**

Constructs a PhotoScene instance of type *type*, with parent *parent*, if any.

void **PhotoScene::**addItem(**SelectItem ****item*)

This function overloads addItem().

Adds *item* to the scene.

void **PhotoScene::**addItem(**QGraphicsItem ****item*)

This function overloads addItem().

Adds *item* to the scene.

void **PhotoScene::**addLabel(**std::string *label***)

Adds *label* to the PhotoScene

void **PhotoScene::**addLabelItem(**SelectItem** **item*, std::string ***label***)

Adds *item* to the Label given by *label*.

void **PhotoScene::**clearItems**()**

This function overloads clearItems().

Clears every item in the scene.

void **PhotoScene::**inactiveAll**() const**

Sets all SelectItems in the PhotoScene to inactive.

SelectItem *PhotoScene::itemAt(QPointF ***point***) **const**

Returns a pointer to the top SelectItem at *point* in scene coordinates. If there is no SelectItem under that position then nullptr is returned.

SelectItem *PhotoScene::itemAt(QPointF ***point***, std::string ***label***)

This function overloads itemAt().

Returns a pointer to the top SelectItem at *point* in scene coordinates and with Label *label*. If there is no SelectItem from that Label under that position then nullptr is returned.

[override virtual protected]void **PhotoScene::**keyPressEvent**(QKeyEvent *event)**

Reimplemented from QGraphicsScene::keyPressEvent().

This function overloads keyPressEvent().

Executed when a key is pressed. *event* contains the key information.

See also QGraphicsScene::keyPressEvent().

[signal]void **PhotoScene::**keyPressed**(PhotoScene::viewerTypes *type*, const** int ***key*)**

Triggered when a key is pressed in the PhotoScene. The signal contains which type of viewer this is as *type* and what key was pressed as *key*.

[override virtual protected]void **PhotoScene::**mouseMoveEvent**(QGraphicsSceneMouseEvent *event)**

Reimplemented from QGraphicsScene::mouseMoveEvent().

This function overloads mouseMoveEvent().

Executed when the mouse moves in the PhotoScene. *event* contains the new position of the mouse. If a mouse button is depressed a mouseMoved() signal is emitted.

See also QGraphicsScene::mouseMoveEvent().

[signal]void **PhotoScene::**mouseMoved**(const QPointF *fromPos*, const QPointF *toPos*, const Qt::KeyboardModifiers *modifiers*)**

Triggered when the mouse moves in the PhotoScene and a mouse button is depressed. The signal contains the starting (*fromPos*) and ending (*toPos*) positions of the mouse and any key modifiers as *modifiers*.

[override virtual protected]void **PhotoScene::**mousePressEvent**(QGraphicsSceneMouseEvent *event)**

Reimplemented from QGraphicsScene::mousePressEvent().

This function overloads mousePressEvent().

Executed when a mouse button is pressed. *event* contains which button was pressed and where in the PhotoScene. The PhotoScene searches for any SelectItems under the clicked point and emits a pointClicked() signal with the result.

See also QGraphicsScene::mousePressEvent().

[override virtual protected]void **PhotoScene::**mouseReleaseEvent**(QGraphicsSceneMouseEvent ****event*****)**

Reimplemented from QGraphicsScene::mouseReleaseEvent().

This function overloads mouseReleaseEvent().

Executed when a mouse button is released. *event* contains which button was released. A mouseReleased() signal is emitted with the PhotoScene type, any mouse movement information, and which button was released.

See also QGraphicsScene::mouseReleaseEvent().

[signal]void **PhotoScene::**mouseReleased**(const PhotoScene::viewerTypes *type*, const QPointF *oldPos*, const QPointF *newPos*, const Qt::MouseButton *button*, const Qt::KeyboardModifiers *modifiers*)**

Triggered when a mouse button is released. The signal contains which type of viewer this is as *type*, he starting (*oldPos*) and ending (*newPos*) positions of the mouse (during the time the button was depressed), which mouse button was released as *button*, and any key modifiers as *modifiers*.

[signal]void **PhotoScene::**pointClicked**(const PhotoScene::viewerTypes *type*, SelectItem ****item*****, const QPointF*point*, const Qt::MouseButton *button*, const Qt::KeyboardModifiers *modifiers*)**

Triggered when a mouse button is depressed in the PhotoScene. The signal contains which type of viewer this is as *type*, any SelectItem that exists under the mouse as *item*, the position of the mouse as *point*, which mouse button was depressed as *button*, and any key modifiers as *modifiers*.

void **PhotoScene::**removeItem**(SelectItem ****item*****)**

This function overloads removeItem().

Adds *item* to the scene.

void **PhotoScene::**setCurrentLabel**(const std::string &*label*)**

Sets the currently active Label to be *label*.

PhotoViewer Class

Viewer for displaying an image and drawing SelectItems on it. More...

Header:	#include <PhotoViewer>
Inherits:	QGraphicsView
Inherited By:	MaskViewer

- List of all members, including inherited members
- Obsolete members

Public Functions

	PhotoViewer(QWidget **parent* = nullptr)
cv::Mat	**QImage2Mat**(const QImage &*src*) const
cv::Mat	**QPixmap2Mat**(QPixmap *px*) const
void	**fitInView**()
void	**resetBrush**(int *size* = 10, Qt::PenCapStyle *capStyle_input* = Qt::RoundCap)
void	**setBrushMode**(Qt::PenCapStyle *cap*)
void	**setFilterControls**(filterControls **fc*)

void	**setPanMode**(bool *mode*)
void	**setPhoto**(QPixmap *pixmap*)
QSize	**setPhotoFromFile**(QString *filename*)
void	**setPhotoFromPixmap**(QPixmap *px*)
void	**testPixmap**()

Reimplemented Public Functions

virtual void	**mouseMoveEvent**(QMouseEvent **event*) override
virtual void	**mousePressEvent**(QMouseEvent **event*) override
virtual void	**mouseReleaseEvent**(QMouseEvent **event*) override
virtual void	**resizeEvent**(QResizeEvent **event*) override
virtual void	**wheelEvent**(QWheelEvent **event*) override

- 78 public functions inherited from QGraphicsView
- 20 public functions inherited from QAbstractScrollArea
- 14 public functions inherited from QFrame
- 222 public functions inherited from QWidget
- 31 public functions inherited from QObject
- 15 public functions inherited from QPaintDevice

Public Slots

void	**setImMask**(EnumConstants::maskTypes *filterName*, EnumConstants::threshold_or_filter *thof* = EnumConstants::FILTER)

void	**zoomedInADifferentView**(int *zoom_input*, float *factor*, QPointF *point*)

- 3 public slots inherited from QGraphicsView
- 19 public slots inherited from QWidget
- 1 public slot inherited from QObject

Signals

void	**changedMask**(EnumConstants::maskTypes *type*)
void	**loadedPhoto**()
void	**zoomed**(int *i*, float *a*, QPointF *point*)

- 1 signal inherited from QGraphicsView
- 4 signals inherited from QWidget
- 2 signals inherited from QObject

Public Variables

QGraphicsPathItem *	**brush**
int	**brushSize**
brushTypes	**brushType**
Qt::PenCapStyle	**capStyle**
int **	**colors**
QPixmap	**currentMap**
cv::Mat	**cvImage**

cv::Mat	**cvThumb**
filterControls *	**filterControl**
QGraphicsPixmapItem *	**filterIm**
bool	**hasPhoto**
QPixmap	**imMask**
QPoint	**lastPos**
maskObjects *	**maskObject**
bool	**paintMode**
QPainterPath	**path**
QPen	**pen**
QGraphicsPixmapItem *	**photo**
QPixmap	**pixmapFilt**
PhotoScene *	**scene**
EnumConstants::maskTypes	**selectedMask**
PhotoScene::viewerTypes	**viewerType**
int	**zoom**

Static Public Members

const QMetaObject	**staticMetaObject**

- 5 static public members inherited from QWidget
- 9 static public members inherited from QObject
- 1 static public member inherited from QPaintDevice

Additional Inherited Members

- 13 properties inherited from QGraphicsView
- 3 properties inherited from QAbstractScrollArea
- 6 properties inherited from QFrame
- 59 properties inherited from QWidget
- 1 property inherited from QObject
- 24 protected functions inherited from QGraphicsView
- 20 protected functions inherited from QAbstractScrollArea
- 4 protected functions inherited from QFrame
- 35 protected functions inherited from QWidget
- 9 protected functions inherited from QObject
- 1 protected function inherited from QPaintDevice
- 1 protected slot inherited from QGraphicsView
- 1 protected slot inherited from QWidget

Detailed Description

Viewer for displaying an image and drawing SelectItems on it.

Member Function Documentation

PhotoViewer:: PhotoViewer**(QWidget** ***parent*** **= nullptr)**

Craetes an instance with parent QWidget *parent*.

cv::Mat PhotoViewer::QImage2Mat**(const QImage &*src*) const**

Converts *src* to a cv::Mat.

cv::Mat PhotoViewer::QPixmap2Mat**(QPixmap *px*) const**

This function overloads QImage2Mat.

Converts *px* to a cv::Mat.

[signal]void PhotoViewer::changedMask**(EnumConstants::maskTypes *type*)**

This signal is emitted when the mask is changed to *type*.

void PhotoViewer::fitInView**()**

Something

[signal]void PhotoViewer::loadedPhoto**()**

This signal is emitted when a photo is loaded.

[override virtual]void PhotoViewer::mouseMoveEvent**(QMouseEvent *event)**

Reimplemented from QGraphicsView::mouseMoveEvent().

[override virtual]void PhotoViewer::mousePressEvent**(QMouseEvent *event)**

Reimplemented from QGraphicsView::mousePressEvent().

[override virtual]void PhotoViewer::mouseReleaseEvent**(QMouseEvent *event)**

Reimplemented from QGraphicsView::mouseReleaseEvent().

void PhotoViewer::resetBrush**(int *size* = 10, Qt::PenCapStyle *capStyle_input* = Qt::RoundCap)**

Resets the internal brush to *size* and style *capStyle_input*.

[override virtual]void PhotoViewer::resizeEvent**(QResizeEvent **event*)**

Reimplemented from QGraphicsView::resizeEvent().

void PhotoViewer::setBrushMode**(Qt::PenCapStyle *cap*)**

Sets the brush mode to *cap*.

void PhotoViewer::setFilterControls**(filterControls **fc*)**

Sets the internal filter controls to *fc*.

[slot]void PhotoViewer::setImMask**(EnumConstants::maskTypes *filterName*, EnumConstants::threshold_or_filter*thof* = EnumConstants::FILTER)**

Sets the internal mask based on *filterName* and *thof*.

void PhotoViewer::setPanMode**(bool *mode*)**

Sets the pan mode to either PAN (*mode* = true) or PAINTBRUSH (*mode* = false).

void PhotoViewer::setPhoto**(QPixmap *pixmap*)**

Sets the internal image based on *pixmap*.

QSize PhotoViewer::setPhotoFromFile**(QString *filename*)**

Sets the internal image based on data from *filename*.

void **PhotoViewer::**setPhotoFromPixmap**(QPixmap *px*)**

Sets the internal image based on *px*.

void **PhotoViewer::**testPixmap**()**

For testing

[override virtual]void **PhotoViewer::**wheelEvent**(QWheelEvent ****event*)**

Reimplemented from QGraphicsView::wheelEvent().

[signal]void **PhotoViewer::**zoomed**(**int ***i*,** float ***a*,** **QPointF** ***point*)**

This signal is emitted when PhotoViewer zooms, indicating zoom count with *i*, zoom factor, *a*, and center of zoom *point*.

[slot]void **PhotoViewer::**zoomedInADifferentView**(**int ***zoom_input*,** float ***factor*,** **QPointF** ***point*)**

Something based on *zoom_input*, *factor*, *point*.

PicPushButton Class

A PushButton. More...

Header:	#include <PicPushButton>
Inherits:	QLabel

- List of all members, including inherited members

- Obsolete members

Public Functions

	PicPushButton(QWidget **parent*)
QPixmap	**addText**(QPixmap *pixmap_input*, QString *text*)
void	**manualSelect**()
void	**resetPixmaps**(QPixmap *pixmap_input*)
void	**setMaskType**(EnumConstants::maskTypes *maskType*)
void	**updatePixmap**()

Reimplemented Public Functions

virtual void	**mousePressEvent**(QMouseEvent **event*) override
virtual void	**paintEvent**(QPaintEvent **event*) override

- 29 public functions inherited from QLabel
- 14 public functions inherited from QFrame
- 222 public functions inherited from QWidget
- 31 public functions inherited from QObject
- 15 public functions inherited from QPaintDevice

Public Slots

void	**otherBoxChecked**(EnumConstants::maskTypes *type*)

- 7 public slots inherited from QLabel
- 19 public slots inherited from QWidget

- 1 public slot inherited from QObject

Signals

void	**checked**(EnumConstants::maskTypes *type*)

- 2 signals inherited from QLabel
- 4 signals inherited from QWidget
- 2 signals inherited from QObject

Public Variables

bool	**check**
EnumConstants::maskTypes	**filter**
int	**h**
QPixmap	**pixmap**
QPixmap	**pixmap_hover**
QPixmap	**pixmap_pressed**
int	**w**

Static Public Members

const QMetaObject	**staticMetaObject**

- 5 static public members inherited from QWidget
- 9 static public members inherited from QObject
- 1 static public member inherited from QPaintDevice

Additional Inherited Members

- 12 properties inherited from QLabel
- 6 properties inherited from QFrame
- 59 properties inherited from QWidget
- 1 property inherited from QObject
- 11 protected functions inherited from QLabel
- 4 protected functions inherited from QFrame
- 35 protected functions inherited from QWidget
- 9 protected functions inherited from QObject
- 1 protected function inherited from QPaintDevice
- 1 protected slot inherited from QWidget

Detailed Description

A PushButton.

Member Function Documentation

PicPushButton:: PicPushButton(**QWidget** ***parent***)

Creates an instance with parent QWidget *parent*.

QPixmap PicPushButton::addText(**QPixmap** ***pixmap_input*****, QString** ***text***)

Returns a QPixmap based on *pixmap_input* with *text* written on it.

[signal]void **PicPushButton::**checked(**EnumConstants::maskTypes** ***type***)

This signal is emitted when something is checked indicating *type*.

void **PicPushButton::**manualSelect**()**

Something

[override virtual]void **PicPushButton::**mousePressEvent**(QMouseEvent** ***event*)**

Reimplemented from QLabel::mousePressEvent().

[slot]void **PicPushButton::**otherBoxChecked**(EnumConstants::maskTypes** ***type*)**

Slot for when another button is pressed taking *type*.

[override virtual]void **PicPushButton::**paintEvent**(QPaintEvent** ***event*)**

Reimplemented from QLabel::paintEvent().

void **PicPushButton::**resetPixmaps**(QPixmap** ***pixmap_input*)**

Resets the pixmaps based on *pixmap_input*.

void **PicPushButton::**setMaskType**(EnumConstants::maskTypes** ***maskType*)**

Sets the mask type to *maskType*.

void **PicPushButton::**updatePixmap**()**

Updates the internal pixmaps.

PolygonSelect Class

The PolygonSelect class provides a class for annotating irregular regions. More...

Header:	#include <PolygonSelect>
Inherits:	SelectItem and QGraphicsPolygonItem

- List of all members, including inherited members
- Obsolete members

Public Functions

	PolygonSelect()
	PolygonSelect(const QJsonObject &*json*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)
	PolygonSelect(QPointF *point*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)
virtual	**~PolygonSelect**() override
void	**checkPoint**(QPointF &*point*)
QPolygonF	**getPolygon**() const
bool	**isVisible**() const
qreal	**magnitude**(QPointF *point*) const
void	**movePoint**(const QPointF &*point*)
QGraphicsScene *	**scene**() const
void	**setMirrorActivePoint**(QPointF *point*) const

void	**setMirrorPolygon**(int *actVert*)
void	**setOpacity**(qreal *val*)

Reimplemented Public Functions

virtual void	**addPoint**(QPointF &*point*, const int *vertex* = UNSELECTED) override
virtual QString	**baseInstructions**() const override
virtual QRectF	**boundingRect**() const override
virtual void	**clickPoint**(const QPointF &*point*) override
virtual PolygonSelect *	**getMirror**() const override
virtual void	**insertVertex**(const int *vertex*, const QPointF &*point*) override
virtual bool	**isInside**(const QPointF &*point*) const override
virtual void	**mirrorHide**() const override
virtual void	**mirrorShow**() const override
virtual void	**moveItem**(const QPointF &*oldPos*, QPointF &*newPos*) override
virtual int	**numberOfVertices**() const override
virtual void	**paint**(QPainter **painter*, const QStyleOptionGraphicsItem **option*, QWidget **widget* = nullptr) override
virtual void	**read**(const QJsonObject &*json*) override

virtual void	**removeVertex**(int *vertex* = UNSELECTED) override
virtual void	**resetActiveVertex**() override
virtual void	**resizeItem**(const int *vertex*, QPointF &*oldP*, QPointF &*newP*) override
virtual void	**rotate**(const QPointF &*from*, const QPointF &*to*) override
virtual void	**rotateMirror**() const override
virtual void	**setMirror**(SelectItem **item*) override
virtual void	**setMirrorActive**() const override
virtual void	**setMirrorAdded**() const override
virtual void	**setMirrorMoved**() const override
virtual void	**setMirrorResized**() const override
virtual void	**setMirrorVertex**(int *vertex*) const override
virtual void	**toPixmap**(QPainter **painter*) override
virtual void	**updateMirrorScene**() const override
virtual void	**updatePen**(QPen *pen*) override
virtual void	**write**(QJsonObject &*json*) const override

- 57 public functions inherited from SelectItem
- 11 public functions inherited from QGraphicsPolygonItem
- 183 public functions inherited from QGraphicsItem
- 6 public functions inherited from QAbstractGraphicsShapeItem

- 183 public functions inherited from QGraphicsItem

Public Variables

const QString	**baseInstruction**

Protected Variables

QVector<QPointF>	**myPoints**
QVector<QRectF>	**myVertices**

Additional Inherited Members

- 1 static public member inherited from SelectItem
- 2 protected functions inherited from SelectItem
- 48 protected functions inherited from QGraphicsItem

Detailed Description

The PolygonSelect class provides a class for annotating irregular regions.

Provides a class for annotating irregular regions in InstaDam. The region is described by a set of vertices.

Member Function Documentation

PolygonSelect:: PolygonSelect**()**

Constructs a PolygonSelect object with the first vertex at 0,0 in scene coordinates

PolygonSelect:: PolygonSelect**(const QJsonObject &*json*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)**

Constructs a PolygonSelect object by reading a QJsonObject and setting the internal vertex list to the values given in *json*. *label* is the Label which owns this object and *item* is the parent QGraphicsItem, if any.

PolygonSelect:: PolygonSelect**(QPointF *point*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)**

Constructs a PolygonSelect object by setting the first vertex to *point*, *label* is the Label which owns this object and *item* is the parent QGraphicsItem, if any.

[override virtual]PolygonSelect::~PolygonSelect**()**

Destructor

[override virtual]void **PolygonSelect::**addPoint**(QPointF &*point*, const** int ***vertex* = UNSELECTED)**

Reimplemented from SelectItem::addPoint().

[override virtual]QString PolygonSelect::baseInstructions**() const**

Reimplemented from SelectItem::baseInstructions().

[override virtual]QRectF PolygonSelect::boundingRect**() const**

Reimplemented from QGraphicsItem::boundingRect().

void PolygonSelect::checkPoint(QPointF &*point*)

Checks whether *point* is inside of the QGraphicsScene which owns this PolygonSelect object. If it is outside the scene then *point* is adjusted so that it is inside.

[override virtual]void PolygonSelect::clickPoint(const QPointF &*point*)

Reimplemented from SelectItem::clickPoint().

[override virtual]PolygonSelect *PolygonSelect::getMirror() const

Reimplemented from SelectItem::getMirror().

QPolygonF PolygonSelect::getPolygon() const

Returns the PolygonSelect object's internal QPolygonF.

[override virtual]void PolygonSelect::insertVertex(const int *vertex*, const QPointF &*point*)

Reimplemented from SelectItem::insertVertex().

[override virtual]bool PolygonSelect::isInside(const QPointF &*point*) const

Reimplemented from SelectItem::isInside().

bool PolygonSelect::isVisible() const

Returns a bool indicating whether the PolygonSelect object is visible (true) in its owning QGraphicsScene or not (false).

qreal PolygonSelect::magnitude(QPointF *point*) const

Returns a qreal indicating the absolute magnitude of a vector from 0,0 to *point*.

[override virtual]void **PolygonSelect::**mirrorHide**() const**

Reimplemented from SelectItem::mirrorHide().

[override virtual]void **PolygonSelect::**mirrorShow**() const**

Reimplemented from SelectItem::mirrorShow().

[override virtual]void **PolygonSelect::**moveItem**(const QPointF &*oldPos*, QPointF &*newPos*)**

Reimplemented from SelectItem::moveItem().

void **PolygonSelect::**movePoint**(const QPointF &*point*)**

Moves the currently active vertex to *point* in scene coordinates

[override virtual]int **PolygonSelect::**numberOfVertices**() const**

Reimplemented from SelectItem::numberOfVertices().

[override virtual]void **PolygonSelect::**paint**(QPainter *_painter_, const QStyleOptionGraphicsItem *_option_, QWidget *_widget_ = nullptr)**

Reimplemented from QGraphicsItem::paint().

[override virtual]void **PolygonSelect::**read**(const QJsonObject &*json*)**

Reimplemented from SelectItem::read().

[override virtual]void **PolygonSelect::**removeVertex**(**int ***vertex* = UNSELECTED)**

[override virtual]void **PolygonSelect::**resetActiveVertex**()**

Reimplemented from SelectItem::resetActiveVertex().

[override virtual]void **PolygonSelect::**resizeItem**(const** int ***vertex*****,** QPointF **&*oldP*****,** QPointF **&*newP*****)**

Reimplemented from SelectItem::resizeItem().

[override virtual]void **PolygonSelect::**rotate**(const** QPointF **&*from*****, const** QPointF **&*to*****)**

Reimplemented from SelectItem::rotate().

Empty function as a PolygonSelect cannot be rotated.

[override virtual]void **PolygonSelect::**rotateMirror**() const**

Reimplemented from SelectItem::rotateMirror().

Empty funtion as a PolygonSelect cannot be rotated.

QGraphicsScene ***PolygonSelect::**scene**() const**

This function overloads SelectItem::scene().

Returns the QGraphicsScene to which this item belongs.

[override virtual]void **PolygonSelect::**setMirror**(SelectItem ****item*****)**

Reimplemented from SelectItem::setMirror().

[override virtual]void **PolygonSelect::**setMirrorActive**() const**

Reimplemented from SelectItem::setMirrorActive().

void **PolygonSelect::**setMirrorActivePoint**(**QPointF ***point*****) const**

Sets the PolygonSelect object's active vertex to *point*.

[override virtual]void **PolygonSelect::**setMirrorAdded**() const**

Reimplemented from SelectItem::setMirrorAdded().

[override virtual]void **PolygonSelect::**setMirrorMoved**() const**

Reimplemented from SelectItem::setMirrorMoved().

void **PolygonSelect::**setMirrorPolygon**(**int ***actVert*****)**

Updates the mirror object and sets *actVert* as the active vertex.

[override virtual]void **PolygonSelect::**setMirrorResized**() const**

Reimplemented from SelectItem::setMirrorResized().

[override virtual]void **PolygonSelect::**setMirrorVertex**(**int ***vertex*****) const**

Reimplemented from SelectItem::setMirrorVertex().

void **PolygonSelect::**setOpacity**(**qreal ***val*****)**

* Sets the opacity of the label to *val*

[override virtual]void **PolygonSelect::**toPixmap**(**QPainter ****painter*****)**

Reimplemented from SelectItem::toPixmap().

[override virtual]void **PolygonSelect::**updateMirrorScene**() const**

Reimplemented from SelectItem::updateMirrorScene().

[override virtual]void **PolygonSelect::**updatePen**(QPen *pen*)**

Reimplemented from SelectItem::updatePen().

[override virtual]void **PolygonSelect::**write**(QJsonObject &*json*) const**

Reimplemented from SelectItem::write().

Project Class

Holds project related information. More...

Header:	#include <Project>

- List of all members, including inherited members

Public Functions

	Project()
	~Project()
void	**addLabel**(QSharedPointer<Label> *lb*)
void	**clearAllLabels**()
int	**getId**()

int	**getImageId**()
QSharedPointer<Label>	**getLabel**(int *index*) const
QVector<QSharedPointer<Label> >	**getLabels**() const
QString	**getName**()
int	**numLabels**() const
void	**resetLabels**()
void	**setId**(int *id*)
void	**setImageId**(int *id*)
void	**setLabel**(int *index*, QSharedPointer<Label> *lb*)
void	**setLabels**(QVector<QSharedPointer<Label> > *lab*)
void	**setName**(QString *name*)

Detailed Description

Holds project related information.

Member Function Documentation

Project:: Project**()**

Creates an instance of Project.

Project::~Project**()**

Destructor.

void **Project::**addLabel**(**QSharedPointer**<**Label**>** ***lb*****)**

Adds Label *lb* to the internal list.

void **Project::**clearAllLabels**()**

Clear all the labels in the Project.

int **Project::**getId**()**

Returns the id of this instance.

int **Project::**getImageId**()**

Returns the image_id of the Project.

QSharedPointer**<**Label**> Project::**getLabel**(**int ***index*****) const**

Returns the Label at index *index*.

QVector**<**QSharedPointer**<**Label**> > Project::**getLabels**() const**

Returns all the Labels in the Project as a QVector.

QString **Project::**getName**()**

Returns the name of the Project.

int **Project::**numLabels**() const**

Returns the current number of Labels in the internal list.

void **Project::**resetLabels**()**

Clears the internal list of Labels.

void **Project::**setId**(**int ***id*****)**

Sets the id of this instance to *id*.

void **Project::**setImageId**(**int ***id*****)**

Set the image_id of the Project to *id*.

void **Project::**setLabel**(**int ***index*****, QSharedPointer<Label> *lb*)**

Sets the label at indes *index* to *lb*.

void **Project::**setLabels**(QVector<QSharedPointer<Label> > *lab*)**

Set all of the labels in the Project with the given *lab*.

void **Project::**setName**(QString *name*)**

Set the name of the Project to *name*.

ProjectList Class

A list of projects. More...

Header:	#include <ProjectList>
Inherits:	QWidget

- List of all members, including inherited members
- Obsolete members

Public Functions

	ProjectList(QWidget **parent* = nullptr)
virtual	**~ProjectList**()
void	**addItems**(QJsonDocument *obj*, QString *databaseURL*, QString *accessToken*)

- 222 public functions inherited from QWidget
- 31 public functions inherited from QObject
- 15 public functions inherited from QPaintDevice

Public Slots

void	**confirmProjectDeletion**(QListWidgetItem **project_name*)
void	**deleteProject**(QListWidgetItem **project_name*)
void	**openProject**(QListWidgetItem **project_name*)

- 19 public slots inherited from QWidget
- 1 public slot inherited from QObject

Signals

void	**instadamClearAll**()
void	**projectDeleted**(int *id*)
void	**projectIdChanged**(int *id*)
void	**projectJsonReceived**(QJsonObject *json*)

- 4 signals inherited from QWidget
- 2 signals inherited from QObject

Public Variables

QString	**useCase**

Static Public Members

const QMetaObject	**staticMetaObject**

- 5 static public members inherited from QWidget
- 9 static public members inherited from QObject
- 1 static public member inherited from QPaintDevice

Additional Inherited Members

- 59 properties inherited from QWidget
- 1 property inherited from QObject
- 35 protected functions inherited from QWidget
- 9 protected functions inherited from QObject
- 1 protected function inherited from QPaintDevice
- 1 protected slot inherited from QWidget

Detailed Description

A list of projects.

Member Function Documentation

ProjectList::ProjectList(QWidget **parent* = nullptr)

Creates a ProjectList with parent QWidget *parent*, if any.

[virtual]ProjectList::~ProjectList()

Destructor.

void ProjectList::addItems(QJsonDocument *obj*, QString *databaseURL*, QString *accessToken*)

Adds Projects to this object based on the input *obj*, *databaseURL*, and *accessToken*.

[slot]void ProjectList::confirmProjectDeletion(QListWidgetItem **project_name*)

Starts a widget to confirm project deletion of *project_name*.

[slot]void ProjectList::deleteProject(QListWidgetItem **project_name*)

Sends a project deletion request for *project_name*.

[signal]void ProjectList::instadamClearAll()

This signal is emitted when a project list is cleared.

[slot]void **ProjectList::**openProject**(QListWidgetItem** ***project_name)**

Slot triggered to open the Project based on *project_name*.

[signal]void **ProjectList::**projectDeleted**(**int ***id*)**

Signal sent when a project, *id*, is deleted.

[signal]void **ProjectList::**projectIdChanged**(**int ***id*)**

This signal is emitted when a project id is changed to *id*.

[signal]void **ProjectList::**projectJsonReceived**(QJsonObject** ***json*)**

This signal is emitted when a new project Json object, *json*, is received.

RectangleSelect Class

The RectangleSelect class provides a class for annotating rectangular regions. More...

Header:	#include <RectangleSelect>
Inherits:	BoxBasedSelector and QGraphicsRectItem

- List of all members, including inherited members
- Obsolete members

Public Functions

	RectangleSelect()
	RectangleSelect(const QJsonObject &_json_, QSharedPointer<Label> _label_ = nullptr, QGraphicsItem *_item_ = nullptr)
	RectangleSelect(QPointF _point_, QSharedPointer<Label> _label_ = nullptr, QGraphicsItem *_item_ = nullptr)
	RectangleSelect(QPointF _point_, qreal _vertSize_, QSharedPointer<Label> _label_ = nullptr, QGraphicsItem *_item_ = nullptr)
virtual	**~RectangleSelect**() override
bool	**isVisible**()
QGraphicsScene *	**scene**()
void	**setOpacity**(qreal _val_)

Reimplemented Public Functions

virtual void	**addPoint**(QPointF &_point_, const int _vertex_ = UNSELECTED) override
virtual QString	**baseInstructions**() const override
virtual QRectF	**boundingRect**() const override
virtual RectangleSelect *	**getMirror**() const override
virtual bool	**isInside**(const QPointF &_point_) const override
virtual void	**mirrorHide**() const override

virtual void	**mirrorShow**() const override
virtual void	**moveItem**(const QPointF &*oldPos*, QPointF &*newPos*) override
virtual void	**paint**(QPainter **painter*, const QStyleOptionGraphicsItem **option*, QWidget **widget* = nullptr) override
virtual void	**rotateMirror**() const override
virtual void	**setMirror**(SelectItem **item*) override
virtual void	**setMirrorActive**() const override
virtual void	**setMirrorAdded**() const override
virtual void	**setMirrorCorners**(QRectF *tlc*, QRectF *blc*, QRectF *trc*, QRectF *brc*) const override
virtual void	**setMirrorMoved**() const override
virtual void	**setMirrorResized**() const override
virtual void	**setMirrorVertex**(int *vertex*) const override
virtual void	**setRectUnchecked**(QRectF *rect*) override
virtual void	**toPixmap**(QPainter **painter*) override
virtual int	**type**() const override
virtual void	**updateMirrorScene**() const override
virtual void	**updatePen**(QPen *pen*) override

- 14 public functions inherited from BoxBasedSelector

- 10 public functions inherited from QGraphicsRectItem
- 57 public functions inherited from SelectItem
- 6 public functions inherited from QAbstractGraphicsShapeItem
- 366 public functions inherited from QGraphicsItem

Public Variables

const QString	**baseInstruction**

Additional Inherited Members

- 1 static public member inherited from SelectItem
- 1 protected function inherited from BoxBasedSelector
- 2 protected functions inherited from SelectItem
- 48 protected functions inherited from QGraphicsItem

Detailed Description

The RectangleSelect class provides a class for annotating rectangular regions.

Provides a class for annotating rectangular regions in InstaDam. The region is described by its top-left and lower-right vertices, and by its angle of rotation.

Member Function Documentation

RectangleSelect:: RectangleSelect**()**

Constructs a RectangleSelect object with all vertices at 0,0 in scene coordinates

RectangleSelect:: RectangleSelect**(const QJsonObject &*json*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)**

Constructs a RectangleSelect object by reading a QJsonObject and setting the internal rectangle and rotation angle to the values given in *json*. *label* is the Label which owns this object and *item* is the parent QGraphicsItem, if any.

RectangleSelect:: RectangleSelect**(QPointF *point*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem**item* = nullptr)**

Constructs a RectangleSelect object by setting all vertices to be a *point*, *label* is the Label which owns this object and *item* is the parent QGraphicsItem, if any.

RectangleSelect:: RectangleSelect**(QPointF *point*, qreal *vertSize*, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)**

Constructs a RectangleSelect object by setting all vertices to be a *point*, *vertSize* indicates the size of the vertex highlight boxes, *label* is the Label which owns this object and *item* is the parent QGraphicsItem, if any.

[override virtual]RectangleSelect::~RectangleSelect**()**

Destructor

[override virtual]void **RectangleSelect::**addPoint**(QPointF &*point*, const** int ***vertex* = UNSELECTED)**

Reimplemented from SelectItem::addPoint().

[override virtual]QString RectangleSelect::baseInstructions**() const**

Reimplemented from SelectItem::baseInstructions().

[override virtual]QRectF RectangleSelect::boundingRect**() const**

Reimplemented from QGraphicsItem::boundingRect().

[override virtual]RectangleSelect *RectangleSelect::getMirror**() const**

Reimplemented from SelectItem::getMirror().

[override virtual]bool **RectangleSelect::**isInside**(const QPointF &*point*) const**

Reimplemented from SelectItem::isInside().

bool **RectangleSelect::**isVisible**()**

Returns whether the RectangleSelect object is visible (true) on the QGraphicsScene or not (false).

[override virtual]void **RectangleSelect::**mirrorHide**() const**

Reimplemented from SelectItem::mirrorHide().

[override virtual]void **RectangleSelect::**mirrorShow**() const**

Reimplemented from SelectItem::mirrorShow().

[override virtual]void **RectangleSelect::**moveItem**(const QPointF &*oldPos*, QPointF &*newPos*)**

Reimplemented from SelectItem::moveItem().

[override virtual]void **RectangleSelect::**paint**(QPainter *****painter*, const QStyleOptionGraphicsItem *****option*, QWidget *****widget* = nullptr)**

Reimplemented from QGraphicsItem::paint().

[override virtual]void **RectangleSelect::**rotateMirror**() const**

Reimplemented from SelectItem::rotateMirror().

QGraphicsScene *RectangleSelect::scene**()**

This function overloads SelectItem::scene().

Returns the QGraphicsScene to which this item belongs.

[override virtual]void **RectangleSelect::**setMirror**(SelectItem ****item***)**

Reimplemented from SelectItem::setMirror().

[override virtual]void **RectangleSelect::**setMirrorActive**() const**

Reimplemented from SelectItem::setMirrorActive().

[override virtual]void **RectangleSelect::**setMirrorAdded**() const**

Reimplemented from SelectItem::setMirrorAdded().

[override virtual]void **RectangleSelect::**setMirrorCorners**(QRectF *tlc*, QRectF *blc*, QRectF *trc*, QRectF *brc*) const**

Reimplemented from BoxBasedSelector::setMirrorCorners().

[override virtual]void **RectangleSelect::**setMirrorMoved**() const**

Reimplemented from SelectItem::setMirrorMoved().

[override virtual]void **RectangleSelect::**setMirrorResized**() const**

Reimplemented from SelectItem::setMirrorResized().

[override virtual]void **RectangleSelect::**setMirrorVertex**(**int ***vertex*****) const**

Reimplemented from SelectItem::setMirrorVertex().

void **RectangleSelect::**setOpacity**(qreal *val*)**

Sets the opacity of the label to *val*

[override virtual]void **RectangleSelect::**setRectUnchecked**(QRectF *rect*)**

Reimplemented from BoxBasedSelector::setRectUnchecked().

[override virtual]void **RectangleSelect::**toPixmap**(QPainter ****painter***)**

Reimplemented from SelectItem::toPixmap().

[override virtual]int **RectangleSelect::**type**() const**

Reimplemented from SelectItem::type().

This function overloads SelectItem::type().

This function returns the type of this item.

See also QGraphicsItem::type() and SelectItem::type().

[override virtual]void **RectangleSelect::**updateMirrorScene**() const**

Reimplemented from SelectItem::updateMirrorScene().

[override virtual]void **RectangleSelect::**updatePen**(QPen *pen*)**

Reimplemented from SelectItem::updatePen().

Register Class

Registers a user. More...

Header:	#include <Register>
Inherits:	QWidget

- List of all members, including inherited members
- Obsolete members

Public Functions

	Register(QWidget **parent* = nullptr)
virtual	**~Register**()

- 222 public functions inherited from QWidget
- 31 public functions inherited from QObject
- 15 public functions inherited from QPaintDevice

Static Public Members

const QMetaObject	**staticMetaObject**

- 5 static public members inherited from QWidget
- 9 static public members inherited from QObject
- 1 static public member inherited from QPaintDevice

Additional Inherited Members

- 59 properties inherited from QWidget
- 1 property inherited from QObject

- 19 public slots inherited from QWidget
- 1 public slot inherited from QObject
- 4 signals inherited from QWidget
- 2 signals inherited from QObject
- 35 protected functions inherited from QWidget
- 9 protected functions inherited from QObject
- 1 protected function inherited from QPaintDevice
- 1 protected slot inherited from QWidget

Detailed Description

Registers a user.

Member Function Documentation

Register:: Register**(QWidget ****parent*** **= nullptr)**

Creates a Register instance with parent QWidget *parent*.

[virtual]Register::~Register**()**

Destructor

RotateCommand Class

The RotateCommand class provides undo and redo actions for rotating a SelectItem on a PhotoScene. More...

Header:	#include <RotateCommand>

Inherits:	QUndoCommand

- List of all members, including inherited members

Public Functions

RotateCommand(SelectItem **item*, const QPointF *oldPos*, const QPointF *newPos*, QUndoCommand **parent* = nullptr)

Reimplemented Public Functions

virtual void	**redo**() override
virtual void	**undo**() override

- 11 public functions inherited from QUndoCommand

Detailed Description

The RotateCommand class provides undo and redo actions for rotating a SelectItem on a PhotoScene.

RotateCommand is used to hold a reference to a SelectItem and the starting and ending points of rotation. Using the undo() and redo() commands the rotation of the item can be undone and redone repeatedly.

See also AddCommand, MoveCommand, AddVertexCommand, MoveVertexCommand, DeleteVertexCommand, ErasePointsCommand, DeleteCommand, and EditLabelCommand.

Member Function Documentation

RotateCommand::RotateCommand(SelectItem **item*, const QPointF *oldPos*, const QPointF *newPos*, QUndoCommand **parent* = nullptr)

Constructs a RotateCommand using the provided *item* and the starting (*oldPos*) and ending (*newPos*) points of rotation so that the rotation of this *item* can be undone and/or redone. *parent* refers to parent QUndoCommand, if any.

[override virtual]void RotateCommand::redo()

Reimplemented from QUndoCommand::redo().

[override virtual]void RotateCommand::undo()

Reimplemented from QUndoCommand::undo().

SelectItem Class

The SelectItem class provides a base class for all annotating items. More...

Header:	#include <SelectItem>
Inherits:	QGraphicsItem
Inherited By:	BoxBasedSelector, FreeDrawSelect, and PolygonSelect

- List of all members, including inherited members
- Obsolete members

Public Types

enum	**SelectType** { Rectangle, Ellipse, Polygon, Freedraw, Freeerase }
enum	**Vertex** { NONE, TOP, BOTTOM, LEFT, RIGHT, UNSELECTED }

Public Functions

	SelectItem(qreal *vertSize* = DEFAULT_SIZE, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)
	SelectItem(QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)
virtual	**~SelectItem**() override
virtual void	**addPoint**(QPointF &*point*, const int *vertex* = ...) = 0
virtual QString	**baseInstructions**() const = 0
virtual void	**clickPoint**(const QPointF &*point*) = 0
void	**flipH**()
void	**flipV**()
QPointF	**getActivePoint**() const
int	**getActiveVertex**() const
bool	**getFromMaskScene**() const

QSharedPointer<Label>	**getLabel**() const
virtual SelectItem *	**getMirror**() const = 0
bool	**getOnMaskScene**() const
QGraphicsItem *	**getParentItem**() const
void	**hideMask**()
virtual void	**insertVertex**(const int *vertex*, const QPointF &*point*) = 0
void	**invertColorForPen**()
virtual bool	**isInside**(const QPointF &*point*) const = 0
bool	**isItemActive**() const
bool	**isItemAdded**() const
void	**itemWasAdded**()
virtual void	**mirrorHide**() const = 0
virtual void	**mirrorShow**() const = 0
virtual void	**moveItem**(const QPointF &*oldPos*, QPointF &*newPos*) = 0
virtual int	**numberOfVertices**() const = 0
virtual void	**read**(const QJsonObject &*json*) = 0
virtual void	**removeVertex**(const int *vertex* = ...) = 0

virtual void	**resetActiveVertex**() = 0
void	**resetState**()
virtual void	**resizeItem**(const int _vertex_, QPointF &_oldP_, QPointF &_newP_) = 0
virtual void	**rotate**(const QPointF &_from_, const QPointF &_to_) = 0
virtual void	**rotateMirror**() const = 0
QGraphicsScene *	**scene**() const
void	**setActiveVertex**(int _h_, int _v_ = ...)
void	**setFromMaskScene**(bool _value_)
void	**setInactive**()
void	**setItemActive**()
void	**setLabel**(QSharedPointer<Label> _label_, bool _init_ = false)
virtual void	**setMirror**(SelectItem *_item_) = 0
virtual void	**setMirrorActive**() const = 0
virtual void	**setMirrorAdded**() const = 0
virtual void	**setMirrorMoved**() const = 0
virtual void	**setMirrorResized**() const = 0
virtual void	**setMirrorVertex**(int _vertex_) const = 0

void	**setMoved**(bool *val*)
void	**setOnMaskScene**(bool *value*)
void	**setOpacity**(qreal *val*)
void	**setRotated**(bool *val*)
void	**showMask**()
void	**sortCorners**(QRectF &*rect*, QPointF &*newPoint*)
virtual void	**toPixmap**(QPainter **painter*) = 0
virtual void	**updateMirrorScene**() const = 0
virtual void	**updatePen**(QPen *pen*) = 0
bool	**wasMoved**() const
bool	**wasPointAdded**() const
bool	**wasResized**() const
bool	**wasRotated**() const
virtual void	**write**(QJsonObject &*json*) const = 0

Reimplemented Public Functions

virtual int	**type**() const override

- 183 public functions inherited from QGraphicsItem

Public Variables

QPen	**highlightPen**
int	**myID**
QPen	**myPen**

Static Public Members

int	**ID**
QSize	**myBounds**
void	**setVertexSize**(qreal *size*)
qreal	**vertexSize**
QPointF	**xoffset**
QPointF	**yoffset**

Protected Functions

void	**checkBoundaries**(const QPointF &*shift*, QRectF &*rect*)
bool	**isInsideRect**(const QRectF &*rect*, const QPointF &*point*) const

- 24 protected functions inherited from QGraphicsItem

Protected Variables

bool	**active**
int	**activeH**
QPointF	**activePoint**
int	**activeV**
int	**activeVertex**
bool	**fromMaskScene**
bool	**hasBeenAdded**
bool	**moved**
QSharedPointer<Label>	**myLabel**
QRectF	**myRect**
int	**mytype**
bool	**onMaskScene**
QPen	**pen**
FreeDrawSelect *	**pixmap**
bool	**pointAdded**
bool	**resized**
bool	**rotated**

SelectItem::SelectType	**selectType**
QPointF	**selectedPoint**

Detailed Description

The SelectItem class provides a base class for all annotating items.

Provides a foundation for all annotation type objects. Common funtions that do not depend on the type of object are implemented in this class, but every subclass of this class must implement all virtual functions.

Current subclasses are:

- RectangleSelect provides selection fo rectanular areas
- EllipseSelect provides selection of elliptical regions
- PolygonSelect provides selection of irregularly shaped regions
- FreeDrawSelect provides selection of regions by brush stroke
- FreeDrawErase is used to erase sections of FreeDrawSelect regions

SelectItem subclasses must have a Label and a PhotoScene owning them.

Member Type Documentation

enum SelectItem:: SelectType

This enum is used to define the type of the SelectItem:

Constant	Value	Description
SelectItem::Rectangle	51	A RectangleSelect object
SelectItem::Ellipse	52	An EllipseSelect object
SelectItem::Polygon	53	A PolygonSelect object
SelectItem::Freedraw	54	A FreeDraeSelect object
SelectItem::Freeerase	55	A FreeEraseSelect object

enum SelectItem:: Vertex

This enum is used to set the active corner of select regions. A vertical and a horizontal selection are combined to indicate the specific corner

```
SelectItem::TOP | SelectItem::LEFT
```

indicates the top-left corner.

Constant	Value	Description
SelectItem::NONE	0x0	Indicates no side of the object
SelectItem::TOP	0x1	Indicates the top of the object
SelectItem::BOTTOM	0x2	Indicates the bottom of the object
SelectItem::LEFT	0x4	Indicates the left of the object
SelectItem::RIGHT	0x8	Indicates the right of the object

SelectItem::UNSELECTED	INT_MAX	Indicates that no vertex has been selected

Member Function Documentation

SelectItem::SelectItem(qreal *vertSize* = DEFAULT_SIZE, QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)

Constructs a SelectItem, using *vertSize* as the vertex box size, *label* as the Label that owns this instance, and *item* as the parent QGraphicsItem.

SelectItem::SelectItem(QSharedPointer<Label> *label* = nullptr, QGraphicsItem **item* = nullptr)

Constructs a SelectItem, using *label* as the Label that owns this instance, and *item* as the parent QGraphicsItem.The vertex box size defaults to 10 pixels.

[override virtual]SelectItem::~SelectItem()

Destroys the instance of SelectItem. The destructor is virtual.

[pure virtual]void SelectItem::addPoint(QPointF &*point*, const int *vertex* = ...)

This pure virtual function adds a point to object being drawn. It is up to the reimplimentation of this function to determine how to handle the input *point* and *vertex*, as not all SelectItem subclasses have vertices.

See also removeVertex() and insertVertex().

[pure virtual]QString SelectItem::baseInstructions() const

Returns a QString representing the instructions on how to draw this SelectItem type on the scene. It is up to the individual reimplementations to construct the string.

[protected]void SelectItem::checkBoundaries(const QPointF &*shift*, QRectF &*rect*)

Checks the boundaries of the object being moved so that it is constrained to the canvas. *rect* is the bounding rectangle of the object and *shift* is the distance the rectangle is being moved. If the movement (*rect* + *shift*) would place one or more of the corners outside of the canvas the movement is halted at the last pixel before the vertex would disappear.

[pure virtual]void **SelectItem::**clickPoint**(const** QPointF **&*point*)**

This pure virtual function is used to signal an object that the mouse has been clicked within its boundaries. It is up to the individual reimplementations to determine what to do with the information. *point* indicated the coordinates (on the canvas) where the click took place.

void **SelectItem::**flipH**()**

Convenience function which flips the horizontal component of the activeVertex (e.g. if the horizontal component of the activeVertex is SelectItem::TOP it becomes SelectItem::BOTTOM and vice versa).

See also setActiveVertex() and flipV().

void **SelectItem::**flipV**()**

Convenience function which flips the vertical component of the activeVertex (e.g. if the vertical component of the activeVertex is SelectItem::LEFT it becomes SelectItem::RIGHT and vice versa).

See also setActiveVertex() and flipH().

QPointF **SelectItem::**getActivePoint**() const**

Returns a QPointF which contains the QGraphicsScene coordinates of the currently active vertex.

int **SelectItem::**getActiveVertex**() const**

Returns an int indicating the currently active vertex. Returns SelectItem::UNSELECTED if there is no currently active vertex.

See also setActiveVertex() and getActivePoint().

bool **SelectItem::**getFromMaskScene**() const**

Returns whether this item was generated by the mask scene

See also setFromMaskScene().

QSharedPointer**<**Label**> SelectItem::**getLabel**() const**

Returns a Label*, pointing to the Label that owns the SelectItem.

See also setLabel().

[pure virtual]SelectItem ***SelectItem::**getMirror**() const**

Returns a SelectItem* to the mirror of this SelectItem. If this SelectItem is a member of the photoScene then it will return the corresponding member from the maskScene.

bool **SelectItem::**getOnMaskScene**() const**

Returns the value of onMaskScene.

See also setFromMaskScene().

QGraphicsItem ***SelectItem::**getParentItem**() const**

Returns the QGraphicItem which is the parent to this item.

void **SelectItem::**hideMask**()**

Hides any items on the mask scene that were generated on the photo scene.

See also showMask().

[pure virtual]void **SelectItem::**insertVertex**(const** int ***vertex*****, const** QPointF **&*****point*****)**

This pure virtual function inserts a new vertex, given as *point*, at position *vertex* in the internal list of vertices. It is up to the reimplementation of this function to determine if and how this is done.

See also addPoint() and removeVertex().

void **SelectItem::**invertColorForPen**()**

Determines what color is the "opposite" of the current pen color. This is used to find a color for highlighting vertices.

[pure virtual]bool **SelectItem::**isInside**(const QPointF &*point*) const**

Returns a bool indicating whether *point* is inside the boundaries of the SelectItem. true indicates the QPointF is within the bounds, false otherwise. It is up to the individual reimplementations to determine how this is done.

[protected]bool **SelectItem::**isInsideRect**(const QRectF &*rect*, const QPointF &*point*) const**

Determines whether *point* is inside of *rect*. Returns true if it is.

bool **SelectItem::**isItemActive**() const**

Returns a bool indicating whether the SelectItem is active (true) or not (false).

See also setItemActive(), setInactive(), and resetState().

bool **SelectItem::**isItemAdded**() const**

Returns a bool indicating whether the SelectItem has been added (true) to a QGraphicsScene or not (false).

See also itemWasAdded().

void **SelectItem::**itemWasAdded**()**

Sets the hasBeenAdded attribute to be true for the SelectItem and its mirror.

See also isItemAdded().

[pure virtual]void **SelectItem::**mirrorHide**() const**

Pure virtual function to hide the mirror item on its scene.

See also QGraphicsItem::hide().

[pure virtual]void **SelectItem::**mirrorShow**() const**

Pure virtual function to show the mirror item on its scene.

See also QGraphicsItem::show().

[pure virtual]void **SelectItem::**moveItem**(const QPointF &*oldPos*, QPointF &*newPos*)**

This pure virtual function moves the SelectItem on the canvas. *oldPos* indicates the position of the mouse at the begining of the move and *newPos* indicates the mouse position at the end of the move. It is up to the individual reimplementations to determine how this movement is interpreted and implemented.

See also rotate() and resizeItem().

[pure virtual]int **SelectItem::**numberOfVertices**() const**

Returns an int indicating the number of vertices in the SelectItem. It is up to the individual reimplementations to do the calculation.

[pure virtual]void **SelectItem::**read**(const QJsonObject &*json*)**

This pure virtual function determines how QJsonObject *json* is converted into the appropriate SelectItem subclass. It is up to the individual reimplementations to determine what each QJsonObject attribute translates to. Any changes to this function must be mirrored in the write() function or data corruption can occur.

See also write().

[pure virtual]void SelectItem::removeVertex(const int *vertex* = ...)

This pure virtual function removes vertex *vertex* from the internal list of vertices. It is up to the individual reimplementations to determine if and how this is done.

See also addPoint() and insertVertex().

[pure virtual]void SelectItem::resetActiveVertex()

This pure virtual function is used to reset the active vertex to the default value of SelectItem::UNSELECTED.

void SelectItem::resetState()

Resets the state of the SelectItem, specifically sets the moved, resized, pointAdded, and rotated attributes to false (along with the mirror counterparts), and sets the activeVertex to SelectItem::UNSELECTED.

See also wasMoved(), wasPointAdded(), wasResized(), wasRotated(), setActiveVertex(), getActiveVertex(), setInactive(), and setItemActive().

[pure virtual]void SelectItem::resizeItem(const int *vertex*, QPointF &*oldP*, QPointF &*newP*)

This pure virtual function is used to resize the item. *vertex* indicates which vertex is moving from point *oldP* to point *newP*. It is up to the individual reimplementations to determine how to do this for each derived class.

See also moveItem() and rotate().

[pure virtual]void SelectItem::rotate(const QPointF &*from*, const QPointF &*to*)

This pure virtual funcion is used to rotate a SelectItem. *from* indicates the starting position of the rotation and *to* indicates the end of the rotation in scene coordinates. It is up to the individual reimplementations to calculate the center of rotation and angle of rotation the SelectItem is undergoing.

See also moveItem() and resizeItem().

[pure virtual]void SelectItem::rotateMirror() const

Pure virtual function to indicate rotation to the mirror SelectItem.

QGraphicsScene *SelectItem::scene() const

Returns the QGraphicsScene to which this item belongs.

void SelectItem::setActiveVertex(int *h*, int *v* = ...)

Sets the activeVertex (the one currently being manipulated) to the given value. For rectangular based objects (RectangleSelect and EllipseSelect) *h* and *v* can be used to set the horizontal and vertical components seperately or they can be logically combined beforehand:

```
setActiveVertex(SelectItem::TOP, SelectItem::Bottom);
```

```
  setActiveVertex(SelectItem::TOP | SelectItem::Bottom);
```

For other subclasses of SelectItem, just the vertex index is needed:

```
  setActiveVertex(5);
```

```
  setActiveVertex(SelectItem::UNSELECTED);
```

indicates that there is no currently active vertex.

See also getActiveVertex() and resetState().

void SelectItem::setFromMaskScene(bool *value*)

Sets the fromMaskScene variable to *value*, indicating whether (true) or not (false) the item was generated from the mask scene.

See also getOnMaskScene().

void **SelectItem::**setInactive**()**

Sets the SelectItem to be inactive (i.e. not the one currently being manipulated in the scene).

See also resetState() and setItemActive().

void **SelectItem::**setItemActive**()**

Sets the SelectItem to be active (i.e. the one currently being manipulated in the scene).

/sa resetState(), setInactive()

See also isItemActive().

void **SelectItem::**setLabel**(QSharedPointer<Label> *label*,** bool ***init* = false)**

Sets the internal variables to associate this object with the given Label *label*. *init* is used to indicate whether this is an initialization of the base class or of a derived class.

[pure virtual]void **SelectItem::**setMirror**(SelectItem *_item_)**

Pure virtual function which sets the mirror for this object, as given by *item*. See the mirror documentation for a description of mirrors.

[pure virtual]void **SelectItem::**setMirrorActive**() const**

Pure virtual function which sets the mirror SelectItem to be active.

[pure virtual]void **SelectItem::**setMirrorAdded**() const**

Pure virtual function which adds the mirror SelectItem to its corresponding scene.

[pure virtual]void **SelectItem::**setMirrorMoved**() const**

Pure virtual function which sets the moved attribute of the SelectItem's mirror.

[pure virtual]void SelectItem::setMirrorResized() const

Pure virtual function which sets the resized attribute of the SelectItem's mirror.

[pure virtual]void SelectItem::setMirrorVertex(int *vertex*) const

Pure virtual function which sets the active vertex of the SelectItem's mirror to *vertex*.

void SelectItem::setMoved(bool *val*)

Set the internal moved variable to *val*, indicating whether (true) or not (false) the item was moved.

See also wasMoved().

void SelectItem::setOnMaskScene(bool *value*)

Sets the onMaskScene to *value*, indicating whether (true) or not (false) the item is on the mask scene.

See also getFromMaskScene().

void SelectItem::setOpacity(qreal *val*)

Set the visual opacity of the item to *val*.

void SelectItem::setRotated(bool *val*)

Set the internal rotated variable to *val*, indicating whether (true) or not (false) the item was rotated.

See also wasRotated().

[static]void SelectItem::setVertexSize(qreal *size*)

Determines the corners of the vertex boxes for plotting by using *size*.

void **SelectItem::**showMask**()**

Shows any items on the mask scene that were generated on the photo scene.

See also hideMask().

void **SelectItem::**sortCorners**(QRectF &*rect*, QPointF &*newPoint*)**

Determines the bounding corners of a rectangle that is being resized. This is done by using the activeVertex member variable (indicating the currently selected vertex) along with the given QRect *rect* and QPointF *newPoint* to determine the position of the moving vertex as well as the new top left and bottom right corners of the rectangle.

[pure virtual]void **SelectItem::**toPixmap**(QPainter ****painter***)**

Rasterizes the SelectItem and draws it on the given *painter*.

[override virtual]int **SelectItem::**type**() const**

Reimplemented from QGraphicsItem::type().

This function overloads QGraphicsItem::type() const.

This function returns the type of this item.

See also QGraphicsItem::type().

[pure virtual]void **SelectItem::**updateMirrorScene**() const**

Pure virtual function which requests an update of the QGraphicsScene which owns the mirror.

[pure virtual]void **SelectItem::**updatePen**(QPen *pen*)**

This pure virtual function is used to update the QGraphicsItem's pen with *pen*.

bool **SelectItem::**wasMoved**() const**

Returns a bool indicating whether the SelectItem was moved (true) or not (false).

See also resetState(), wasPointAdded(), wasResized(), and wasRotated().

bool **SelectItem::**wasPointAdded**() const**

Returns a bool indicating whether a point was added (true) or not (false) to the SelectItem.

See also resetState(), wasMoved(), wasResized(), and wasRotated().

bool **SelectItem::**wasResized**() const**

Returns a bool indicating whether the SelectItem was resized (true) or not (false).

See also resetState(), wasMoved(), wasPointAdded(), and wasRotated().

bool **SelectItem::**wasRotated**() const**

Returns a bool indicating whether the SelectItem was rotates (true) or not (false).

See also resetState(), wasMoved(), wasPointAdded(), and wasResized().

[pure virtual]void **SelectItem::**write**(**QJsonObject **&*json*) const**

This pure virtual function determines the SelectItem is converted into a QJsonObject representation *json*. It is up to the individual reimplementations to determine what attributes of the derived classes need to be saved in order to completely reconstruct the state of the current SelectItem. Any changes to this function must be mirrored in the read() function or data corruption can occur.

See also read().

StartingWidget Class

Starting widget for selecting user mode, either Local or Server. More...

Header:	#include <StartingWidget>
Inherits:	QWidget

- List of all members, including inherited members
- Obsolete members

Public Functions

	StartingWidget(QWidget **parent* = nullptr)
virtual	**~StartingWidget**()

- 222 public functions inherited from QWidget
- 31 public functions inherited from QObject
- 15 public functions inherited from QPaintDevice

Static Public Members

const QMetaObject	**staticMetaObject**

- 5 static public members inherited from QWidget
- 9 static public members inherited from QObject
- 1 static public member inherited from QPaintDevice

Additional Inherited Members

- 59 properties inherited from QWidget
- 1 property inherited from QObject
- 19 public slots inherited from QWidget
- 1 public slot inherited from QObject
- 4 signals inherited from QWidget
- 2 signals inherited from QObject
- 35 protected functions inherited from QWidget

- 9 protected functions inherited from QObject
- 1 protected function inherited from QPaintDevice
- 1 protected slot inherited from QWidget

Detailed Description

Starting widget for selecting user mode, either Local or Server.

Member Function Documentation

StartingWidget:: StartingWidget**(QWidget ****parent*** **= nullptr)**

Constructs a Qwidget with parent QWidget *parent*, if any.

[virtual]StartingWidget::~StartingWidget**()**

Destructor

UserPrivilege Class

The UserPrivilege provedes a class for setting user privileges. More...

Header:	#include <UserPrivilege>
Inherits:	QWidget

- List of all members, including inherited members
- Obsolete members

Public Functions

	UserPrivilege(QWidget **parent* = nullptr)
virtual	**~UserPrivilege**()

- 222 public functions inherited from QWidget
- 31 public functions inherited from QObject
- 15 public functions inherited from QPaintDevice

Signals

void	**on_pushButton_2_clicked**()
void	**on_pushButton_clicked**()

- 4 signals inherited from QWidget
- 2 signals inherited from QObject

Public Variables

QString	**privilege**
QString	**userDetails**

Static Public Members

const QMetaObject	**staticMetaObject**

- 5 static public members inherited from QWidget
- 9 static public members inherited from QObject
- 1 static public member inherited from QPaintDevice

Additional Inherited Members

- 59 properties inherited from QWidget
- 1 property inherited from QObject
- 19 public slots inherited from QWidget
- 1 public slot inherited from QObject
- 35 protected functions inherited from QWidget
- 9 protected functions inherited from QObject
- 1 protected function inherited from QPaintDevice
- 1 protected slot inherited from QWidget

Detailed Description

The UserPrivilege provedes a class for setting user privileges.

Member Function Documentation

UserPrivilege:: UserPrivilege**(QWidget **parent* = nullptr)**

Constructs a UserPrivilege instance with parent *parent*, if any.

[virtual]UserPrivilege::~UserPrivilege**()**

Destructor.

[signal]void **UserPrivilege::**on_pushButton_2_clicked**()**

Emitted when button is pushed.

[signal]void **UserPrivilege::**on_pushButton_clicked**()**

Emitted when button is pushed.

---

chooseLabelDialog Class

Defines a custom QDialog to pick labels from. More...

Header:	#include <chooseLabelDialog>
Inherits:	QDialog

- List of all members, including inherited members
- Obsolete members

Public Functions

chooseLabelDialog(Project **currentPro*)

Reimplemented Public Functions

virtual void	**mouseMoveEvent**(QMouseEvent **event*)
virtual void	**mousePressEvent**(QMouseEvent **event*)

- 8 public functions inherited from QDialog
- 222 public functions inherited from QWidget
- 31 public functions inherited from QObject
- 15 public functions inherited from QPaintDevice

Public Slots

void	**checkLabel**(QSharedPointer<Label> *label*)

- 5 public slots inherited from QDialog
- 19 public slots inherited from QWidget
- 1 public slot inherited from QObject

Signals

void	**labelPicked**(QSharedPointer<Label> *label*)

- 3 signals inherited from QDialog
- 4 signals inherited from QWidget
- 2 signals inherited from QObject

Public Variables

QVector<LabelButtonFilter *>	**labelButtons**
QPoint	**mpos**

Static Public Members

const QMetaObject	**staticMetaObject**

- 5 static public members inherited from QWidget
- 9 static public members inherited from QObject
- 1 static public member inherited from QPaintDevice

Additional Inherited Members

- 2 properties inherited from QDialog
- 59 properties inherited from QWidget
- 1 property inherited from QObject
- 6 protected functions inherited from QDialog
- 35 protected functions inherited from QWidget

- 9 protected functions inherited from QObject
- 1 protected function inherited from QPaintDevice
- 1 protected slot inherited from QWidget

Detailed Description

Defines a custom QDialog to pick labels from.

Member Function Documentation

chooseLabelDialog::chooseLabelDialog**(Project** ***currentPro*)**

Creates a chooseLabelDialog instance with the given *currentPro*.

[slot]void **chooseLabelDialog::**checkLabel**(QSharedPointer<Label>** ***label*)**

Called when a label button is checked, given by *label*.

[signal]void **chooseLabelDialog::**labelPicked**(QSharedPointer<Label>** ***label*)**

This signal is emitted when *label* is picked.

[virtual]void **chooseLabelDialog::**mouseMoveEvent**(QMouseEvent** ***event*)**

Reimplemented from QWidget::mouseMoveEvent().

Called when the mouse moves with the given *event*.

[virtual]void **chooseLabelDialog::**mousePressEvent**(QMouseEvent** ***event*)**

Reimplemented from QWidget::mousePressEvent().

Called when a mouse button is pressed with the given *event*.

---

fCheckBox Class

Checbox widget for filterDialog. More...

Header:	#include <fCheckBox>
Inherits:	QCheckBox

- List of all members, including inherited members
- Obsolete members

Public Functions

fCheckBox(EnumConstants::maskTypes *maskType*, int *propNums*, EnumConstants::threshold_or_filter *tf*, QWidget **parent*)

- 6 public functions inherited from QCheckBox
- 21 public functions inherited from QAbstractButton
- 222 public functions inherited from QWidget
- 31 public functions inherited from QObject
- 15 public functions inherited from QPaintDevice

Signals

void	**fStateChanged**(EnumConstants::maskTypes *selectedMask*, EnumConstants::threshold_or_filter *tf*)
void	**filterValueChanged**(EnumConstants::maskTypes *selectedMask*, int *propNum*, int *value*, EnumConstants::threshold_or_filter *tf*)

- 1 signal inherited from QCheckBox

- 4 signals inherited from QAbstractButton
- 4 signals inherited from QWidget
- 2 signals inherited from QObject

Public Variables

int	**propNum**
EnumConstants::maskTypes	**selectedMask**
EnumConstants::threshold_or_filter	**thof**

Static Public Members

const QMetaObject	**staticMetaObject**

- 5 static public members inherited from QWidget
- 9 static public members inherited from QObject
- 1 static public member inherited from QPaintDevice

Additional Inherited Members

- 1 property inherited from QCheckBox
- 11 properties inherited from QAbstractButton
- 59 properties inherited from QWidget
- 1 property inherited from QObject
- 5 public slots inherited from QAbstractButton
- 19 public slots inherited from QWidget
- 1 public slot inherited from QObject
- 7 protected functions inherited from QCheckBox
- 14 protected functions inherited from QAbstractButton
- 35 protected functions inherited from QWidget
- 9 protected functions inherited from QObject
- 1 protected function inherited from QPaintDevice
- 1 protected slot inherited from QWidget

Detailed Description

Checbox widget for filterDialog.

Member Function Documentation

fCheckBox::fCheckBox**(EnumConstants::maskTypes *maskType*,** int ***propNums*, EnumConstants::threshold_or_filter *tf*, QWidget ****parent*)**

Creates an instance based on *maskType*, *propNums*, *tf*, and parent QWidget *parent*, if any.

[signal]void **fCheckBox::**fStateChanged**(EnumConstants::maskTypes *selectedMask*, EnumConstants::threshold_or_filter *tf*)**

This signal is emitted when the state changes giving *selectedMask* and *tf*.

[signal]void **fCheckBox::**filterValueChanged**(EnumConstants::maskTypes *selectedMask*,** int ***propNum*,** int ***value*, EnumConstants::threshold_or_filter *tf*)**

This signal is emitted when the filter value changes with *selectedMask*, *propNum*, *value*, and *tf*.

---

fSlider Class

Slider widget for filterDialog. More...

Header:	#include <fSlider>

Inherits:	QSlider

- List of all members, including inherited members
- Obsolete members

Public Functions

fSlider(EnumConstants::maskTypes *maskType*, int *propNums*, EnumConstants::threshold_or_filter *tf*, QWidget **parent*)

- 7 public functions inherited from QSlider
- 21 public functions inherited from QAbstractSlider
- 222 public functions inherited from QWidget
- 31 public functions inherited from QObject
- 15 public functions inherited from QPaintDevice

Signals

void	**fSliderReleased**(EnumConstants::maskTypes *selectedMask*, EnumConstants::threshold_or_filter *tf*)
void	**filterValueChanged**(EnumConstants::maskTypes *selectedMask*, int *propNum*, int *value*, EnumConstants::threshold_or_filter *thof*)

- 6 signals inherited from QAbstractSlider
- 4 signals inherited from QWidget
- 2 signals inherited from QObject

Public Variables

int	**propNum**
EnumConstants::maskTypes	**selectedMask**

EnumConstants::threshold_or_filter	**thof**

Static Public Members

const QMetaObject	**staticMetaObject**

- 5 static public members inherited from QWidget
- 9 static public members inherited from QObject
- 1 static public member inherited from QPaintDevice

Additional Inherited Members

- 2 properties inherited from QSlider
- 11 properties inherited from QAbstractSlider
- 59 properties inherited from QWidget
- 1 property inherited from QObject
- 3 public slots inherited from QAbstractSlider
- 19 public slots inherited from QWidget
- 1 public slot inherited from QObject
- 5 protected functions inherited from QSlider
- 8 protected functions inherited from QAbstractSlider
- 35 protected functions inherited from QWidget
- 9 protected functions inherited from QObject
- 1 protected function inherited from QPaintDevice
- 1 protected slot inherited from QWidget

Detailed Description

Slider widget for filterDialog.

Member Function Documentation

fSlider::fSlider(EnumConstants::maskTypes *maskType*, int *propNums*, EnumConstants::threshold_or_filter *tf*, QWidget **parent*)

Creates an instance based on *maskType*, *propNums*, *tf*, and parent QWidget *parent*, if any.

[signal]void fSlider::fSliderReleased(EnumConstants::maskTypes *selectedMask*, EnumConstants::threshold_or_filter*tf*)

This signal is emitted when the state changes giving *selectedMask* and *tf*.

[signal]void fSlider::filterValueChanged(EnumConstants::maskTypes *selectedMask*, int *propNum*, int *value*, EnumConstants::threshold_or_filter *thof*)

This signal is emitted when the filter value changes with *selectedMask*, *propNum*, *value*, and *thof*.

---

fSpinBox Class

Spinbox widget for filterDialog. More...

Header:	#include <fSpinBox>
Inherits:	QSpinBox

- List of all members, including inherited members
- Obsolete members

Public Functions

fSpinBox(EnumConstants::maskTypes *maskType*, int *propNums*, EnumConstants::threshold_or_filter *tf*, QWidget **parent*)

- 17 public functions inherited from QSpinBox
- 30 public functions inherited from QAbstractSpinBox
- 222 public functions inherited from QWidget
- 31 public functions inherited from QObject

- 15 public functions inherited from QPaintDevice

Public Slots

void	**displayValue**(int *value*)

- 1 public slot inherited from QSpinBox
- 4 public slots inherited from QAbstractSpinBox
- 19 public slots inherited from QWidget
- 1 public slot inherited from QObject

Signals

void	**fSlotChanged**(EnumConstants::maskTypes *selectedMask*, EnumConstants::threshold_or_filter *tf*)
void	**filterValueChanged**(EnumConstants::maskTypes *selectedMask*, int *propNum*, int *value*, EnumConstants::threshold_or_filter *tf*)

- 2 signals inherited from QSpinBox
- 1 signal inherited from QAbstractSpinBox
- 4 signals inherited from QWidget
- 2 signals inherited from QObject

Public Variables

int	**propNum**
EnumConstants::maskTypes	**selectedMask**
EnumConstants::threshold_or_filter	**thof**

Static Public Members

```
const QMetaObject    staticMetaObject
```

- 5 static public members inherited from QWidget
- 9 static public members inherited from QObject
- 1 static public member inherited from QPaintDevice

Additional Inherited Members

- 9 properties inherited from QSpinBox
- 12 properties inherited from QAbstractSpinBox
- 59 properties inherited from QWidget
- 1 property inherited from QObject
- 5 protected functions inherited from QSpinBox
- 20 protected functions inherited from QAbstractSpinBox
- 35 protected functions inherited from QWidget
- 9 protected functions inherited from QObject
- 1 protected function inherited from QPaintDevice
- 1 protected slot inherited from QWidget

Detailed Description

Spinbox widget for filterDialog.

Member Function Documentation

fSpinBox::fSpinBox**(EnumConstants::maskTypes *maskType*,** int ***propNums*,** **EnumConstants::threshold_or_filter *tf*, QWidget **parent*)**

Creates an instance based on *maskType*, *propNums*, *tf*, and parent QWidget *parent*, if any.

[slot]void **fSpinBox::**displayValue**(**int ***value*)**

Sets the display value to *value*.

[signal]void **fSpinBox::**fSlotChanged(**EnumConstants::maskTypes** ***selectedMask***, **EnumConstants::threshold_or_filter** ***tf***)

This signal is emitted when the state changes giving *selectedMask* and *tf*.

[signal]void **fSpinBox::**filterValueChanged**(EnumConstants::maskTypes** ***selectedMask***, int ***propNum***, int ***value***, **EnumConstants::threshold_or_filter** ***tf***)

This signal is emitted when the filter value changes with *selectedMask*, *propNum*, *value*, and *tf*.

---

filterControls Class

Defines the properties of the mask and conducts the filtering operations. More...

Header:	#include <filterControls>
Inherits:	QObject

- List of all members, including inherited members
- Obsolete members

Public Functions

	filterControls()
void	**defineProperties**()
cv::Mat	**filtAndGeneratePixmaps**(cv::Mat *image*, EnumConstants::maskTypes *selectedFilter*)

cv::Mat	**filterFunc**(cv::Mat *image*, EnumConstants::maskTypes *selectedFilter*)
void	**im2pixmap**(EnumConstants::maskTypes *selectedFilter*)
QPixmap	**thumb2pixmap**(cv::Mat *thumb*, EnumConstants::maskTypes *selectedFilter*)

- 31 public functions inherited from QObject

Public Slots

void	**assignVal**(EnumConstants::maskTypes *maskType*, int *propNum*, int *value*, EnumConstants::threshold_or_filter *thof*)
void	**setLabelMask**(QSharedPointer<Label> *label*)

- 1 public slot inherited from QObject

Signals

void	**valAssigned**(EnumConstants::maskTypes *maskType*, EnumConstants::threshold_or_filter *thof*)

- 2 signals inherited from QObject

Public Variables

cv::Mat	**edges**
cv::Mat	**img**
QPixmap	**labelMask**

PhotoViewer *	**photoViewer**
std::vector<filterPropertiesMeta *>	**properties**
QPixmap	**qAlpha**
QPixmap	**qImg**
QPixmap	**qImgThumb**

Static Public Members

const QMetaObject	**staticMetaObject**

- 9 static public members inherited from QObject

Additional Inherited Members

- 1 property inherited from QObject
- 9 protected functions inherited from QObject

Detailed Description

Defines the properties of the mask and conducts the filtering operations.

Member Function Documentation

filterControls::filterControls**()**

Creates an instance

[slot]void **filterControls::**assignVal**(EnumConstants::maskTypes *maskType*,** int ***propNum*,** int ***value*, EnumConstants::threshold_or_filter *thof*)**

Slot that sets the int *value* to the appropriate property indexed by *maskType*, *propNum* and *thof*.

void **filterControls::**defineProperties**()**

Defines the properties of the different masks.

cv::Mat **filterControls::**filtAndGeneratePixmaps**(**cv::Mat ***image*, EnumConstants::maskTypes *selectedFilter*)**

Returns a masked image based in the input *image* and *selectedFilter*.

cv::Mat **filterControls::**filterFunc**(**cv::Mat ***image*, EnumConstants::maskTypes *selectedFilter*)**

Filters the cv::Mat *image* based on the selected maskTypes *selectedFilter* and returns a binary image cv::Mat.

void **filterControls::**im2pixmap**(EnumConstants::maskTypes *selectedFilter*)**

Binarizes the image and converts it to a pixmap using *selectedFilter*.

[slot]void **filterControls::**setLabelMask**(QSharedPointer<Label> *label*)**

Obtains the label mask from *label* and sets it to be used as a mask for the LABELMASK filter operation.

QPixmap filterControls::thumb2pixmap**(**cv::Mat ***thumb*, EnumConstants::maskTypes *selectedFilter*)**

Returns a thubnail pixmap for the filter selection bar at the bottom of InstaDam, based on *thumb* and *selectedFilter*.

[signal]void filterControls::valAssigned(EnumConstants::maskTypes *maskType*, EnumConstants::threshold_or_filter*thof*)

This signal is emitted when a value is assigned with *maskType* and *thof*.

---

filterPropertiesMeta Class

Header: #include <filterPropertiesMeta>

- List of all members, including inherited members

Public Functions

filterPropertiesMeta(std::vector<filterProperty *> *fp*, int *nc*, EnumConstants::maskTypes *mt*)

Public Variables

EnumConstants::maskTypes	**maskType**
int	**numControls**
std::vector<filterProperty *>	**propertylist**

Detailed Description

Holds information about the particular maskType *mt* including the number of controls and the list of properties *fp*.

Member Function Documentation

filterPropertiesMeta::filterPropertiesMeta**(std::vector<filterProperty *>** ***fp*****, int** ***nc*****, EnumConstants::maskTypes** ***mt*****)**

Creates an instance based on *fp*, *nc*, and *mt*.

---

filterProperty Class

Holds the filter properties. More...

Header:	#include <filterProperty>

- List of all members, including inherited members

Public Functions

	filterProperty(std::string *propertyName*, btnTypes *bt*, int *propMin*, int *propVal*, int *propMax*, evenOdds *eo*, EnumConstants::threshold_or_filter *thof*, bool *sb*, bool *show_flag* = true)
void	**sliderAssign**(int *sliderVal*)

Public Variables

btnTypes	**btnType**
evenOdds	**evenOdd**

int	**max**
int	**min**
QString	**name**
bool	**showProp**
bool	**signalBool**
EnumConstants::threshold_or_filter	**threshold_filter**
int	**val**

Detailed Description

Holds the filter properties.

Initializes the filter properties to the provided parameter values

Member Function Documentation

filterProperty::filterProperty**(**std::string ***propertyName*****,** btnTypes ***bt*****,** int ***propMin*****,** int ***propVal*****,** int ***propMax*****,** evenOdds***eo*****,** EnumConstants::threshold_or_filter ***thof*****,** bool ***sb*****,** bool ***show_flag*** **= true)**

Creates an instance based on *propertyName*, *bt*, *propMin*, *propVal*, *propMax*, *eo*, *thof*, *sb*, and *show_flag*.

void **filterProperty::**sliderAssign(int ***sliderVal***)

Assigns the slider value to the nearest odd or even numbber close to *sliderVal*

---

newproject Class

Creates a dialog for starting a new project. More...

Header:	#include <newproject>
Inherits:	QDialog

- List of all members, including inherited members
- Obsolete members

Public Functions

	newproject(QWidget **parent* = nullptr)
virtual	**~newproject**()

- 8 public functions inherited from QDialog
- 222 public functions inherited from QWidget
- 31 public functions inherited from QObject
- 15 public functions inherited from QPaintDevice

Signals

void	**sendProject**()

- 3 signals inherited from QDialog
- 4 signals inherited from QWidget

- 2 signals inherited from QObject

Public Variables

QString	**accessToken**
QString	**databaseURL**
Project *	**newPr**
bool	**runningLocally**

Static Public Members

const QMetaObject	**staticMetaObject**

- 5 static public members inherited from QWidget
- 9 static public members inherited from QObject
- 1 static public member inherited from QPaintDevice

Additional Inherited Members

- 2 properties inherited from QDialog
- 59 properties inherited from QWidget
- 1 property inherited from QObject
- 5 public slots inherited from QDialog
- 19 public slots inherited from QWidget
- 1 public slot inherited from QObject
- 6 protected functions inherited from QDialog
- 35 protected functions inherited from QWidget
- 9 protected functions inherited from QObject
- 1 protected function inherited from QPaintDevice
- 1 protected slot inherited from QWidget

Detailed Description

Creates a dialog for starting a new project.

Member Function Documentation

newproject::newproject**(QWidget** ***parent* **= nullptr)**

Creates a new instance with parent QWidget *parent*, if any

[virtual]newproject::~newproject**()**

Destructor

[signal]void **newproject::**sendProject**()**

Signal emitted when a new project is sent.

---

serverProjectName Class

Something. More...

Header:	#include <serverProjectName>
Inherits:	QWidget

- List of all members, including inherited members

- Obsolete members

Public Functions

	serverProjectName(QWidget **parent* = nullptr)
virtual	**~serverProjectName**()

- 222 public functions inherited from QWidget
- 31 public functions inherited from QObject
- 15 public functions inherited from QPaintDevice

Signals

void	**on_pushButton_clicked**()

- 4 signals inherited from QWidget
- 2 signals inherited from QObject

Public Variables

Ui::serverProjectName *	**ui**

Static Public Members

const QMetaObject	**staticMetaObject**

- 5 static public members inherited from QWidget
- 9 static public members inherited from QObject
- 1 static public member inherited from QPaintDevice

Additional Inherited Members

- 59 properties inherited from QWidget
- 1 property inherited from QObject
- 19 public slots inherited from QWidget
- 1 public slot inherited from QObject
- 35 protected functions inherited from QWidget
- 9 protected functions inherited from QObject
- 1 protected function inherited from QPaintDevice
- 1 protected slot inherited from QWidget

Detailed Description

Something.

Member Function Documentation

serverProjectName::serverProjectName**(QWidget *_parent_ = nullptr)**

Constructs a serverProjectName instance with parent *parent*, if any.

[virtual]serverProjectName::~serverProjectName**()**

Destructor.

[signal]void **serverProjectName::**on_pushButton_clicked**()**

This signal is emitted when button is clicked.