Name: Pavan Policherla

Student access ID: ec3449

Project: Change Request 1

Date: 6/28/2016

Group Number:1

***Everything in italic should be changed as appropriate by you and should not be italic when submitted****. Also remember code is not changed until the Refactoring stage, so don’t put “I changed” or similar until section 4 of the report.*

(Zoom slider and rotation by 180°)

1. **Change Request and concepts*:***

A zoom slider needs to be added that can be implemented either in the main window of the GUI itself or in a sub menu of the tools menu. The next change that needs to be made is the addition of options to rotate the image by 180° in either direction. This option will definitely need to be only added to the tools submenu rotate.

1. **Concept Location:**

*Explain the methodology that you have used to* ***locate each significant concept*** *that was part of your change request.*

*Using Table X for dependency search, list all the files in the order that you have visited them (1st column). Explain how you have found each file (2nd column). You can simply read the source code or any other software tools that you want to use.*

*In the 3rd column, mention if the class is related to the concept. Use one of the following terms:*

* *Use* ***“Unchanged”*** *if the class has no relation to the concept but you have visited it.*
* *Use* ***“Propagating”*** *if you read the source code of the class and it guided you to the location of the concept, but you will not change it.*
* *Use* ***“Located”*** *if the class will be changed.*

*In the 4th column, write what you have learned about the class/file.*

*Draw a* ***partial class dependency graph****. It must contain all the classes that you visited and all the dependencies between these classes that you understood. Mark the classes that were* ***“Located”*** *with red text,* ***“Propagating”*** *with orange text and* ***“Unchanged”*** *with black text.*

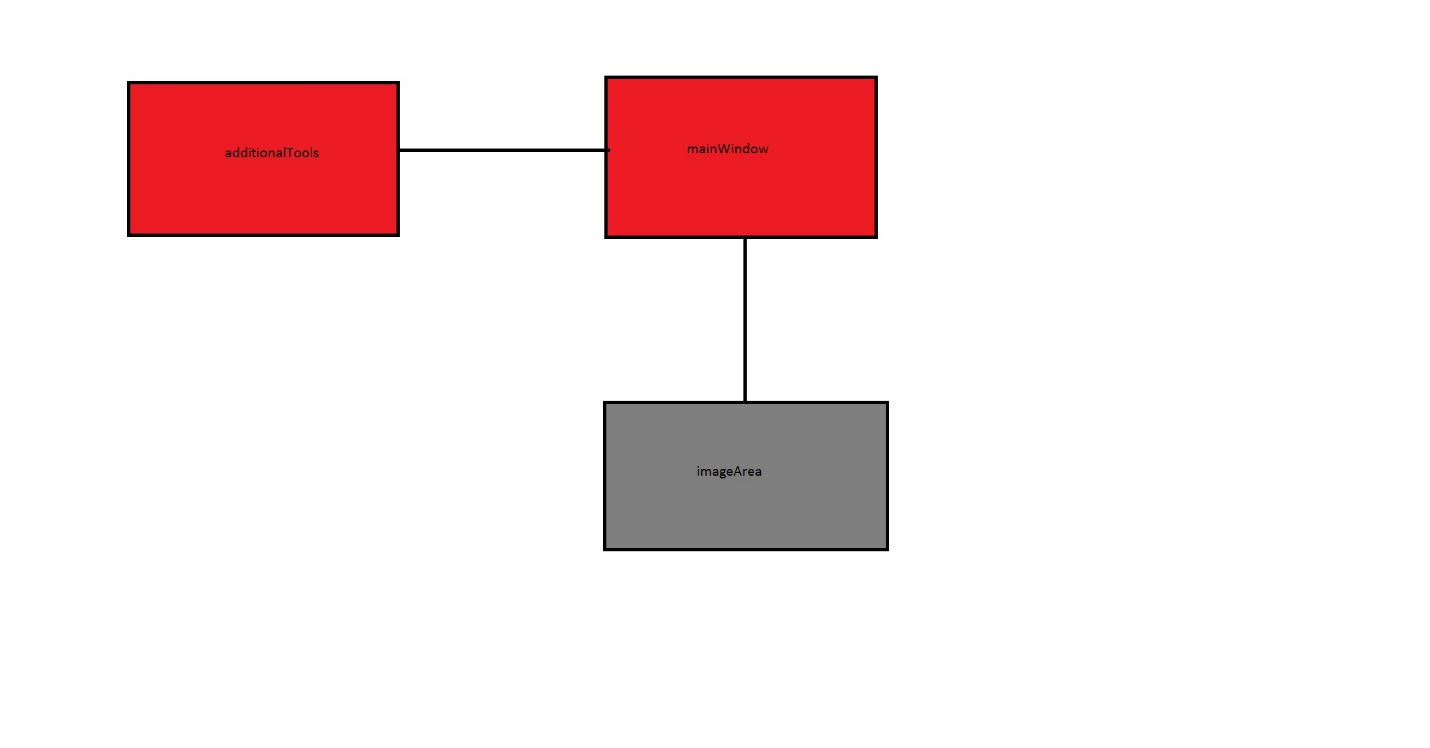
**Table 1. Dependency Search: Zoom**

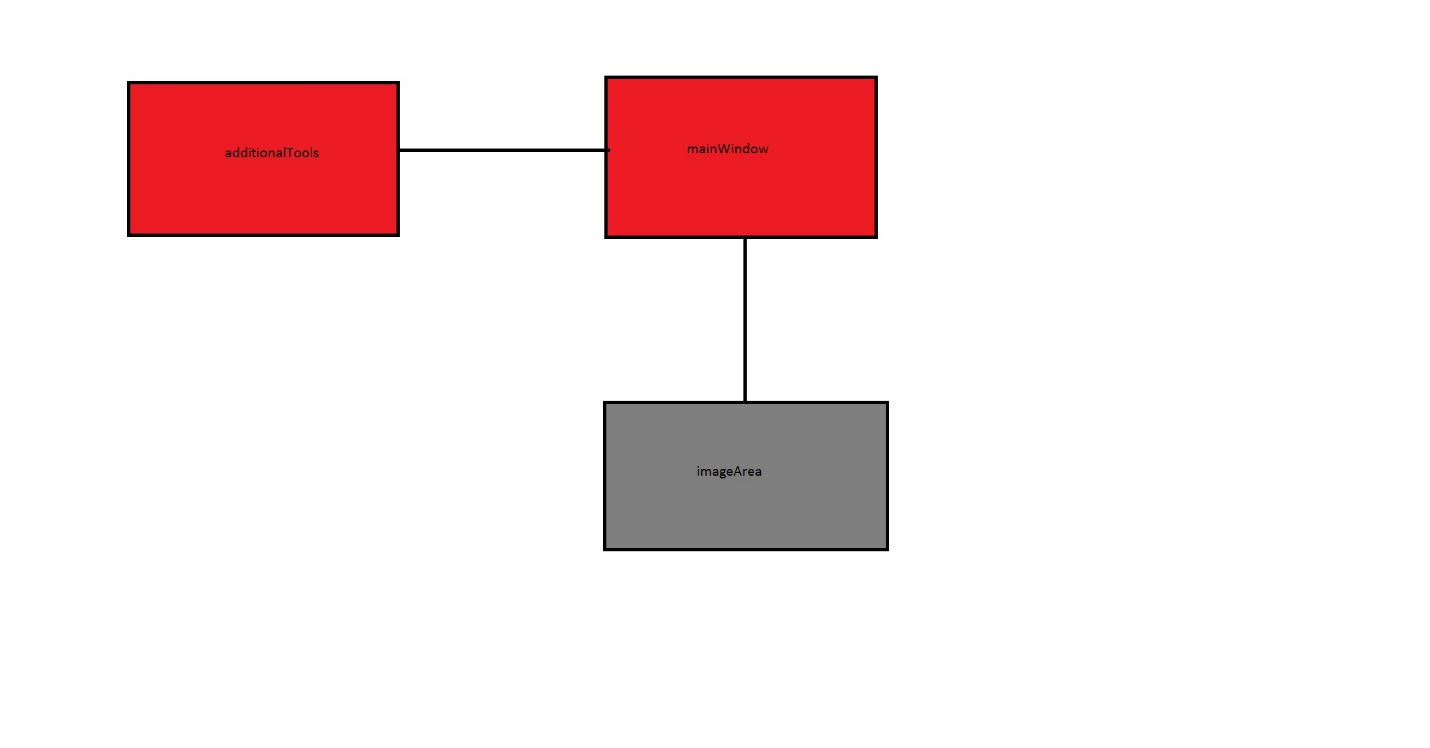
|  |  |  |  |
| --- | --- | --- | --- |
| **Class/file name** | **Tool used** | **Mark** | **Comments** |
| Additional Tools | Find in files | Located | This class seems to hold the base structure for a lot of other tool classes. |
| Image area | View call hierarchy | Unchanged | This class have its on zoom function that makes a call to the additional tools version of the method |
| Mainwindow | Find in files | Located | The methods here are the overarching implementation of the zoom functions. Also they are the implementations of the available zoom features already in the menu that do not need to be changed. |
|  |  |  |  |
|  |  |  |  |

**Table 2. Dependency Search: Rotate**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class/file name** | **Tool used** | **Mark** | **Comments** |
| Image area | Find in Files | Unchanged | Like in other finds in this class it merely already implements the actual function into predefined tools that will not be augmented |
| Additional tools | Find in files | Located | Here I have found the actual function that will actually rotate an image in either direction and degree. |
| MainWindow | Find in files | Located | Here is where the actual tools and option for the rotate function are built into the GUI. |
|  |  |  |  |
|  |  |  |  |

**Zoom Dependency**

****

**Rotate Dependency Graph**

*Using Table Y for grep search, list all the queries (2nd column) you try for each concept (1st column). The number of results by each query should be recorded in the 3rd column. Mention the correct class/file in column 4 and the tool for this query in column 5.*

*In the last column, write what you have learned about the class/file.*

**Table Y. Grep Search**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Concept** | **Query** | **#Results** | **Target class/file** | **Tool used** | **Comments** |
| Zoom | Zoom | 90 | Main window | Find in file | The zoom function seems to have a lot of dependencies in other classes that require a lot of attention to add functionality to |
| Rotate | Rotate | 42 | Main window | Find in file | Much like with the zoom function here there seems to be the tools that implement buttons that make calls to the actual rotate functions. |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1. **Impact Analysis:**

*Do a complete impact analysis based on the result of section 2. Using Table 1 to list the classes that you visited. At the beginning rows, include the class where you have located the concept, i.e. the class that will be changed (2nd column). Explain how you have found each of the classes, i.e. which tools have you used (3rd column).*

*In the 3rd column, mention if the class is related to the concept. Use one of the following terms:*

* *Use* ***“Unchanged”*** *if the class has no relation to the concept but you have visited it.*
* *Use* ***“Propagating”*** *if you read the source code of the class and it guided you to the location of the concept, but you will not change it.*
* *Use* ***“Impacted”*** *if the class will be changed.*

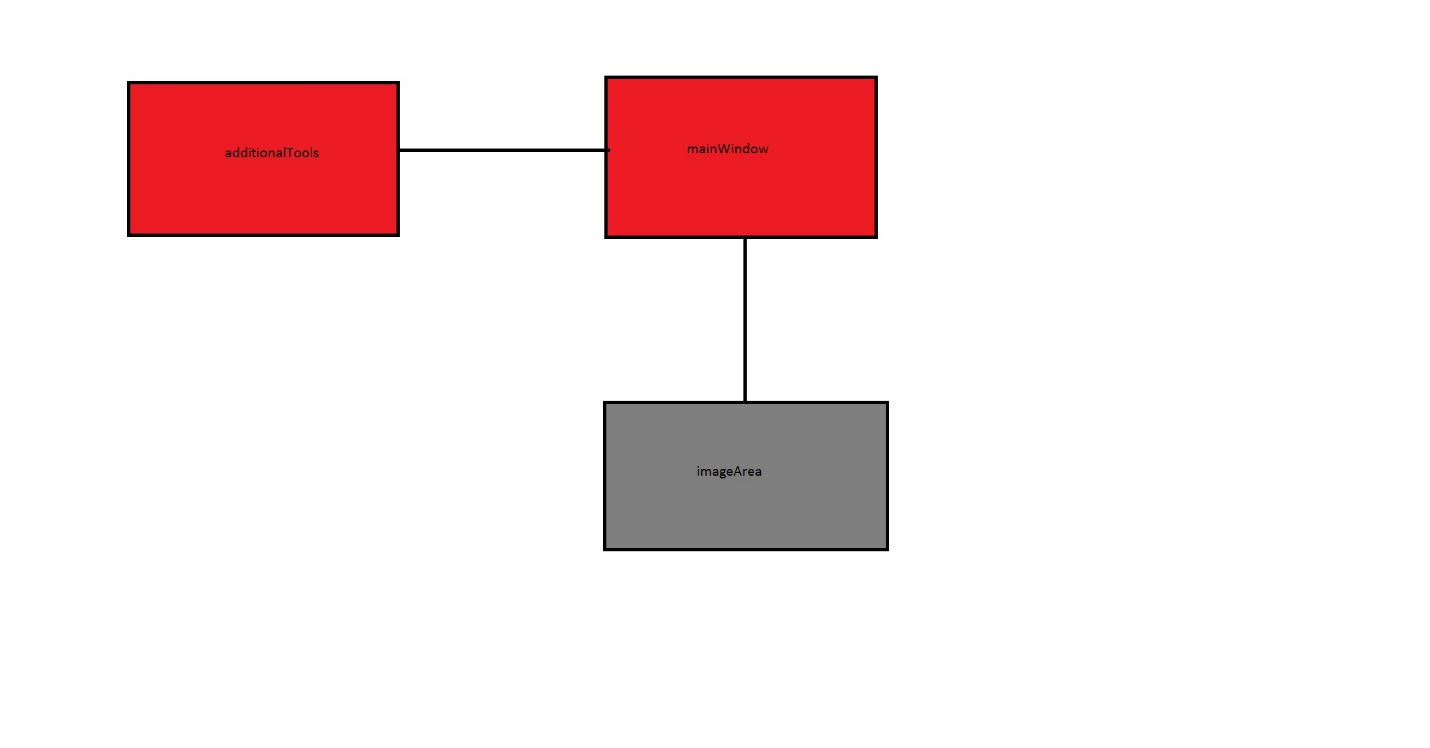
*Write short comments explaining what have you learned about each class. What other tools you would like to have in Visual Studio so that impact analysis would be faster?*

*Draw a* ***partial class interaction graph****. It must contain all the classes that you visited and all the dependencies between these classes that you understood. Mark the classes that were* ***“Impacted”*** *with red text,* ***“Propagating”*** *with orange text and* ***“Unchanged”*** *with black text.*

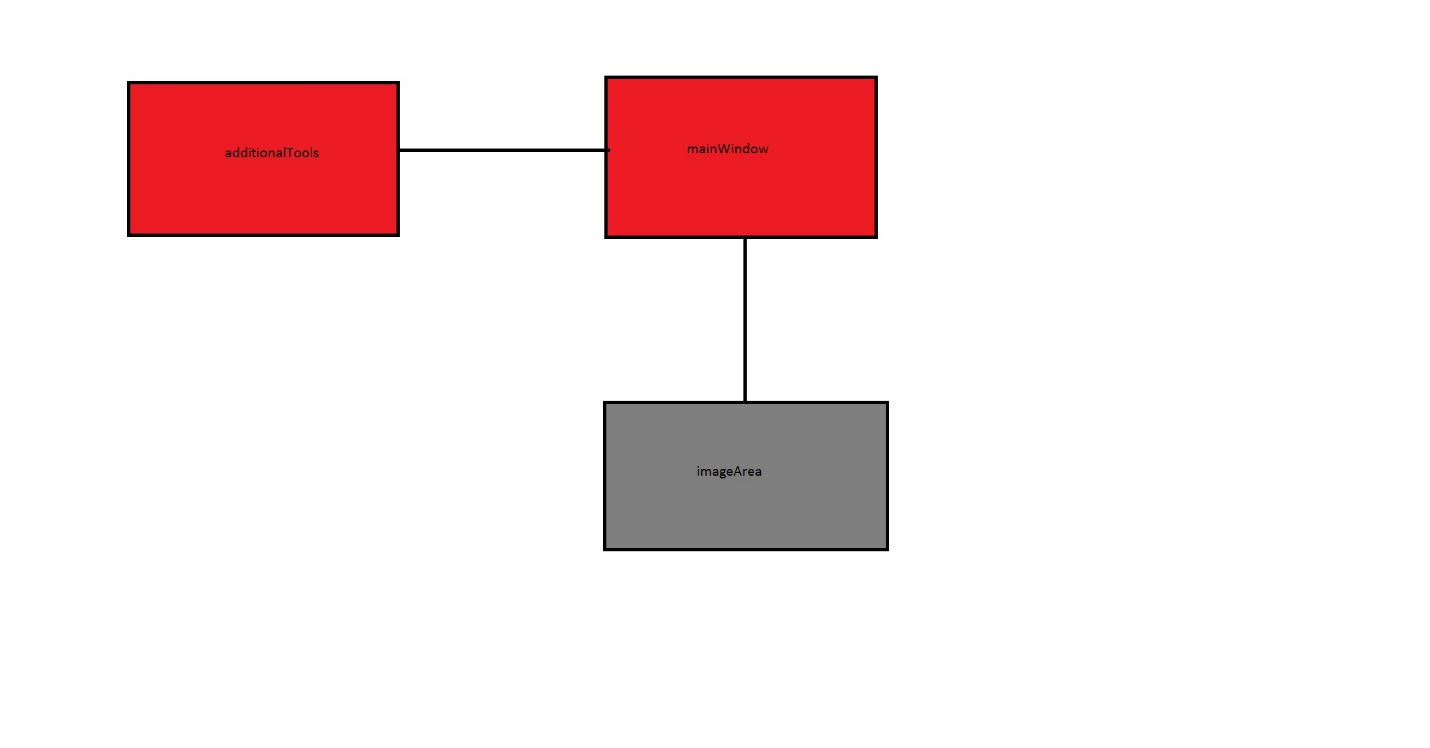
**Table 1. The list of all the classes visited during impact analysis for both the zoom slider and the rotate function.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Class name** | **Tool used** | **Mark** | **Comments** |
|  | Main Window | Find references | impacted | This class is where we will have the tools built and added to the GUI for the user to interact with. |
|  | Image Area | View call hierarchy | impacted | This is the class that has the functions that the added tools will use to perform actions |
|  | Additional Tools | View call hierarchy | unchanged | This class houses some functions that the functions in image area uses. They will not be changed but they are called in the functions that I added to the source files. |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Interaction graph zoom slider**

****

**Interaction graph rotate**

****

1. **Prefactoring:**

*Please provide a detailed journal entry describing how you went about performing prefactoring for this change request. Write down the type of your refactoring in the 3rd column (i.e. “Extract a superclass” or use the terms on https://sourcemaking.com/refactoring).*

**Table 2. Prefactoring Code Files Zoom Slider**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Code File Names** | **Refactoring Issue** | **Lines of Code** | | |
| **Added** | **Deleted** | **Total** |
|  | Mainwindow | Composing methods | 28 | 0 | 28 |
|  |  |  |  |  |  |

**Table 2. Prefactoring Code Files rotate**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Code File Names** | **Refactoring Issue** | **Lines of Code** | | |
| **Added** | **Deleted** | **Total** |
|  | Mainwindow | Composing methods | 13 | 0 | 13 |
|  |  |  |  |  |  |

1. **Actualization:**

Complete Table 3 and Table 4. *Record where (column 2, Table 4) and why (column 3, Table 4) you made changes in the source code.*

**Table 3. Actualization Summary zoom slider**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Code Files** | | | | | |
| Visited | Changed | Added | Propagating | Unchanged | Added to Changed Set |
| #3 | #1 | #0 | #1 | #1 | #0 |

**Table 4. Actualization Code Files zoom slider**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Code File Name** | **Task** | **Lines of Code** | | |
| **Added** | **Deleted** | **Total** |
| 1 | Main Window | Implement zoom slider object | 7 | 0 | 7 |
| 2 | Main window | Implement zoom function slider uses to zoom | 20 | 0 | 20 |
| 3 | Main window | Include library for sliders | 1 | 0 | 1 |

**Table 3. Actualization Summary rotate**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Code Files** | | | | | |
| Visited | Changed | Added | Propagating | Unchanged | Added to Changed Set |
| #3 | #1 | #0 | #1 | #1 | #0 |

**Table 4. Actualization Code Files rotate**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Code File Name** | **Task** | **Lines of Code** | | |
| **Added** | **Deleted** | **Total** |
| 1 | Main Window | Create and connect rotate options | 13 | 0 | 13 |

1. **Postfactoring:**

*Please provide a detailed journal entry describing how you went about performing postfactoring for this change request. Write down the type of your refactoring in the 3rd column (i.e. “Extract a superclass” or use the terms on https://sourcemaking.com/refactoring).*

**Table 5. Postfactoring Code Files zoom slider**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Code File Names** | **Refactoring Issue** | **Lines of Code** | | |
| **Added** | **Deleted** | **Total** |
| 1 | Main Window | None | 0 | 0 | 0 |
|  |  |  |  |  |  |

**Table 5. Postfactoring Code Files rotate**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Code File Names** | **Refactoring Issue** | **Lines of Code** | | |
| **Added** | **Deleted** | **Total** |
| 1 | Main Window | None | 0 | 0 | 0 |
|  |  |  |  |  |  |

1. **Verification:**

*Please provide a detailed journal entry describing how you went about performing verification for this change request.*

**Table 6. Statement Verification Zoom slider**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Code File Name** | **Coverage of Application** | | | **Tests Failed** | **Bugs Found** |
| **Total Statements** | **Covered Statements** | **%** |
| 1 | MainWindow | 12 | 12 | 100 | 0 | 0 |
|  |  |  |  |  |  |  |

**Table 6. Statement Verification rotate**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Code File Name** | **Coverage of Application** | | | **Tests Failed** | **Bugs Found** |
| **Total Statements** | **Covered Statements** | **%** |
| 1 | MainWindow | 12 | 12 | 100 | 0 | 0 |

1. **Sources:** *Include any sources that you cited or used information from*

[*https://sourcemaking.com/refactoring*](https://sourcemaking.com/refactoring)

*http://doc.qt.io/qt-5/qtwidgets-widgets-sliders-example.html*

1. **Highlighted Source Code:**

*Attach or cut and paste the code of the classes that you changed. Highlight the code that was changed or added. Use YELLOW for modified code RED for deleted code, and GREEN for added code.*

*If you only changed one method in a large file, only include that method and the file name it’s from. Likewise, if you only changed a line or two in an event map or resource file, only include a few of the surrounding lines and the file name. Do not include thousands of lines of code that you did not change!*

**Zoom**

**Mainwindow.cpp**

#include <QSlider>

void MainWindow::zoomSlider(int level)

{

switch (level)

{

case 0:

zoomPercentageAct(0.5);

break;

case 1:

zoomPercentageAct(1.0);

break;

case 2:

zoomPercentageAct(1.5);

break;

case 3:

zoomPercentageAct(2.0);

break;

case 4:

zoomPercentageAct(3.0);

break;

}

}

QSlider \*zoom = new QSlider(Qt::Horizontal, this);

zoom->setTickPosition(QSlider::TicksBothSides);

zoom->setInvertedAppearance(false);

zoom->setRange(0,4);

zoom->setValue(1);

mStatusBar->addPermanentWidget(zoom,1);

connect(zoom, SIGNAL(valueChanged(int)), this, SLOT(zoomSlider(int)));

mainWindow.h

void zoomSlider(int level);

**Rotate:**

**MainWindow.cpp**

QAction \*rotate180LAction = new QAction(tr("Counter-clockwise 180°"), this);

rotate180LAction->setIcon(QIcon::fromTheme("object-rotate-left-180°", QIcon(":/media/actions-icons/object-rotate-left.png")));

rotate180LAction->setIconVisibleInMenu(true);

connect(rotate180LAction, SIGNAL(triggered()), this, SLOT(rotateLeftImageAct()));

connect(rotate180LAction, SIGNAL(triggered()), this, SLOT(rotateLeftImageAct()));

rotateMenu->addAction(rotate180LAction);

QAction \*rotate180RAction = new QAction(tr("clockwise 180°"), this);

rotate180RAction->setIcon(QIcon::fromTheme("object-rotate-right-180°", QIcon(":/media/actions-icons/object-rotate-right.png")));

rotate180RAction->setIconVisibleInMenu(true);

connect(rotate180RAction, SIGNAL(triggered()), this, SLOT(rotateRightImageAct()));

connect(rotate180RAction, SIGNAL(triggered()), this, SLOT(rotateRightImageAct()));

rotateMenu->addAction(rotate180RAction);