Name: Joy Evans

Student access ID: er0471

Project: Project 1 – Change Request 1

Date: November 8, 2018

Group Number: 4

Change Request 1 (Dependency Search)

1. **Change Request and concepts:**

Flip Horizontally and Vertically

Add a new feature that can flip the image horizontally and vertically.

Flip, horizontal, vertical, rotate

1. **Concept Location:**

Explain the methodology that you have used to **locate each significant concept (use either dependency search or grep search)** 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 (use starUML)**. It must contain all the classes that you visited and all the dependencies among these classes that you understood. Mark the classes that were **“Located”** with red text, **“Propagating”** with orange text and **“Unchanged”** with black text.

**Table flip. Dependency Search: Flip Horizontally and Vertically**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class/file name** | **Tool used** | **Mark** | **Comments** |
| No files found |  |  | No files were found containing this concept. |

**Table horizontal. Dependency Search: Flip Horizontally and Vertically**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class/file name** | **Tool used** | **Mark** | **Comments** |
| No files found |  |  | No files were found containing this concept. |

**Table vertical. Dependency Search: Flip Horizontally and Vertically**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class/file name** | **Tool used** | **Mark** | **Comments** |
| No files found |  |  | No files were found containing this concept. |

**Table rotate. Dependency Search: Flip Horizontally and Vertically**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class/file name** | **Tool used** | **Mark** | **Comments** |
| additionaltools.cpp  void rotateImage() | Find in Files | Located | This class is used to rotate the image by 90 degrees. |
| imagearea.cpp  void rotateImage() | Find in Files | Propagating | The class calls the additional tools class to rotate image and sends a new image size. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

****

1. **Impact Analysis:**

Do a complete impact analysis based on the result of section 2. Using Table Z 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 (1st column). Explain how you have found each of the classes, i.e. which tools have you used (2nd column).

In the 3rd column, 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 you have learned about each class and what other tools you would like to have in Visual Studio so that impact analysis would be faster.

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

**Table rotate. The list of all the classes visited during impact analysis.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class name** | **Tool used** | **Mark** | **Comments** |
| additionaltools.cpp  void rotateImage() | Find in Files | Impacted | This class is used to rotate the image by 90 degrees. |
| imagearea.cpp  void rotateImage() | Find in Files | Impacted | The class calls the additional tools class to rotate image and sends a new image size. |
| imagearea.h | Find in Files | Impacted | This file calls additionaltools declarations using the additionaltools header file. |
| additionaltools.h | Find in Files | Impacted | This file declares the rotateImage class. |
| MainWindow | Find in Files | Impacted |  |
|  |  |  |  |

****

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 2nd column (e.g. “Extract a superclass” or use the terms on https://sourcemaking.com/refactoring).

**Table 3. Prefactoring Code Files**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **File Name** | **Refactoring Issue** | **Lines of Code** | | |
| **Added** | **Deleted** | **Total** |
| imagearea.cpp | Change needed to add Boolean flipImage method | 5 | 0 | 5 |
| imagearea.h | Declaration of flipImage method needed | 1 | 0 | 1 |
| additionaltools.cpp | Implementation of flip tool added | 17 | 0 | 17 |
| additionaltools.h | Declaration of flip tool added | 1 | 0 | 1 |
| mainwindow.cpp | Addition of flip QAction needed to be added | 28 | 0 | 28 |
| mainwindow.h | Flip QActions declared | 2 | 0 | 2 |

1. **Actualization:**

Complete Table 4 and Table 5. Describe the number of files you change in Table 4 and record where (column 1, Table 5) and why (column 2, Table 5) you made changes in the source code.

**Table 4. Actualization Summary**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Code Files** | | | | | |
| Visited | Changed | Added | Propagating | Unchanged | Added to Changed Set |
| **6**  **imagearea.cpp/.h**  **additionaltools.cpp/.h**  **mainwindow.cpp/.h** | **6** | **6** | **0** | **0** | **6** |

**Table 5. Actualization Code Files**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **File Name** | **Task** | **Lines of Code** | | |
| **Added** | **Deleted** | **Total** |
| imagearea.cpp | Add flipImage method to ImageArea | 5 | 0 | 5 |
| imagearea.h | Declare flipImage method | 1 | 0 | 1 |
| additionaltools.cpp | Implement flipImage method for horizontal and vertical flipping | 17 | 0 | 17 |
| additionaltools.h | Declare flipImage method | 1 | 0 | 1 |
| mainwindow.cpp | Implement QAction button for toolbar to perform flipImage | 28 | 0 | 28 |
| mainwindow.h | Declare QAction flipImage method | 2 | 0 | 2 |

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 2nd column (e.g. “Extract a superclass” or use the terms on https://sourcemaking.com/refactoring).

**Table 6. Postfactoring Code Files**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **File Name** | **Refactoring Issue** | **Lines of Code** | | |
| **Added** | **Deleted** | **Total** |
| imagearea.cpp | Change needed to add Boolean flipImage method | 5 | 0 | 5 |
| imagearea.h | Declaration of flipImage method needed | 1 | 0 | 1 |
| additionaltools.cpp | Implementation of flip tool added | 17 | 0 | 17 |
| additionaltools.h | Declaration of flip tool added | 1 | 0 | 1 |
| mainwindow.cpp | Addition of flip QAction needed to be added | 28 | 0 | 28 |
| mainwindow.h | Flip QActions declared | 2 | 0 | 2 |

1. **Verification:**

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

**Table 7. Statement Verification**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **File Name** | **Coverage of Application** | | | **Tests Failed** | **Bugs Found** |
| **Total Statements** | **Covered Statements** | **%** |
| additonaltools.cpp | 1 | 1 | 0% | 1 | 1: Using QTransform method rotate resulted in the paint object not flipping the image properly (i.e. flipped in the wrong direction) |
| additonaltools.cpp | 1 | 1 | 100 | 0 | 0: changing the rotate method to scale and using QImage instead provide the desired results when testing. |
| additionaltools.cpp additionaltools.cpp/void flipImage(bool flag) | 1 | 1 | 0 | 1 | After attempting to change the code for the flip functions using the rotate function, I wrote a message on the paint canvas. When click the flip horizontal and vertical buttons, the image flipped but in the wrong direction. |
| additionaltools.cpp/void flipImage(bool flag) | 1 | 1 | 100 | 0 | After changing the code to use the scale function, the image flipped correctly both horizontally and vertically. |

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

<http://doc.qt.io/qt-5/qtwidgets-painting-transformations-example.html>

<http://doc.qt.io/qt-5/qimage.html>

<http://doc.qt.io/qt-5/qaction.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!

**mainwindow.h**

void rotateLeftImageAct();

void rotateRightImageAct();

void flipVertImageAct();

void flipHorImageAct();

**additionaltools.h**

/\*\*

\* @brief flip image

\*

\* @param flag horizontal or vertical

\*/

void flipImage(bool flag);

**imagearea.h**

/\*\*

\* @brief flip image

\*

\* @param flag horizontal or vertical

\*/

void flipImage(bool flag);

**imagearea.cpp**

void ImageArea::flipImage(bool flag)

{

mAdditionalTools->flipImage(flag);

emit sendNewImageSize(mImage->size());

}

**Additionaltools.cpp**

void AdditionalTools::flipImage(bool flag)

{

QTransform transform;

if (flag)

{

transform.scale(-1,1);

}

else

{

transform.scale(1,-1);

}

mPImageArea->setImage(mPImageArea->getImage()->transformed(transform));

mPImageArea->resize(mPImageArea->getImage()->rect().right() + 6,

mPImageArea->getImage()->rect().bottom() + 6);

mPImageArea->update();

mPImageArea->setEdited(true);

mPImageArea->clearSelection();

}

void AdditionalTools::rotateImage(bool flag)

{

QTransform transform;

if(flag)

{

transform.rotate(90);

}

else

{

transform.rotate(-90);

}

mPImageArea->setImage(mPImageArea->getImage()->transformed(transform));

mPImageArea->resize(mPImageArea->getImage()->rect().right() + 6,

mPImageArea->getImage()->rect().bottom() + 6);

mPImageArea->update();

mPImageArea->setEdited(true);

mPImageArea->clearSelection();

}

**Mainwindow.cpp**

void MainWindow::flipVertImageAct()

{

getCurrentImageArea()->flipImage(false);

}

void MainWindow::flipHorImageAct()

{

getCurrentImageArea()->flipImage(true);

}

QMenu \*flipMenu = new QMenu(tr("Flip"));

QAction \*flipVertAction = new QAction(tr("Vertically"), this);

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

flipVertAction->setIconVisibleInMenu(true);

connect(flipVertAction, SIGNAL(triggered()), this, SLOT(flipVertImageAct()));

flipMenu->addAction(flipVertAction);

QAction \*flipHorAction = new QAction(tr("Horizontally"), this);

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

flipHorAction->setIconVisibleInMenu(true);

connect(flipHorAction, SIGNAL(triggered()), this, SLOT(flipHorImageAct()));

flipMenu->addAction(flipHorAction);

mToolsMenu->addMenu(flipMenu);