Name: Joy Evans

Student access ID: er0471

Project: Change Request 2

Date: November 29, 2018

Group Number: 4

Change Request 2 (Dependency Search)

1. **Change Request and concepts:**

Mirror Mode (vertical).

Add a new capability named vertical mirror mode. In this mirror mode, the canvas is divided by left and right half. All the figures that are drawn in one half should be mirrored to the other half each time. E.g. mirror the left half:

Concepts: Mirror, draw, canvas

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 Mirror. Dependency Search: Mirror Mode (vertical)**

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

**Table draw. Dependency Search: Mirror Mode (vertical)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class/file name** | **Tool used** | **Mark** | **Comments** |
| Imagearea.cpp/.h | Find in Files | Located | This class appears to show how the image is printed onto the canvas. Using this method may allow me to mirror the image unto the canvas. |

**Table canvas. Dependency Search: Mirror Mode (vertical)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class/file name** | **Tool used** | **Mark** | **Comments** |
| Addtionaltools.cpp/.h  Mainwindow.cpp/h  Imagearea.h | Find in Files | Located | The following files contain code for the imagearea and buttons and tools used in the window that will require changing to implement the mirror mode function. |

****

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 | Find in Files | Impacted | Found here are tools such as rotate and flip that are similar to the mirror function and can be used as a guideline to create the mirror function. |
| imagearea.cpp | Find in Files | Impacted | I believe the find in files tool is sufficient to find classes. It would be nice if the class diagram was easier to read and analyze. |
| imagearea.h | Find in Files | Impacted | Mirror function will be declared here. |
| additionaltools.h | Find in Files | Impacted | Mirror method that is used to implement tool will be declared here. |
| MainWindow.cpp/.h | Find in Files | Impacted | This is the mainwindow where the canvas and tools appear. This will need to be edited to include the new mirror vertical function. |
|  |  |  |  |

****

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** |
| sprayinstrument.cpp | Code contained a complicated expression. Expression was replaced with a simpler variable | 29 | 44 | 63 |
| imagearea.cpp | Change needed to add Boolean mirrorVert method | 5 | 0 | 5 |
| imagearea.h | Declaration of mirrorVert method needed | 1 | 0 | 1 |
| additionaltools.cpp | Implementation of mirror vertical tool added | 45 | 0 | 45 |
| additionaltools.h | Declaration of mirror vertical tool added | 1 | 0 | 1 |
| mainwindow.cpp | Addition of mirror vertial needed to be added | 18 | 0 | 18 |
| mainwindow.h | Mirror vertical 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 |
| **7**  **imagearea.cpp/.h**  **additionaltools.cpp/.h**  **mainwindow.cpp/.h**  **sprayinstrument.cpp** | **7** | **0** | **0** | **0** | **6** |

**Table 5. Actualization Code Files**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **File Name** | **Task** | **Lines of Code** | | |
| **Added** | **Deleted** | **Total** |
| imagearea.cpp | Add mirrorVert method to ImageArea | 5 | 0 | 5 |
| imagearea.h | Declare mirrorVert method | 1 | 0 | 1 |
| additionaltools.cpp | Implement mirrorVert method for horizontal and vertical flipping | 27 | 0 | 27 |
| additionaltools.h | Declare mirrorvERT method | 1 | 0 | 1 |
| mainwindow.cpp | Implement QAction button for toolbar to perform mirrorVert | 28 | 0 | 28 |
| mainwindow.h | Declare QAction mirrorVert method | 2 | 0 | 2 |
| Sprayinstrument.epp | Refactor code fragment | 29 | 44 | 63 |

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 mirrorVert method | 5 | 0 | 5 |
| imagearea.h | Declaration of mirrorVert method needed | 1 | 0 | 1 |
| additionaltools.cpp | Implementation of mirror tool added | 27 | 0 | 27 |
| additionaltools.h | Declaration of mirror tool added | 1 | 0 | 1 |
| mainwindow.cpp | Addition of mirror QAction needed to be added | 28 | 0 | 28 |
| mainwindow.h | mirror QActions declared | 2 | 0 | 2 |
| sprayinstrument.cpp | Complicated expression used. Replaced with a simple variable for calculations | 29 | 44 | 63 |

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: Attempted to use scale method, similar to flip function, to flip mirror image to right side when performing a left vertical mirror image. Instead, canvas shrunk to a smaller size and picture drawn was harder to see. |
| additonaltools.cpp | 1 | 1 | 0% | 0 | Attempted to use mirror image alone without setting dimension for left and right side. Canvas would be blank upon clicking mirror vertical button. |
| additionaltools.cpp | 1 | 1 | 0% | 0 | After specifying dimension using setClipRect, canvas would flip left side and right but it would show mirrored image upside down. |
| additionaltools.cpp/void mirrorVertImage(bool flag) | 1 | 1 | 100% | 0 | After adjusting width and height and dimension in setClipRect formula, mirror image is shown correctly upon clicking the appropriate Mirror Vertically button. |

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>

http://doc.qt.io/qt-5/qpainter.html#clipRegion

<http://doc.qt.io/qt-5/qpainter.html#setClipRect-2>

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!

**sprayinstrument.cpp**

void SprayInstrument::paint(ImageArea &imageArea, bool isSecondaryColor, bool)

{

//refactored code - initialize variable

int paintInstrument = sqrt(DataSingleton::Instance()->getPenSize() \* imageArea.getZoomFactor());

QPainter painter(imageArea.getImage());

if (isSecondaryColor)

{

painter.setPen(QPen(DataSingleton::Instance()->getSecondaryColor(),

paintInstrument,

Qt::SolidLine, Qt::RoundCap, Qt::RoundJoin));

}

else

{

painter.setPen(QPen(DataSingleton::Instance()->getPrimaryColor(),

paintInstrument,

Qt::SolidLine, Qt::RoundCap, Qt::RoundJoin));

}

int x, y;

//refactored code - implement variable

for (int i(0); i < 12; i++)

{

switch (i) {

case 0: case 1: case 2: case 3:

x = (qrand() % 5 - 2)

\* paintInstrument;

y = (qrand() % 5 - 2)

\* paintInstrument;

break;

case 4: case 5: case 6: case 7:

x = (qrand() % 10 - 4)

\* paintInstrument;

y = (qrand() % 10 - 4)

\* paintInstrument;

break;

case 8: case 9: case 10: case 11:

x = (qrand() % 15 - 7)

\*paintInstrument;

y = (qrand() % 15 - 7)

\* paintInstrument;

break;

}

painter.drawPoint(mEndPoint.x() + x,

mEndPoint.y() + y);

}

//Joy Evans - er0471

//Code included a complicated expression " sqrt(DataSingleton::Instance()->getPenSize() \* imageArea.getZoomFactor())". To solve this code smell, I extracted a variable and created int paintInstrument to perform the same function.

//QPainter painter(imageArea.getImage());

//if(isSecondaryColor)

//{

// painter.setPen(QPen(DataSingleton::Instance()->getSecondaryColor(),

// sqrt(DataSingleton::Instance()->getPenSize() \* imageArea.getZoomFactor()),

// Qt::SolidLine, Qt::RoundCap, Qt::RoundJoin));

//}

//else

//{

// painter.setPen(QPen(DataSingleton::Instance()->getPrimaryColor(),

// sqrt(DataSingleton::Instance()->getPenSize() \* imageArea.getZoomFactor()),

// Qt::SolidLine, Qt::RoundCap, Qt::RoundJoin));

//}

//int x, y;

//for(int i(0); i < 12; i++)

//{

// switch(i) {

// case 0: case 1: case 2: case 3:

// x = (qrand() % 5 - 2)

// \* sqrt(DataSingleton::Instance()->getPenSize() \* imageArea.getZoomFactor());

// y = (qrand() % 5 - 2)

// \* sqrt(DataSingleton::Instance()->getPenSize() \* imageArea.getZoomFactor());

// break;

// case 4: case 5: case 6: case 7:

// x = (qrand() % 10 - 4)

// \* sqrt(DataSingleton::Instance()->getPenSize() \* imageArea.getZoomFactor());

// y = (qrand() % 10 - 4)

// \* sqrt(DataSingleton::Instance()->getPenSize() \* imageArea.getZoomFactor());

// break;

//case 8: case 9: case 10: case 11:

// x = (qrand() % 15 - 7)

// \* sqrt(DataSingleton::Instance()->getPenSize() \* imageArea.getZoomFactor());

// y = (qrand() % 15 - 7)

// \* sqrt(DataSingleton::Instance()->getPenSize() \* imageArea.getZoomFactor());

//break;

// }

// painter.drawPoint(mEndPoint.x() + x,

// mEndPoint.y() + y);

//}

imageArea.setEdited(true);

painter.end();

imageArea.update();

}

**mainwindow.h**

void mirrorVertLeftImageAct();

void mirrorVertRightImageAct();

**additionaltools.h**

void flipImage(bool flag);

/\*\*

void mirrorVertImage(bool flag);

/\*\*

**imagearea.h**

void flipImage(bool flag);

void mirrorVertImage(bool flag);

**imagearea.cpp**

void ImageArea::mirrorVertImage(bool flag)

{

mAdditionalTools->mirrorVertImage(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::mirrorVertImage(bool flag)

{

int height = mPImageArea->getImage()->height();

int width = mPImageArea->getImage()->width();

QPainter paintObj(mPImageArea->getImage());

QTransform transform;

if (flag)

{

QImage newImage = mPImageArea->getImage()->mirrored(true,false);

paintObj.setClipRect(width/2,0,width,height);

paintObj.drawImage(0, 0, newImage);

}

else

{

QImage newImage = mPImageArea->getImage()->mirrored(true, false);

paintObj.setClipRect(0, 0, width / 2, height);

paintObj.drawImage(0, 0, newImage);

}

mPImageArea->update();

mPImageArea->setEdited(true);

mPImageArea->clearSelection();

}

**Mainwindow.cpp**

QMenu \*mirrorMenu = new QMenu(tr("Mirror Vertically"));

QAction \*mirrorVertLeftAction = new QAction(tr("Left"), this);

connect(mirrorVertLeftAction, SIGNAL(triggered()), this, SLOT(mirrorVertLeftImageAct()));

mirrorMenu->addAction(mirrorVertLeftAction);

QAction \*mirrorVertRightAction = new QAction(tr("Right"), this);

connect(mirrorVertRightAction, SIGNAL(triggered()), this, SLOT(mirrorVertRightImageAct()));

mirrorMenu->addAction(mirrorVertRightAction);

mToolsMenu->addMenu(mirrorMenu);

QMenu \*zoomMenu = new QMenu(tr("Zoom"));

{

getCurrentImageArea()->flipImage(true);

}

void MainWindow::mirrorVertLeftImageAct()

{

getCurrentImageArea()->mirrorVertImage(true);

}

void MainWindow::mirrorVertRightImageAct()

{

getCurrentImageArea()->mirrorVertImage(false);

}