Name:

Student access ID:

Project:

Date:

Group Number:

***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.*

(Make effect work on a selected area)

1. **Change Request and concepts:**

For this change request what is intended is that when in easy paint you will be able to select a specific area of a picture or image, and then from that selected area apply an effect to the area specifically.

Concepts to not here are apply effect.

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:*

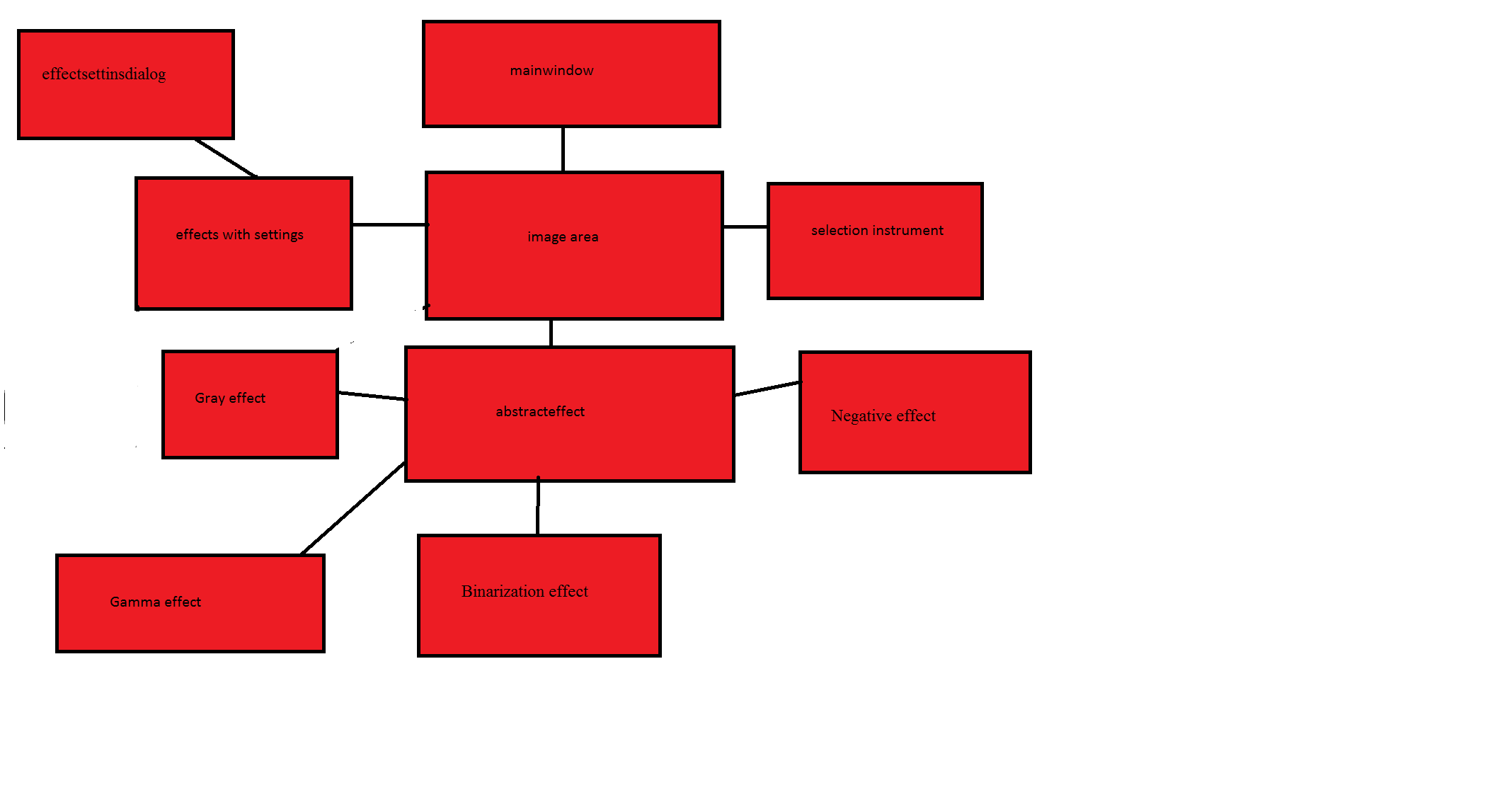
* *Use* ***“Unchanged”*** *if the class has no relation to the concept but you have visited it.*
* *Use* ***“Propagating”*** *if you read the source code of the class and it guided you to the location of the concept, but you will not change it.*
* *Use* ***“Located”*** *if the class will be changed.*

*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: Effect**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class/file name** | **Tool used** | **Mark** | **Comments** |
| GammaEffect | Find in files | Located | This class has the apply effect function that will need to be changed so that it will notice the selected area |
| Gray effect | Class diagram | located | Same as gamma effect |
| Binarization effect | Class diagram | located | Same as gamma effect |
| effectwithsettings | Class diagram | located | This class has the virtual function that the effects with dialogs will use to apply the effect it needs to be overloaded |
| Effectsettingsdialog | Class diagram | Located | Will have to be changed to handle the effect with settings changes |
| abstract effect | Class diagram | located | The apply effect here also needs to be overloaded |
| image area | Class diagram | located | Here the algorithm inside the apply effect function needs to be changed to account for the selected area so it will display the right outcome. |
| selection instrument | Class diagram | located | Here is where an apply effect function that will use the selection tool will be implemented |
| Main window | Class diagram | located | The effects act function will need some modification |
| Negative effect | Class diagram | located | Applyeffect function will need to be changed |

****

*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 1. Grep Search**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Concept** | **Query** | **#Results** | **Target class/file** | **Tool used** | **Comments** |
| Applyeffect | Applyeffect | 26 | All files | Find in files | Found all the classes with the apply effect function that will need to be changed |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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:*

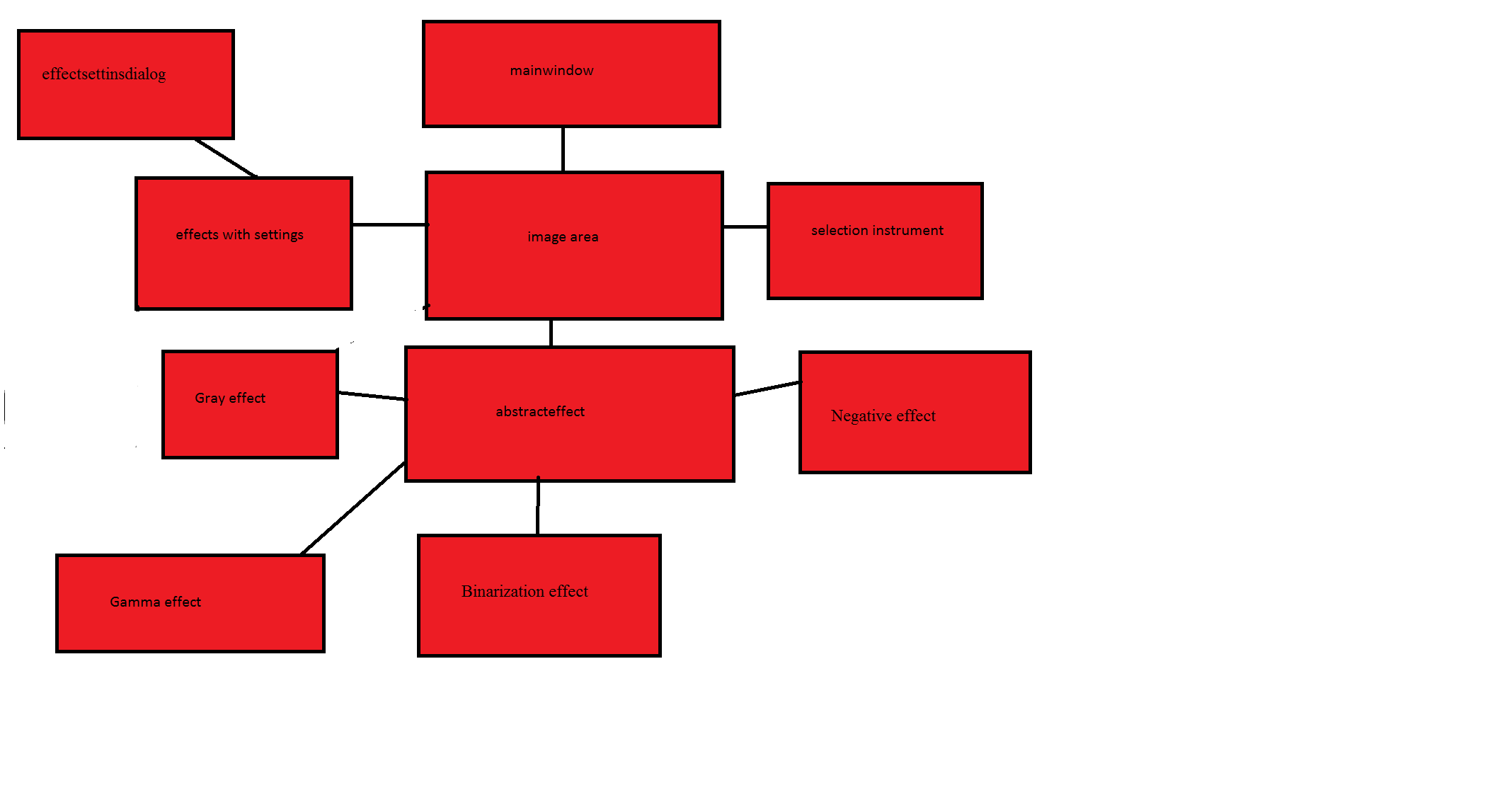
* *Use* ***“Unchanged”*** *if the class has no relation to the concept but you have visited it.*
* *Use* ***“Propagating”*** *if you read the source code of the class and it guided you to the location of the concept, but you will not change it.*
* *Use* ***“Impacted”*** *if the class will be changed.*

*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.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Class name** | **Tool used** | **Mark** | **Comments** |
|  | GammaEffect | Find in files | impacted | Apply effect will need to be changed |
|  | Gray effect | Class diagram | impacted | Apply effect will need to be changed |
|  | Binarization effect | Class diagram | impacted | Apply effect will need to be changed |
|  | effectwithsettings | Class diagram | impacted | Apply effect will need to be changed |
|  | Effectsettingsdialog | Class diagram | Impacted | Function will change to incorporate other changes in functionality69 |
|  | abstract effect | Class diagram | impacted | Apply effect will need to be overloaded |
|  | image area | Class diagram | impacted | Apply effect will need to be changed |
|  | selection tool | Class diagram | impacted | Apply effect function will need to be implemented |
|  | Main window | Class diagram | Impacted | Effects act function is changed |
|  | Negative effect | Class diagram | Impacted | Apply effect will need to be changed |



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**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Code File Names** | **Refactoring Issue** | **Lines of Code** | | |
| **Added** | **Deleted** | **Total** |
|  | Main window.cpp | Function needs expanding | 8 | 0 | 8 |
|  | Image area.cpp | Need to include library and function needs expanding | 13 | 0 | 13 |
|  | Abstracteffect.h | Function header needed to be changed | 0 | 0 | 0 |
|  | Effectsettingsdialog.h | Function header need to be changed | 0 | 0 | 0 |
|  | Effectsettingsdialog.cpp | Add library, expand function and connect functions | 18 | 0 | 18 |
|  | Effectswithsettings.h | Header needs to be changed | 0 | 0 | 0 |
|  | Effectswithsettings.cpp | Function header and algorithm changes implemented | 0 | 0 | 0 |
|  | Selectioninstrument.h | Added libraries and added function | 3 | 0 | 0 |
|  | Selectioninstrument.cpp | Added libraries and implemented function | 11 | 0 | 11 |
|  | Grayeffect.h | Function header changed | 0 | 0 | 0 |
|  | Grayeffect.cpp | Made change to function to reflect new header | 15 | 0 | 15 |
|  | Gammaeffect.h | Changed function headers | 0 | 0 | 0 |
|  | Gammaeffect.cpp | Made changes that reflected the header changes to functions | 15 | 0 | 15 |
|  | Binarizationeffect.h | Changed function header | 0 | 0 | 0 |
|  | Binarizationeffect.cpp | Made changes to the function | 15 | 0 | 15 |
|  | Negativeeffect.h | Changed function header | 0 | 0 | 0 |
|  | Negativeeffect.cpp | Made changes to function to reflect header changes | 21 | 0 | 21 |

In prefactoring I went through the classes that needed to be changed and looked at any function that pertained to effects and made changes to them. For that actual effect classes most of the changes were the same across the board so those changes were all the same. For many of the classes minor changes needed to be made to headers and libraries needed to be included. Some of the classes had to have the libraries included in both the header and the implementation files otherwise an error would occur.

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**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Code Files** | | | | | |
| Visited | Changed | Added | Propagating | Unchanged | Added to Changed Set |
| 10 | 10 | 0 | 0 | 0 | 10 |

**Table 4. Actualization Code Files**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Code File Names** | **Task** | **Lines of Code** | | |
| **Added** | **Deleted** | **Total** |
|  | Main window.cpp | Function needs expanding | 8 | 0 | 8 |
|  | Image area.cpp | Need to include library and function needs expanding | 13 | 0 | 13 |
|  | Abstracteffect.h | Function header needed to be changed | 0 | 0 | 0 |
|  | Effectsettingsdialog.h | Function header need to be changed | 0 | 0 | 0 |
|  | Effectsettingsdialog.cpp | Add library, expand function and connect functions | 18 | 0 | 18 |
|  | Effectswithsettings.h | Header needs to be changed | 0 | 0 | 0 |
|  | Effectswithsettings.cpp | Function header and algorithm changes implemented | 0 | 0 | 0 |
|  | Selectioninstrument.h | Added libraries and added function | 3 | 0 | 0 |
|  | Selectioninstrument.cpp | Added libraries and implemented function | 11 | 0 | 11 |
|  | Grayeffect.h | Function header changed | 0 | 0 | 0 |
|  | Grayeffect.cpp | Made change to function to reflect new header | 15 | 0 | 15 |
|  | Gammaeffect.h | Changed function headers | 0 | 0 | 0 |
|  | Gammaeffect.cpp | Made changes that reflected the header changes to functions | 15 | 0 | 15 |
|  | Binarizationeffect.h | Changed function header | 0 | 0 | 0 |
|  | Binarizationeffect.cpp | Made changes to the function | 15 | 0 | 15 |
|  | Negativeeffect.h | Changed function header | 0 | 0 | 0 |
|  | Negativeeffect.cpp | Made changes to function to reflect header changes | 21 | 0 | 21 |

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**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Code File Names** | **Refactoring Issue** | **Lines of Code** | | |
| **Added** | **Deleted** | **Total** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

No postfactoring performed. When compiled and program ran the files functioned and no errors occurred. And I could not determine anyways to further optimize the files, basically I was lucky enough to not find any sort of redundant code.

1. **Verification:**

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

**Table 6. Statement Verification**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Code File Name** | **Coverage of Application** | | | **Tests Failed** | **Bugs Found** |
| **Total Statements** | **Covered Statements** | **%** |
|  | Effectsettingsdialog | 20 | 20 | 100 | 0 | 0 |
|  | Effectswithsettings | 2 | 2 | 100 | 0 | 0 |
|  | imagearea | 7 | 7 | 100 | 0 | 0 |
|  | selectioninstrument | 5 | 5 | 100 | 0 | 0 |
|  | Abstracteffect | 1 | 1 | 100 | 0 | 0 |
|  | Grayeffect | 11 | 11 | 100 | 0 | 0 |
|  | Gammaeffect | 11 | 11 | 100 | 0 | 0 |
|  | Binarizationeffect | 11 | 11 | 100 | 0 | 0 |
|  | Negativeeffect | 11 | 11 | 100 | 0 | 0 |

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

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

*http://stackoverflow.com/questions/14167785/qt-troubles-with-painting-selection-box*

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!*

*Effectsettingdialog.cpp*

#include <qsignalmapper.h>

QSignalMapper mapper;

connect(mOkButton, &QPushButton::clicked, this, [&]{applyMatrix(topLeft, bottomRight); });

connect(mApplyButton, &QPushButton::clicked, this, [&]{applyMatrix(topLeft, bottomRight); })

void EffectSettingsDialog::applyMatrix(QPoint\* topLeft, QPoint\* bottomRight)

int width = 0;

int height = 0;

int q, w;

if (topLeft == NULL || bottomRight == NULL)

{

width = mImage.width();

height = mImage.height();

q = 2;

w = 2;

}

else

{

q = topLeft->x();

w = topLeft->y();

width = bottomRight->x();

height = bottomRight->y();

}

for(int i = w+2; i < height - 2; ++i)

{

for(int j = q+2; j < width - 2; ++j

effectsettingsdialog.h

explicit EffectSettingsDialog(const QImage &img, AbstractEffectSettings \*settingsWidget, QPoint\* = NULL, QPoint\* = NULL, QWidget \*parent = 0);

void applyMatrix(QPoint\* = NULL, QPoint\* = NULL);

abstracteffect.h

virtual void applyEffect(ImageArea &imageArea, QPoint \*topLeft = NULL, QPoint\* bottomRight = NULL) = 0;

binarizationeffect.cpp

void BinarizationEffect::applyEffect(ImageArea &imageArea, QPoint \*topLeft, QPoint \*bottomRight)

makeBinarization(imageArea, 200, 100, topLeft, bottomRight);

void BinarizationEffect::makeBinarization(ImageArea &imageArea, int coeff1, int coeff2, QPoint\* topLeft, QPoint\* bottomRight)

int width = 0;

int height = 0;

int q, w;

if (topLeft == NULL || bottomRight == NULL)

{

width = imageArea.getImage()->width();

height = imageArea.getImage()->height();

q = 0;

w = 0;

}

else

{

q = topLeft->x();

w = topLeft->y();

width = bottomRight->x();

height = bottomRight->y();

for (int x(q); x < width; x++)

{

for (int y(w); y < height; y++)

binarizationeffect.h

void applyEffect(ImageArea &imageArea, QPoint \*topLeft = NULL, QPoint \*bottomRight = NULL);

void makeBinarization(ImageArea &imageArea, int coeff1, int coeff2, QPoint\* topLeft= NULL, QPoint\* bottomRight= NULL);

effectwithsettings.cpp

void EffectWithSettings::applyEffect(ImageArea &imageArea, QPoint\* topLeft, QPoint\* bottomRight)

{

EffectSettingsDialog dlg(\*imageArea.getImage(), getSettingsWidget(), topLeft, bottomRight);

Effectwithsettings.h

virtual void applyEffect(ImageArea &imageArea, QPoint\* = NULL, QPoint\* = NULL)

gammaeffect.cpp

void GammaEffect::applyEffect(ImageArea &imageArea, QPoint \*topLeft, QPoint \*bottomRight)

makeGamma(imageArea, 2, topLeft, bottomRight);

void GammaEffect::makeGamma(ImageArea &imageArea, float modificator, QPoint\* topLeft,QPoint\* bottomRight)

int width = 0;

int height = 0;

int q, w;

if (topLeft == NULL||bottomRight==NULL)

{

width = imageArea.getImage()->width();

height = imageArea.getImage()->height();

q = 0;

w = 0;

}

else

{

q = topLeft->x();

w = topLeft->y();

width = bottomRight->x();

height = bottomRight->y();

}

for(int x(q); x < width; x++)

{

for(int y(w); y < height; y++)

gammaeffect.h

void applyEffect(ImageArea &imageArea, QPoint \*topLeft = NULL, QPoint \*bottomRight = NULL);

void makeGamma(ImageArea &imageArea, float modificator, QPoint\* = NULL, QPoint\* = NULL);

grayeffect.cpp

void GrayEffect::applyEffect(ImageArea &imageArea, QPoint\* topLeft , QPoint\* bottomRight)

int width = 0;

int height = 0;

int q, w;

if (topLeft == NULL || bottomRight == NULL)

{

width = imageArea.getImage()->width();

height = imageArea.getImage()->height();

q = 0;

w = 0;

}

else

{

q = topLeft->x();

w = topLeft->y();

width = bottomRight->x();

height = bottomRight->y();

}

for(int i(q); i < width; i++)

{

for(int y(w); y < height; y++)

{

Grayeffect.h

void applyEffect(ImageArea &imageArea, QPoint\* topLeft = NULL, QPoint\* bottomRight = NULL);

negativeeffect.cpp

void NegativeEffect::applyEffect(ImageArea &imageArea, QPoint\* topLeft, QPoint\* bottomRight)

int width = 0;

int height = 0;

int q, w;

if (topLeft == NULL || bottomRight == NULL)

{

width = imageArea.getImage()->width();

height = imageArea.getImage()->height();

q = 0;

w = 0;

}

else

{

q = topLeft->x();

w = topLeft->y();

width = bottomRight->x();

height = bottomRight->y();

for (int x(q); x < width; x++)

{

for (int y(w); y < height; y++)

{

Negativeeffect.h

void applyEffect(ImageArea &imageArea, QPoint\* topLeft = NULL, QPoint\* bottomRight = NULL);

imagearea.cpp

#include <iostream>

SelectionInstrument \*instrument = dynamic\_cast <SelectionInstrument\*> (mInstrumentsHandlers.at(DataSingleton::Instance()->getInstrument()));

if (instrument == NULL)

{

}

else

{

mEffectHandler = mEffectsHandlers.at(effect);

instrument->applyEffectSelected(\*this, mEffectHandler, effect);

if (effect == NEGATIVE)

mUndoStack->undo();

mUndoStack->clear();

}

Selectioninstrument.cpp

#include "../effects/abstracteffect.h"

void SelectionInstrument::pasteImage(ImageArea &imageArea, bool undoable)

void SelectionInstrument::applyEffectSelected(ImageArea &imageArea, AbstractEffect \*abstr, EffectsEnum effect)

{

if (effect != NEGATIVE)

abstr->applyEffect(imageArea, &mTopLeftPoint, &mBottomRightPoint);

else {

copyImage(imageArea);

mSelectedImage.invertPixels();

pasteImage(imageArea, false);

}

Selectioninstrument.h

#include "../effects/abstracteffect.h"

#include "../easypaintenums.h"

void pasteImage(ImageArea &imageArea, bool undoable = true)

void applyEffectSelected(ImageArea &imageArea, AbstractEffect \*abstr, EffectsEnum);

mainwindow.cpp

{

if (ImageArea \*imageArea = getCurrentImageArea())

{

imageArea->applyEffect(mEffectsActMap.key(currentAction));

}

else

{

QAction \*currentAction = static\_cast<QAction\*>(sender());