Name: Caleb Latimer

Student access ID: ej1297

Project: easyPaint

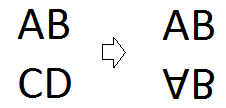
Date: 3/24/2017

Group Number: 1

**Mirror Mode (horizontal)**

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

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



Note the users must be allowed to decide which half to mirror.

Important, but external implementation

Irrevelant to concept location

Significant concept

1. **Concept Location:**

I went with a dependency search for this change request, I started my search in main.

**Table 1. Dependency Search: canvas**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class/file name** | **Tool used** | **Mark** | **Comments** |
| Main | No tool, starting point | Unchanged | Main calls mainwindow and datasingleton |
| mainwindow | Go to definition | Changed | Class contains instantiation of imagearea which is essentially the canvas. Will be changed due to new capability to take in user input |
| datasingleton | Go to definition | Unchanged | Implementation affects menu and instruments, capability is not an instrument |
| Image area | Go to definition | Changed | Core area that I need to work with to implement mirror function |
| Additional tools | Go to definition | Changed | Contains information related to tools which is what the change will be implemented as |



1. **Impact Analysis:**

**Table 1. The list of all the classes visited during impact analysis.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Class name** | **Tool used** | **Mark** | **Comments** |
| 1 | Main | No tool, starting point | Unchanged | Calls mainwindow and datasingleton |
| 2 | mainwindow | Go to definition | Changed | Calls easypaintenums, imagearea,datasingleton, settings dialog, palettebar, zoombar, toolbar |
| 3 | datasingleton | Go to definition | Unchanged | Calls easypaintenums |
| 4 | Image area | Go to definition | Changed | Calls easypaintenums, additionalTools,undocommand, all instrument files and all effect files |
| 5 | Additional tools | Go to definition | Changed | Calls image area, resizeDialog, |
| 6 | EasyPaintenums | Find all references | Unchanged | None |
| 7 | Settings dialog | Find all references | Unchanged | none |
| 8 | palettebar | Find all references | unchanged | none |
| 9 | zoombar | Find all references | Unchanged | none |
| 10 | Toolbar | Find all references | Unchanged | none |
| 11 | Undocommand | Find all references | Unchanged | None |
| 12 | resizedialog | Find all references | Unchanged | But could be useful if the arguments for width and height are passed here |
| 13 | Instrument files | Go to definition | Unchanged | None |
| 14 | Effect files | Go to definition | Unchanged | none |



1. **Prefactoring:**

There is no prefactorting for the menu option all of the code will be new code, however due to the requirements I did refactor my prior change request for 45, horizontal and vertical line by extracting a function.

**Table 2. Prefactoring Code Files**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Code File Names** | **Refactoring Issue** | **Lines of Code** | | |
| **Added** | **Deleted** | **Total** |
| 1 | lineinstrument | Extract a function | 16 | 27 | 43 |

1. **Actualization:**

**Table 3. Actualization Summary**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Code Files** | | | | | |
| Visited | Changed | Added | Propagating | Unchanged | Added to Changed Set |
| 6 | 6 | 0 | 0 | 5 | 0 |

**Table 4. Actualization Code Files**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Code File Name** | **Task** | **Lines of Code** | | |
| **Added** | **Deleted** | **Total** |
| 1 | Mainwindow.cpp | New menu dropdown for mirror mode | 15 | 0 | 15 |
| 2 | Mainwindow.h | Included header for new function | 2 | 0 | 17 |
| 3 | Imagearea.h | Included header for new function | 2 | 0 | 19 |
| 4 | Imagearea.cpp | Implement area copy function | 8 | 0 | 27 |
| 5 | Additionaltools.h | Included header for new function | 2 | 0 | 29 |
| 6 | Additionaltools.cpp | Implement redrawing | 30 | 0 | 59 |

1. **Postfactoring:**

No postfactoring was done in this request

1. **Verification:**

**Table 6. Statement Verification**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Code File Name** | **Coverage of Application** | | | **Tests Failed** | **Bugs Found** |
| **Total Statements** | **Covered Statements** | **%** |
| 1 | Imagearea.cpp | 2 | 2 | 100% | 0 | 1 |
| 2 | Main.cpp | 3 | 3 | 100% | 0 | 1 |

**Test 1:** Fully examine the mirror() function for implementation. Change to true,false for horizontal flip only. Passed, does render on canvas. Try for true,true for a horizontal flip at a diagnol positon. Passed

**Test 2:** Modify mini-menu with option to pick segment of canvas for source. Passed, drop-down functional, feature complete. Test Additional component for splitting canvas of multiple sizes, passed. Test overwriting flipped segment and called option for reverse effect, passed. Run program with varying figures and positions. Passed

1. **Sources:**

Combining images: <https://forum.qt.io/topic/24320/combining-images-sided-by-side-using-qpainter-and-qimage/2>

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

Safayeth, doing a walkthrough of change and working to implement a menu as a team with paired-programming was a great resource

1. **Highlighted Source Code:**

**Mainwindow.h**

void horizontalCpyTop(); /\*ej1297\*/

void horizontalCpyBot();

**Mainwindow.cpp**

/\*Implementation of the mirror option to the menu\*/

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

QAction \*horizontalTopMirrorAction = new QAction(tr("Horizontal Mirror (Top)"), this);

connect(horizontalTopMirrorAction, SIGNAL(triggered()), this, SLOT(horizontalCpyTop()));

mirrorMenu->addAction(horizontalTopMirrorAction);

QAction \*horizontalBotMirrorAction = new QAction(tr("Horizontal Mirror (Bot)"), this);

connect(horizontalBotMirrorAction, SIGNAL(triggered()), this, SLOT(horizontalCpyBot()));

mirrorMenu->addAction(horizontalBotMirrorAction);

/\*ej1297\*/

/\*Implementation of mirror action\*/

void MainWindow::horizontalCpyTop()

{

getCurrentImageArea()->horizontalMirrorImageTop();

}

/\*ej1297\*/

void MainWindow::horizontalCpyBot()

{

getCurrentImageArea()->horizontalMirrorImageBot();

}

/\*ej1297\*/

**Imagearea.h**

void horizontalMirrorImageTop(); /\*ej1297\*/

void horizontalMirrorImageBot();

**Imagearea.cpp**

/\*Image area function implemented\*/

void ImageArea::horizontalMirrorImageTop()

{

mAdditionalTools->mirrorImageTop(mImage->width(), mImage->height());

emit sendNewImageSize(mImage->size());

}

void ImageArea::horizontalMirrorImageBot()

{

mAdditionalTools->mirrorImageBot(mImage->width(), mImage->height());

emit sendNewImageSize(mImage->size());

}

/\*ej1297\*/

**Additionaltools.h**

void mirrorImageTop(int width, int height); /\*ej1297\*/

void mirrorImageBot(int width, int height); /\*ej1297\*/

**Additionaltools.cpp**

/\*mirror Image implemented\*/

void AdditionalTools::mirrorImageTop(int width, int height)

{

QImage copy = mPImageArea->getImage()->copy(0, 0, width, height / 2);

mPImageArea->setImage(mPImageArea->getImage()->mirrored(false, true)); // This function is key, using (false,true) is a horizontal mirror, using (true,false) is a vertical mirror and using (true, true) is both kind of?

QImage result(width, height, QImage::Format\_RGB32);

QPainter painter;

painter.begin(&result);

painter.drawImage(0, 0, mPImageArea->getImage()->copy(0, 0, width, height));

painter.drawImage(0, 0, copy);

painter.end();

mPImageArea->setImage(result);

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

mPImageArea->update();

mPImageArea->setEdited(true);

mPImageArea->clearSelection();

}

void AdditionalTools::mirrorImageBot(int width, int height)

{

QImage copy = mPImageArea->getImage()->copy(0, 0, width, height);

mPImageArea->setImage(mPImageArea->getImage()->mirrored(false, true)); // This function is key, using (false,true) is a horizontal mirror, using (true,false) is a vertical mirror and using (true, true) is both kind of?

QImage result(width, height, QImage::Format\_RGB32);

QPainter painter;

painter.drawImage(0, 0, copy);

painter.begin(&result);

painter.drawImage(0, 0, mPImageArea->getImage()->copy(0, 0, width, height));

painter.end();

mPImageArea->setImage(result);

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

mPImageArea->update();

mPImageArea->setEdited(true);

mPImageArea->clearSelection();

mirrorImageTop(width, height);

}

/\*ej1297\*/