Mike Torchia

Dz7995

Easy Paint Change Request

3/3/2016

Group 1

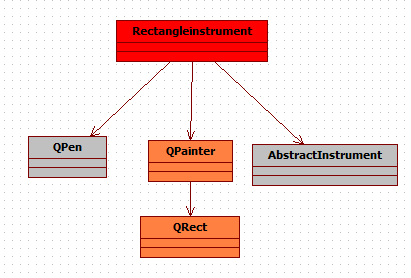
Change Request 1 – Rounded Rectangle

1. **Change Request and Concepts**
   1. Rounded Rectangle
      1. Change the tool that draws rectangles into a tool that draws rounded rectangles.
   2. Concepts
      1. Tool
      2. Draw(s)
      3. Rectangle(s)
      4. Rounded Rectangle(s)
2. **Concept Location**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Concept** | **Query** | **#Results** | **Target Class/File** | **Tool Used** | **Comments** |
| Draw | 1 | Matching Lines: 65  Matching Files: 15 | N/A | Grep | Found instrument in search. Redo search based on that |
| Instrument | 2 | Matching Lines: 790  Matching Files: 58 | N/A | Grep | Too many to inspect. Redo search |
| Rectangle | 3 | Matching Lines: 51  Matching Files: 11 | Rectangleinstrument.h | Grep | Found the rectangleinstrument header file |
| Rectangleinstrument | 4 | Matching Lines: 32  Matching Files: 5 | Rectangleinstrument.cpp | Grep | Found rectangleinstrument cpp file. Found the drawRect function. |
| drawRect | 5 | Matching Lines: 23  Matching Files: 6 | QPainter.h | Grep | Found all the draw functions for all of the shapes |
| Rounded | 6 | Matching Lines: 16  Matching Files: 7 | QPainter.h | Grep | Found drawRoundedRect function. Used to draw rounded rectangles |

1. **Impact Analysis**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | **Class Name** | **Tool Used** | **Mark** | **Comments** |
| 1 | Rectangleinsturment | Reading Source Code | Impacted | Concept located here |
| 2 | QPen | Reading Source Code | Unchanged |  |
| 3 | QPainter | Reading Source Code | Propagated | Contains the drawRoundedRect function that will be put into Rectangleinstrument class. |
| 4 | AbstractInstrument | Reading Source Code | Unchanged |  |
| 5 | QRect | Reading Source Code | Propagated | Contains the class and values that are to be pass into drawRoundedRect function |



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Code Files** | | | | | |
| Visited | Changed | Added | Propagating | Unchanged | Added to Changed Set |
| 3 | 1 | 0 | 0 | 2 | 1 |

1. **Actualization Summary**
2. **Actualization Code Files**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Code File Name** | **Task** | **Lines of Code** | | |
| **Added** | **Deleted** | **Total** |
| 1 | rectangleinstrument.cpp | Change code so tool draws rounded rectangles | 1 | 0 | 1 |
| 2 | qpainter.h | Contained solution function to be added to rectqangleinstrument.cpp | 0 | 0 | 0 |

1. **Verification**
   1. Saved changes to code
   2. Built and executed new source
   3. Expected results were rectangle tool that draws rectangles with rounded corners
   4. Achieved expected results after first change

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Code File Name** | **Coverage of Applications** | | | **Tests Failed** | **Bugs Found** |
| **Total Statements** | **Covered Statements** | **%** |
| **1** | **Rectangleinstrument.cpp** | **1** | **1** | **100** | **0** | **0** |

1. **Highlighted Source Code**

*Change located in rectangleinstrument.cpp*

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

{

QPainter painter(imageArea.getImage());

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

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

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

if(isSecondaryColor)

{

painter.setBrush(QBrush(DataSingleton::Instance()->getSecondaryColor()));

}

if(mStartPoint != mEndPoint)

{

// painter.drawRect(QRect(mStartPoint, mEndPoint)); //old function before change request

painter.drawRoundedRect(QRect(mStartPoint, mEndPoint), 10, 10); //new function after change request

}

imageArea.setEdited(true);

// int rad(DataSingleton::Instance()->getPenSize() + round(sqrt((mStartPoint.x() - mEndPoint.x()) \*

// (mStartPoint.x() - mEndPoint.x()) +

// (mStartPoint.y() - mEndPoint.y()) \*

// (mStartPoint.y() - mEndPoint.y()))));

// mPImageArea->update(QRect(mStartPoint, mEndPoint).normalized().adjusted(-rad, -rad, +rad, +rad));

painter.end();

imageArea.update();

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Concept** | **Query** | **# of Results** | **Target Class/File** | **Tool Used** | **Comments** |
| Color Tool Bar | 1 | 0 | N/A | Grep | Found Nothing. Concept too specific |
| Color | 2 | Matching Lines: 365  Matching Files: 45 | N/A | Grep | Too many results. Refine search |
| Toolbar | 3 | Matching Lines: 79  Matching File: 10 | Toolbar.h  Toolbar.cpp | Grep | Looks like it creates the toolbars and defines some signals. Found palettebar.h in search. Continue search |
| Palette | 4 | Matching Lines: 88  Matching Files:9 | Palettebar.h  Palettebar.cpp | Grep | Reading through source code found PalettButton functions. |
| PaletteButton | 5 | Matching Lines: 32  Matching Files: 31 | Palettebutton.cpp | Grep | Found function for handling mouse click events on buttons. Concept Found |