Name: Mike Torchia

Access ID: dz7995

Project: 2

Date 3/8/16

Group Number: 1

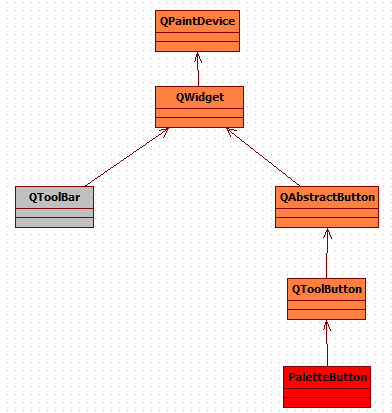
Change Request 2: Add Custom Colors

1. **Change Request and Concepts**: On the bottom of the application, there is a color tool bar for choosing default colors, add a new capability that a user can double click on the color to pick the new color from a color palette.
   1. Key Concepts:
      1. Color tool bar
      2. Color palette
      3. Palette
      4. Color

***(Concept Location table too big for page. On Next Page)***

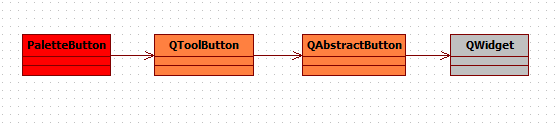
1. **Concept Location: Dependency Search**

|  |  |  |  |
| --- | --- | --- | --- |
| Class/File Name | Tool Used | Mark | Comments |
| QPaintDevice | Class Diagram | Propagated | Base Class. QWidget Next |
| QWidget | Class Diagram | Propagated | AbstractEffectSettings Next |
| AbstractEffectSettings | Class Diagram | Unchanged | Back to QWidget to check another neighbor. QToolBar Next |
| QToolBar | Class Diagram | Unchanged | Nothing found. Back to QWidget to check another neighbor. QAbstractButton next. |
| QAbstractButton | Class Diagram | Propagated | Found mousePressEvent used for ImageArea. The right concept but for the wrong class. QToolButton next |
| QToolButton | Class Diagram | Propagated | Found mousePressEvent definition. Function used for clicking on buttons on a QToolBar. Need to inspect next class. PaletteButton next |
| PaletteButton | Class Diagram | Located | Definition for mousePressEvent for the buttons on the palette |



1. **Impact Analysis**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Class Name** | **Tool Used** | **Mark** | **Comments** |
| 1 | PaletteButton | Reading Source Code | Impacted | Location here |
| 2 | QToolButton | Reading Source Code | Propagated | Propagating class containing mousePressEvent |
| 3 | QAbstractButton | Reading Source Code | Propagated | Propagating class containing mousePressEvent |
| 4 | QWidget | Reading Source Code | Unchanged |  |



1. **Actualization Summary**

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

1. **Actualization Code Files**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Code File Name** | **Task** | **Lines of Code** | | |
| **Added** | **Deleted** | **Total** |
| 1 | Palettebutton.h | Added function prototype for mouseDoubleClickEvent | 1 | 0 | 1 |
| 2 | Colorchooser.cpp | Extracted mousePressEvent function to be used in the mouseDoubleClickEvent function (same event happens in both) | 0 | 0 | 0 |
| 3 | Palettebutton.cpp | Used extracted function to define mouseDoubleClickEvent (both left and right click) | 25 | 0 | 25 |

1. **Verification**
   1. Saved changes to source codes
   2. Built and executed new source
   3. Expected results were to have color palette pop up upon double clicking colors on the bottom of the application (both left and right click)
   4. Achieved expected results after first change

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Code File Name** | **Coverage of Applications** | | | **Tests Failed** | **Bugs Found** |
| **Total Statements** | **Covered Statements** | **%** |
| 1 | Palettebutton.cpp | 2 | 2 | 100 | 0 | 0 |

1. **Highlighted Source Code**

*Changes located in palettebutton.cpp*

*Code below is all added code (new functionality)*

void PaletteButton::mouseDoubleClickEvent(QMouseEvent \*event)

{

//if left button double clicked

if (event->button() == Qt::LeftButton)

{

/\*color is equal to the color that is choosen from the pop up dialog\*/

QColor color = QColorDialog::getColor(mColor, this);

/\*if the color is valid. set the primary color to the newly picked color and then emit the signal\*/

if (color.isValid())

{

DataSingleton::Instance()->setPrimaryColor(color);

emit colorPicked();

}

}

/\*Repeat the same process for the right click\*/

else if (event->button() == Qt::RightButton)

{

QColor color = QColorDialog::getColor(mColor, this);

if (color.isValid())

{

DataSingleton::Instance()->setSecondaryColor(color);

emit colorPicked();

}

}

}

*Changes located in palettebutton.h*

#ifndef QPALETTEBUTTON\_H

#define QPALETTEBUTTON\_H

#include "../datasingleton.h"

#include <QToolButton>

class PaletteButton : public QToolButton

{

Q\_OBJECT

public:

PaletteButton(const QColor &color);

signals:

void colorPicked();

private:

QColor mColor;

protected:

void mousePressEvent(QMouseEvent \*event);

void mouseDoubleClickEvent(QMouseEvent \*event);

};

#endif // QPALETTEBUTTON\_H