Специальность: 09.02.07 Информационные системы и программирование

Проверил:

\_\_\_\_\_\_\_\_\_\_\_ Антонов М.Б.

Дата: «» \_\_\_\_\_ 2020 г.

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

(Университет ИТМО)

факультет среднего профессионального образования

Отчет

о Лабораторной работе №16

по теме: Рисование в Qt

по дисциплине: Разработка программных модулей

Выполнил:

студент группы Y2433

\_\_\_\_\_\_\_\_\_\_\_ Сафин Р.М.

Санкт-Петербург 2020

Содержание

[1 Постановка задачи 3](#_Toc35698558)

[1.1 Цель 3](#_Toc35698559)

[1.2 Задачи 3](#_Toc35698560)

[1.3 Задание 3](#_Toc35698561)

[2 Диаграммы 4](#_Toc35698562)

[3 Исходный код](#_Toc35698563) 5

[4 ЛИСТИНГ ВЫполнения программы 9](#_Toc35698564)

[5 Вывод 1](#_Toc35698565)0

1. Постановка задачи
   1. Цель

Познакомиться с основами работы графических сцен и основы рисования в Qt.

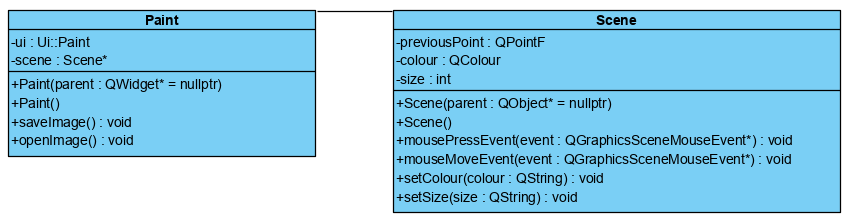
* 1. Задачи
* изучить основы работы с QPaint;
* создать простейшее приложение для рисования.
  1. Задание

Требуется создать приложение для рисования по типу кисти Paint мышкой.

Требуется создать виджет для выбора текущего цвета рисования.

Результат рисования можно сохранять в .bmp файл или любой другой формат изображения, используя QFileDialog.

Можно открыть png изображение и отредактировать его.

1. Диаграммы

*Рисунок 1 – Диаграмма классов*

1. Исходный код

Листинг 1 — Код файла main.cpp

#include "paint.h"

#include <QApplication>

int main(int argc, char \*argv[])

{

QApplication a(argc, argv);

Paint w;

w.show();

return a.exec();

}

*Листинг 2 – Код файла paint.cpp*

/\*\*

\* @file paint.cpp

\* @brief painting window

\* @author Roman Safin

\*/

#include "paint.h"

#include "ui\_paint.h"

#include <QMessageBox>

Paint::**Paint**(QWidget \*parent) : QWidget(parent), ui(new Ui::Paint)

{

ui->setupUi(this);

scene = new Scene();

ui->graphicsView->setScene(scene);

QCursor cursor = QCursor(QPixmap(":/Brush.png"));

ui->graphicsView->setCursor(cursor);

connect(ui->color, &QComboBox::currentTextChanged, scene, &Scene::setColor);

connect(ui->size, &QLineEdit::textChanged, scene, &Scene::setSize);

connect(ui->save, &QPushButton::clicked, this, &Paint::saveImage);

connect(ui->open, &QPushButton::clicked, this, &Paint::openImage);

}

Paint::~***Paint***()

{

delete ui;

}

void Paint::**saveImage**()

{

QString fileName = QFileDialog::getSaveFileName

(this, tr("Save file"),"",

tr("Images(\* .png \* .jpg)"));

QImage image(ui->graphicsView->scene()->width(),

ui->graphicsView->scene()->height(),

QImage::Format\_RGB32);

image.fill(QColor(Qt::white).rgb());

QPainter painter(&image);

ui->graphicsView->scene()->render(&painter);

painter.end();

if(fileName.indexOf(".png") != -1 || fileName.indexOf(".jpg") != -1)

image.save(fileName);

else

image.save(fileName.append(".png"));

}

void Paint::**openImage**()

{

QString fileName = QFileDialog::getOpenFileName

(this, tr("Open file"),"",

tr("Images(\* .png \* .jpg)"));

Scene\* sceneImage = new Scene;

sceneImage->addPixmap(QPixmap(fileName));

ui->graphicsView->setScene(sceneImage);

sceneImage->setSize(ui->size->displayText());

sceneImage->setColor(ui->color->currentText());

connect(ui->color, &QComboBox::currentTextChanged,

sceneImage, &Scene::setColor);

connect(ui->size, &QLineEdit::textChanged,

sceneImage, &Scene::setSize);

}

Листинг 3 – Код файла scene.cpp

/\*\*

\* @file scene.h

\* @brief painting scene

\* @author Roman Safin

\*/

#include "scene.h"

Scene::**Scene**(QObject \*parent) : QGraphicsScene(parent)

{

}

Scene::~***Scene***()

{

}

void Scene::***mousePressEvent***(QGraphicsSceneMouseEvent \*event)

{

addEllipse(event->scenePos().x()-(size/2), event->scenePos().y()-(size/2),

size, size, QPen(Qt::NoBrush), QBrush(colour));

previousPoint = event->scenePos();

}

void Scene::***mouseMoveEvent***(QGraphicsSceneMouseEvent \*event)

{

addLine(previousPoint.x(), previousPoint.y(),

event->scenePos().x(), event->scenePos().y(),

QPen(colour, size,Qt::SolidLine,Qt::RoundCap));

previousPoint = event->scenePos();

}

void Scene::**setColor**(QString colour)

{

this->colour = colour;

}

void Scene::**setSize**(QString size)

{

this->size = size.toInt();

}

Листинг 4 – Код файла scene.h

/\*\*

\* @file scene.h

\* @brief painting scene

\* @author Roman Safin

\*/

#ifndef SCENE\_H

#define SCENE\_H

#include <QGraphicsScene>

#include <QGraphicsSceneMouseEvent>

class **Scene** : public QGraphicsScene

{

Q\_OBJECT

private:

/\*\*

\* @brief previous point coordinates

\*/

QPointF previousPoint;

/\*\*

\* @brief brush colour

\*/

QColor colour;

/\*\*

\* @brief brush size

\*/

int size;

public:

explicit **Scene**(QObject \*parent = nullptr);

~***Scene***();

private:

/\*\*

\* @brief mouse button pressed

\* @param event

\*/

void ***mousePressEvent***(QGraphicsSceneMouseEvent \*event);

/\*\*

\* @brief mouse moved

\* @param event

\*/

void ***mouseMoveEvent***(QGraphicsSceneMouseEvent \*event);

public slots:

/\*\*

\* @brief set colour

\* @param colour

\*/

void **setColor**(QString colour);

/\*\*

\* @brief set brush size

\* @param size

\*/

void **setSize** (QString size);

};

#endif // SCENE\_H

Листинг 5 – Код файла paint.h

/\*\*

\* @file paint.h

\* @brief painting window

\* @author Roman Safin

\*/

#ifndef PAINT\_H

#define PAINT\_H

#include <QWidget>

#include <QFileDialog>

#include <scene.h>

QT\_BEGIN\_NAMESPACE

namespace **Ui** { class **Paint**; }

QT\_END\_NAMESPACE

class **Paint** : public QWidget

{

Q\_OBJECT

private:

Ui::Paint \*ui;

Scene\* scene;

public:

**Paint**(QWidget \*parent = nullptr);

~***Paint***();

private slots:

/\*\*

\* @brief save image

\*/

void **saveImage**();

/\*\*

\* @brief open image

\*/

void **openImage**();

};

#endif // PAINT\_H

Листинг 6 – Код файла res.qrc

<RCC>

<qresource prefix="/">

<file>brush.png</file>

</qresource>

</RCC>

1. ЛИСТИНГ ВЫполнения программы

На рисунке 2 представлен интерфейс работы программы.



Рисунок 2 – Интерфейс программы



Рисунок 3 — Открытие сохранённого изображения в программе

1. Вывод

В ходе выполнения лабораторной работы была простейшая программа для рисования мышью с возможностью открытия и сохранения файлов.