**Отчет по проекту**

**Bitmap editor**

Ларов Даниил

5103, МФТИ

**Общая задача:**

Создание простейшего растрового редактора с возможностью редактирования.

**Используемый язык:**

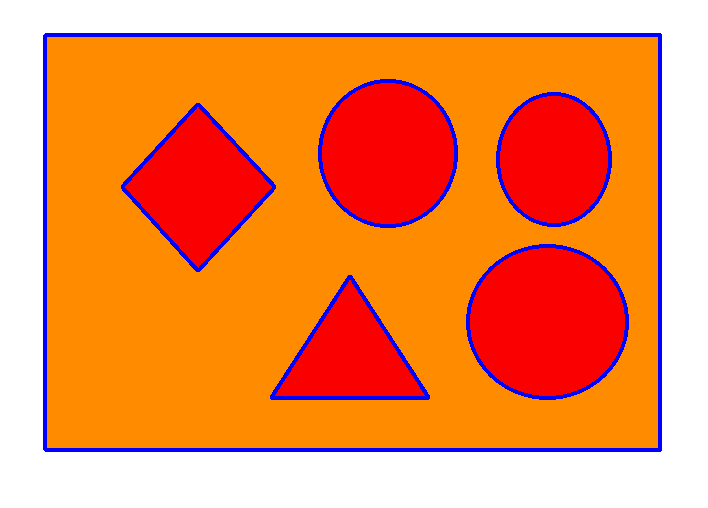
Стандартная библиотека С++ и Qt.

**Входные данные:**

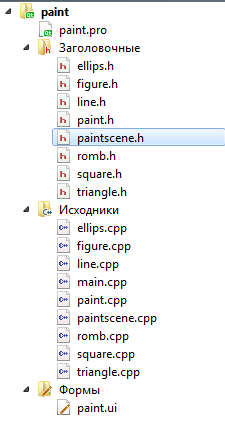
Отсутствуют или изображение формата png или png;

**Выходные данные:**

Файл – изображение формата .jpeg или .png



Структура проекта:



В структуру проекта входят следующие файлы:

* paint.h - заголовочный файл виджета, в котором будет располагаться графическая сцена для рисования;
* paint.cpp - соответственно файл исходных кодов для этого виджета;
* paintscene.h - заголовочный файл кастомизированной графической сцены
* paintscene.cpp - файл исходных кодов кастомизированной графической сцены.
* figure.h - заголовочный файл базового класса фигур;
* figure.cpp - файл исходных кодов класса фигур;
* romb.h - заголовочный файл класса Ромб;
* romb.cpp - файл исходных кодов класса Ромб;
* square.h - заголовочный файл класса Прямоугольник;
* square.cpp - файл исходных кодов класса Прямоугольник;
* triangle.h - заголовочный файл класса Треугольник;
* triangle.cpp - файл исходных кодов класса Треугольник;
* elipse.h - заголовочный файл класса Эллипс;
* elips.cpp - файл исходных кодов класса Эллипс;
* line.h - заголовочный файл класса Прямая линия;
* line.cpp - файл исходных кодов класса Прямая линия;
* paint.ui - форма главного окна приложения;

Описание функций виджетов

paint.h

В данном файле объявляется кастомизированная графическая сцена, а также таймер со слотом для этого таймера, который служит для корректной обработки изменения размеров окна приложения. Кроме того, здесь мы определяем клавиши для нажатия а также функции, которые будут вызываться при их нажатии.

paint.cpp

В данном классе происходит добавление кастомизированной графической сцены в объект класса QGraphicsView. Также здесь происходит добавление кнопок, полей заполнения и пр., подробно описывается поведение функций при их вызове.

Само рисование происходит исключительно в кастомизированной графической сцене.

paintscene.h

Описание того, что и как именно будет рисоваться определяется здесь. За то, какой объект будет рисовать пользователь отвечает переменная **Style.**

**Style:**

**0 – При нажатии на сцену и проведении мыши добавляется последовательность эллипсов (линия).**

**1 – При нажатии на сцену будет происходить закраска области**

**2 – Добавление на сцену различных объектов. Нажатие на сцену устанавливает начальную точку, а проведение мыши и отпускание кнопки даёт конечную точку, что меняет размер фигуры.**

Также здесь описана функция filling(Функция закраски области), перечисляется типы используемых фигур.

Q\_PROPERTY, которое отвечает за хранение текущего типа фигуры для отрисовки. Список типов задаётся с помощью enum FigureTypes. Также присутствует объявление объекта tempFigure , который является экземпляром класса Figure . Данный класс является базовым в соответствии с парадигмой полиморфизма, поэтому он используется для временного хранения фигуры, которая отрисовывается на графической сцене, поскольку все основные методы для задания размеров фигуры с помощью мыши реализованы в базовом классе Figure.

Также переопределены методы событий от мыши. В методе mousePressEvent создаётся геометрическая фигура для отрисовки, а в методе mouseMoveEvent производится перерисовка фигуры в зависимости от положения курсора мыши до тех пор, пока кнопка мыши не будет отпущена.

paintscene.cpp

В данном файле вся работа с рисованием происходит в методах mouseMoveEvent и mousePressEvent. Также здесь описывается функция закраски области. Подробное идея и алгоритм описывается далее.

Описание классов фигур

Figure унаследован от QGraphicsItem , и там реализован весь основной функционал в данном классе. Классы romb, square, triangle, elipse, line унаследованы от него, а в них реализован только метод отрисовки самой фигуры: Romb, Square, Triangle, Elipse, Line.

figure.h

Заголовочный файл базового класса для создания геометрических фигур. Он предоставляет основной функционал для определения границ отрисовки геометрической фигуры вне зависимости от того, какая фигура будет отрисовываться.

Для отрисовки используются две точки: стартовая и конечная. Отрисовка начинается из стартовой точки и она не меняется своего местоположения, а вот конечная точка является координатой противоположного угла прямоугольной области в которой отрисовывается фигура. В зависимости от этих точек перерисовываются и все остальные ключевые точки фигур, а соответственно и границы этих фигур.

romb.h

Он наследуется от базового класса Figure и переопределяет метод paint(), который наследован от предка класса Figure, то есть от QGraphicsItem. В этом методе будет реализована логика отрисовки данной геометрической фигуры.

Square.h, Triangle.h, Elipse.h и Line.h определяются аналогично.

Описание Алгоритма закраски

**Общая идея алгоритма:**

Имеем контур, который необходимо закрасить, начальную позицию (x,y), которую выбрал пользователь а также цвет, который мы закрашиваем и цвет, которым закрашиваем.

**Описание**

1) Проходим по позиции y вниз(++) и вверх (--) до тех пор пока цвет на котором мы стоим совпадает с тем который нужно закрасить.

1. Сначала при перемещении компоненты x определяется ширина линии позиции y для последующего закраски линии на этой позиции.
2. Запоминается крайняя левая и крайняя правая позиция.
3. Идет сравнение этих крайних левых и правых позиций с теми, которые были на позиции y-1. Если разница между ними существенна, то запоминаются обе позиции как для правых так и левых крайних позиций в массиве данный **reml** и **remr.**
4. На сцену добавляется линия на этой широте y.

2) В цикле рекурсивно вызываем следующую функцию, причем передаваемая позиция x – среднее арифметическое от запомненных в **reml** и **remr.** Также вызываем последнюю позицию y, но с измененной компонентой x (равна среднему арифметическому от последних крайних левых и правых позиций x).

Рекурсивный вызов происходит либо до тех пор пока все разрывы не будут вызваны либо до тех пор, пока не будет вызвано слишком много рекурсий (проверяется переменной deep и сравнивается с maxdeep).