# Практические упражнения по Основам создания UI

Цель упражнения: получить навыки в описании элементов QML, управлении их расположением и обработке сигналов.

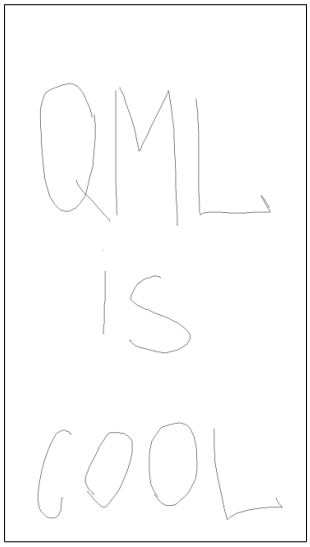
Задача: расширить функционал примера простой рисовалки, добавив возможность выбора цветов и толщины линий.

## Основные задания

1. Сделать предварительную настройку проекта:
    1. Скачать архив шаблона проекта qml-basics для Sailfish OS SDK (альтернативно можно для Qt SDK).
    2. Распаковать скачанный архив в домашнюю директорию или альтернативную директорию для проектов.
    3. Из Sailfish OS IDE открыть qml-basics.pro.
    4. Настроить проект, выбрав комплект для i486.
    5. В настройках запуска проекта отметить «Enable receiving updates with Qt QmlLive».
    6. Настроить запуск в режиме отладки с помощью сборки RPM-пакета.
    7. Проверить, что проект собирается и запускается в эмуляторе.
2. В qml-basics.qml добавить код простой рисовалки из слайдов, не забыть про

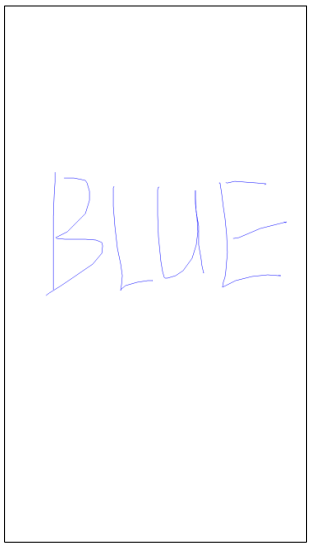
    ```
    import "painter.js" as Painter
    ```

    Проверить, что линии рисуются.




3. В корневой Item добавить свойство paintColor типа string, привязанное к значению "blue".
4. В обработчике сигнала pressed элемента MouseArea при создании новой линии (*lines*.push) заменить цвет, который указывается в поле stroke, на paintColor.

    Проверить, что линии рисуются синим цветом.




5. Сделать кнопку круглой.

6. Выделить код холста, на котором рисуются линии, в отдельный файл LinesCanvas.qml:

```
import QtQuick 2.6
import "painter.js" as Painter

Canvas {
    property var lines: []

    onPaint: {
        var ctx = getContext("2d");
        ctx.clearRect(region.x, region.y, region.width, region.height);
        Painter.paintLines(ctx, lines);
    }
}
```

7. В qml-basics.qml заменить Canvas на LinesCanvas, убрать не нужные больше импорты и свойства.

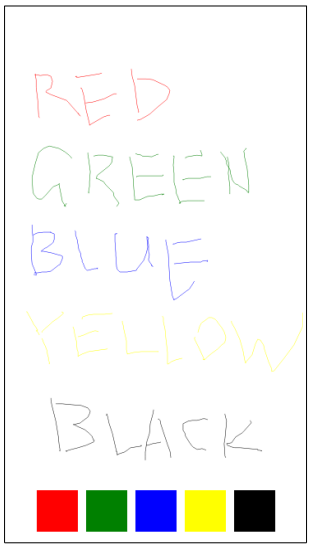
   Пересобрать и запустить проект, проверить, что линии рисуются.

8. Добавить в папку qml элемент цветной кнопки ColorButton.qml:

```
import QtQuick 2.6

MouseArea {
    property alias color: rectangle.color

    width: 100; height: 100

    Rectangle {
        id: rectangle
        anchors.fill: parent
    }
}
```


   Изучить этот код и понять, что он делает.

9. В qml-basics.qml добавить красную кнопку, разместив её с помощью свойств anchors по центру горизонтали корневого Item на расстоянии 20 пикселов от его нижней границы.




10. Назначить обработчик сигнала clicked добавленной кнопки, который присваивает paintColor значение "red".

    Пересобрать и запустить проект, проверить, что линии рисуются красным цветом.

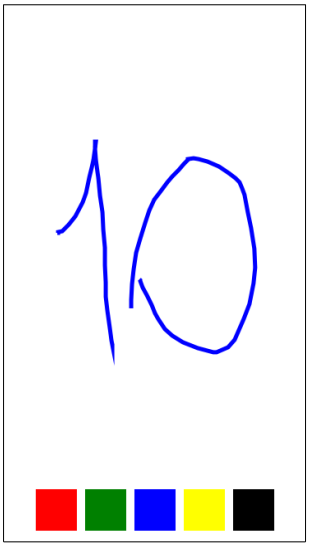
11. Добавить кнопки зелёного, синего, жёлтого и чёрного цветов. Поместить все кнопки в элемент

типа Row, расположенный с помощью свойств anchors по центру горизонтали корневого Item на расстоянии 20 пикселов от его нижней границы.

12. Для каждой кнопки назначить обработчик сигнала clicked, который присваивает paintColor соответствующий цвет.
13. Реализовать вставку пяти однотипных кнопок с помощью элемента типа Repeater, у которого свойство model описано с помощью ListModel:

```
ListModel {
    ListElement { color: "red" }
    ListElement { color: "green" }
    ListElement { color: "blue" }
    ListElement { color: "yellow" }
    ListElement { color: "black" }
}
```

При таком подходе в описании свойства delegate значение соответствующего цвета можно получить через model.color.



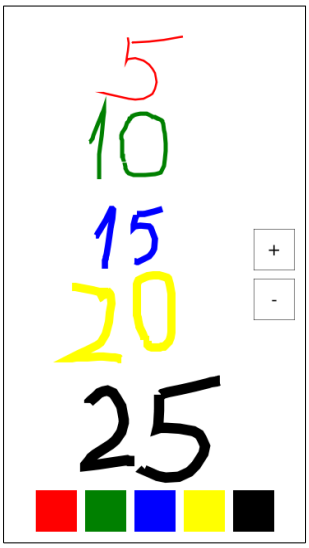
14. Для кнопок добавить обработчик сигнала pressAndHold, который удаляет все нарисованные линии (присвоить пустой массив свойству lines). Важно не забыть вызвать перерисовку.
15. При создании новой линии (*lines*.push) указать ширину (width: 10).

    Проверить, что линии стали шире.

## Дополнительные задания

16. В корневой Item добавить свойство paintWidth, определяющее толщину линий.
17. Создать новый элемент TextButton, реализующий кнопку с текстом.
18. Добавить кнопки со знаками «+» и «-», которые позволяют изменять свойство paintWidth, соответственно, увеличивая или уменьшая его с шагом 5.

    Проверить, что можно изменять ширину линий.



19. Заменить обработчик сигнала pressAndHold цветных кнопок так, чтобы он удалял только линии цвета кнопки. Для этого будет полезен метод filter().
20. Добавить второй холст, на котором будет рисоваться новая линия, и переноситься на основной только по сигналу released элемента типа MouseArea. Это нужно для существенного ускорения отрисовки при большом количестве уже нарисованных линий.