Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего профессионального образования

«Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского»

**Институт информационных технологий, математики и механики**

**Кафедра программной инженерии**

**Практическая работа №2**

**«Основы конструирования интерфейсов пользователя»**

**Выполнил: студент группы 381808-02**

**Шульман Е. А.**

**Проверил: Карчков Д. А.**

**Нижний Новгород**

**2021**

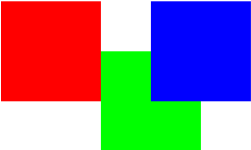
**Цель работы**

Освоить базовые навыки построения пользовательских интерфейсов, позиционирования, отрисовки и перемещения элементов. Научиться анимировать элементы. Научиться создавать диалоги и взаимодействовать с ними.

**Шаги**:

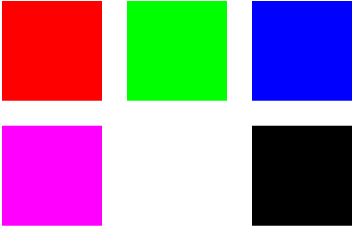
1) Создать новый проект со стандартной заготовкой приложения.

2) Нарисовать 3 квадрата красного, зелёного и синего цветов следующим образом:



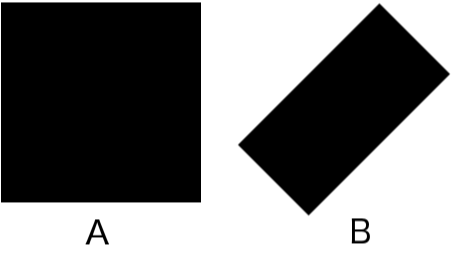
3) Поместить текст “Квадрат” белого цвета по центру синего квадрата.

4) Нарисовать 5 квадратов с использованием Column и Row следующим образом:



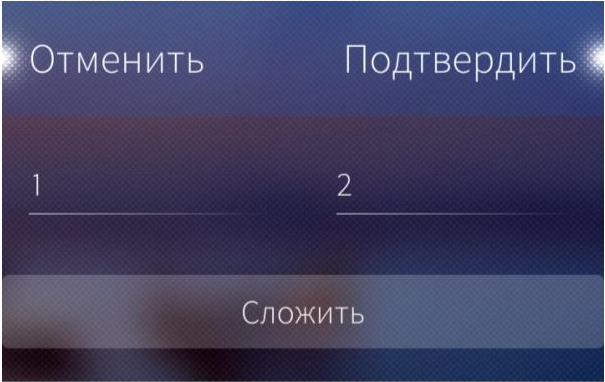
5) Нарисовать те же 5 квадратов с использованием Grid.

6) Сделать из квадрата “A” прямоугольник “B” с использованием объектов Translate, Scale и Rotation.

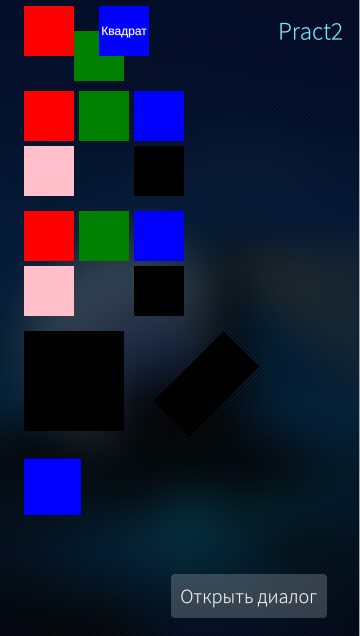
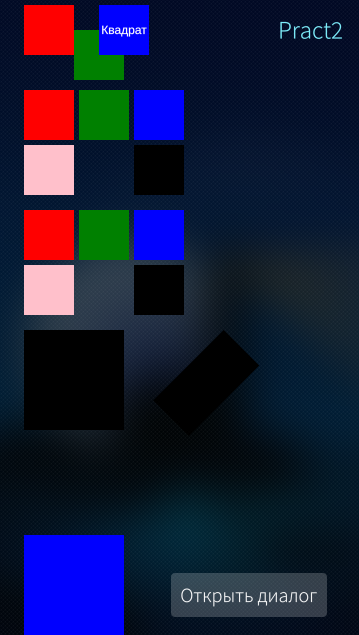


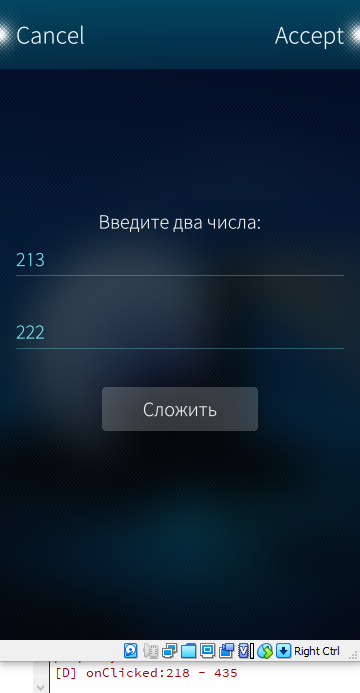
7) Нарисовать квадрат и анимировать его перемещение вниз с увеличением его размера. Документация по анимации доступна по адресу http://doc.qt.io/qt-5/qml-qtquick-animation.html.

8) Реализовать диалог с двумя текстовыми полями, в которые вводятся числа. После нажатия на кнопку “Подтвердить” в консоль выводится сумма чисел. Для преобразования строк к числам использовать функцию parseInt(“42”). Валидацией и обработкой ошибок можно пренебречь.



**Демонстрация работы**

** **

****

**Заключение**

В ходе практической работы было реализовано приложение, содержащее все описанные шаги. Подробно были разобраны следующие элементы: Rectangle, Text и их свойства. Были рассмотрены способы позиционирования элементов на экране. Познакомились с различными контейнерами элементов. Рассмотрели Translate, Scale и Rotation. Научились анимировать перемещение элементов. А также научились использовать диалоги для взаимодействия с пользователем.

**Приложения**

1. Код FirstPage.qml:

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

id: *page*

allowedOrientations: Orientation.All

SilicaFlickable {

anchors.fill: *parent*

PullDownMenu {

MenuItem {

text: *qsTr*("Show Page 2")

onClicked: *pageStack*.push(*Qt*.resolvedUrl("SecondPage.qml"))

}

}

contentHeight: *column*.height

Column {

id: *column*

width: *page*.width

spacing: Theme.paddingLarge

PageHeader {

title: *qsTr*("Pract2")

}

Item {

x: 50

y: 20

Rectangle {

width: 100

height: 100

color: "red"

}

Rectangle {

x: 100

y: 50

width: 100

height: 100

color: "green"

}

Rectangle {

x: 150

width: 100

height: 100

color: "blue"

Text {

text: "Квадрат"

font.family: "Helvetica"

font.pointSize: 18

color: "white"

anchors.centerIn: *parent*

}

}

}

Item {

x: 50

y: 190

Row {

spacing: 10

Column {

spacing: 10

Rectangle {

width: 100

height: 100

color: "red"

}

Rectangle {

width: 100

height: 100

color: "pink"

}

}

Rectangle {

width: 100

height: 100

color: "green"

}

Column {

spacing: 10

Rectangle {

width: 100

height: 100

color: "blue"

}

Rectangle {

width: 100

height: 100

color: "black"

}

}

}

}

Item {

x: 50

y: 430

Grid {

rows: 2

columns: 3

spacing: 10

Rectangle {

width: 100

height: 100

color: "red"

}

Rectangle {

width: 100

height: 100

color: "green"

}

Rectangle {

width: 100

height: 100

color: "blue"

}

Rectangle {

width: 100

height: 100

color: "pink"

}

Rectangle {

width: 100

height: 100

color: "transparent"

}

Rectangle {

width: 100

height: 100

color: "black"

}

}

}

Item {

x: 50

y: 670

Rectangle {

width: 200

height: 200

color: "black"

}

Rectangle {

width: 200

height: 200

color: "black"

transform: [

Scale {

xScale: 0.5

yScale: 1

},

Rotation {

angle: 45

},

Translate {

x: + 400

}]

}

}

Item {

x: 50

y: 900

Rectangle {

id: *animationRectangle*

color: "blue"

width: 100

height: 100

ParallelAnimation {

running: true

NumberAnimation {

target: *animationRectangle*

property: "y"

to: 180

duration: 10000

}

NumberAnimation {

target: *animationRectangle*

property: "width"

to: 200

duration: 10000

}

NumberAnimation {

target: *animationRectangle*

property: "height"

to: 200

duration: 10000

}

}

}

}

Component {

id: *dialog*

Dialog {

DialogHeader{}

Column {

anchors.centerIn: *parent*

width: *parent*.width

spacing: Theme.paddingMedium

Label {

anchors.horizontalCenter: *parent*.horizontalCenter

text: "Введите два числа:"

}

TextField {

id: *field1*

width: *parent*.width

inputMethodHints: *Qt*.ImhFormattedNumbersOnly

placeholderText: *label*

color: Theme.highlightColor

}

TextField {

id: *field2*

width: *parent*.width

inputMethodHints: *Qt*.ImhFormattedNumbersOnly

placeholderText: *label*

color: Theme.highlightColor

}

Button {

anchors.horizontalCenter: *parent*.horizontalCenter

text: "Сложить"

onClicked: *console*.log(parseInt(*field1*.text) + parseInt(*field2*.text));

}

}

onAccepted: *console*.log(parseInt(*field1*.text) + parseInt(*field2*.text));

onRejected: helloLabel.text = "Отменил действие"

}

}

Item {

x: 500

y: 1200

Button {

text: "Открыть диалог"

anchors.centerIn: *parent*

onClicked: {

onClicked: *pageStack*.push(*dialog*)

}

}

}

}

}

}

1. Ссылка на проект: https://github.com/egorshul/QtProject/tree/main/pract2