Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Московский государственный технический университет имени Н.Э.Баумана

(национальный исследовательский университет)»

(МГТУ им. Н.Э. Баумана)

Факультет ИУ – «Информатика и системы управления»

Кафедра ИУ-3 – «Информационные системы и телекоммуникации»

Отчет по лабораторной работе №1

по дисциплине «Разработка мультисервисных приложений для мобильных операционных систем»

Студент группы ИУ3-81Б О.С. Курзина

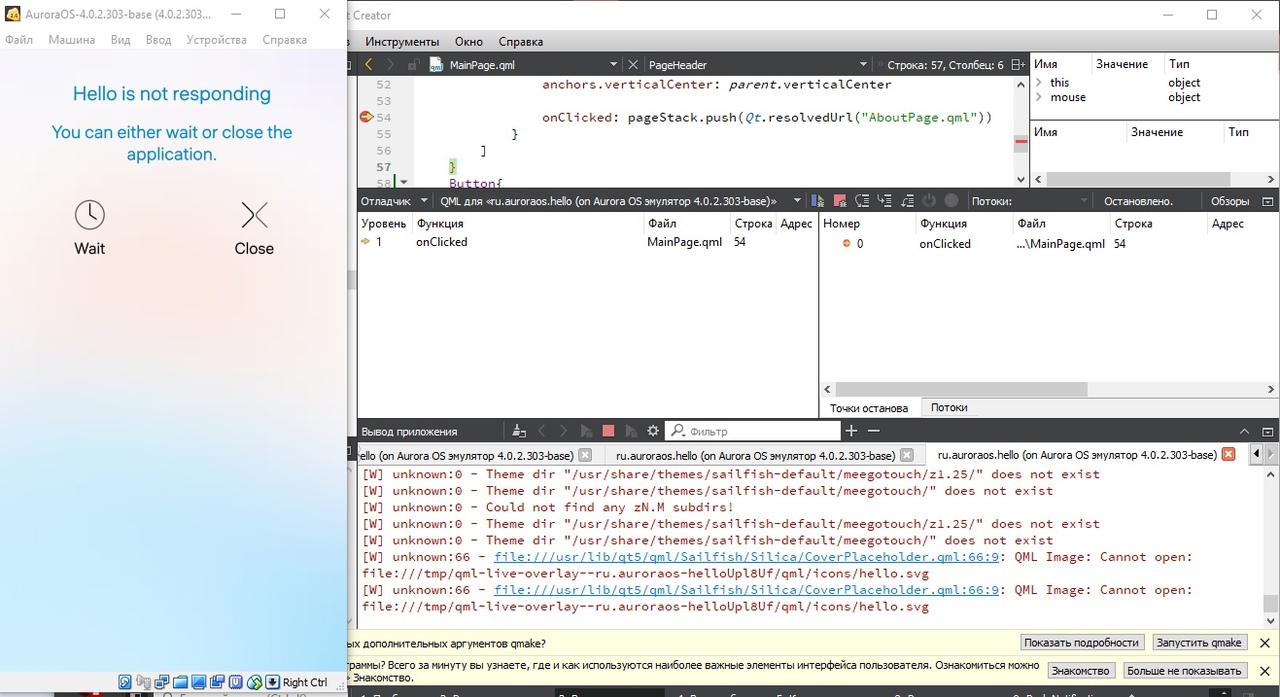
Преподаватель Д.А. Видьманов

Москва, 2024 г.

1. **ЦЕЛЬ**

Ознакомление с ОС Аврора. Установление нужных компонентов. Запуск первого приложения.

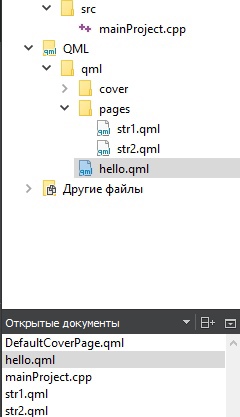
1. **ЗАДАНИЕ**
2. Отладка и профилирование
3. Задать точки останова, запустить приложение в режиме отладки



1. Изменить размеры окна приложения

https://sun9-54.userapi.com/impg/l-LlqBkVx6VEC-30c5vnoAkDYDcWZzGME72Okw/0tUoE-hEKbY.jpg?size=369x33&quality=96&sign=41ac8b231056eaf94f0670c451a74647&type=album

1. Переименовать основные .cpp и .qml- файлы



1. Напишите приложение, которое выводит текст в консоль после нажатия на кнопку (Button)

Код:

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

objectName: "task1Page"

allowedOrientations: Orientation.All

PageHeader {

objectName: "pageHeader"

title: qsTr("Task1")

extraContent.children: [

Column{

Button {

text: "Task2"

onClicked: pageStack.push(Qt.resolvedUrl("Task2.qml"))

}

Button {

text: "Task3"

onClicked: pageStack.push(Qt.resolvedUrl("Task3.qml"))

}

}

]

}

Rectangle {

anchors.centerIn: parent

id: printRect

width: 400

height: 400

color: "lightgrey"

Button {

text: "Print to console"

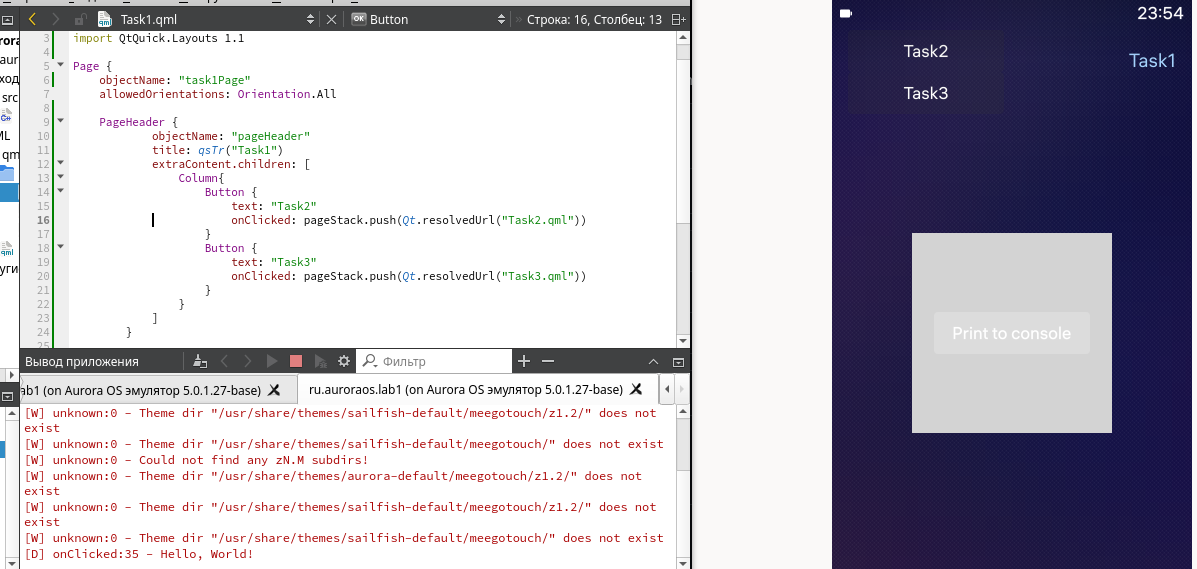
anchors.centerIn: parent

onClicked: console.log("Hello, World!")

}

}

}



1. Создайте проект, у которого в главном окне содержится три кнопки, к каждой из которых подключен слот, а уже эти слоты передают сигнал в один единый слот с номером нажатой кнопки.

Код:

Page {

objectName: "task2Page"

allowedOrientations: Orientation.All

PageHeader {

objectName: "pageHeader"

title: qsTr("Task2")

extraContent.children: [

Column{

Button {

text: "Task1"

onClicked: pageStack.push(Qt.resolvedUrl("Task1.qml"))

}

Button {

text: "Task3"

onClicked: pageStack.push(Qt.resolvedUrl("Task3.qml"))

}

}

]

}

Rectangle {

id: root

width: 400

height: 400

anchors.centerIn: parent

color: "lightgrey"

signal buttonClicked(int buttonNumber)

Button {

text: "Button 1"

anchors.centerIn: parent

onClicked: {

root.buttonClicked(1)

}

}

Button {

text: "Button 2"

anchors.bottom: parent.bottom

anchors.horizontalCenter: parent.horizontalCenter

onClicked: {

root.buttonClicked(2)

}

}

Button {

text: "Button 3"

anchors.top: parent.top

anchors.horizontalCenter: parent.horizontalCenter

onClicked: {

root.buttonClicked(3)

}

}

Connections {

target: root

onButtonClicked: {

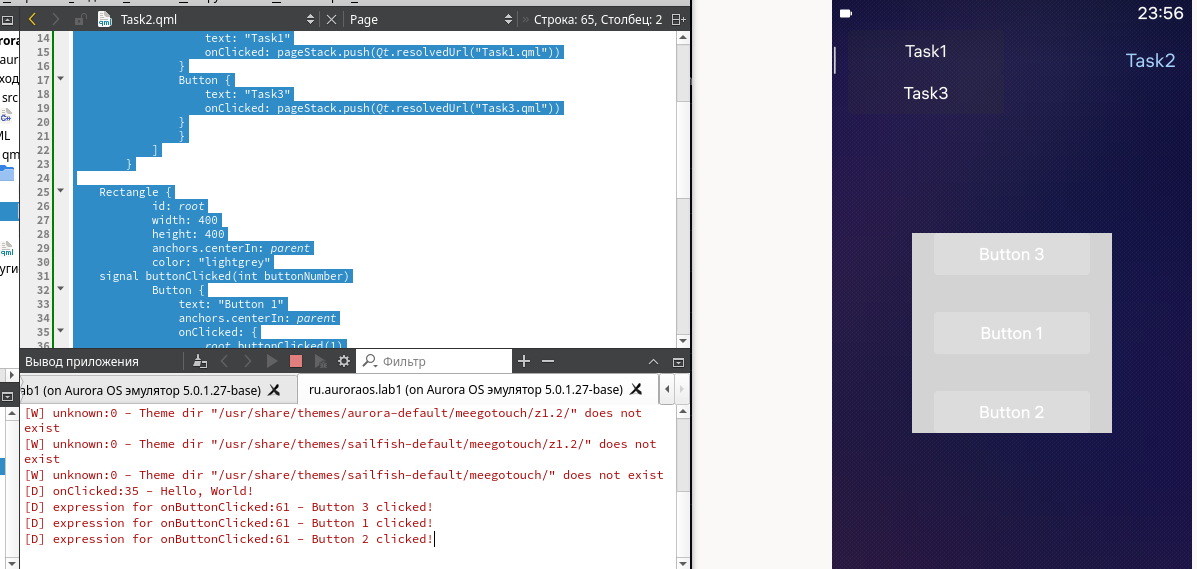
console.log("Button", buttonNumber, "clicked!")

}

}

}

}



4. Реализовать компонент кнопки, который предоставляет сигналы: clicked, doubleClicked, pressed, released

Код:

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

objectName: "mainPage"

allowedOrientations: Orientation.All

PageHeader {

objectName: "pageHeader"

title: *qsTr*("Laba1")

extraContent.children: [

IconButton {

objectName: "aboutButton"

icon.source: "image://theme/icon-m-about"

anchors.verticalCenter: *parent*.verticalCenter

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

}

]

}

Button {

signal doubleClicked()

signal pressed()

signal released()

anchors.centerIn: *parent*

MouseArea {

anchors.fill: *parent*

onPressed: {

*parent*.pressed(*console*.log("pressed"));

}

onReleased: {

*parent*.released(*console*.log("released"))

}

onDoubleClicked: {

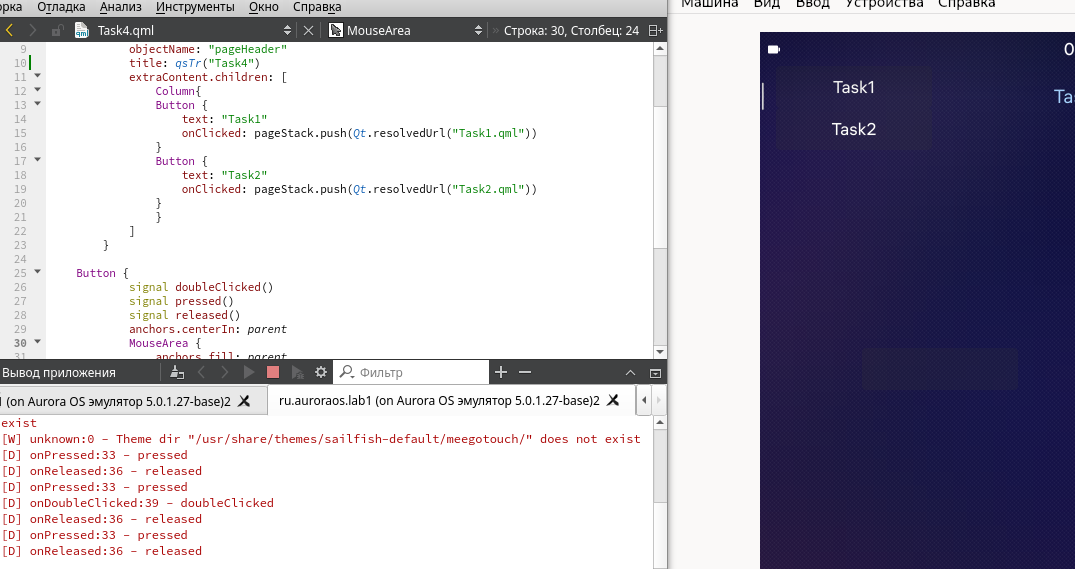
*parent*.doubleClicked(*console*.log("doubleClicked"))

}

}

}

}



5. Написать приложение, которое позволяет перетаскивать, вращать и масштабировать изображение двумя пальцами.

Код:

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

objectName: "task3Page"

allowedOrientations: Orientation.All

PageHeader {

objectName: "pageHeader"

title: qsTr("Task3")

extraContent.children: [

Column{

Button {

text: "Task1"

onClicked: pageStack.push(Qt.resolvedUrl("Task1.qml"))

}

Button {

text: "Task2"

onClicked: pageStack.push(Qt.resolvedUrl("Task2.qml"))

}

}

]

}

PinchArea {

id: pinchArea

property real \_statTargetScale

property real \_statTargetRotation

property real \_statRadiusX

property real \_statRadiusY

function initiateTransformation(pinch){

\_statTargetScale = target.scale

\_statTargetRotation = target.rotation

\_statRadiusX = target.x - pinch.center.x

\_statRadiusY = target.y - pinch.center.y

}

function updateTransformation(pinch){

var rotationSin = Math.sin(pinch.rotation + Math.PI/180)

var rotationCos = Math.cos(pinch.rotation + Math.PI/180)

var radiusX = pinch.scale\*(\_statRadiusX\*rotationCos - \_statRadiusY\*rotationSin)

var radiusY = pinch.scale\*(\_statRadiusX\*rotationSin - \_statRadiusY\*rotationCos)

target.x =pinch.center.x + radiusX

target.y =pinch.center.y + radiusY

target.scale = \_statTargetScale + pinch.scale

target.rotation = \_statTargetRotation + pinch.rotation

}

anchors.fill: parent

onPinchStarted: initiateTransformation(pinch)

onPinchUpdated: updateTransformation(pinch)

onPinchFinished: updateTransformation(pinch)

}

Flickable{

anchors.centerIn: parent

width: 200

height: 200

contentWidth: target.width

contentHeight: target.height

Image {

id: target

source: "img/cat.png"

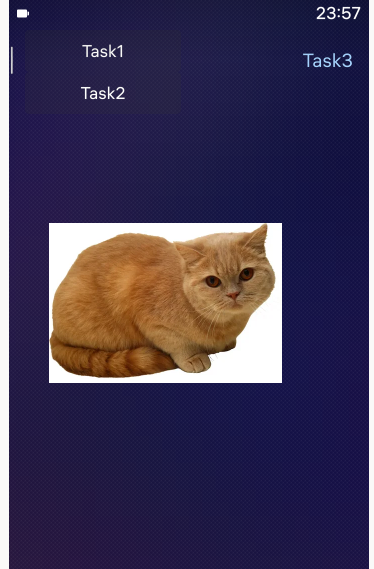
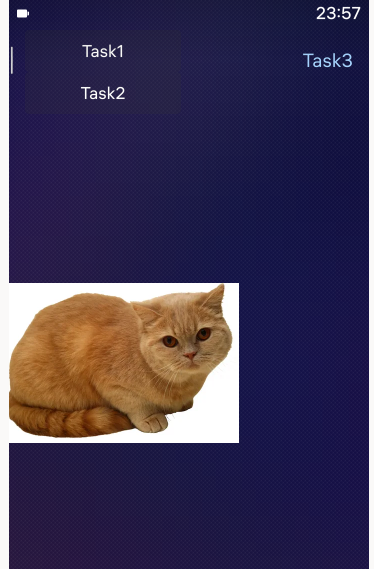
sourceSize.width: Math.max(Screen.width, Screen.height)

transformOrigin: Item.TopLeft

}

}

}



1. **ВЫВОД**

Мы ознакомились с ОС Аврора. Создали первый проект. Научились создавать кнопки и изменять (увеличивать/крутить) изображение.