МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования   
**«Национальный исследовательский   
Нижегородский государственный университет им. Н.И. Лобачевского»**

**(ННГУ)**

**Институт информационных технологий, математики и механики**

**Кафедра Математического обеспечения и суперкомпьютерных технологий**

Направление подготовки: «Фундаментальная информатика и информационные технологии»

Профиль подготовки: «Инженерия программного обеспечения»

**ОТЧЕТ**

**По лабораторной работе №8**

по дисциплине «Технология программирования мобильных систем»

**Выполнила:** студентка группы 381906-3

Сорокина Екатерина Вячеславовна

**Руководитель:**

Доцент, кандидат технических наук

Борисов Николай Анатольевич

Нижний Новгород  
2022

**Содержание**

[1. Цель 3](#_Toc123151104)

[2. Постановка задачи 4](#_Toc123151105)

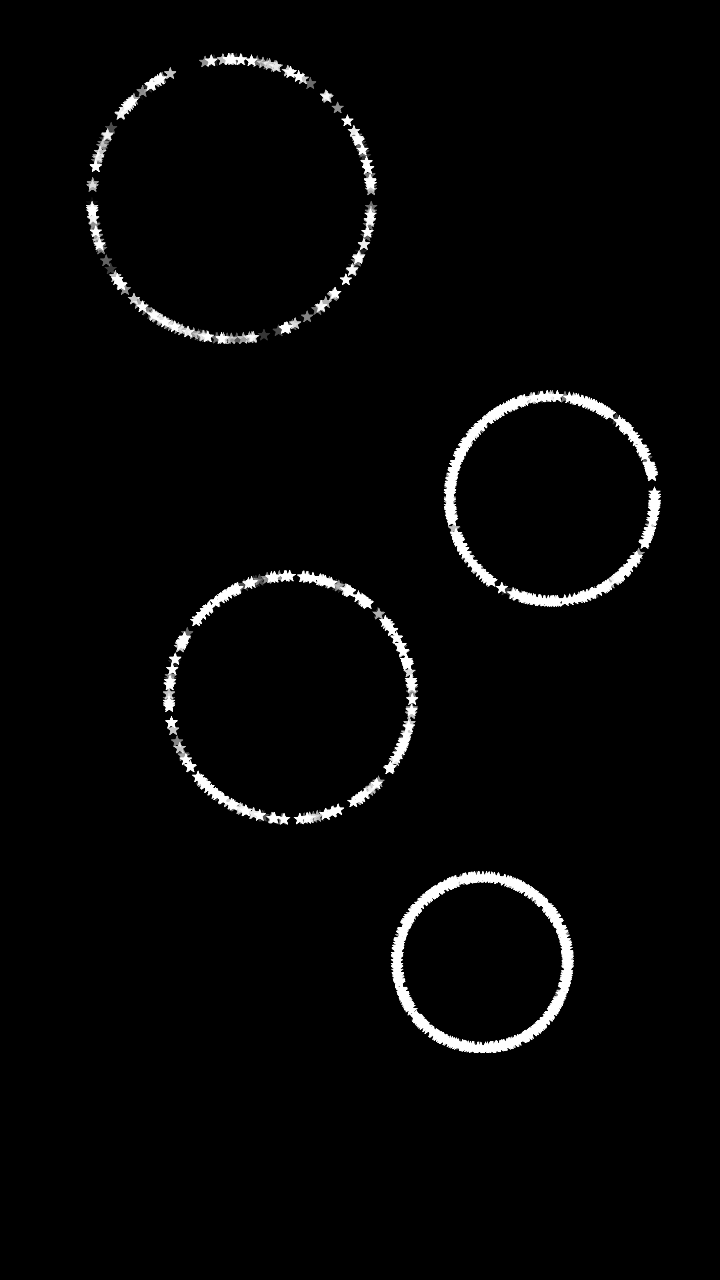
[3. Решение поставленной задачи 5](#_Toc123151106)

[4. Руководство пользователя 9](#_Toc123151107)

[Приложение 13](#_Toc123151108)

1. Цель

Цель: научиться использовать объект Canvas для рисования. Освоить реализацию анимации с помощью спрайтов. Научиться использовать частицы для создания визуальных эффектов. Изучить объекты мультимедиа для встраивания музыки и видео в приложение.

1. Постановка задачи
2. Нарисуйте простой дом с использованием объекта Canvas.  
   
3. Выберите спрайты со страницы <http://www.gameart2d.com/freebies.html> и сделайте анимацию с их использованием.
4. Реализуйте приложение, выбрасывающее частицы в месте нажатия на экран.
5. Реализуйте фонтан из частиц с использованием излучателя и гравитации.  
   
6. Реализуйте плеер для проигрывания музыкального файла. Музыкальные файлы можно найти на странице <http://www.stephaniequinn.com/samples.htm>.
7. Реализуйте приложение для воспроизведения видео с помощью объекта Video. Видео доступно по ссылке <http://clips.vorwaerts-gmbh.de/big_buck_bunny.ogv>.
8. Для приложения из задания №5 добавьте возможность переключения между музыкальными файлами для проигрывания с помощью объекта PlayList.
9. Решение поставленной задачи

Создаём проект со стандартной заготовкой приложения.

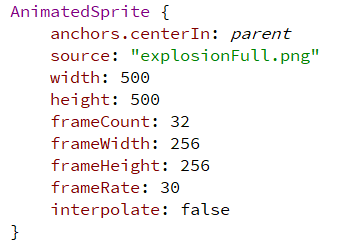
**Задание 1.**

Вызов Canvas и последовательная отрисовка элементов домика в плоском виде



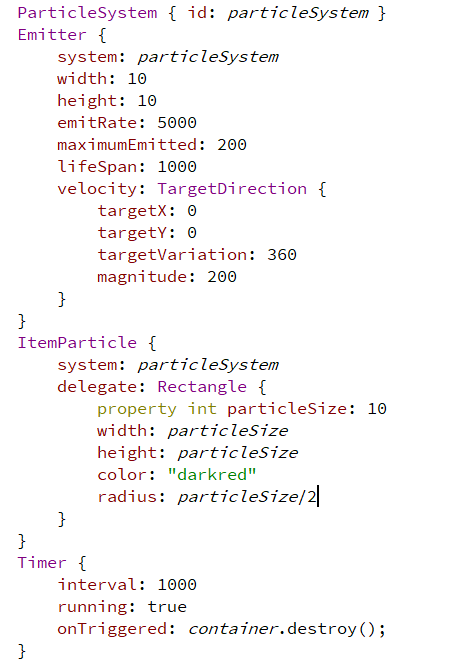
**Задание 2.**

Вызов AnimatedSprite для анимации спрайта по картинке, указанноцй по ссылке в свойстве source

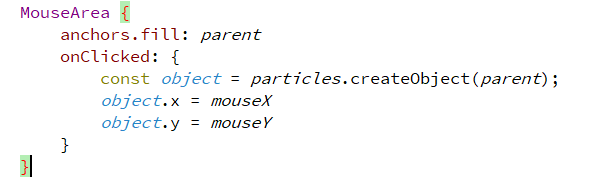


**Задание 3.**

Создание системы частиц, излучателя, контролирующего их количество, время жизни и направление излучения и таймера, который уничтожает их через промежуток времени

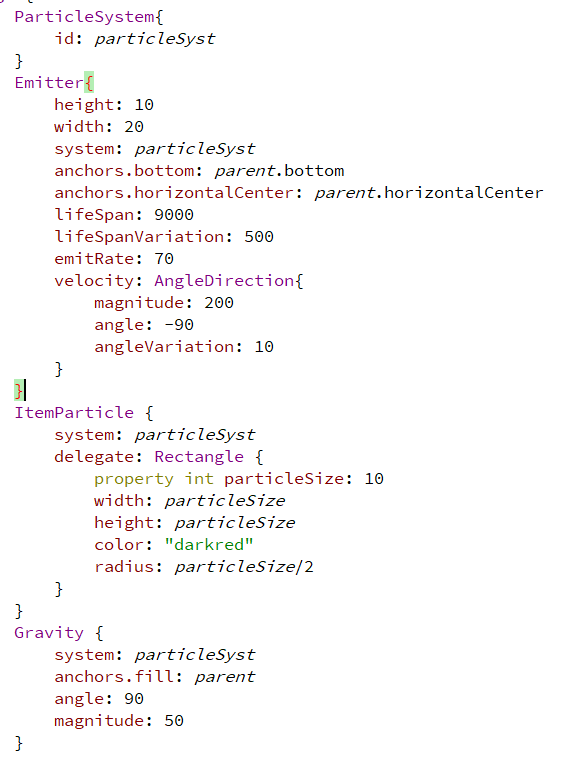


Появление частиц в месте клика мышки



**Задание 4.**

Создание системы частиц, излучателя, контролирующего их количество, время жизни и направление излучения и таймера, который уничтожает их через промежуток времени и гравитации для притягивания их

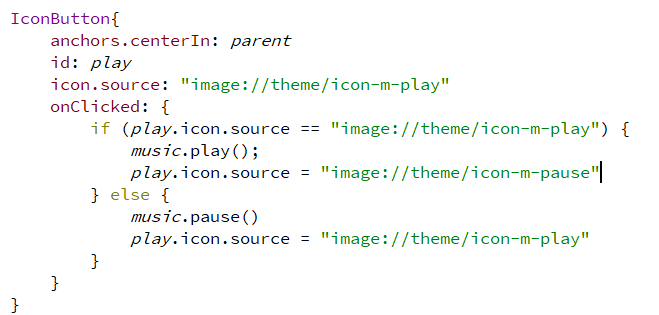


**Задание 5.**

Создание аудио элемента по ссылке из source

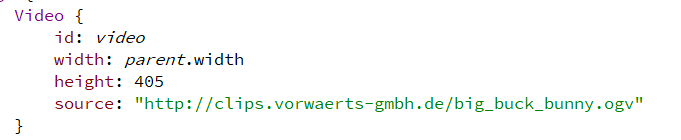


Кнопка для проигрывания и остановки трека



**Задание 6.**

Создание видео элемента по ссылке из source



Кнопки для проигрывания и остановки, перемотки вперед или назад



**Задание7.**

Создание плейлиста из нескольких аудио элементов



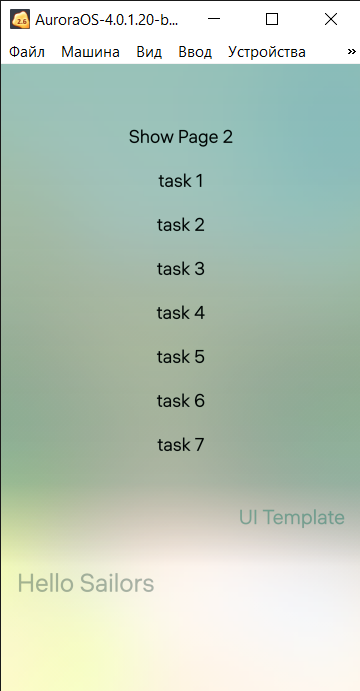
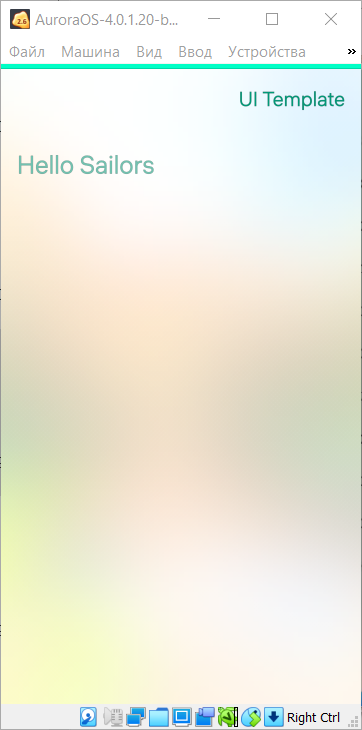
Кнопки для проигрывания и остановки трека, перехода к следующей композиции или предыдущей



Осталось только сохранить все изменения и запустить эмулятор.

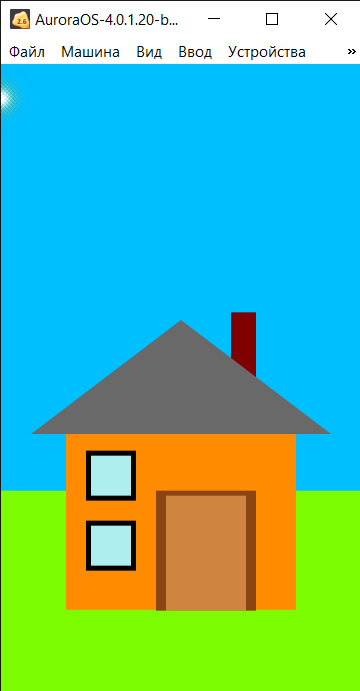
1. Руководство пользователя

При запуске приложения пользователь попадает на страницу FirstPage. Потянув вниз, открывается меню с нашими страницами.

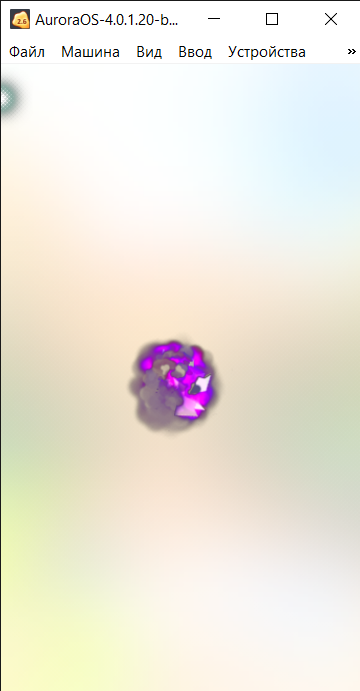


При нажатии на каждую из страниц, она загрузит нам наше задание.

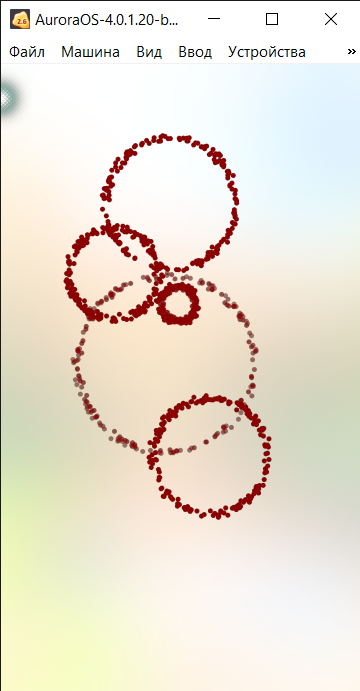
Задание 1:



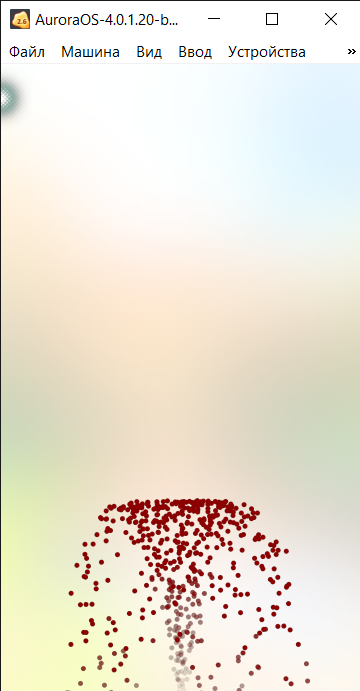
Задание 2:



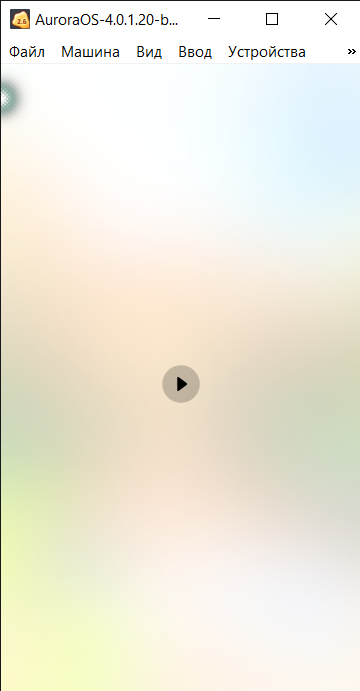
Задание 3:



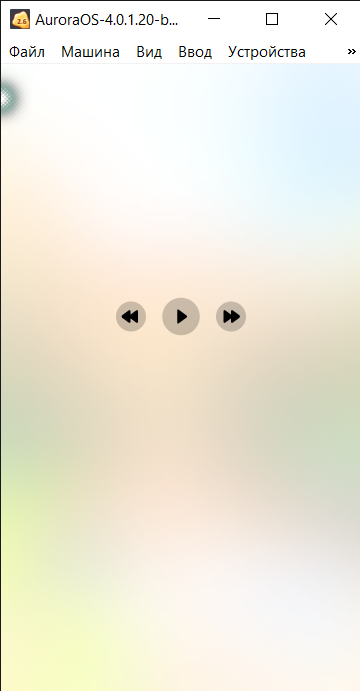
Задание 4:



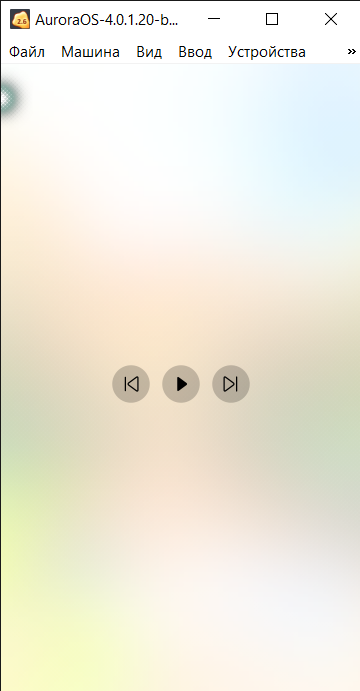
Задание 5:



Задание 6:



Задание 7:



Приложение

// FirstPage.qml

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

id: *page*

// The effective value will be restricted by ApplicationWindow.allowedOrientations

allowedOrientations: Orientation.All

// To enable PullDownMenu, place our content in a SilicaFlickable

SilicaFlickable {

anchors.fill: *parent*

// PullDownMenu and PushUpMenu must be declared in SilicaFlickable, SilicaListView or SilicaGridView

PullDownMenu {

MenuItem {

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

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

}

MenuItem{

text: *qsTr*("task 1")

onClicked: *pageStack*.push(*Qt*.resolvedUrl("task\_1.qml"))

}

MenuItem{

text: *qsTr*("task 2")

onClicked: *pageStack*.push(*Qt*.resolvedUrl("task\_2.qml"))

}

MenuItem{

text: *qsTr*("task 3")

onClicked: *pageStack*.push(*Qt*.resolvedUrl("task\_3.qml"))

}

MenuItem{

text: *qsTr*("task 4")

onClicked: *pageStack*.push(*Qt*.resolvedUrl("task\_4.qml"))

}

MenuItem{

text: *qsTr*("task 5")

onClicked: *pageStack*.push(*Qt*.resolvedUrl("task\_5.qml"))

}

MenuItem{

text: *qsTr*("task 6")

onClicked: *pageStack*.push(*Qt*.resolvedUrl("task\_6.qml"))

}

MenuItem {

text: *qsTr*("task 7")

onClicked: *pageStack*.push(*Qt*.resolvedUrl("task\_7.qml"))

}

}

// Tell SilicaFlickable the height of its content.

contentHeight: *column*.height

// Place our content in a Column. The PageHeader is always placed at the top

// of the page, followed by our content.

Column {

id: *column*

width: *page*.width

spacing: Theme.paddingLarge

PageHeader {

title: *qsTr*("UI Template")

}

Label {

x: Theme.horizontalPageMargin

text: *qsTr*("Hello Sailors")

color: Theme.secondaryHighlightColor

font.pixelSize: Theme.fontSizeExtraLarge

}

}

}

}

// task\_1.qml

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

Canvas{

width: *parent*.width

height: *parent*.height

onPaint: {

var *ctx* = *getContext*("2d");

*ctx*.fillStyle = "lawngreen";

*ctx*.fillRect(0,0,*width*,*height*);

*ctx*.fillStyle = "deepskyblue";

*ctx*.fillRect(0, 0, *width*, *height*/1.5);

*ctx*.fillStyle = "darkorange";

*ctx*.fillRect(*width*/2-230, *height*/3+310, 460, 355)

*ctx*.fillStyle = "maroon"

*ctx*.fillRect(*width*/2+100, *height*/3+70, 50, 150)

*ctx*.fillStyle = "dimgray"

*ctx*.beginPath()

*ctx*.moveTo(*width*/2, *height*/2.5);

*ctx*.lineTo(*width*/2+300, *height*/2+100);

*ctx*.lineTo(*width*/2-300, *height*/2+100);

*ctx*.fill();

*ctx*.closePath();

*ctx*.fillStyle = "saddlebrown"

*ctx*.fillRect(*width*/2-50, *height*/1.5, 200, 240)

*ctx*.fillStyle = "peru"

*ctx*.fillRect(*width*/2-30, *height*/1.5+10, 160, 230)

*ctx*.fillStyle = "black"

*ctx*.fillRect(*width*/4-10,*height*/1.5-80,100,100)

*ctx*.fillRect(*width*/4-10,*height*/1.5+60,100,100)

*ctx*.fillStyle = "paleturquoise"

*ctx*.fillRect(*width*/4,*height*/1.5-70,80,80)

*ctx*.fillRect(*width*/4,*height*/1.5+70,80,80)

}

}

}

//task\_2.qml

import QtQuick 2.0

import Sailfish.Silica 1.0

Page {

AnimatedSprite {

anchors.centerIn: *parent*

source: "explosionFull.png"

width: 500

height: 500

frameCount: 32

frameWidth: 256

frameHeight: 256

frameRate: 30

interpolate: false

}

}

//task\_3.qml

import QtQuick 2.0

import Sailfish.Silica 1.0

import QtQuick.Particles 2.0

Page {

Component {

id: *particles*

Item {

id: *container*

width: 200

height: 200

ParticleSystem { id: *particleSystem* }

Emitter {

system: *particleSystem*

width: 10

height: 10

emitRate: 5000

maximumEmitted: 200

lifeSpan: 1000

velocity: TargetDirection {

targetX: 0

targetY: 0

targetVariation: 360

magnitude: 200

}

}

ItemParticle {

system: *particleSystem*

delegate: Rectangle {

property int particleSize: 10

width: *particleSize*

height: *particleSize*

color: "darkred"

radius: *particleSize*/2

}

}

Timer {

interval: 1000

running: true

onTriggered: *container*.destroy();

}

}

}

MouseArea {

anchors.fill: *parent*

onClicked: {

const *object* = *particles*.createObject(*parent*);

*object*.x = *mouseX*

*object*.y = *mouseY*

}

}

}

//task\_4.qml

import QtQuick 2.0

import Sailfish.Silica 1.0

import QtQuick.Particles 2.0

Page {

ParticleSystem{

id: *particleSyst*

}

Emitter{

height: 10

width: 20

system: *particleSyst*

anchors.bottom: *parent*.bottom

anchors.horizontalCenter: *parent*.horizontalCenter

lifeSpan: 9000

lifeSpanVariation: 500

emitRate: 70

velocity: AngleDirection{

magnitude: 200

angle: -90

angleVariation: 10

}

}

ItemParticle {

system: *particleSyst*

delegate: Rectangle {

property int particleSize: 10

width: *particleSize*

height: *particleSize*

color: "darkred"

radius: *particleSize*/2

}

}

Gravity {

system: *particleSyst*

anchors.fill: *parent*

angle: 90

magnitude: 50

}

}

//task\_5.qml

import QtQuick 2.0

import Sailfish.Silica 1.0

import QtMultimedia 5.6

Page {

Audio{

id: *music*

source: "jony-kometa.mp3"

// hasAudio: true

// source: "https://previews.customer.envatousercontent.com/files/423515637/preview.mp3"

volume: 0.5

}

IconButton{

anchors.centerIn: *parent*

id: *play*

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

onClicked: {

if (*play*.icon.source == "image://theme/icon-m-play") {

*music*.play();

*play*.icon.source = "image://theme/icon-m-pause"

} else {

*music*.pause()

*play*.icon.source = "image://theme/icon-m-play"

}

}

}

}

//task\_6.qml

import QtQuick 2.0

import Sailfish.Silica 1.0

import QtMultimedia 5.6

Page {

Video {

id: *video*

width: *parent*.width

height: 405

source: "http://clips.vorwaerts-gmbh.de/big\_buck\_bunny.ogv"

}

Row {

anchors.horizontalCenter: *parent*.horizontalCenter

anchors.top: *video*.bottom

anchors.topMargin: 50

IconButton {

icon.source: "image://theme/icon-m-media-rewind"

onClicked: *video*.seek(*video*.position - 5000)

}

IconButton {

id: *play*

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

onClicked: {

if (*play*.icon.source == "image://theme/icon-m-play") {

*video*.play();

*play*.icon.source = "image://theme/icon-m-pause"

} else {

*video*.pause()

*play*.icon.source = "image://theme/icon-m-play"

}

}

}

IconButton {

icon.source: "image://theme/icon-m-media-forward"

onClicked: *video*.seek(*video*.position + 5000)

}

}

}

// task\_7.qml

import QtQuick 2.0

import Sailfish.Silica 1.0

import QtMultimedia 5.6

Page {

Audio {

id: *audio*

playlist: Playlist {

id: *playlist*

playbackMode: Playlist.Loop

// PlaylistItem { source: "https://previews.customer.envatousercontent.com/files/422044190/preview.mp3" }

// PlaylistItem { source: "https://previews.customer.envatousercontent.com/files/423219930/preview.mp3" }

// PlaylistItem { source: "https://previews.customer.envatousercontent.com/files/423515637/preview.mp3" }

PlaylistItem { source: "jony-kometa.mp3" }

PlaylistItem { source: "jony-nikak.mp3" }

PlaylistItem { source: "jony-pustota.mp3" }

}

}

Row {

anchors.centerIn: *parent*

IconButton {

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

onClicked: {

*audio*.playlist.previous()

}

}

IconButton {

id: *play*

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

onClicked: {

if (*play*.icon.source == "image://theme/icon-m-play") {

*audio*.play();

*play*.icon.source = "image://theme/icon-m-pause"

} else {

*audio*.pause()

*play*.icon.source = "image://theme/icon-m-play"

}

}

}

IconButton {

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

onClicked: {

*audio*.playlist.next()

}

}

}

}