МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ 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 программной инженерии**

Направление подготовки:

«Фундаментальная информатика и информационные технологии»

**ОТЧЕТ**

по Практическому заданию №7:

**«Интерфейс пользователя как граф состоянии»**

**Выполнила:** студентка группы 381606-2

Колосова Евгения Александровна

**Научный руководитель:** Доцент кафедры программной инженерии

Борисов Николай Анатольевич

Нижний Новгород  
2019

Введение

В этой лабораторной работе поднята задача по созданию интерфейсов пользователя как графов состояния в мобильных приложениях с Sailfish OS.

Целью работы является приобретение основных навыков создания пользовательского интерфейса, конфигурируемого состояниями, реализовывания анимированных переходов при смене состояний и создание собственных QML компонентов. Требуется получить в итоге корректно работающую программу. Следует выполнить все этапы из постановки задачи этой лабораторной работы.

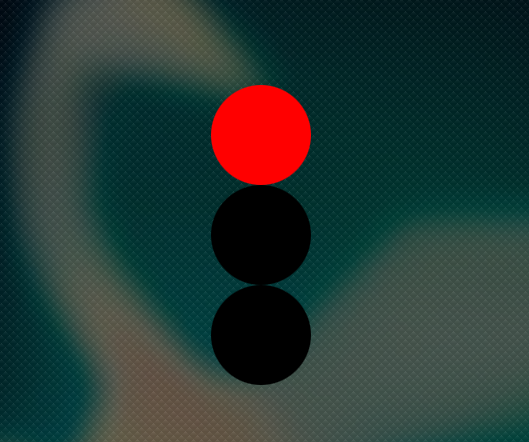
Постановка задачи

Для достижения в результате выполнения работы положительного результата, проект следует реализовывать в соответствии со следующими пунктами:

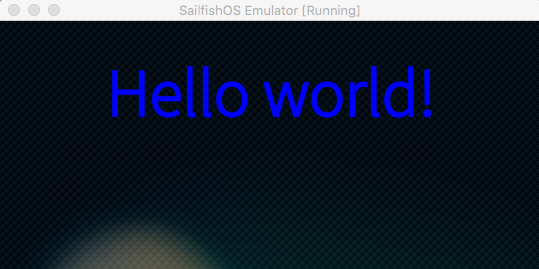
Основная цель работы: научиться организовывать многостраничное приложение, использовать контейнеры Silica, вытягиваемые меню и обложку приложения.

*Шаги:*

1. Создать приложение, отображающее светофор. На экране должно присутствовать 3 разноцветных сигнала, которые загораются и гаснут в том же порядке, что и сигналы светофора. Сделать автоматическую смену состояний.



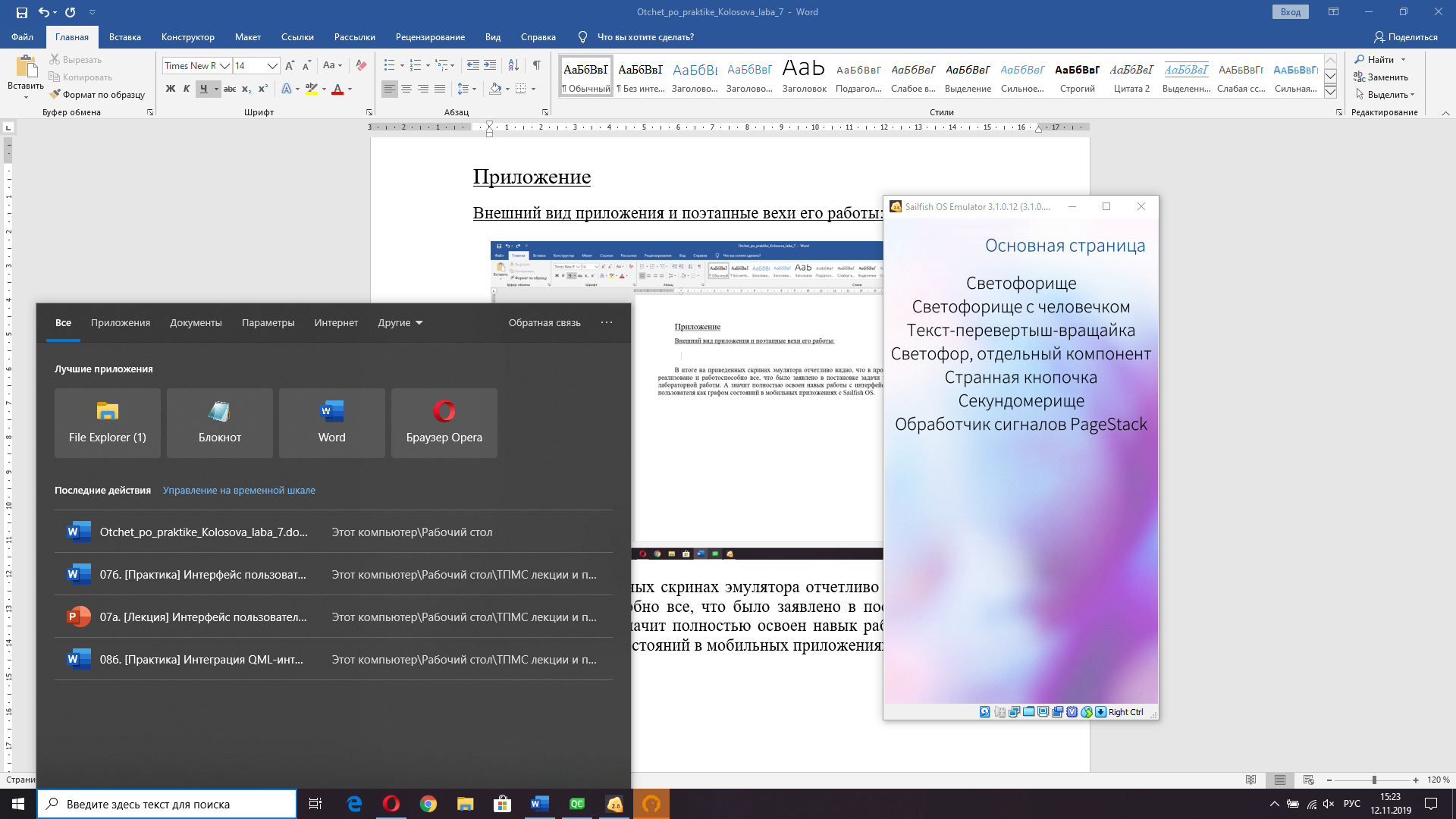
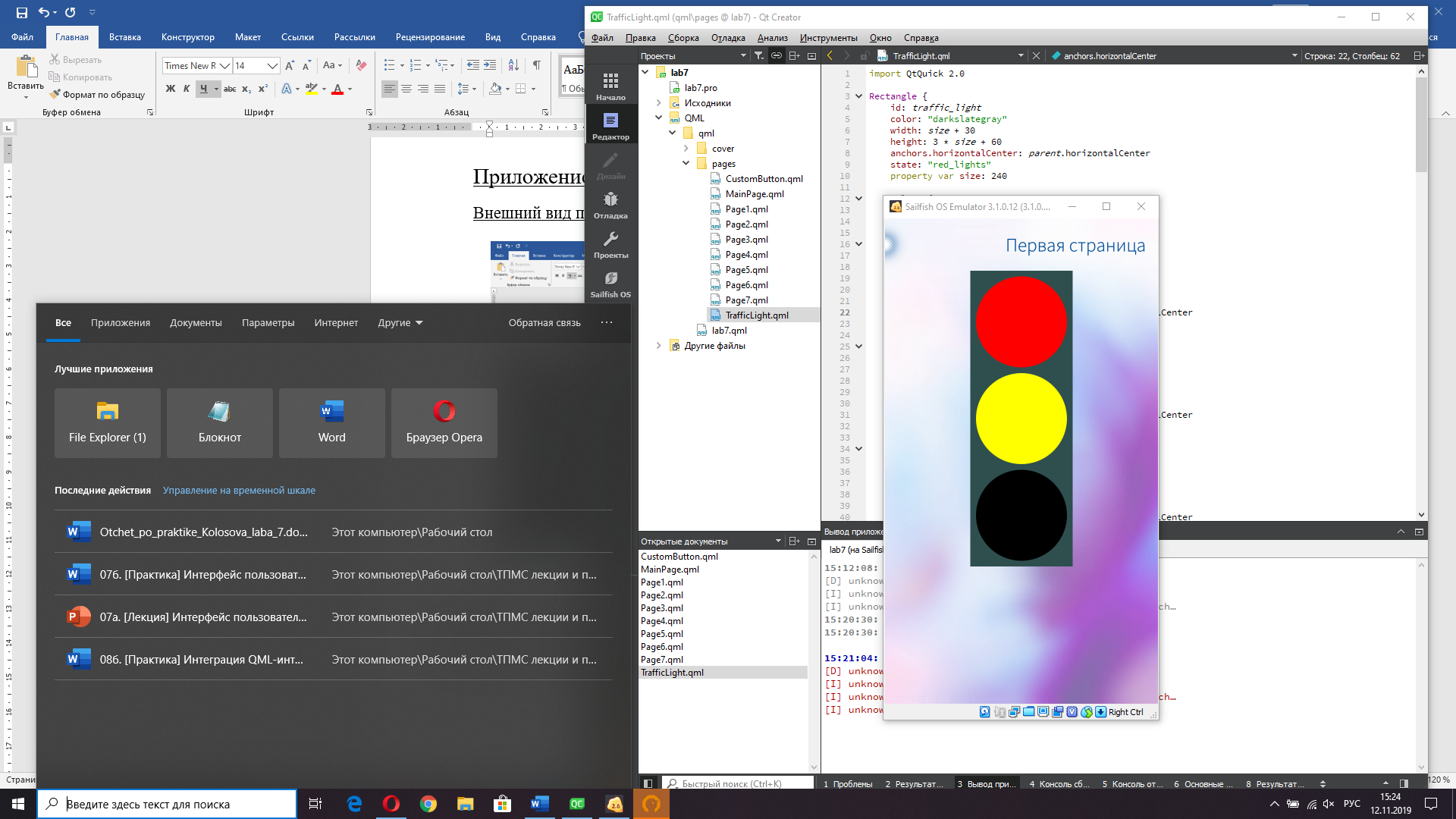
1. Доработать задание 1 так, чтобы во время зеленого сигнала светофора из одного конца экрана в другой плавно двигалась иконка человечка.
2. Создать приложение, отображающее строку текста вверху экрана. При нажатии на текст он должен плавно перемещаться вниз экрана, поворачивать на 180 градусов и менять цвет. Когда нажатие прекращается, он должен так же плавно возвращаться в исходное положение.

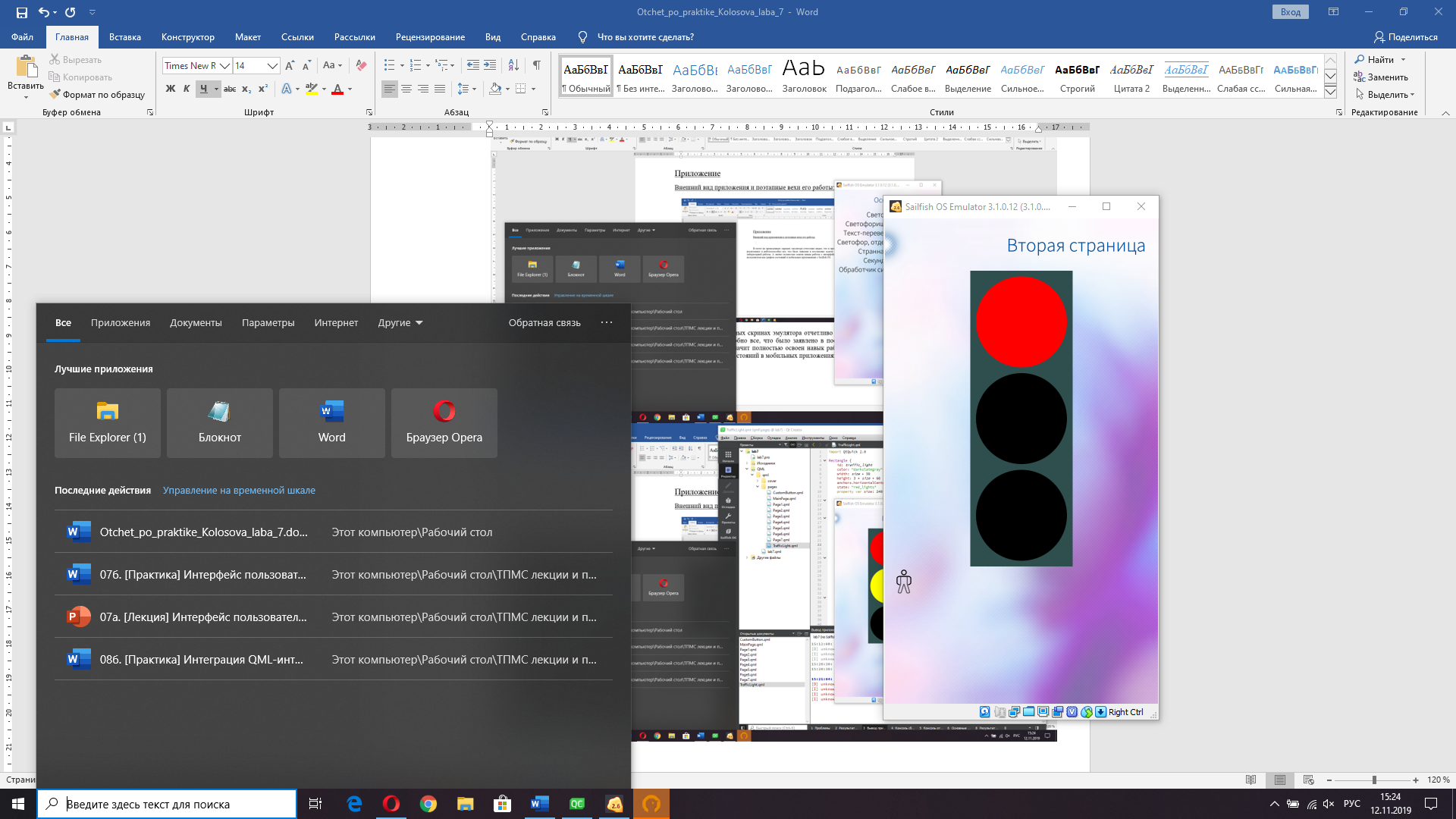
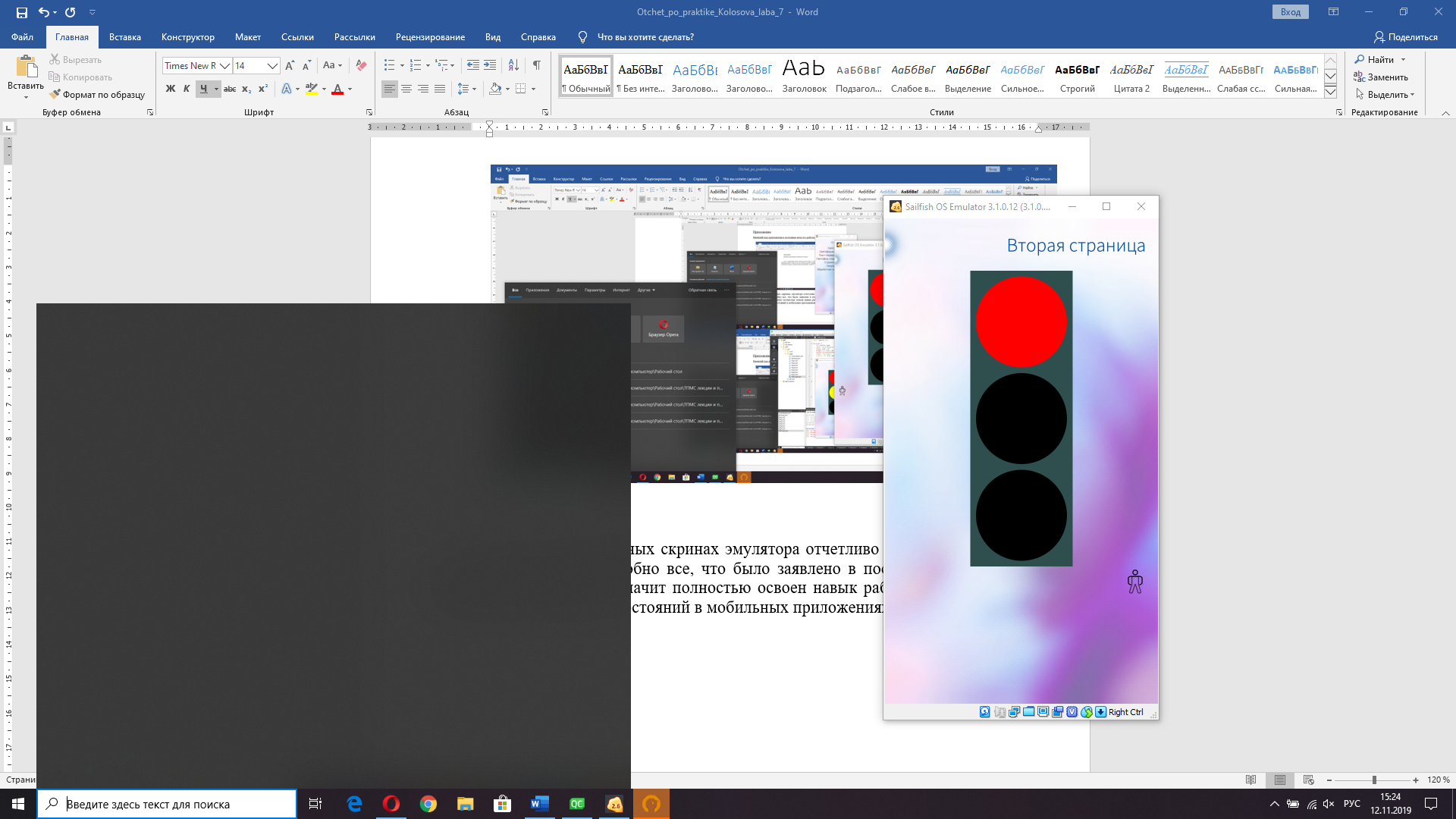


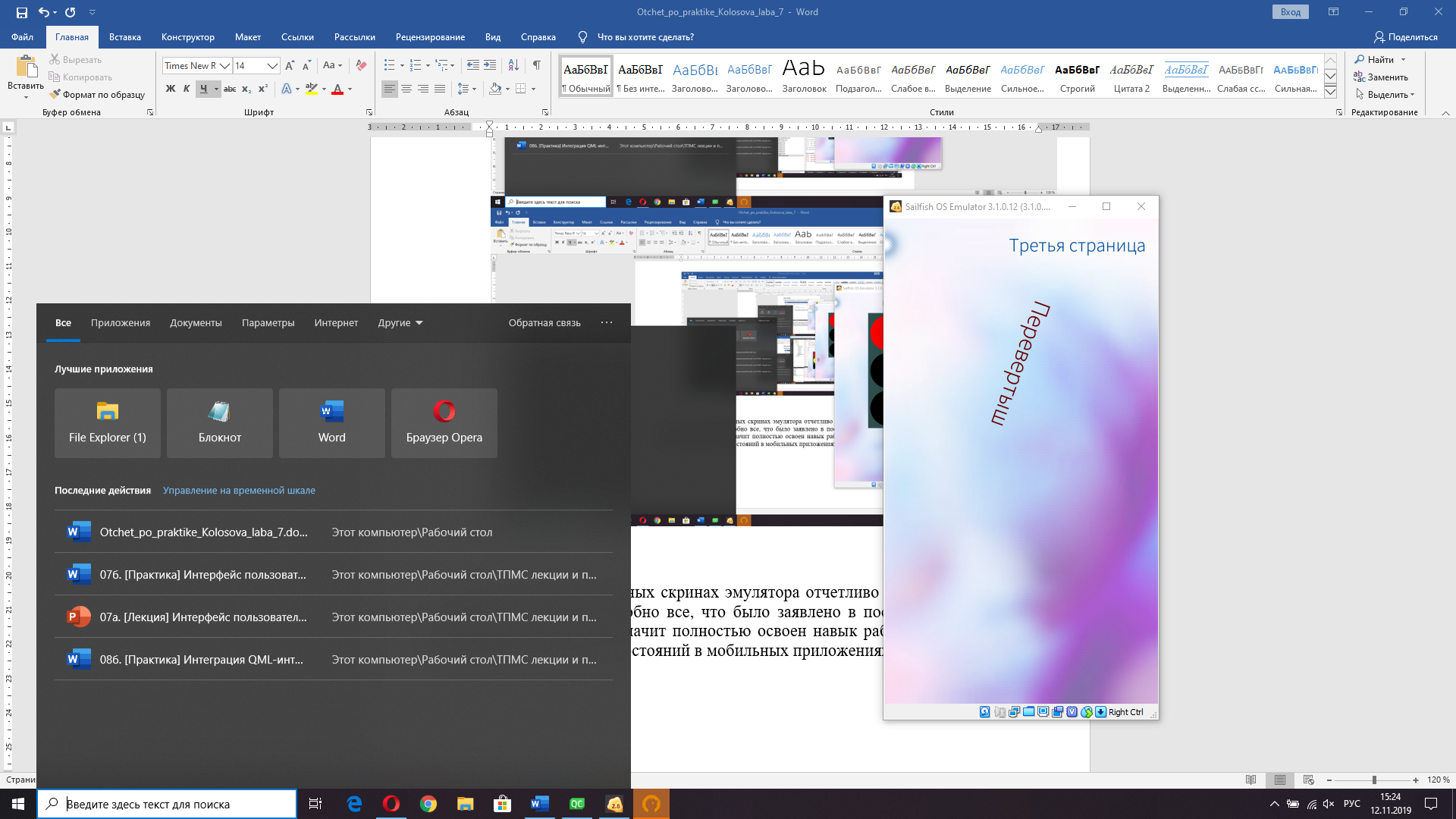
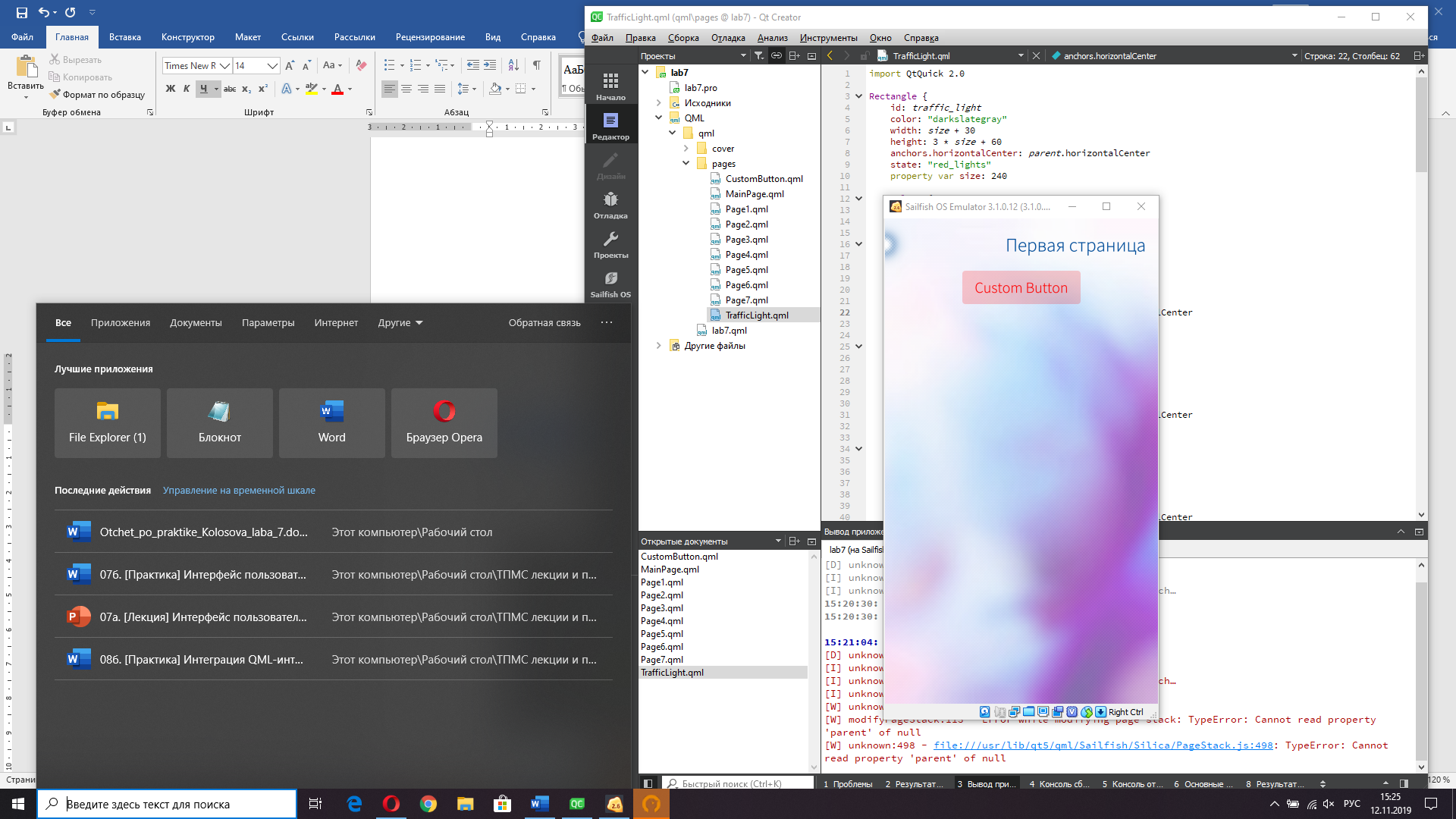
1. Выделить сигналы светофора из задания 1 в отдельный компонент и использовать его.
2. Создать QML компонент со свойством по умолчанию, который берет значение свойства text любого объявленного внутри него объекта и создает Button с тем же текстом. Добавить возможность задавать цвет кнопки при объявлении компонента.
3. Создать приложение-секундомер. На экране должны отображаться значения часов, минут и секунд. Секундомер запускается по сигналу кнопки, при повторном нажатии секундомер останавливается. Для отображения часов, минут и секунд использовать собственные QML компоненты.
4. Добавить обработчик сигналов PageStack, подсчитывающий количество добавленных и удаленных страниц в PageStack.

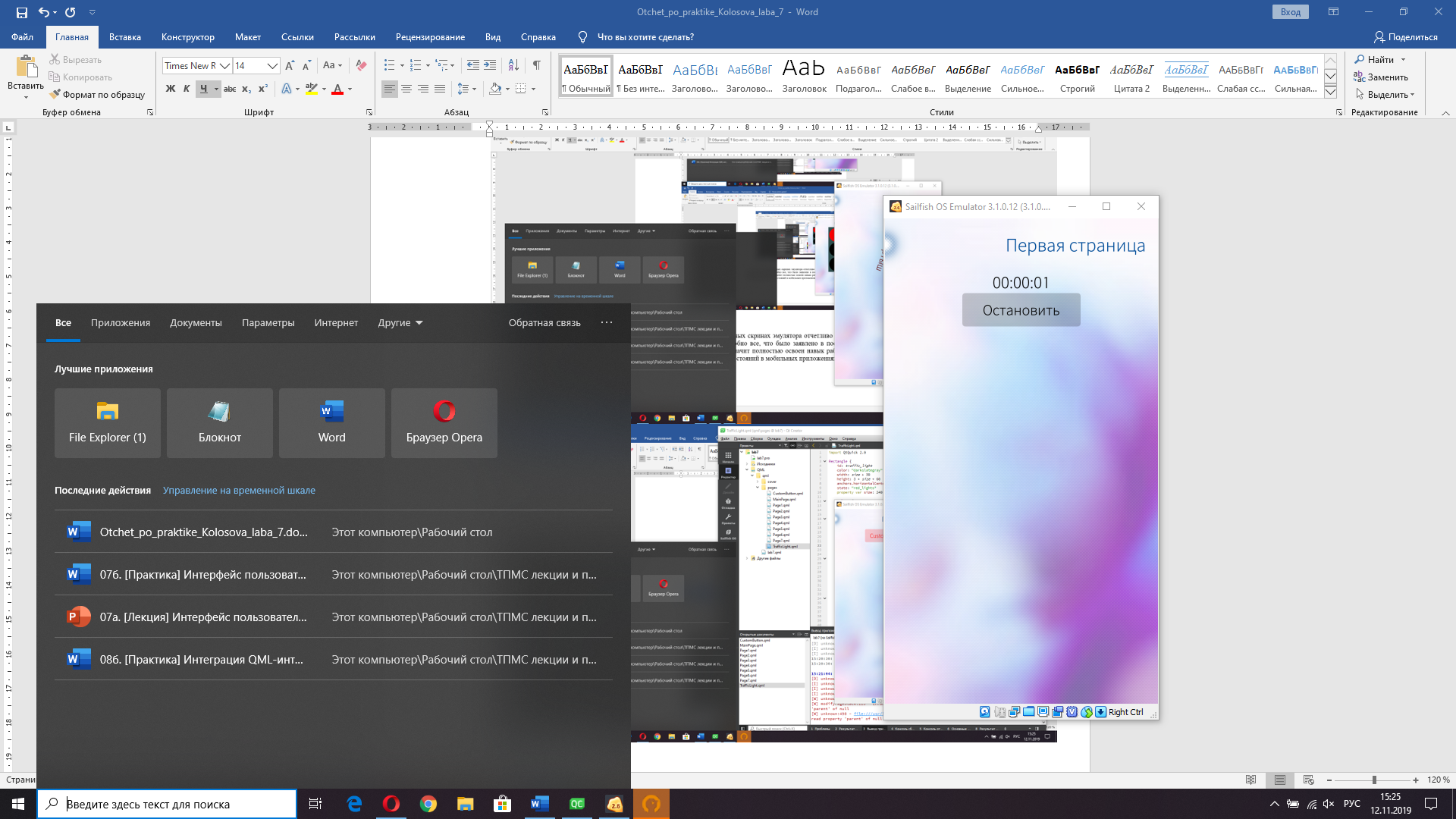
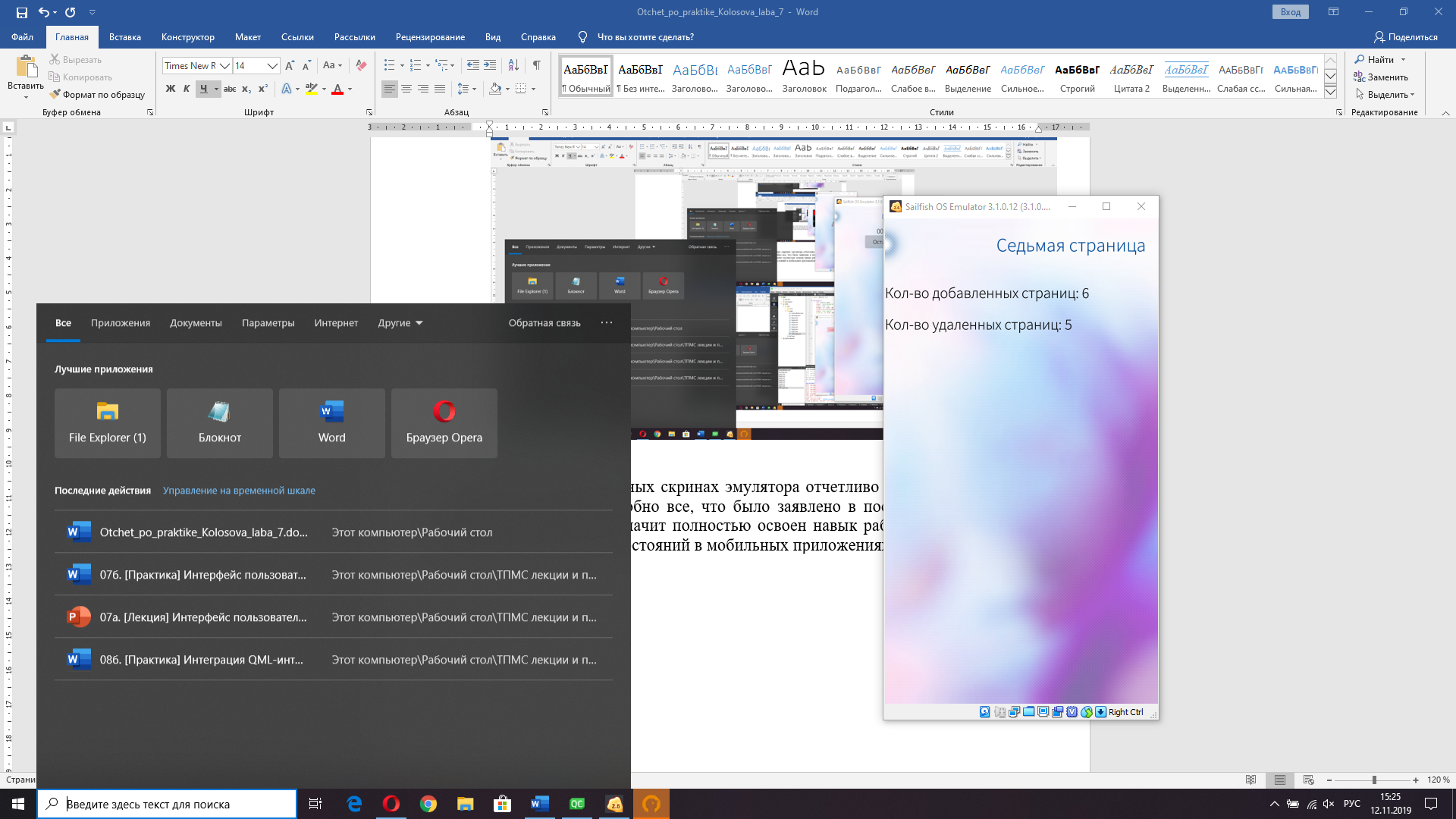
Приложение

Внешний вид приложения и поэтапные вехи его работы:

В итоге на приведенных скринах эмулятора отчетливо видно, что в проекте реализовано и работоспособно все, что было заявлено в постановке задачи этой лабораторной работы. А значит полностью освоен навык работы с интерфейсами пользователя как графом состояний в мобильных приложениях с Sailfish OS.

Код проекта:

MainPage.qml

1. import QtQuick 2.0
2. import Sailfish.Silica 1.0
3. Page {
4. **id: page**
5. allowedOrientations: Orientation.All
7. SilicaListView {
8. **anchors.fill: parent**
9. header: PageHeader { title: "Основная страница" }
11. model: ListModel {
12. **id: listModel**
13. Component.onCompleted: {
14. append({"name": "Светофорище"});
16. append({"name": "Светофорище с человечком"});
18. append({"name": "Текст-перевертыш-вращайка"});
20. append({"name": "Светофор, отдельный компонент"});
22. **append({"name": "Странная кнопочка"});**
24. append({"name": "Секундомерище"});
26. append({"name": "Обработчик сигналов PageStack"});
27. **}**
28. }
30. delegate: ListItem {
31. width: ListView.view.width
32. **contentHeight: label\_name.height**
34. Label {
35. id: label\_name
36. text: model.name
37. **font.pixelSize: 48**
38. anchors.horizontalCenter: parent.horizontalCenter
39. }
41. onClicked: {
42. **var idx = parseInt(index) + 1;**
43. pageStack.push(Qt.resolvedUrl("Page" + idx.toString() + ".qml"));
44. }
45. }
46. }
47. **}**

Page1.qml

1. import QtQuick 2.0
2. import Sailfish.Silica 1.0
4. Page {
5. **id: page**
7. allowedOrientations: Orientation.All
8. property int size: 240
10. **Component.onCompleted: change\_state()**
12. SilicaFlickable {
13. anchors.fill: parent
14. contentHeight: column.height
16. Column {
17. id: column
18. width: page.width
20. **PageHeader { title: "Первая страница"}**
22. Rectangle {
23. id: traffic\_light
24. color: "darkslategray"
25. **width: size + 30**
26. height: 3 \* size + 60
27. anchors.horizontalCenter: parent.horizontalCenter
28. state: "red\_lights"
30. **Column {**
31. anchors.centerIn: parent
32. spacing: 15
34. Rectangle {
35. **id: red\_light**
36. width: size
37. height: size
38. radius: size
39. color: "black"
40. **anchors.horizontalCenter: parent.horizontalCenter**
41. }
43. Rectangle {
44. id: yellow\_light
45. **width: size**
46. height: size
47. radius: size
48. color: "black"
49. anchors.horizontalCenter: parent.horizontalCenter
50. **}**
52. Rectangle {
53. id: green\_light
54. width: size
55. **height: size**
56. radius: size
57. color: "black"
58. anchors.horizontalCenter: parent.horizontalCenter
59. }
60. **}**
62. states: [
63. State {
64. name: "none"
65. **},**
67. State {
68. name: "red\_lights"
69. PropertyChanges {
70. **target: red\_light**
71. color: "red"
72. }
73. },
75. **State {**
76. name: "switch\_to\_green"
77. PropertyChanges {
78. target: red\_light
79. color: "red"
80. **}**
81. PropertyChanges {
82. target: yellow\_light
83. color: "yellow"
84. }
85. **},**
87. State {
88. name: "yellow\_lights"
89. PropertyChanges {
90. **target: yellow\_light**
91. color: "yellow"
92. }
93. },
95. **State {**
96. name: "green\_lights"
97. PropertyChanges {
98. target: green\_light
99. color: "green"
100. **}**
101. }
102. ]
104. transitions: [
105. **Transition {**
106. from: "red\_lights"
107. to: "switch\_to\_green"
108. SequentialAnimation {
109. PauseAnimation {
110. **duration: 3000**
111. }
112. ColorAnimation {
113. targets: [red\_light, yellow\_light, green\_light]
114. property: "color"
115. **duration: 0**
116. }
117. ScriptAction {
118. script: change\_state()
119. }
120. **}**
121. },
123. Transition {
124. from: "switch\_to\_green"
125. **to: "green\_lights"**
126. SequentialAnimation {
127. PauseAnimation {
128. duration: 1000
129. }
130. **ColorAnimation {**
131. targets: [red\_light, yellow\_light, green\_light]
132. property: "color"
133. duration: 0
134. }
135. **ScriptAction {**
136. script: change\_state()
137. }
138. }
139. },
141. Transition {
142. from: "green\_lights"
143. to: "yellow\_lights"
145. **SequentialAnimation {**
146. PauseAnimation {
147. duration: 3000
148. }
149. ColorAnimation {
150. **target: green\_light**
151. property: "color"
152. to: "black"
153. duration: 0
154. }
155. **PauseAnimation {**
156. duration: 500
157. }
158. ColorAnimation {
159. target: green\_light
160. **property: "color"**
161. to: "green"
162. duration: 0
163. }
164. PauseAnimation {
165. **duration: 500**
166. }
167. ColorAnimation {
168. target: green\_light
169. property: "color"
170. **to: "black"**
171. duration: 0
172. }
173. PauseAnimation {
174. duration: 500
175. **}**
176. ColorAnimation {
177. target: green\_light
178. property: "color"
179. to: "green"
180. **duration: 0**
181. }
182. PauseAnimation {
183. duration: 500
184. }
185. **ColorAnimation {**
186. target: green\_light
187. property: "color"
188. to: "black"
189. duration: 0
190. **}**
191. PauseAnimation {
192. duration: 500
193. }
194. ColorAnimation {
195. **target: green\_light**
196. property: "color"
197. to: "green"
198. duration: 0
199. }
200. **PauseAnimation {**
201. duration: 500
202. }
203. ColorAnimation {
204. target: green\_light
205. **property: "color"**
206. to: "black"
207. duration: 0
208. }
210. **ColorAnimation {**
211. targets: [red\_light, yellow\_light, green\_light]
212. property: "color"
213. duration: 0
214. }
215. **ScriptAction {**
216. script: change\_state()
217. }
218. }
219. },
221. Transition {
222. from: "yellow\_lights"
223. to: "red\_lights"
225. **SequentialAnimation {**
226. PauseAnimation {
227. duration: 1000
228. }
229. ColorAnimation {
230. **targets: [red\_light, yellow\_light, green\_light]**
231. property: "color"
232. duration: 0
233. }
234. ScriptAction {
235. **script: change\_state()**
236. }
237. }
238. }
239. ]
240. **}**
241. }
242. }
244. **function** change\_state() {
245. **switch(traffic\_light.state) {**
246. **case** "red\_lights":
247. traffic\_light.state = "switch\_to\_green";
248. **break**;
249. **case** "switch\_to\_green":
250. **traffic\_light.state = "green\_lights";**
251. **break**;
252. **case** "green\_lights":
253. traffic\_light.state = "yellow\_lights";
254. **break**;
255. **case "yellow\_lights":**
256. traffic\_light.state = "red\_lights";
257. **break**;
258. }
259. }
260. **}**

Page2.qml

1. import QtQuick 2.0
2. import Sailfish.Silica 1.0
4. Page {
5. **id: page**
7. allowedOrientations: Orientation.All
8. property int size: 240
10. **Component.onCompleted: change\_state()**
12. SilicaFlickable {
13. anchors.fill: parent
14. contentHeight: column.height
16. Column {
17. id: column
18. width: page.width
20. **PageHeader { title: "Вторая страница"}**
22. Rectangle {
23. id: traffic\_light
24. color: "darkslategray"
25. **width: size + 30**
26. height: 3 \* size + 60
27. anchors.horizontalCenter: parent.horizontalCenter
28. state: "red\_lights"
30. **Column {**
31. anchors.centerIn: parent
32. spacing: 15
34. Rectangle {
35. **id: red\_light**
36. width: size
37. height: size
38. radius: size
39. color: "black"
40. **anchors.horizontalCenter: parent.horizontalCenter**
41. }
43. Rectangle {
44. id: yellow\_light
45. **width: size**
46. height: size
47. radius: size
48. color: "black"
49. anchors.horizontalCenter: parent.horizontalCenter
50. **}**
52. Rectangle {
53. id: green\_light
54. width: size
55. **height: size**
56. radius: size
57. color: "black"
58. anchors.horizontalCenter: parent.horizontalCenter
59. }
60. **}**
62. states: [
63. State {
64. name: "red\_lights"
65. **PropertyChanges {**
66. target: red\_light
67. color: "red"
68. }
69. },
71. State {
72. name: "switch\_to\_green"
73. PropertyChanges {
74. target: red\_light
75. **color: "red"**
76. }
77. PropertyChanges {
78. target: yellow\_light
79. color: "yellow"
80. **}**
81. },
83. State {
84. name: "yellow\_lights"
85. **PropertyChanges {**
86. target: yellow\_light
87. color: "yellow"
88. }
89. },
91. State {
92. name: "green\_lights"
93. PropertyChanges {
94. target: green\_light
95. **color: "green"**
96. }
97. }
98. ]
100. **transitions: [**
101. Transition {
102. from: "red\_lights"
103. to: "switch\_to\_green"
104. SequentialAnimation {
105. **PauseAnimation {**
106. duration: 3000
107. }
108. ColorAnimation {
109. targets: [red\_light, yellow\_light, green\_light]
110. **property: "color"**
111. duration: 0
112. }
113. ScriptAction {
114. script: change\_state()
115. **}**
116. }
117. },
119. Transition {
120. **from: "switch\_to\_green"**
121. to: "green\_lights"
122. SequentialAnimation {
123. PauseAnimation {
124. duration: 1000
125. **}**
126. ColorAnimation {
127. targets: [red\_light, yellow\_light, green\_light]
128. property: "color"
129. duration: 0
130. **}**
131. ScriptAction {
132. script: change\_state()
133. }
134. }
135. **},**
137. Transition {
138. from: "green\_lights"
139. to: "yellow\_lights"
141. SequentialAnimation {
142. PauseAnimation {
143. duration: 3000
144. }
145. **ColorAnimation {**
146. target: green\_light
147. property: "color"
148. to: "black"
149. duration: 0
150. **}**
151. PauseAnimation {
152. duration: 500
153. }
154. ColorAnimation {
155. **target: green\_light**
156. property: "color"
157. to: "green"
158. duration: 0
159. }
160. **PauseAnimation {**
161. duration: 500
162. }
163. ColorAnimation {
164. target: green\_light
165. **property: "color"**
166. to: "black"
167. duration: 0
168. }
169. PauseAnimation {
170. **duration: 500**
171. }
172. ColorAnimation {
173. target: green\_light
174. property: "color"
175. **to: "green"**
176. duration: 0
177. }
178. PauseAnimation {
179. duration: 500
180. **}**
181. ColorAnimation {
182. target: green\_light
183. property: "color"
184. to: "black"
185. **duration: 0**
186. }
187. PauseAnimation {
188. duration: 500
189. }
190. **ColorAnimation {**
191. target: green\_light
192. property: "color"
193. to: "green"
194. duration: 0
195. **}**
196. PauseAnimation {
197. duration: 500
198. }
199. ColorAnimation {
200. **target: green\_light**
201. property: "color"
202. to: "black"
203. duration: 0
204. }
206. ColorAnimation {
207. targets: [red\_light, yellow\_light, green\_light]
208. property: "color"
209. duration: 0
210. **}**
211. ScriptAction {
212. script: change\_state()
213. }
214. }
215. **},**
217. Transition {
218. from: "yellow\_lights"
219. to: "red\_lights"
221. SequentialAnimation {
222. PauseAnimation {
223. duration: 1000
224. }
225. **ColorAnimation {**
226. targets: [red\_light, yellow\_light, green\_light]
227. property: "color"
228. duration: 0
229. }
230. **ScriptAction {**
231. script: change\_state()
232. }
233. }
234. }
235. **]**
236. }
238. Image {
239. id: image\_person
240. **source: "image://theme/icon-m-person"**
241. state: "one\_side"
243. states: [
244. State {
245. **name: "one\_side"**
246. PropertyChanges {
247. target: image\_person
248. x: 10
249. }
250. **},**
252. State {
253. name: "other\_side"
254. PropertyChanges {
255. **target: image\_person**
256. x: parent.width - 100
257. }
258. }
259. ]
261. transitions: [
262. Transition {
263. NumberAnimation {
264. property: "x"
265. **duration: 3000**
266. }
267. }
268. ]
269. }
270. **}**
271. }
273. **function** change\_state() {
274. **switch**(traffic\_light.state) {
275. **case "red\_lights":**
276. traffic\_light.state = "switch\_to\_green";
277. **break**;
278. **case** "switch\_to\_green":
279. traffic\_light.state = "green\_lights";
280. **break;**
281. **case** "green\_lights":
282. image\_person.state = (image\_person.state == "one\_side") ? "other\_side" : "one\_side";
283. traffic\_light.state = "yellow\_lights";
284. **break**;
285. **case "yellow\_lights":**
286. traffic\_light.state = "red\_lights";
287. **break**;
288. }
289. }
290. **}**

Page3.qml

1. import QtQuick 2.0
2. import Sailfish.Silica 1.0
4. Page {
5. **id: page**
7. allowedOrientations: Orientation.All
9. SilicaFlickable {
10. **anchors.fill: parent**
11. contentHeight: column.height
13. Column {
14. id: column
15. **width: page.width**
17. PageHeader { title: "Третья страница"}
19. Label {
20. **id: label**
21. text: "Перевертыш"
22. color: "black"
23. font.pixelSize: 64
24. anchors.horizontalCenter: parent.horizontalCenter
25. **state: "normal"**
27. states: [
28. State {
29. name: "normal"
31. PropertyChanges {
32. target: label
33. y: 100
34. }
35. **PropertyChanges {**
36. target: label
37. color: "red"
38. }
39. PropertyChanges {
40. **target: label**
41. rotation: 0
42. }
43. },
45. **State {**
46. name: "inverted"
48. PropertyChanges {
49. target: label
50. **y: 500**
51. }
52. PropertyChanges {
53. target: label
54. color: "black"
55. **}**
56. PropertyChanges {
57. target: label
58. rotation: 180
59. }
60. **}**
61. ]
63. transitions: [
64. Transition {
65. **NumberAnimation {**
66. property: "y"
67. duration: 3000
68. }
69. RotationAnimation {
70. **direction: RotationAnimation.Counterclockwise**
71. duration: 3000
72. }
73. ColorAnimation {
74. duration: 3000
75. **}**
76. }
77. ]
79. MouseArea {
80. **anchors.fill: parent**
81. onPressed: label.state = (label.state == "normal") ? "inverted" : "normal"
82. onReleased: label.state = (label.state == "normal") ? "inverted" : "normal"
83. }
85. **}**
86. }
87. }
88. }

Page4.qml

1. import QtQuick 2.0
2. import Sailfish.Silica 1.0
3. import "C:/Users/Evgen128/SailfishOSProjects/lab7/pages/TrafficLight.qml"
5. **Page {**
6. id: page
8. allowedOrientations: Orientation.All
9. Component.onCompleted: traffic\_light.change\_state()
11. SilicaFlickable {
12. anchors.fill: parent
13. contentHeight: column.height
15. **Column {**
16. id: column
17. width: page.width
19. PageHeader { title: "Четвертая страница"}
21. TrafficLight {
22. id: traffic\_light
23. }
24. }
25. **}**
26. }

Page5.qml

1. import QtQuick 2.0
2. import Sailfish.Silica 1.0
3. import "C:/Users/Evgen128/SailfishOSProjects/lab7/pages/CustomButton.qml"
5. **Page {**
6. id: page
8. allowedOrientations: Orientation.All
10. **SilicaFlickable {**
11. anchors.fill: parent
12. contentHeight: column.height
14. Column {
15. **id: column**
16. width: page.width
18. PageHeader { title: "Первая страница"}
20. **CustomButton {**
21. anchors.horizontalCenter: parent.horizontalCenter
22. color\_button: "red"
24. Text { text: "Custom Button" }
25. **}**
27. }
28. }
29. }

Page6.qml

1. import QtQuick 2.0
2. import Sailfish.Silica 1.0
4. Page {
5. **id: page**
7. allowedOrientations: Orientation.All
9. SilicaFlickable {
10. **anchors.fill: parent**
12. Column {
13. id: column
14. width: page.width
16. PageHeader { title: "Первая страница"}
18. Timer {
19. id: timer
20. **property var ticks: 0**
21. interval: 1000
22. repeat: **true**
24. onTriggered: time.text = dispaly\_time(++ticks);
25. **}**
27. Text {
28. id: time
29. text: dispaly\_time(timer.ticks)
30. **font.pixelSize: 46**
31. anchors.horizontalCenter: parent.horizontalCenter
32. }
34. Button {
35. **id: button**
36. anchors.horizontalCenter: parent.horizontalCenter
37. text: timer.running ? "Остановить" : "Запустить"
39. onClicked: {
40. **timer.running ? timer.stop() : timer.start();**
41. }
42. }
43. }
44. }
46. **function** dispaly\_time(ticks) {
47. **var** hh = Math.floor(ticks / 3600);
48. **var** mm = Math.floor((ticks % 3600) / 60);
49. **var** ss = ticks % 60
51. **return** pad(hh, 2) + ":" + pad(mm, 2) + ":" + pad(ss, 2);
52. }
54. **function** pad(n, width) {
55. **n = n + "";**
56. **return** n.length >= width ? n : **new** Array(width - n.length + 1).join("0") + n;
57. }
58. }

Page7.qml

1. import QtQuick 2.0
2. import Sailfish.Silica 1.0
4. Page {
5. **id: page**
7. allowedOrientations: Orientation.All
9. SilicaFlickable {
10. **anchors.fill: parent**
12. Column {
13. id: column
14. width: page.width
15. **spacing: Theme.paddingLarge**
17. PageHeader { title: "Седьмая страница"}
19. Label {
20. **id: label\_count\_pushed**
21. text: "Кол-во добавленных страниц: " + addedPages
22. }
24. Label {
25. **id: label\_count\_poped**
26. text: "Кол-во удаленных страниц: " + deletedPages
27. }
28. }
29. }
30. **}**

CustomButton.qml

1. import QtQuick 2.0
2. import Sailfish.Silica 1.0
4. Button {
5. **default property var some\_text**
6. property alias color\_button: button.color
8. id: button
9. text: some\_text.text
10. **}**

TrafficLight.qml

1. import QtQuick 2.0
3. Rectangle {
4. id: traffic\_light
5. **color: "darkslategray"**
6. width: size + 30
7. height: 3 \* size + 60
8. anchors.horizontalCenter: parent.horizontalCenter
9. state: "red\_lights"
10. **property var size: 240**
12. Column {
13. anchors.centerIn: parent
14. spacing: 15
16. Rectangle {
17. id: red\_light
18. width: size
19. height: size
20. **radius: size**
21. color: "black"
22. anchors.horizontalCenter: parent.horizontalCenter
23. }
25. **Rectangle {**
26. id: yellow\_light
27. width: size
28. height: size
29. radius: size
30. **color: "black"**
31. anchors.horizontalCenter: parent.horizontalCenter
32. }
34. Rectangle {
35. **id: green\_light**
36. width: size
37. height: size
38. radius: size
39. color: "black"
40. **anchors.horizontalCenter: parent.horizontalCenter**
41. }
42. }
44. states: [
45. **State {**
46. name: "red\_lights"
47. PropertyChanges {
48. target: red\_light
49. color: "red"
50. **}**
51. },
53. State {
54. name: "switch\_to\_green"
55. **PropertyChanges {**
56. target: red\_light
57. color: "red"
58. }
59. PropertyChanges {
60. **target: yellow\_light**
61. color: "yellow"
62. }
63. },
65. **State {**
66. name: "yellow\_lights"
67. PropertyChanges {
68. target: yellow\_light
69. color: "yellow"
70. **}**
71. },
73. State {
74. name: "green\_lights"
75. **PropertyChanges {**
76. target: green\_light
77. color: "green"
78. }
79. }
80. **]**
82. transitions: [
83. Transition {
84. from: "red\_lights"
85. **to: "switch\_to\_green"**
86. SequentialAnimation {
87. PauseAnimation {
88. duration: 3000
89. }
90. **ColorAnimation {**
91. targets: [red\_light, yellow\_light, green\_light]
92. property: "color"
93. duration: 0
94. }
95. **ScriptAction {**
96. script: change\_state()
97. }
98. }
99. },
101. Transition {
102. from: "switch\_to\_green"
103. to: "green\_lights"
104. SequentialAnimation {
105. **PauseAnimation {**
106. duration: 1000
107. }
108. ColorAnimation {
109. targets: [red\_light, yellow\_light, green\_light]
110. **property: "color"**
111. duration: 0
112. }
113. ScriptAction {
114. script: change\_state()
115. **}**
116. }
117. },
119. Transition {
120. **from: "green\_lights"**
121. to: "yellow\_lights"
123. SequentialAnimation {
124. PauseAnimation {
125. **duration: 3000**
126. }
127. ColorAnimation {
128. targets: [red\_light, yellow\_light, green\_light]
129. property: "color"
130. **duration: 0**
131. }
133. ScriptAction {
134. script: change\_state()
135. **}**
136. }
137. },
139. Transition {
140. **from: "yellow\_lights"**
141. to: "red\_lights"
143. SequentialAnimation {
144. PauseAnimation {
145. **duration: 1000**
146. }
147. ColorAnimation {
148. targets: [red\_light, yellow\_light, green\_light]
149. property: "color"
150. **duration: 0**
151. }
152. ScriptAction {
153. script: change\_state()
154. }
155. **}**
156. }
157. ]
159. **function** change\_state() {
160. **switch(traffic\_light.state) {**
161. **case** "red\_lights":
162. traffic\_light.state = "switch\_to\_green";
163. **break**;
164. **case** "switch\_to\_green":
165. **traffic\_light.state = "green\_lights";**
166. **break**;
167. **case** "green\_lights":
168. traffic\_light.state = "yellow\_lights";
169. **break**;
170. **case "yellow\_lights":**
171. traffic\_light.state = "red\_lights";
172. **break**;
173. }
174. }
175. **}**