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 №3:

**«Основы конструирования интерфейсов пользователя»**

**Выполнила:** студентка группы 381606-2

Колосова Евгения Александровна

**Научный руководитель:** Доцент кафедры программной инженерии

Борисов Николай Анатольевич

Нижний Новгород  
2019

Введение

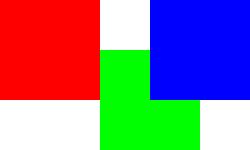
В данной лабораторной работе представлена задача конструирования основных элементов пользовательских интерфейсов в Sailfish OS.

Целью работы является освоение основных навыков работы интуитивно понятными пользователю интерфейсами, методы позиционирования, отрисовки и перемещения элементов. Также нужно научиться анимировать различные элементы и создавать диалоги, а также взаимодействие с ними.

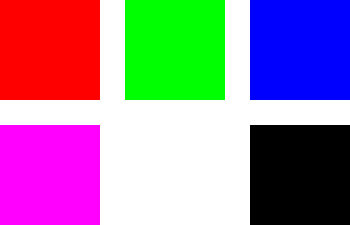
Постановка задачи

Для достижения в результате выполнения работы положительного результата, проект следует реализовывать по следующим пунктам:

1. Создать новый проект со стандартной заготовкой приложения.
2. Нарисовать 3 квадрата красного, зелёного и синего цветов следующим образом:



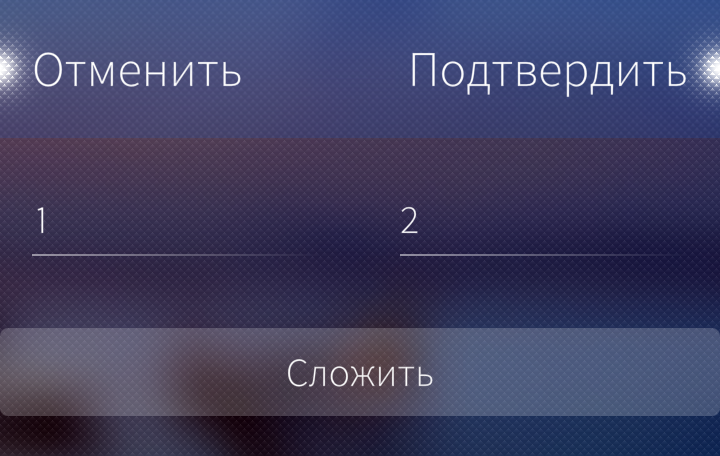
1. Нарисовать 5 квадратов с использованием Column и Row следующим образом:



1. Нарисовать те же 5 квадратов с использованием Grid.
2. Сделать из квадрата “A” прямоугольник “B” с использованием объектов Translate, Scale и Rotation.

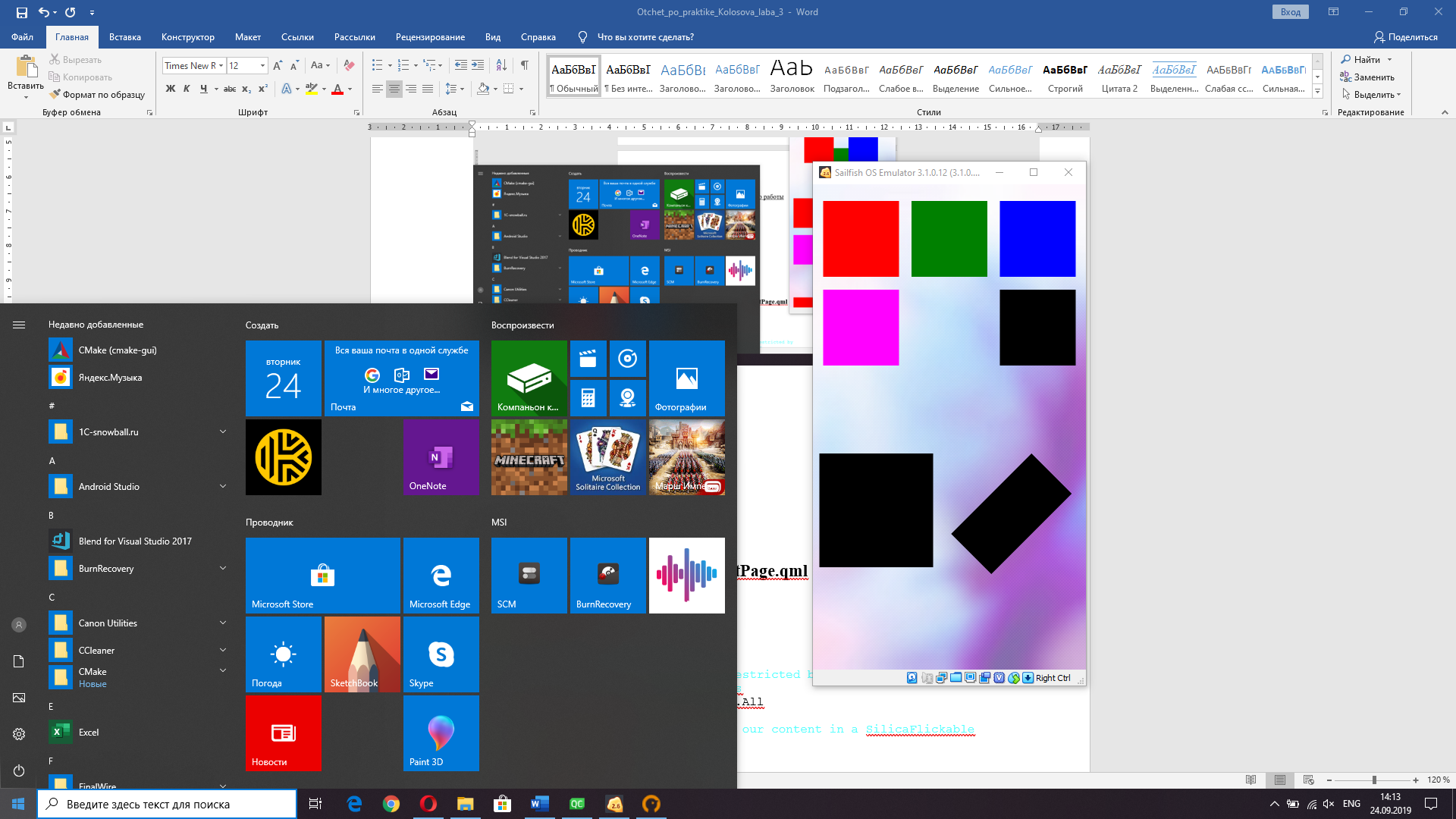
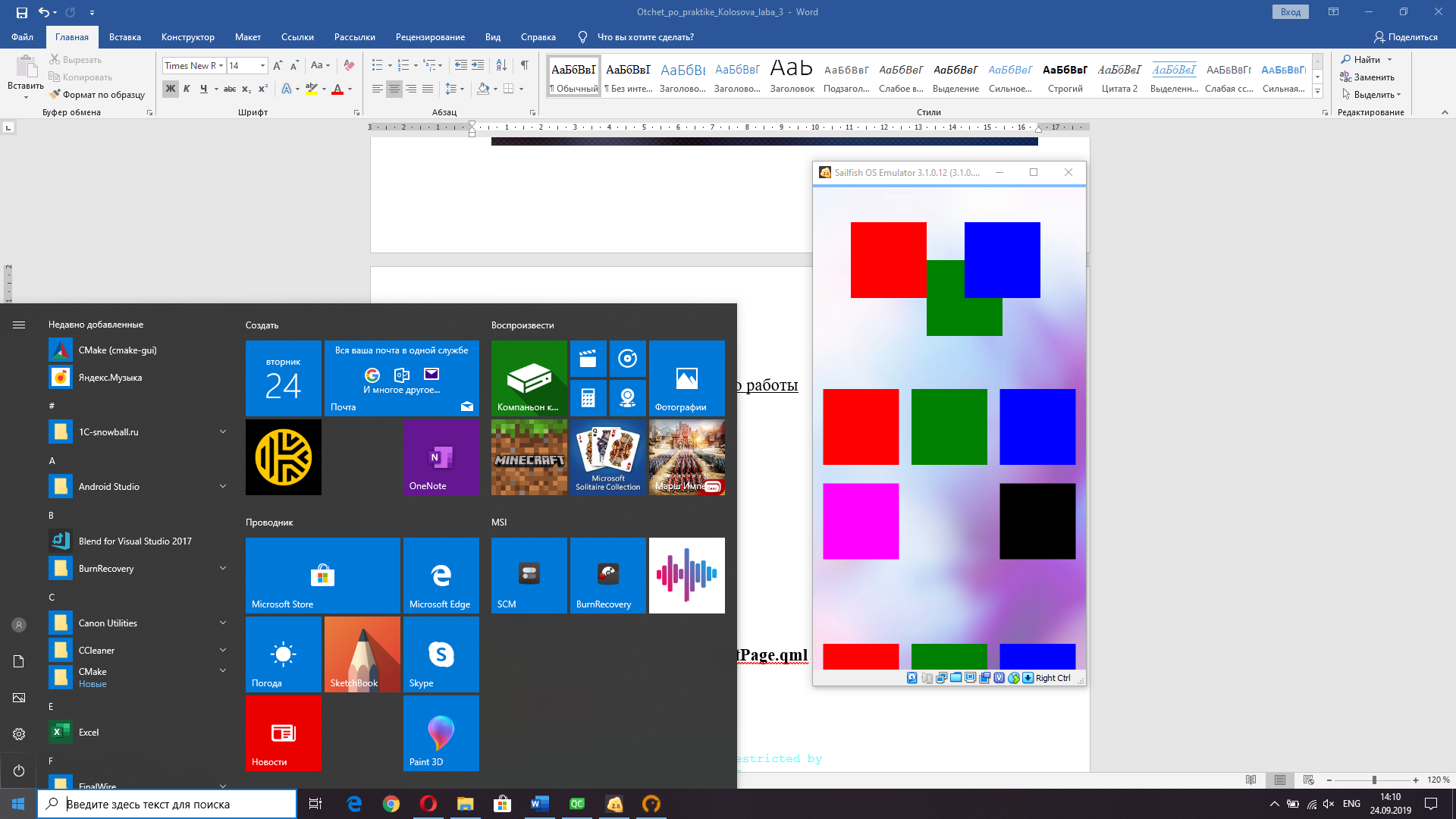


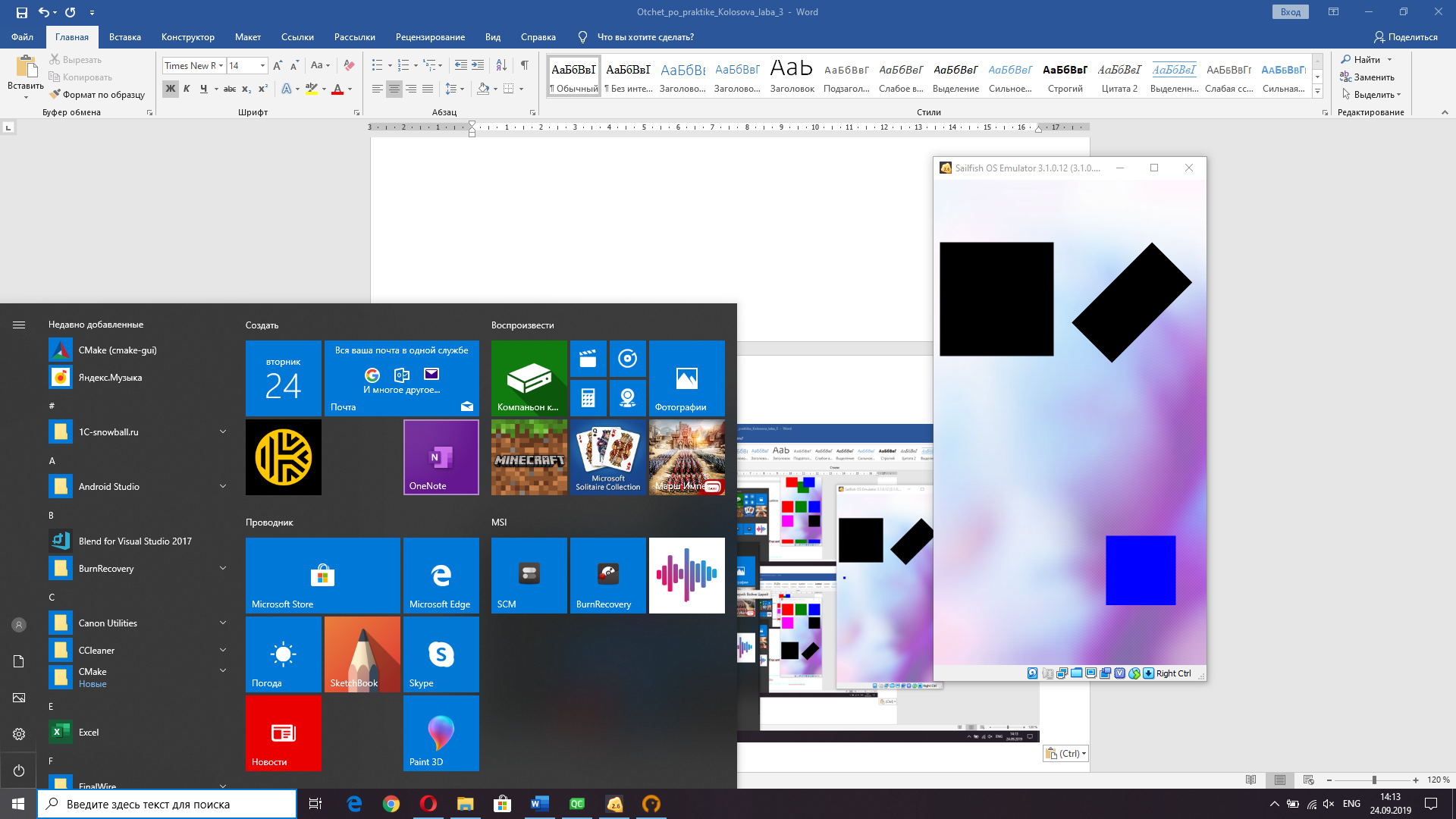
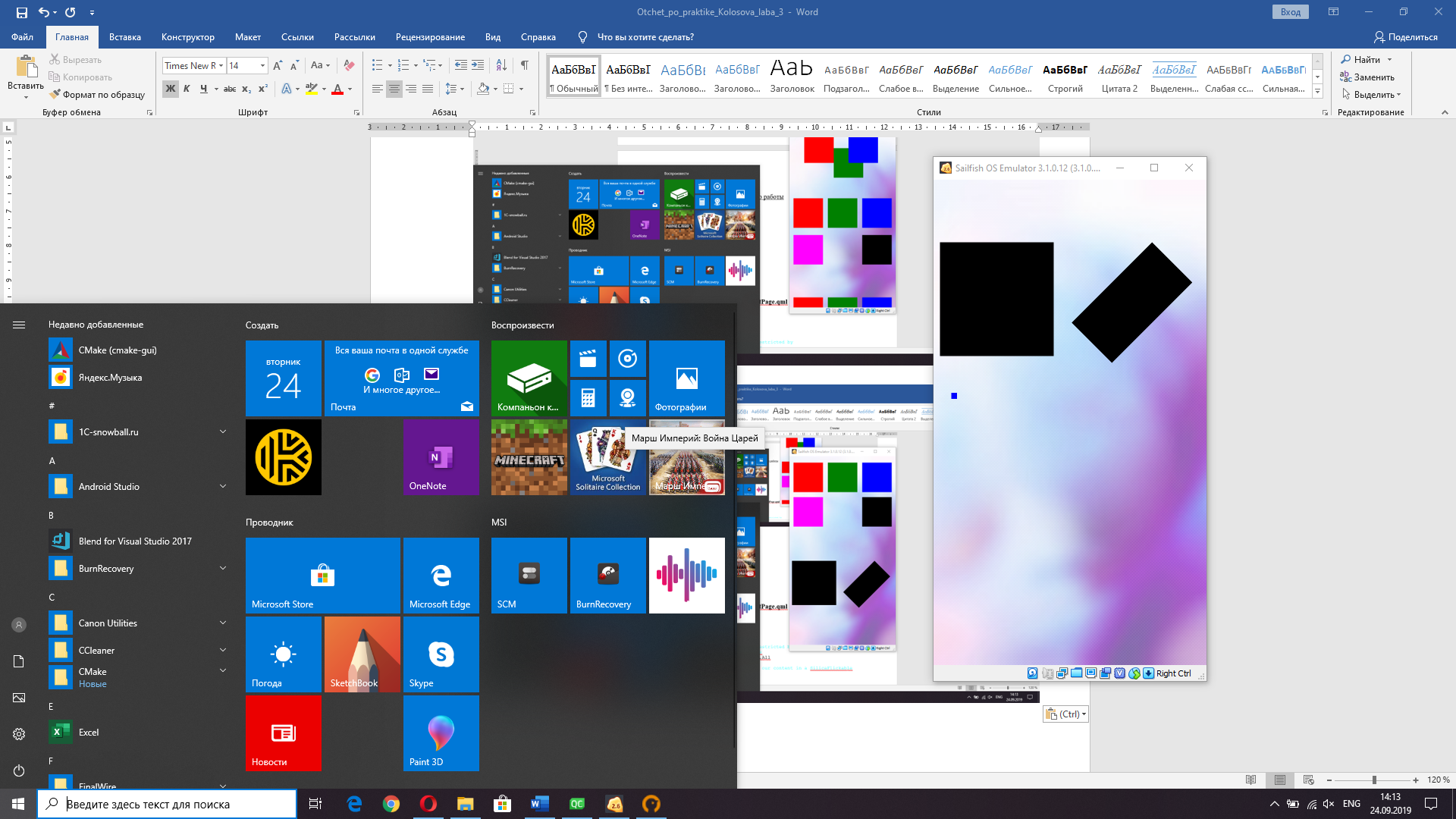
1. Нарисовать квадрат и анимировать его перемещение вниз с увеличением его размера. Документация по анимации доступна по адресу <http://doc.qt.io/qt-5/qml-qtquick-animation.html>.
2. Реализовать диалог с двумя текстовыми полями, в которые вводятся числа. После нажатия на кнопку “Подтвердить” в консоль выводится сумма чисел. Для преобразования строк к числам использовать функцию *parseInt(“42”).* Валидацией и обработкой ошибок можно пренебречь.

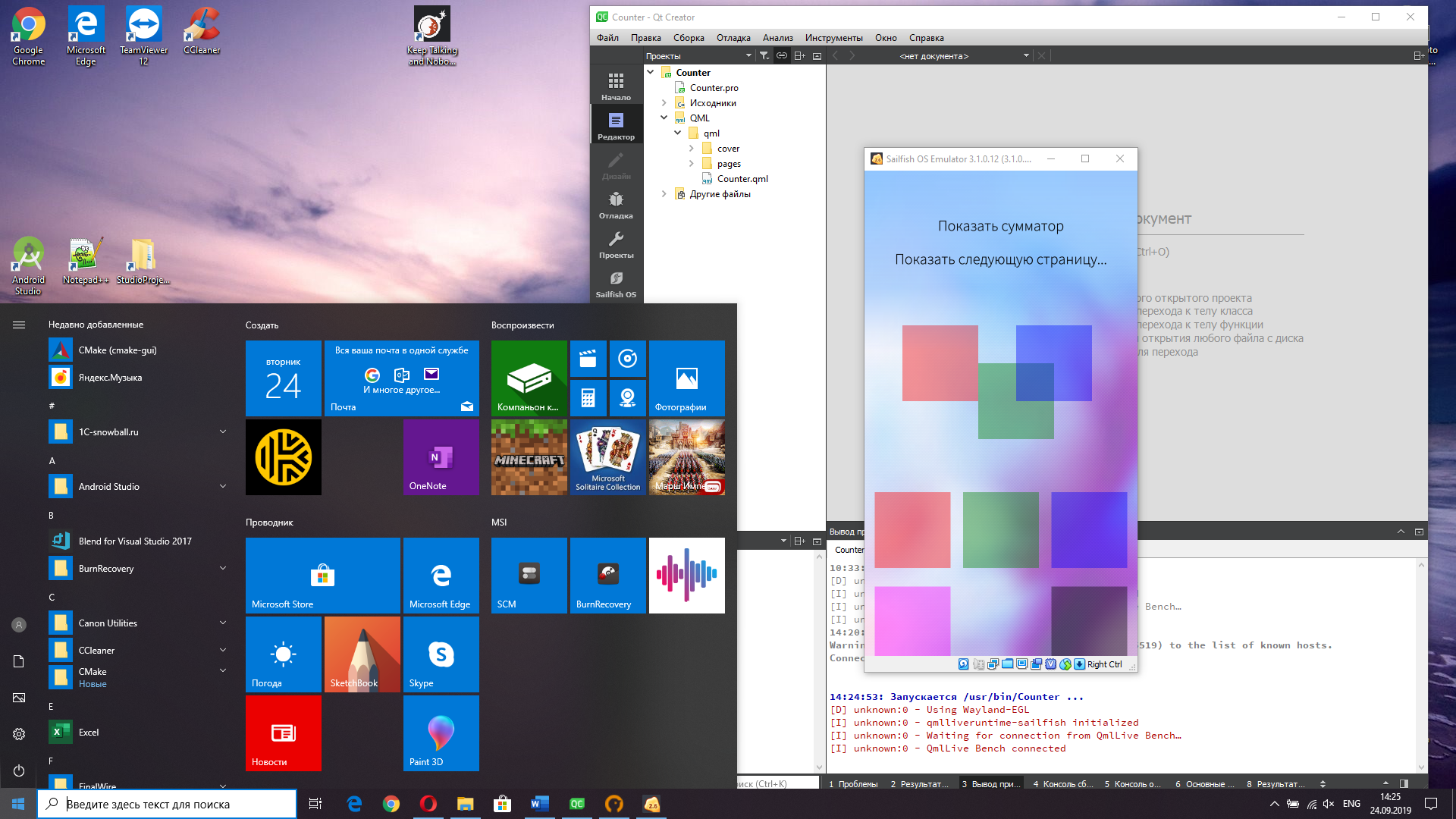


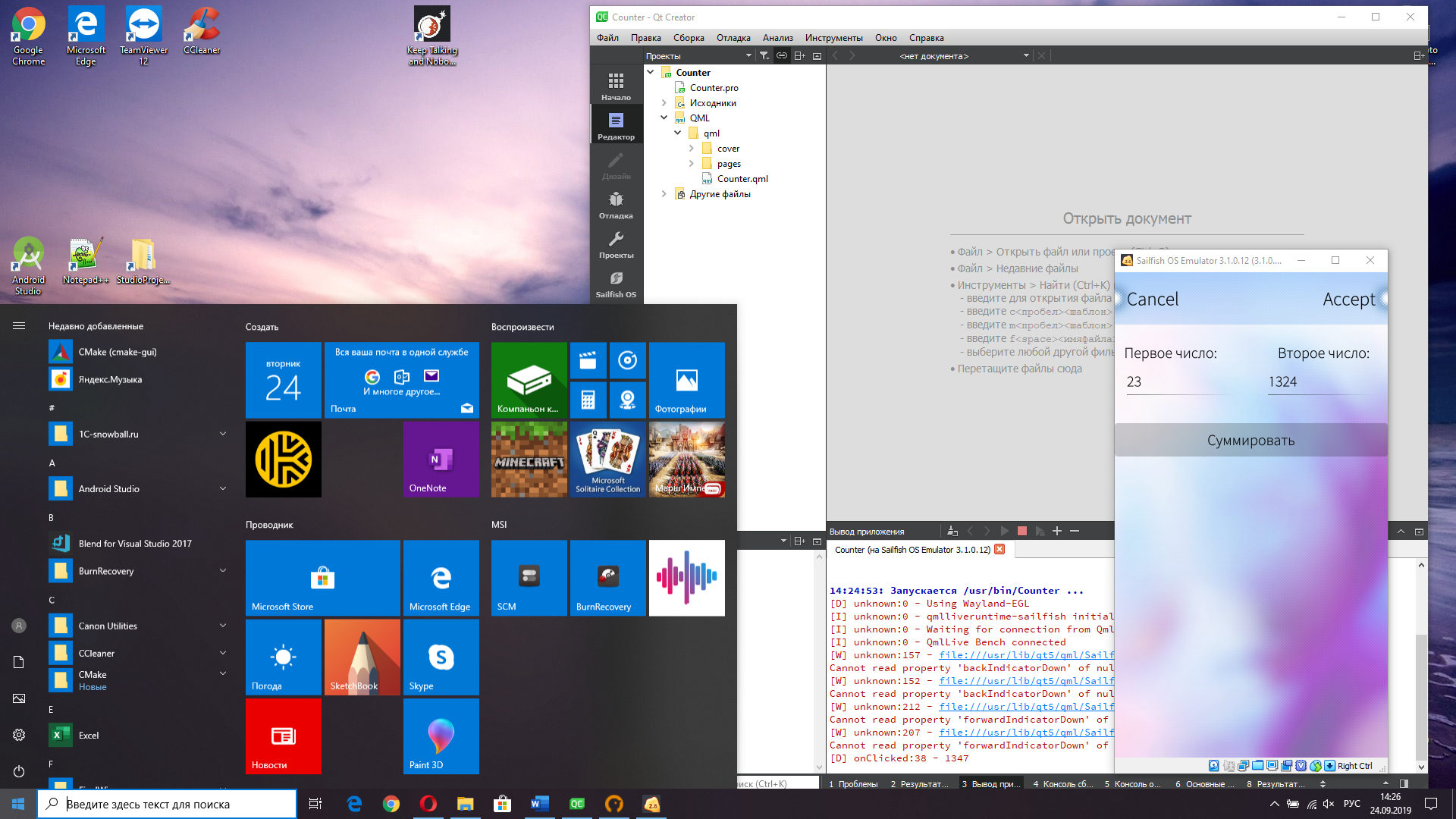
Приложение

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 задаче этой лабораторной работы. А значит полностью освоен навык конструирования пользовательских интерфейсов с нужными элементами, анимацией и диалогами.

Код проекта:

FirstPage.qml

1. import QtQuick 2.0
2. import Sailfish.Silica 1.0
4. Page {
5. **id: page**
6. allowedOrientations: Orientation.All
7. SilicaFlickable {
8. anchors.fill: parent
9. contentHeight: column.height
11. PullDownMenu {
12. MenuItem {
13. text: qsTr("Показать сумматор")
14. onClicked: pageStack.push(Qt.resolvedUrl("SecondPage.qml"))
15. **}**
16. MenuItem {
17. text: qsTr("Показать следующую страницу...")
18. onClicked: pageStack.push(Qt.resolvedUrl("ThirdPage.qml"))
19. }
20. **}**
22. Column {
23. id: column
24. width: page.width
25. **property int sizeRect: 200**
26. spacing: Theme.paddingLarge
27. Rectangle {
28. width: page.width
29. height: column.sizeRect \* 2.5
30. **color: "transparent"**
31. Rectangle {
32. width: column.sizeRect
33. height: column.sizeRect
34. x: column.sizeRect/2
35. **y: column.sizeRect/2**
36. color: "red"
37. Rectangle {
38. width: column.sizeRect
39. height: column.sizeRect
40. **x: column.sizeRect**
41. y: column.sizeRect/2
42. color: "green"
43. Rectangle {
44. width: column.sizeRect
45. **height: column.sizeRect**
46. x: column.sizeRect/2
47. y: -column.sizeRect/2
48. color: "blue"
49. }
50. **}**
51. }
52. }
53. Rectangle {
54. width: parent.width
55. **height: column.sizeRect + column.sizeRect/12**
56. color: "transparent"
57. Row {
58. anchors.horizontalCenter: parent.horizontalCenter
59. anchors.verticalCenter: parent.verticalCenter
60. **spacing: column.sizeRect / 6;**
61. Rectangle {
62. width: column.sizeRect;
63. height: column.sizeRect
64. color: "red"
65. **}**
66. Rectangle {
67. width: column.sizeRect;
68. height: column.sizeRect
69. color: "green"
70. **}**
71. Rectangle {
72. width: column.sizeRect;
73. height: column.sizeRect
74. color: "blue"
75. **}**
76. }
77. }
78. Rectangle {
79. width: parent.width
80. **height: column.sizeRect + column.sizeRect/12**
81. color: "transparent"
82. Row {
83. anchors.horizontalCenter: parent.horizontalCenter
84. anchors.verticalCenter: parent.verticalCenter
85. **spacing: column.sizeRect / 6;**
86. Rectangle {
87. width: column.sizeRect;
88. height: column.sizeRect
89. color: "magenta"
90. **}**
91. Rectangle {
92. width: column.sizeRect;
93. height: column.sizeRect
94. color: "transparent"
95. **}**
96. Rectangle {
97. width: column.sizeRect;
98. height: column.sizeRect
99. color: "black"
100. **}**
101. }
102. }
103. Rectangle {
104. width: parent.width
105. **height: column.sizeRect \* 2 + column.sizeRect \* 2**
106. color: "transparent"
107. Grid{
108. anchors.horizontalCenter: parent.horizontalCenter
109. anchors.verticalCenter: parent.verticalCenter
110. **columns: 3**
111. rows: 2
112. spacing: column.sizeRect / 6
113. Rectangle {
114. width: column.sizeRect;
115. **height: column.sizeRect**
116. color: "red"
117. }
118. Rectangle {
119. width: column.sizeRect;
120. **height: column.sizeRect**
121. color: "green"
122. }
123. Rectangle {
124. width: column.sizeRect;
125. **height: column.sizeRect**
126. color: "blue"
127. }
128. Rectangle {
129. width: column.sizeRect;
130. **height: column.sizeRect**
131. color: "magenta"
132. }
133. Rectangle {
134. width: column.sizeRect;
135. **height: column.sizeRect**
136. color: "transparent"
137. }
138. Rectangle {
139. width: column.sizeRect;
140. **height: column.sizeRect**
141. color: "black"
142. }
143. }
144. }
145. **Rectangle {**
146. width: parent.width
147. height : column.sizeRect\*1.5
148. color: "transparent"
149. Rectangle {
150. **id: startRect**
151. width: column.sizeRect\*1.5
152. height: column.sizeRect\*1.5
153. x: width/18
154. y: height/18
155. **color: "black"**
156. Rectangle {
157. width: parent.width
158. height: parent.height
159. color: parent.color
160. **z: 1**
161. transform: [Scale { xScale: 0.5; yScale: 1 }, Rotation { angle: 45}, Translate {x: column.sizeRect \* 2.8} ]
162. }
163. }
164. }
165. **Rectangle {**
166. width: parent.width
167. height: column.sizeRect \* 4;
168. color: "transparent"
169. Rectangle {
170. **id: scaleRect;**
171. y: 50;
172. x: 10;
173. width: 0;
174. height: 0;
175. **color: "blue";**
176. SequentialAnimation {
177. loops: Animation.Infinite;
178. id: animation;
179. running: **true**
180. **ParallelAnimation {**
181. PropertyAnimation {
182. target: scaleRect;
183. property: "y";
184. from: 50; to: page.width - column.sizeRect - 25; duration: 2500;
185. **}**
186. PropertyAnimation {
187. target: scaleRect;
188. property: "x";
189. from: 10; to: page.width - column.sizeRect - 25; duration: 2500;
190. **}**
191. PropertyAnimation {
192. target: scaleRect;
193. property: "width";
194. from: 0; to: column.sizeRect; duration: 2500;
195. **}**
196. PropertyAnimation {
197. target: scaleRect;
198. property: "height";
199. from: 0; to: column.sizeRect; duration: 2500;
200. **}**
201. }
202. ParallelAnimation {
203. PropertyAnimation {
204. target: scaleRect;
205. **property: "y";**
206. from: parent.width - column.sizeRect - 50; to: 50; duration: 2500;
207. }
208. PropertyAnimation {
209. target: scaleRect;
210. **property: "x";**
211. from: page.width - column.sizeRect - 25; to: 10; duration: 2500;
212. }
213. PropertyAnimation {
214. target: scaleRect;
215. **property: "width";**
216. from: column.sizeRect; to: 0; duration: 2500;
217. }
218. PropertyAnimation {
219. target: scaleRect;
220. **property: "height";**
221. from: column.sizeRect; to: 0; duration: 2500;
222. }
223. }
224. }
225. **}**
226. }
227. }
228. }
229. }

SecondPage.qml

1. import QtQuick 2.0
2. import Sailfish.Silica 1.0
4. Dialog {
5. **id: dialog;**
6. Column {
7. anchors.fill: parent;
8. id: column;
9. width: parent.width;
10. **spacing: Theme.paddingMedium;**
11. DialogHeader {}
12. Row {
13. spacing: 160;
14. x: 25;
15. **Label {**
16. text: "Первое число:"
17. }
18. Label {
19. text: "Второе число:"
20. **}**
21. }
22. Row {
23. width: parent.width;
24. spacing: 25;
25. **TextField {**
26. id: number1;
27. width: parent.width/2 - 12.5;
28. }
29. TextField {
30. **id: number2;**
31. width: parent.width/2 - 12.5;
32. }
33. }
34. Button {
35. **width: parent.width;**
36. text: "Суммировать";
37. onClicked: {
38. console.log(parseInt(number1.text) + parseInt(number2.text));
39. }
40. **}**
41. }
42. }