

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НЕФТИ И ГАЗА  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)  
ИМЕНИ И.М. ГУБКИНА»

ФАКУЛЬТЕТ КОМПЛЕКСНОЙ БЕЗОПАСНОСТИ ТЭК  
КАФЕДРА БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

**Лабораторная работа №15**  
по дисциплине «Специализированные языки и технологии  
программирования»  
на тему «Работа с ресурсами Qt, стили и кастомизация»

Выполнил студент:  
группы КА-22-06  
Воронин Алексей Дмитриевич

Преподаватель:  
Греков Владимир Сергеевич

Москва, 2025

Оглавление	
Цель работы .....	3
Задание 1: Анализ примерного проекта.....	4
Задание 2: Реализация дополнительного функционала .....	4
ЗАКЛЮЧЕНИЕ .....	9
Контрольные вопросы .....	10

## **Цель работы**

- Изучить систему ресурсов Qt (qrc)
- Освоить механизмы создания и применения анимаций
- Научиться кастомизировать интерфейс приложения
- Разработать расширение функциональности примерного проекта

## **Задание 1: Анализ примерного проекта**

1. Изучить структуру проекта [projects/animations\\_aurora · main · omprussia / Education / Разработка приложений на Qt · GitLab](https://github.com/omprussia/Animations_Aurora)
2. Проанализировать используемые ресурсы
3. Разобрать реализованные анимации

## **Задание 2: Реализация дополнительного функционала**

1. Добавление новых типов анимаций:
  - Эффекты перехода между состояниями
  - Сложные составные анимации
2. Создание кастомных компонентов:
  - Анимированные кнопки
  - Интерактивные элементы управления

```
import QtQuick 2.0
import Sailfish.Silica 1.0

Page {
    id: page
    allowedOrientations: Orientation.All

    property color color1: "red"
    property color color2: "green"
    property bool isRed: true
    property real orbitRadius: 100
    property int orbitDuration: 1000
    property int orbitLoops: 5

    Rectangle {
        id: circle
        width: Theme.itemSizeExtraLarge
        height: Theme.itemSizeExtraLarge
        radius: width / 2
        color: color1
        anchors.centerIn: parent
        scale: 1
        Behavior on color { ColorAnimation { duration: 800 } }
    }

    // Пулсация
    SequentialAnimation {
        id: pulseAnim
        loops: 3
        NumberAnimation { target: circle; property: "scale"; to: 1.3;
            duration: 200 }
        NumberAnimation { target: circle; property: "scale"; to: 1.0;
            duration: 200 }
    }
}
```

```

Item {
  id: orbitContainer
  anchors.fill: parent
  visible: false

  Rectangle {
    id: orbCircle1
    width: Theme.itemSizeExtraLarge
    height: Theme.itemSizeExtraLarge
    radius: width / 2
    color: circle.color
  }

  Rectangle {
    id: orbCircle2
    width: Theme.itemSizeLarge
    height: Theme.itemSizeLarge
    radius: width / 2
    color: circle.color
  }

  Path {
    id: orbitPath1
    startX: page.width / 2 + orbitRadius
    startY: page.height / 2
    PathArc {
      x: page.width / 2 - orbitRadius
      y: page.height / 2
      radiusX: orbitRadius
      radiusY: orbitRadius
    }
    PathArc {
      x: page.width / 2 + orbitRadius
      y: page.height / 2
      radiusX: orbitRadius
      radiusY: orbitRadius
    }
  }

  Path {
    id: orbitPath2
    startX: page.width / 2 - orbitRadius
    startY: page.height / 2
    PathArc {
      x: page.width / 2 + orbitRadius
      y: page.height / 2
      radiusX: orbitRadius
      radiusY: orbitRadius
    }
    PathArc {
      x: page.width / 2 - orbitRadius
      y: page.height / 2
      radiusX: orbitRadius
      radiusY: orbitRadius
    }
  }

  ParallelAnimation {
    id: orbitGroup

```

```

loops: orbitLoops
onStopped: {
    orbitContainer.visible = false
    circle.visible = true
}

PathAnimation {
    target: orbCircle1
    path: orbitPath1
    duration: orbitDuration
    orientation: PathAnimation.NoOrientation
}

PathAnimation {
    target: orbCircle2
    path: orbitPath2
    duration: orbitDuration
    orientation: PathAnimation.NoOrientation
}
}
}

SequentialAnimation {
    id: colorChangeAnim
    PropertyAnimation { target: circle; property: "color"; to:
    "black"; duration: 500 }
    PropertyAnimation {
        target: circle
        property: "color"
        to: isRed ? color2 : color1
        duration: 500
    }
    onStopped: isRed = !isRed
}

Column {
    anchors.horizontalCenter: parent.horizontalCenter
    anchors.bottom: parent.bottom
    anchors.bottomMargin: Theme.paddingLarge
    spacing: Theme.paddingMedium

    Button {
        text: "Пульсация"
        onClicked: pulseAnim.start()
    }

    Button {
        text: "Вращение"
        onClicked: {

            orbCircle1.color = circle.color
            orbCircle2.color = circle.color
            circle.visible = false
            orbitContainer.visible = true
            orbitGroup.stop()
            orbitGroup.start()
        }
    }

    Button {
        text: "Изменить цвет"
    }
}

```

```
    onClicked: colorChangeAnim.start()
}
}
}
```

При нажатии на кнопку вращения появляется второй круг, и они начинают вращаться вокруг общего центра. При нажатии на Пульсацию круг начинает увеличиваться и уменьшаться в размерах. При нажатии на изменение цвета круг плавно становится черным, а потом так же плавно зеленым.

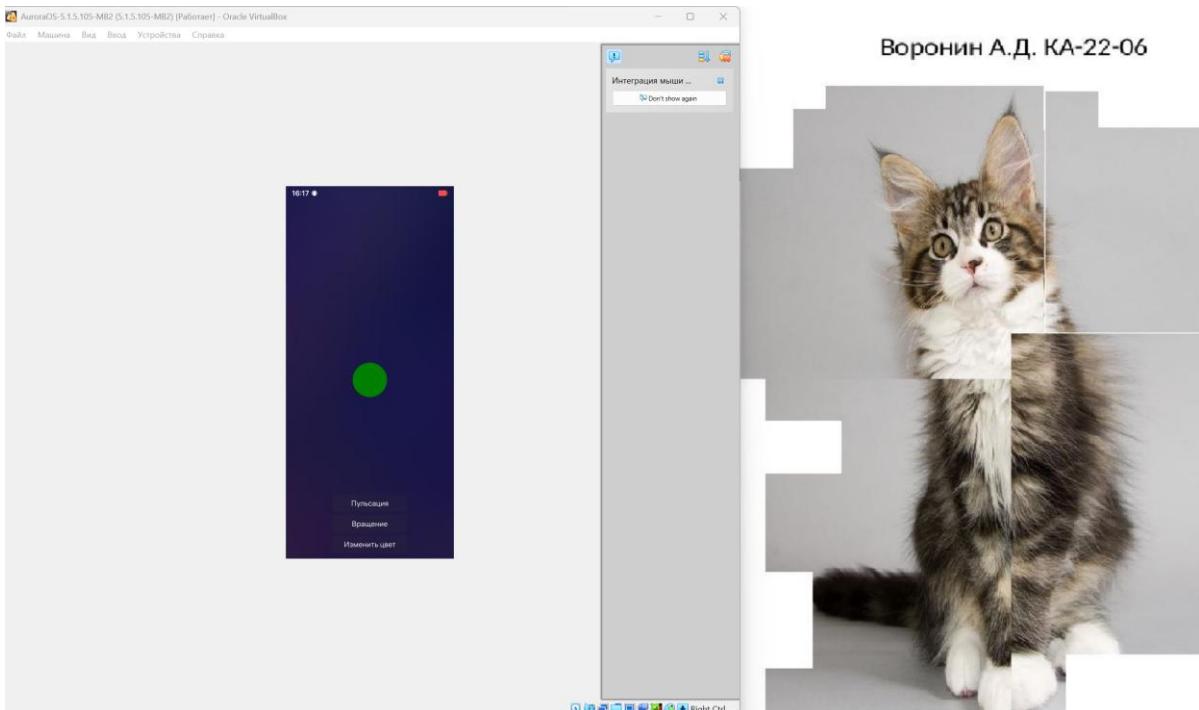
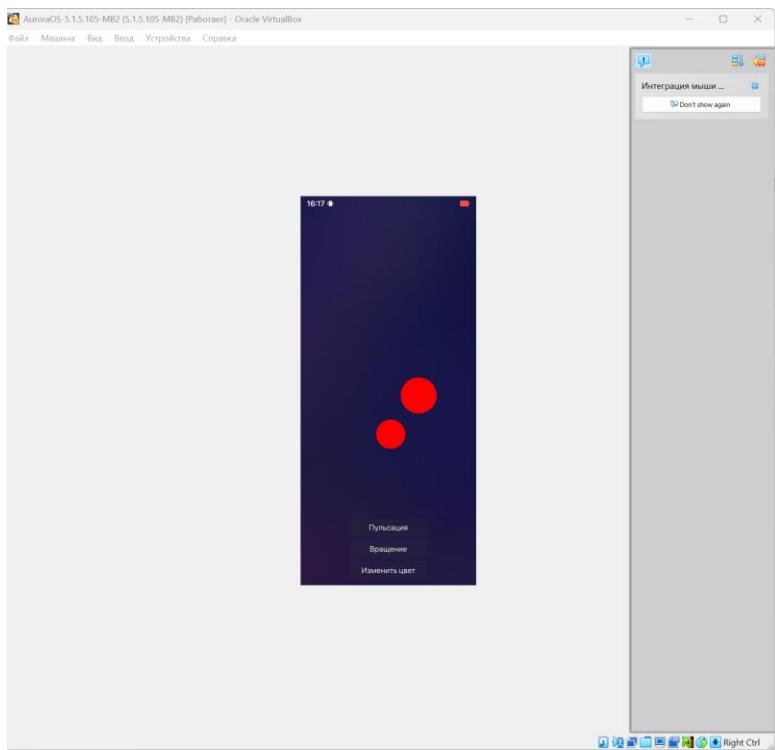


Рисунок 1 – Изменение цвета



Воронин А.Д. КА-22-06

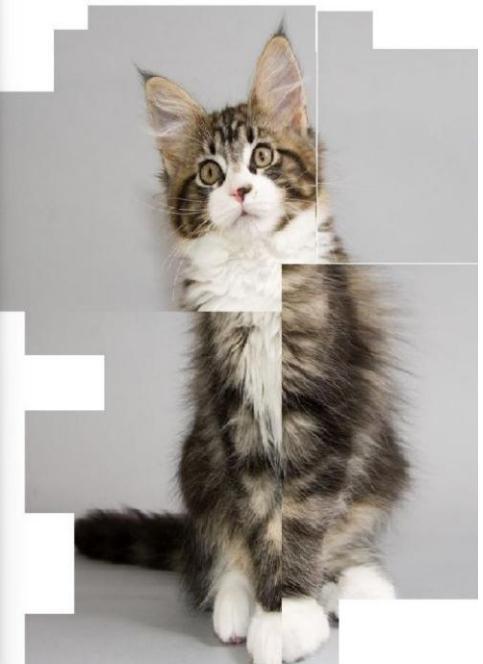


Рисунок 2 – Вращение

## **ЗАКЛЮЧЕНИЕ**

В заключение, подводя итог проделанной работе, все поставленные в начале цели и задания данной лабораторной работы были достигнуты в полном объеме.

## **Контрольные вопросы**

### **1. Как организована система ресурсов в Qt?**

Система ресурсов в Qt организована с помощью файла коллекции ресурсов (.qrc). В процессе сборки проекта файл ресурсов подключается через систему сборки, что обеспечивает автоматическую упаковку ресурсов вместе с приложением. В коде приложения доступ к ресурсам осуществляется через специальные пути, начинающиеся с префикса :/.

### **2. Какие типы анимаций вы использовали в работе?**

- SequentialAnimation
- NumberAnimation
- PropertyAnimation

### **3. Как реализована кастомизация стандартных компонентов?**

Брали стандартные элементы (Button, Page, Column) в свойствах меняли им параметры (text, color).