

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
Направление подготовки: «Фундаментальная информатика и
информационные технологии»

Отчет по зачетному проекту
«Инструменты разработки мобильных приложений»

Выполнил:
студент группы 381908-4
Рябцев М. В.

Нижний Новгород
2022 г.

Оглавление

Постановка задачи	3
Руководство программиста.....	5
Руководство пользователя.....	7
Вывод.....	10
Приложение.....	11

Постановка задачи

Создать приложение-игру, в которой нужно перетаскивать фигуры в нужные отверстия, чтобы выполнить цель. Приложение будет использовать технологию Drag'n'Drop.

Руководство программиста

Приложение начинается с создания переменных для хранения состояний фигур:

```
property bool figure1: false
property bool figure2: false
property bool figure3: false
property bool figure4: false
```

Затем, создаем зоны DropArea на каждую фигуру, куда нужно будет их перетаскивать:

```
DropArea {
  id: drop1
  x: 100; y: 100
  width: 100; height: 200

  Rectangle {
    anchors.fill: parent
    color: parent.containsDrag ? "green" : "grey"
  }
}
```

Создаем сами фигуры:

```
Rectangle {
  id: blueRect
  x: 250; y: 700
  width: 100; height: 200
  color: "blue"
  border.color: "lightgreen"
  border.width: dragArea.drag.active ? 10 : 0

  Drag.active: dragArea.drag.active
  Drag.hotSpot.x: 25
  Drag.hotSpot.y: 100

  MouseArea {
    id: dragArea
    anchors.fill: parent
    // drag.axis: "YAxis"
    drag.target: parent
    onReleased: {
      if (drop1.containsDrag) {
        parent.parent.figure1 = true
      } else {
        parent.parent.figure1 = false
      }
    }
  }
}
```

И, наконец, создаем надпись с индикаторами, стоят ли фигуры на своих местах:

```

Label {
  id: label
  text: parent.figure1 && parent.figure2 && parent.figure3 &&
parent.figure4 ? "Готово!" : "Перетащите все фигуры на места"
  anchors.centerIn: parent
}

Row {
  anchors.horizontalCenter: parent.horizontalCenter
  y: 600
  spacing: 10

  Rectangle {
    width: 20
    height: width
    radius: 10
    color: parent.parent.figure1 ? "lightgreen" : "red"
  }

  Rectangle {
    width: 20
    height: width
    radius: 10
    color: parent.parent.figure2 ? "lightgreen" : "red"
  }

  Rectangle {
    width: 20
    height: width
    radius: 10
    color: parent.parent.figure3 ? "lightgreen" : "red"
  }

  Rectangle {
    width: 20
    height: width
    radius: 10
    color: parent.parent.figure4 ? "lightgreen" : "red"
  }
}

```

Руководство пользователя

После запуска программы пользователем, открывается страница с фигурами, разбросанными по полю. Задача – перетащить фигуры на нужные места.

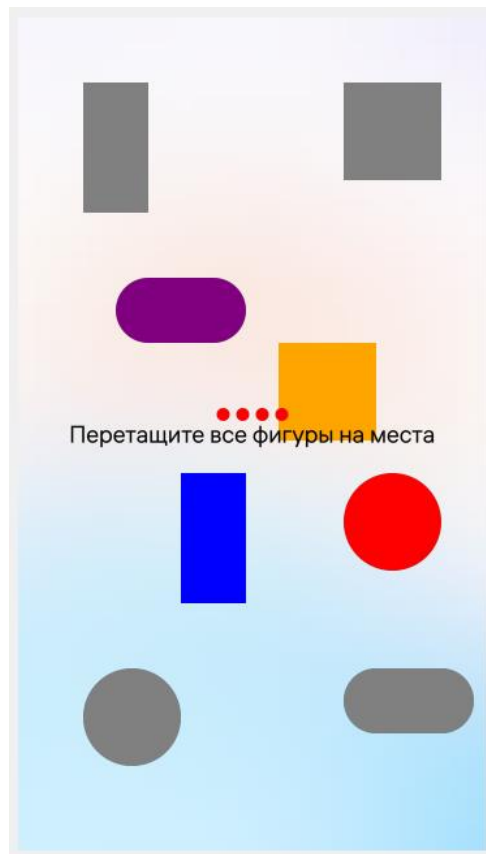


Рис 1. Выбор месяца

После того, как все фигуры окажутся на своих местах, индикаторы загорятся зеленым цветом.

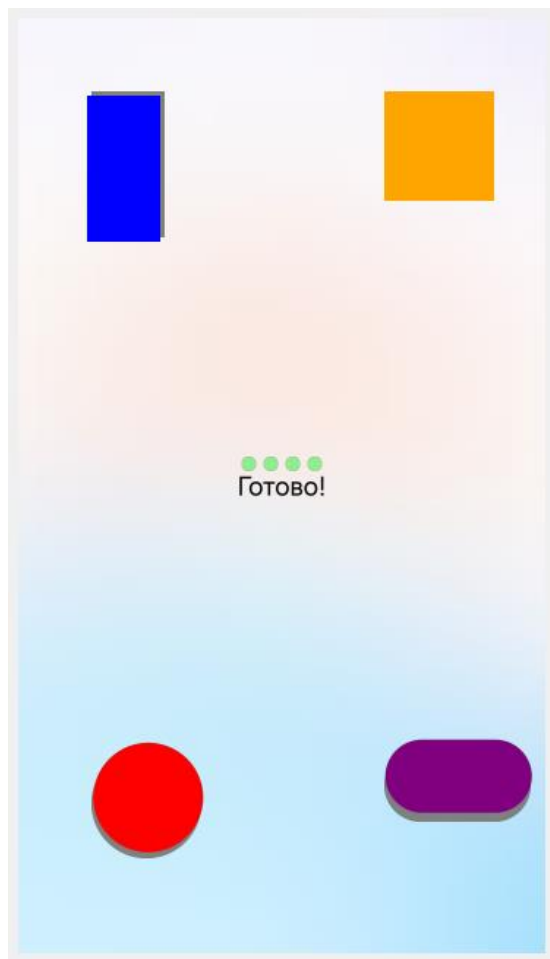


Рис 2. Список дней

Вывод

В процессе выполнения данной лабораторной работы я создал собственное приложение, реализующее игру.

Приложение

```
import QtQuick 2.0
import Sailfish.Silica 1.0
import "."

Page {

    property bool figure1: false
    property bool figure2: false
    property bool figure3: false
    property bool figure4: false

    DropArea {
        id: drop1
        x: 100; y: 100
        width: 100; height: 200

        Rectangle {
            anchors.fill: parent
            color: parent.containsDrag ? "green" : "grey"
        }
    }

    Rectangle {
        id: blueRect
        x: 250; y: 700
        width: 100; height: 200
        color: "blue"
        border.color: "lightgreen"
        border.width: dragArea.drag.active ? 10 : 0

        Drag.active: dragArea.drag.active
        Drag.hotSpot.x: 25
        Drag.hotSpot.y: 100

        MouseArea {
            id: dragArea
            anchors.fill: parent
            // drag.axis: "YAxis"
            drag.target: parent
            onReleased: {
                if (drop1.containsDrag) {
                    parent.parent.figure1 = true
                } else {
                    parent.parent.figure1 = false
                }
            }
        }
    }

    DropArea {
        id: drop2
        x: 500; y: 100
        width: 150; height: 150

        Rectangle {
            anchors.fill: parent
            color: parent.containsDrag ? "green" : "grey"
        }
    }
}
```

```

Rectangle {
    id: orangeRect
    x: 400; y: 500
    width: 150; height: 150
    color: "orange"
    border.color: "lightgreen"
    border.width: dragArea2.drag.active ? 10 : 0

    Drag.active: dragArea2.drag.active
    Drag.hotSpot.x: 25
    Drag.hotSpot.y: 100

    MouseArea {
        id: dragArea2
        anchors.fill: parent
        // drag.axis: "YAxis"
        drag.target: parent
        onReleased: {
            if (drop2.containsDrag) {
                parent.parent.figure2 = true
            } else {
                parent.parent.figure2 = false
            }
        }
    }
}

DropArea {
    id: drop3
    x: 100; y: 1000
    width: 150; height: 150

    Rectangle {
        anchors.fill: parent
        color: parent.containsDrag ? "green" : "grey"
        radius: 75
    }
}

Rectangle {
    id: redCircle
    x: 500; y: 700
    width: 150; height: 150
    color: "red"
    radius: 75
    border.color: "lightgreen"
    border.width: dragArea3.drag.active ? 10 : 0

    Drag.active: dragArea3.drag.active
    Drag.hotSpot.x: 25
    Drag.hotSpot.y: 100

    MouseArea {
        id: dragArea3
        anchors.fill: parent
        // drag.axis: "YAxis"
        drag.target: parent
        onReleased: {
            if (drop3.containsDrag) {
                parent.parent.figure3 = true
            } else {
                parent.parent.figure3 = false
            }
        }
    }
}

```

```

    }
  }
}

DropArea {
  id: drop4
  x: 500; y: 1000
  width: 200; height: 100

  Rectangle {
    anchors.fill: parent
    color: parent.containsDrag ? "green" : "grey"
    radius: 100
  }
}

Rectangle {
  id: purpleOval
  x: 150; y: 400
  width: 200; height: 100
  color: "purple"
  radius: 100
  border.color: "lightgreen"
  border.width: dragArea4.drag.active ? 10 : 0

  Drag.active: dragArea4.drag.active
  Drag.hotSpot.x: 25
  Drag.hotSpot.y: 100

  MouseArea {
    id: dragArea4
    anchors.fill: parent
    // drag.axis: "YAxis"
    drag.target: parent
    onReleased: {
      if (drop4.containsDrag) {
        parent.parent.figure4 = true
      } else {
        parent.parent.figure4 = false
      }
    }
  }
}

Label {
  id: label
  text: parent.figure1 && parent.figure2 && parent.figure3 &&
parent.figure4 ? "Готово!" : "Перетащите все фигуры на места"
  anchors.centerIn: parent
}

Row {
  anchors.horizontalCenter: parent.horizontalCenter
  y: 600
  spacing: 10

  Rectangle {
    width: 20
    height: width
    radius: 10
    color: parent.parent.figure1 ? "lightgreen" : "red"
  }
}

```

```
Rectangle {
  width: 20
  height: width
  radius: 10
  color: parent.parent.figure2 ? "lightgreen" : "red"
}

Rectangle {
  width: 20
  height: width
  radius: 10
  color: parent.parent.figure3 ? "lightgreen" : "red"
}

Rectangle {
  width: 20
  height: width
  radius: 10
  color: parent.parent.figure4 ? "lightgreen" : "red"
}
}
```