

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 381908-2  
Чаусов Н. Д.

Нижний Новгород  
2022 г.

## Оглавление

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

## **Постановка задачи**

Создать приложение-игру «Ним», в котором игрок может играть с компьютером.

Игрок выбирает число предметов в кучке и затем по очереди с компьютером делает ходы, выбирая, какое число предметов убрать из кучки.

# Руководство программиста

Создаем игровое поле:

```
Grid {
  columns: 5
  id: grid
  spacing: 50
}
```

Функция заполнения поля выбранным числом предметов:

```
Component.onCompleted: {
  lastPos = count
  console.log(lastPos)

  for (var i = 0; i < count; i++) {
    const newObject = Qt.createQmlObject('
      import QtQuick 2.0

      Rectangle {
        width: 50
        height: 50
        radius: 25
        y: 500
      }
    ',
                                          grid
    )
    newObject.color = "#" +
Math.floor(Math.random()*16777215).toString(16);
  }
}
```

Функция хода:

```
function take(taken, isPlayer) {
  if (lastPos > 0) {
    for (var i = lastPos - 1; i > lastPos - taken - 1; i--) {
      grid.children[i].visible = false
    }
    lastPos = lastPos - taken
  }

  if (isPlayer) {
    if (lastPos === 0) {
      btnRow.visible = false
      returnBtn.visible = true
      annLabel.text = "Вы играли"
      return
    }
    annLabel.text = "Ходит компьютер"
    var randomAmount = parseInt(Math.random() * 3)
    for (i = 0; randomAmount > lastPos; i++) randomAmount--
    if (randomAmount === 0) randomAmount++

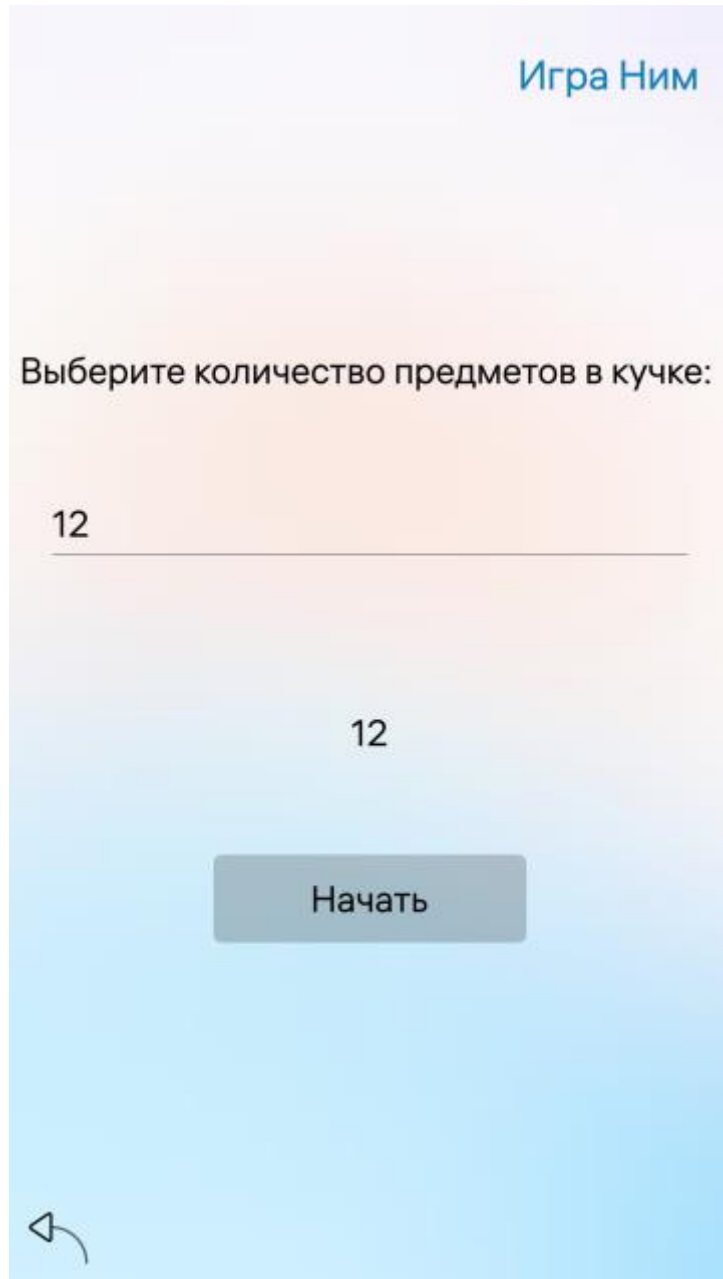
    delay(1000, function() {
      take(randomAmount, false)
    })
  }

  if (!isPlayer) {
```

```
    if (lastPos === 0) {  
        btnRow.visible = false  
        returnBtn.visible = true  
        annLabel.text = "Вы проиграли"  
        return  
    }  
    annLabel.text = "Ходит игрок"  
}  
}
```

# Руководство пользователя

После запуска программы пользователем, открывается страница с выбором числа предметов.



Игра Ним

Выберите количество предметов в кучке:

12

12

Начать

Рис 1. Выбор

Пользователю открывается игровое поле. Он начинает ход, выбирая число предметов.



Рис 2. Ход игрока

Затем ходит компьютер.



Рис 3. Ход компьютера



Побеждает тот, кто забирает последние предметы.

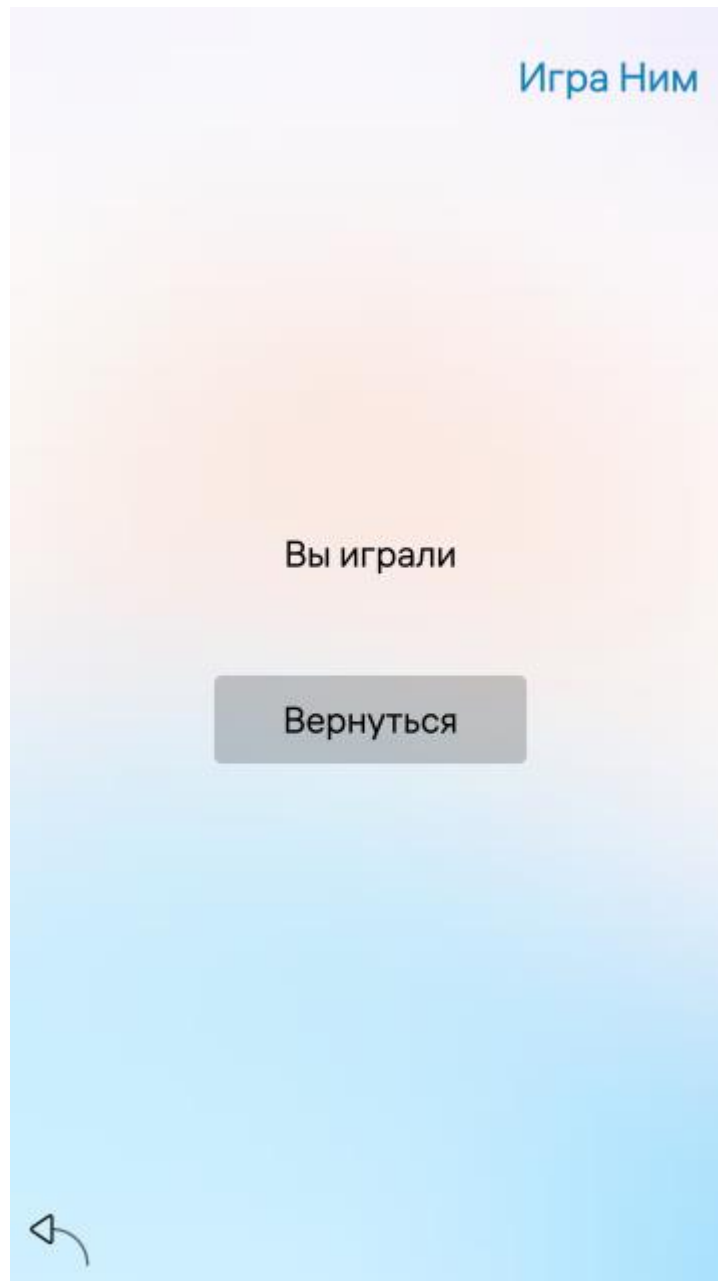


Рис 4. Победа

## **Вывод**

Выполняя данную лабораторную работу я создал приложение-игру, использующие различные технологии.

## Приложение

```
import QtQuick 2.0
import Sailfish.Silica 1.0

Page {
    PageHeader {
        objectName: "pageHeader"
        title: "Игра Ним"
    }

    IconButton {
        objectName: "aboutButton"
        icon.source: "image://theme/icon-m-rotate-left"
        anchors.bottom: parent.bottom
        anchors.left: parent.left

        onClicked: {
            pageStack.clear()
            pageStack.replace(Qt.resolvedUrl("../MainPage.qml"))
        }
    }

    Column {

        anchors.centerIn: parent
        spacing: 100

        Label {
            anchors.horizontalCenter: parent.horizontalCenter
            text: "Выберите количество предметов в кучке: "
        }

        TextField {
            anchors.horizontalCenter: parent.horizontalCenter
            placeholderText: "Количество"
            onTextChanged: count = parseInt(text)
        }

        Label {
            anchors.horizontalCenter: parent.horizontalCenter
            text: count
        }

        Button {
            anchors.horizontalCenter: parent.horizontalCenter
            text: "Начать"
            onClicked: {
                if (count > 1) {
                    pageStack.replace(Qt.resolvedUrl("GamePage.qml"))
                }
            }
        }
    }
}

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

Page {
```

```

property int lastPos: 0

PageHeader {
    objectName: "pageHeader"
    title: "Игра Ним"
}

IconButton {
    objectName: "aboutButton"
    icon.source: "image://theme/icon-m-rotate-left"
    anchors.bottom: parent.bottom
    anchors.left: parent.left

    onClicked: {
        pageStack.clear()
        pageStack.replace(Qt.resolvedUrl("../MainPage.qml"))
    }
}

Column {
    anchors.centerIn: parent

    spacing: 100

    Label {
        id: annLabel
        text: "Ходит игрок"
        anchors.horizontalCenter: parent.horizontalCenter
    }

    Grid {
        columns: 5
        id: grid
        spacing: 50

        Label {
            text: "Выберите число предметов"
            visible: btnRow.visible
        }

        Row {
            id: btnRow
            spacing: 20
            anchors.horizontalCenter: parent.horizontalCenter

            Button {
                text: "1"
                width: 100
                onClicked: take(1, true)
            }

            Button {
                text: "2"
                width: 100
                onClicked: take(2, true)
            }

            Button {
                text: "3"
                width: 100
                onClicked: take(3, true)
            }
        }
    }
}

```

```

    }
}

Button {
    anchors.horizontalCenter: parent.horizontalCenter
    text: "Вернуться"
    onClicked: pageStack.replace(Qt.resolvedUrl("MainPage.qml"))
    id: returnBtn
    visible: false
}

}

Timer {
    id: timer
}

function delay(delayTime, cb) {
    timer.interval = delayTime;
    timer.repeat = false;
    timer.triggered.connect(cb);
    timer.start();
}

function take(taken, isPlayer) {
    if (lastPos > 0) {
        for (var i = lastPos - 1; i > lastPos - taken - 1; i--) {
            grid.children[i].visible = false
        }
        lastPos = lastPos - taken
    }

    if (isPlayer) {
        if (lastPos === 0) {
            btnRow.visible = false
            returnBtn.visible = true
            annLabel.text = "Вы играли"
            return
        }
        annLabel.text = "Ходит компьютер"
        var randomAmount = parseInt(Math.random() * 3)
        for (i = 0; randomAmount > lastPos; i++) randomAmount--
        if (randomAmount === 0) randomAmount++

        delay(1000, function() {
            take(randomAmount, false)
        })
    }

    if (!isPlayer) {
        if (lastPos === 0) {
            btnRow.visible = false
            returnBtn.visible = true
            annLabel.text = "Вы проиграли"
            return
        }
        annLabel.text = "Ходит игрок"
    }
}

Component.onCompleted: {
    lastPos = count
    console.log(lastPos)
}

```

```

for (var i = 0; i < count; i++) {
  const newObject = Qt.createObject('
    import QtQuick 2.0

    Rectangle {
      width: 50
      height: 50
      radius: 25
      y: 500
    }
    ',
                                                                    grid
                                                                    )
    newObject.color = "#" +
Math.floor(Math.random()*16777215).toString(16);
  }
}

```