|  |
| --- |
| https://lh6.googleusercontent.com/QcftzNtI05T0Y6fjdSh1Rr2rt8oqZ1IvnLvbn1jLJ7CCyteVir3k-xBLv4SL1wAgWJsRhmmJSR0UW-RP63_GQenE4vVWv05BRoZTsmIcBccVTnfxwmsnNMvjg599x9SqZd8E3dkd |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА** - **Российский технологический университет»**  **РТУ МИРЭА** |

Институт Информационных Технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения (ИППО)

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2** | |
| **по дисциплине** | |
| «Разработка программных приложений» | |
|  | |
| Выполнил студент группы ИКБО-02-18 | Юркевич Г.А. |
| Принял ассистент кафедры ИППО | Строганкова Н.В. |

|  |  |  |
| --- | --- | --- |
| Лабораторные работы выполнены | «\_\_»\_\_\_\_\_\_\_2020 г. |  |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2020 г. |  |

Москва 2020

**Содержание**

[1. Цель лабораторной работы 3](#_heading=h.26in1rg)

2. Общие требования [3](#_heading=h.2et92p0)

3. Задание [3](#_heading=h.3dy6vkm)

4. Поэтапное выполнение лабораторной работы [4](#_heading=h.4d34og8)

5. Результат работы [4](#_heading=h.lnxbz9)

6.Вывод [4](#_heading=h.17dp8vu)

[Список использованных источников 9](#_heading=h.35nkun2)

1. Цель лабораторной работы

Целью данной лабораторной работы является ознакомление со средой разработки программных приложений Android Studio, а также создание приложения из двух активити.

**2. Общие требования**

Среда разработки по умолчанию - Android Studio. Язык программирования - Java.

По своему желанию, можно использовать Kotlin, а также XCode/Swift при наличии соответствующего оборудования.

Решения должны быть размещены в виде проекта в git репозитории (github/bitbucket).

Ссылку на git-репозиторий и отчет по выполненной работе прислать на почту: strogankova-mirea@yandex.ru.

Минимальная версия API – 19/

**2. Задание**

Реализовать приложение, состоящее из двух активити:

Первое активити — сплеш скрин из первой лабораторной, только показывается не

больше чем время загрузки.

Второе активити представляет собой список и ViewPager сформированный из json-файла: https://raw.githubusercontent.com/wesleywerner/ancient-

tech/02decf875616dd9692b31658d92e64a20d99f816/src/data/techs.ruleset.json

Этот JSON файл — массив «технологий» в игре Цивилизация. В каждой технологии вас должны интересовать поля:

● graphic – картинка (относительноhttps://raw.githubusercontent.com/wesleywerner/ancient- tech/02decf875616dd9692b31658d92e64a20d99f816/src/images/tech/)

● name - название

● helptext - дополнительная информация (может не быть)

Второе активити должно начинаться с фрагмента со списком, в элементе списка

маленькая картинка 64dp и название.

Список кликабельный. При клике на технологии открывается ViewPager c фрагментом, в котором соответствующая картинка большого размера (не больше ширины экрана-20dp) и описанием технологии. При свайпе вправо описание должно меняться на описание предмета выше в списке, соответственно влево - на описание ниже в списке.

Опционально приложение может предоставлять Content Provider с данными

технологий.

Загрузка JSON должна происходить в момент показа сплеш скрина. Загрузка картинок, в процессе показа списка (естественно не в UI потоке), имена технологий не должны прыгать в процессе загрузки. Необходимо минимизировать использование трафика.

**4. Поэтапное выполнение лабораторной работы**

Создается сплэш скрин, которое переходит во второе активити. Реализован список с технологиями содержащий картинку, название и дополнительную информацию если таковая присутствует. Реализованна кликабельность списка. При нажатии на технологии открываеться ViewPager в котором реализован свайп в право и свайп в лево. Тестирование показало работоспособность приложения и выполнение поставленной задачи.

**5. Результат работы**

**SplashScreen.kt**

package com.example.laba2

import android.content.Intent

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

class SplashScreen : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

val intent = Intent (this, MainActivity::class.java)

startActivity(intent)

}

}

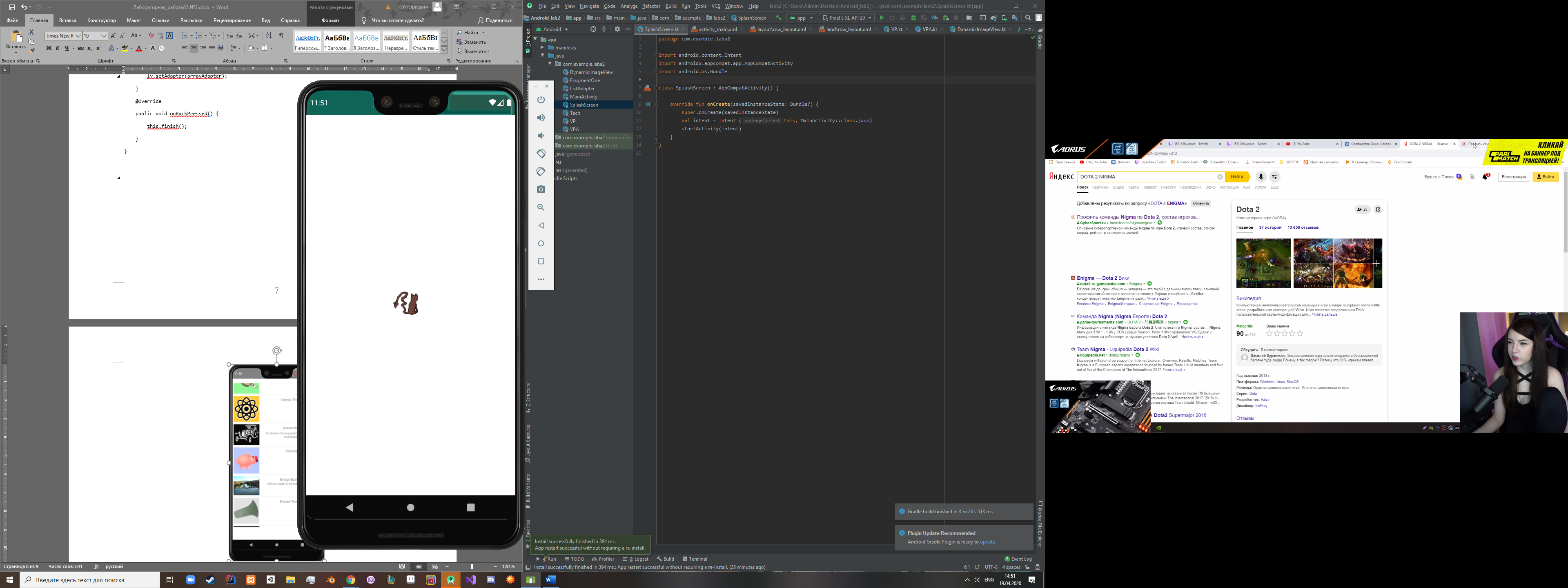


Рисунок 1 Splash screen

**MainActivity.kt**

package com.example.laba2

import android.os.AsyncTask

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

import android.widget.AdapterView

import kotlinx.android.synthetic.main.activity\_main.\*

import org.json.JSONArray

import java.net.URL

import android.content.Intent

import android.view.View

import android.widget.AdapterView.OnItemClickListener

import javax.net.ssl.HttpsURLConnection

class MainActivity : AppCompatActivity() {

var list = ArrayList<Tech>()

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

val url = "https://raw.githubusercontent.com/wesleywerner/ancient-tech/02decf875616dd9692b31658d92e64a20d99f816/src/data/techs.ruleset.json"

AsyncTaskHandleJson().execute(url)

technologies\_list.setOnItemClickListener(object : OnItemClickListener {

override fun onItemClick(parent: AdapterView<\*>, view: View, pos: Int, id: Long) {

val intent = Intent(view!!.context, VP::class.java)

var rPos = pos - 1

var lPos = pos + 1

if (pos == 0) rPos = list.lastIndex

if (pos == list.lastIndex) lPos = 0

intent.putExtra("1", list[rPos].graphic)

intent.putExtra("2", list[pos].graphic)

intent.putExtra("3", list[lPos].graphic)

intent.putExtra("4", list[rPos].name)

intent.putExtra("5", list[pos].name)

intent.putExtra("6", list[lPos].name)

intent.putExtra("7", list[rPos].helptext)

intent.putExtra("8", list[pos].helptext)

intent.putExtra("9", list[lPos].helptext)

startActivity(intent)

}

})

}

inner class AsyncTaskHandleJson : AsyncTask<String, String, String>() {

override fun doInBackground(vararg jsonUrl: String?): String {

val connection = URL(jsonUrl[0]).openConnection() as HttpsURLConnection

val jsonData: String

try{

connection.connect()

jsonData = connection.inputStream.use { it.reader().use{ reader -> reader.readText()} }

}finally {

connection.disconnect()

}

return jsonData

}

override fun onPostExecute(result: String?) {

super.onPostExecute(result)

HJson(result)

}

}

fun HJson(jsonString: String?) {

val jsonArray = JSONArray(jsonString)

var jsonLineIndex = 1

val adapter = ListAdapter(this, list)

while(jsonLineIndex < jsonArray.length())

{

val jsonObject = jsonArray.getJSONObject(jsonLineIndex)

list.add(

Tech(

jsonObject.getString("graphic"),

jsonObject.getString("name"),

if (jsonObject.has("helptext"))

jsonObject.getString("helptext")

else

""

)

)

jsonLineIndex++

}

technologies\_list.adapter = adapter

}

}

**VP.kt**

package com.example.laba2

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

import kotlinx.android.synthetic.main.activity\_view\_pager.\*

class VP : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_view\_pager)

val VPA = VPA(supportFragmentManager)

val RImage = intent.extras!!.getString("1")

val Image = intent.extras!!.getString("2")

val LImage = intent.extras!!.getString("3")

val RName = intent.extras!!.getString("4")

val Name = intent.extras!!.getString("5")

val LName = intent.extras!!.getString("6")

val RDescription = intent.extras!!.getString("7")

val Description = intent.extras!!.getString("8")

val LDescription = intent.extras!!.getString("9")

VPA.addFragment(FragmentOne(RImage!!, RName!!, RDescription!!), "R")

VPA.addFragment(FragmentOne(Image!!, Name!!, Description!!), "X")

VPA.addFragment(FragmentOne(LImage!!, LName!!, LDescription!!), "L")

viewPager.adapter = VPA

viewPager.currentItem = 1

}

}

**VPA.kt**

package com.example.laba2

import androidx.fragment.app.Fragment

import androidx.fragment.app.FragmentManager

import androidx.fragment.app.FragmentPagerAdapter

class VPA(manager: FragmentManager) : FragmentPagerAdapter(manager){

private val fragmentList : MutableList<Fragment> = ArrayList()

private val titleList : MutableList<String> = ArrayList()

override fun getItem(pos: Int): Fragment {

return fragmentList[pos]

}

override fun getCount(): Int {

return fragmentList.size

}

fun addFragment(fragment: Fragment, title: String){

fragmentList.add(fragment)

titleList.add(title)

}

override fun getPageTitle(pos: Int): CharSequence? {

return titleList[pos]

}

}

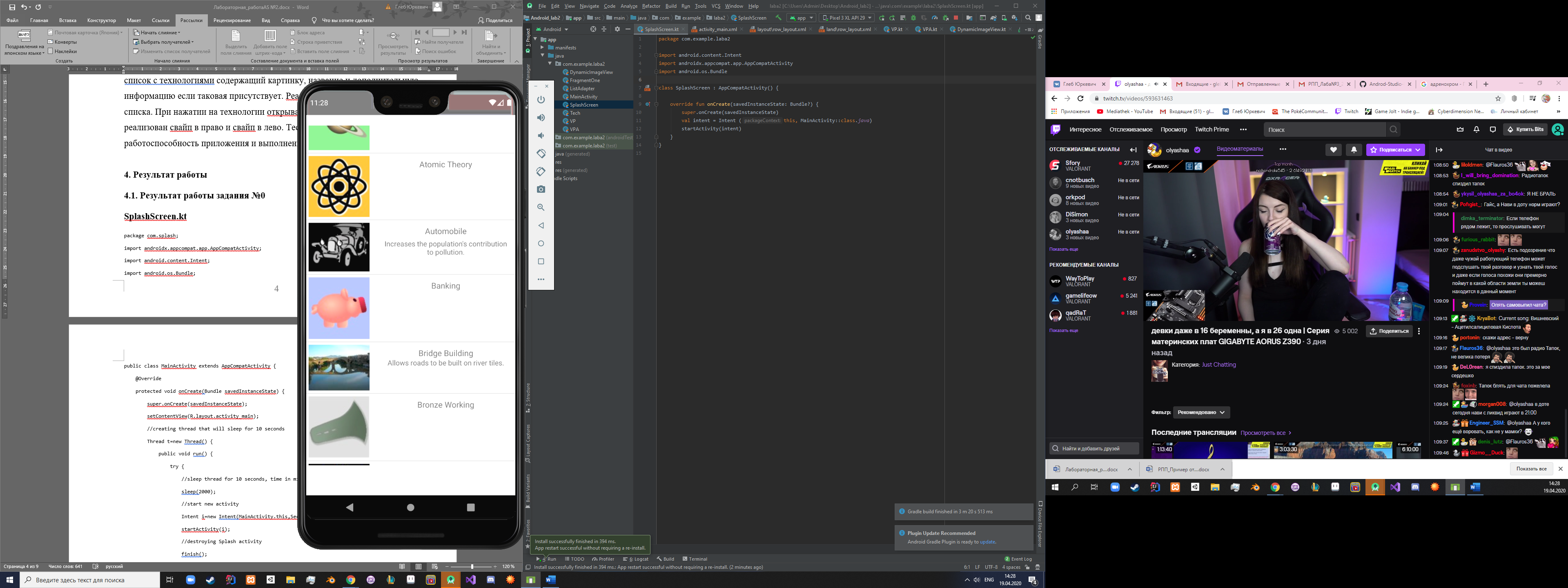


Рисунок 2 ListView

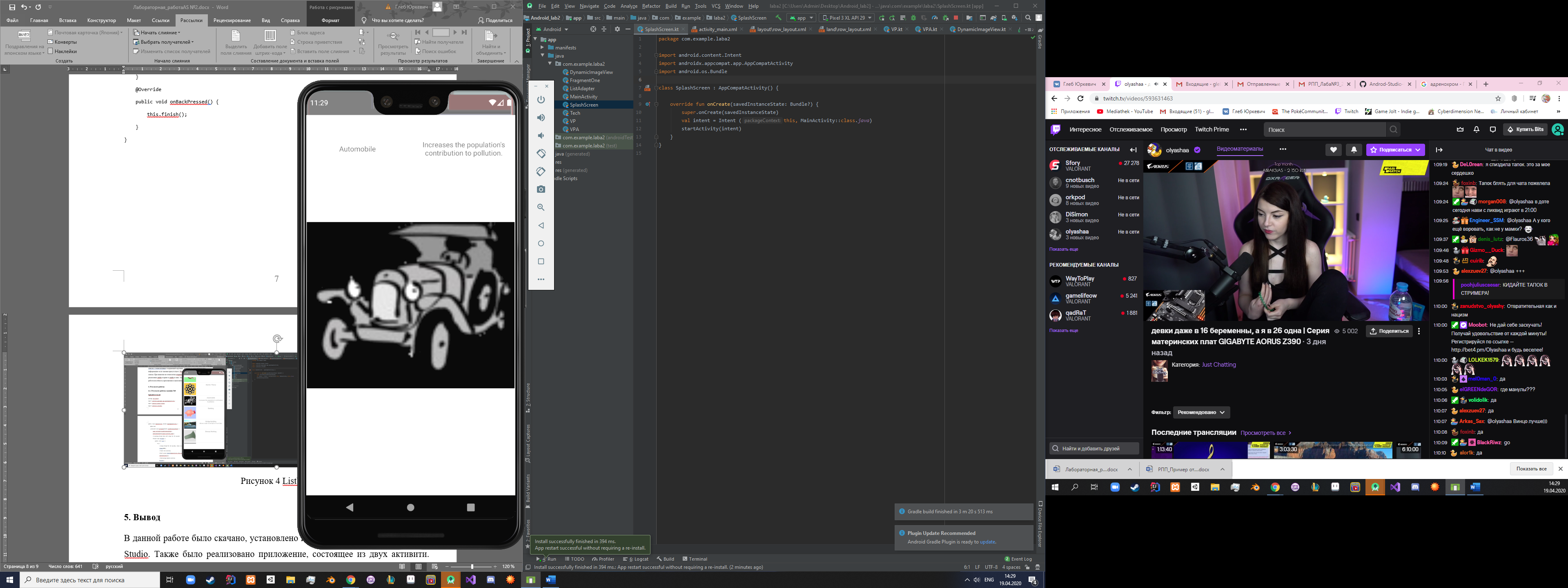


Рисунок 3 Fragment

**5. Вывод**

В данной работе было скачано, установлено и запущено приложение Android Studio. Также было реализовано приложение, состоящее из двух активити. Приложение было протестировано.

**Список использованных источников**

1. Главный сайт по Android: <https://developer.android.com/>
2. Канал Google на YouTube:

<http://www.youtube.com/channel/UC_x5XG1OV2P6uZZ5FSM9Ttw>

1. Онлайн-курсы по Android:

– Udacity:

<https://eu.udacity.com/course/new-android-fundamentals--ud851>

1. <https://stackoverflow.com/>
2. <https://github.com>
3. <https://google.com/>