

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

Мобильное приложение для расчёта материалов чистовой отделки жилых помещений

Автор: студент группы КЭ-405

Тодуа Левани Романович

Научный руководитель: к.п.н., доцент каф. ЭВМ

Алтухова Мария Александровна

Актуальность темы

2

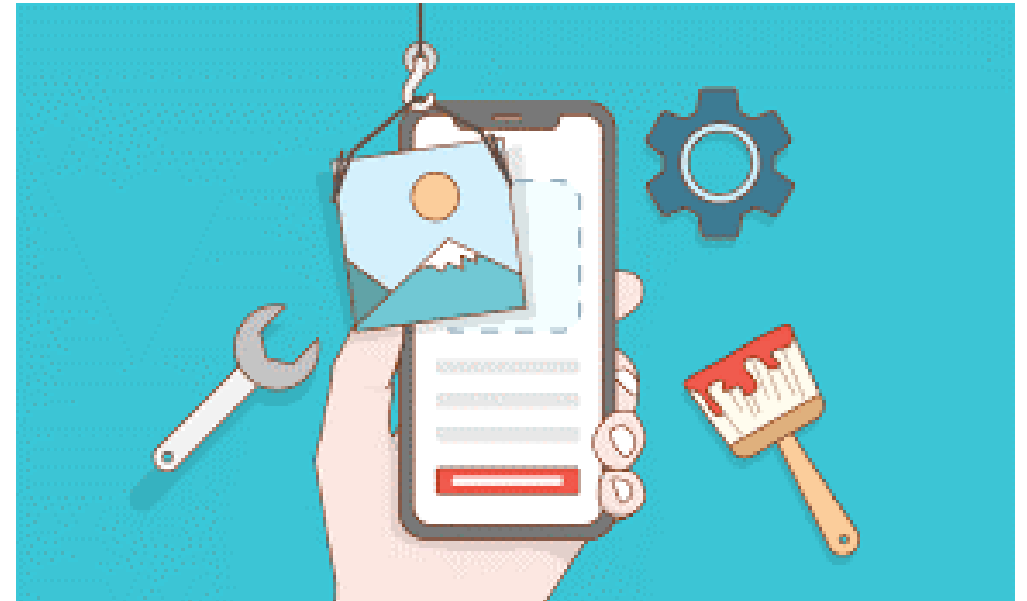
Актуальность заключается в том, чтобы упростить этапы чистовой отделки помещения для обычного человека, предоставляя ему возможность наглядно видеть материал наложенный на модель его помещения и рассчитывать необходимое количество материала.



Цель

3

Разработка мобильного приложения для широкого круга пользователей, которое будет предназначено для облегчения задачи выбора и расчета количества материалов для внутренней отделки помещения с возможностью визуализации выбранного варианта на трёхмерной модели помещения.



Задачи

- 1) провести аналитический обзор аналогов по тематике работы;
- 2) разработать алгоритмы обработки изображения обоев и построения трёхмерных визуализаций жилых помещений с наложением текстур;
- 3) спроектировать интерфейс приложения;
- 4) разработать приложение и произвести его отладку;
- 5) протестировать работу приложения.

Обзор аналогов

5

Таблица 1 - Сравнительный анализ аналогов

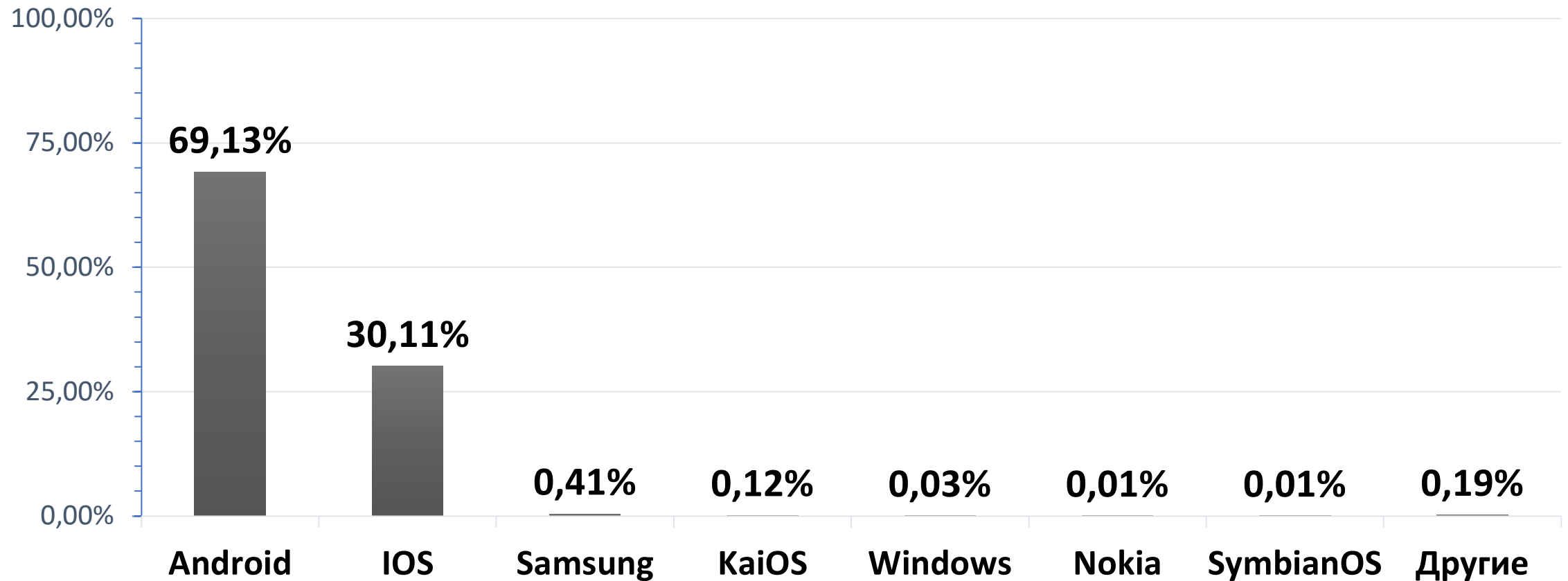
	Название приложения	Построение трёхмерной модели	Получение текстуры обоев	Заполнение модели текстурами	Возможность сравнения вариантов
	Planner5D	Да	Нет	Да	Да
	Homestyler	Да	Нет	Да	Да
	Строительный калькулятор (разработчик: Will Maze)	Нет	Нет	Нет	Нет
	Строительный калькулятор (разработчик: CodART.PRO)	Нет	Нет	Нет	Да

Продолжение таблицы 1



Название приложения	Расчёт количества материалов	Доступность широкому кругу пользователей	Независимость приложения от поставщиков
 Planner5D	Нет	Да	Нет
 Homestyler	Нет	Да	Нет
 Строительный калькулятор (разработчик: Will Maze)	Да	Да	Нет
 Строительный калькулятор (разработчик: CodART.PRO)	Да	Да	Нет

Анализ основных технологических решений 7

Операционная система: Android (69,13%)



Язык программирования: Kotlin

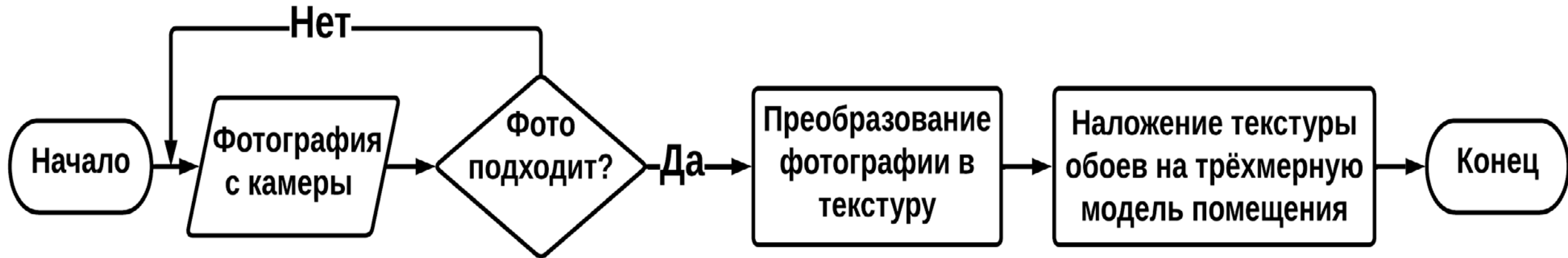
Языки программирования	Java 	Kotlin 
Скорость компиляции	Быстрее Kotlin на 12-15%	Медленнее Java на 12-15%
Компактность	Объёмнее Kotlin на 30-40%	Компактнее Java на 30-40%
Безопасная работа с обнуляемыми переменными	Нет	Есть

Интегрированная среда разработки (IDE): Android Studio

IDE	Android Studio	Eclipse	IntelliJ IDEA
Языки программирования	Java, C, C++, Kotlin	Java, C, C++, C#, JavaScript, Python	Java, Scala, Groovy, Kotlin, JavaScript, TypeScript, SQL
Операционная система	Windows, MacOS, Linux	Любая ОС, которая поддерживает Java	Windows, MacOS, Linux
Цена	Бесплатно	Бесплатно	От Бесплатно До 499\$/год

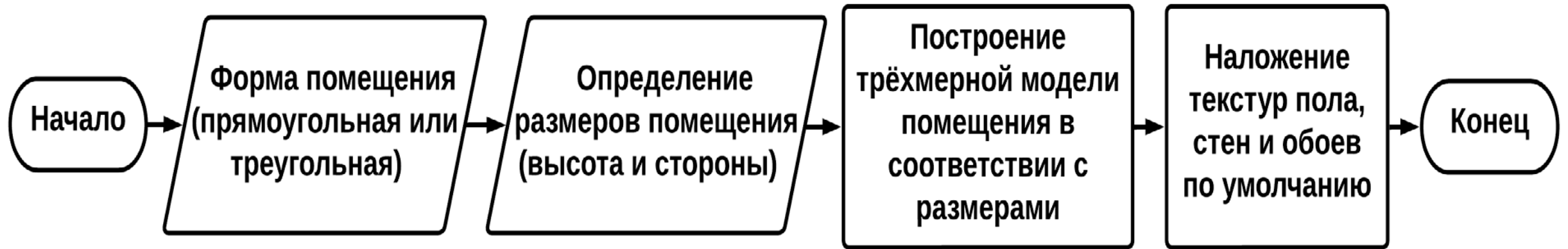
Разработка алгоритма обработки изображения обоев

10

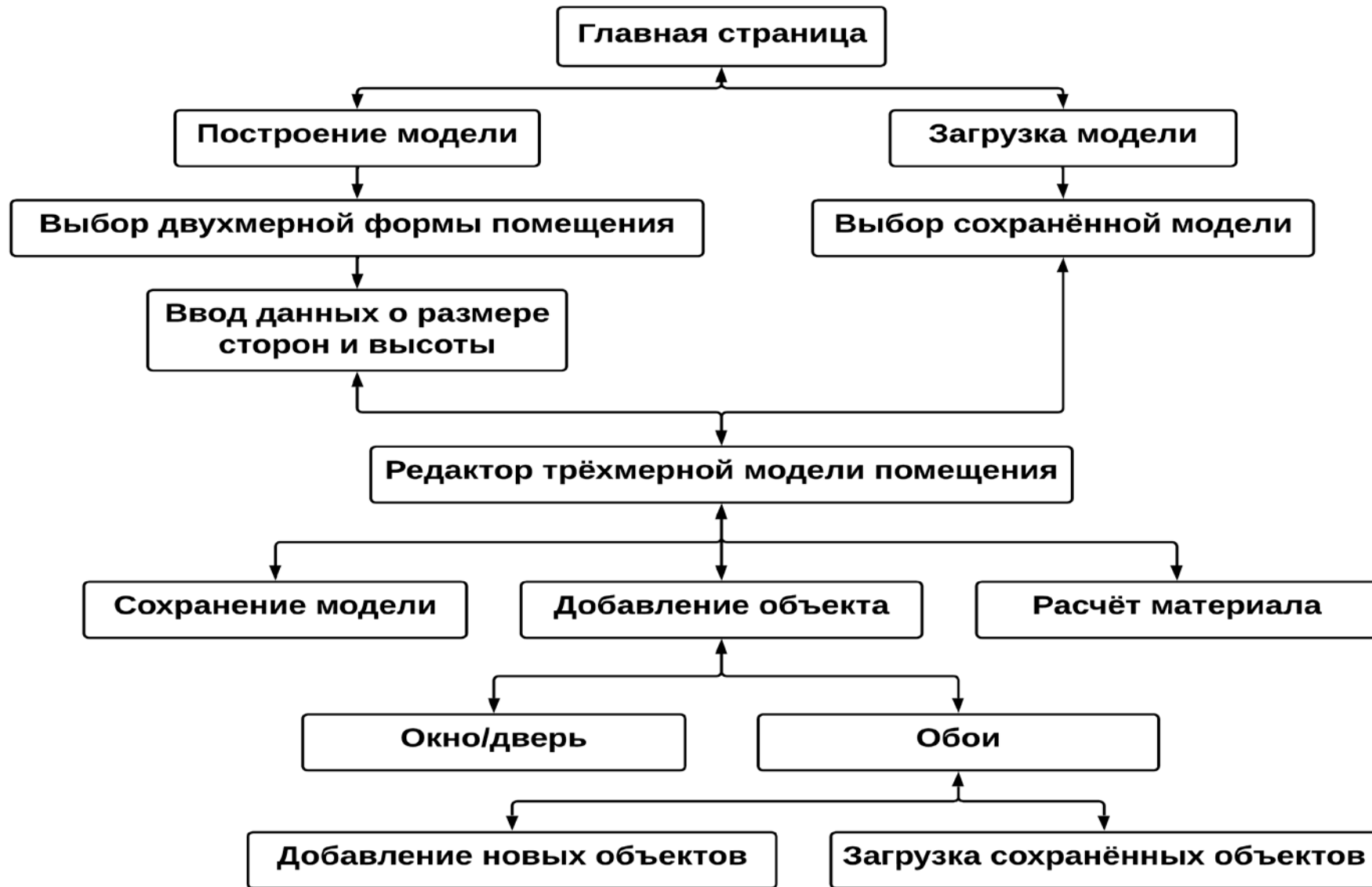


Разработка алгоритма построения трёхмерных визуализаций жилых помещений с наложением текстур

11



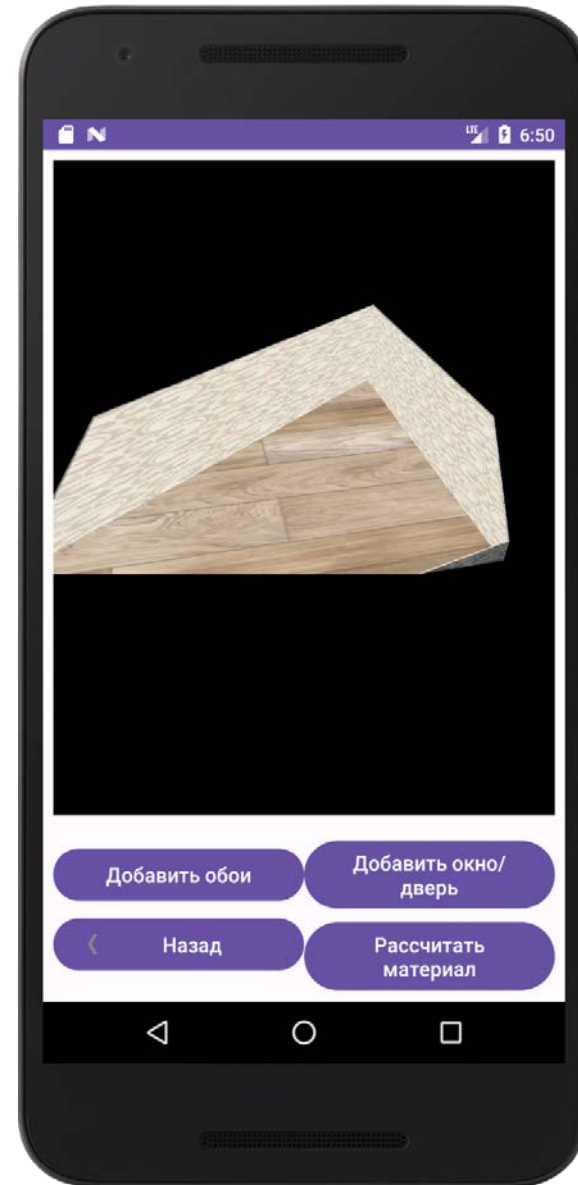
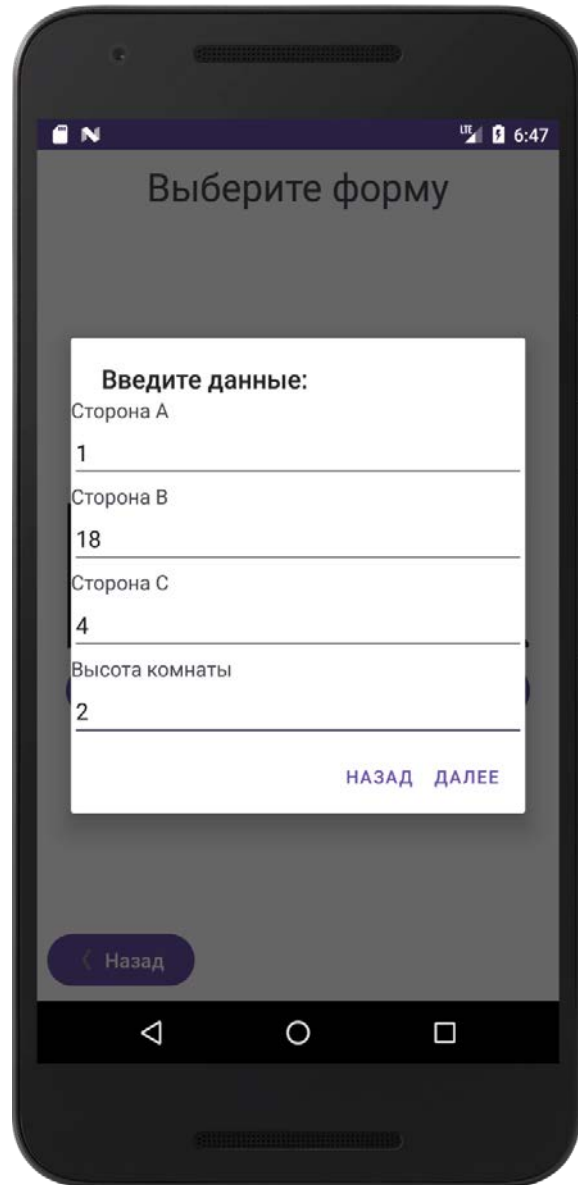
Проектирование интерфейса приложения 12

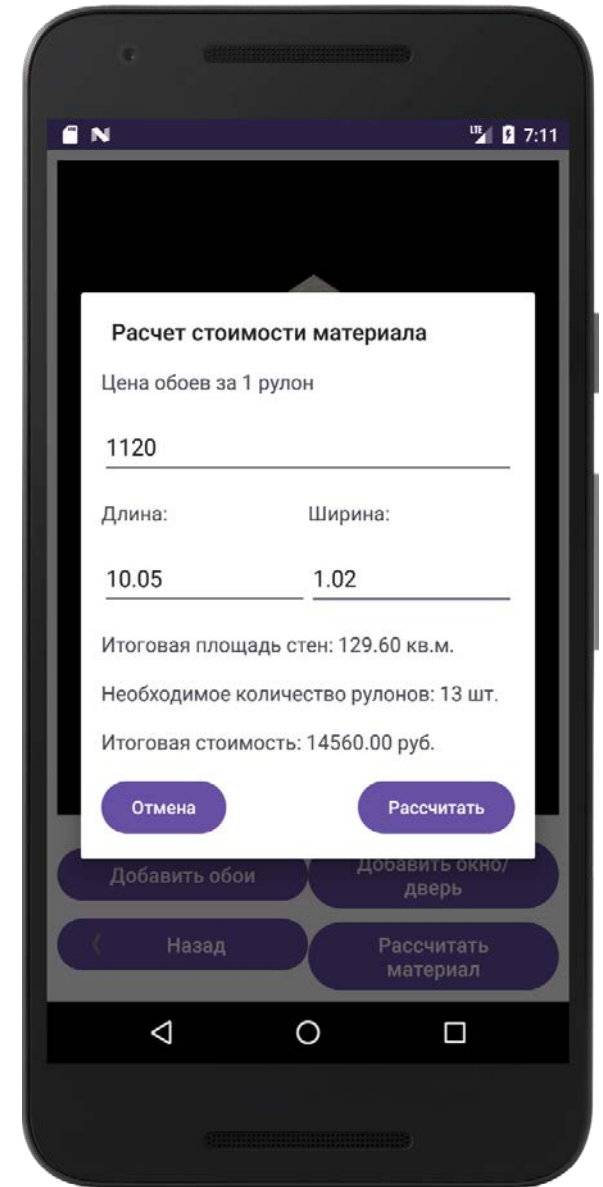
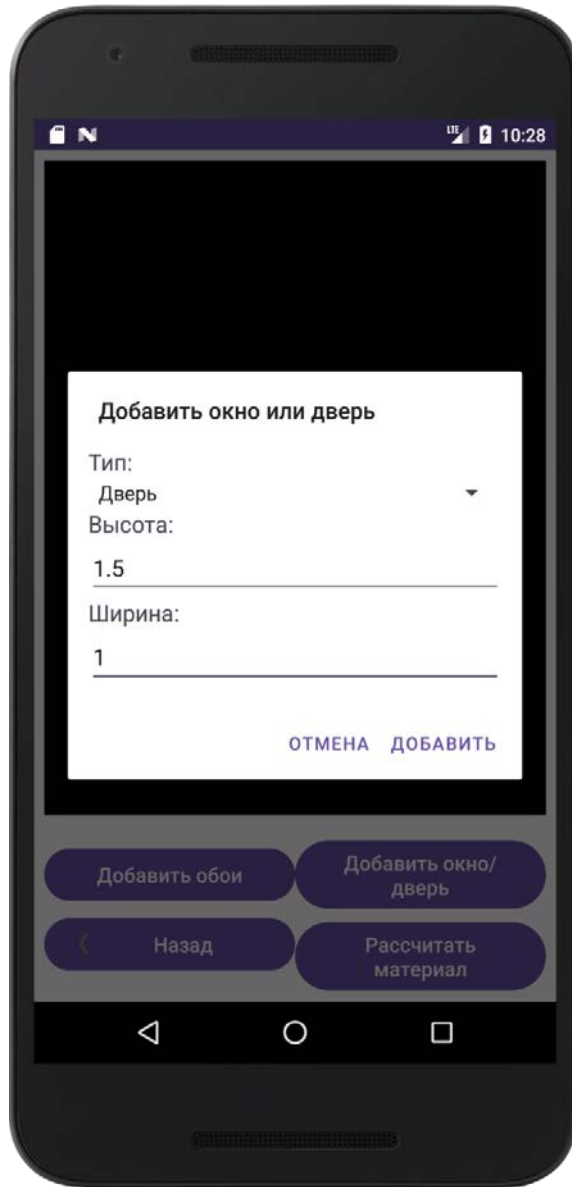
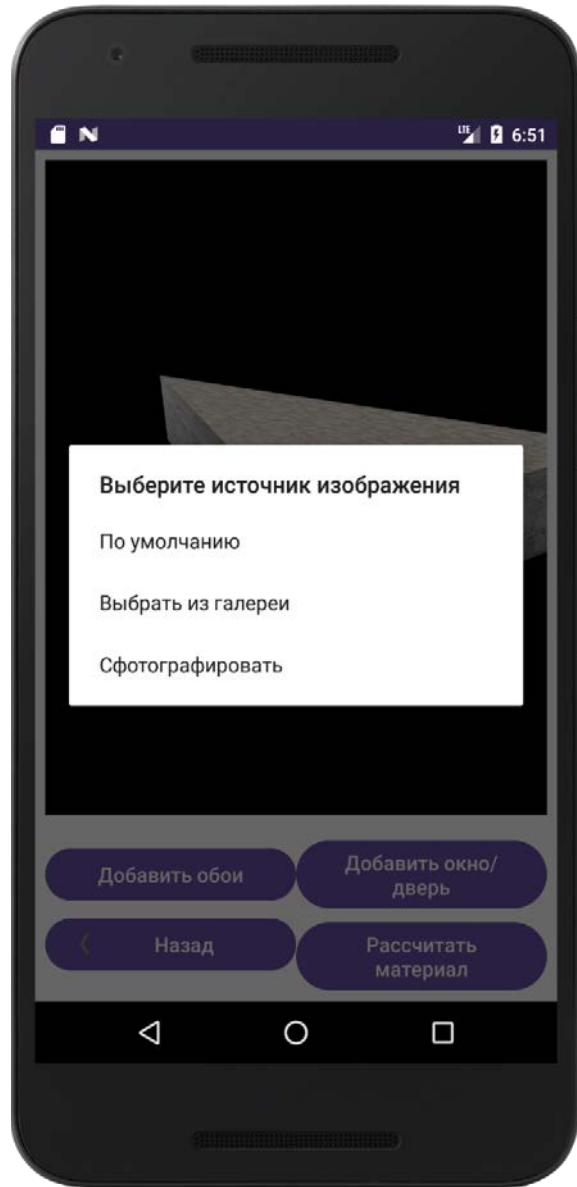


Разработка приложения

13



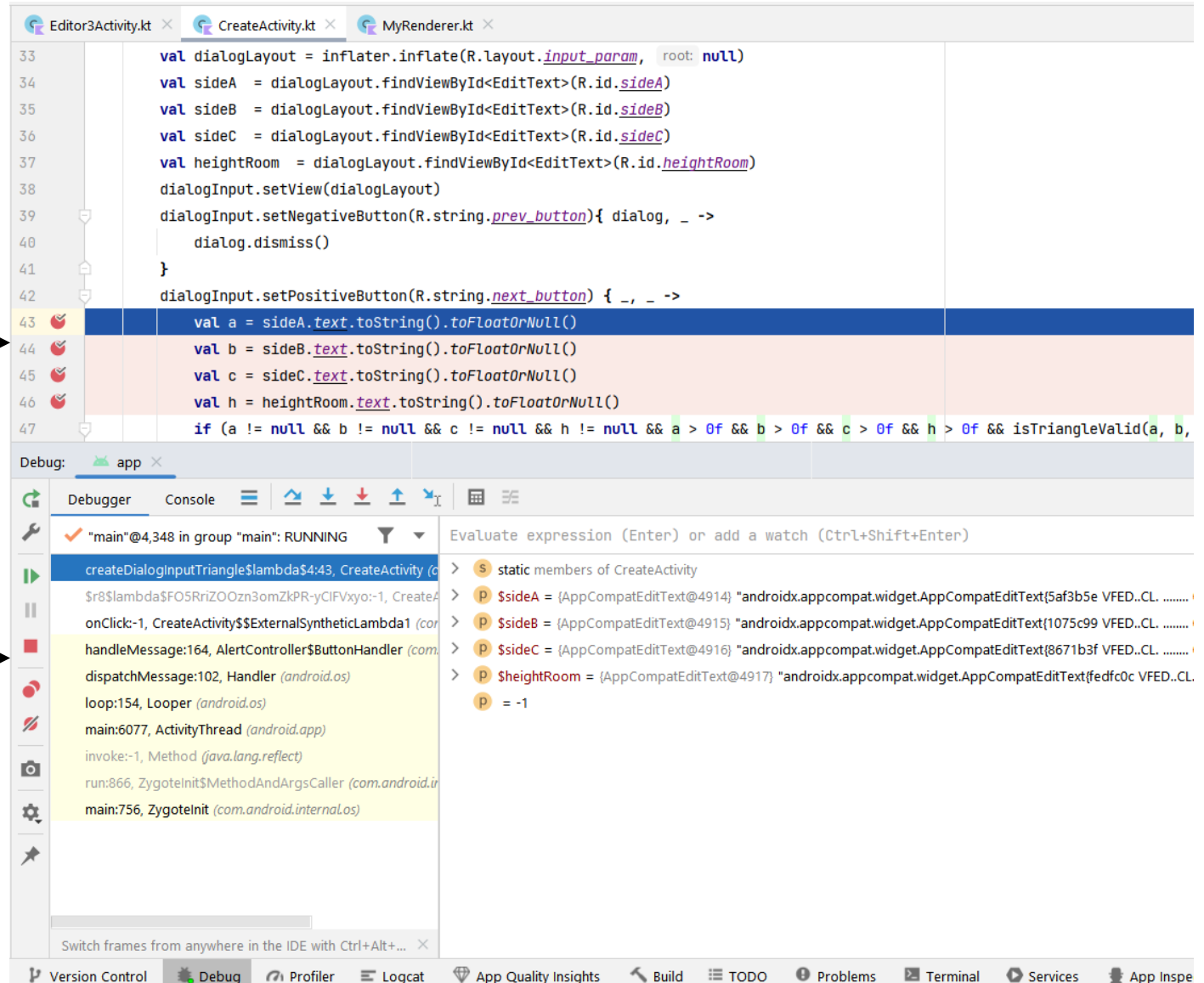




Отладка приложения

16

Точки останова



```
Editor3Activity.kt | CreateActivity.kt | MyRenderer.kt
33 |         val dialogLayout = inflater.inflate(R.layout.input_param, root: null)
34 |         val sideA = dialogLayout.findViewById<EditText>(R.id.sideA)
35 |         val sideB = dialogLayout.findViewById<EditText>(R.id.sideB)
36 |         val sideC = dialogLayout.findViewById<EditText>(R.id.sideC)
37 |         val heightRoom = dialogLayout.findViewById<EditText>(R.id.heightRoom)
38 |         dialogInput.setView(dialogLayout)
39 |         dialogInput.setNegativeButton(R.string.prev_button){ dialog, _ ->
40 |             dialog.dismiss()
41 |         }
42 |         dialogInput.setPositiveButton(R.string.next_button) { _, _ ->
43 |             val a = sideA.text.toString().toFloatOrNull()
44 |             val b = sideB.text.toString().toFloatOrNull()
45 |             val c = sideC.text.toString().toFloatOrNull()
46 |             val h = heightRoom.text.toString().toFloatOrNull()
47 |             if (a != null && b != null && c != null && h != null && a > 0f && b > 0f && c > 0f && h > 0f && isTriangleValid(a, b,
```

Debugger: app

Debugger Console

main:756, ZygoteInit (com.android.internal.os)

main:6077, ActivityThread (android.app)

loop:154, Looper (android.os)

dispatchMessage:102, Handler (android.os)

handleMessage:164, AlertController\$ButtonHandler (com.android.internal.os)

onClick:-1, CreateActivity\$\$ExternalSyntheticLambda1 (com.android.internal.os)

onClickListener:1, CreateActivity (com.android.internal.os)

createDialogInputTriangle\$lambda\$4:43, CreateActivity (com.android.internal.os)

main:4348 in group "main": RUNNING

static members of CreateActivity

sideA = {AppCompatActivity@4914} "androidx.appcompat.widget.AppCompatActivity{5af3b5e VFED..CL

sideB = {AppCompatActivity@4915} "androidx.appcompat.widget.AppCompatActivity{1075c99 VFED..CL

sideC = {AppCompatActivity@4916} "androidx.appcompat.widget.AppCompatActivity{8671b3f VFED..CL

heightRoom = {AppCompatActivity@4917} "androidx.appcompat.widget.AppCompatActivity{fedfc0c VFED..CL

= -1

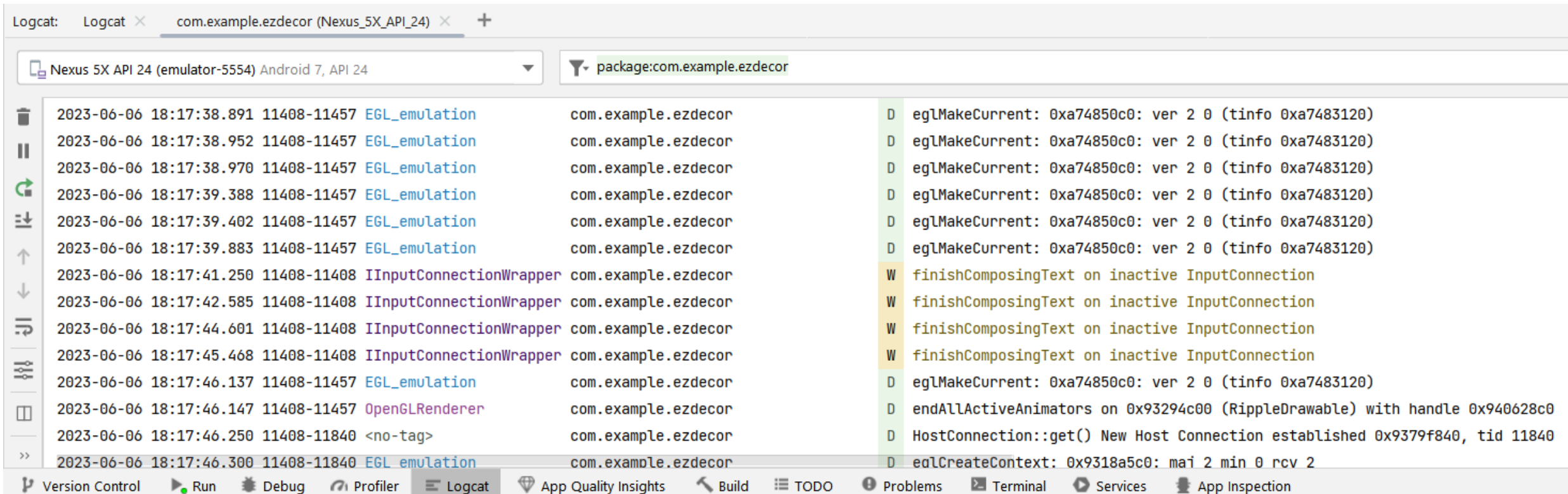
Окно режима Debug



Отладка приложения

17

Окно инструмента Logcat:



The screenshot shows the Logcat window in an IDE. The window title is "Logcat: Logcat x com.example.ezdecor (Nexus_5X_API_24) x +". The device selected is "Nexus 5X API 24 (emulator-5554) Android 7, API 24". The filter is set to "package:com.example.ezdecor". The log messages are as follows:

```
2023-06-06 18:17:38.891 11408-11457 EGL_emulation com.example.ezdecor D eglMakeCurrent: 0xa74850c0: ver 2 0 (tinfo 0xa7483120)
2023-06-06 18:17:38.952 11408-11457 EGL_emulation com.example.ezdecor D eglMakeCurrent: 0xa74850c0: ver 2 0 (tinfo 0xa7483120)
2023-06-06 18:17:38.970 11408-11457 EGL_emulation com.example.ezdecor D eglMakeCurrent: 0xa74850c0: ver 2 0 (tinfo 0xa7483120)
2023-06-06 18:17:39.388 11408-11457 EGL_emulation com.example.ezdecor D eglMakeCurrent: 0xa74850c0: ver 2 0 (tinfo 0xa7483120)
2023-06-06 18:17:39.402 11408-11457 EGL_emulation com.example.ezdecor D eglMakeCurrent: 0xa74850c0: ver 2 0 (tinfo 0xa7483120)
2023-06-06 18:17:39.883 11408-11457 EGL_emulation com.example.ezdecor D eglMakeCurrent: 0xa74850c0: ver 2 0 (tinfo 0xa7483120)
2023-06-06 18:17:41.250 11408-11408 IInputConnectionWrapper com.example.ezdecor W finishComposingText on inactive InputConnection
2023-06-06 18:17:42.585 11408-11408 IInputConnectionWrapper com.example.ezdecor W finishComposingText on inactive InputConnection
2023-06-06 18:17:44.601 11408-11408 IInputConnectionWrapper com.example.ezdecor W finishComposingText on inactive InputConnection
2023-06-06 18:17:45.468 11408-11408 IInputConnectionWrapper com.example.ezdecor W finishComposingText on inactive InputConnection
2023-06-06 18:17:46.137 11408-11457 EGL_emulation com.example.ezdecor D eglMakeCurrent: 0xa74850c0: ver 2 0 (tinfo 0xa7483120)
2023-06-06 18:17:46.147 11408-11457 OpenGLRenderer com.example.ezdecor D endAllActiveAnimators on 0x93294c00 (RippleDrawable) with handle 0x940628c0
2023-06-06 18:17:46.250 11408-11840 <no-tag> com.example.ezdecor D HostConnection::get() New Host Connection established 0x9379f840, tid 11840
2023-06-06 18:17:46.300 11408-11840 EGL_emulation com.example.ezdecor D eglCreateContext: 0x9318a5c0: mai 2 min 0 rcv 2
```

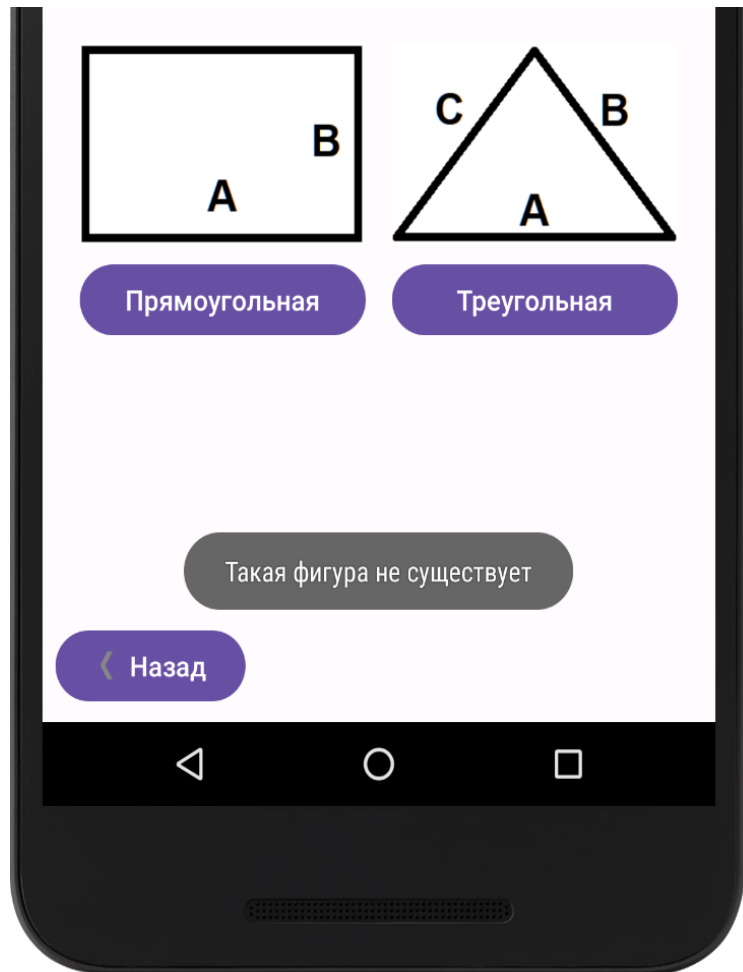
The IDE interface at the bottom shows tabs for Version Control, Run, Debug, Profiler, Logcat (selected), App Quality Insights, Build, TODO, Problems, Terminal, Services, and App Inspection.

Тестирование работы приложения

18

Модульное тестирование:

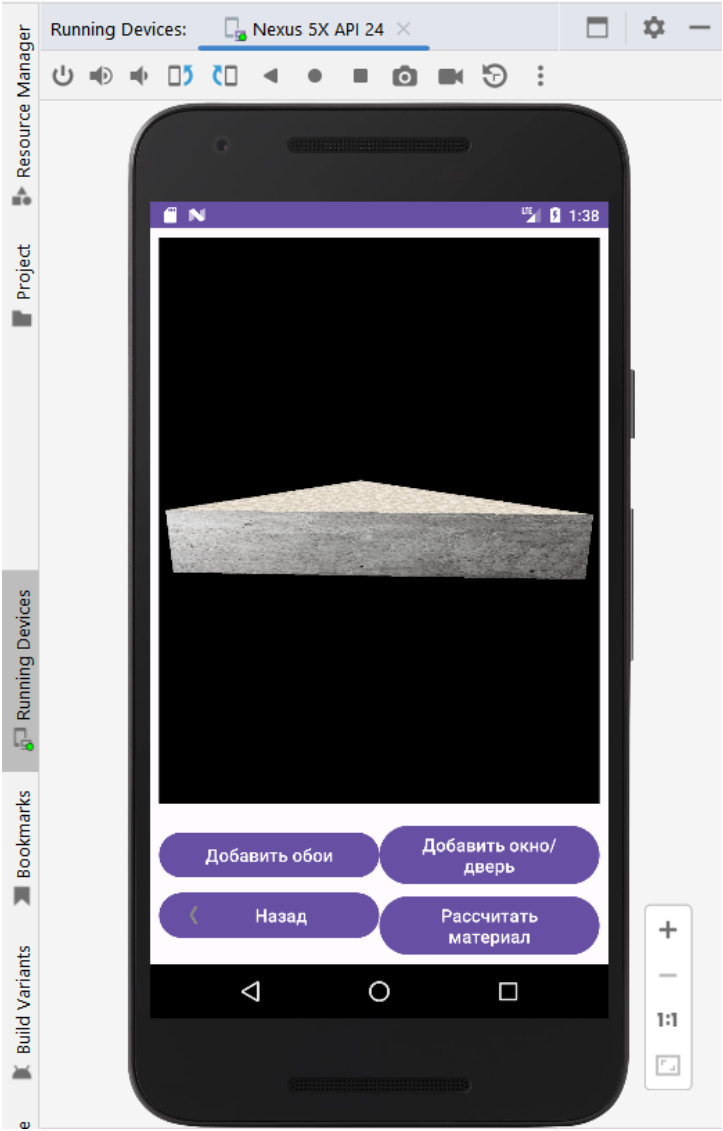
Интеграционное тестирование:



```
33 ● val sideA = this.intent.getFloatExtra( name: "SideA", defaultValue: 0f)
34 ● val sideB = this.intent.getFloatExtra( name: "SideB", defaultValue: 0f)
35 ● val heightRoom = this.intent.getFloatExtra( name: "HeightR", defaultValue: 0f)
36 triangleForm = this.intent.getBooleanExtra( name: "triangleForm", defaultValue: false)
37 if(triangleForm) {
38 ● sideC = this.intent.getFloatExtra( name: "SideC", defaultValue: 0f)
```

```
48 ● val intent = Intent( packageContext: this, Editor3Activity::class.java)
49 ● intent.putExtra( name: "SideA", a)
50 ● intent.putExtra( name: "SideB", b)
51 ● intent.putExtra( name: "SideC", c)
52 ● intent.putExtra( name: "HeightR", h)
53 ● intent.putExtra( name: "triangleForm", value: true)
54 ● startActivity(intent)
```

Системное тестирование:



Device Manager

Virtual Physical

Create device

Device	API	Size on Disk	Actions
Nexus 5X API 24 Android 7.0 Google APIs x86	24	9.1 GB	[Stop] [Folder] [Edit] [More]

Nexus 5X API 24

Summary

API level	24
Resolution	1080 × 1920
dp	412 × 732
ABI list	x86
Available storage	5,761 MB

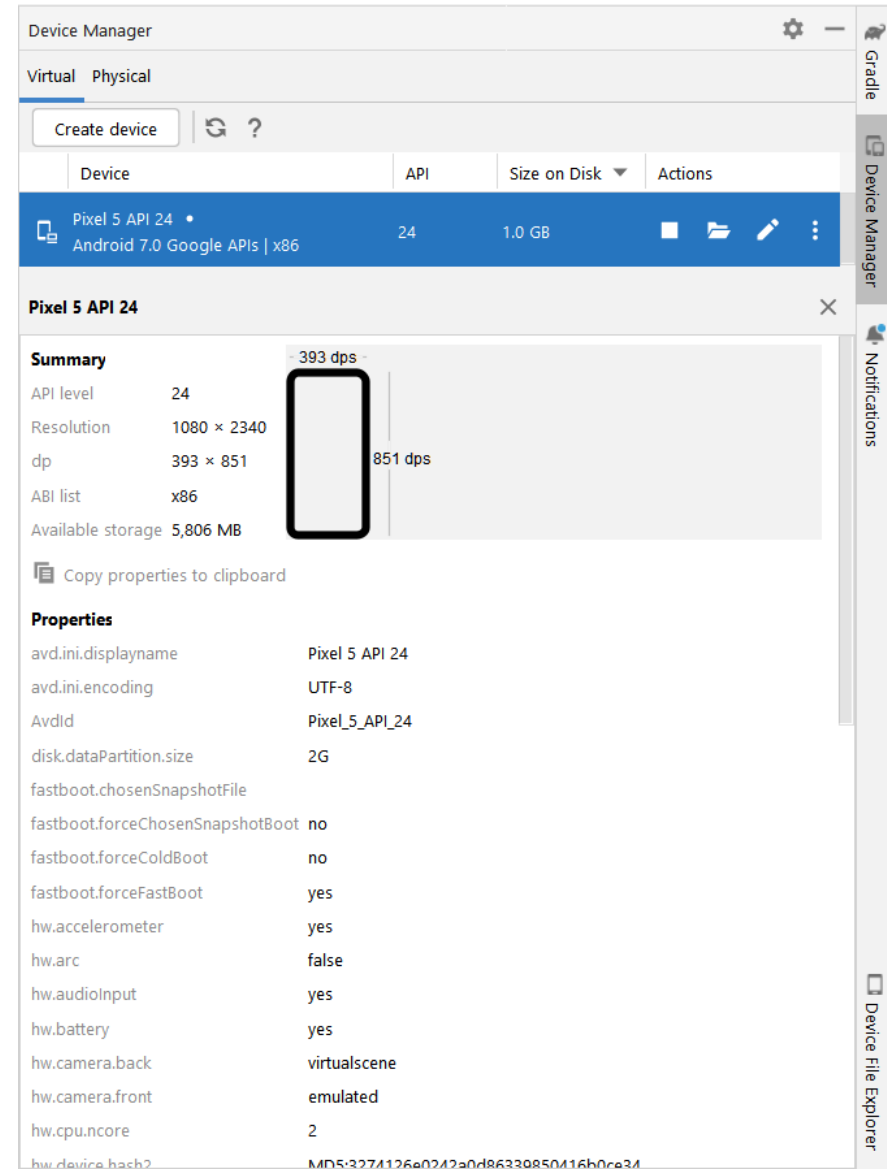
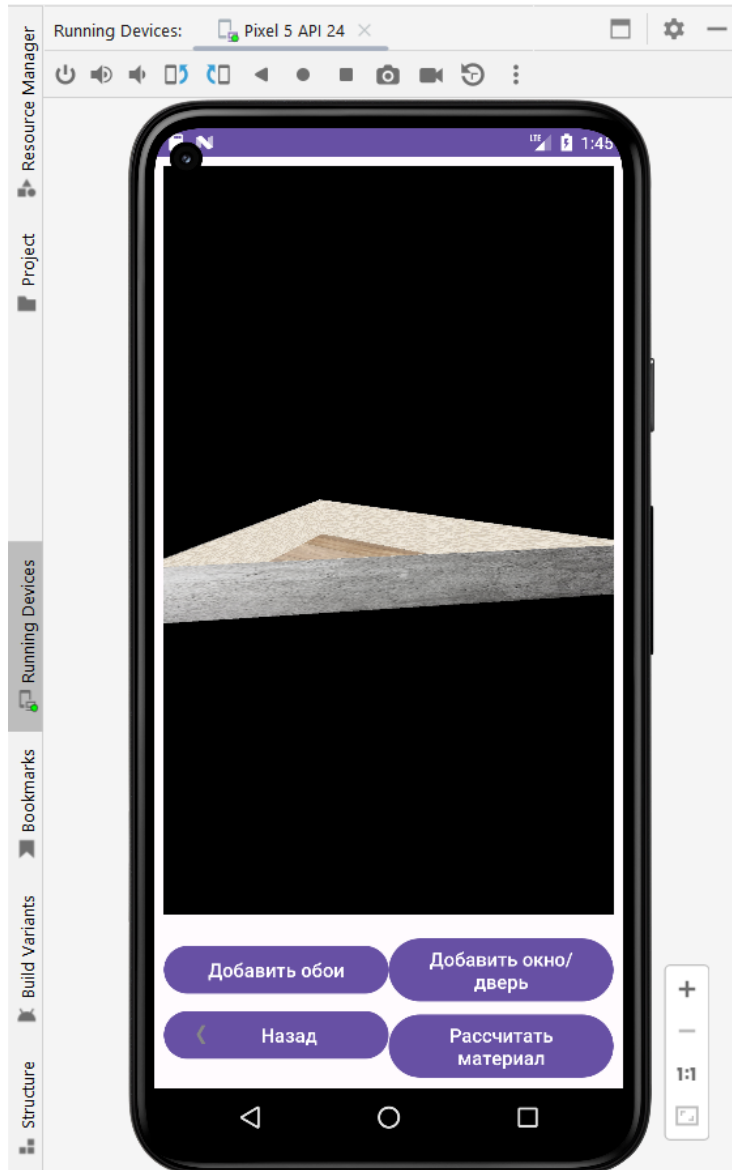
412 dps

732 dps

Copy properties to clipboard

Properties

avd.ini.displayname	Nexus 5X API 24
avd.ini.encoding	UTF-8
AvdId	Nexus_5X_API_24
disk.dataPartition.size	6442450944
fastboot.chosenSnapshotFile	
fastboot.forceChosenSnapshotBoot	no
fastboot.forceColdBoot	no
fastboot.forceFastBoot	yes
hw.accelerometer	yes
hw.arc	false



- проведён аналитический обзор аналогов по тематике работы;
- разработаны алгоритмы обработки изображения обоев и построения трёхмерных визуализаций жилых помещений с наложением текстур;
- спроектирован интерфейс приложения;
- разработано приложение (частично некорректно) и произведена его отладка;
- протестирована работа приложения.

Спасибо за внимание!