

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**КУРСОВАЯ РАБОТА**  
**по дисциплине «Разработка приложений для мобильных платформ»**  
**Тема: Soundboard**

Студент гр. 1303	_____	Самохин К.А.
Студент гр. 1303	_____	Иевлев Е.А.
Студент гр. 1303	_____	Чернуха В.В.
Преподаватель	_____	Заславский М.М.

Санкт-Петербург  
2025

## ЗАДАНИЕ

Студент Самохин К.А.

Студент Иевлев Е.А.

Студент Чернуха В.В.

Группа 1303

Тема: Soundboard

Исходные данные:

Необходимо реализовать простое Android–приложение, реализующее функции soundboard: запись пользовательских семплов, конфигурация клавиш на экране, запись треков и возможность поделиться, макросы.

Содержание пояснительной записки:

«Содержание»

«Введение»

«Сценарии использования»

«Пользовательский интерфейс»

«Разработанное приложение»

«Выводы»

«Приложения»

«Список использованных источников»

Предполагаемый объем пояснительной записки:

Не менее 15 страниц.

Дата выдачи задания: 05.02.2025

Дата сдачи реферата: 25.03.2025

Дата защиты реферата: 25.03.2025

Студент	_____	Самохин К.А.
Студент	_____	Иевлев Е.А.
Студент	_____	Чернуха В.В.
Преподаватель	_____	Заславский М.М.

## АННОТАЦИЯ

В рамках курсовой работы разработано Android-приложение, представляющее собой простой Soundboard, в котором можно создавать свои собственные треки при помощи записи и делиться ими через встроенный плеер, конфигурировать клавиши и макросы на экране.

Для разработки использованы технологии Kotlin, Android SDK, Gson, Oboe.

Найти исходники можно по ссылке: <https://github.com/moevm/adfmp1h25-sound>

## SUMMARY

As part of the coursework, an Android application was developed, which is a simple Soundboard, in which you can create your own tracks using recording and share them through the built-in player, configure keys and macros on the screen. Kotlin, Android SDK, Gson, Oboe technologies were used for development.

The source code can be found at the link: <https://github.com/moevm/adfmp1h25-sound>

## СОДЕРЖАНИЕ

1.	Введение	7
1.1.	Актуальность проблемы	7
1.2.	Постановка задачи	7
1.3.	Предлагаемое решение	7
1.4.	Почему решение необходимо реализовать как мобильное приложение	8
2.	Сценарии использования	9
2.1.	Сценарий «Игра на клавиатуре»	9
2.2.	Сценарий «Запись макросов»	9
2.3.	Сценарий «Игра под "минус"»	10
2.4.	Сценарий «Запись»	10
2.5.	Сценарий «Удаление макроса»	10
2.6.	Сценарий «Просмотр информации о разработчиках»	11
2.7.	Сценарий «Просмотр списка записанных треков»	11
2.8.	Сценарий «Поиск записанного трека»	11
2.9.	Сценарий «Просмотр руководства по использованию»	11
2.10.	Сценарий «Прослушивание записанного трека»	12
2.11.	Сценарий «Удалить записанный трек»	12
2.12.	Сценарий «Поделиться записанным треком»	12
2.13.	Сценарий «Переименовать записанный трек»	13
2.14.	Сценарий «Прикрепить семпл к клавише»	13
2.15.	Сценарий «Удалить семпл с клавиши»	14
3.	Пользовательский интерфейс	15
3.1.	Макет интерфейса с графом переходов	15
3.2.	Целевые устройства, обоснование требований и максимально подробные характеристики	15
4.	Разработанное приложение	18
4.1.	Краткое описание	18

4.2	Схема архитектуры	18
4.2.1	Используемые технологии	18
4.2.2	Архитектура приложения	18
4.3	Использованные библиотеки	19
5.	Выводы	21
5.1	Достигнутые результаты	21
5.2	Недостатки и пути для улучшения	21
5.3	Будущее развитие решения	21
6.	Литература	22
7.	Приложения	23
7.1	Инструкция для пользователя	23
7.2	Снимки экрана приложения	24

# **1. ВВЕДЕНИЕ**

## **1.1. Актуальность проблемы**

Актуальность приложения Soundboard для Android заключается в предоставлении возможности простого, понятного инструмента создания треков для пользователей. Soundboard-приложение предоставляют пользователям возможность создавать и делиться семплами и музыкой, что способствует развитию креативности и развлечению. Это может быть интересно как для обычных пользователей, так и для создателей контента.

## **1.2. Постановка задачи**

Задача проекта заключается в разработке Android-приложения, которое позволяет пользователям создавать треки, играя на клавишах приложения. Приложение должно предоставлять следующие возможности:

- Создание собственных треков: пользователь может записать трек, играя на клавишах приложения.
- Конфигурация клавиш: возможность настроить свои собственные пресеты.
- Настройка макросов: возможность записать семпл на макрос и воспроизводить его.
- Встроенный плеер: возможность прослушать записанные треки и поделиться ими.

Кроме того, приложение должно быть интуитивно понятным и минималистичным, чтобы пользователи могли легко освоить его функционал без обучения и быстро выполнять необходимые операции.

## **1.3. Предлагаемое решение**

Разработать Android-приложение, которое позволит пользователям создавать и хранить музыкальные треки, отправлять их через социальные сети. Приложение будет поддерживать музыкальные файлы типа .wav, обеспечивая

при этом интуитивно понятный интерфейс. Для реализации функционала будет использована современная высокопроизводительная звуковая библиотека `oboe`.

#### **1.4. Почему решение необходимо реализовать как мобильное приложение**

- Мобильные устройства всегда находятся под рукой, что позволяет пользователю создавать треки в любое время и из любого места.
- Мобильные платформы предоставляют простые инструменты для интеграции с различными социальными сетями и мессенджерами, что упрощает отправку созданных треков.
- Приложение может работать локально без необходимости постоянного доступа к интернету, что обеспечивает его функциональность в любых условиях.
- Мобильные устройства имеют встроенный микрофон, который может пригодиться для записи семплов или треков.



## **2. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ**

### **2.1. Сценарий «Игра на клавиатуре»**

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Пользователь открывает приложение.
2. Пользователь переходит к экрану «soundboard».
3. Пользователь использует пресет по умолчанию или выбирает другой на верхней панели
4. Пользователь играет по клавишам

### **2.2. Сценарий «Запись макросов»**

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Аналогично пунктам сценария «Игра на клавиатуре»
2. Пользователь выбирает пустой (серый) макрос на панели справа и однократно нажимает на него - макрос переходит в состояние записи (красный)
3. Пользователь играет по клавишам музыкальный рисунок (семпл), после чего повторным нажатием на макрос завершает запись (макрос становится зелёным)

### **2.3. Сценарий «Игра под "минус"»**

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Аналогично пунктам сценария «Игра на клавиатуре»
2. Пользователь выбирает один или несколько непустых (зелёных) макросов с записанным на каждый семплом на панели справа и

однократно нажимает на выбранный(-ые) - макрос(-ы) переходит(-ят) в состояние готовности циклического проигрывания (жёлтый(-ые))

3. Пользователь нажимает кнопку "Play/Pause" на верхней панели и играет по клавишам под "минус", воспроизводимый макросами.

## **2.4. Сценарий «Запись»**

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Аналогично пунктам сценария «Игра на клавиатуре»
2. Пользователь выбирает один или несколько непустых (зелёных) макросов с записанным на каждый семплом на панели справа и однократно нажимает на выбранный(-ые) - макрос(-ы) переходит(-ят) в состояние готовности циклического проигрывания (жёлтый(-ые))
3. Пользователь нажимает кнопку "Record" на верхней панели и играет по клавишам под "минус", воспроизводимый макросами, записывая аудиофайл.
4. После повторного нажатия кнопки "Record" аудиозапись появляется на экране «My tracks»

## **2.5. Сценарий «Удаление макроса»**

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Пользователь открывает приложение.
2. Пользователь переходит к экрану «soundboard».
3. На панели сверху пользователь нажимает кнопку "корзина" и выбирает на панели справа непустой (серый макрос) которых хочет удалить и нажатием на этот макрос удаляет его

4. Повторным нажатием на кнопку корзины пользователь выходит из режима удаления макросов

## **2.6. Сценарий «Просмотр информации о разработчиках»**

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Пользователь открывает приложение.
2. Пользователь переходит к экрану «About devs».
3. Пользователь получает информацию о разработчиках

## **2.7. Сценарий «Просмотр списка записанных треков»**

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Пользователь открывает приложение.
2. Пользователь переходит к экрану «My tracks».
3. Пользователь получает список своих треков

## **2.8. Сценарий «Поиск записанного трека»**

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Пользователь открывает приложение.
2. Пользователь переходит к экрану «My tracks».
3. Пользователь использует сортировку (по дате)
4. Пользователь вводит в строку поиска название
5. Пользователь получает треки, подходящие под условия поиска

## **2.9. Сценарий «Просмотр руководства по использованию»**

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Пользователь открывает приложение.
2. Пользователь переходит к экрану «Help».
3. Пользователь получает руководство к приложению

**2.10. Сценарий «Прослушивание записанного трека»**

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Пользователь открывает приложение.
2. Пользователь переходит к экрану «My tracks».
3. Пользователь нажимает на трек
4. Открывается плеер и начинает играть трек

**2.11. Сценарий «Удалить записанный трек»**

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Пользователь открывает приложение.
2. Пользователь переходит к экрану «My tracks».
3. Пользователь зажимает иконку трека, при желании выбирает несколько треков
4. Пользователь нажимает удалить в всплывающем меню действий

**Альтернативный сценарий:**

1. Пользователь выполняет сценарий «Прослушивание записанного трека»
2. Пользователь нажимает кнопку мусорки для удаления трека

**2.12. Сценарий «Поделиться записанным треком»**

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Пользователь открывает приложение.

2. Пользователь переходит к экрану «My tracks».
3. Пользователь зажимает иконку трека, при желании выбирает несколько треков
4. Пользователь нажимает поделиться в всплывающем меню действий

**Альтернативный сценарий:**

1. Пользователь выполняет сценарий «Прослушивание записанного трека»
2. Пользователь нажимает кнопку, похожую на граф из 3 узлов, для того чтобы поделиться треком

**2.13. Сценарий «Переименовать записанный трек»**

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Пользователь открывает приложение.
2. Пользователь переходит к экрану «My tracks».
3. Пользователь зажимает иконку трека
4. Пользователь нажимает кнопку переименовывания в всплывающем меню действий

**2.14. Сценарий «Прикрепить семпл к клавише»**

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Пользователь открывает приложение.
2. Пользователь переходит к экрану "Configure sounds".
3. Пользователь нажимает на клавишу со знаком "+".
4. ИЗ списка сохранённых на устройстве аудиофайлов пользователь выбирает тот, который хочет прикрепить к нажатой клавише.

**2.15. Сценарий «Удалить семпл с клавиши»**

**Действующее лицо:**

Пользователь

**Основной сценарий:**

1. Пользователь открывает приложение.
2. Пользователь переходит к экрану "Configure sounds".
3. Пользователь нажимает на клавишу, с которой необходимо удалить семпл.
4. Приложение запрашивает подтверждение действия при помощи диалогового окна.

### 3. ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС

#### 3.1. Макет интерфейса с графом переходов

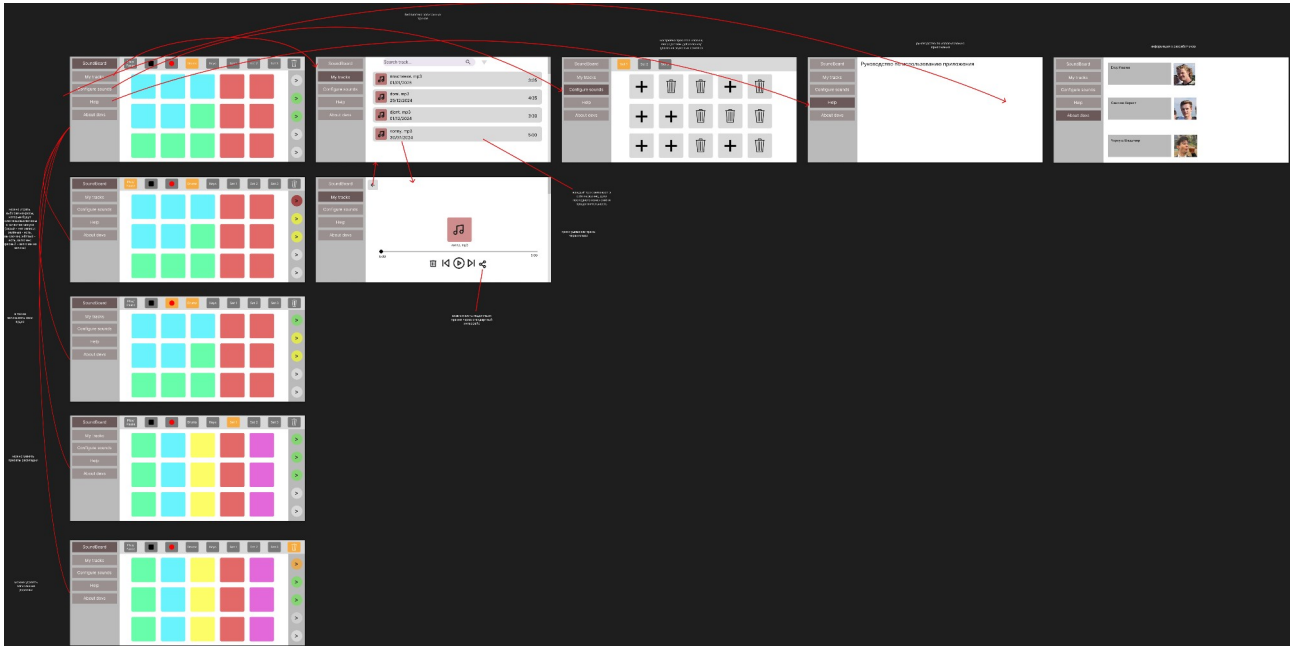


Рисунок 1 – Макет интерфейса с графом переходов

#### 3.2 Целевые устройства, обоснование требований и максимально подробные характеристики

Тип устройства: Смартфон

Модель: Google Pixel 9 Pro

Характеристики:

Версия API 35

Экран:

hw.lcd.density 480

hw.lcd.height 2856

hw.lcd.width 1280

Остальное:

avd.ini.displayname Pixel 9 Pro API 35

avd.ini.encoding UTF-8

AvdId Pixel\_9\_Pro\_API\_35

disk.dataPartition.size 2G

fastboot.chosenSnapshotFile  
fastboot.forceChosenSnapshotBoot no  
fastboot.forceColdBoot no  
fastboot.forceFastBoot yes  
hw.accelerometer yes  
hw.arc false  
hw.audioInput yes  
hw.battery yes  
hw.camera.back virtualscene  
hw.camera.front emulated  
hw.cpu.ncore 2  
hw.device.hash2 MD5:73e7b35d09e3a8055043aca4688e0dad  
hw.device.manufacturer Google  
hw.device.name pixel\_9\_pro  
hw.dPad no  
hw.gps yes  
hw.gpu.enabled yes  
hw.gpu.mode auto  
hw.initialOrientation portrait  
hw.keyboard yes  
hw.mainKeys no  
hw.ramSize 11548  
hw.sdCard yes  
hw.sensors.orientation yes  
hw.sensors.proximity yes  
hw.trackBall no  
image.androidVersion.api 35  
image.sysdir.1  
system-images/android-35/google\_api\_playstore/x86\_64/  
PlayStore.enabled true



runtime.network.latency none  
runtime.network.speed full  
showDeviceFrame yes  
skin.dynamic yes  
tag.display Google Play  
tag.displaynames Google Play  
tag.id google\_apis\_playstore  
tag.ids google\_apis\_playstore  
vm.heapSize 256

## **4. РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ**

### **4.1 Краткое описание**

Разработанное приложение представляет собой Android-решение для создания, хранения музыкальных треков.

Пользователи могут создавать свои музыкальные треки, играя клавишами приложения, при этом есть возможность настроить макросы и сконфигурировать клавиши. После записи трека он сохраняется на устройство.

Интерфейс приложения интуитивно понятен: пользователи могут легко освоиться с клавишами soundboard, плеером и меню конфигурации. В приложении предусмотрены возможности для быстрой отправки треков через социальные сети и мессенджеры, что упрощает процесс передачи композиций другим людям.

Музыкальные треки пользователя хранятся локально в файловой системе на устройстве и не требует подключения к интернету для работы с ним.

### **4.2 Схема архитектуры**

#### **4.2.1 Используемые технологии**

- **Язык программирования:** Kotlin
- **Генерация Kotlin кода:** KSP
- **UI:** XML
- **Звуковая библиотека:** oboe
- **Сериализация и десериализация:** Gson
- **Набор инструментов для разработки приложений:** Android SDK

#### **4.2.2 Архитектура приложения**

Приложение разработано с использованием XML, архитектура строилась по принципам MVVM (Model-View-ViewModel). XML отвечает за View (отображение), а логика реализуется в коде (Activity). Можно выделить компоненты архитектуры:

- Модель (Model)

Обеспечивает хранение и обработку данных (Song.kt)

- Представление (View)

Обеспечивает отображение UI, взаимодействие с пользователем (все XML-файлы разметки)

- Контроллер/ViewModel

Обработка логики, связь между Model и View (kotlin файлы Activity)

### 4.3 Используемые библиотеки

- AndroidX (Jetpack)
  - androidx.core:core-ktx: Расширение для Android Core API, упрощающее использование API Android с помощью Kotlin.
  - androidx.lifecycle:lifecycle-runtime-ktx: Предоставляет компоненты для реализации жизненного цикла Android-приложений, упрощая управление жизненным циклом компонентов.
  - androidx.appcompat: Совместимость с новыми фичами на старых устройствах.
  - androidx.constraintlayout: Гибкая система верстки (ConstraintLayout).
  - androidx.lifecycle: Компоненты для управления жизненным циклом:
    1. livedata-ktx (LiveData с поддержкой Kotlin).
    2. viewmodel-ktx (ViewModel для MVVM).
  - androidx.navigation: Навигация между фрагментами:
    1. navigation-fragment-ktx (Fragments + Kotlin).
    2. navigation-ui-ktx (UI-компоненты навигации).

- `androidx.activity`: Улучшенный Activity API.
- Google/Material
  - `material`: Material Design компоненты (кнопки, карточки и т.д.).
- Тестирование
  - `junit`: Юнит-тесты (JUnit 4).
  - `androidx.junit`: Инструментальные тесты (JUnit для Android).
  - `androidx.espresso:espresso-core`: UI-тесты (Espresso).
- Gson
  - `gson`: Библиотека для сериализации и десериализации Java-объектов в JSON и обратно.
- Нативные компоненты
  - `externalNativeBuild`: Сборка нативного кода через CMake:
  - `arguments("-DANDROID_STL=c++_shared")`: Используется общая STL (Standard Template Library) для C++.
  - `path = file("src/main/cpp/CmakeLists.txt")`: Указывает на файл конфигурации CMake.
  - `prefab = true`: Поддержка Prefab (упрощение интеграции нативных библиотек).

## **5. ВЫВОДЫ**

### **5.1 Достигнутые результаты**

В ходе работы было разработано мобильное приложение «Soundboard», позволяющее создавать собственные музыкальные треки, играя на клавишах устройства. В приложении доступны функции: игра на клавишах приложения, запись и сохранение треков, настройка макросов и пресетов клавиш, плеер для созданных треков с возможностью поделиться.

### **5.2 Недостатки и пути для улучшения**

- 1) Можно улучшить качество записи треков.
- 2) Добавить поддержку не только .wav, но и других популярных форматов (.mp3, .flac).

### **5.3 Будущее развитие решения**

В качестве развития решения возможны:

- 1) Локализация для поддержки нескольких языков.
- 2) Улучшение ui для поддержки большего количества смартфонов

## 6. ЛИТЕРАТУРА

1. Ссылка на GitHub. - [Электронный ресурс]. - URL: <https://github.com/moevm/adfmp1h25-sound> (дата обращения: 05.02.2025).
2. Android Studio. – [Электронный ресурс]. – URL: <https://developer.android.com/studio> (дата обращения: 03.02.2025).
3. Kotlin. – [Электронный ресурс]. – URL: <https://kotlinlang.org/> (дата обращения: 03.02.2025).
4. Овое. – [Электронный ресурс]. – URL: <https://github.com/google/oboe> (дата обращения: 03.02.2025).

## 7. ПРИЛОЖЕНИЯ

### 7.1 Инструкция для пользователя

1. На странице “Soundboard” пользователю доступен функционал проигрывания семплов и записи собственных треков. Кнопки на верхней панели позволяют начать (кнопка записи) и завершить (кнопка стоп) запись музыкальной дорожки.

Кнопки с изображением барабана и клавиш пианино позволяют выбрать стандартные ударный и клавишный пресеты. Кнопки с цифрами 1, 2 и 3 позволяют выбрать пресеты, которые пользователь может настроить самостоятельно на странице “Sound configuration”. Кнопка с рисунком корзины включает режим удаления макросов по нажатию.

На правой панели находятся кнопки макросов (по умолчанию серые). Нажатие на серую клавишу делает её красной через 2 секунды и сразу после этого начнёт запись выбранного макроса - играйте во время записи. Нажатие на красную кнопку остановит запись и зафиксирует макрос, перекрасив кнопку в желтый. Нажатие на желтую кнопку начнет воспроизведение макроса в режиме loop. Повторное нажатие остановит воспроизведение по окончании проигрывания loop'a.

2. На странице “My tracks” пользователю представлены все аудиозаписи с возможностью поиска по названию и выбора для прослушивания. Выбранный трек можно удалить или отправить через стандартный интерфейс.

3. Страница “Sound configuration” позволяет пользователю настроить собственные пресеты, разместив на кнопках семплы, сохраненные на устройстве. Всего возможно создать 3 пресета. Для сохранения изменений необходимо нажать на соответствующую кнопку на верхней панели. Добавлять следует только .wav файлы, иначе вы обрекаете себя на то, что что-то может где-то не заработать.

## 7.2 Снимки экрана приложения

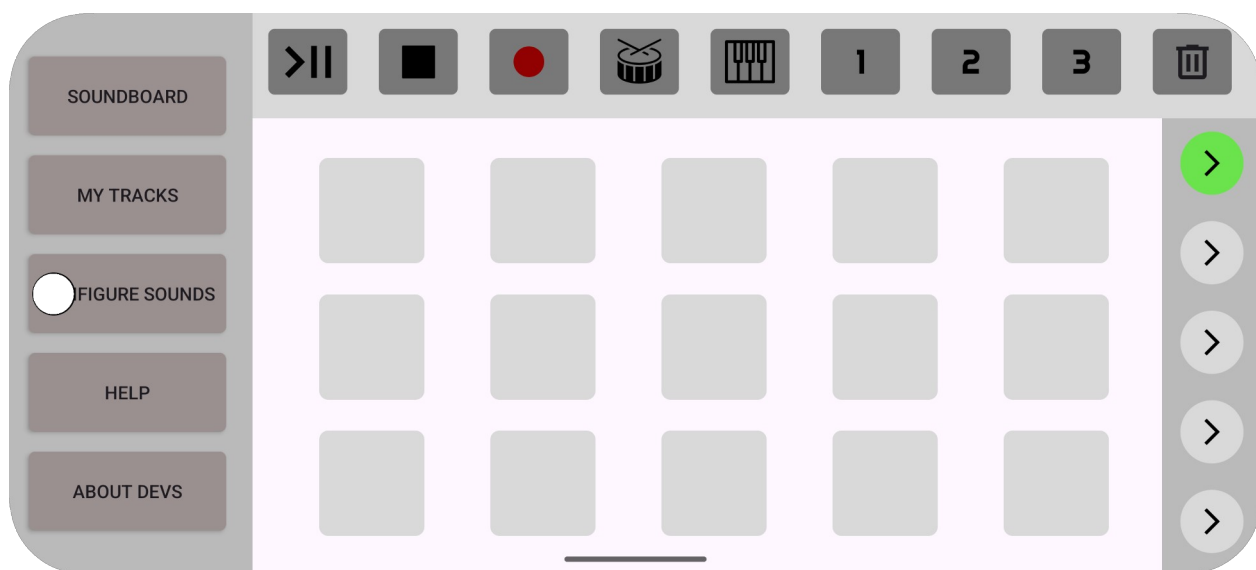


Рисунок 2 – Главный экран

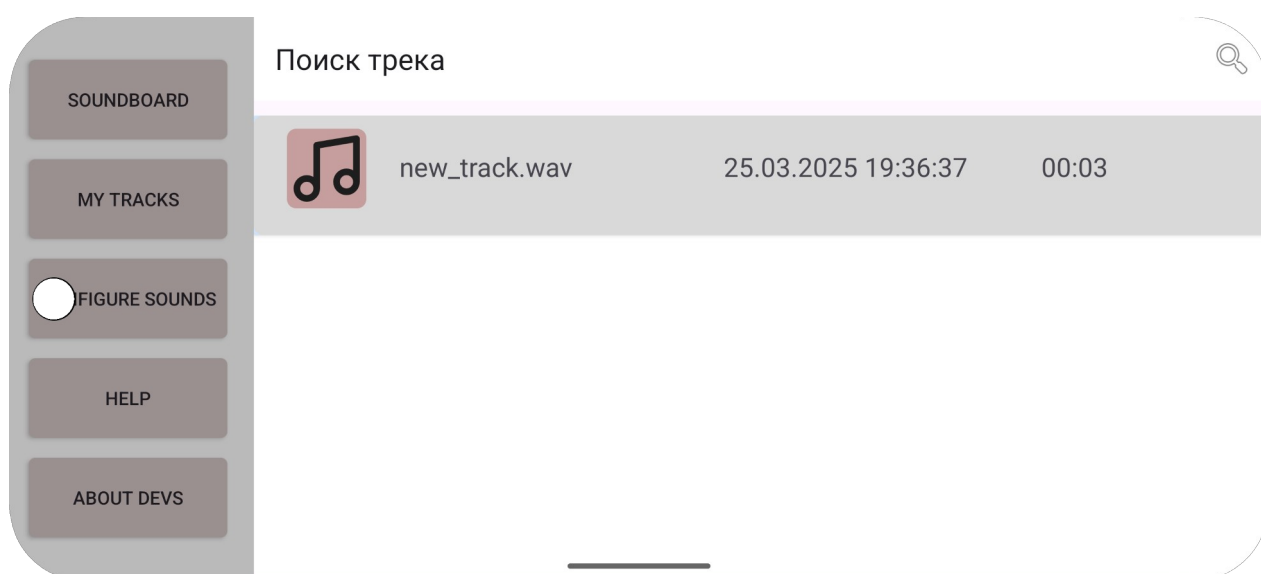


Рисунок 3 – Экран записанных треков



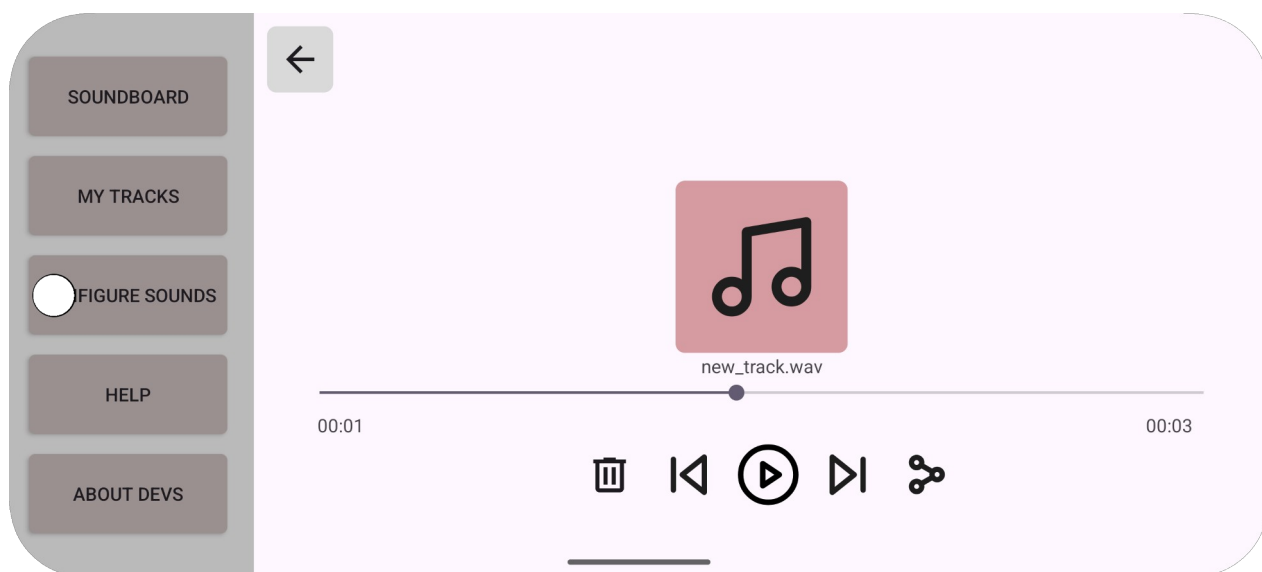


Рисунок 4 – Экран плеера

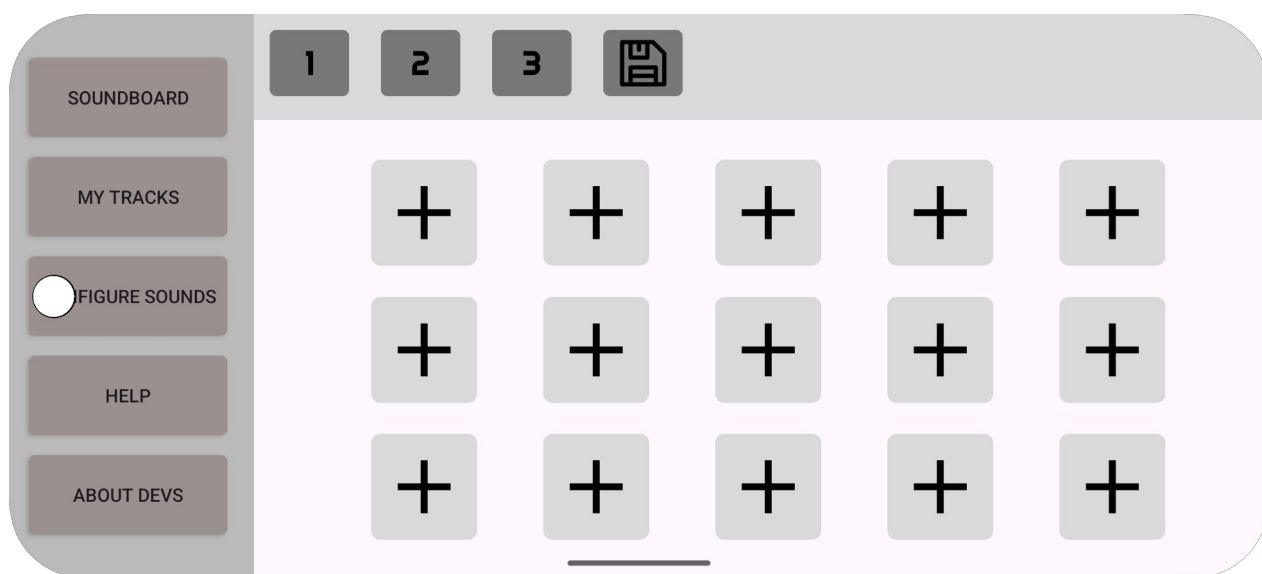


Рисунок 5 – Экран конфигурирования клавиш

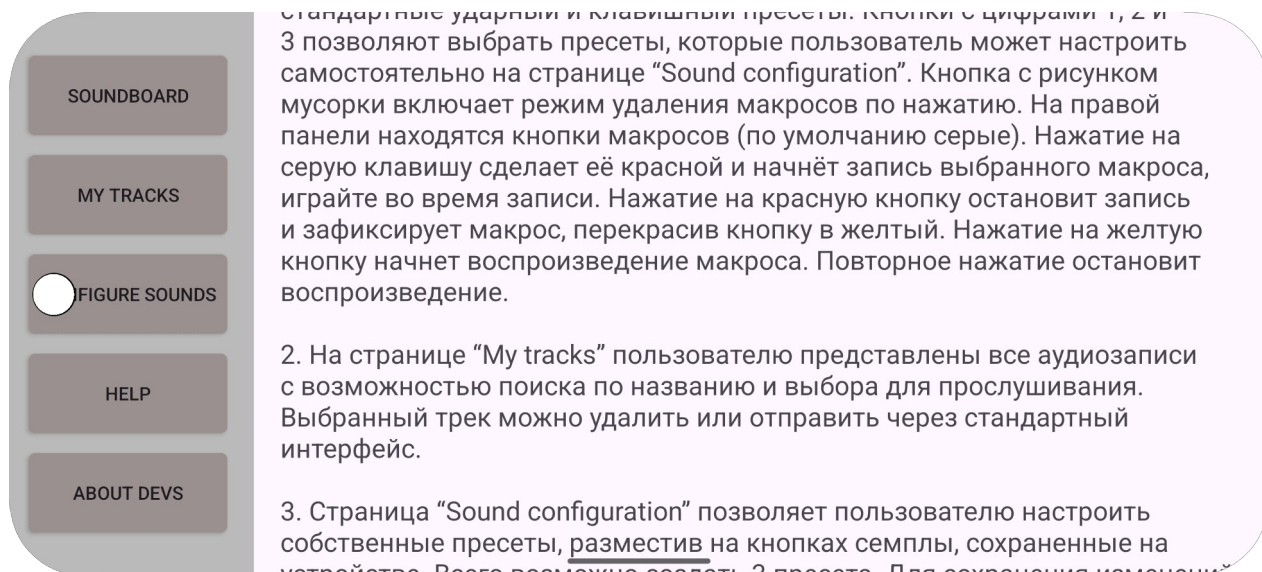


Рисунок 6 – Экран помощи

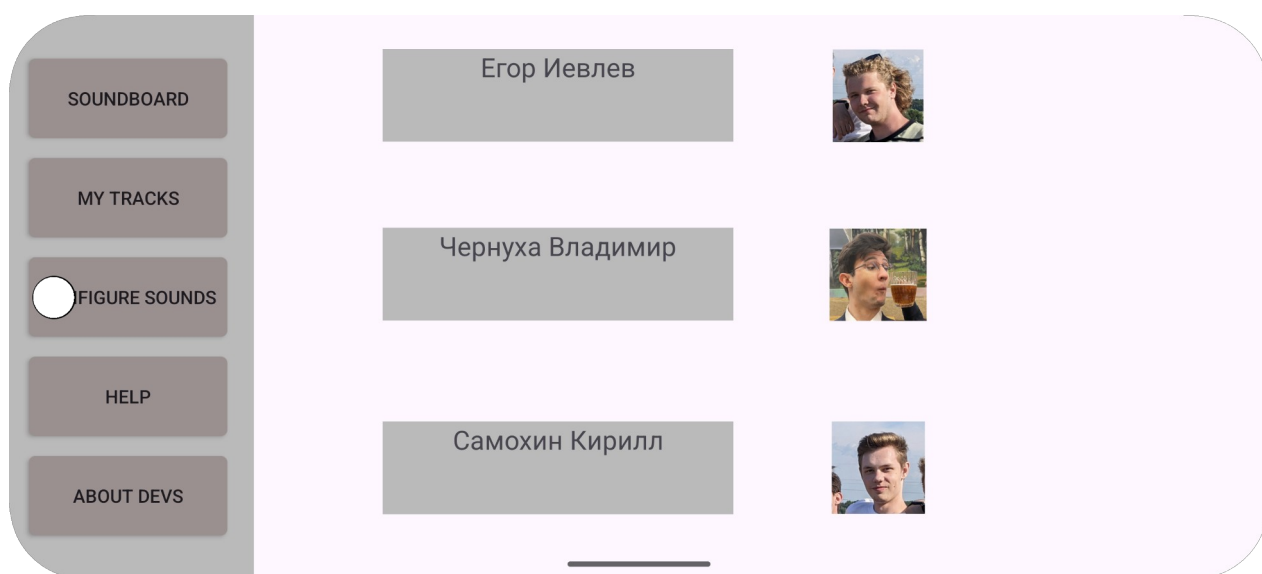


Рисунок 7 – Экран о разработчиках