**Interactive Lock**

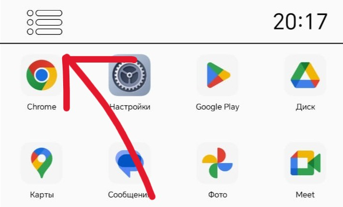
**Введение**

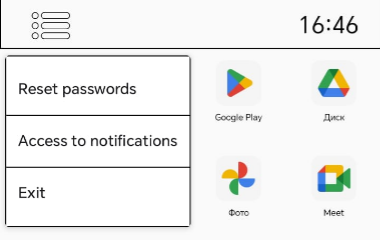
Interactive Lock – это инновационный лаунчер для Android, разработанный на Kotlin с использованием Jetpack Compose. Он предоставляет пользователям уникальный и гибкий способ защиты приложений, выходящий за рамки стандартных пин-кодов и паролей. Interactive Lock позволяет персонализировать защиту каждого приложения индивидуально, предлагая несколько методов разблокировки и настраиваемый интерфейс.

**Основные характеристики**

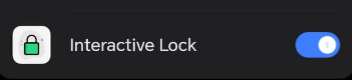
* **Индивидуальная защита приложений:** Выберите уникальный метод блокировки для каждого приложения.
* **Разнообразие методов блокировки:**
  + **BongoCat Lock:** Разблокировка на основе последовательности нажатий на лапки BongoCat. Веселый и интуитивно понятный способ.
  + **Voice Lock:** Разблокировка голосом. Произнесите заданную фразу для доступа.
  + **Text Lock:** Традиционный пароль. Надежный и проверенный метод.
  + **No Lock:** Отключение блокировки для конкретного приложения.
* **Легкая настройка:** Простой и интуитивно понятный процесс настройки блокировки для каждого приложения.
* **Восстановление доступа:** Механизм сброса пароля/последовательности с использованием контрольного вопроса.
* **Ненавязчивые уведомления:** Визуальные индикаторы уведомлений (красные кружочки) на иконках приложений.
* **Разработано на Kotlin с использованием Jetpack Compose:** Современная и поддерживаемая кодовая база.

**Как это работает (Руководство пользователя)**

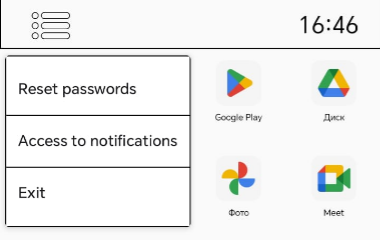
1. **Установка:**
   * Установите Interactive Lock из магазина приложений.
   * Предоставьте необходимые разрешения для корректной работы лаунчера (доступ к микрофону для Voice Lock, доступ к уведомлениям для индикаторов уведомлений).
2. **Настройка блокировки приложения:**
   * На главном экране Interactive Lock найдите иконку приложения, которое хотите защитить.
   * Нажмите и удерживайте (долгое нажатие) иконку приложения.
   * Появится меню выбора типа блокировки: BongoCat Lock, Voice Lock, Text Lock, No Lock.
   * Выберите желаемый тип блокировки.
3. **Настройка выбранного типа блокировки:**
   * **BongoCat Lock:** Запомните и повторите показанную последовательность нажатий на лапки BongoCat.
   * **Voice Lock:** Запишите секретную голосовую фразу.
   * **Text Lock:** Придумайте и введите надежный пароль.
4. **Включение уведомлений:**
   * Перейдите в меню в левом верхнем углу. 
   * Найдите пункт “Access to notifications”



* + Перейдите в настройки уведомлений и дайте разрешение приложению Interactive Lock



1. **Смена блокировки:**
   * Повторите шаги 2 и 3, чтобы изменить тип блокировки или пароль/последовательность для приложения.
2. **Сброс всех блокировок:**
   * Если вы забыли пароль, фразу или последовательность BongoCat, нажмите на кнопку “Reset passwords” (В раскрывающимся списке после нажатия на кнопку меню в левом верхнем углу экрана).



* + Ответьте на контрольный вопрос, который вы задали при первом запуске Interactive Lock.
  + После успешного ответа все блокировки будут сброшены, и вы сможете настроить их заново.

**Техническая документация (для разработчиков)**

**Язык программирования:** Kotlin

**Версия JDK:** 11

**Минимальный SDK:** 28 (Android 9.0)

**Целевой SDK:** 35 (Android 14)

**UI Framework:** Jetpack Compose

**Архитектура: MVVM (Model-View-ViewModel)**

**Model:**

* **AppModel -** основная модель данных приложения
* **AppModelsAdapter -** адаптер для работы со списком приложений

**View:**

* **AppView -** представление для отдельного приложения
* **Compose UI** компоненты в активностях

**ViewModel:**

* **AppViewModel -** управление состоянием и логикой приложения

**Основные зависимости**

dependencies {

    // Compose

    implementation(libs.androidx.activity.compose)

    implementation(libs.androidx.ui)

    implementation(libs.androidx.ui.graphics)

    implementation(libs.androidx.material3)

    // Утилиты

    implementation("com.github.GrenderG:Toasty:1.5.2")

    implementation(libs.androidx.core.ktx)

    implementation(libs.androidx.lifecycle.runtime.ktx)

}

**Структура проекта**

**UI (ui/)**

* **MainActivity.kt** - главный экран приложения

Composable функции для отображения:

* + Списка приложений
  + Меню настроек
  + Индикаторов уведомлений

**ViewModel (viewModel/)**

* **AppViewModel.kt:**
  + Управление состоянием приложения
  + Обработка пользовательских действий
  + Взаимодействие с хранилищем данных

**Model (model/)**

* **AppModel.kt:**
  + Данные приложения (имя, иконка, пакет)
  + Тип блокировки
  + Логика работы с паролями
* **AppModelsAdapter.kt:**
  + Адаптер для работы со списком приложений

**Locks (locks/)**

* **BongoLockActivity.kt** - реализация BongoCat блокировки
* **VoiceActivity.kt** - голосовая блокировка
* **TextLockActivity.kt** - текстовая блокировка
* **NoLockActivity.kt** - отсутствие блокировки
* **SetLockActivity.kt** - настройка типа блокировки

**Сервисы (notifyService/)**

* **NLService.kt** - сервис для обработки уведомлений

**Хранение данных:** SharedPreferences.

**Расширяемость и вклад**

Interactive Lock спроектирован с учетом возможности расширения.

В планах:

* Добавить новые типы блокировок
* Улучшить UI
* Оптимизация производительности

**Известные проблемы и ограничения**

* **Хранение блокировок без шифрования**
* **Не очень большой выбор типов блокировок**

**Тестирование**

**В проекте реализованы модульные тесты для двух основных классов:**

1. **AppModelTest - тесты для модели приложения**
2. **AppViewModelTest - тесты для ViewModel приложения**

**AppModelTest тестирует основную логику работы с моделью приложения (AppModel). Реализованы тесты для проверки установки пароля и корректного определения типа блокировки.**

**AppViewModelTest тестирует логику ViewModel, которая управляет взаимодействием между UI и моделью данных.**

**Проверяет обработку уведомления в SharedPreferences при различных сценариях разблокировки (успешной и нет), корректность передачи операции установки пароля в модель, корректность получения имени из модели и делегирования запуска активности.**

**Используемые технологии тестирования:**

**JUnit4 - основной фреймворк тестирования**

**Mockito - для создания mock-объектов**