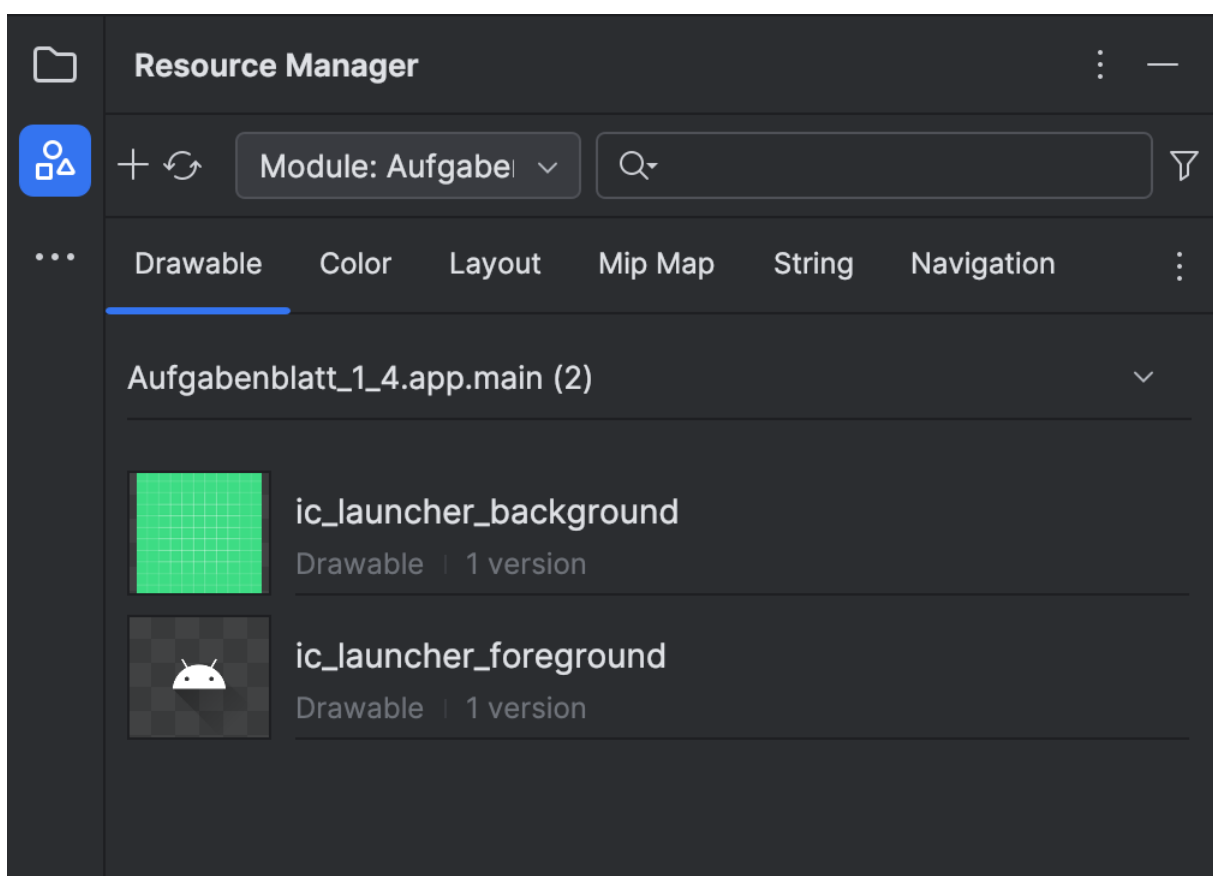


Aufgabe 1.4

Info: ChatGPT 4o wurde teilweise zu Recherchezwecken verwendet.

- ☐ Create sample projects with **File > New > New Project** and inspect:
 - Resource Manager Window
 - Project Window
 - Structure Window

Resource Manager Window:



Drawable: Bilder, Icons, Vektorgrafiken

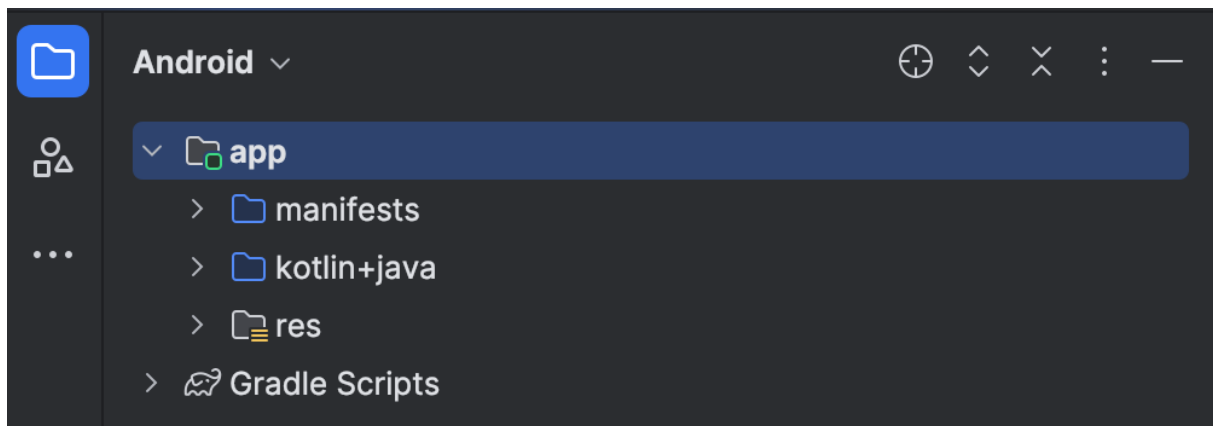
Color: Farben aus colors.xml

Layout: XML-Dateien für das Layout Design

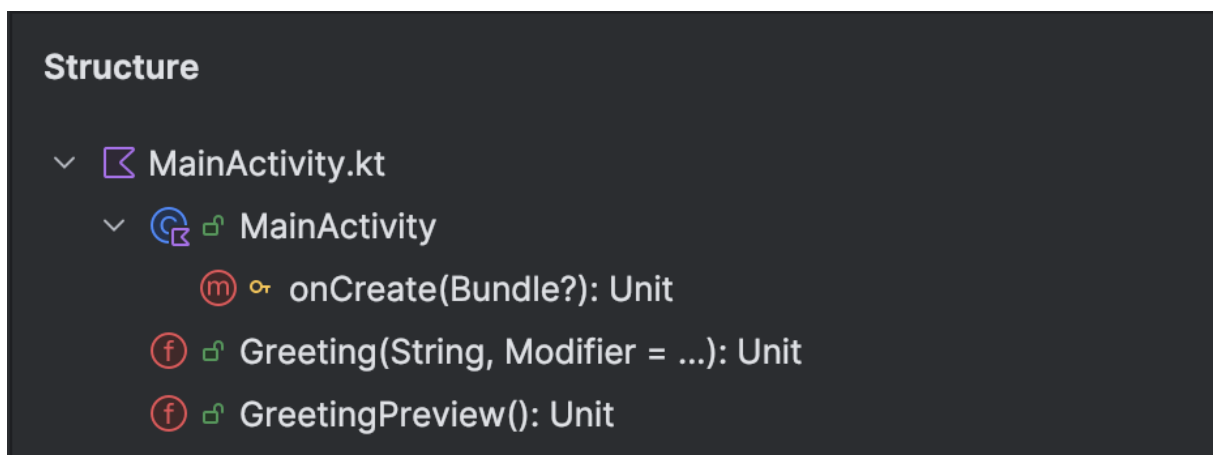
Mip Map: Launcher Icons – Verbessert die Darstellung von Icons auf verschiedenen Bildschirmgrößen und Auflösungen

String: Texte aus strings.xml

Navigation: Wurde in älteren Android Studio Versionen dazu verwendet, die Navigationsgraphik (nav_graph.xml) visuell zu verwalten, wird nun im Navigation Editor gehandled

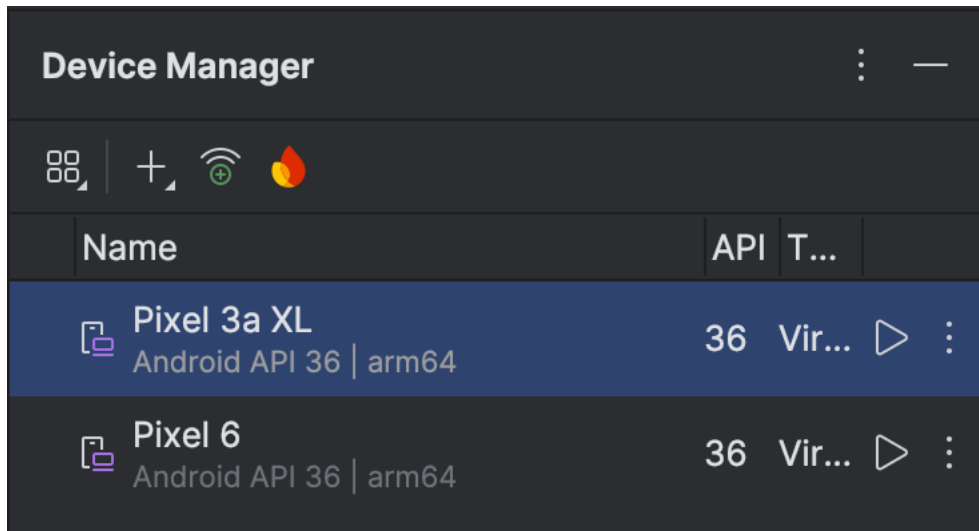
Project Window:

Zeigt die ganze Struktur einer Android Applikation. Es enthält Code, Ressourcen, Konfigurationsdateien und Build Skripte.

Structure Window:

Zeigt die wichtigsten Elemente einer Datei (z.B. einer Klasse) in einer Baumstruktur.

- ☐ Install and launch the Android Emulator
 - ☐ Optional: Enable hardware acceleration
 - ☐ Create and manage virtual devices with different properties
 - ☐ Explore how to perform basic control actions with preinstalled apps
 - ☐ Explore how to use extended controls (location, displays, sensors, ...) with preinstalled apps

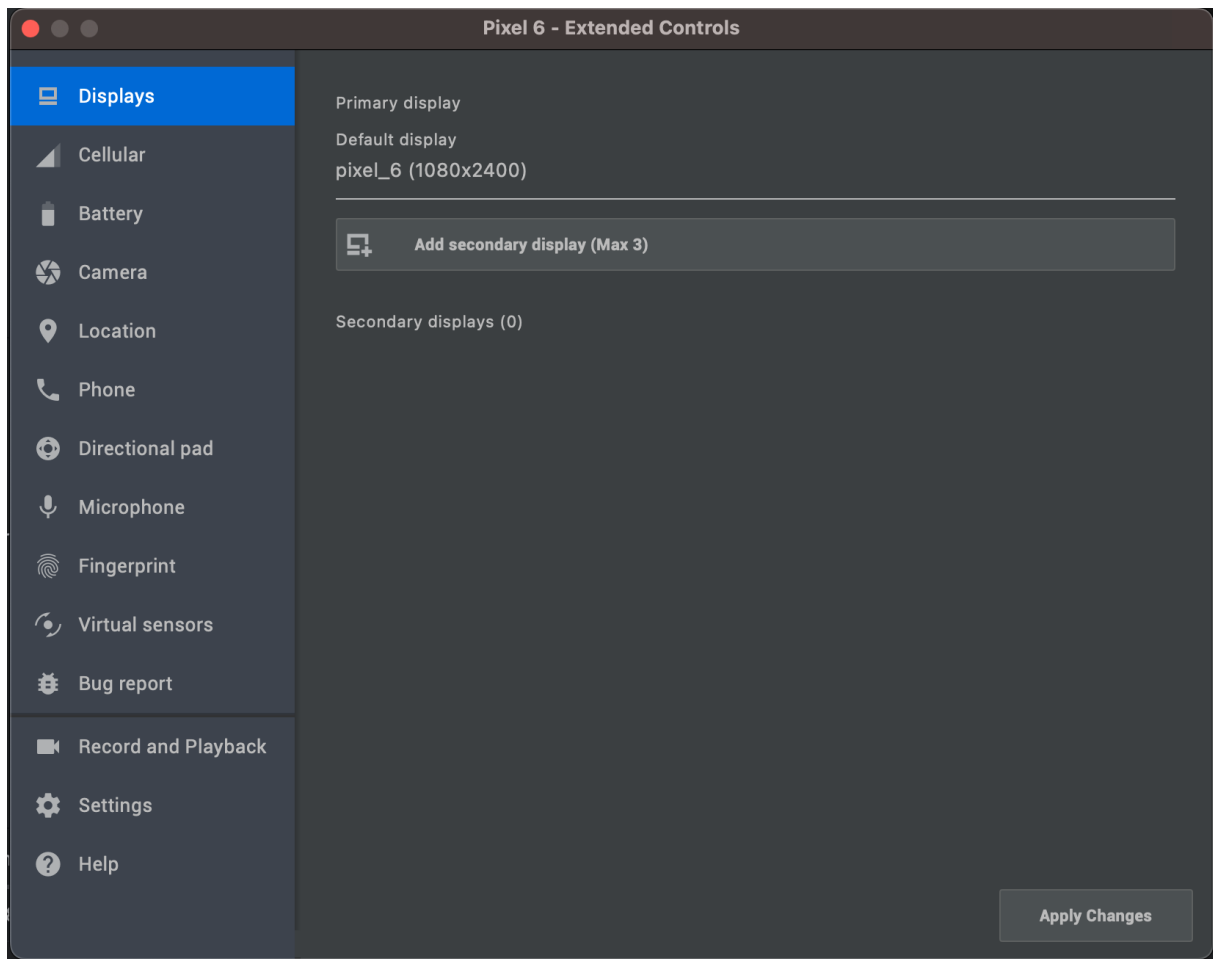


2 Android Devices mit unterschiedlichen Properties (z.B. RAM, Modell usw.). Zum Ändern (⋮) drücken.

Basic Controls:

Oben in der Bar kann man das Device unter anderem Ausschalten, Lautstärke ändern, rotieren Screenshots machen und den Screen recorden.

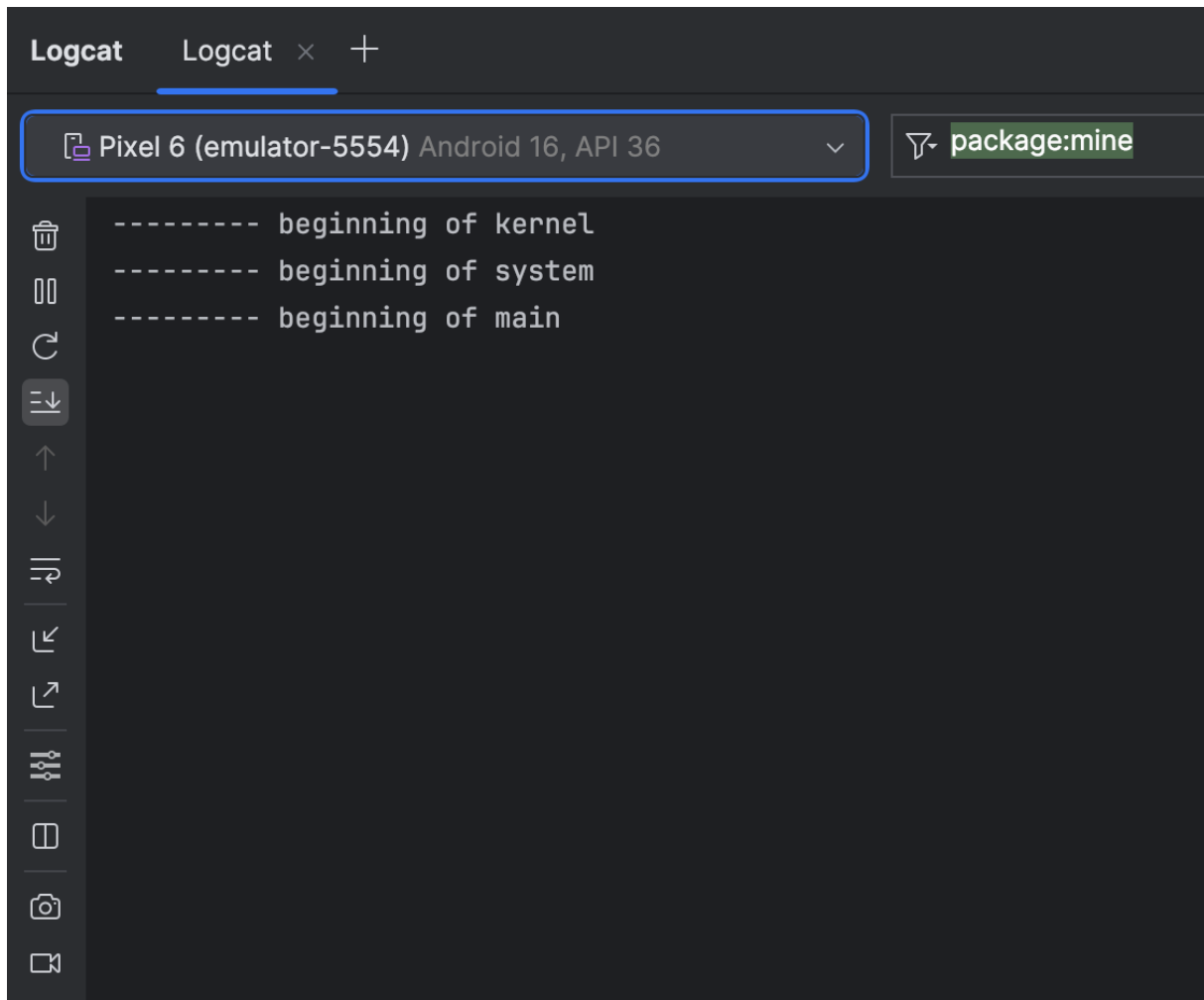
Ein Mausklick simuliert einen „Touch“ und bei gedrückter Maustaste wischen kann man scrollen am Bildschirm.

Extended Controls:

Mit einem Klick auf (:) im Emulator gelangt man zu den erweiterten Controls. In diesem Fenster kann man die Sensoren des Devices testen / simulieren.

☐ Explore Logcat

- ☐ Get familiar with the Logcat window in Android Studio
- ☐ Explain the purpose of logcat and how messages can be filtered



Logcat zeigt alle Logs des Android-Emulators bzw. echten Geräts:

Log.e(„TAG“, „Error“)	Error (E/)
Log.w(„TAG“, „Warning“)	Warning (W/)
Log.i(„TAG“, „Information“)	Information (I/)
Log.d(„TAG“, „Debugging-Info“)	Debug (D/)
Log.v(„TAG“, „Detaillierte Logs“)	Verbose (V/)

Filter:

```
14      import android.util.Log
15
16      class MainActivity : AppCompatActivity() {
17          override fun onCreate(savedInstanceState: Bundle?) {
18              super.onCreate(savedInstanceState)
19              Log.d( tag: "MainActivity", msg: "onCreate wurde aufgerufen")
19          }
19      }
```

Pixel 6 (emulator-5554) Android 16, API 36 tag:main

----- PROCESS STARTED (5233) for package aaU.ss25.apde.aufgabenblatt_1_4 -----
2025-03-20 11:11:35.565 5233-5233 MainActivity aaU.ss25.apde.aufgabenblatt_1_4 D onCreate wurde aufgerufen

□ Inspect the Device File Explorer Window in Android Studio

Device Explorer			
Pixel 6 Android API 36			
Files Processes			
Name	Permissions	Date	Size
✓ /	drwxr-xr-x	2025-03-11 08:49	4 KB
> / apex	drwxr-xr-x	2025-03-20 10:40	2 KB
> / bin	lrw-r--r--	2025-03-11 08:42	11 B
> / bootstrap-apex	drwxr-xr-x	2025-03-20 10:40	280 B
> / cache	drwxrwx---	2025-03-11 08:42	4 KB
> / config	drwxr-xr-x	1970-01-01 01:00	0 B
> / d	lrw-r--r--	2025-03-11 08:42	17 B
> / data	drwxrwx--x	2025-03-20 10:41	4 KB
> / data_mirror	drwxrwx---	2025-03-20 10:40	180 B

Zeigt das komplette Filesystem des Emulators bzw. Geräts. In diesem Manager kann man Dateien hoch- und herunterladen sowie Dateien löschen.

4.2 Gradle

Gradle is used as build system in Android Studio. Android studio automatically generates gradle build files. These build files might be sufficient for small projects but need to be manually edited for release builds or more complex software.

- ☐ Get a closer look at gradle and describe the terms, e.g., build type, product flavor, build variants, dependencies.
- ☐ Describe the build process that converts a project into an Android Application Package (APK).
- ☐ Describe following files used in gradle:
gradle.properties, settings.gradle[.kts], build.gradle[.kts]

Build Type:

„Build Type“ ist eine Definition verschiedener Build-Konfigurationen:

- debug: für Entwickler (mehr Logs, Debugging-Tools)
- release: optimiert für den Endnutzer

Product Flavor:

Ermöglicht verschiedene App-Variationen mit eigenen Konfigurationen (z.B. Gratis- und Premium-Version)

Build Variants:

Eine Kombination aus Build Type und Product Flavor (z.B freeDebug, freeRelease, premiumDebug, premiumRelease)

Dependencies:

Gradle verwendet Dependencies zum Hinzufügen von Bibliotheken

Build Process:

Ein Android-Projekt wird in mehreren Schritten zu einer .apk-Datei umgewandelt:

- 1) Quellcode wird kompiliert → Kotlin zu Bytecode
- 2) Ressourcen werden verarbeitet → XMLs / Bilder werden ins apk-Format gebracht
- 3) Dex-Optimierung → Bytecode wird in .dex-Dateien umgewandelt
- 4) ProGuard / R8 (optional) → Code-Minimierung und -Optimierung
- 5) Signierung → .apk wird mit einem Zertifikat signiert
- 6) APK wird generiert und ist bereit für die Installation

gradle.properties:

Enthält globale Gradle-Optionen und Umgebungsvariablen

```
android.useAndroidX=true
# Kotlin code style for this project: "official" or "obsolete":
kotlin.code.style=official
```


settings.gradle.kts:

Definiert, welche Module im Projekt enthalten sind

```
pluginManagement {
    repositories {
        google {
            content {
                includeGroupByRegex("com\\.android.*")
                includeGroupByRegex("com\\.google.*")
                includeGroupByRegex("androidx.*")
            }
        }
        mavenCentral()
        gradlePluginPortal()
    }
}

dependencyResolutionManagement {
    repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS)
    repositories {
        google()
        mavenCentral()
    }
}

rootProject.name = "Aufgabenblatt_1_4"
include(...projectPaths: ":app")
```

build.gradle.kts:

Ist die Hauptkonfigurationsdatei für Module (z.B. app/build.gradle.kts) und beinhaltet außerdem noch Abhängigkeiten, Plugins und Build-Typen.

```
plugins {  
    ⚡ alias(libs.plugins.androidApplication)  
    alias(libs.plugins.jetbrainsKotlinAndroid)  
}  
  
android {  
    namespace = "aau.ss25.apde.aufgabenblatt_1_4"  
    compileSdk = 34  
  
    defaultConfig {  
        applicationId = "aau.ss25.apde.aufgabenblatt_1_4"  
        minSdk = 26  
        targetSdk = 34  
        versionCode = 1  
        versionName = "1.0"  
  
        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"  
        vectorDrawables {  
            useSupportLibrary = true  
        }  
    }  
  
    buildTypes {  
        release {  
            isMinifyEnabled = false  
            proguardFiles(  
                getDefaultProguardFile(name: "proguard-android-optimize.txt"),  
                "proguard-rules.pro"  
            )  
        }  
    }  
    compileOptions {
```