SVEUČILIŠTE U RIJECI

FAKULTET INFORMATIKE I DIGITALNIH TEHNOLOGIJA

Preddiplomski studij informatike

2.Seminarski rad

Mrežni i mobilni operacijski sustavi

Autor: Zara Čubranić

U Rijeci, siječanj 2024.

Sadržaj

[ZADATAK 3](#_Toc156442836)

[KREIRANJE MOBILNE APLIKACIJE 4](#_Toc156442837)

[APLIKACIJA SA SENZOROM SVJETLINE 6](#_Toc156442838)

[KOD APLIKACIJE 7](#_Toc156442839)

[PROFILIRANJE APLIKACIJE 10](#_Toc156442840)

## ZADATAK

Potrebno je kreirati mobilnu aplikaciju koristeći alat po vlastitom izboru, preferirajući Expo platformu. Fokus je na implementaciji osnovnih značajki, senzora, interakciji korisnika te osnovnom profiliranju aplikacije.

Senzori koje mogu uključiti u mobilnu aplikaciju:

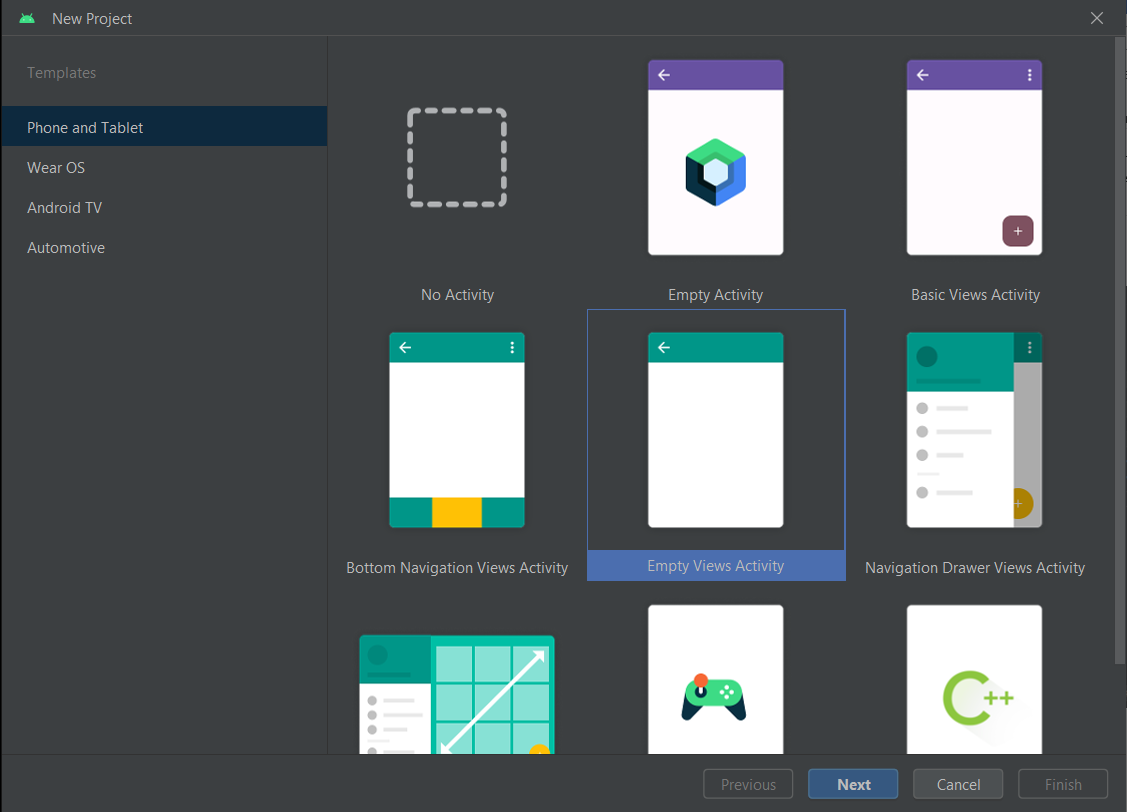
* Akcelometar
* Barometar
* Senzor pomaka
* Žiroskop
* Senzor svjetla
* Senzor magnetskog toka
* Pedometar

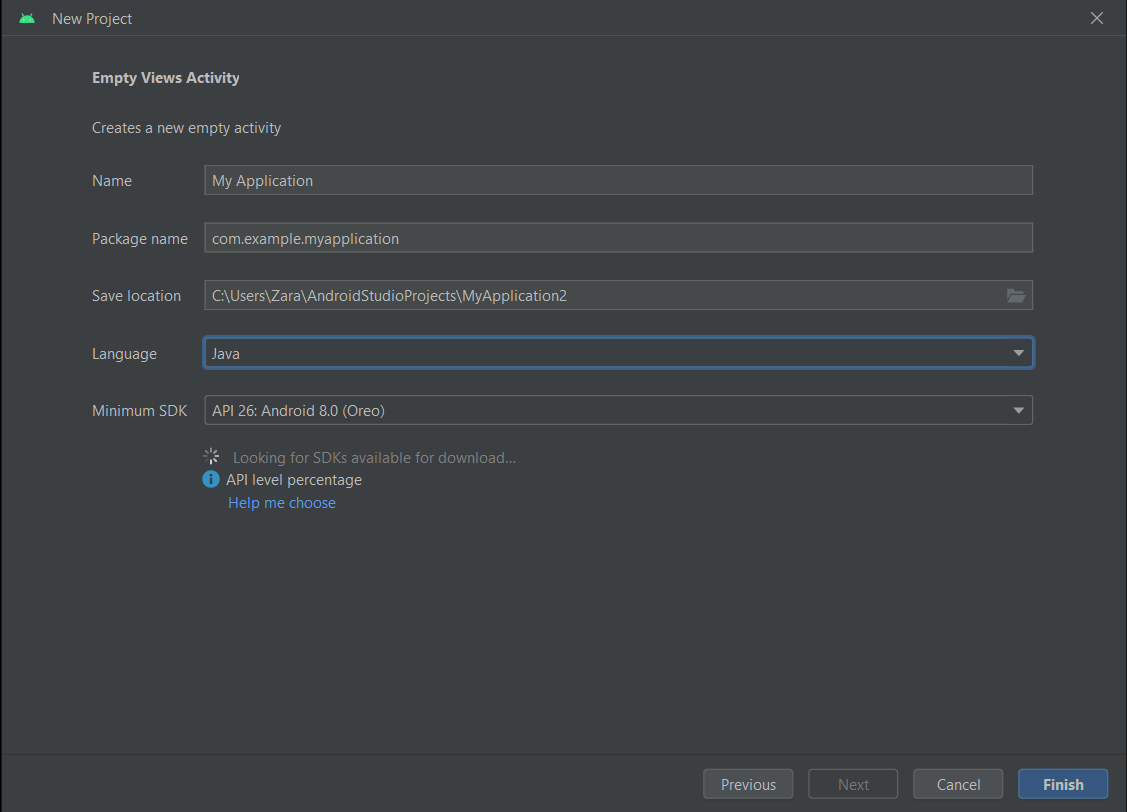
Za svoju aplikaciju sam izabrala senzor svjetline.

## KREIRANJE MOBILNE APLIKACIJE

Budući da imam Android mobitel i iskustva u korištenju Android Studia i programskog jezika Java, odlučila sam se da ću to koristiti u izradi svoje mobilne aplikacije.

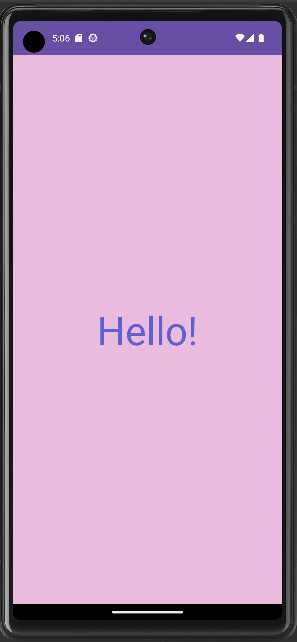
Nakon otvaranja Android Studia, stvorila sam novi projekt te odabrala Empty Views Activity. Zatim sam promjenila jezik iz Kotlina u Java.





Napravila sam jednostavnu aplikaciju sa tekstom i pozadinom u boji. Boje sam najprije dodala u datoteku colors.xml (res/values/colors.xml). U nastavku slijedi kod datoteke activity\_main.xml (res/layout/activity\_main.xml).

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/pink"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/helloTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textColor="@color/blue"  
 android:text="Hello!"  
 android:textSize="60sp"  
 android:fontFamily="@font"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>



## APLIKACIJA SA SENZOROM SVJETLINE

Kroz Expo platformu, dodala sam senzor za ambijentalnu svjetlost u svoj Android projekt. Instalirala sam senzor naredbom expo install expo-sensors, omogućujući aplikaciji praćenje okoliša. U projektu koristim SensorManager i SensorEventListener za praćenje promjena svjetlosti.

Aplikacija reagira na okoliš prikazujući sliku mjeseca na tamno plavoj pozadini kada je mrak, a mjesec se pomiče s lijeva na desno. Kada je svjetlo, prikazuje se slika sunca na svijetlo plavoj pozadini, a sunce također ide s lijeva na desno. Tekstualni prikaz se mijenja ovisno o situaciji (Laku noć ili Dobar dan).

## KOD APLIKACIJE

MainActivity.java

package com.example.myapplication;  
  
import android.animation.ObjectAnimator;  
import android.graphics.Color;  
import android.hardware.Sensor;  
import android.hardware.SensorEvent;  
import android.hardware.SensorEventListener;  
import android.hardware.SensorManager;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.ImageView;  
import android.widget.TextView;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class MainActivity extends AppCompatActivity implements SensorEventListener {  
  
 private SensorManager sensorManager;  
 private Sensor lightSensor;  
 private TextView greetingTextView;  
 private ImageView imageView;  
 private View backgroundView;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 greetingTextView = findViewById(R.id.*helloTextView*);  
 imageView = findViewById(R.id.*imageView*);  
 backgroundView = findViewById(R.id.*backgroundView*);  
  
 sensorManager = (SensorManager) getSystemService(*SENSOR\_SERVICE*);  
  
 // Provjera dostupnosti senzora svjetline  
 lightSensor = sensorManager.getDefaultSensor(Sensor.*TYPE\_LIGHT*);  
  
 if (lightSensor != null) {  
 sensorManager.registerListener(this, lightSensor, SensorManager.*SENSOR\_DELAY\_NORMAL*);  
 } else {  
 greetingTextView.setText("Senzor svjetla nije dostupan");  
 }  
  
 ObjectAnimator animator = ObjectAnimator.*ofFloat*(imageView, "translationX", -500f, 500f);  
 animator.setDuration(5000);  
 animator.setRepeatCount(ObjectAnimator.*INFINITE*);  
 animator.start();  
 }  
  
 @Override  
 public void onSensorChanged(SensorEvent event) {  
 if (event.sensor.getType() == Sensor.*TYPE\_LIGHT*) {  
 float ambientLight = event.values[0];  
  
 if (ambientLight < 10) {  
 backgroundView.setBackgroundColor(getResources().getColor(R.color.*darkBlue*));  
 } else {  
 backgroundView.setBackgroundColor(getResources().getColor(R.color.*lightBlue*));  
 }  
  
 // Prilagodba teksta i slike na temelju svjetlosti  
 if (ambientLight < 10) {  
 imageView.setImageResource(R.drawable.*moon*);  
 setGreetingText("Laku noć");  
 } else {  
 imageView.setImageResource(R.drawable.*sun*);  
 setGreetingText("Dobar dan");  
 }  
 }  
 }  
  
 @Override  
 public void onAccuracyChanged(Sensor sensor, int accuracy) {  
 }  
  
 private void setGreetingText(String greeting) {  
 greetingTextView.setText(greeting);  
 }  
  
 @Override  
 protected void onDestroy() {  
 sensorManager.unregisterListener(this);  
 super.onDestroy();  
 }  
}

activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/blue"  
 tools:context=".MainActivity">  
  
 <View  
 android:id="@+id/backgroundView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"/>  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/moon"  
 android:layout\_centerInParent="true" />  
  
 <TextView  
 android:id="@+id/helloTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Hello!"  
 android:textSize="24sp"  
 android:textColor="@android:color/white"  
 android:layout\_centerInParent="true"/>  
</RelativeLayout>

colors.xml

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="black">#FF000000</color>  
 <color name="white">#FFFFFFFF</color>  
 <color name="pink">#EBBBDD</color>  
 <color name="blue">#5C62D6</color>  
 <color name="darkBlue">#001F3F</color>  
 <color name="lightBlue">#ADD8E6</color>  
</resources>

**onCreate():**

Metoda onCreate() aktivira se pri pokretanju aplikacije te je odgovorna za inicijalizaciju raznih elemenata i postavljanje početnih uvjeta.

Nakon toga, inicijalizira se SensorManager kako bi se omogućila komunikacija s senzorima uređaja. Provjerava se dostupnost senzora svjetline (lightSensor), te se, ako je prisutan, registrira SensorEventListener za praćenje promjena na tom senzoru. Ukoliko senzor nije dostupan, postavlja se odgovarajuća poruka u TextView-u.

Na kraju, koristi se ObjectAnimator kako bi se postigla animacija pomicanja slike sunca ili mjeseca s lijeva na desno kao vizualni efekt tijekom izvršavanja aplikacije.

**onSensorChanged():**

Metoda onSensorChanged(SensorEvent event) aktivira se svaki put kad senzor svjetline zabilježi promjenu. Argument event sadrži podatke o promjeni. Unutar ove metode, prvo se iz event izdvaja intenzitet ambijentalne svjetlosti (ambientLight).

Nakon toga, ovisno o intenzitetu svjetlosti, mijenjaju se vizualni elementi aplikacije. Ako je intenzitet manji od 10, postavlja se tamna pozadina (darkBlue), slika mjeseca te odgovarajući pozdravni tekst ("Laku noć"). U suprotnom, postavlja se svijetla pozadina (lightBlue), slika sunca i odgovarajući pozdravni tekst ("Dobar dan").

## PROFILIRANJE APLIKACIJE

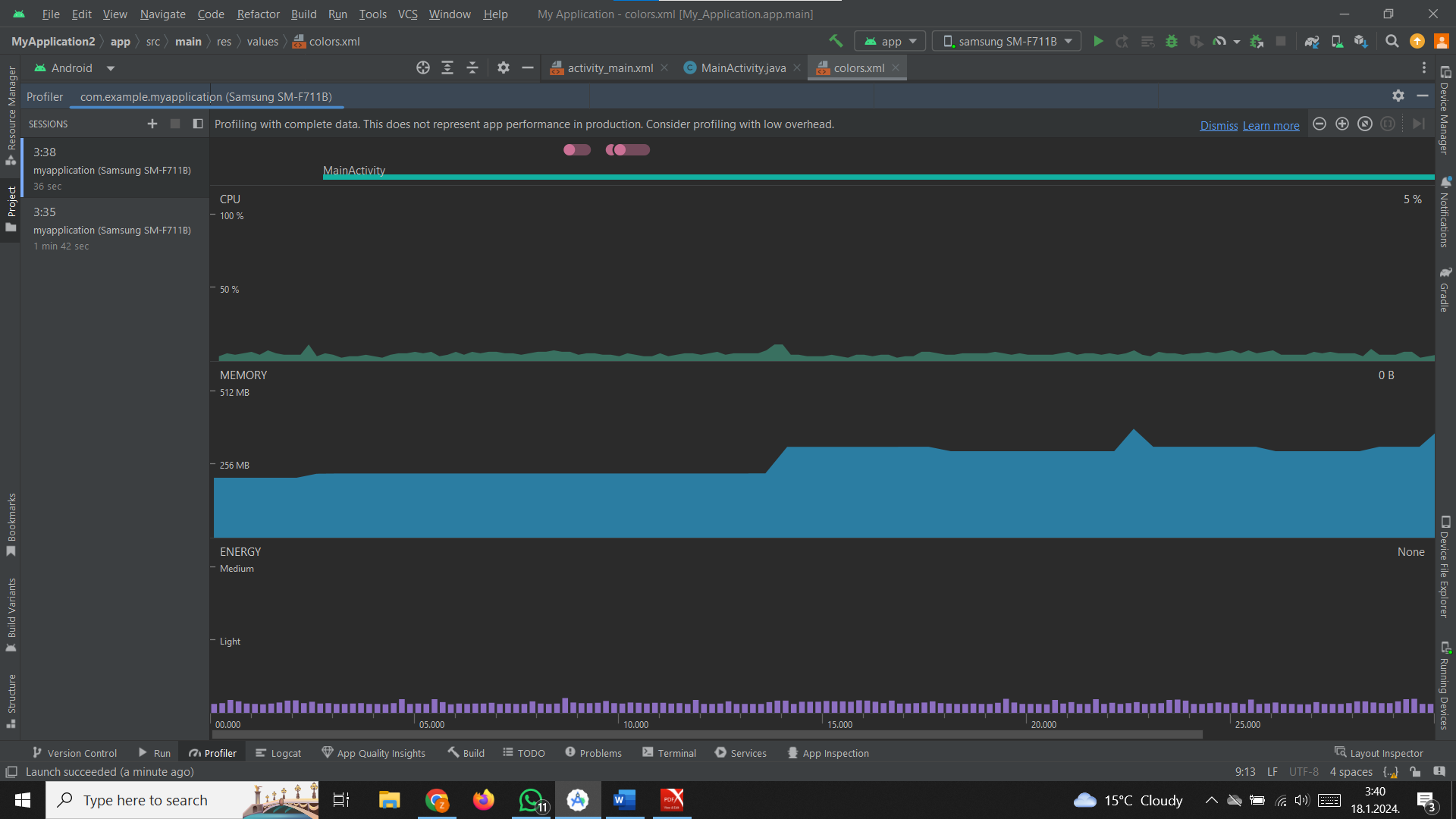
Za profiliranje aplikacije koristila sam alat Android Studia - Profiler, podijeljen na tri ključna dijela: CPU, Memoriju i Energetsku potrošnju.

Na vrhu je prikazana interakcija s korisnikom. Svaka roza točka je pritisak korisnika na ekran.

Onda je CPU. Zelena "brda" prikazuju stvarnu upotrebu CPU-a u stvarnom vremenu, izraženu postotkom ukupnog dostupnog CPU vremena.

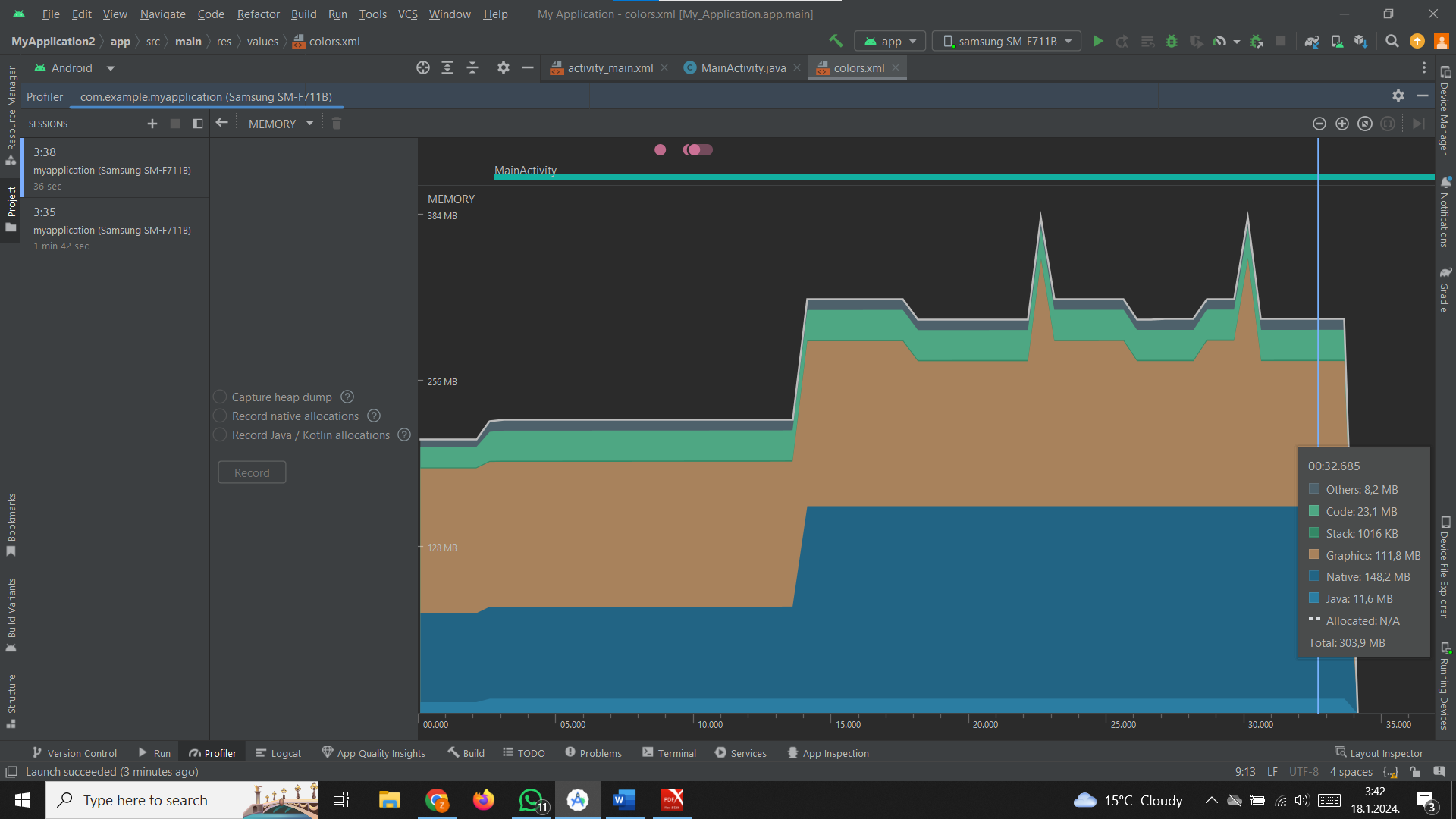
Ispod CPU dijela nalazi se Memory, koji služi za upravljanje memorijom i identificiranje nekorištenih objekata radi poboljšanja iskorištene memorije.

Zadnji segment, Energy Profiler, pomaže u pronalaženju područja gdje aplikacija troši više energije nego što je potrebno.



Odlučila sam izdvojiti graf koji ilustrira korištenje memorije. Značajne promjene na grafu primijetile su se prilikom promjene svjetlosnih uvjeta u prostoriji. Dvije izbočine na grafu dogodile su se kada je svjetlost u prostoriji promijenjena, što je zahtijevalo prilagodbu pozadine, animacije i teksta unutar aplikacije.

Graf jasno pokazuje dinamiku promjene u korištenju memorije tijekom različitih događaja, posebno reagiranje na promjene svjetline. Legenda na slici pruža detaljan pregled dijelova grafa koji su odgovorili na specifične promjene u svjetlini, omogućavajući preciznu interpretaciju utjecaja tih događaja na ponašanje memorije aplikacije.



Ove informacije iz Profilera pružaju sveobuhvatan uvid u performanse aplikacije, omogućujući optimizaciju i poboljšanje korisničkog iskustva

Analizom slika, može se zaključiti da je inicijalno pokretanje aplikacije najviše potrošilo energije, dok se nakon toga održava relativno konstantna potrošnja.