**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**імені ТАРАСА ШЕВЧЕНКА**

****

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра прикладних інформаційних систем**

**Звіт до лабораторної роботи №4**

**з курсу**

**«Архітектура мобільних операційних систем і пристроїв»**

*Студентки 4 курсу*

*групи ПП-42*

*спеціальності 122 «Комп'ютерні науки»*

*ОП «Прикладне програмування»*

Матвіїв А.Ю.

*Викладач:*

Краснощок В.М.

**Київ – 2024**

***Тема:*** Робота із анімацією та макетами в ОС Android.

***Мета:*** Отримати досвід роботи із анімацією та ознайомитись із основними макетами для побудови користувацького інтерфейсу додатків під ОС Android.

***Хід роботи:***

Створіть новий проект laba4 у якій буде робота з анімацією.

В папку drawable добавимо фото до якої будем робити анімацію:



Добавимо код в 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:id="@+id/main"

   android:layout\_width="match\_parent"

   android:layout\_height="match\_parent"

   tools:context=".MainActivity">

   <ImageView

       android:id="@+id/imageView"

       android:layout\_width="340dp"

       android:layout\_height="608dp"

       app:srcCompat="@drawable/fox"

       tools:layout\_editor\_absoluteX="35dp"

       tools:layout\_editor\_absoluteY="61dp" />

</androidx.constraintlayout.widget.ConstraintLayout>

Добавимо файл, який буде описувати анімацію.

ship\_anim.xml:

<?xml version="1.0" encoding="utf-8"?>

<set xmlns:android="http://schemas.android.com/apk/res/android">

   <rotate

       android:duration="3333"

       android:fromDegrees="0"

       android:toDegrees="1080"

       android:pivotX="50%"

       android:pivotY="50%"

       android:repeatCount="infinite"

       android:repeatMode="reverse" />

   <translate

       android:duration="1900"

       android:fromXDelta="-50%p"

       android:toXDelta="50%p"

       android:repeatCount="infinite"

       android:repeatMode="reverse" />

   <translate

       android:duration="1300"

       android:fromYDelta="-50%p"

       android:toYDelta="50%p"

       android:startOffset="123"

       android:repeatCount="infinite"

       android:repeatMode="reverse" />

   <alpha

       android:duration="500"

       android:fromAlpha="1.0"

       android:toAlpha="0.3"

       android:repeatCount="infinite"

       android:repeatMode="reverse" />

   <scale

       android:duration="10000"

       android:fromXScale="0.0"

       android:toXScale="2.5"

       android:fromYScale="0.0"

       android:toYScale="2.5"

       android:pivotX="50%"

       android:pivotY="50%"

       android:repeatCount="infinite"

       android:repeatMode="reverse" />

</set>

MainActivity.kt:

package com.example.laba4

import android.os.Bundle

import android.view.animation.AnimationUtils

import android.widget.ImageView

import androidx.activity.enableEdgeToEdge

import androidx.appcompat.app.AppCompatActivity

import androidx.core.view.ViewCompat

import androidx.core.view.WindowInsetsCompat

class MainActivity : AppCompatActivity() {

   override fun onCreate(savedInstanceState: Bundle?) {

       super.onCreate(savedInstanceState)

*enableEdgeToEdge*()

       setContentView(R.layout.*activity\_main*)

       ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.*main*)) **{** v, insets **->**

val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())

           v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)

           insets

**}**

val shipView: ImageView = findViewById(R.id.*imageView*)

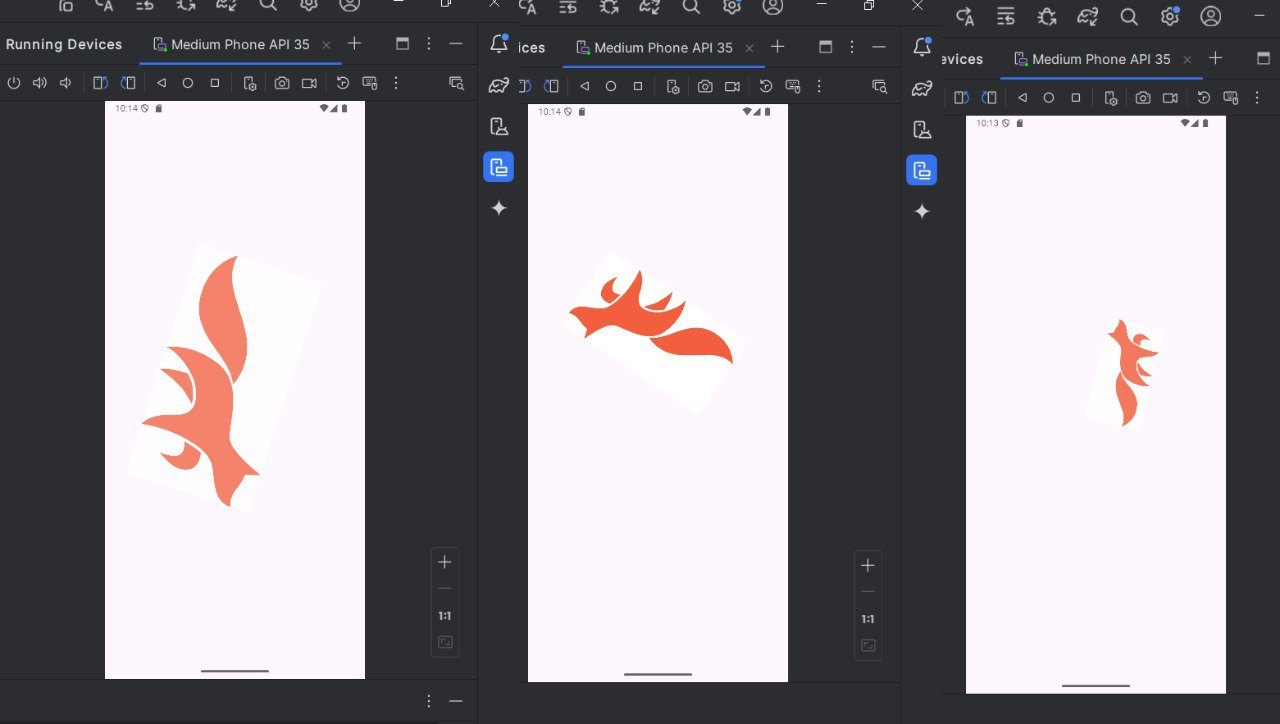
       val shipAnim = AnimationUtils.loadAnimation(this, R.anim.*ship\_anim*)

       shipView.startAnimation(shipAnim)

   }

}

Результат роботи:  
Бачимо, що анімація працює.



Продовжимо написання додатку з попередньої роботи. Добавимо локалізацію для французької мови.

Змінимо activity\_main.xml:  
<LinearLayout

   xmlns:android="http://schemas.android.com/apk/res/android"

   android:layout\_width="match\_parent"

   android:layout\_height="match\_parent"

   android:orientation="vertical"

   android:padding="16dp">

   <TextView

       android:id="@+id/textView"

       android:layout\_width="wrap\_content"

       android:layout\_height="wrap\_content"

       android:layout\_marginBottom="16dp"

       android:text=""

       android:textSize="40sp" />

   <Button

       android:id="@+id/button"

       android:layout\_width="220dp"

       android:layout\_height="120dp"

       android:layout\_marginBottom="16dp"

       android:text="ПП-42"

       android:textSize="24sp" />

   <ImageView

       android:id="@+id/flagImageView"

       android:layout\_width="308dp"

       android:layout\_height="205dp"

       android:layout\_marginBottom="68dp"

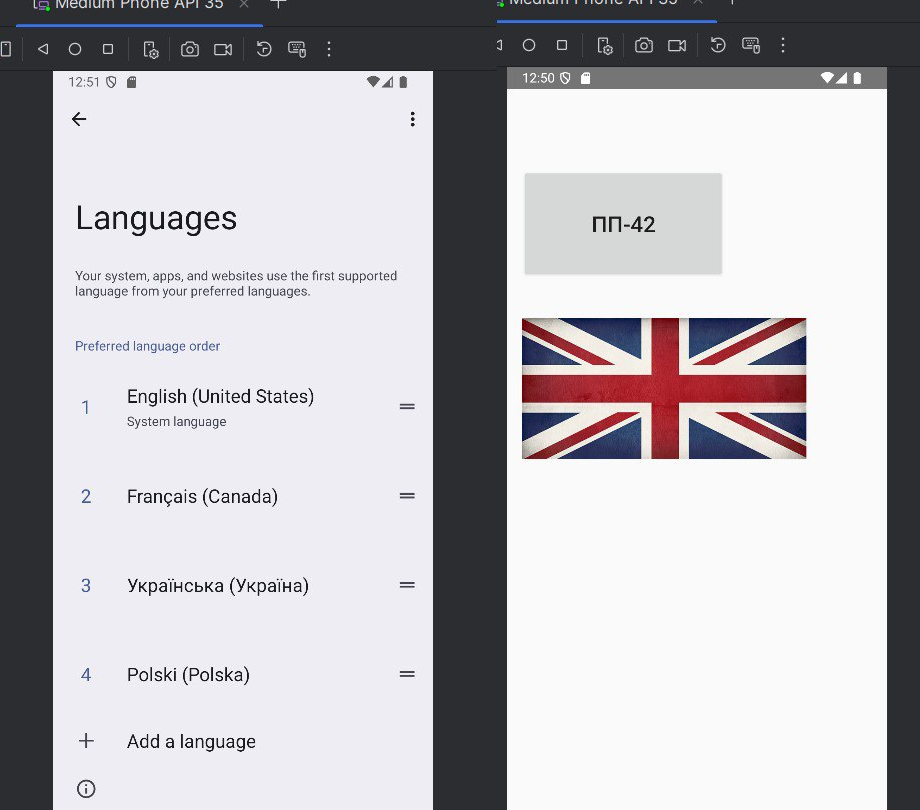
       android:src="@drawable/flag\_default" />

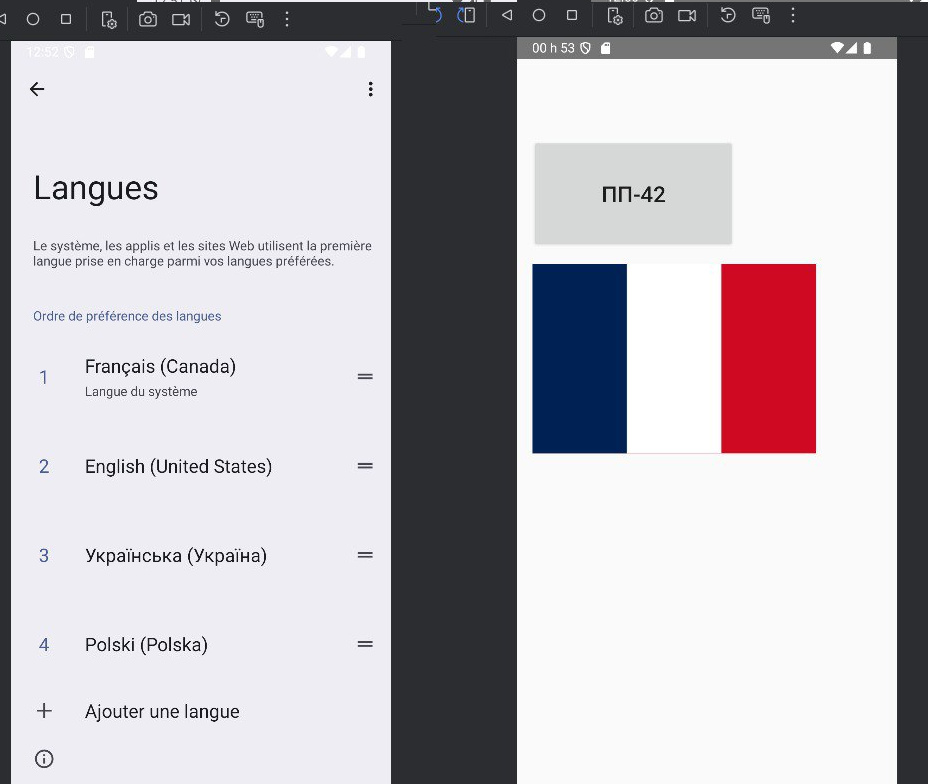
</LinearLayout>

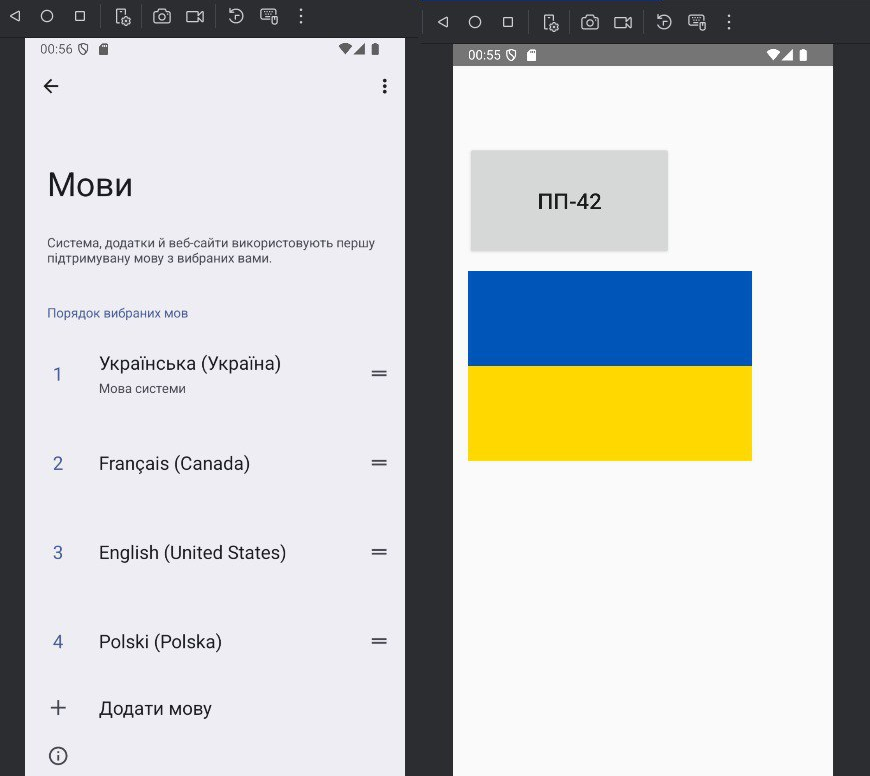
Добавимо новий прапор Франції:



Переглянемо результат:







Створимо проект RelativeLayoutSample:

Добавимо даний код в

MainActivity.kt:

package com.example.relativelayoutsample

import android.os.Bundle

import androidx.activity.enableEdgeToEdge

import androidx.appcompat.app.AppCompatActivity

import androidx.core.view.ViewCompat

import androidx.core.view.WindowInsetsCompat

import android.widget.Button

import android.widget.EditText

import android.widget.TextView

import java.util.Locale

class MainActivity : AppCompatActivity() {

   override fun onCreate(savedInstanceState: Bundle?) {

       super.onCreate(savedInstanceState)

*enableEdgeToEdge*()

       setContentView(R.layout.*activity\_main*)

       // Отримати елементи інтерфейсу

       val textView: TextView = findViewById(R.id.*label*)

       val buttonOk: Button = findViewById(R.id.*ok*)

       val buttonCancel: Button = findViewById(R.id.*cancel*)

       val editText: EditText = findViewById(R.id.*entry*)

       // Визначити мову пристрою

       val currentLocale = Locale.getDefault()

       when (currentLocale.*language*) {

           "uk" -> {

               textView.*text* = getString(R.string.*label\_text*)

               buttonOk.*text* = getString(R.string.*ok*)

               buttonCancel.*text* = getString(R.string.*cancel*)

               editText.*hint* = getString(R.string.*entry\_hint*)

           }

           "fr" -> {

               textView.*text* = "Entrez des données :"

               buttonOk.*text* = "OK"

               buttonCancel.*text* = "Annuler"

               editText.*hint* = "quelque chose..."

           }

           else -> {

               textView.*text* = getString(R.string.*label\_text*)

               buttonOk.*text* = getString(R.string.*ok*)

               buttonCancel.*text* = getString(R.string.*cancel*)

               editText.*hint* = getString(R.string.*entry\_hint*)

           }

       }

   }

}

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:id="@+id/main"

   android:layout\_width="match\_parent"

   android:layout\_height="match\_parent"

   tools:context=".MainActivity">

   <!-- Label TextView -->

   <TextView

       android:id="@+id/label"

       android:layout\_width="0dp"

       android:layout\_height="wrap\_content"

       android:text="@string/label\_text"

       app:layout\_constraintTop\_toTopOf="parent"

       app:layout\_constraintStart\_toStartOf="parent"

       app:layout\_constraintEnd\_toEndOf="parent"

       android:layout\_marginTop="16dp" />

   <!-- Entry EditText -->

   <EditText

       android:id="@+id/entry"

       android:layout\_width="0dp"

       android:layout\_height="wrap\_content"

       android:hint="@string/entry\_hint"

       android:background="@android:drawable/edit\_text"

       app:layout\_constraintTop\_toBottomOf="@id/label"

       app:layout\_constraintStart\_toStartOf="parent"

       app:layout\_constraintEnd\_toEndOf="parent"

       android:layout\_marginTop="8dp" />

   <!-- OK Button -->

   <Button

       android:id="@+id/ok"

       android:layout\_width="wrap\_content"

       android:layout\_height="wrap\_content"

       android:text="@android:string/ok"

       app:layout\_constraintTop\_toBottomOf="@id/entry"

       app:layout\_constraintEnd\_toEndOf="parent"

       android:layout\_marginTop="8dp" />

   <!-- Cancel Button -->

   <Button

       android:id="@+id/cancel"

       android:layout\_width="wrap\_content"

       android:layout\_height="wrap\_content"

       android:text="@android:string/cancel"

       app:layout\_constraintTop\_toTopOf="@id/ok"

       app:layout\_constraintEnd\_toStartOf="@id/ok"

       android:layout\_marginEnd="8dp" />

</androidx.constraintlayout.widget.ConstraintLayout>

string.xml:

<resources>

   <string name="app\_name">RelativeLayoutSample</string>

   <string name="label\_text">Введіть дані:</string>

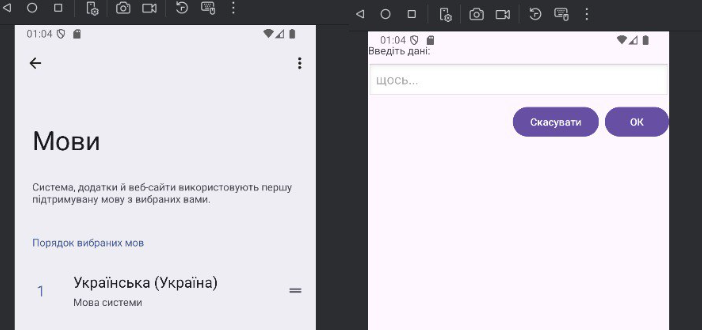
   <string name="entry\_hint">щось...</string>

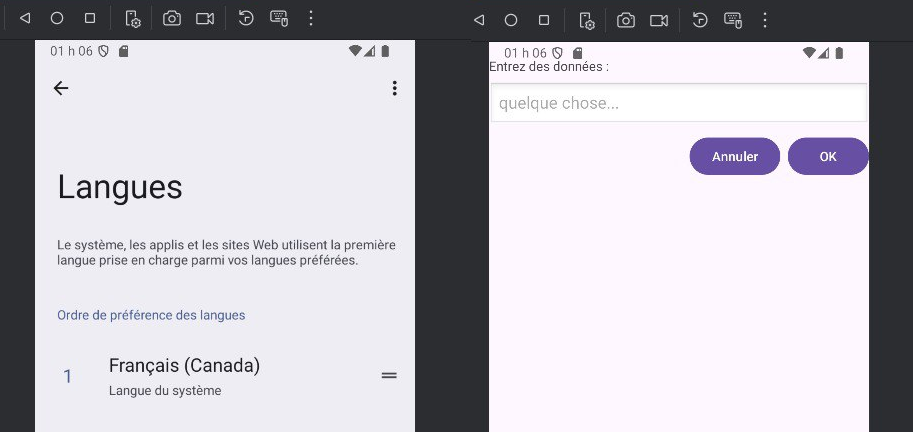
   <string name="cancel">Скасувати</string>

   <string name="ok">ОК</string>

</resources>

Результат роботи. Є дві версії: українська мова і французька.





***Висновок:*** На даній лабораторній роботі я отримала навички роботи з анімацією та макетами в Android Studio. Зокрема, я навчилася створювати динамічні елементи інтерфейсу та адаптувати додаток під різні мови.

***Контрольні запитання:***

1. ***Наскільки добре платформа Android справляється з анімацією?***

Android має потужні інструменти для створення різноманітних анімацій. Платформа пропонує широкий спектр можливостей для анімування властивостей об'єктів, перехідних ефектів між екранами та створення складних анімаційних послідовностей. З кожною новою версією Android інструменти для анімації стають все більш досконалими та легкими у використанні.

1. ***Чи існує спосіб зупинки відтворення анімації?***

Так, існує кілька способів зупинити відтворення анімації в Android. Один з найпоширеніших – це використання методу cancel() для об'єкта Animator. Також можна змінити значення властивостей об'єкта, які анімуються, щоб зупинити анімацію в потрібному місці.

1. ***Які типи операцій підтримуються анімаціями руху?***

Анімації руху в Android дозволяють змінювати різні властивості об'єктів, такі як:

* Переміщення об'єкта в просторі.
* Зміна ширини та висоти об'єкта.
* Зміна прозорості об'єкта.
* Зміна кута повороту об'єкта.
* Зміна розміру об'єкта відносно центру.
* Зміна кольору об'єкта.

1. ***Чи може бути використаний елемент-контейнер LinearLayout для розміщення всіх дочірніх елементів-уявлень View один за одним (по вертикалі)?***

Так, LinearLayout є одним з найпоширеніших виджетів для вертикального розміщення елементів. Він дозволяє розміщувати дочірні елементи один під одним або один поруч. Для вертикального розміщення необхідно встановити атрибут android:orientation="vertical".

1. ***Назвіть віджети, які використовуються для створення користувацького інтерфейсу.***

Android пропонує широкий спектр віджетів для створення користувацького інтерфейсу. Деякі з найпоширеніших:

* TextView: Для відображення тексту.
* Button: Для створення кнопок.
* ImageView: Для відображення зображень.
* EditText: Для введення тексту користувачем.
* CheckBox: Для створення прапорців.
* RadioButton: Для створення радіокнопок.
* Spinner: Для створення спадного списку.
* ListView: Для створення списків.
* GridView: Для створення сіток.
* RecyclerView: Більш ефективний варіант ListView для великих списків.