

Practice 12. Solunin

Создаем cell-анимацию

Добавляем кадры нашей анимации, и создаем xml файл с названием нашей анимации и добавляем туда значения

```
<?xml version="1.0" encoding="utf-8"?>
<animation-list xmlns:android="http://schemas.android.com/apk/res/android"
    android:oneshot="false">
    <item android:drawable="@drawable/first" android:duration="100" />
    <item android:drawable="@drawable/second" android:duration="100" />
    <item android:drawable="@drawable/third" android:duration="100" />
    <item android:drawable="@drawable/four" android:duration="100" />
</animation-list>
```

MainActivity

```
package com.example.myapplication13

import android.graphics.drawable.AnimationDrawable
import android.os.Bundle
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 img : ImageView = findViewById(R.id.ivAnim)
        // устанавливаем ресурс анимации
        img.setBackgroundResource(R.drawable.cat_animation);
        // получаем объект анимации
        val frameAnimation = img.background as AnimationDrawable
        frameAnimation.start()
    }
}
```

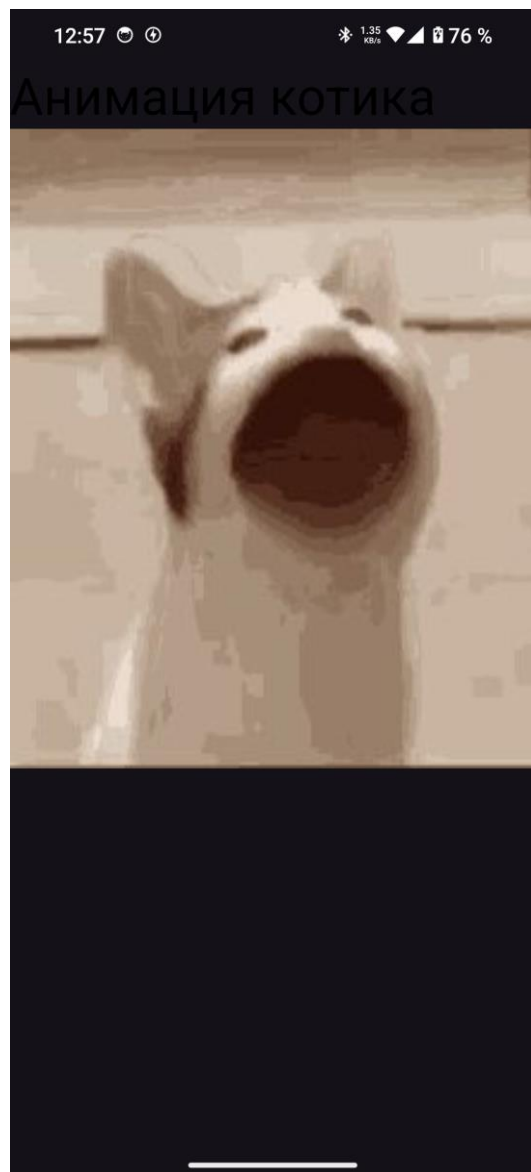
Activity_main

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical">
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Анимация котика"
    android:textColor="@color/black"
    android:textSize="40sp"
/>
<ImageView
    android:id="@+id/ivAnim"
    android:layout_width="match_parent"
    android:layout_height="500dp"
    android:layout_gravity="center"
/>

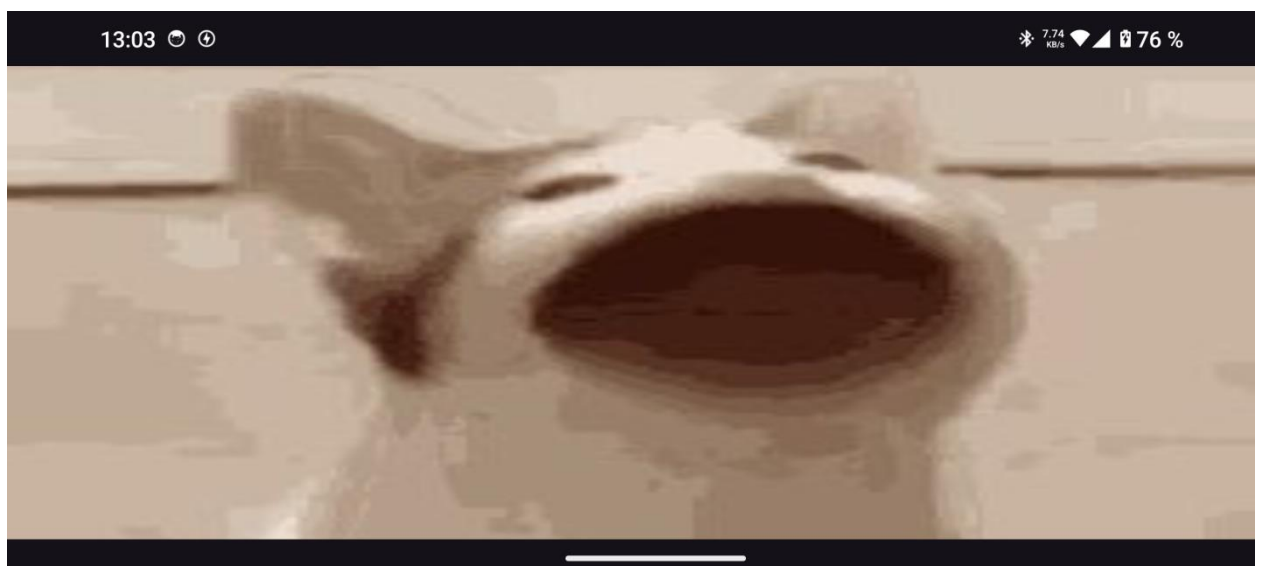
</LinearLayout>
```

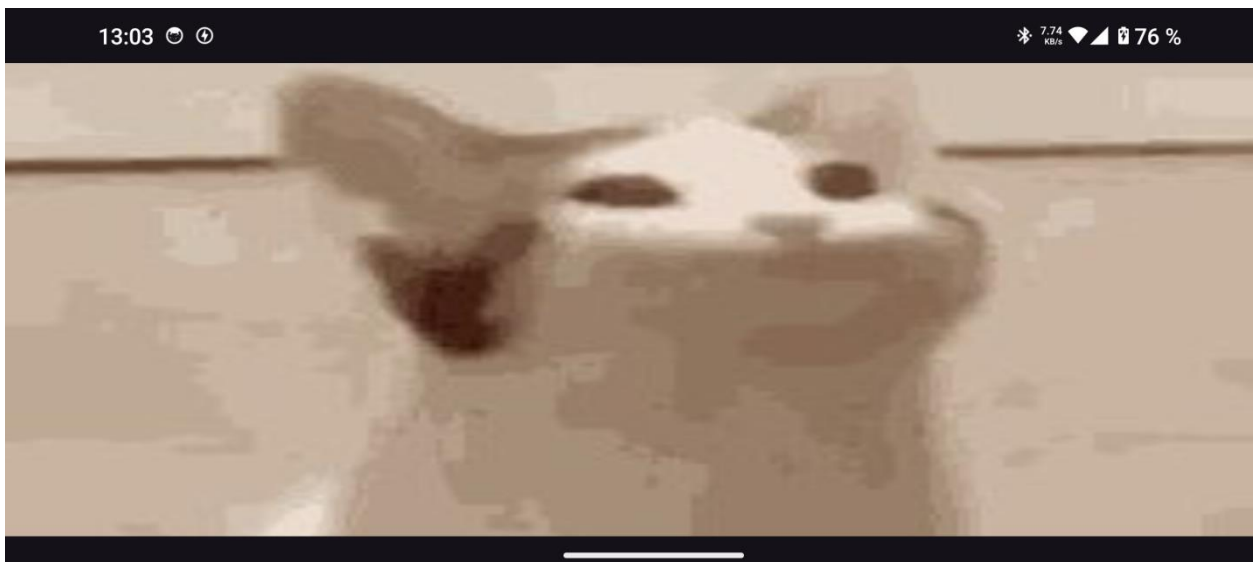
Result





Этаже анимация но теперь горизонтально





Дальше делаем twin анимацию. Создаем папку anim и так создаем xml файл в который прописываем

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/linear_interpolator">
    <scale android:fromXScale="1.0" android:toXScale="0.5"
        android:fromYScale="1.0" android:toYScale="0.5"
        android:pivotX="50%" android:pivotY="50%" android:duration="4500"
        android:repeatCount="infinite" android:repeatMode="reverse" />
    <rotate
        android:fromDegrees="0.0"
        android:toDegrees="180.0"
        android:pivotX="50%"
        android:pivotY="50%" />
    <alpha android:fromAlpha="1.0" android:toAlpha="0.1"
        android:duration="1000"
        android:repeatCount="infinite" android:repeatMode="reverse" />
    <translate android:fromXDelta="0.0"
        android:toXDelta="25.0"
        android:fromYDelta="20.0"
        android:toYDelta="80.0"
        android:duration="1000"
        android:repeatMode="reverse"
        android:repeatCount="infinite" />
</set>
```

Используя предложенное фото, и меняем наши файлы

MainActivity

```
package com.example.myapplication13

import android.graphics.drawable.AnimationDrawable
import android.os.Bundle
import android.view.animation.Animation
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 imgCat : ImageView = findViewById(R.id.ivAnimCat)

        val anim : Animation =
            AnimationUtils.loadAnimation(this,R.anim.new_animation)
        imgCat.startAnimation(anim)

    }
}

```

activity_main

```

<?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/ivAnimCat"
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:src="@drawable/cat"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

Result twin animation

13:09



0.99 KB/s



77 %



13:09



0.00 KB/s



77 %

