

Создание анимированных элементов на экране в Android Studio

Андрюченко Иван Сергеевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье описывается процесс создания анимации элементов на экране в приложении android на языке программирования Java. Для создания анимации использовались среда разработки Android Studio и изображение звезды из стандартных рисунков android. В результате было создано приложение, в котором, при нажатии на кнопку, запускалась анимация движения звезды.

Ключевые слова: Android-разработка, анимация, пользовательский интерфейс, мобильное приложение, Android Studio.

Creating animated elements on the screen in Android Studio

Andrienko Ivan Sergeevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article describes the process of creating animation of elements on the screen in an android application in the Java programming language. To create the animation, the Android Studio development environment and a star image from standard android drawings were used. As a result, an application was created in which, when a button was pressed, an animation of the movement of a star was launched.

Keywords: Android-development, animation, user interface, mobile application, Android Studio.

1 Введение

1.1 Актуальность

Сегодня все современные приложения имеют анимированные объекты. Они помогают ориентироваться в функционале приложения, акцентируют внимание на нужных деталях и помогают узнаваемости бренда. Также анимация придает приложению более качественный внешний вид. Android включает в себя различные API анимации в зависимости от того, какой тип анимации хочет добавить разработчик в свой пользовательский интерфейс (UI).

1.2 Обзор исследований

В своей работе С.В. Ярош выявил проблемы, возникающие при создании анимированных интерфейсов. выявил критерии для дальнейшего проведения сравнительного анализа программных средств для создания анимированных интерфейсов [1]. Д.Д. Федорова описала как анимационные решения элементов информационной среды одновременно используют несколько типов анимации по степени интерактивности. [2]. В своей работе Д. Саффер описывает основные элементы анимации, а затем показывает, как использовать их в мобильном приложении, веб-виджете и устройстве. [3]. В своей работе В. Хэд написал ускоренный курс по теории и практике motion design для веб-дизайнеров, специалистов по UX и интерфейсных разработчиков. [4]

1.3 Цель исследования

Цель исследования – создать приложение, проигрывающее анимацию рисунка на экране.

2 Материалы и методы

Процесс создания мобильного приложения проделан в среде Android Studio, с использованием языка программирования Java.

3 Результаты и обсуждения

В Android Studio существует множество анимаций. Для простоты их можно поделить на 3 вида: базовые, комплексные и «низкоуровневые» анимации. В данной работе будет продемонстрирован пример применения базовой анимации, с помощью пакета animation.

Перед созданием анимации необходимо создать новый проект используя шаблон «Empty activity». Минимальное SDK необходимо выбрать от API 11, так как ниже пакет не предоставляется. После создания проекта, в activity_main.xml необходимо добавить элемент, который будет анимирован. Это может быть текст, изображение, кнопка, виджет и т.п. В данной работе будет добавлено множество анимаций изображению звезды. С помощью вкладки «Design» перетаскиваем ImageView на экран и выбираем любое изображение. Так же добавим кнопку (рис. 1). С помощью нее анимация будет проигрываться по нажатию.

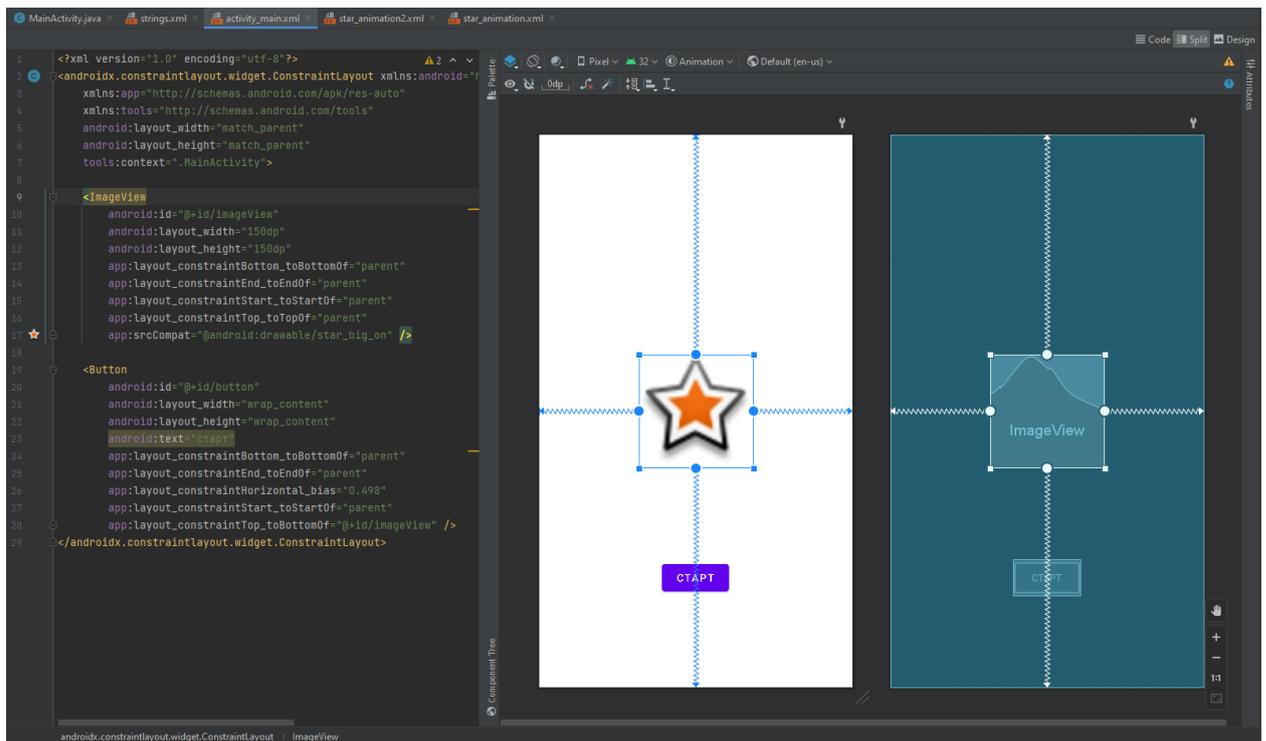


Рисунок 1 – Добавление элементов на экран

Теперь необходимо создать папку, где будут находиться анимации в виде кода. В папке app находится каталог для ресурсов – res. В нем создаем android resource directory и выбираем тип ресурсов anim (рис. 2).

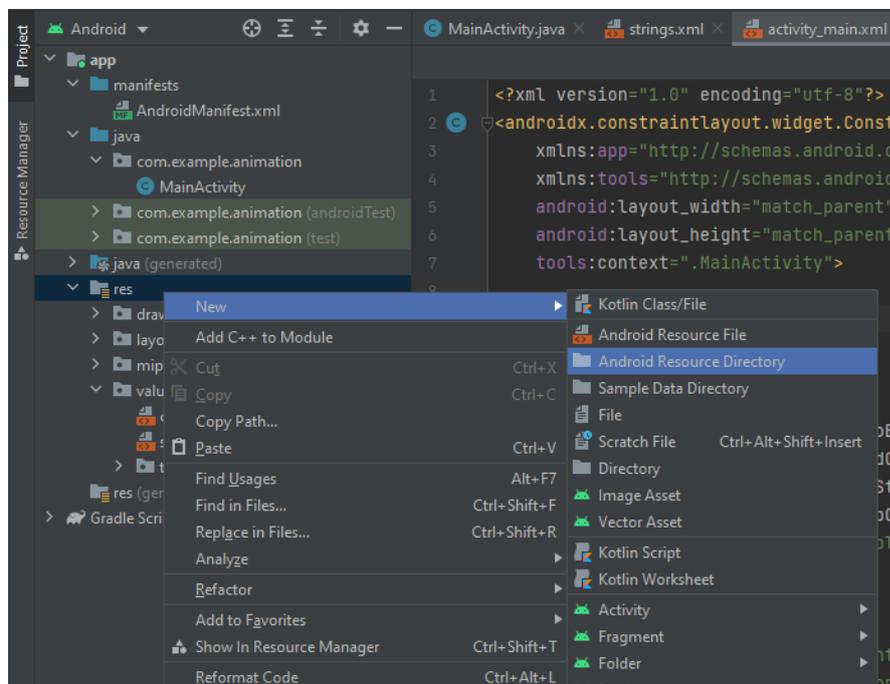


Рисунок 2 – Создание папки для файлов анимации

В созданной папке будут содержаться xml файлы с кодом анимации. Создаем такой файл, а именно animation resource fail. Даем ему название star_animation (рис.3).

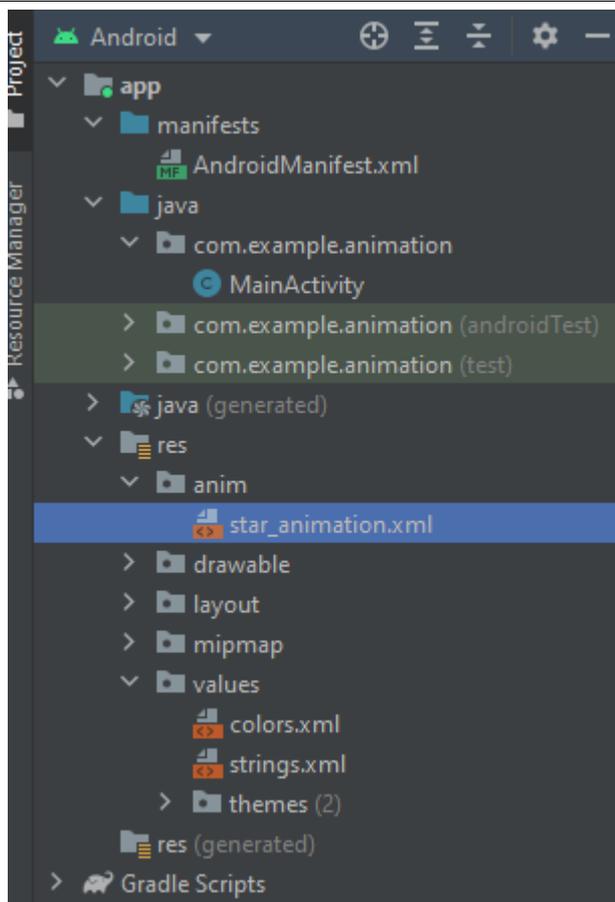


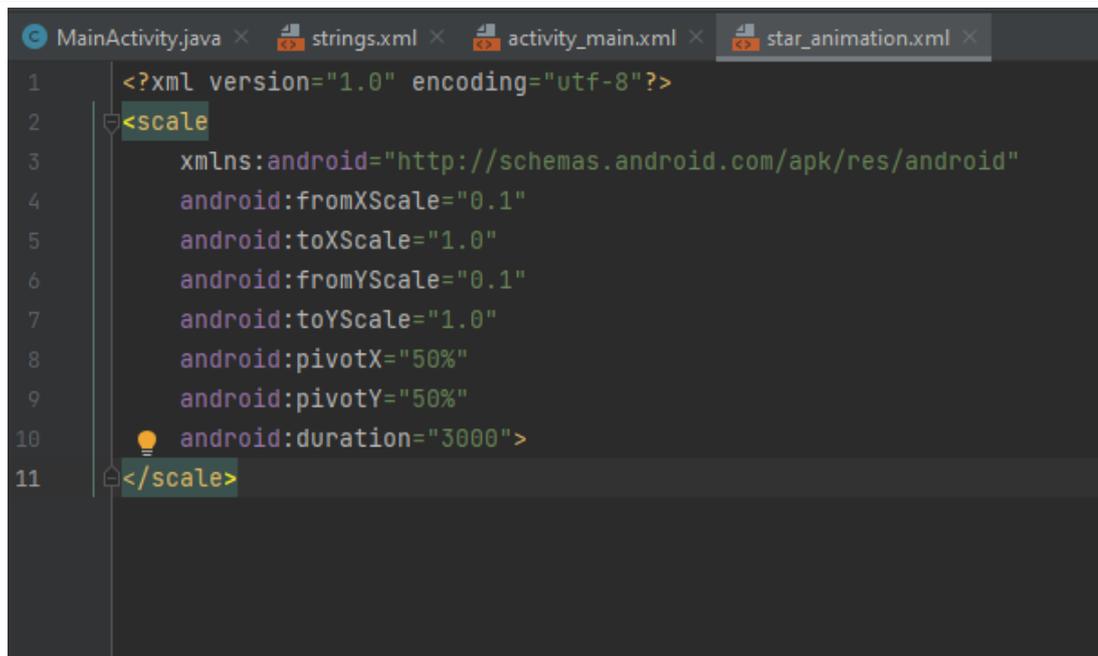
Рисунок 3 – Создание файла Animation resource

В созданном файле зададим анимацию. Это может быть элемент `alpha` – создающий эффект плавного изменения прозрачности, `scale` – меняющий размер объекта, `translate` – создающий эффект движения объекта, `rotate` – предназначенный для анимации вращения и элемент `set` – позволяющий комбинировать перечисленные анимации.

После выбора элемента необходимо задать атрибуты анимации. У каждого элемента свои собственные атрибуты, это могут быть координаты смещения объекта, угол вращения, значение прозрачности и т.п. У всех элементов есть один и тот же атрибут – `duration` – продолжительность анимации в миллисекундах.

При создании анимации важно понимать, что анимация может перемещаться или изменять размер, при этом границы объекта, в котором хранится анимация, не будут автоматически подстраиваться под нее. Несмотря на это, анимация все равно будет нарисована за пределами ее видимости и не будет обрезана.

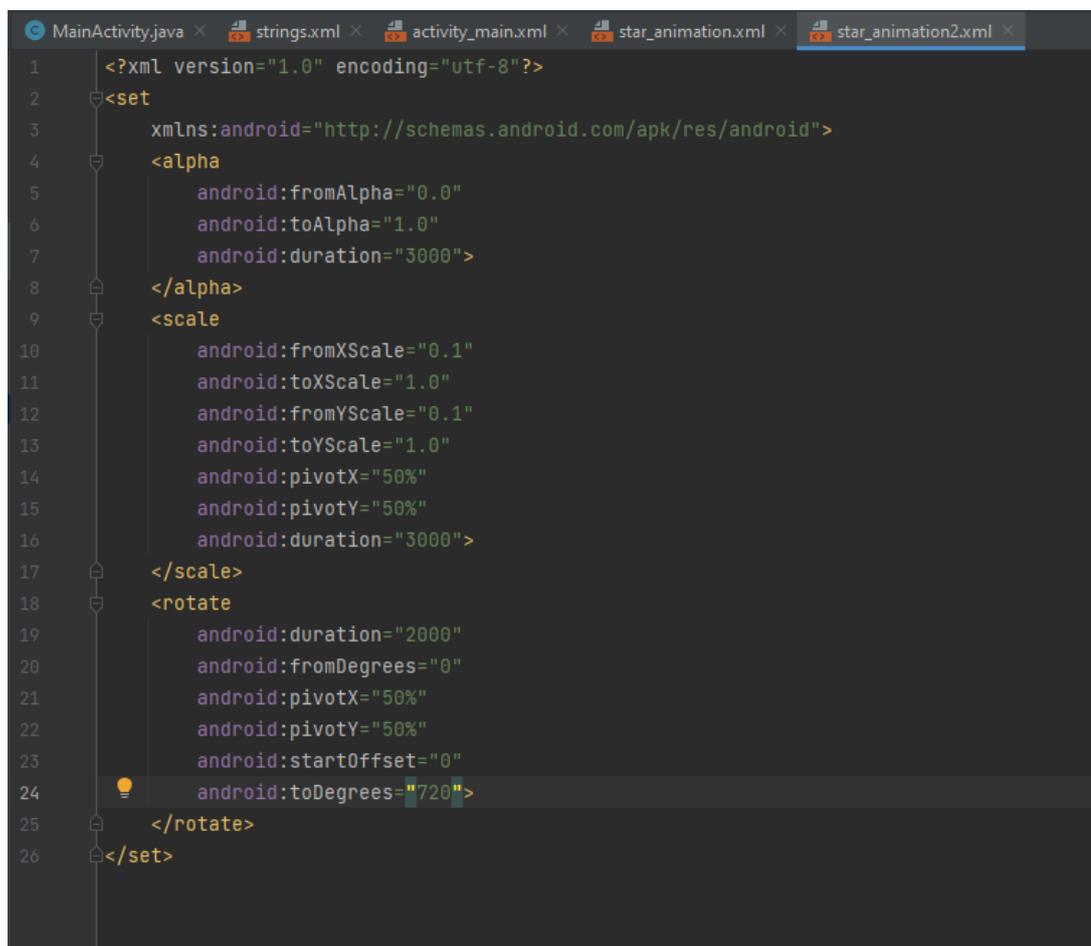
Создадим анимацию плавного увеличения звезды (рис. 4).



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <scale
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:fromXScale="0.1"
5     android:toXScale="1.0"
6     android:fromYScale="0.1"
7     android:toYScale="1.0"
8     android:pivotX="50%"
9     android:pivotY="50%"
10    android:duration="3000">
11 </scale>
```

Рисунок 4 – Создание анимации изменения размера объекта

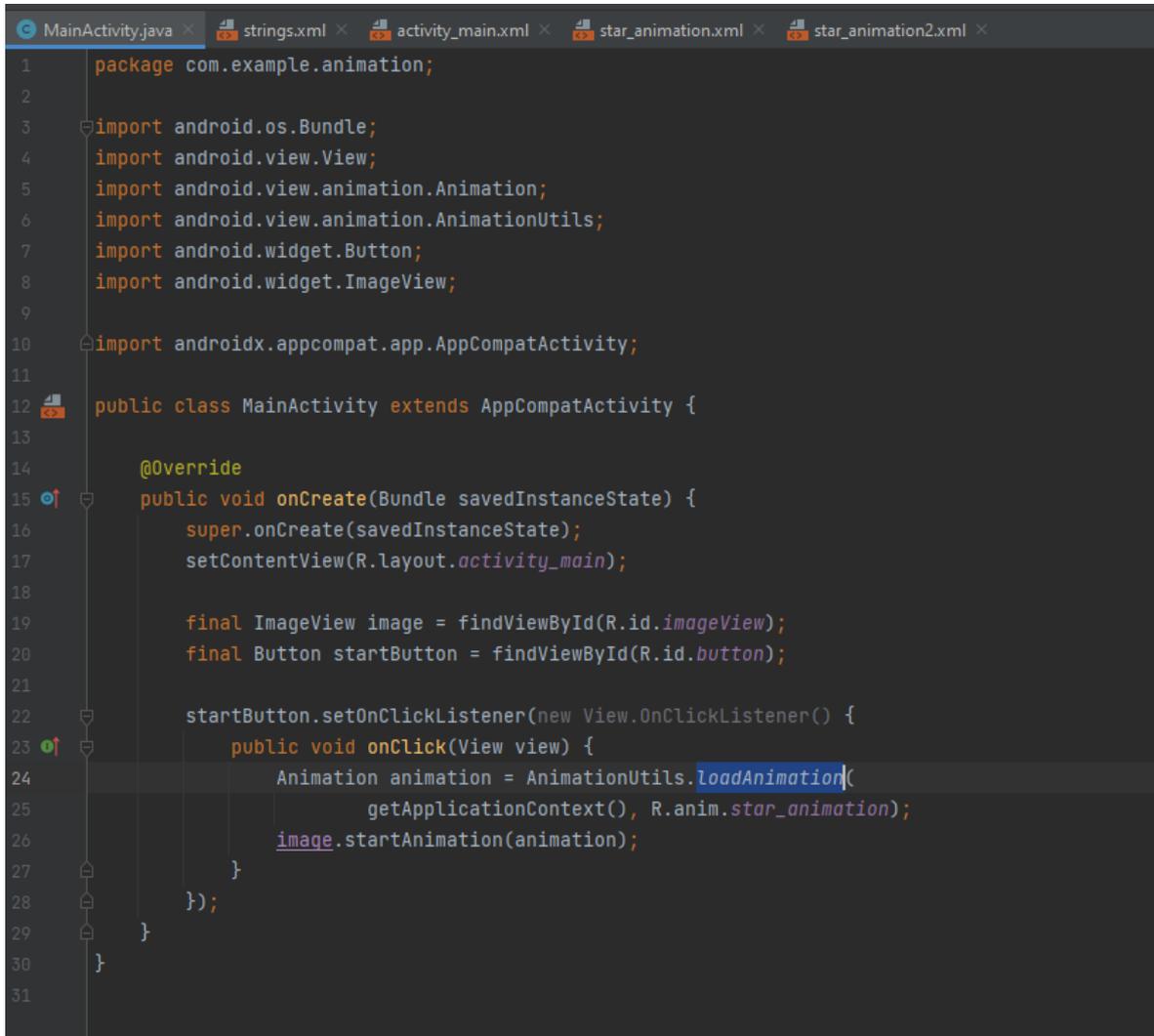
Дополнительно создадим еще один файл `star_animation2.xml` и пропишем комбинированную анимацию, создающую эффект появления, увеличения и прокручивания звезды вокруг своей оси дважды (рис. 5).



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <set
3     xmlns:android="http://schemas.android.com/apk/res/android">
4     <alpha
5         android:fromAlpha="0.0"
6         android:toAlpha="1.0"
7         android:duration="3000">
8     </alpha>
9     <scale
10        android:fromXScale="0.1"
11        android:toXScale="1.0"
12        android:fromYScale="0.1"
13        android:toYScale="1.0"
14        android:pivotX="50%"
15        android:pivotY="50%"
16        android:duration="3000">
17    </scale>
18    <rotate
19        android:duration="2000"
20        android:fromDegrees="0"
21        android:pivotX="50%"
22        android:pivotY="50%"
23        android:startOffset="0"
24        android:toDegrees="720">
25    </rotate>
26 </set>
```

Рисунок 5 – Создание комбинированной анимации

Пропишем код для работы программы. Импортируем необходимые классы. Далее находим изображение и кнопку на экране с помощью `findViewById`. Так как анимация будет воспроизводиться по кнопке добавим атрибут `OnClick`. Загружаем анимацию в `animation` используя `loadAnimation` и передаем элементу `image` (рис. 6).



```
1 package com.example.animation;
2
3 import android.os.Bundle;
4 import android.view.View;
5 import android.view.animation.Animation;
6 import android.view.animation.AnimationUtils;
7 import android.widget.Button;
8 import android.widget.ImageView;
9
10 import androidx.appcompat.app.AppCompatActivity;
11
12 public class MainActivity extends AppCompatActivity {
13
14     @Override
15     public void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18
19         final ImageView image = findViewById(R.id.imageView);
20         final Button startButton = findViewById(R.id.button);
21
22         startButton.setOnClickListener(new View.OnClickListener() {
23             public void onClick(View view) {
24                 Animation animation = AnimationUtils.loadAnimation(
25                     getApplicationContext(), R.anim.star_animation);
26                 image.startAnimation(animation);
27             }
28         });
29     }
30 }
31
```

Рисунок 6 – Код программы

Включаем эмулятор и запускаем приложение (рис. 7).

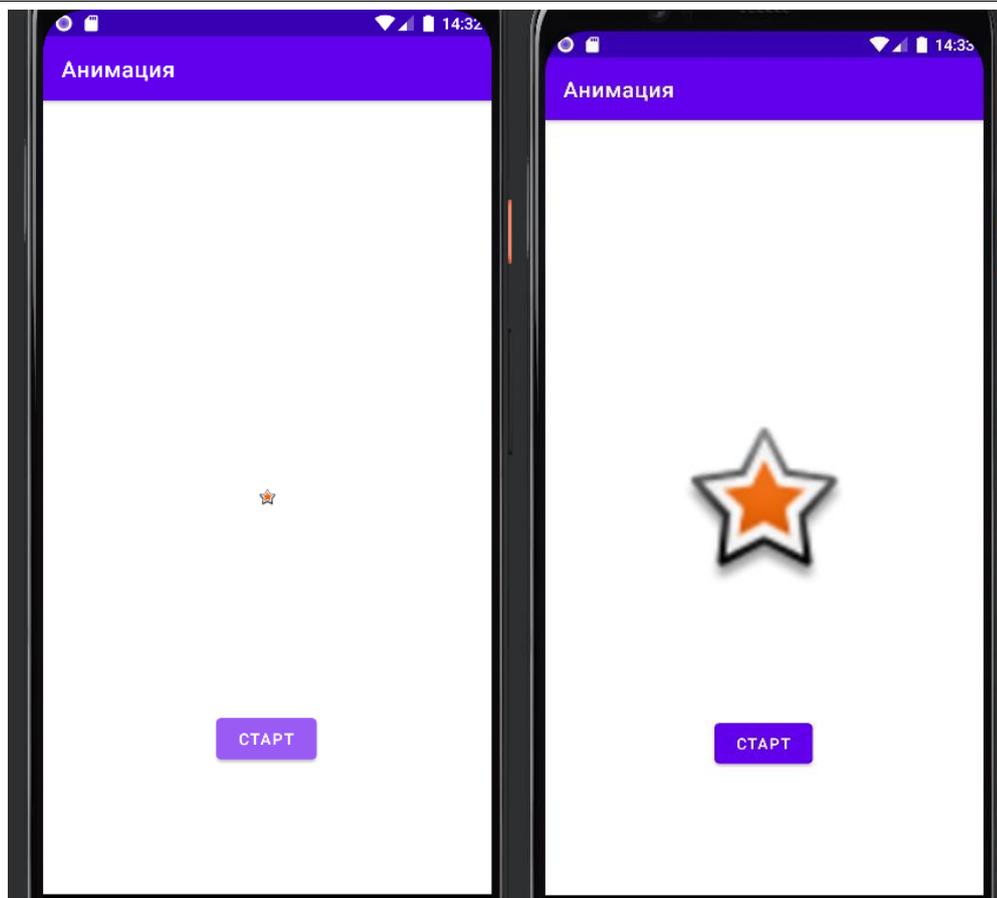


Рисунок 7 – Запуск первой анимации

Так же запустим комбинированную анимацию, предварительно изменив в загрузке анимации путь на `R.anim.star_animation2` (рис.8).

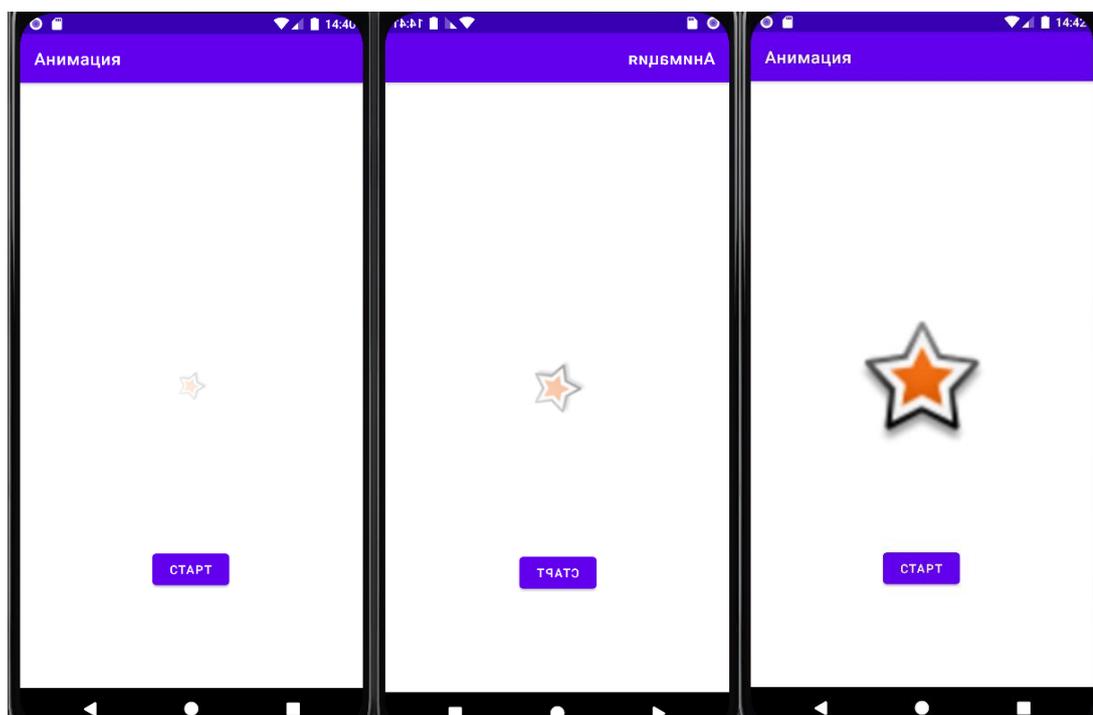


Рисунок 8 – Запуск комбинированной анимации

Выводы

В данной работе был описан процесс создания анимации элементов на экране в приложении android на языке программирования Java. В результате было создано приложение, в котором, при нажатии на кнопку, запускалась анимация движения звезды.

Библиографический список

1. Ярош С.В. Исследование проблем создания анимированных интерфейсов. // Альманах научных работ молодых ученых университета ИТМО. XLVII научная и учебно-методическая конференция Университета ИТМО по тематикам: экономика; менеджмент, инноватика. 2018. С. 330-332.
2. Федорова Д.Д. Роль и виды интерактивной анимации в обучающих приложениях // Новые информационные технологии в науке. Сборник статей Международной научно-практической конференции. 2018. С. 168-170.
3. Saffer D. Microinteractions: Full Color Edition: Designing with Details. Massachusetts: O'Reilly Media, 2013. 170 с.
4. Head V. Designing Interface Animation. Rosenfeld Media, 2016. 240 с.
5. Климов А. Android: Анимация преобразований. URL: <https://developer.alexanderklimov.ru/android/animation/tweenanimation.php>
6. Introduction to animations | Android Developers URL: <https://developer.android.com/develop/ui/views/animations/overview>