|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА — Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт информационных технологий

Кафедра математического обеспечения и стандартизации информационных технологий

**Отчет по практической работе №11**

по дисциплине

«Разработка мобильных приложений»

|  |  |
| --- | --- |
| **Выполнил:**  Студент группыИКБО-33-21 | Привезенцев Д.А. |
| **Проверил:**  Ассистент | Новичков Д.Е. |

Москва 2023 г.

**СОДЕРЖАНИЕ**

[1. ЗАДАНИЕ 1 3](#_Toc136818319)

[1.1 Условие задания 3](#_Toc136818320)

[1.2 Решение задания 3](#_Toc136818321)

[2. ЗАДАНИЕ 2 6](#_Toc136818322)

[2.1 Условие задания 6](#_Toc136818325)

[2.2 Решение задания 6](#_Toc136818326)

[ВЫВОД 7](#_Toc136818327)

# ЗАДАНИЕ 1

## Условие задания

В ранее реализованном приложении для каждого фрагмента создать набор анимация для обработки его появления и исчезновения.

## Решение задания

Для добавления анимаций появления и исчезновения для каждого фрагмента в приложении были внесены следующие изменения. В коде был добавлен слушатель кликов на кнопку button2. При клике на кнопку происходит навигация к фрагменту thirdFragment с передачей дополнительных данных bundle2. Далее, для фрагмента была получена ссылка на ImageView с идентификатором image. В этом ImageView был установлен фоновый ресурс анимированного вектора, используя метод setBackgroundResource. Затем, для получения объекта AnimatedVectorDrawable из фонового ресурса, был вызван метод getBackground на ImageView. Для запуска анимации был добавлен обработчик кликов на ImageView с использованием лямбда-выражения. При клике на ImageView вызывается метод start у объекта AnimatedVectorDrawable, запускающий анимацию. Код представлен на Рисунке 1.

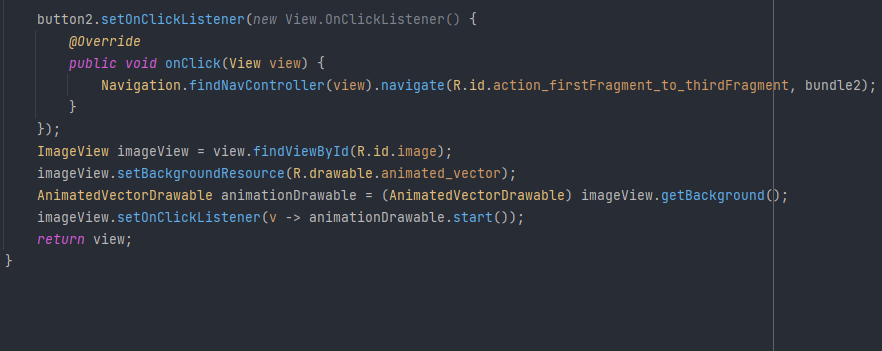


Рисунок 1 — Код навигации в модуле FirstFragment

Этот код обеспечивает анимированное появление и исчезновение фрагмента при навигации, создавая привлекательный и интерактивный пользовательский интерфейс. Анимации могут быть настроены и изменены для каждого фрагмента в соответствии с требованиями и дизайном приложения. Сама же анимация представлена на Рисунках 2-3.

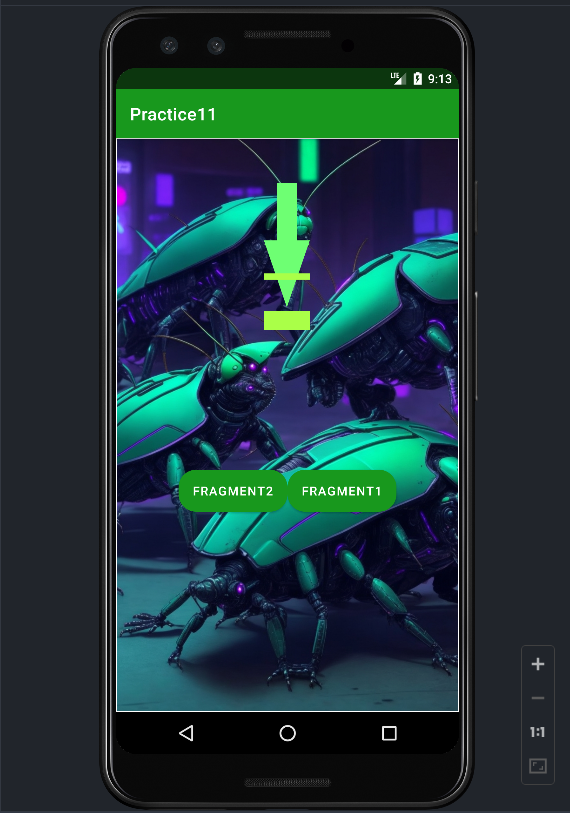


Рисунок 2 — Анимация в начальной точке

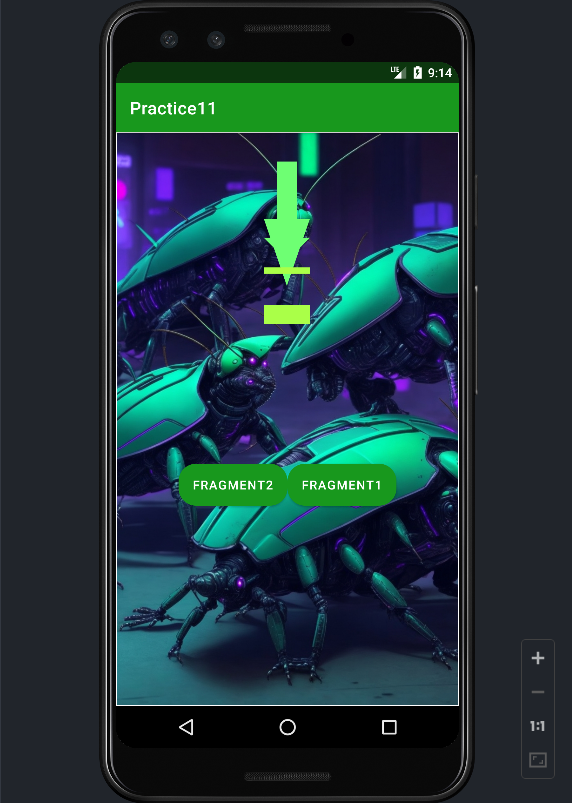


Рисунок 3 — Анимация в конечной точке

# ЗАДАНИЕ 2



## Условие задания

Добавить в проект ресурс формата векторной анимации и внедрить его в своё приложение.

## Решение задания

Для добавления векторной анимации в приложение был использован ресурс формата векторной анимации animated\_vector.xml. Данный ресурс был размещен в директории res/drawable проекта. В коде XML разметки был добавлен элемент ImageView, с идентификатором image, который использует векторную анимацию в качестве источника изображения (app:srcCompat="@drawable/animated\_vector"). Элемент ImageView имеет заданные размеры layout\_width и layout\_height и располагается внутри ConstraintLayout, используя ограничения ConstraintLayout. Таким образом, после добавления векторной анимации в проект и внедрения ее в элемент ImageView, при запуске приложения данная анимация будет отображаться на экране в соответствии с заданными параметрами. Файловая структура с добавленным animated\_vector.xml изображена на Рисунке 4.

В результате выполнения данного задания была успешно добавлена векторная анимация в приложение. Это позволяет создать более интерактивный и привлекательный пользовательский интерфейс, обогатив его анимацией.

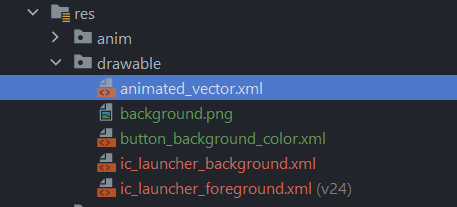


Рисунок 4 — файловая система с animated\_vector.xml

# ВЫВОД

В первом задании мы реализовали набор анимаций для каждого фрагмента в нашем приложении. Для этого мы создали различные анимационные файлы, определили анимации появления и исчезновения для каждого фрагмента, и применили их в соответствующих фрагментах. Это позволило нам добавить эффекты перехода между фрагментами и улучшить пользовательский опыт. Во втором задании мы добавили векторную анимацию в наше приложение. Для этого мы создали ресурс формата векторной анимации и внедрили его в элемент ImageView в нашей разметке. Это позволило нам отобразить векторную анимацию на экране и добавить интерактивность и привлекательность в наше приложение. В результате работы мы создали более привлекательный и пользовательски удобный интерфейс, что может положительно сказаться на восприятии и впечатлении пользователей от нашего приложения.