

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ДЛЯ РОЗПІЗНАВАННЯ ОБРАЗІВ НА ОСНОВІ НЕЙРОМЕРЕЖІ. РОЗРОБКА КОРИСТУВАЦЬКОГО ІНТЕРФЕЙСУ.

Вінницький національний технічний університет

Анотація

В роботі досліджено ряд макетів для створення дружнього користувача інтерфейсу на платформі Android з використанням сучасної мови програмування Kotlin. За результатами аналізу обрано оптимальний макет для вирішення поставленої задачі.

Ключові слова: Android, kotlin, інтерфейс, макет, layout.

Abstract

This paper examines a number of layouts to create a user-friendly UI on the Android platform using the modern Kotlin programming language. According to the results of the analysis, the optimal solution for the task was selected.

Keywords: Android, kotlin, layout, interface.

З появою нових операційних систем, таких як Android та iOS, стали актуальними задачі, які раніше вирішувались для комп'ютерних операційних систем. Вибір макета життєво важливий для призначеного користувача інтерфейсу Android. Макет - це те, що дозволяє позиціонувати і розміщувати інші компоненти. Добре розуміння того, що пропонується в кожному з наявних шаблонів, може зробити різницю між простим і складним призначенням для користувача інтерфейсом як з точки зору програміста, так і користувача. Це особливо актуально, якщо потрібно підтримувати цілий ряд пристроїв.[1]

Для кожного додатка виникає необхідність використання деяких макетів, що допомагає зробити інтерфейс простим у використанні, модерним та цікавим. Саме тому актуальним є дослідження існуючих макетів для створення інтерфейсу, та вибір найбільш оптимального рішення в залежності від поставленої задачі.

Операційна система Android підтримує дві мови програмування: Java та Kotlin. Для проведення дослідження будемо використовувати мову програмування Kotlin. Макет, яка буде використовуватися Android Studio, - це ConstraintLayout, але це не єдиний макет, який можна використовувати з редактором макетів. Існує п'ять підтримуваних макетів[1]:

- ConstraintLayout;
- FrameLayout;
- LinearLayout;
- RelativeLayout;
- TableLayout.

Незважаючи на те, що ConstraintLayout є рекомендованим макетом для використання, все ще існує багато проектів Android, які використовують оригінальні макети, і деяким програмістам просто не подобається ConstraintLayout. Отже, варто знати, як працюють більш прості макети.

Серед оригінальних макетів найбільш часто використовуються RelativeLayout і LinearLayout, а FrameLayout посідає третє місце. Останні два, TableLayout і GridLayout, підходять для спеціалізованих

типів призначеного для користувача інтерфейсу, а в Android Studio 3 більш-менш не підтримуються в редакторі макетів, тому вам доведеться безпосередньо працювати з їх властивостями. З цієї причини їх краще уникати.[2]

ConstraintLayout - це новий макет, доступний у сховищі підтримки Android для створення гнучких та ефективних макетів. На відміну від попередніх макетів, він не входить у залежність від бібліотеки підтримки. Це дозволяє легко відправляти часті версії, тому рамкові версії Android не впливають на них безпосередньо. Цей макет був представлений у травні 2016 року під час заходу Google I / O. Макет має вищі переваги, кращі показники та більшу гнучкість порівняно з іншими макетами. Ця компоновка назад сумісна з API 9 (Android 2.3). Система ConstraintLayout складається з трьох частин: обмеження, рівняння та розв'язувач. Обмеження - це взаємозв'язок між вашими поглядами і визначається під час налаштування інтерфейсу користувача. Після створення цих відносин система переведе їх у лінійну систему рівнянь. У наступних частинах цієї серії ми детальніше розглянемо обмеження.[2]

FrameLayout призначений для блокування області на екрані для відображення одного елемента. Як правило, FrameLayout слід використовувати для утримання єдиного дочірнього представлення, оскільки це може бути важко організувати дитячі перегляди таким чином, щоб масштабувались до різних розмірів екрана, без того, щоб діти перетиналися між собою. Однак ви можете додати декілька дітей до FrameLayout і контролювати їх положення в межах FrameLayout, призначивши гравітацію кожній дитині, використовуючи атрибут android:layout_gravity.[3]

У студії макет LinearLayout представлений двома варіантами - горизонтальні та вертикальні. Макет LinearLayout виражає всі дочерні об'єкти в одночасному створенні - вертикально або горизонтально. Направлення задається при допомозі атрибута орієнтації android:orientation:"horizontal"("vertical"). Всі дочерні елементи містяться в одному з інших, так як вертикальний список компонентів буде містити лише один дочірній елемент, який є незалежним від того, який є досить широким. Горизонтальне розташування списку буде розміщувати елементи в одній строці з високою, рівномірно високою самою високою дочірньою елементом списку.[4]

RelativeLayout (відносна розмітка) знаходиться в розділі Layouts і дозволяє дочірнім компонентам визначати свою позицію щодо батьківського компонента або щодо сусідніх дочірніх елементів (ідентифікатора елемента). У RelativeLayout дочірні елементи розташовані так, що якщо перший елемент розташований по центру екрана, інші елементи, вирівняні щодо першого елемента, будуть вирівняні відносно центру екрана. При такому розташуванні, при оголошенні розмітки в XML-файлі, елемент, на який будуть посилатися для позиціонування інші об'єкти уявлення, повинен бути оголошений раніше, ніж інші елементи, які звертаються до нього за його ідентифікатором. [5]

Макет, який упорядковує своїх дітей у рядки та стовпці. TableLayout складається з декількох об'єктів android.widget.TableRow, кожен з яких визначає рядок. Контейнери TableLayout не відображають межі рядків для їх рядків, стовпців або комірок. Кожен рядок має нуль або більше комірок; кожна комірка може містити один об'єкт View. У таблиці є стільки стовпців, скільки рядків із найбільшою кількістю комірок. Таблиця може залишати клітинки порожніми. Клітини можуть охоплювати стовпці, як це можливо в HTML.[5]

Розглянувши різні засоби для створення користувацького інтерфейсу було обрано кілька макетів. Для вирішення поставленої задачі узгоджено та застосовано такі макети: ConstraintLayout, FrameLayout, LinearLayout. Саме таке поєднання методів дає найбільш гнучкі можливості для роботи з макетами, не навантажуючи UI-потік, а також дозволяє обробляти різні сценарії подій.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Android Programming In Kotlin: Layouts. [Електронний ресурс] : [Веб-сайт] – Електронні дані. - Режим доступу: <https://www.i-programmer.info/programming/android/13260-android-programming-in-kotlin-layouts.html?start=2>
2. Getting started with constraintlayout in kotlin. [Електронний ресурс] : [Веб-сайт] – Електронні дані. - Режим доступу: <https://pusher.com/tutorials/constraintlayout-kotlin-part-1>
3. LinearLayout. [Електронний ресурс] : [Веб-сайт] – Електронні дані. - Режим доступу: <http://developer.alexanderklimov.ru/android/layout/linearlayout.php#bottomline>

4. Android Programming In Kotlin: RelativeLayout. [Електронний ресурс] : [Веб-сайт] – Електронні дані. - Режим доступу: <http://developer.alexanderklimov.ru/android/layout/relativelayout.php>
5. TableLayout. [Електронний ресурс] : [Веб-сайт] – Електронні дані. - Режим доступу: <https://developer.android.com/reference/kotlin/android/widget/TableLayout>

Левицька Юлія Русланівна – студентка групи 1KI-16б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: yulia@snapp.team

Науковий керівник: Ткаченко Олександр Миколайович – к.т.н, доцент, відповідальний за наукову роботу на кафедрі обчислювальної техніки, Вінницький національний технічний університет, Вінниця, e-mail: alextk1960@gmail.com

Levytska Yuliia - student of the group 1KI-16b, Faculty of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: yulia@snapp.team

Scientific adviser: Alexander Tkachenko - Ph.D., Associate Professor, responsible for scientific work at the Department of Computer Engineering, Vinnitsa National Technical University, Vinnitsa, e-mail: alextk1960@gmail.com