

INFORMATION TECHNOLOGIES

Імплементація специфічного UI мобільного додатку у Android Lollipop

О. Г. Хамула, Н. В. Сорока

Українська Академія Друкарства, м. Львів, Україна

Paper received 21.06.2016; Accepted for publication 05.07.2016.

Анотація. Розглядаються нововведення в Android Lollipop ОС з точки зору користувацького інтерфейсу. Наводиться аналіз концепцій дизайну і варіанти їх реалізації з точки зору розробника. Надаються рекомендації по впровадженні UI особливостей концепту музичного програвача.

Ключові слова: UI, користувацький інтерфейс, мобільний додаток, прототип, дизайн, музичний програвач, Android 5, Lollipop.

Постановка проблеми. Незважаючи на значну кількість досліджень в галузі інформаційної технології розробки мобільних додатків, розробники періодично стикаються із необхідністю вибору кращих методів імплементації специфічного користувацького інтерфейсу. Це пов'язано із регулярними оновленнями мобільних операційних систем.

Компанія Google ближче до середини грудня 2015 збрала статистику своєї мобільної операційної системи Android за листопад. Зайняло це 7 днів, з 1 по 7 грудня, статистика збиралася в офіційному магазині Play Store, доступному з Android 2.2 і вищих версій.

Основний приріст частки користувачів в минулому місяці трапився у Android 5.x Lollipop. У жовтні частка цієї версії ледь перевищувала 25%, а тепер це значення зросло до 29,5%. Зокрема, версія Android 5.1 набрала за місяць 3,1%.

Android 4.4 KitKat на ринку присутня більше двох років і поки продовжує лідувати з часткою 36,6%, проте її спад виявився максимальним, склавши 1,2%. Всі інші версії Android 4.x також трохи втратили, в сумі 2,1%. У Android Jelly Bean частка дорівнює 2,9%, менше ніж у вийшла раніше за неї Gingerbread з 3,4%.

Через два місяці після релізу Android 6.0 Marshmallow швидких темпів її поширення традиційно не спостерігається. У момент появи оновлення Android 6.0.1 версія 6.0 добралася всього до 0,5% Android-пристроїв, збільшивши свою частку за місяць на 0,2%. Відповідно, 99,5% пристроїв не можуть похвалитися наявністю найбільш актуальної версії Android.

Виділення невіршених раніше частин загальної проблеми, котрим присвячується означена стаття. Дослідженням інформаційних систем мобільних

операційних систем присвячують свої наукові роботи багато авторів. Проте, Android Lollipop являється відносно новою ОС і на даний момент немає досліджень по проблемі активного залучення в дизайн System bars.

Питання впровадження в інтерфейс додатку анімації також є актуальним враховуючи, що робота базується на власному концепті мобільного додатку і відповідно оригінальним анімаційним ефектам.

Формування цілей статті. На основі дизайну користувацького інтерфейсу концепту мобільного додатку, дослідити варіанти його реалізації. Розглянути нововведення в UI для Android Lollipop ОС. Зокрема, провести аналіз проблеми активного залучення в дизайн System bars. Надати рекомендації по можливостях імплементації анімаційних ефектів, на основі концепту.

Виклад основного матеріалу дослідження з повним обґрунтуванням отриманих наукових результатів. Наш концепт додатку мобільного програвача базується, з точки зору візуального інтерфейсу – UI, на основах Material Design, про те із оригінальними рішеннями: відмова від використання тіней, активне залучення системних блоків операційної системи (Status Bar і Navigation Bar), анімація основних іконок, динамічна кольорова палітра інтерфейсу.

Відмова від використання тіней. Тіні в Material Design покликані розділяти елементи інтерфейсу у просторі по осі z. За закладеною ідеєю і особливістю UX, елементи інтерфейсу користувача розміщені в одній площині (Рис. 1). Зокрема акцент на одній площині диктується вертикальним свайпом, який в парі із суцільним кольором викликає відчуття єдності трьох екранів.

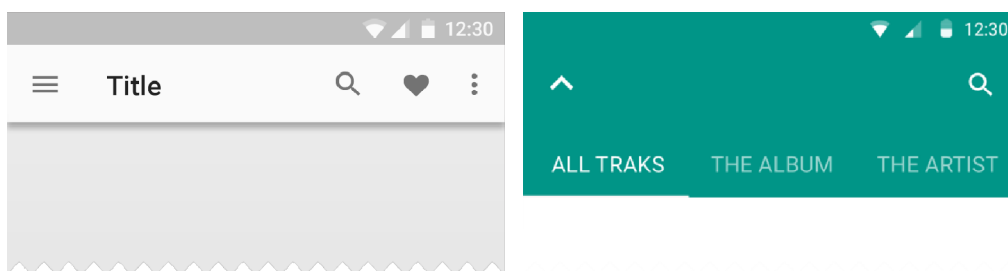


Рис. 1. Використання тіней у Material Design і їх відсутність у концепті.

Активне залучення в дизайн системних блоків операційної системи (Status Bar і Navigation Bar). За замовчуванням, при розробці додатку системні блоки забарв-

лені у чорний колір (Рис. 2). Починаючи із Android 5, для розробників відкрились додаткові можливості по стилізації цих блоків: забарвлення кольором, прозорість [1].

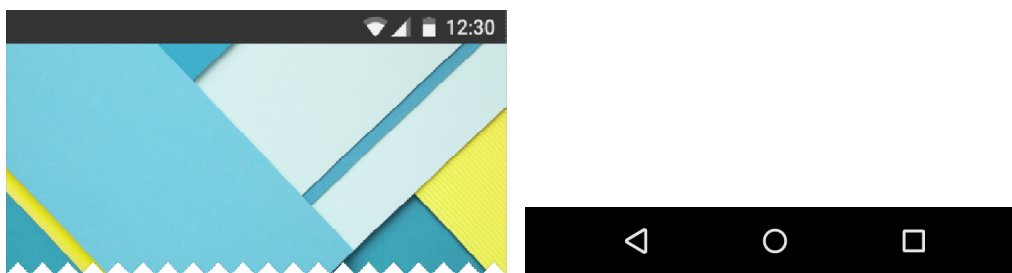


Рис. 2. Стандартний темний (Dark) тип Status і Navigation блоків.

Внесення змін до Status Bar полягають у використанні прозорості 20% чорного кольору (#000000). Це необхідно коли музичний файл містить зображення альбому до якого відноситься. Поверх цього зображення відображається загальна інформація, відповідно для її кращого відображення рекомендується використовувати підкладку. Відповідно в даному випадку використовується Translucent параметр:

```
<style name="AppTheme" parent="AppTheme.Base">
<item
name="android:windowTranslucentStatus">true</item>
```

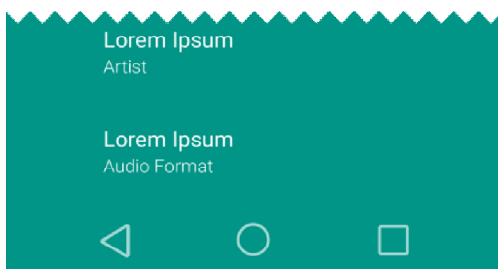


Рис. 3. Вигляд Navigation Bar концепту музичного програвача для Android 5+.

Для реалізації зміни кольору системного блоку навігації є декілька варіантів реалізації:

1. Редагування xml-файлу "values-v21/style.xml"

```
<item name="android:navigationBarColor">@color/
/navigationbar_color</item>
```

2. Редагування xml-файлу "values/style.xml"

```
<item name="android:navigationBarColor" tools:
:targetApi="21">@color/navigationbar_color</item>
```

3. Програмно

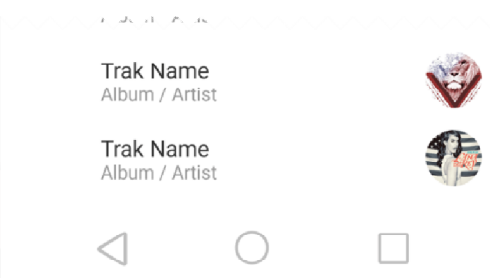
```
if (Build.VERSION.SDK_INT >= 21)
getWindow().setNavigationBarColor(getResources().getColor(R.color.navigationbar_color));
```

Анімація основних іконок. В дизайні концепту робиться акцент на простоту і чистоту у візуальному стилі і це досягається, також завдяки використанню іконок. Іконки рекомендується створювати у професійних програмах для векторної графіки, як приклад Adobe Illustrator, або використовувати готові із пакету іконок наданих компанією розробником ОС – Google.

Наша рекомендація полягає у використанні ресурсів іконок у форматі svg (Рис. 4). Оскільки даний формат представлення графіки дозволяє якісно застосовувати анімацію і зберігати пропорції в залежності із розміром екрану. Scalable Vector Graphics (скорочено SVG) – специфікація мови розмітки, що базується на XML та формат файлів для двовимірної векторної графіки, як статичної, так і анімованої та інтерактивної. SVG може бути вик-

</style>

Враховуючи логіку навігації концепту, що здебільшого полягає у вертикальному свайпі, постала необхідність динамічної стилізації системного блоку Navigation (Рис. 3). На головному екрані навігації і в крайній верхній точці(екран з інформацією про музичний файл, який активний на даний момент) системний блок зафарбовується кольором #019587, або #ff4b3b., іконки навігації приймають білий колір #ffffff. На екрані із списком музичних файлів(крайня нижня точка концепту при свайпі) Navigation Bar приймає параметр "Transparent" [2].



лючно декларативним, або містити описи сценаріїв. Зображення можуть містити зовнішні посилання шляхом застосування простих XLink-ів. Ця специфікація є відкритим стандартом, розробленим робочою групою SVG Working Group організації World Wide Web Consortium. Використання даного формату в Android проекті ми реалізували завдяки бібліотеці 1.2.2-beta-1-tweaked-2 [3].

Для створення іконок: press_to_play(okay_icon).svg, press_to_stop(2lines_icon).svg і progress_bar.svg використовувалося програмне забезпечення від компанії Adobe, Illustrator CC.

Приклад представлення res_to_play(okay_icon).svg у кодї:

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Generator: Adobe Illustrator 15.0.1, SVG Export
Plug-In . SVG Version: 6.00 Build 0) -->
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG
1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg version="1.1" id="Layer_2"
xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" x="0px"
y="0px"
width="280.184px" height="198.702px"
viewBox="-2.603 -1.977 280.184 198.702"
enable-background="new -2.603 -1.977
280.184 198.702" xml:space="preserve">
```



Рис. 4. Вигляд основних svg-іконок і часової шкали.

```
<polyline fill="none" stroke="#FFFFFF" stroke-  
width="4.5" stroke-miterlimit="10" points="0.091,114.9  
80.005,194.814
```

```
274.886,-0.065 "/>
```

```
</svg>
```

Враховуючи вищенаведений код, який описує графічний елемент за допомогою точок, ми можемо програмно змінювати ці параметри. В результаті отримувати ефектні, для користувача, анімаційні ефекти.

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямку. Користувачський інтерфейс, створеного нами концепту музичного програвача для Android ОС, базується на основних принципах концепції дизайну від Google. Проте нововведення запропоновані і реалізовані в прототипі відкривають нові границі для взаємодії з користувачем.

Грунтуючись на вищезгаданому, рекомендується активно використовувати UI ресурси з типом фалів

svg. Цей тип надає можливість використання одного ресурсу для різних розмірів екранів, оскільки масштабування проходить без втрати якості, також видозмінювати графічний елемент за допомогою функцій описаних в кодї.

Jon Wiley, один із керуючих директорів відділу дизайну компанії Google, наводить рекомендацію по активному впровадженню нових моделей взаємодії, зберігаючи лише базові принципи. Відповідно задається активний розвиток в цьому.

Перспективою подальшого дослідження є проведення тестування реалізованого концепту. Особливу увагу необхідно сконцентрувати на системному блоці навігації і блоці відображення статусу. Також, даний користувачський інтерфейс концепту передбачає взаємодію із керуванням жестами. Дана адаптація інтерфейсу є логічним і наступним етапом реалізації проекту в цілому.

ЛІТЕРАТУРА

1. Google Design. Layout Structure [Електронний ресурс]. – Режим доступу: <https://www.google.com/design/spec/layout/structure.html#structure-system-bars>
2. Developer Android. Using the Material Theme [Електронний ресурс]. – Режим доступу: <http://developer.android.com/training/material/theme.html>
3. Androidsvg [Електронний ресурс]. – Режим доступу: <http://mvnrepository.com/artifact/de.greenrobot/androidsvg/1.2.2-beta-1-tweaked-2>

REFERENCES

1. Google Design. Layout Structure. WEB resource: <https://www.google.com/design/spec/layout/structure.html#structure-system-bars>
2. Developer Android. Using the Material Theme. WEB resource: <http://developer.android.com/training/material/theme.html>
3. Androidsvg. WEB resources: <http://mvnrepository.com/artifact/de.greenrobot/androidsvg/1.2.2-beta-1-tweaked-2>

How to implement a special user interface in Android Lollipop

Soroka N., Khamula O.

Abstract. We review a new user interface structure in Android Lollipop. We describe our innovative mobile design concept, with special logic of user interaction, and visual elements for support that logic. The Android Lollipop uses material design style like a default, however our visual elements need non-standard implementation. A main accent is on custom status and navigation bars, also on svg animations.

Keywords: UI, concept, app, android.

Имплементация специфического UI мобильного приложения на Android Lollipop

Н. В. Сорока, О. Г. Хамула

Аннотация. Мы рассматриваем новую структуру пользовательского интерфейса в Android Lollipop. Мы описываем нашу инновационную концепцию мобильного дизайна, с особой логикой взаимодействия с пользователем, а также с визуальными элементами для поддержки этой логики. Android Lollipop использует "material" стиль по умолчанию, однако наши визуальные элементы имеют нестандартную реализацию, которая и описана в статье. Основной акцент делается на адаптированном дизайне "StatusBar&NavBar" элементов, а также на SVG анимации.

Ключевые слова: интерфейс, концепт, мобильное приложение, android.