Menu

Menu là một thành phần giao diện người dùng phổ biến trong nhiều loại ứng dụng. Để cung cấp một trải nghiệm người dùng quen thuộc và nhất quán, bạn nên sử dụng các API Menu để trình bày hành động người dùng và các tùy chọn khác trong hoạt động của mình.

Bắt đầu với Android 3.0 (API mức 11), các thiết bị dựa trên nền tảng Android không còn phải cung cấp một nút *Menu* chuyên dụng nữa. Với sự thay đổi này, các ứng dụng Android cần tránh khỏi sự phụ thuộc vào bảng điều khiển menu 6 mục truyền thống này mà thay vào đó cung cấp một thanh hành động để trình bày các hành động người dùng thông dụng.

Mặc dù thiết kế và trải nghiệm người dùng đối với một số mục menu đã thay đổi, ngữ nghĩa để định nghĩa tập hợp hành động và tùy chọn thì vẫn dựa trên các API Menu. Hướng dẫn này trình bày cách tạo ba loại menu hay trình bày hành động cơ bản trên tất cả phiên bản Android:

Menu tùy chọn và thanh hành động

Menu tùy chọn là tập hợp các mục menu cơ bản cho một hoạt động. Đó là nơi bạn nên đặt các hành động có tác động chung tới ứng dụng, chẳng hạn như "Tìm kiếm," "Soạn e-mail" và "Cài đặt."

Nếu bạn đang phát triển cho phiên bản Android 2.3 hoặc thấp hơn, người dùng có thể hiện bảng điều khiển menu tùy chọn bằng cách nhấn nút *Menu*.

Trên phiên bản Android 3.0 trở lên, các mục từ menu tùy chọn được trình bày bởi thanh hành động, là sự kết hợp giữa các mục hành động trên màn hình và các tùy chọn tràn. Bắt đầu với phiên bản Android 3.0, nút *Menu* bị bỏ đi (một số thiết bị không có), vì thế bạn nên chuyển sang sử dụng thanh hành động để cho phép truy cập vào hành động và các tùy chọn khác.

Menu ngữ cảnh và chế độ hành động theo ngữ cảnh

Menu ngữ cảnh là một menu nổi xuất hiện khi người dùng thực hiện nhấp giữ trên một phần tử. Nó cung cấp các hành động ảnh hưởng tới nội dung hoặc khung ngữ cảnh được chọn.

Khi phát triển cho phiên bản Android 3.0 trở lên, thay vào đó, bạn nên sử dụng chế độ hành động theo ngữ cảnh để kích hoạt các hành động trên nội dung được chọn. Chế độ này hiển thị các mục hành động ảnh hưởng tới nội dung được chọn trong một thanh ở trên cùng của màn hình và cho phép người dùng chọn nhiều mục.

Menu bật lên

Menu bật lên sẽ hiển thị danh sách các mục trong một danh sách thẳng đứng được neo vào dạng xem đã gọi ra menu. Nên cung cấp một phần tràn gồm các hành động liên quan tới nội dung cụ thể hoặc nhằm cung cấp các tùy chọn cho phần thứ hai của một lệnh. Các hành động trong một menu bật lên **không** nên trực tiếp ảnh hưởng tới nội dung tương ứng—đó là việc của hành động ngữ cảnh . Thay vào đó, menu bật lên áp dụng cho các hành động mở rộng liên quan tới các vùng nội dung trong hoạt động của bạn.

Định nghĩa một Menu trong XML

Đối với tất cả các loại menu, Android cung cấp một định dạng XML chuẩn để định nghĩa các mục menu. Thay vì xây dựng một menu trong mã của hoạt động của bạn, bạn nên định nghĩa một menu và tất cả các mục của nó trong một tài nguyên menu XML. Khi đó, bạn có thể bung tài nguyên menu (tải nó như một đối tượng Menu) trong hoạt động hoặc phân đoạn của mình.

Sử dụng một tài nguyên menu là một cách làm hay vì một vài lý do:

- Nó dễ trực quan hóa cấu trúc menu trong XML hơn.
- Nó tách riêng nội dung cho menu với mã hành vi của ứng dụng của bạn.
- Nó cho phép bạn tạo các cấu hình menu phái sinh cho các phiên bản nền tảng, kích cỡ màn hình khác nhau và các cấu hình khác bằng cách tận dụng khuôn khổ tài nguyên ứng dụng.

Để định nghĩa menu, hãy tạo một tệp XML bên trong thư mục `res/menu/` dự án của bạn và xây dựng menu với các phần tử sau:

`<menu>`

Định nghĩa một Menu, đó là một bộ chứa các mục menu. Phần tử `<menu>` phải là một nút gốc cho tệp và có thể giữ một hoặc nhiều phần tử `<item>` và `<group>`.

`<item>`

Tạo một MenuItem, nó biểu diễn một mục đơn trong một menu. Phần tử này có thể chứa một phần tử `<menu>` được lồng nhau để tạo một menu con.

`<group>`

Một bộ chứa tùy chọn, vô hình cho các phần tử `&lt;item&gt;`. Nó cho phép bạn phân loại các mục menu sao cho chúng chia sẻ các tính chất như trạng thái hiện hoạt và khả năng hiển thị. Để biết thêm thông tin, hãy xem phần nói về Tạo Nhóm Menu.

Sau đây là một menu ví dụ có tên là `game_menu.xml`:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
   <item android:id="@+id/new_game"
        android:icon="@drawable/ic_new_game"
        android:title="@string/new_game"
        android:showAsAction="ifRoom"/>
   <item android:id="@+id/help"
        android:icon="@drawable/ic_help"
        android:title="@string/help" />
</menu>
```

Phần tử <item> hỗ trợ một vài thuộc tính bạn có thể sử dụng để định nghĩa biểu hiện bên ngoài và hành vi của một mục. Các mục trong menu trên bao gồm những thuộc tính sau:

android:id

Một ID tài nguyên duy nhất đối với mục, nó cho phép ứng dụng có thể nhận ra mục đó khi người dùng chọn nó.

android:icon

Một tham chiếu tới một nội dung vẽ được để dùng làm biểu tượng của mục.

android:title

Một tham chiếu tới một xâu để dùng làm tiêu đề của mục.

android:showAsAction

Quy định thời điểm và cách thức mục này nên xuất hiện như một mục hành động trong thanh hành động.

Đây là những thuộc tính quan trọng nhất bạn nên sử dụng, nhưng còn nhiều thuộc tính sẵn có khác.

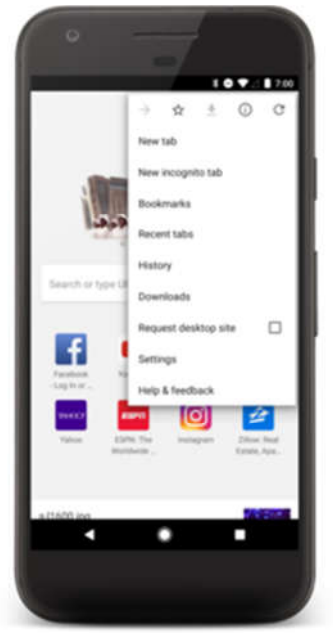
Bạn có thể thêm một menu con vào một mục trong bất kỳ menu nào (ngoại trừ menu con) bằng cách thêm một phần tử <menu> làm con của <item>. Các menu con thường hữu ích khi ứng dụng của bạn có nhiều chức năng mà có thể được tổ chức thành các chủ đề, như các mục trong thanh menu của một ứng dụng PC (Tệp, Chỉnh sửa, Dạng xem, v.v.). Ví dụ:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
   <item android:id="@+id/file"
        android:title="@string/file" >
      <!-- "file" submenu -->
      <menu>
```

```
        <item android:id="@+id/create_new"
            android:title="@string/create_new" />
        <item android:id="@+id/open"
            android:title="@string/open" />
    </menu>
  </item>
</menu>
```

Để sử dụng menu trong hoạt động của mình, bạn cần bung tài nguyên menu (chuyển tài nguyên XML thành một đối tượng có thể lập trình) bằng cách sử dụng MenuInflater.inflate(). Trong những phần sau, bạn sẽ biết cách bung một menu đối với mỗi loại menu.

Tạo một Menu Tùy chọn



Hình 1. Các menu tùy chọn trong Trình duyệt, trên Android 2.3.

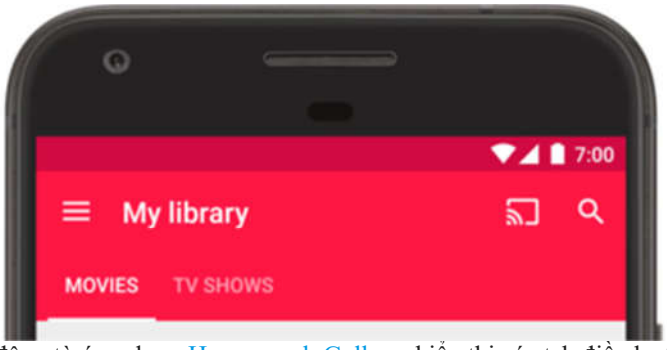
Menu tùy chọn là nơi bạn nên đưa vào hành động và các tùy chọn khác liên quan tới ngữ cảnh hoạt động hiện tại, chẳng hạn như "Tìm kiếm," "Soạn e-mail," và "Cài đặt."

Nơi mà các mục trong menu tùy chọn của bạn xuất hiện trên màn hình sẽ phụ thuộc vào phiên bản mà bạn phát triển ứng dụng của mình cho:

Nếu bạn phát triển ứng dụng của mình cho phiên bản **Android 2.3.x (API mức 10) hoặc thấp hơn**, nội dung của menu tùy chọn sẽ xuất hiện ở dưới cùng màn hình khi người dùng nhấn nút *Menu* như minh họa trong hình 1. Khi được mở, phần hiển thị đầu tiên là menu biểu tượng với tối đa sáu mục menu. Nếu menu của bạn bao gồm

nhiều hơn sáu mục, Android sẽ đặt mục thứ sáu và phần còn lại vào một menu tràn mà người dùng có thể mở bằng cách chọn *Thêm nữa*.

Nếu bạn phát triển ứng dụng của mình cho phiên bản **Android 3.0 (API mức 11) và cao hơn**, các mục từ menu tùy chọn sẵn ở trong thanh hành động. Theo mặc định, hệ thống đặt tất cả các mục trong phần tràn hành động mà người dùng có thể hiện bằng biểu tượng tràn hành động phía bên phải của thanh hành động (hoặc bằng cách nhấn nút *Menu* của thiết bị nếu có). Để kích hoạt truy cập nhanh vào các hành động quan trọng, bạn có thể đưa lên một vài mục xuất hiện trong thanh hành động bằng cách thêm android:showAsAction="ifRoom" vào phần tử <item>tương ứng (xem hình 2).



Hình 2. Thanh hành động từ ứng dụng Honeycomb Gallery, hiển thị các tab điều hướng và một mục hành động máy ảnh (cộng với nút tràn hành động).

Bạn có thể khai báo các mục cho menu tùy chọn từ lớp con Activity của bạn hoặc một lớp con Fragment. Nếu cả hoạt động của bạn và (các) phân đoạn đều khai báo các mục cho menu tùy chọn, chúng sẽ được kết hợp lại trong UI. Các mục của hoạt động xuất hiện trước, sau đó là các mục của từng phân đoạn theo thứ tự phân đoạn được thêm vào hoạt động. Nếu cần, bạn có thể sắp xếp lại các mục menu bằng thuộc tính android:orderInCategory trong mỗi <item> mà bạn cần di chuyển.

Để quy định menu tùy chọn cho một hoạt động, hãy khống chế onCreateOptionsMenu() (các phân đoạn cung cấp phương pháp gọi lại onCreateOptionsMenu() của chính mình). Trong phương pháp này, bạn có thể bung tài nguyên menu của mình (được định nghĩa trong XML) vào Menu được cung cấp trong phương pháp gọi lại. Ví dụ:

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.game_menu, menu);
    return true;
}
```

Bạn cũng có thể thêm các mục menu bằng cách sử dụng add() và truy xuất các mục bằng findItem() để xem lại tính chất của chúng bằng các API MenuItem.

Nếu bạn phát triển ứng dụng của mình cho phiên bản Android 2.3.x và thấp hơn, hệ thống gọi onCreateOptionsMenu() để tạo menu tùy chọn khi người dùng mở menu lần đầu tiên. Nếu bạn phát triển cho phiên bản Android 3.0 vào cao hơn, hệ thống sẽ gọi onCreateOptionsMenu() khi bắt đầu hoạt động để hiển thị các mục cho thanh hành động.

Xử lý sự kiện nhấp

Khi người dùng chọn một mục từ menu tùy chọn (bao gồm các mục hành động trong thanh hành động), hệ thống sẽ gọi phương pháp onOptionsItemSelected() của hoạt động của bạn. Phương pháp này thông qua MenuItem được chọn. Bạn có thể nhận biết mục bằng cách gọi getItemId(), nó trả về ID duy nhất cho mục menu (được định nghĩa bởi thuộc tính android:id trong tài nguyên menu hoặc bằng một số nguyên được cấp cho phương pháp add()). Bạn có thể khớp ID này với các mục menu đã biết để thực hiện hành động phù hợp. Ví dụ:

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
   // Handle item selection
   switch (item.getItemId()) {
      case R.id.new_game:
         newGame();
         return true;
      case R.id.help:
         showHelp();
         return true;
      default:
         return super.onOptionsItemSelected(item);
   }
}
```

Khi bạn xử lý thành công một mục menu, trả về true. Nếu không xử lý được mục menu, bạn nên gọi triển khai siêu lớp của onOptionsItemSelected() (triển khai mặc định trả về sai).

Nếu hoạt động của bạn bao gồm các phân đoạn, trước tiên hệ thống sẽ gọi onOptionsItemSelected() cho hoạt động, rồi mới cho từng phân đoạn (theo thứ tự thêm phân đoạn) tới khi trả về true hoặc tất cả phân đoạn đều được gọi.

Thay đổi các mục menu vào thời gian chạy

Sau khi hệ thống gọi onCreateOptionsMenu(), nó sẽ giữ lại một thực thể của Menu mà bạn đưa vào và sẽ không gọi lại onCreateOptionsMenu() trừ khi menu bị vô hiệu hóa vì lý do nào đó. Tuy nhiên, bạn chỉ nên sử dụng onCreateOptionsMenu() để tạo trạng thái menu ban đầu chứ không phải để thực hiện thay đổi trong vòng đời của hoạt động.

Nếu bạn muốn sửa đổi menu tùy chọn dựa trên các sự kiện xảy ra trong vòng đời của hoạt động, bạn có thể làm vậy trong phương pháp onPrepareOptionsMenu(). Phương pháp này chuyển cho bạn đối tượng Menu như hiện đang có để bạn có thể sửa đổi nó, chẳng hạn như thêm, xóa bỏ, hoặc vô hiệu hóa các mục. (Phân đoạn cũng cung cấp lệnh gọi lại onPrepareOptionsMenu().)

Trên phiên bản Android 2.3.x và thấp hơn, hệ thống gọi onPrepareOptionsMenu() mỗi lần người dùng mở menu tùy chọn (nhấn nút *Menu*).

Trên phiên bản Android 3.0 trở lên, menu tùy chọn được coi như luôn mở khi các mục menu được trình bày trong thanh hành động. Khi một sự kiện xảy ra và bạn muốn thực hiện một cập nhật menu, bạn phải gọi invalidateOptionsMenu() để yêu cầu hệ thống gọi onPrepareOptionsMenu().

Tạo một Menu Ngữ cảnh

Hình 3. Ảnh chụp màn hình một menu ngữ cảnh nổi (trái) và thanh hành động ngữ cảnh (phải).

Menu ngữ cảnh sẽ đưa ra các hành động ảnh hưởng tới một mục hoặc khung ngữ cảnh cụ thể trong UI. Bạn có thể cung cấp một menu ngữ cảnh cho bất kỳ dạng xem nào, nhưng chúng thường được sử dụng nhiều nhất cho các mục trong một ListView, GridView, hoặc các bộ sưu tập dạng xem khác mà người dùng có thể thực hiện hành động trực tiếp trên mỗi mục.

Có hai cách để cung cấp các hành động ngữ cảnh:

Trong một menu ngữ cảnh nổi. Menu xuất hiện như một danh sách nổi gồm nhiều mục menu (tương tự như một hộp thoại) khi người dùng thực hiện nhấp giữ (nhấn và giữ) trên một dạng xem có khai báo hỗ trợ menu ngữ cảnh. Người dùng có thể thực hiện hành động ngữ cảnh trên một mục vào một thời điểm.

Trong chế độ hành động theo ngữ cảnh. Chế độ này là một hệ thống triển khai ActionMode có chức năng hiển thị một *thanh hành động ngữ cảnh* ở bên trên màn hình với các mục hành động ảnh hưởng tới (các) mục được chọn. Khi chế độ này hiện hoạt, người dùng có thể thực hiện một hành động trên nhiều mục ngay lập tức (nếu ứng dụng của bạn cho phép).

Tạo một menu ngữ cảnh nổi

Để cung cấp một menu ngữ cảnh nổi:

1. Đăng ký View mà menu ngữ cảnh nên được liên kết với bằng cách gọi registerForContextMenu() và chuyển cho nó View.

   Nếu hoạt động của bạn sử dụng một ListView hoặc GridView và bạn muốn từng mục cung cấp cùng menu ngữ cảnh, hãy đăng ký tất cả mục cho một menu ngữ cảnh bằng cách chuyển ListView hoặc GridView cho registerForContextMenu().

2. Triển khai phương pháp onCreateContextMenu() trong Activity hoặc Fragment của bạn.

   Khi dạng xem được đăng ký nhận được một sự kiện nhấp giữ, hệ thống sẽ gọi phương pháp onCreateContextMenu() của bạn. Đây là nơi bạn định nghĩa các mục menu, thường bằng cách bung một tài nguyên menu. Ví dụ:

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
                                ContextMenuInfo menuInfo) {
  super.onCreateContextMenu(menu, v, menuInfo);
  MenuInflater inflater = getMenuInflater();
  inflater.inflate(R.menu.context_menu, menu);
}
```

   MenuInflater cho phép bạn bung menu ngữ cảnh từ một tài nguyên menu. Các tham số của phương pháp gọi lại bao gồm View mà người dùng đã chọn và một đối tượng ContextMenu.ContextMenuInfo cung cấp thông tin bổ sung về mục được chọn. Nếu hoạt động của bạn có một vài dạng xem mà mỗi dạng cung cấp một menu ngữ cảnh khác nhau, bạn có thể sử dụng những tham số này để xác định menu ngữ cảnh nào cần bung.

3. Triển khai onContextItemSelected().

   Khi người dùng chọn một mục menu, hệ thống sẽ gọi phương pháp này để bạn có thể thực hiện hành động phù hợp. Ví dụ:

```
@Override
public boolean onContextItemSelected(MenuItem item) {
   AdapterContextMenuInfo info = (AdapterContextMenuInfo) item.getMenuInfo();
   switch (item.getItemId()) {
      case R.id.edit:
         editNote(info.id);
         return true;
      case R.id.delete:
         deleteNote(info.id);
         return true;
      default:
         return super.onContextItemSelected(item);
   }
}
```

Phương pháp `getItemId()` sẽ truy vấn ID cho mục menu được chọn, bạn nên gán mục này cho từng mục menu trong XML bằng cách sử dụng thuộc tính `android:id` như trình bày trong phần về Định nghĩa một Menu trong XML.

Khi bạn xử lý thành công một mục menu, trả về `true`. Nếu bạn không xử lý mục menu, bạn nên chuyển mục menu đó tới triển khai siêu lớp. Nếu hoạt động của bạn bao gồm nhiều phân đoạn, hoạt động sẽ nhận được lệnh gọi lại này trước. Bằng cách gọi siêu lớp khi chưa được xử lý, hệ thống sẽ chuyển sự kiện tới phương pháp gọi lại tương ứng trong từng phân đoạn, lần lượt (theo thứ tự thêm phân đoạn) tới khi `true` hoặc `false` được trả về. (Triển khai mặc định cho `Activity` và `android.app.Fragment` sẽ trả về `false`, vì thế bạn nên luôn gọi siêu lớp khi chưa được xử lý.)

Sử dụng chế độ hành động theo ngữ cảnh

Chế độ hành động theo ngữ cảnh là một triển khai hệ thống `ActionMode` tập trung vào tương tác người dùng hướng tới việc thực hiện các hành động theo ngữ cảnh. Khi một người dùng kích hoạt chế độ này bằng cách chọn một mục, một *thanh hành động ngữ cảnh* sẽ xuất hiện bên trên màn hình để trình bày các hành động mà người dùng có thể thực hiện trên (các) mục đang được chọn. Trong khi chế độ này được kích hoạt, người dùng có thể chọn nhiều mục (nếu bạn cho phép), bỏ chọn mục, và tiếp tục điều hướng trong hoạt động (miễn là bạn sẵn lòng cho phép). Chế độ hành động bị vô hiệu hóa và thanh hành động ngữ cảnh biến mất khi người dùng bỏ chọn tất cả các mục, nhấn nút QUAY LẠI, hoặc chọn hành động *Xong* ở phía bên trái của thanh.

Nếu bạn đang phát triển cho phiên bản Android 3.0 (API mức 11) hoặc cao hơn, bạn nên sử dụng chế độ hành động theo ngữ cảnh để trình bày các hành động ngữ cảnh, thay vì sử dụng menu ngữ cảnh nổi.

Đối với các dạng xem cung cấp hành động ngữ cảnh, bạn nên thường xuyên gọi ra chế độ hành động theo ngữ cảnh khi xảy ra một trong hai sự kiện sau (hoặc cả hai):

- Người dùng thực hiện nhấp giữ trên dạng xem.
- Người dùng chọn một hộp kiểm hoặc một thành phần UI tương tự trong dạng xem.

Cách ứng dụng của bạn gọi ra chế độ hành động theo ngữ cảnh và định nghĩa hành vi cho từng hành động phụ thuộc vào thiết kế của bạn. Cơ bản có hai thiết kế:

- Đối với các hành động ngữ cảnh trên các dạng xem riêng lẻ, tùy ý.
- Đối với các hành động ngữ cảnh hàng loạt trên các nhóm mục trong một ListView hoặc GridView (cho phép người dùng chọn nhiều mục và thực hiện một hành động trên tất cả).

Các phần sau mô tả phần thiết lập cần thiết đối với từng kịch bản.

Kích hoạt chế độ hành động theo ngữ cảnh cho các dạng xem riêng lẻ

Nếu muốn gọi ra chế độ hành động theo ngữ cảnh chỉ khi người dùng chọn các dạng xem cụ thể, bạn nên:

1. Triển khai giao diện ActionMode.Callback. Trong các phương pháp gọi lại của giao diện, bạn có thể quy định các hành động cho thanh hành động ngữ cảnh, hồi đáp các sự kiện nhấp trên mục hành động, và xử lý các sự kiện vòng đời khác đối với chế độ hành động.
2. Gọi startActionMode() khi bạn muốn hiển thị thanh (chẳng hạn như khi người dùng nhấp giữ dạng xem).

Ví dụ:

1. Triển khai giao diện ActionMode.Callback:

```
private ActionMode.Callback mActionModeCallback = new ActionMode.Callback() {

  // Called when the action mode is created; startActionMode() was called
  @Override
  public boolean onCreateActionMode(ActionMode mode, Menu menu) {
     // Inflate a menu resource providing context menu items
     MenuInflater inflater = mode.getMenuInflater();
     inflater.inflate(R.menu.context_menu, menu);
     return true;
  }

  // Called each time the action mode is shown. Always called after onCreateActionMode, but
  // may be called multiple times if the mode is invalidated.
  @Override
  public boolean onPrepareActionMode(ActionMode mode, Menu menu) {
     return false; // Return false if nothing is done
  }

  // Called when the user selects a contextual menu item
  @Override
```

```
    public boolean onActionItemClicked(ActionMode mode, MenuItem item) {
        switch (item.getItemId()) {
            case R.id.menu_share:
                shareCurrentItem();
                mode.finish(); // Action picked, so close the CAB
                return true;
            default:
                return false;
        }
    }

    // Called when the user exits the action mode
    @Override
    public void onDestroyActionMode(ActionMode mode) {
        mActionMode = null;
    }
};
```

Lưu ý rằng những phương pháp gọi lại sự kiện này hầu như giống với các phương pháp gọi lại đối với menu tùy chọn, khác ở chỗ từng phương pháp cũng chuyển đối tượng ActionMode được liên kết với sự kiện đó. Bạn có thể sử dụng các API ActionMode để thực hiện những thay đổi khác nhau với CAB, chẳng hạn như sửa đổi tiêu đề và phụ đề bằng setTitle() và setSubtitle() (hữu ích khi muốn cho biết có bao nhiêu mục được chọn).

Cũng lưu ý rằng các bộ mẫu trên sẽ đặt biến mActionMode là rỗng khi chế độ hành động bị hủy. Ở bước tiếp theo, bạn sẽ thấy cách nó được khởi tạo và việc lưu biến thành viên trong hoạt động hoặc phân đoạn của bạn có thể hữu ích như thế nào.

2. Gọi startActionMode() để kích hoạt chế độ hành động theo ngữ cảnh khi phù hợp, chẳng hạn như để hồi đáp lại một sự kiện nhấp giữ trên một View:

```
someView.setOnLongClickListener(new View.OnLongClickListener() {
    // Called when the user long-clicks on someView
    public boolean onLongClick(View view) {
        if (mActionMode != null) {
            return false;
        }

        // Start the CAB using the ActionMode.Callback defined above
        mActionMode = getActivity().startActionMode(mActionModeCallback);
        view.setSelected(true);
        return true;
    }
});
```

Khi bạn gọi startActionMode(), hệ thống sẽ trả về ActionMode được tạo. Bằng cách lưu điều này trong một biến thành viên, bạn có thể thực hiện thay đổi thanh hành động theo ngữ cảnh để hồi đáp những sự kiện khác. Trong mẫu trên, ActionMode được sử dụng để đảm bảo rằng thực thể ActionMode không được tạo lại nếu nó đã hiện hoạt, bằng cách kiểm tra xem thành viên có rỗng không trước khi khởi động chế độ hành động.

Kích hoạt hành động theo ngữ cảnh hàng loạt trong ListView hoặc GridView

Nếu bạn có một bộ sưu tập các mục trong một ListView hoặc GridView (hoặc một phần mở rộng khác của AbsListView) và muốn cho phép người dùng thực hiện các hành động hàng loạt, bạn nên:

- Triển khai giao diện AbsListView.MultiChoiceModeListener và đặt nó cho nhóm dạng xem bằng setMultiChoiceModeListener(). Trong các phương pháp gọi lại của trình nghe, bạn có thể quy định các hành động cho thanh hành động theo ngữ cảnh, hồi đáp các sự kiện nhấp trên các mục hành động, và xử lý các phương pháp gọi lại khác được kế thừa từ giao diện ActionMode.Callback.
- Gọi setChoiceMode() bằng tham đối CHOICE_MODE_MULTIPLE_MODAL.

Ví dụ:

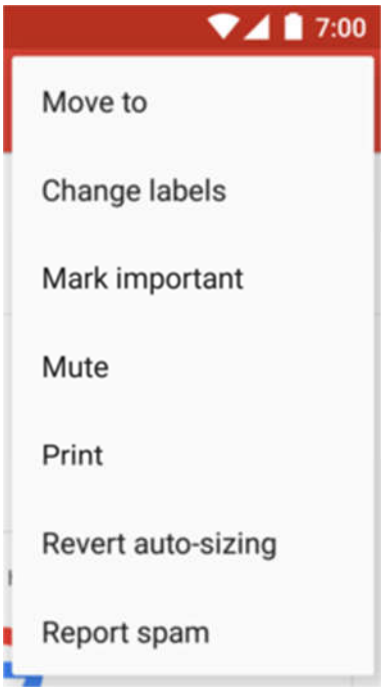
```
ListView listView = getListView();
listView.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE_MODAL);
listView.setMultiChoiceModeListener(new MultiChoiceModeListener() {

    @Override
    public void onItemCheckedStateChanged(ActionMode mode, int position,
                                          long id, boolean checked) {
        // Here you can do something when items are selected/de-selected,
        // such as update the title in the CAB
    }

    @Override
    public boolean onActionItemClicked(ActionMode mode, MenuItem item) {
        // Respond to clicks on the actions in the CAB
        switch (item.getItemId()) {
            case R.id.menu_delete:
                deleteSelectedItems();
                mode.finish(); // Action picked, so close the CAB
                return true;
            default:
                return false;
        }
    }

    @Override
    public boolean onCreateActionMode(ActionMode mode, Menu menu) {
```

```
        // Inflate the menu for the CAB
        MenuInflater inflater = mode.getMenuInflater();
        inflater.inflate(R.menu.context, menu);
        return true;
    }

    @Override
    public void onDestroyActionMode(ActionMode mode) {
        // Here you can make any necessary updates to the activity when
        // the CAB is removed. By default, selected items are deselected/unchecked.
    }

    @Override
    public boolean onPrepareActionMode(ActionMode mode, Menu menu) {
        // Here you can perform updates to the CAB due to
        // an invalidate() request
        return false;
    }
});
```

Vậy là xong. Lúc này, khi người dùng chọn một mục bằng nhấp giữ, hệ thống sẽ gọi phương pháp onCreateActionMode() và hiển thị thanh hành động theo ngữ cảnh với các hành động được quy định. Trong khi thanh hành động theo ngữ cảnh hiển thị, người dùng có thể chọn thêm mục.

Trong một số trường hợp mà các hành động ngữ cảnh cung cấp các mục hành động chung, bạn có thể muốn thêm một hộp kiểm hoặc một phần tử UI tương tự để cho phép người dùng chọn các mục, vì họ có thể không phát hiện được hành vi nhấp giữ. Khi một người dùng chọn hộp kiểm, bạn có thể gọi ra chế độ hành động theo ngữ cảnh bằng cách thiết đặt mục danh sách tương ứng về trạng thái đã chọn bằng setItemChecked().

Tạo một Menu Bật lên



Hình 4. Menu bật lên trong ứng dụng Gmail, được neo vào nút tràn ở trên cùng bên phải.

PopupMenu là một menu mô thái được neo vào một View. Nó xuất hiện bên dưới dạng xem dấu neo nếu có khoảng trống, hoặc bên trên dạng xem nếu không. Nó có ích cho việc:

Cung cấp một menu kiểu tràn cho các hành động mà *liên quan tới* nội dung cụ thể (chẳng hạn như tiêu đề e-mail của Gmail như minh họa trong hình 4).

- Cung cấp một phần thứ hai của câu lệnh (chẳng hạn như một nút được đánh dấu "Thêm" có chức năng tạo ra một menu bật lên với các tùy chọn "Thêm" khác nhau).
- Cung cấp một danh sách thả xuống tương tự như Spinner, nó không giữ lại một lựa chọn liên tục.

Nếu bạn định nghĩa menu của mình trong XML, sau đây là cách bạn có thể hiển thị menu bật lên:

1. Khởi tạo một PopupMenu bằng hàm dựng của nó, có chức năng đưa ứng dụng hiện tại Context và Viewtới menu mà sẽ được neo.
2. Sử dụng MenuInflater để bung tài nguyên menu của bạn vào đối tượng Menu được trả về bởi PopupMenu.getMenu(). Trên API mức 14 trở lên, bạn có thể sử dụng PopupMenu.inflate() thay thế.
3. Gọi PopupMenu.show().

Ví dụ, sau đây là một nút với thuộc tính android:onClick có chức năng hiển thị một menu bật lên:

```
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/ic_overflow_holo_dark"
    android:contentDescription="@string/descr_overflow_button"
    android:onClick="showPopup" />
```

Khi đó, hoạt động có thể hiển thị menu bật lên như sau:

```
public void showPopup(View v) {
    PopupMenu popup = new PopupMenu(this, v);
    MenuInflater inflater = popup.getMenuInflater();
    inflater.inflate(R.menu.actions, popup.getMenu());
    popup.show();
}
```

Trong API mức 14 trở lên, bạn có thể kết hợp hai dòng có chức năng bung menu bằng PopupMenu.inflate().

Menu bị bỏ qua khi người dùng chọn một mục hoặc chạm vào bên ngoài vùng menu. Bạn có thể lắng nghe báo hiệu sự kiện bỏ bằng cách sử dụng PopupMenu.OnDismissListener.

Xử lý sự kiện nhấp

Để thực hiện một hành động khi người dùng chọn một mục menu, bạn phải triển khai giao diện PopupMenu.OnMenuItemClickListener và đăng ký nó với PopupMenu của mình bằng cách gọi setOnMenuItemclickListener(). Khi người dùng chọn một mục, hệ thống sẽ gọi lệnh gọi lại onMenuItemClick() trong giao diện của bạn.

Ví dụ:

```
public void showMenu(View v) {
    PopupMenu popup = new PopupMenu(this, v);

    // This activity implements OnMenuItemClickListener
    popup.setOnMenuItemClickListener(this);
    popup.inflate(R.menu.actions);
    popup.show();
}

@Override
public boolean onMenuItemClick(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.archive:
            archive(item);
            return true;
```

```
        case R.id.delete:
            delete(item);
            return true;
        default:
            return false;
    }
}
```

Tạo Nhóm Menu

Nhóm menu là một tập hợp các mục menu chia sẻ những đặc điểm nhất định. Với một nhóm, bạn có thể :

- Hiển thị hoặc ẩn tất cả các mục bằng setGroupVisible()
- Kích hoạt hoặc vô hiệu hóa tất cả các mục bằng setGroupEnabled()
- Quy định xem tất cả các mục có thể chọn hay không bằng setGroupCheckable()

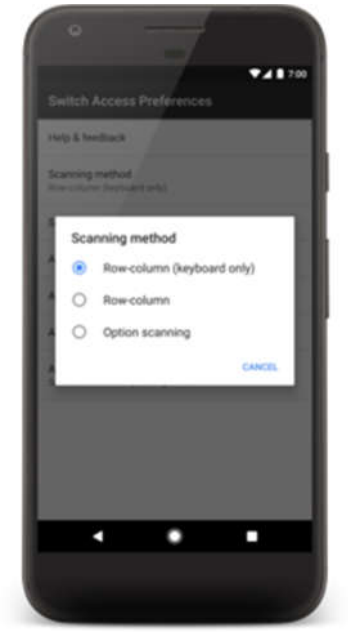
Bạn có thể tạo một nhóm bằng cách lồng các phần tử <item> bên trong một phần tử <group>vào tài nguyên menu của bạn hoặc bằng cách quy định một ID nhóm bằng phương pháp add().

Sau đây là một ví dụ về tài nguyên menu bao gồm một nhóm:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/menu_save"
        android:icon="@drawable/menu_save"
        android:title="@string/menu_save" />
    <!-- menu group -->
    <group android:id="@+id/group_delete">
        <item android:id="@+id/menu_archive"
            android:title="@string/menu_archive" />
        <item android:id="@+id/menu_delete"
            android:title="@string/menu_delete" />
    </group>
</menu>
```

Các mục nằm trong nhóm xuất hiện ở cùng cấp như mục đầu tiên—tất cả ba mục trong menu đều là các mục đồng cấp. Tuy nhiên, bạn có thể sửa đổi các đặc điểm của hai mục trong nhóm bằng cách tham chiếu ID nhóm và sử dụng các phương pháp được liệt kê bên trên. Hệ thống cũng sẽ không bao giờ tách riêng các mục đã ghép nhóm. Ví dụ, nếu bạn khai báo android:showAsAction="ifRoom" cho từng mục, chúng sẽ hoặc đều xuất hiện trong thanh hành động hoặc đều xuất hiện trong phần tràn hành động.

Sử dụng mục menu có thể chọn



Hình 5. Ảnh chụp màn hình một menu con với các mục có thể chọn.

Một mục có thể có ích như một giao diện để bật và tắt các tùy chọn, bằng cách sử dụng một hộp kiểm cho các tùy chọn độc lập, hoặc nút chọn một cho các nhóm tùy chọn loại trừ lẫn nhau. Hình 5 minh họa một menu con với các mục có thể chọn bằng các nút chọn một.

Bạn có thể định nghĩa hành vi có thể chọn cho các mục menu riêng lẻ bằng cách sử dụng thuộc tính `android:checkable` trong phần tử `<item>`, hoặc cho toàn bộ nhóm với thuộc tính `android:checkableBehavior` trong phần tử `<group>`. Ví dụ, tất cả các mục trong nhóm menu này có thể chọn bằng một nút chọn một:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <group android:checkableBehavior="single">
    <item android:id="@+id/red"
          android:title="@string/red" />
    <item android:id="@+id/blue"
          android:title="@string/blue" />
  </group>
</menu>
```

Thuộc tính `android:checkableBehavior` chấp nhận hoặc:

`single`

Chỉ chọn được một mục từ nhóm (nút chọn một)

all

Có thể chọn được tất cả các mục (hộp kiểm)

none

Không chọn được mục nào

Bạn có thể áp dụng một trạng thái được chọn mặc định cho một mục bằng cách sử dụng thuộc tính android:checked trong phần tử <item> và thay đổi nó trong mã bằng phương pháp setChecked().

Khi chọn một mục có thể chọn, hệ thống sẽ gọi phương pháp gọi lại mục được chọn tương ứng (chẳng hạn như onOptionsItemSelected()). Chính ở đây bạn phải đặt trạng thái của hộp kiểm, vì hộp kiểm hay nút chọn một đều không tự động thay đổi trạng thái của nó. Bạn có thể truy vấn trạng thái hiện tại của mục (như trước khi người dùng chọn) bằng isChecked() và sau đó đặt trạng thái được chọn bằng setChecked(). Ví dụ:

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
  switch (item.getItemId()) {
    case R.id.vibrate:
    case R.id.dont_vibrate:
      if (item.isChecked()) item.setChecked(false);
      else item.setChecked(true);
      return true;
    default:
      return super.onOptionsItemSelected(item);
  }
}
```

Nếu bạn không đặt trạng thái được chọn bằng cách này, khi đó trạng thái hiển thị của mục (hộp kiểm hoặc nút chọn một) sẽ không thay đổi khi người dùng chọn nó. Khi bạn đặt trạng thái, hoạt động sẽ giữ nguyên trạng thái được chọn của mục đó để khi người dùng mở menu sau, trạng thái được chọn mà bạn đặt sẽ được hiển thị.

Thêm Mục Menu dựa trên Ý định

Đôi khi bạn sẽ muốn một mục menu khởi chạy một hoạt động bằng cách sử dụng một Intent (dù đó là một hoạt động trong ứng dụng của bạn hay một ứng dụng khác). Khi bạn biết ý định mà mình muốn sử dụng và có một mục menu cụ thể sẽ khởi tạo ý định, bạn có thể thực thi ý định bằng startActivity() trong phương pháp gọi lại phù hợp theo mục được chọn (chẳng hạn như lệnh gọi lại onOptionsItemSelected()).

Tuy nhiên, nếu bạn không chắc chắn rằng thiết bị của người dùng chứa một ứng dụng xử lý ý định đó thì việc thêm một mục menu gọi nó ra có thể dẫn đến mục menu

không hoạt động, do ý định có thể không phân giải thành một hoạt động. Để giải quyết điều này, Android cho phép bạn linh hoạt thêm các mục menu vào menu của mình khi Android tìm các hoạt động trên thiết bị để xử lý ý định của bạn.

Để thêm các mục menu dựa trên các hoạt động sẵn có mà chấp nhận ý định:

1. Định nghĩa một ý định bằng thể loại CATEGORY_ALTERNATIVE và/hoặc CATEGORY_SELECTED_ALTERNATIVE, cộng với bất kỳ yêu cầu nào khác.
2. Gọi Menu.addIntentOptions(). Sau đó, Android tìm kiếm bất kỳ ứng dụng nào có thể thực hiện ý định và thêm chúng vào menu của bạn.

Nếu không có ứng dụng được cài đặt mà thỏa mãn ý định thì không có mục menu nào được thêm vào.

Ví dụ:

```
@Override
public boolean onCreateOptionsMenu(Menu menu){
   super.onCreateOptionsMenu(menu);

   // Create an Intent that describes the requirements to fulfill, to be included
   // in our menu. The offering app must include a category value of
Intent.CATEGORY_ALTERNATIVE.
   Intent intent = new Intent(null, dataUri);
   intent.addCategory(Intent.CATEGORY_ALTERNATIVE);

   // Search and populate the menu with acceptable offering applications.
   menu.addIntentOptions(
       R.id.intent_group,  // Menu group to which new items will be added
       0,      // Unique item ID (none)
       0,      // Order for the items (none)
       this.getComponentName(),   // The current activity name
       null,   // Specific items to place first (none)
       intent, // Intent created above that describes our requirements
       0,      // Additional flags to control items (none)
       null);  // Array of MenuItems that correlate to specific items (none)

   return true;
}
```

Đối với mỗi hoạt động được tìm thấy mà cung cấp một bộ lọc ý định khớp với ý định được định nghĩa, một mục menu được thêm, bằng cách sử dụng giá trị trong android:label của bộ lọc ý định làm tiêu đề của mục menu và biểu tượng của ứng dụng làm biểu tượng của mục menu. Phương pháp addIntentOptions() trả về số mục menu được thêm.

Cho phép hoạt động của bạn được thêm vào các menu khác

Bạn cũng có thể cung cấp các dịch vụ của hoạt động của mình cho các ứng dụng khác, vì vậy ứng dụng của bạn có thể nằm trong menu của các ứng dụng khác (đảo ngược vai trò nêu trên).

Để được nằm trong menu của ứng dụng khác, bạn cần định nghĩa một bộ lọc ý định như bình thường, nhưng đảm bảo thêm các giá trị CATEGORY_ALTERNATIVE và/hoặc CATEGORY_SELECTED_ALTERNATIVE cho thể loại bộ lọc ý định. Ví dụ:

```
<intent-filter label="@string/resize_image">
    ...
    <category android:name="android.intent.category.ALTERNATIVE" />
    <category android:name="android.intent.category.SELECTED_ALTERNATIVE" />
    ...
</intent-filter>
```