ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Mobile Programing

Chapter 4.1. Button and Clickable Image

Note

- This slide is based on Google Android code labs slides
- Original slides: https://drive.google.com/drive/folders/1eu-LXxiHocSktGYpG04PfE9Xmr_pBY5P

4.1 Buttons and clickable images

Contents

- User interaction
- Buttons
- Clickable images
- Floating action button
- Common gestures

User interaction

Users expect to interact with apps

- Tapping or clicking, typing, using gestures, and talking
- Buttons perform actions
- Other UI elements enable data input and navigation

User interaction design

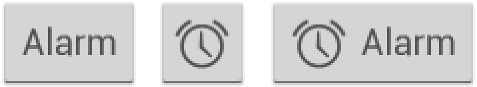
Important to be obvious, easy, and consistent:

- Think about how users will use your app
- Minimize steps
- Use UI elements that are easy to access, understand, use
- Follow Android best practices
- Meet user's expectations

Buttons

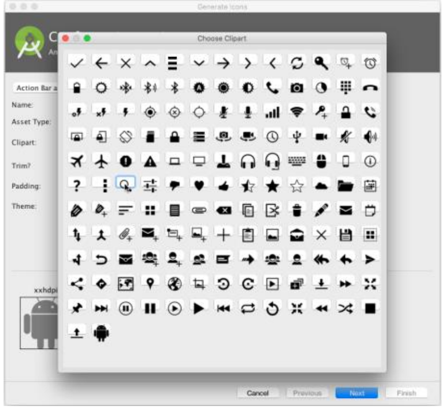
Button

- View that responds to tapping (clicking) or pressing
- Usually text or visuals indicate what will happen when tapped
- State: normal, focused, disabled, pressed, on/off



Button image assets

1. Right-click app/res/drawable
2. Choose **New > Image Asset**
3. Choose **Action Bar and Tab Items** from drop down menu
4. Click the **Clipart:** image (the Android logo)



Experiment:

2. Choose **New > Vector Asset**

Responding to button taps

- *In your code*: Use `OnClickListener` event listener.
- *In XML*: use `android:onClick` attribute in the XML layout:

```
<Button
    android:id="@+id/button_send"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_send"
    android:onClick="sendMessage" />
```

android:onClick

Setting listener with onClick callback

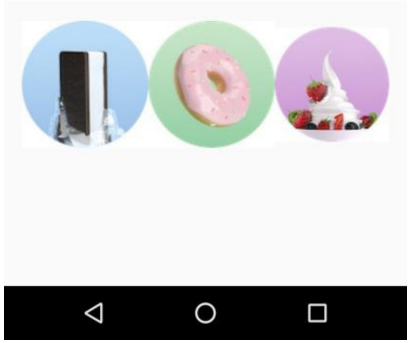
```
Button button = findViewById(R.id.button);

button.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        // Do something in response to button click
    }
});
```

Clickable images

ImageView

- `ImageView` with `android:onClick` attribute
- Image for `ImageView` in **app>src>main>res>drawable** folder in project



Responding to ImageView taps

- *In your code*: Use `OnClickListener` event listener.
- *In XML*: use `android:onClick` attribute in the XML layout:

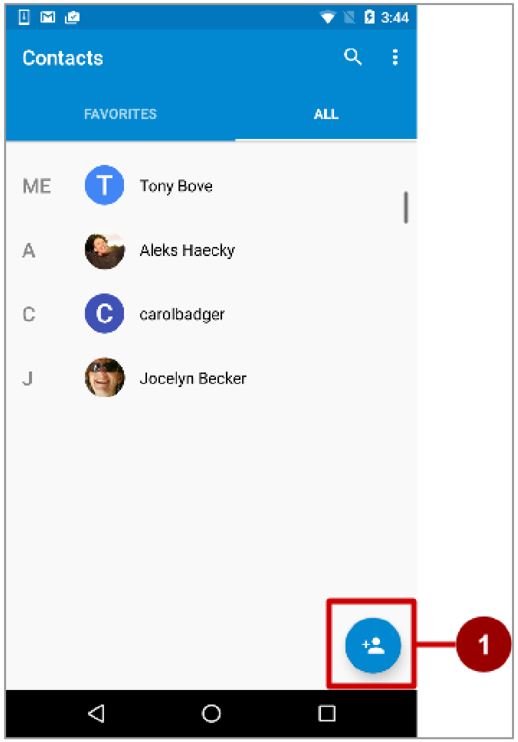
```
<ImageView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:src="@drawable/donut_circle"
      android:onClick="orderDonut"/>
```

android:onClick

Floating action button

Floating Action Buttons (FAB)

- Raised, circular, floats above layout
- Primary or "promoted" action for a screen
- One per screen



For example:

Add Contact button in Contacts app

Using FABs

- Start with Basic Activity template
- Layout:

```
<android.support.design.widget.FloatingActionButton
        android:id="@+id/fab"
        android:layout_gravity="bottom|end"
        android:layout_margin="@dimen/fab_margin"
        android:src="@drawable/ic_fab_chat_button_white"

        .../>
```

FAB size

- 56 x 56 dp by default
- Set mini size (30 x 40 dp) with `app:fabSize` attribute:
    - `app:fabSize="mini"`
- Set to 56 x 56 dp (default):
    - `app:fabSize="normal"`

Common Gestures

Touch Gestures

Touch gestures include:

- long touch
- double-tap
- fling
- drag
- scroll
- pinch

Don’t depend on touch gestures for app's basic behavior!

Detect gestures

Classes and methods are available to help you handle gestures.

- GestureDetectorCompat class for common gestures
- MotionEvent class for motion events

Detecting all types of gestures

1. Gather data about touch events.
2. Interpret the data to see if it meets the criteria for any of the gestures your app supports.

Read more about how to handle gestures in the Android developer documentation

Learn more

- Input Controls
- Drawable Resources
- Floating Action Button
- Radio Buttons
- Specifying the Input Method Type
- Handling Keyboard Input
- Text Fields
- Buttons
- Spinners
- Dialogs
- Fragments
- Input Events
- Pickers
- Using Touch Gestures
- Gestures design guide

What's Next?

- Concept Chapter: 4.1 Buttons and clickable images
- Practical: 4.1 Clickable images

END