

# ITW202: Mobile Application

## Unit IV: Developing for Android

Ms. Sonam Wangmo

Gyalpozhing College of Information Technology  
Royal University of Bhutan

April 29, 2021

# Mobile Application Development

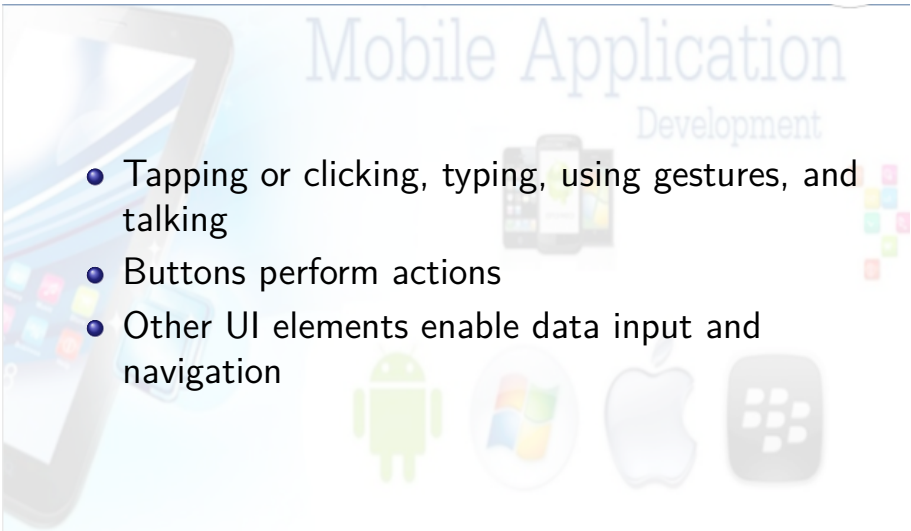
## User Interaction



# User Interaction



# Users expect to interact with apps

- 
- The background of the slide features a light blue gradient. On the left, there is a large, stylized illustration of a tablet. In the center, the words "Mobile Application Development" are written in a large, light blue, sans-serif font. Below this text, there are four circular icons representing different mobile operating systems: Android (a green robot), Windows (the four-pane logo), Apple (the silver apple logo), and BlackBerry (a grey square with a white grid). To the right of the text, there is a small cluster of colorful squares in various colors (red, yellow, green, blue, pink).
- Tapping or clicking, typing, using gestures, and talking
  - Buttons perform actions
  - Other UI elements enable data input and navigation

# User Interaction



# Button

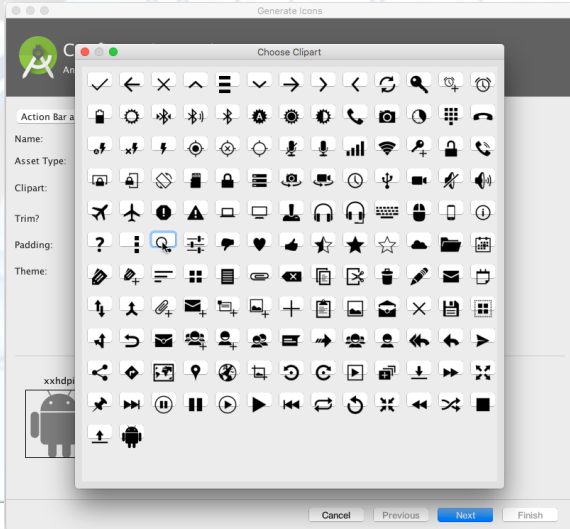
- View that responds to tapping (clicking) or pressing
- Usually text or visuals indicate what will happen when tapped
- State: normal, focused, disabled, pressed, on/off

# Button image assets

## Mobile Application Development

- 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)

# Button image assets





# 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
    }
});
```

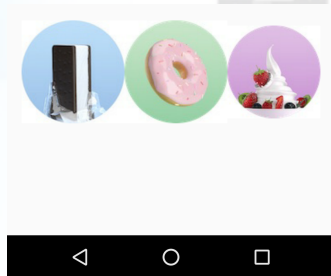
# Mobile Application Development

**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

# Mobile Application Development

## Floating action button



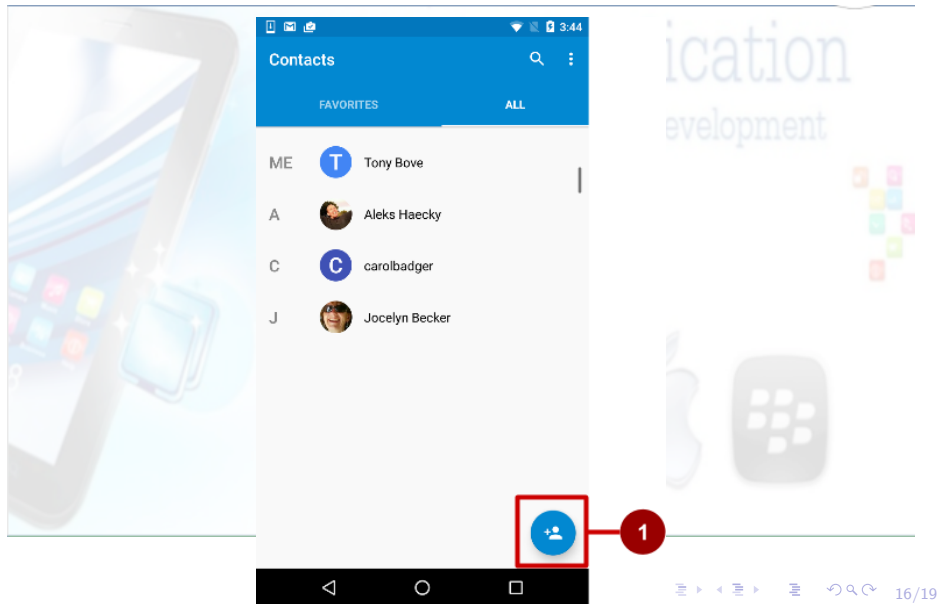
# Floating Action Buttons (FAB)

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

For example:

**Add Contact** button in Contacts app

# Floating Action Buttons (FAB)





# 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

## Mobile Application Development

- 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"

# Mobile Application Development

**THANK YOU**

