2019 级 本科

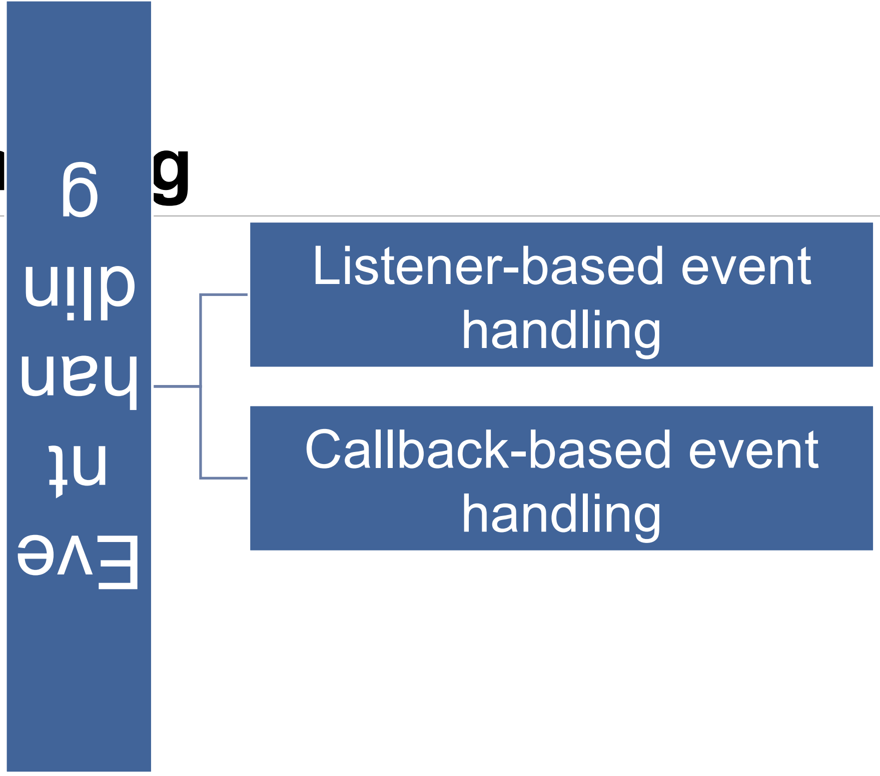
移动应用开发

张昕

zhangxin@cust.edu.cn

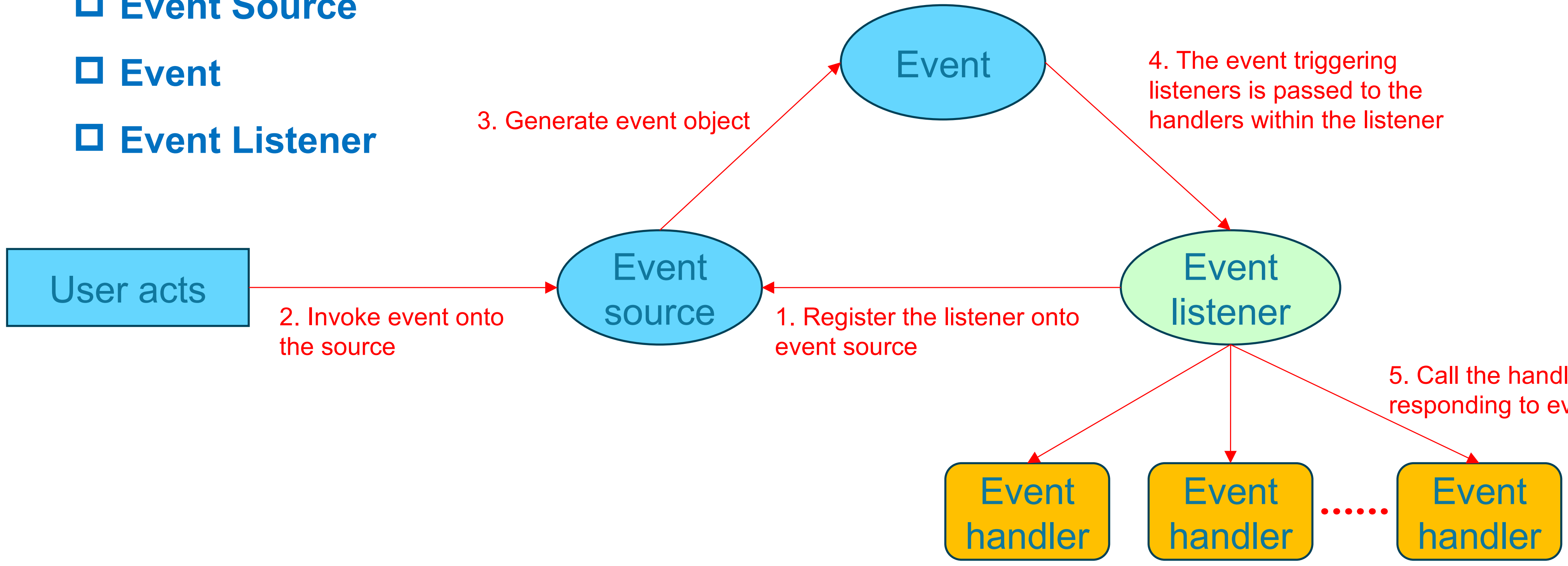
2022

Event Han g


Event handling
Listener-based event handling
Callback-based event handling

Listener-based Event Handling

- **Listener-based event handling model: Set event listener onto UI components**
  - **Three modules**
    - **Event Source**
    - **Event**
    - **Event Listener**



User acts
Event source
Event
Event listener
Event handler
Event handler
Event handler
......
1. Register the listener onto event source
2. Invoke event onto the source
3. Generate event object
4. The event triggering listeners is passed to the handlers within the listener
5. Call the handler for responding to event

Callback-based Event Handling

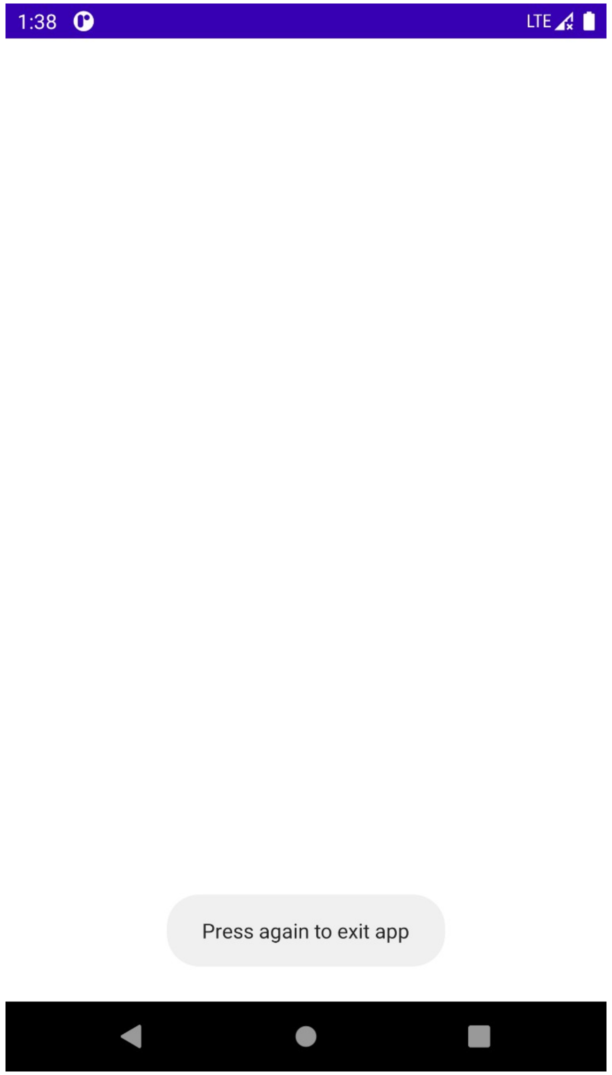
- **Override the specific callback methods of Android components or the callback methods of Activity**
  - **Means: extending GUI component class, then override the event handling methods in the class**
    - **Android designed the callback methods in all the GUI components**

```
For example, View class has the callback methods
     boolean onKeyDown(int keyCode, KeyEvent event)
     boolean onKeyLongPress(int keyCode, KeyEvent event)
     boolean onKeyShortcut(int keyCode, KeyEvent event)
     boolean onKeyUp(int keyCode, KeyEvent event)
     boolean onTouchEvent(int keyCode, KeyEvent event)
     boolean onTrackballEvent(int keyCode, KeyEvent event)
Referring to some particular events, callback-based event handling doesn’t work.
Listener-based event handling is the only choice.
```

Physical button-related Event handling

- **A standard Android device might have multiple physical buttons, e.g., volume buttons, etc.**
  - Volume Button
    - KEYCODE_VOLUME_UP
    - KEYCODE_VOLUME_DOWN
  - Back Button
    - KEYCODE_BACK
  - Menu Button
    - KEYCODE_MENU
  - When handling physical button-related events, there are usually three callback methods
    - onKeyUp()
      - When releasing the button
    - onKeyDown()
      - When pressing (not released) the button
    - onKeyLongPress()
      - When keeping pressing the button

Demo



```
public class MainActivity extends Activity {
    private long exitTime = 0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        if(keyCode == KeyEvent.KEYCODE_BACK){
            exit();
            return true;
        }
        return super.onKeyDown(keyCode, event);
    }
    public void exit(){
        if((System.currentTimeMillis() - exitTime) > 2000){
            Toast.makeText(getApplicationContext(), "Press again to exit app",
Toast.LENGTH_LONG).show();
            exitTime = System.currentTimeMillis();
        }else{
            finish();
            System.exit(0);
        }
    }
}
```

Touch screen event handling

- **Single-click event**
  - Android provides setOnClickListener() to components for listening to the potential event

```
public static interface View.OnClickListener(){
    public void onClick(View v);
}
```

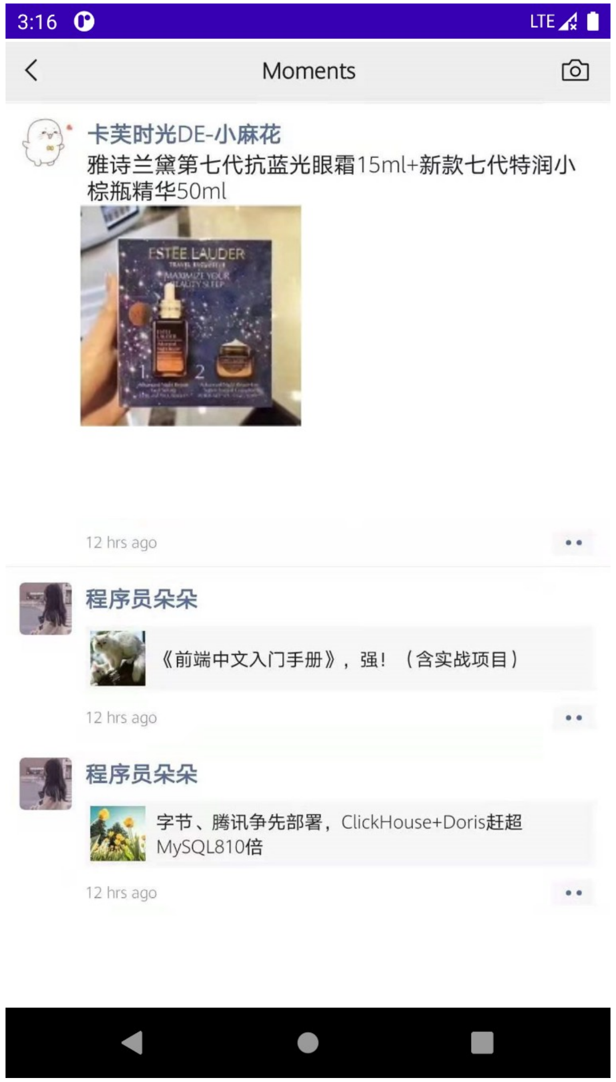
```
Example:
Button btn = new Button(this);
btn.setOnClickListener(new View.OnClickListener(){
    @override
    public void onClick(View v){
        Toast.makeText(MainActivity.this, “Button clicked”, Toast.LENGTH_SHORT).show();
    }
});
```

Touch screen event handling

- **Long-click event**
  - **Android provides setOnLongClickListener() to components for listening to the potential event**

```
public static interface View.OnLongClickListener(){
    public boolean onLongClick(View v);
}
```

Demo



卡芙时光DE-小麻花
雅诗兰黛第七代抗蓝光眼霜15ml+新款七代特润小棕瓶精华50ml

12 hrs ago

程序员朵朵
《前端中文入门手册》，强！（含实战项目）

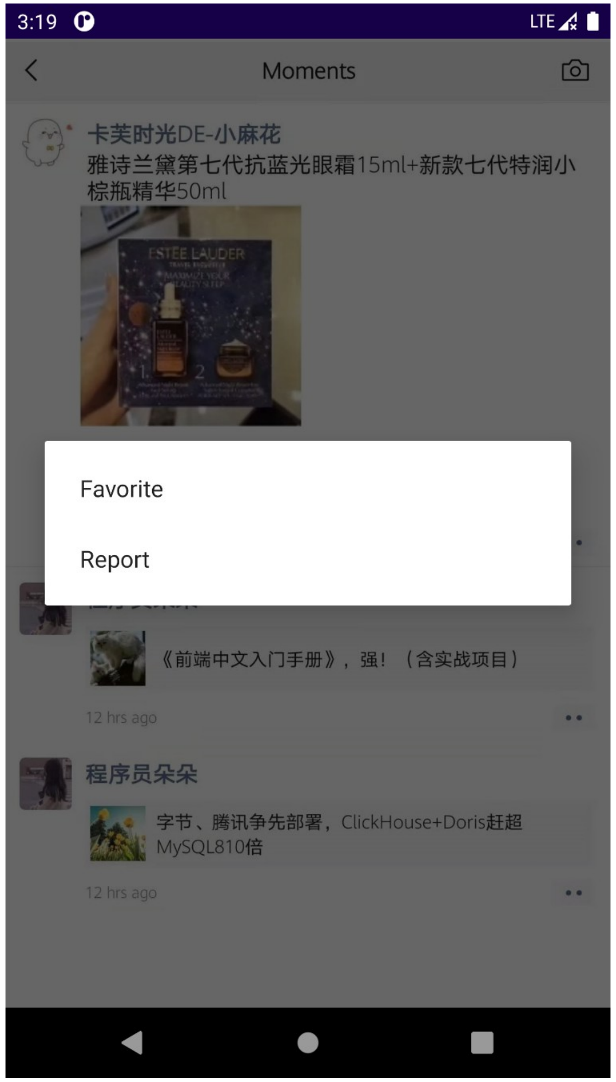
12 hrs ago

程序员朵朵
字节、腾讯争先部署，ClickHouse+Doris赶超MySQL810倍

12 hrs ago

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.longpressevent.MainActivity">
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="110dp"
        android:layout_marginTop="2dp"
        android:src="@drawable/wei_top"/>
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="50dp"
        android:src="@drawable/ico_144_144a"/>
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/wei_down"/>
</LinearLayout>
```

Demo


3:19 LTE

Moments

卡芙时光DE-小麻花
雅诗兰黛第七代抗蓝光眼霜15ml+新款七代特润小棕瓶精华50ml

Favorite

Report

《前端中文入门手册》，强！（含实战项目）

12 hrs ago

程序员朵朵

字节、腾讯争先部署，ClickHouse+Doris赶超MySQL810倍

12 hrs ago

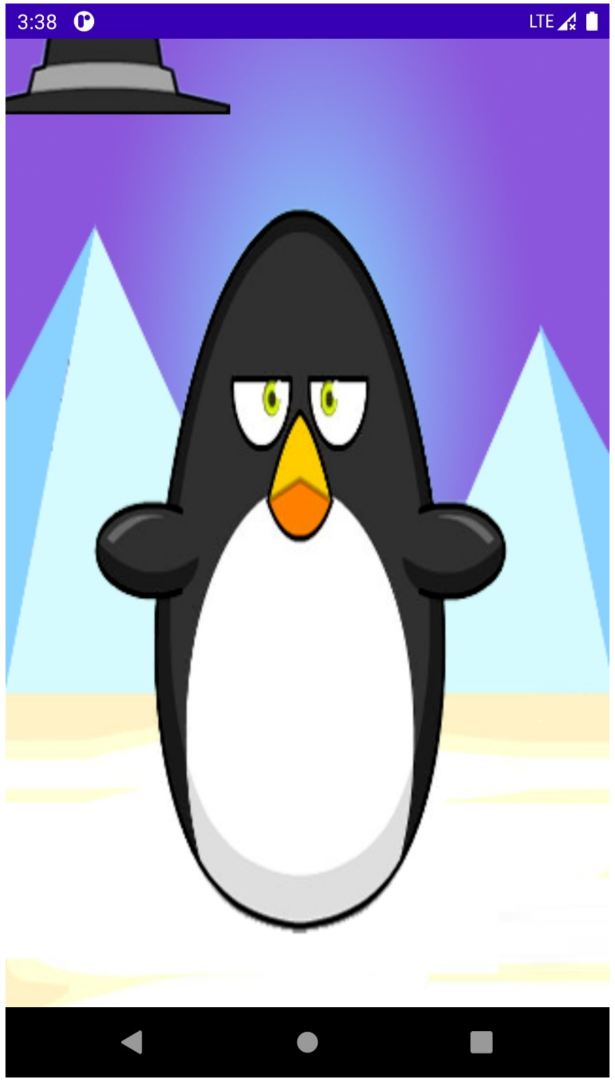
```java
public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ImageView imageView = (ImageView)findViewById(R.id.imageView);
        imageView.setOnLongClickListener(new View.OnLongClickListener(){
            @Override
            public boolean onLongClick(View v) {
                registerForContextMenu(v);
                openContextMenu(v);
                return true;
            }
        });
    }
    @Override
    public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenu.ContextMenuInfo menuInfo) {
        super.onCreateContextMenu(menu, v, menuInfo);
        menu.add("Favorite");
        menu.add("Report");
    }
}
```

Touch screen event handling

- Touch screen event
  - Android provides setOnTouchListener() to components for listening to the potential event

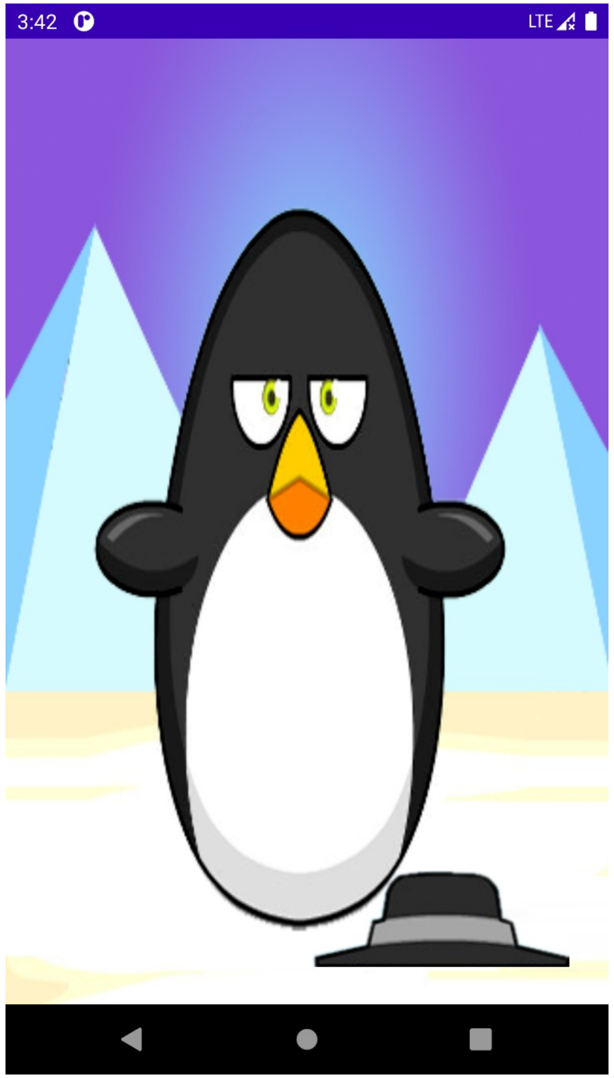
```
public static interface View.OnTouchListener(){
    public abstract boolean onTouch(View v, MotionEvent event);
}
```

Demo



```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/relativeLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background"
    tools:context="com.example.touchevents.MainActivity">
</RelativeLayout>
```

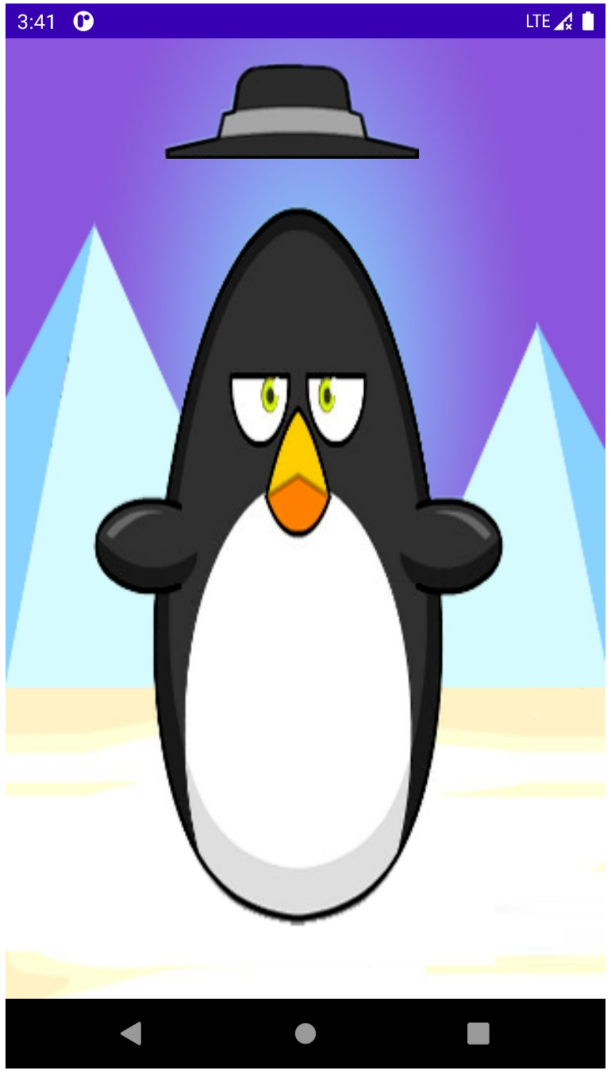
Demo



```
public class HatView extends View {
    public float bitmapX;
    public float bitmapY;
    public HatView(Context context){
        super(context);
        bitmapX = 65;
        bitmapY = 0;
    }
    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
        Paint paint = new Paint();
        Bitmap bitmap =
                BitmapFactory.decodeResource(this.getResources(),

                              R.drawable.hat);
        canvas.drawBitmap(bitmap,bitmapX,bitmapY,paint);
        if(bitmap.isRecycled()){
            bitmap.recycle();
        }
    }
}
```

Demo



```
public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        RelativeLayout relativeLayout =
                       (RelativeLayout)findViewById(R.id.relativeLayout);
        final HatView hat = new HatView(MainActivity.this);
        hat.setOnTouchListener(new View.OnTouchListener(){
            @Override
            public boolean onTouch(View v, MotionEvent event) {
                hat.bitmapX = event.getX()-80;
                hat.bitmapY = event.getY()-50;
                hat.invalidate();
                return true;
            }
        });
        relativeLayout.addView(hat);
    }
}
```

Gesture Detection

- Android provides GestureDetector class for detecting gestures
  - When creating GestureDector, a GestureDetector.OnGestureListener object that represents a listener to respond to the gestures
  - GestureDetector.OnGestureListener contains event handling methods:

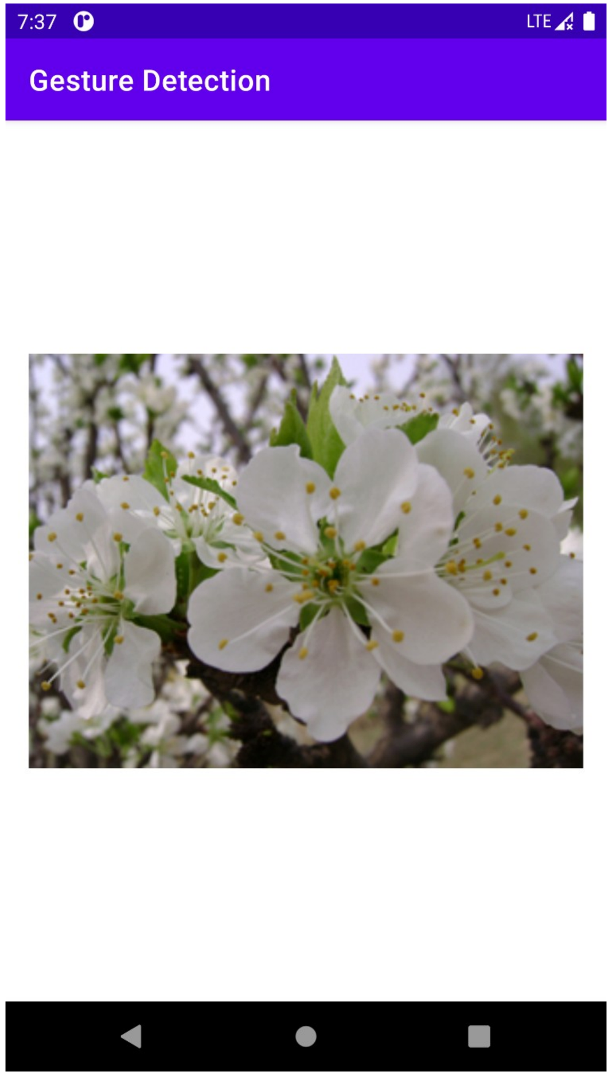
```
boolean onDown(MotionEvent e)
    when touching the screen
boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY)
    when touching the screen and sliding the finger
abstract void onLongPress(MotionEvent e)
    when long touching the screen
boolean onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY)
    when touching the screen and continuously sliding towards up or down
void onShowPress(MotionEvent e)
    when touching the screen without any movement and without releasing
boolean onSingleTapUp(MotionEvent e)
    when tapping up the screen
```

Gesture Detection

- **Procedures**
  - Create a GestureDetector object and implement the GestureDetector.OnGestureListener instance
  - Bind listeners onto TouchEvent and send TouchEvent to GestureDetector of Activity

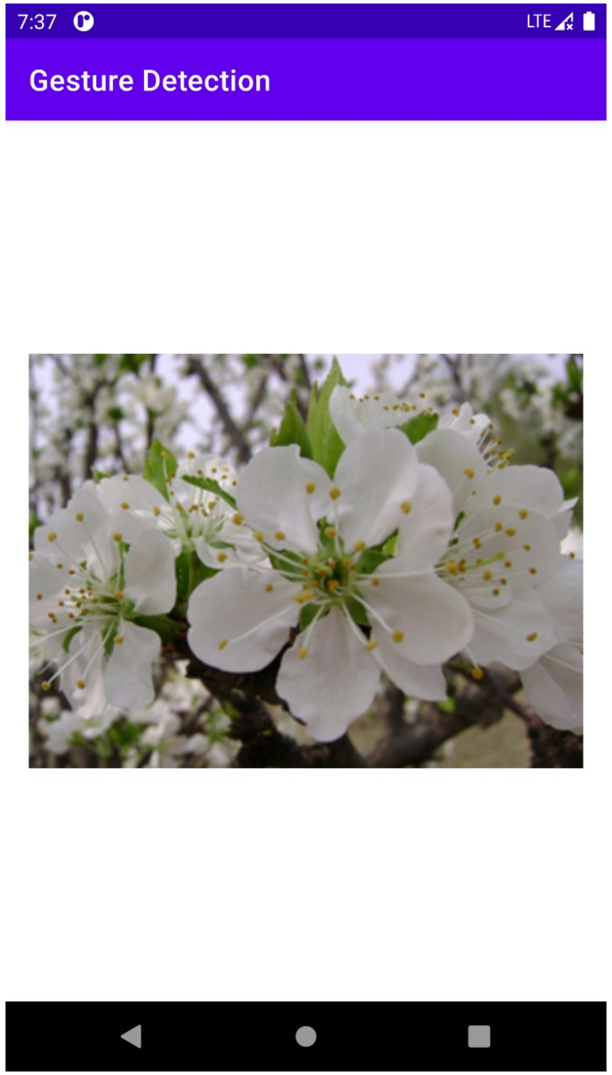
```
boolean onDown(MotionEvent e)
    when touching the screen
boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY)
    when touching the screen and sliding the finger
abstract void onLongPress(MotionEvent e)
    when long touching the screen
boolean onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY)
    when touching the screen and continuously sliding towards up or down
void onShowPress(MotionEvent e)
    when touching the screen without any movement and without releasing
boolean onSingleTapUp(MotionEvent e)
    when tapping up the screen
```

Demo



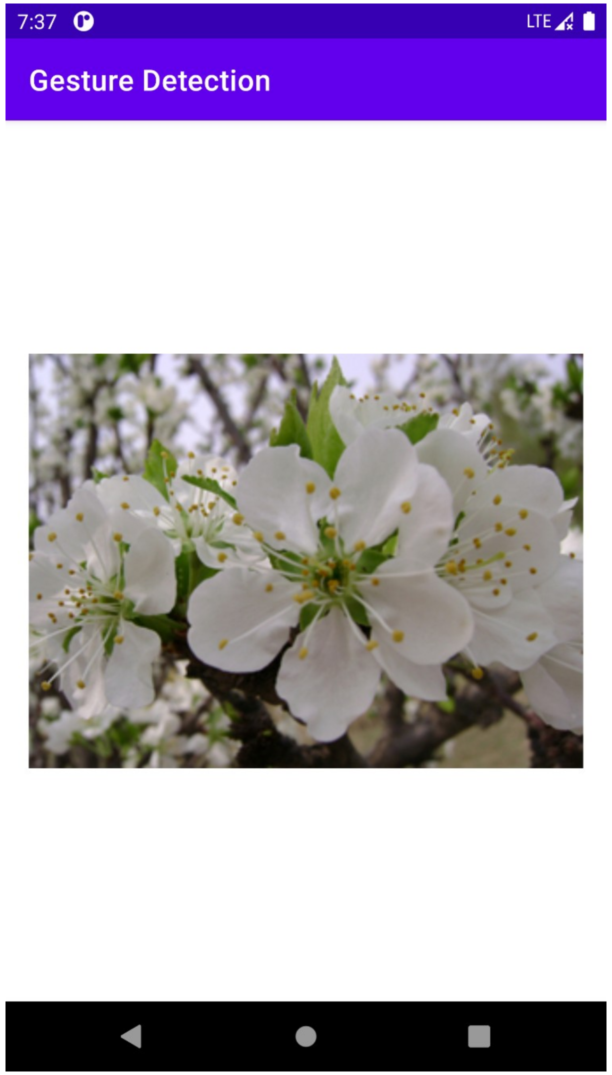
```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.gesturedetection.MainActivity">
    <ViewFlipper
        android:id="@+id/flipper"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </ViewFlipper>
</RelativeLayout>
```

Demo



```
public class MainActivity extends AppCompatActivity implements GestureDetector.OnGestureListener {
    ViewFlipper flipper;
    GestureDetector detector;
    Animation[] animation = new Animation[4];
    final int distance = 50;
    private int[] images = new int[]{R.drawable.img01,R.drawable.img02,R.drawable.img03,
            R.drawable.img04,R.drawable.img05,R.drawable.img06,R.drawable.img07,
            R.drawable.img08,R.drawable.img09};
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        detector = new GestureDetector(this,this);
        flipper = (ViewFlipper)findViewById(R.id.flipper);
        for(int i = 0; i < images.length; i++){
            ImageView imageView = new ImageView(this);
            imageView.setImageResource(images[i]);
            flipper.addView(imageView);
        }
        animation[0] = AnimationUtils.loadAnimation(this,R.anim.slide_in_left);
        animation[1] = AnimationUtils.loadAnimation(this,R.anim.slide_out_left);
        animation[2] = AnimationUtils.loadAnimation(this,R.anim.slide_in_right);
        animation[3] = AnimationUtils.loadAnimation(this,R.anim.slide_out_right);
    }
```

Demo



```java
    @Override
    public boolean onDown(MotionEvent e) {
        return false;
    }
    @Override
    public void onShowPress(MotionEvent e) {
    }
    @Override
    public boolean onSingleTapUp(MotionEvent e) {
        return false;
    }
    @Override
    public boolean onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY) {
        return false;
    }
    @Override
    public void onLongPress(MotionEvent e) {
    }
    @Override
    public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY) {
        if(e1.getX() - e2.getX() > distance){
            flipper.setInAnimation(animation[2]);
            flipper.setOutAnimation(animation[1]);
            flipper.showPrevious();
            return true;
        }else if(e2.getX() - e1.getX() > distance){
            flipper.setInAnimation(animation[0]);
            flipper.setOutAnimation(animation[3]);
            flipper.showNext();
            return true;
        }
        return false;
    }
    @Override
    public boolean onTouchEvent(MotionEvent event) {
        return detector.onTouchEvent(event);
    }
}
```