



LẬP TRÌNH ANDROID

Giảng Viên: Phùng Mạnh Dương
Trường ĐH Công nghệ, ĐHQGHN



CHƯƠNG 5

CĂN BẢN VỀ GIAO DIỆN

- ☐ View & View Events
- ☐ View Group & AdapterViews
- ☐ Layout
- ☐ Menu
- ☐ Dialog

View

- ❑ **Khối cơ bản** cho các thành phần giao diện.
- ❑ Chiếm một vùng không gian **hình chữ nhật** trên màn hình.
- ❑ Phụ trách việc tự **vẽ** thành phần giao diện cũng như **xử lý sự kiện** liên quan đến thành phần đó.
- ❑ Một số View điển hình:
 - Button
 - ToogleButton
 - CheckBox
 - RatingBar
 - AutoCompleteTextView

Button

- ❑ Thành phần giao diện cho phép **click** để thực hiện một hành động nào đó

- ❑ VD: UIButton

```
<Button
    android:id="@+id/button"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_marginLeft="10dip"
    android:text="@string/press_me_string" >
</Button>
```

```
// Get a reference to the Press Me Button
final Button button = (Button) findViewById(R.id.button);

// Set an OnClickListener on this Button
// Called each time the user clicks the Button
button.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {

        // Maintain a count of user presses
        // Display count as text on the Button
        button.setText("Got Pressed:" + ++count);

    }
});
```

ToggleButton

- ❑ Một nút bấm **2 trạng thái** thể hiện được nhấn hoặc không.

- ❑ VD: UIToggleButton

```
<ToggleButton
    android:id="@+id/togglebutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textOff="@string/start_string"
    android:textOn="@string/stop_string"
    android:textSize="24sp" />
```

```
final LinearLayout bg = (LinearLayout) findViewById(R.id.linearlayout);

// Get a reference to the ToggleButton
final ToggleButton button = (ToggleButton) findViewById(R.id.togglebutton);

// Set an OnClickListener on the ToggleButton
button.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {

        // Toggle the Background color between a light and dark color
        if (button.isChecked()) {
            bg.setBackgroundColor(0xFFF3F3F3);
        } else {
            bg.setBackgroundColor(0xFF000000);
        }
    }
});
```

CheckBox

☐ Ô chọn 2 trạng thái.

☐ VD: UICheckBox

```
<CheckBox
    android:id="@+id/checkbox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/im_not_checked_string"
    android:textSize="24sp"/>
```

```
// Get a reference to the CheckBox
final CheckBox checkbox = (CheckBox) findViewById(R.id.checkbox);

// Set an OnClickListener on the CheckBox
checkbox.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {

        // Check whether CheckBox is currently checked
        // Set CheckBox text accordingly
        if (checkbox.isChecked()) {
            checkbox.setText("I'm checked");
        } else {
            checkbox.setText("I'm not checked");
        }
    }
});
```

RatingBar

- ❑ Thành phần giao diện bao gồm 1 hàng các ngôi sao.
- ❑ Người dùng có thể click hoặc di qua các ngôi sao để chọn.
- ❑ VD: UIRatingBar

```
<RatingBar
    android:id="@+id/ratingbar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:numStars="4"
    android:stepSize="1.0" >
</RatingBar>
```

```
final RatingBar bar = (RatingBar) findViewById(R.id.ratingbar);

bar.setOnRatingBarChangeListener(new OnRatingBarChangeListener() {

    // Called when the user swipes the RatingBar
    @Override
    public void onRatingChanged(RatingBar ratingBar, float rating, boolean fromUser) {
        tv.setText("Rating:" + rating);
    }
});
```

AutoCompleteTextView

- ❑ Một trường text cho phép người dùng nhập đồng thời **tự động gợi ý** theo kí tự nhập.
- ❑ VD: UIAutoCompleteTextView (Upload)

```
<AutoCompleteTextView
    android:id="@+id/autocomplete_country"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="5dp"
    android:textSize="24sp"/>
```

```
// Get a reference to the AutoCompleteTextView
AutoCompleteTextView textView = (AutoCompleteTextView) findViewById(R.id.autocomplete_country);

// Create an ArrayAdapter containing country names
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
    R.layout.list_item, COUNTRIES);

// Set the adapter for the AutoCompleteTextView
textView.setAdapter(adapter);
```


Các hoạt động View cơ bản

- ❑ Thiết lập **hiển thị** hoặc **ẩn** View (setVisibility)
- ❑ Thiết lập trạng thái (setChecked)
- ❑ Thiết lập **lắng nghe** và xử lý sự kiện (setOnClickListener).
- ❑ Thiết lập **thuộc tính**: Độ trong suốt, màu nền, kích thước ...
- ❑ Cho phép **focus** hoặc yêu cầu focus cho view.

Xử lý sự kiện

- ❑ Nguồn sự kiện:
 - Chạm
 - Bàn phím, trackball, D-Pad
 - Sinh ra bởi Hệ thống như thay đổi lifecycle
- ❑ Xử lý sự kiện: thông qua giao diện lắng nghe sự kiện Listener. Phổ biến:
 - `OnClickListener.onClick()`
 - `OnLongClickListener.onLongClick()`
 - `OnFocusChangeListener.onFocusChange()`
 - `OnKeyListener.onKey()`

Đơn vị đo sử dụng trong thiết kế giao diện

- ❑ Đơn vị đo sử dụng trong thiết kế giao diện:
 - dp: Density-independent pixel (=1px với màn hình 160 dpi)
 - sp: Scale-independent pixel
 - pt: Point (= 1/72 inch)
 - px: pixel (không nên dùng)
- ❑ Android định nghĩa 6 loại mật độ màn hình:
 - *ldpi* (low) ~120dpi
 - *mdpi* (medium) ~160dpi
 - *hdpi* (high) ~240dpi
 - *xhdpi* (extra-high) ~320dpi
 - *xxhdpi* (extra-extra-high) ~480dpi
 - *xxxhdpi* (extra-extra-extra-high) ~640dpi



ViewGroup

- ❑ Được sử dụng để nhóm và tổ chức một nhóm các View.
- ❑ Một số ViewGroup được định nghĩa:
 - RadioGroup
 - TimePicker
 - DatePicker
 - WebView
 - MapView
 - Gallery
 - Spinner

RadioGroup

- ❑ Bao gồm một tập các nút Radio.
- ❑ Tại một thời điểm **chỉ một nút có thể được chọn**.
- ❑ VD: UIRadioGroup (SV Tự làm)

```
<RadioGroup
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
>

<RadioButton
    android:id="@+id/choice1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/choice_1_string"
    android:textSize="24sp"/>

<RadioButton
    android:id="@+id/choice2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/choice_2_string"
    android:textSize="24sp"/>

<RadioButton
    android:id="@+id/choice3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/choice_3_string"
    android:textSize="24sp" />
</RadioGroup>

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    final TextView tv = (TextView) findViewById(R.id.textView);

    // Define a generic listener for all three RadioButtons in the RadioGroup
    final OnClickListener radioListener = new OnClickListener() {
        @Override
        public void onClick(View v) {
            RadioButton rb = (RadioButton) v;
            tv.setText(rb.getText() + " chosen");
        }
    };

    final RadioButton choice1 = (RadioButton) findViewById(R.id.choice1);
    // Called when RadioButton choice1 is clicked
    choice1.setOnClickListener(radioListener);

    final RadioButton choice2 = (RadioButton) findViewById(R.id.choice2);
    // Called when RadioButton choice2 is clicked
    choice2.setOnClickListener(radioListener);

    final RadioButton choice3 = (RadioButton) findViewById(R.id.choice3);
    // Called when RadioButton choice3 is clicked
    choice3.setOnClickListener(radioListener);
}
```

TimePicker

- ❑ Cho phép người dùng lựa chọn thời gian
- ❑ VD: `UITimePicker`

```
<TextView
    android:id="@+id/timeDisplay"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=""
    android:textSize="24sp" />

<Button
    android:id="@+id/pickTime"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/change_the_time_string"
    android:textSize="24sp" />
```

```
// The callback received when the user "sets" the time in the dialog
private TimePickerDialog.OnTimeSetListener mTimeSetListener = new TimePickerDialog.OnTimeSetListener() {
    public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
        mHour = hourOfDay;
        mMinute = minute;
        updateDisplay();
    }
};

@Override
protected Dialog onCreateDialog(int id) {
    switch (id) {
        case TIME_DIALOG_ID:
            return new TimePickerDialog(this, mTimeSetListener, mHour, mMinute,
                false);
    }
    return null;
}
```

DatePicker

□ Cho phép người dùng lựa chọn ngày tháng

□ VD: UIDatePicker

```
<TextView
    android:id="@+id/dateDisplay"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=""
    android:textSize="24sp" />

<Button
    android:id="@+id/pickDate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/change_the_date_string"
    android:textSize="24sp" />
```

```
// The callback received when the user "sets" the date in the Dialog
private DatePickerDialog.OnDateSetListener mDateSetListener = new DatePickerDialog.OnDateSetListener() {

    public void onDateSet(DatePicker view, int year, int monthOfYear,
        int dayOfMonth) {
        mYear = year;
        mMonth = monthOfYear;
        mDay = dayOfMonth;
        updateDisplay();
    }
};

// Create and return DatePickerDialog
@Override
protected Dialog onCreateDialog(int id) {
    switch (id) {
        case DATE_DIALOG_ID:
            return new DatePickerDialog(this, mDateSetListener, mYear, mMonth,
                mDay);
    }
    return null;
}
```

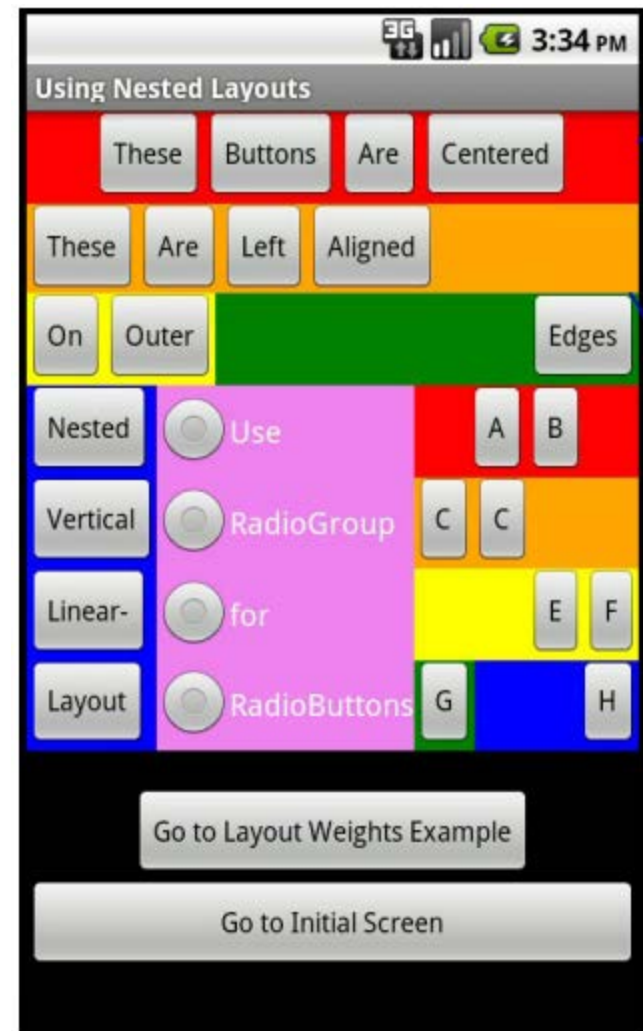
Layout

Layout

- ❑ Layout quy định cách thức hiển thị các thành phần giao diện.
- ❑ Một số loại layout:
 - LinearLayout
 - RelativeLayout
 - TableLayout
 - GridView
 - FrameLayout
 - ~~AbsoluteLayout~~

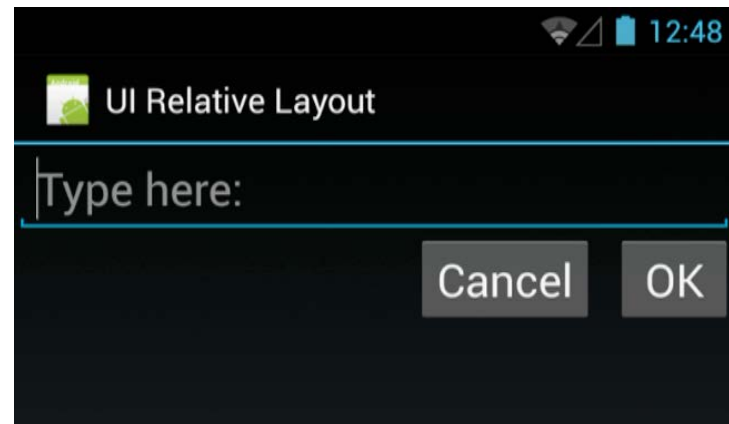
LinearLayout

- ❑ Đặt các thành phần theo 1 dòng hay 1 cột
- ❑ Có thể lồng ghép để tạo thành các khối dòng và cột kết hợp.
- ❑ VD: `UILinearLayout`



RelativeLayout

- ❑ Cho phép chỉ định vị trí của một thành phần so với thành phần khác.
- ❑ Việc chỉ định vị trí tương đối được thực hiện qua ViewID.
- ❑ Các cách chỉ định bao gồm:
 - `layout_alignParentTop`
 - `layout_alignParentLeft`
 - `layout_alignLeft`
 - `layout_alignRight`
 - `layout_below`
 - `layout_centerHorizontal`
- ❑ VD: `UIRelativeLayout`



TableLayout

- ❑ Nhóm các thành phần thành hàng và cột.
- ❑ Tương tự như HTML table, số dòng và cột xác định tự động, không cần phải khai báo.
- ❑ Các view được đặt bên trong TableRow
- ❑ VD: UI_TableLayout

