

LAB 1. ỨNG DỤNG ANDROID ĐẦU TIÊN

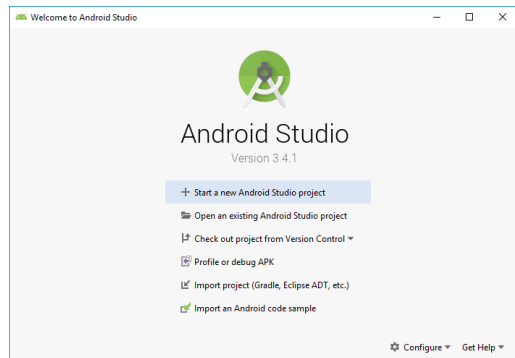
1. MỤC TIÊU:

- Cài đặt môi trường làm việc Android Studio.
- Hiểu cấu trúc cơ bản của Android project.
- Cài đặt máy ảo Genymotion.
- Chạy ứng dụng Android đầu tiên.

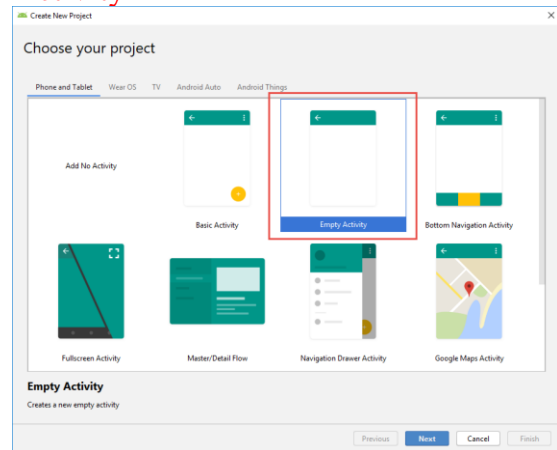
2. THỰC HIỆN:

2.1. Bài 1: Ứng dụng Android đầu tiên.

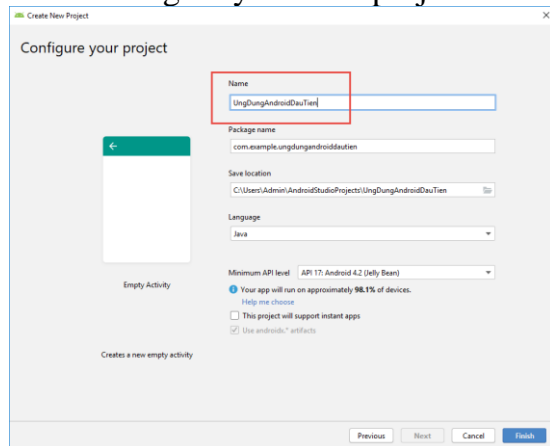
B1 – Start a new Android Studio Project



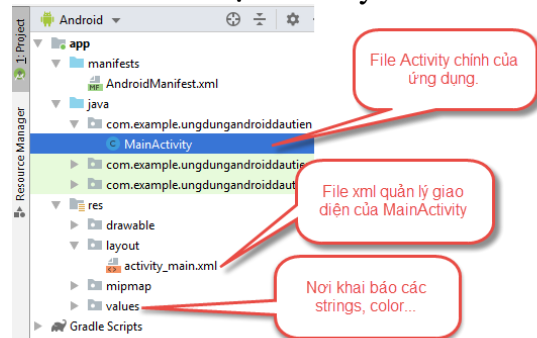
B2 – Choose your project → Empty Activity



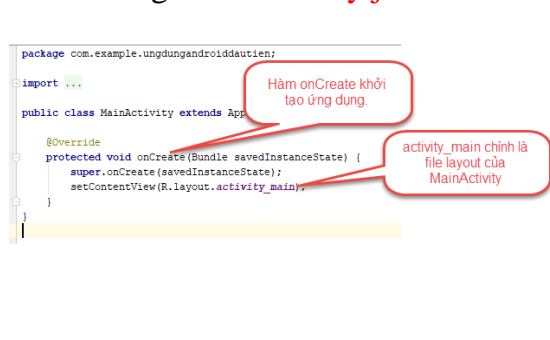
B3 – Configure your new project



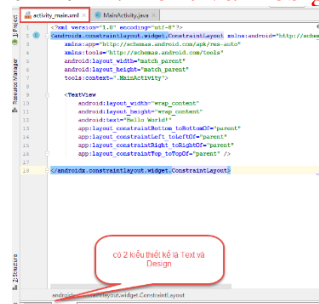
B4 – Các thư mục chính yếu



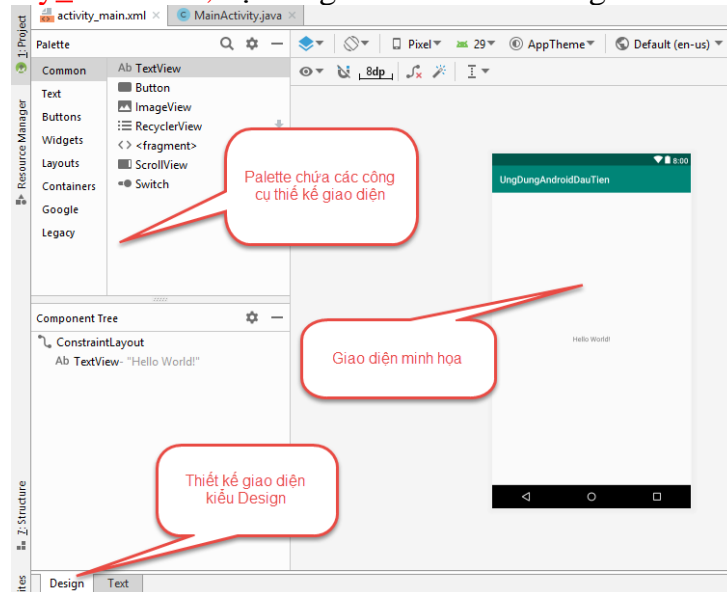
B5 – Trong MainActivity.java



B6 – Trong activity_main.xml, có 2 kiểu thiết kế: Text và Design



B7 - Trong **activity_main.xml**, bật sang kiểu thiết kế Design:



B8 - Trong **activity_main.xml**, bật sang kiểu thiết kế **Text** → sửa Layout của **activity_main.xml** thành **LinearLayout**



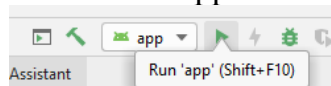
B9 – Trong **activity_main.xml**, xóa phần TextView có sẵn để thay thế bằng một TextView tự thiết kế.



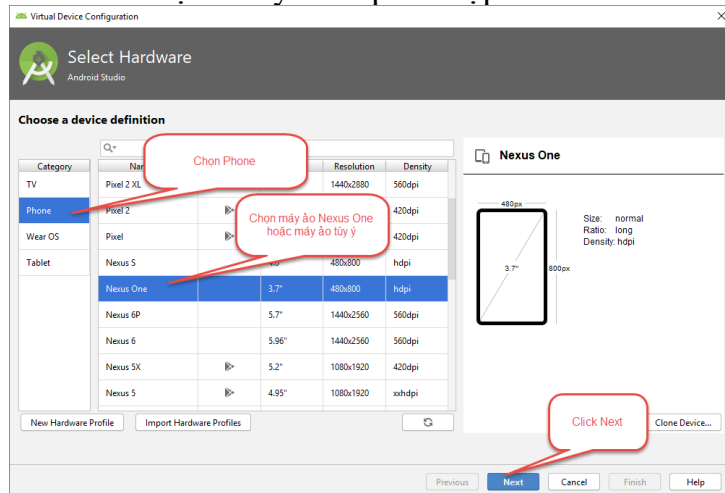
B10 – Trong `activity_main.xml`, thiết kế một `TextView` có nội dung: “**Đây là ứng dụng android đầu tiên của tôi!**”, cỡ chữ **30sp**, màu **đỏ**, canh giữa chữ và canh chữ nằm giữa layout.
→ Cập nhật đoạn code sau.

```
<TextView
    android:text="Đây là ứng dụng Android đầu tiên của tôi!"
    android:textSize="30sp"
    android:textColor="@android:color/holo_red_light"
    android:textAlignment="center"
    android:layout_gravity="center"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```

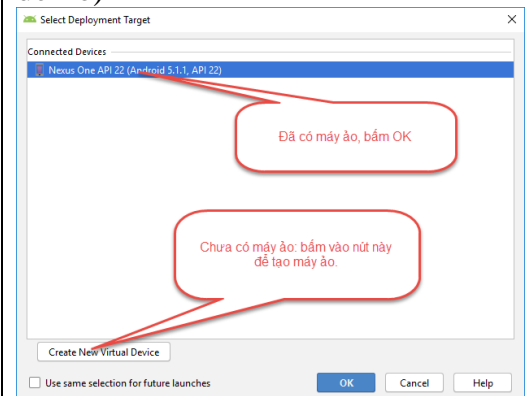
B11 – Run ‘app’



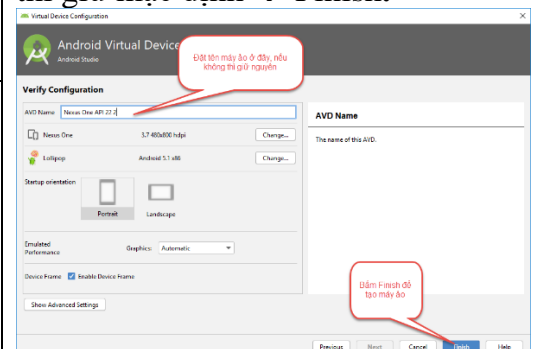
B13 – Chọn máy ảo phù hợp → bấm Next



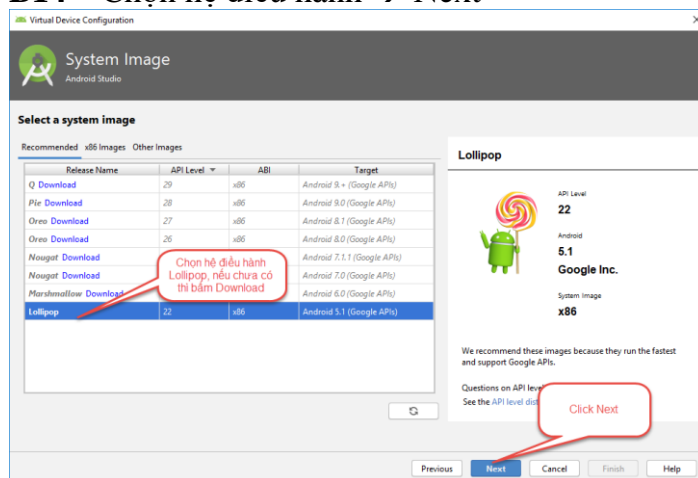
B12 – Kiểm tra xem đã có máy ảo chưa, nếu chưa có → Bấm vào nút **Create New Virtual Device** để tạo máy ảo. (Chỉ tạo máy ảo 1 lần, những lần sau sẽ dùng lại máy ảo này để demo)



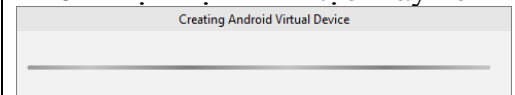
B15 – Đặt tên cho máy ảo, nếu không thì giữ mặc định → Finish.

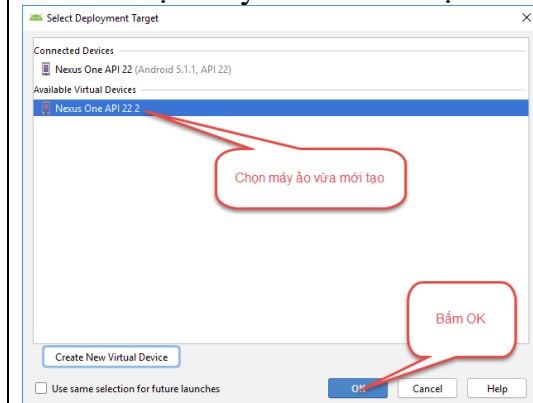
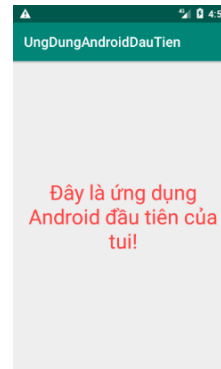


B14 – Chọn hệ điều hành → Next



B16 – Đợi một lát để tạo máy ảo

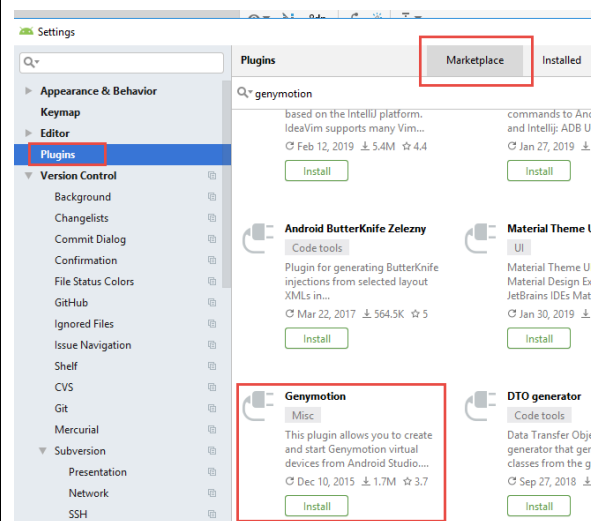


B17 – Chọn máy ảo vừa mới tạo → OK**B18 – Kiểm tra****2.2. Bài 2: Chương trình giả lập máy ảo Genymotion****B1 – Đăng ký và cài đặt Genymotion**

- Truy cập địa chỉ: <https://www.genymotion.com/account/create/>
- Điền thông tin và nhấn Create Account.
- Check mail để kích hoạt tài khoản Genymotion.
- Tải Genymotion bản Personal with VirtualBox.
- Cài đặt Genymotion.

B2 – Cài đặt plugin Genymotion

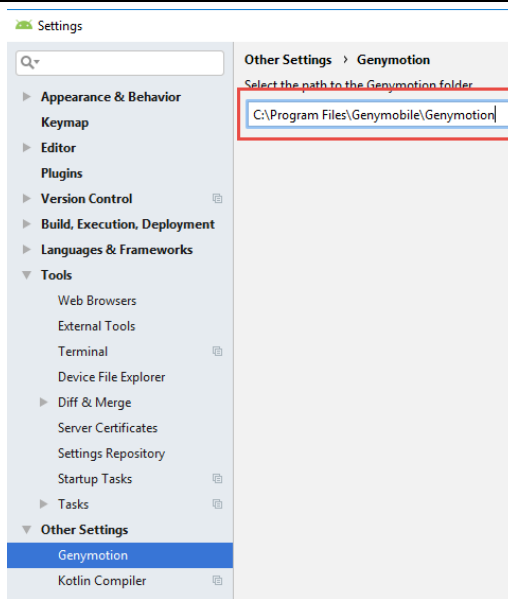
Trong Android Studio → File → Settings → Plugins → Marketplace → Tìm “genymotion” → Install → Restart IDE.

**B4 – Cài đặt điện thoại ảo trong Genymotion.**

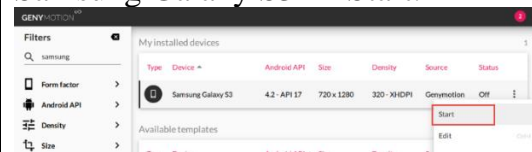
Tìm “samsung galaxy” → Samsung Galaxy S3 → Install.



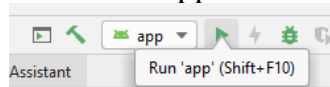
B3 – Chọn đường dẫn cho Plugin Genymotion. Trong Android Studio → File → Settings → Other Settings → Genymotion → Browse tới đường dẫn của Genymotion.



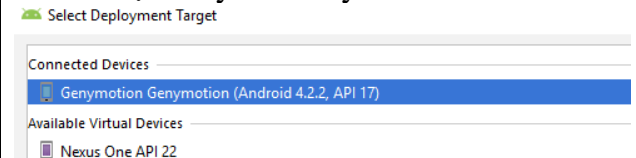
B5 – Khởi động máy ảo Genymotion. Samsung Galaxy S3 → Start.



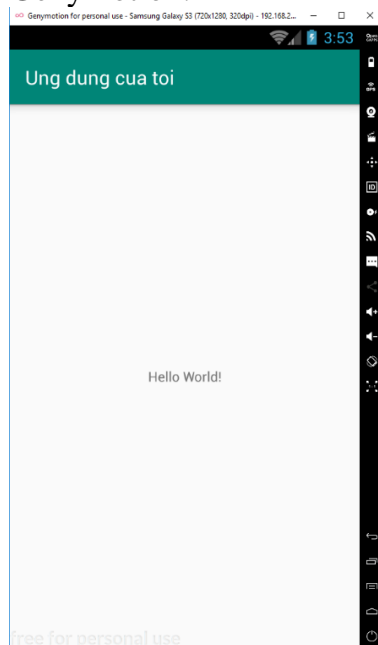
B6 – Run ‘app’



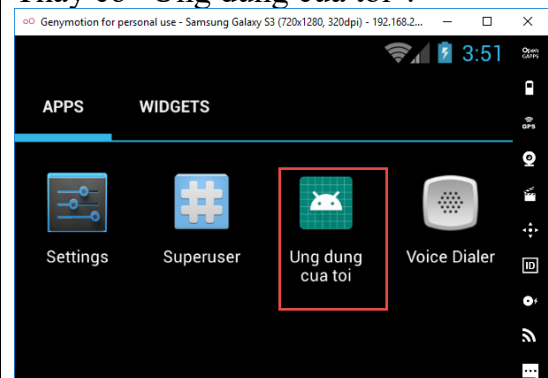
B7 – Chọn máy ảo Genymotion



B8 – Quan sát màn hình kết quả trên máy ảo Genymotion.



B9 – Quan sát màn hình ứng dụng máy ảo Genymotion. Thấy có “Ung dung của toi”.



2.3. Bài 3: Ứng dụng Vysor – phần mềm Ảnh xạ điện thoại vào máy tính

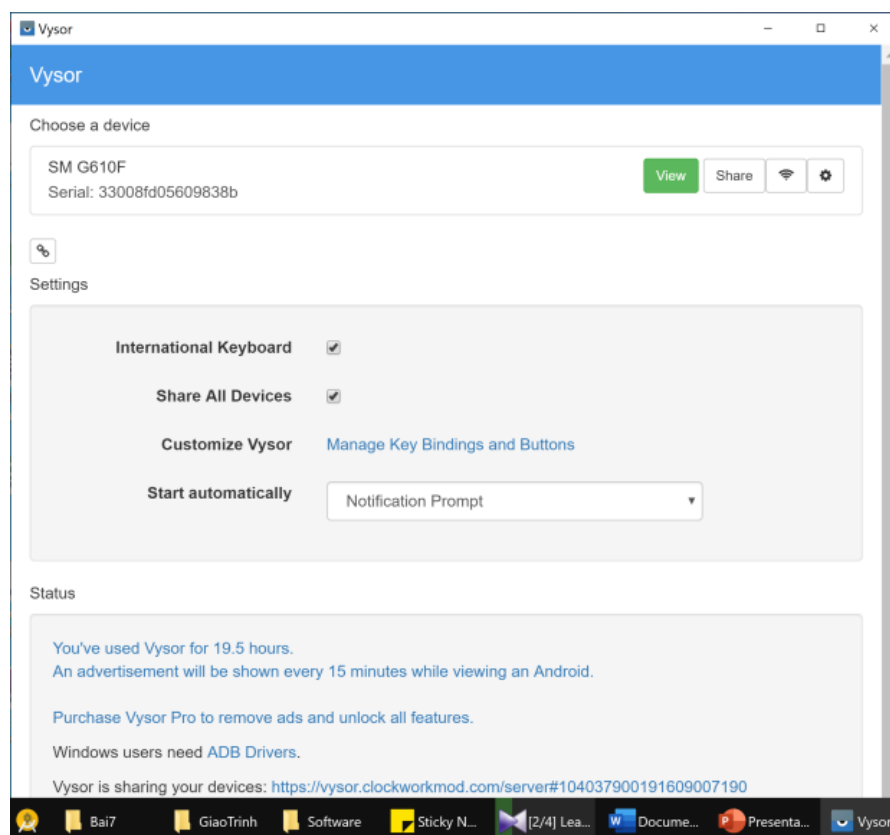
Phần mềm Vysor rất hay, nó cho phép chúng ta tương tác 2 chiều. Có thể từ Vysor trong máy tính điều khiển điện thoại và ngược lại.

Rất tiện lợi cho Giảng Viên trình chiếu giảng dạy, các bạn Sinh Viên trình chiếu báo cáo đồ án, các team work làm việc nhóm, triển khai dự án.

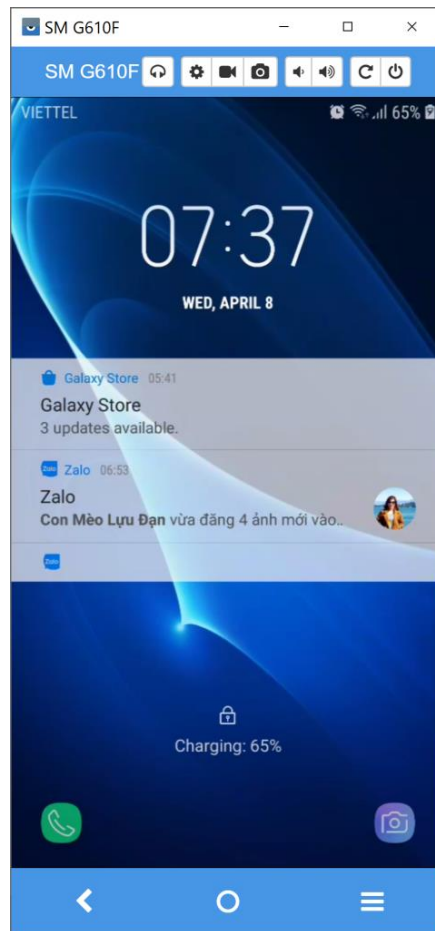
| | |
|--|---|
| Bước 1: Vào https://www.vysor.io/ | Bước 2: Chọn phiên bản để download:  |
|--|---|

Nhấn Download, chương trình sẽ hỏi phiên bản cài đặt.

Tải xong thì bấm vào nó để cài đặt, vô cùng đơn giản, bấm rồi chờ xít là xong, Vysor chạy lên và ta sẽ thấy các thiết bị điện thoại sẽ được hiển thị như hình bên dưới, dĩ nhiên là chỉ điện thoại nào đã cấu hình USB Debugging.



Ta muốn Ảnh xạ điện thoại nào vào máy tính thì chọn View ở thiết bị đó, kết quả:



Tới đây ta có thể điều khiển điện thoại ngay trên Máy tính của mình, có thể trình chiếu... rất hay. Xem màn hình toàn cảnh:



3. Bài tập:

Bài 1: Trình bày cách tạo máy ảo Android Emulator

Bài 2: Trình bày cách sử dụng Android Emulator: cấu hình, gọi điện, nhắn tin, xoay màn hình, giả lập cảm biến...

Bài 3: So sánh sự khác biệt về tốc độ, chức năng giữa máy ảo Android Emulator và máy ảo Android Genymotion

Bài 4: Trình bày cấu hình Debugging bằng USB cho thiết bị di động

Bài 5: Muốn máy tính kết nối với thiết bị di động để trình chiếu và tương tác qua lại thì dùng phần mềm nào? Hãy trình bày chi tiết cách tải và sử dụng.

Bài 6: Hãy trình bày các bước để một phần mềm thiết kế trong Android Studio có thể chạy được trên thiết bị di động.

LAB 2. CÁC LAYOUT CƠ BẢN

Layout là nơi để tổ chức sắp xếp control/View trên giao diện và thường được sử dụng đầu tiên khi ta thiết kế giao diện. Các Layout này được Android Studio bố trí trong Palette/Layouts:

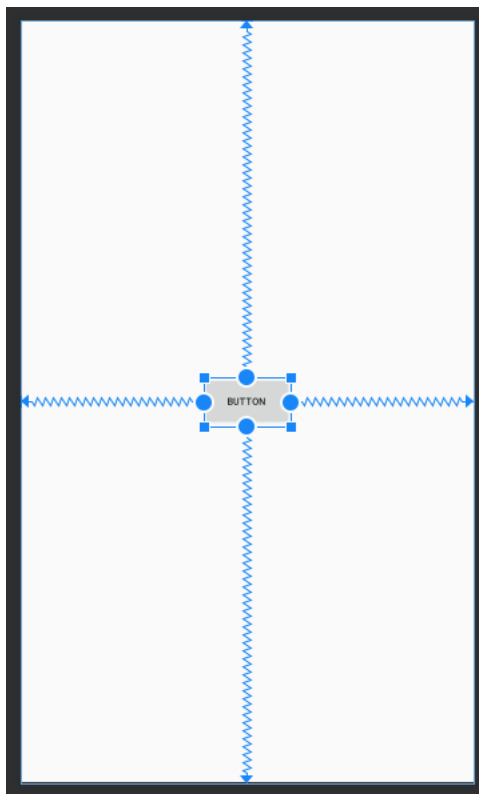
1. MỤC TIÊU:

- Trong bài này sẽ trình bày về các layout:
 - o ConstraintLayout
 - o Frame Layout
 - o Linear Layout
 - o Table Layout
 - o Relative layout
 - o LinearLayout
- Đọc thêm về các loại layout ở đây:
<https://developer.android.com/guide/topics/ui/layout>

2. LÝ THUYẾT:

2.1. Constraint Layout

Constraint layout dùng để định vị trí tương đối của Một control/View trên màn hình với các phần tử khác trong layout:



Ta có thể xác định một constraint cho một hay nhiều mặt của một view bằng chế độ kết nối bất kỳ sau đây :

- Điểm neo nằm trên một View khác.
- Một cạnh của layout.
- Một guideline.

Android cung cấp các loại Constraint sau: Relative positioning, Margins, Centering positioning, Visibility behavior, Dimension constraints, Chains.

Layout này được áp dụng để thiết kế các màn hình Responsive không lệ thuộc vào độ phân giải.

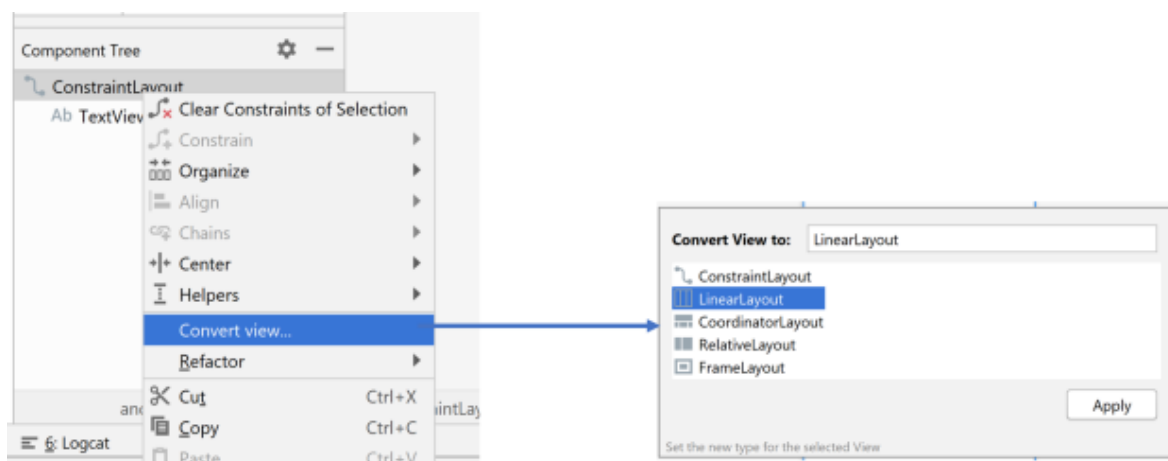
Cấu trúc XML của ConstraintLayout được mô tả như sau:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button2"
        android:layout_width="123dp"
        android:layout_height="70dp"
        android:text="Button"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"

    </androidx.constraintlayout.widget.ConstraintLayout>
```

Android Studio còn cung cấp chức năng chuyển đổi từ ConstraintLayout qua các layout khác(và ngược lại), bằng cách bấm chuột phải vào Layout bất kỳ/chọn Convert View ==>Chọn layout đích mà ta muốn chuyển đổi:



2.2. Frame Layout

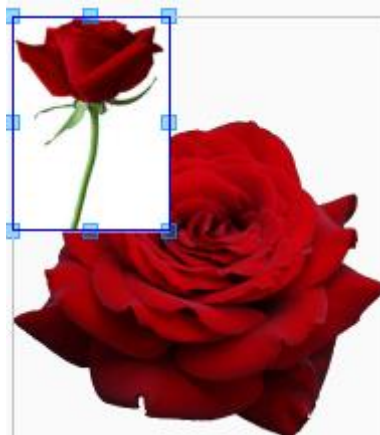
Sử dụng trong các trường hợp xây dựng bố cục tổ chức hiển thị một đối tượng duy nhất.

Đối tượng mặc định vị trí top-left trên FrameLayout, View khai báo sau sẽ chồng lên View khai báo trước, có thể sử dụng thuộc tính Gravity để thiết lập lại vị trí.

Cấu trúc xml của Frame Layout được mô tả như sau:

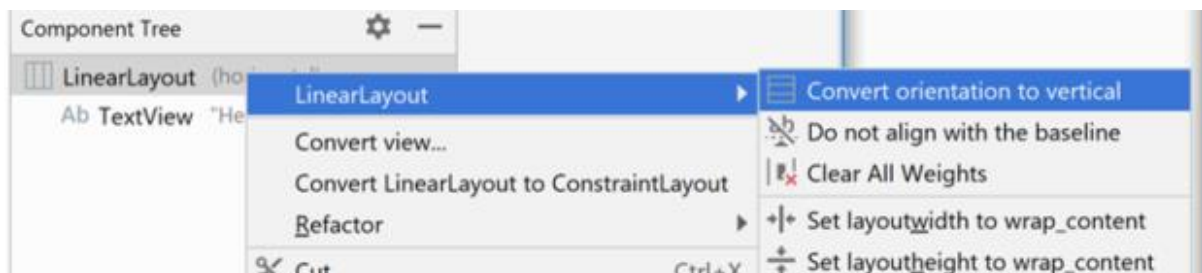
```
<FrameLayout android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/h0"
        android:id="@+id/imageView3"/>
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/h1"
        android:id="@+id/imageView4" />
</FrameLayout>
```

Ta thấy hình h1 khai báo sau nên h1 sẽ chồng lên h0, cả hai hình này đều được neo ở góc top-left của màn hình:



2.3. Linear Layout

Linear Layout lại vừa có Horizontal và Vertical, ta có thể bấm chuột phải vào nó để đổi tiếp:



Sử dụng trong các trường hợp xây dựng bố cục tổ chức hiển thị các đối tượng theo một chiều duy nhất (chiều dọc-vertical hoặc ngang-horizontal).

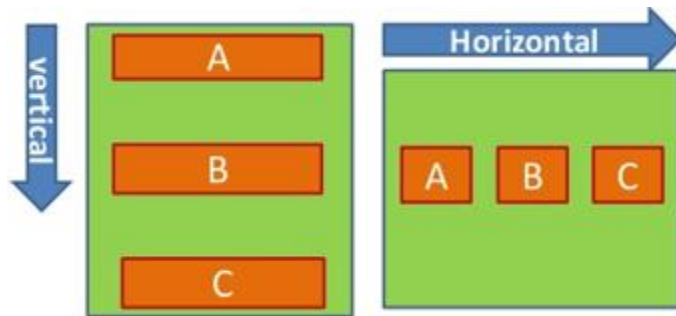
Đối tượng mặc định vị trí top left trên LinearLayout, có thể sử dụng thuộc tính Gravity để thiết lập lại vị trí.

Cấu trúc XML của LinearLayout có dạng như sau:

```
<LinearLayout
```

```
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/linearLayout1"
android:orientation="horizontal"
android:layout_width="match_parent"
android:layout_height="match_parent">
```

Ta thay đổi chiều dọc (`android:orientation="vertical"`), chiều ngang (`android:orientation="horizontal"`) các View sẽ tự động sắp xếp:

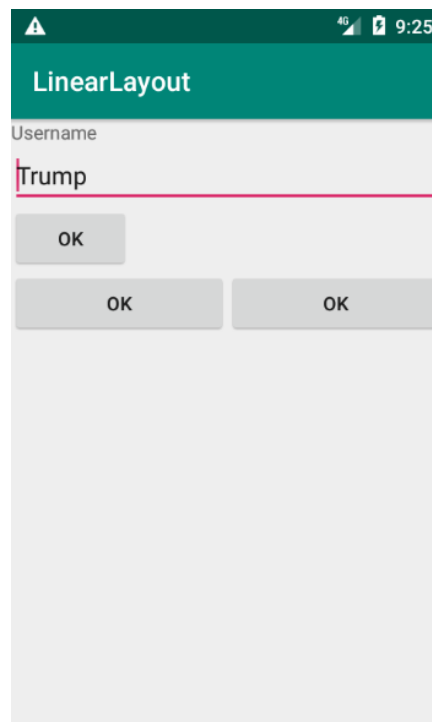


LinearLayout hỗ trợ thuộc tính text `android:gravity` để căn chỉnh chữ trong View, `android:layout_gravity` để căn chỉnh vị trí tương đối của View trong layout.



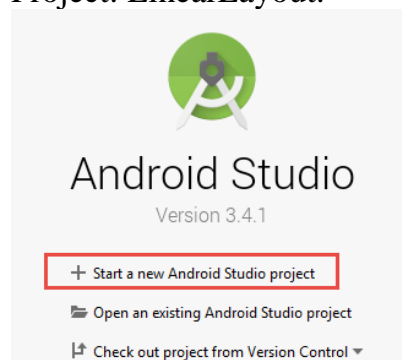
Ngoài ra LinearLayout còn hỗ trợ nhiều thuộc tính khác như `android:padding` để tăng độ dày của View, `android:layout_margin` để tăng/giảm khoảng cách giữa các View.

Ví dụ: Sử dụng Linear Layout, thiết kế giao diện như hình bên dưới.

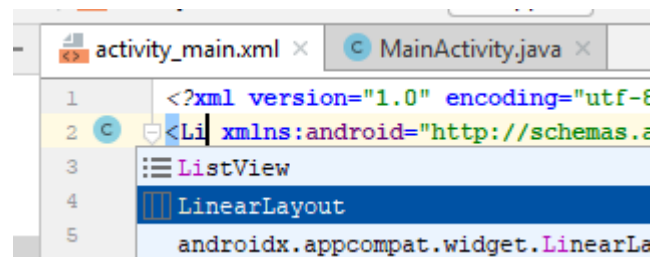


Hướng dẫn:

B1 - Mở Android Studio → New Project → Empty Activity → Đặt tên Project: LinearLayout.



B2 – Mở **activity_main.xml** sửa kiểu layout sử dụng thành **LinearLayout**.



B3 – Trong **activity_main.xml** thêm vào TextView, EditText, Button như sau:

```
<TextView
    android:text="Username"
    android:layout_gravity="center"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
<EditText
    android:text="Trump"
    android:id="@+id/edtName"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
<Button
    android:id="@+id/btnOk"
    android:text="OK"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

B4 - Trong **activity_main.xml** tiếp tục thêm vào LinearLayout, TextView, EditText, Button như sau:

```

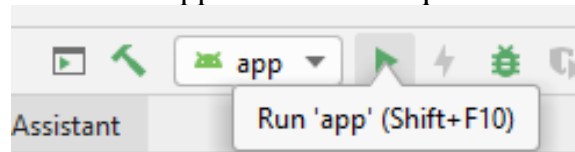
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <Button
        android:layout_weight="1"
        android:text="OK"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
    <Button
        android:layout_weight="1"
        android:text="OK"
        android:id="@+id/btnOk2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
    />
</LinearLayout>

```

B6 - Kết quả



B5 – Run ‘app’ kiểm tra kết quả.



2.4. Table Layout

TableLayout kế thừa từ LinearLayout, cho phép hiển thị các đối tượng theo nhiều dòng (TableRow). Mỗi dòng có thể chứa nhiều View, mỗi View được xem là một cột.

Đặc biệt TableLayout lấy dòng có số lượng View nhiều nhất làm số cột, tức là nếu TableLayout có 3 dòng: dòng đầu tiên có 2 View, dòng thứ hai có 5 View và dòng cuối cùng có 4 View thì cột được xác định cho TableLayout này là 5.

<TableLayout

```

xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent">
    <Tablerow>
        <Button/>
    </Tablerow>

```

</TableLayout>

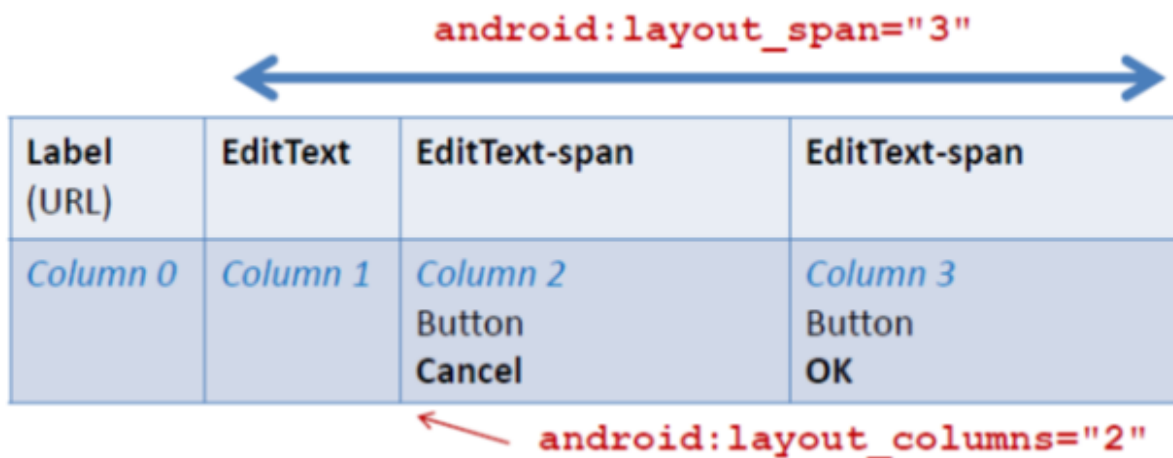
TableLayout cho phép ta Meger(trộn) các ô trong TableRow bằng cách thiết lập thuộc tính layout_span cho View trong TableRow:

```

<TableRow>
    <TextView android:text="URL:" />
    <EditText
        android:id="@+id/entry"
        android:layout_span="3" />
</TableRow>

```

Hay `layout_column` để di chuyển vị trí của View tới một cột nào đó trong `TableRow`:



Dùng `stretchColumns` để dẫn đều các control, các cell (ta thường dùng dấu "*" để cấu hình cho toàn bộ các cột, hoặc nhập số để cấu hình cho cột được chỉ định):

```

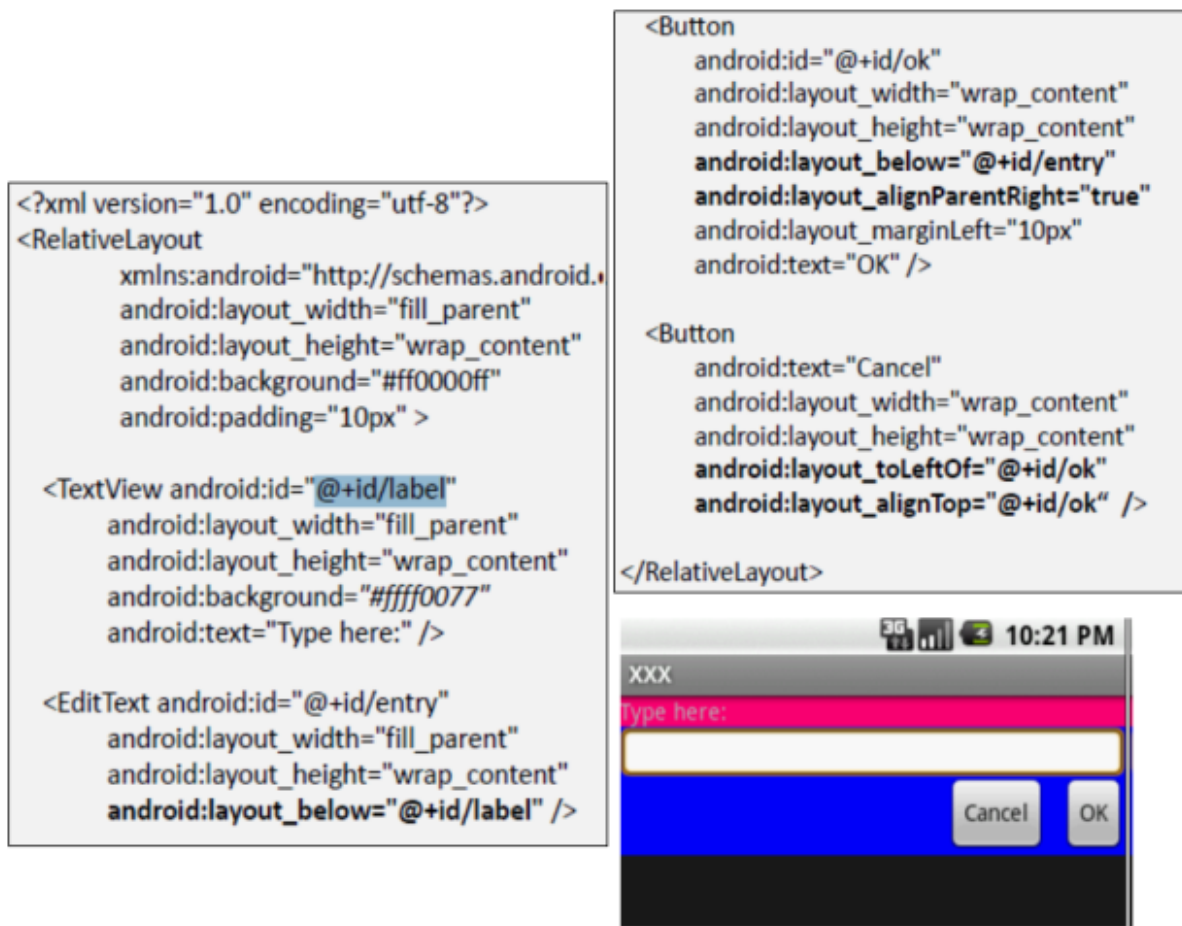
<TableLayout
    android:stretchColumns="*"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

```

2.5. Relative layout

`RelativeLayout` cho phép sắp xếp các View theo vị trí tương đối giữa các View khác trên giao diện (kể cả View chứa nó). Thường nó dựa vào Id của các View khác để sắp xếp theo vị trí tương đối.

Do đó khi làm `RelativeLayout` ta phải chú ý là đặt Id của View cho chuẩn xác, nếu sau khi Layout xong mà ta lại đổi Id của các View thì giao diện sẽ bị xáo trộn (do đó nếu đổi ID thì phải đổi luôn các tham chiếu khác sao cho khớp với Id mà ta mới đổi).



Trên giao diện lúc kéo thả các View trên RelativeLayout ta tự căn chỉnh, lúc này các View sẽ tự động sắp xếp theo cách ta kéo thả trên màn hình.

3. Bài tập:

Bài 1: Để sắp xếp các view trên giao diện theo chiều đứng và chiều nằm ngang thì ta nên dùng Layout nào?

Bài 2: ConstraintLayout dùng để làm gì ?

Bài 3: Muốn sắp xếp các View theo dạng dòng và cột thì ta dùng Layout nào ?

Bài 4: Các thuộc tính nào bắt buộc phải khai báo trong XML cho các View ?

Bài 6: Sử dụng layout phù hợp và các view để thiết kế màn hình sau:

CẬP NHẬT THÔNG TIN CÁ NHÂNCMND: CMND ở đâySố phone: Số phone ở đâyĐịa chỉ: Địa chỉ ở đây

CẬP NHẬT

LÀM LẠI

Bài 7: Viết chương trình chuyển đổi độ C qua độ F và ngược lại. Yêu cầu sử dụng LinearLayout để sắp xếp các View trên giao diện

**FAHRENHEIT/CELSIUS DEGREE
CONVERTER**

Fahrenheit:

nhập độ f ở đây

Celsius:

nhập độ c ở đâyCONVERT TO
CELSIUSCONVERT TO
FAHRENHEIT

CLEAR

Bài 8: Viết chương trình giải phương trình bậc 1 với giao diện dưới đây:

GIẢI PHƯƠNG TRÌNH BẬC 1

Nhập a:

nhập a ở đây

Nhập b:

nhập b ở đây

Kết quả:

x=

RESET

GIẢI

THOÁT

Bài 9: Viết giao diện đăng nhập như sau. Yêu cầu sử dụng LinearLayout

OnClickXML

TextView

Username ở đây

Password

Password ở đây

ĐĂNG NHẬP

THOÁT

Bài 10: Kết hợp các Layout và các View để thiết kế màn hình máy tính bỏ túi như sau:

