SCIENCES
SORBONNE
UNIVERSITÉ

MultipleActivities Example

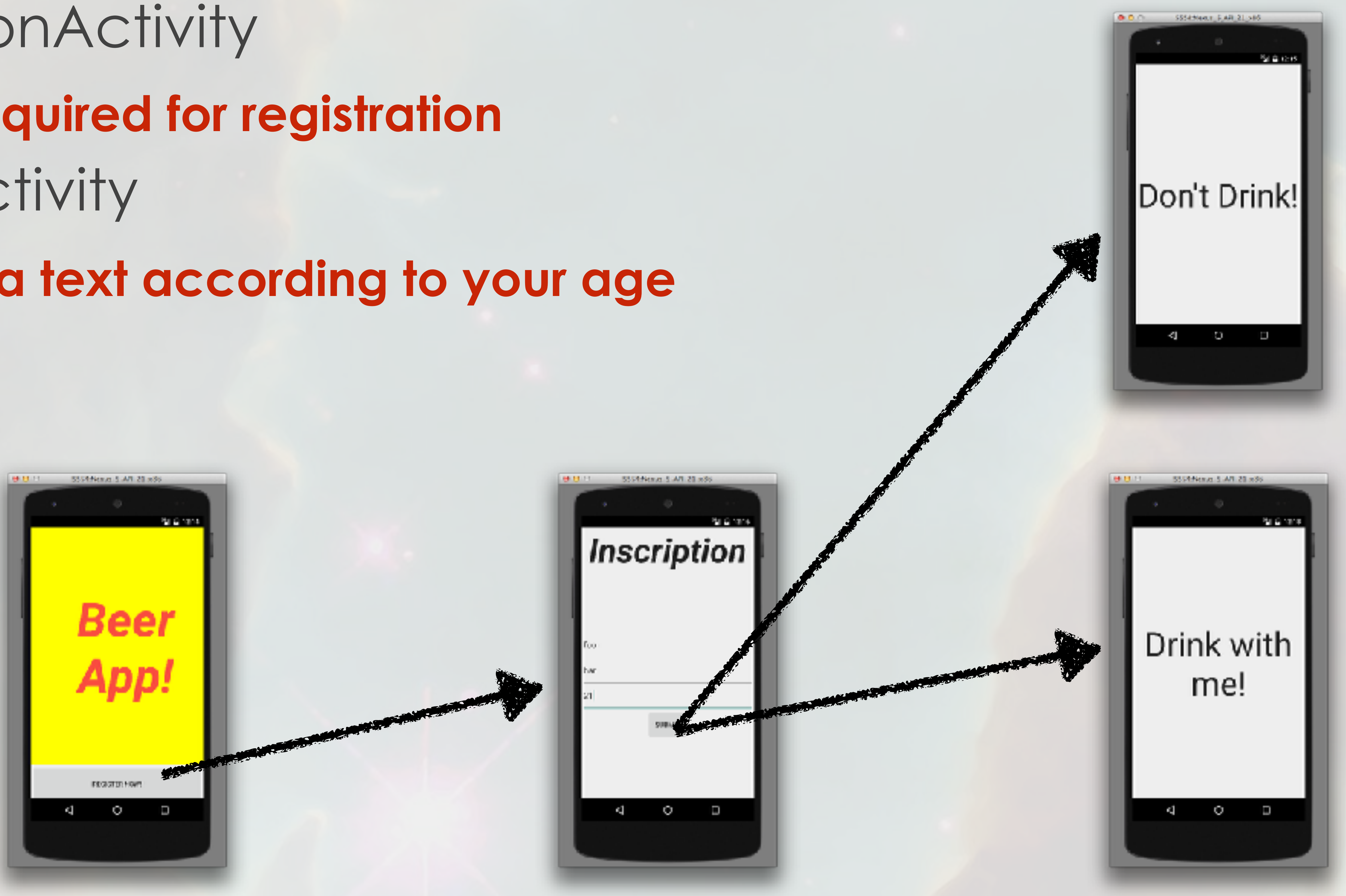
Renault@lrde.epita.fr

LRDE

BeerApp Example

Three activities

- MainActivity
  - **Start screen with a "register button"**
- InscriptionActivity
  - **Fields required for registration**
- ResultActivity
  - **Display a text according to your age**



Use XML to store static values

Group constants in XML files

- Good practice since you will not use magic static values in your code
- Avoid to repeat multiple time same text
- Put all constants into **res/value/string.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Beer App!</string>
    <string name="register_now">Register now!</string>
    <string name="inscription">Inscription</string>
    <string name="name">Name</string>
    <string name="firstName">First Name</string>
    <string name="age">Age</string>
    <string name="submit">Submit</string>
    <string name="okDrink">Drink with me!</string>
    <string name="koDrink">"Don't drink!"</string>
 </resources>
```

Build an App based on Localisation

Localisation

Bière Appli!

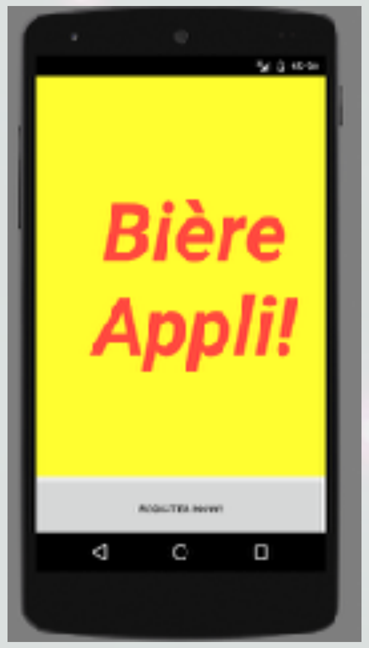
- Adapt your texts based on the language of the user
- Copy the **res/value** directory into **res/value-fr**
- **Edit res/value-fr/strings.xml**

```
<resources>
     <string name="app_name">Beer App!</string>
     <string name="register_now">Register now!</string>
     <string name="inscription">Inscription</string>
     <string name="name">Name</string>
     <string name="firstName">First Name</string>
     <string name="age">Age</string>
     <string name="submit">Submit</string>
     <string name="okDrink">Drink with me!</string>
     <string name="koDrink">"Don't drink!"</string>
  </resources>
```

- Android will select the correct file based on the Locale

Build an App based on Localisation

Localisation



- Adapt your texts based on the language of the user
- Copy the **res/value** directory into **res/value-fr**
- **Edit res/value-fr/strings.xml**

```
<resources>
     <string name="app_name">Beer App!</string>
     <string name="register_now">Register now!</string>
     <string name="inscription">Inscription</string>
     <string name="name">Name</string>
     <string name="firstname">Firstname</string>
     <string name="age">Age</string>
     <string name="submit">Submit</string>
     <string name="okDrink">Drink with me!</string>
     <string name="koDrink">"Don't drink!"</string>
  </resources>
```

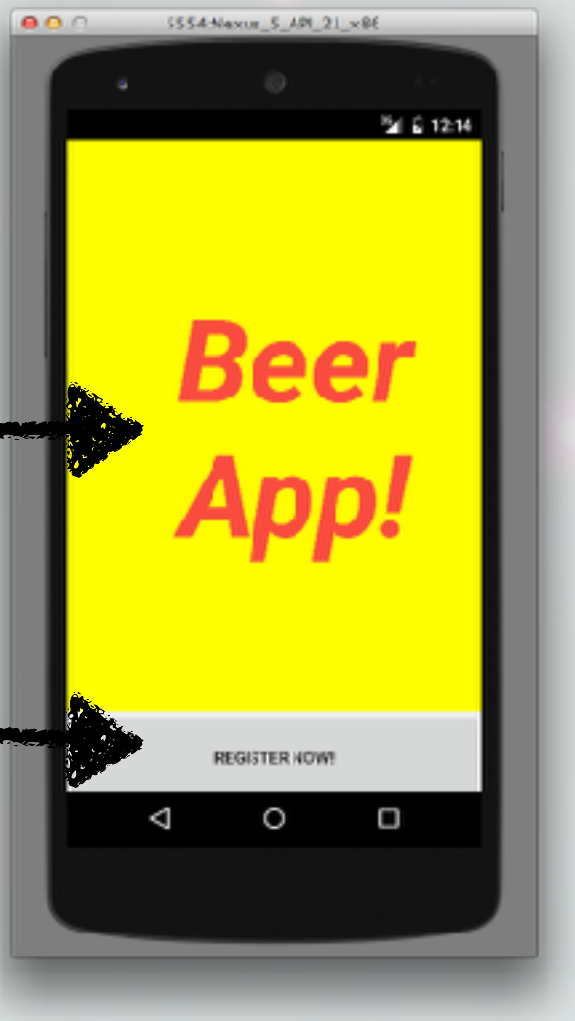
You can change the local of the device through
System > Preferences > Language

- Android will select the correct file based on the Locale

Main GUI (1/2)

Three elements in the hierarchy of views

- A LinearLayout
- A TextView
- A Button



GUI description: activity_main.xml

```
<LinearLayout xmlns:android=
     "http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:baselineAligned="false"
    android:weightSum="1">
```

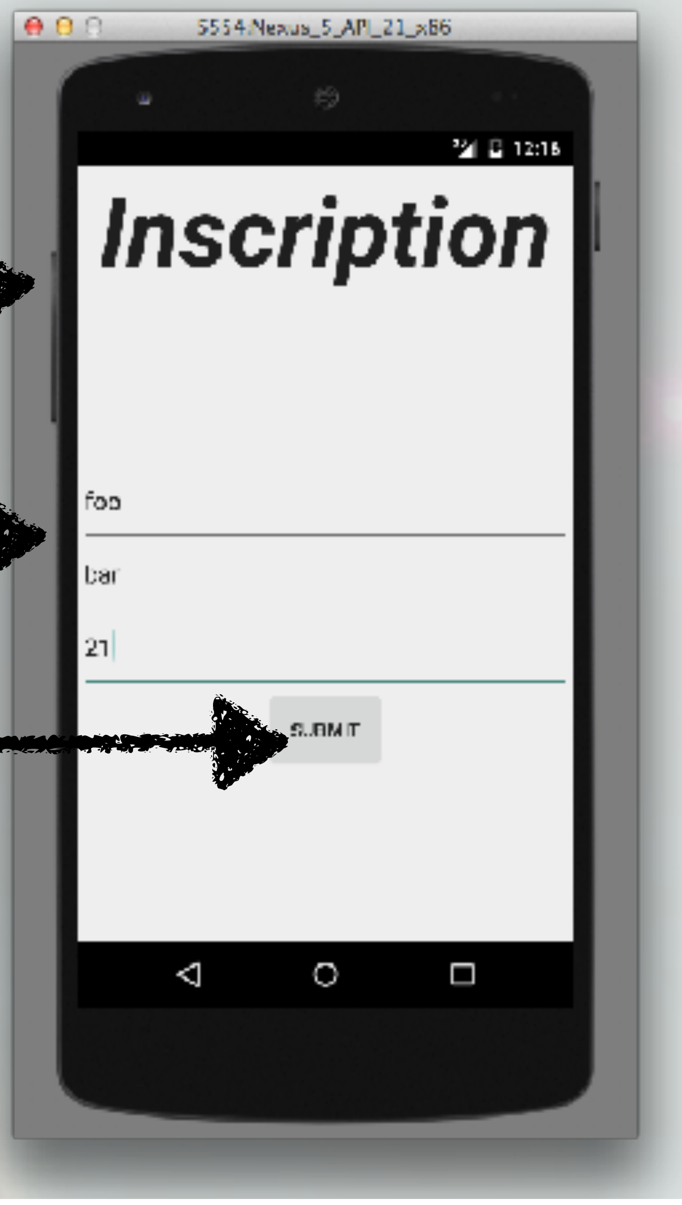
Main GUI (2/2)

```
    <TextView
        android:layout_width="396dp"
        android:layout_height="497dp"
        android:textAppearance=
                    "?android:attr/textAppearanceLarge"
        android:text="@string/app_name"
        android:id="@+id/display_name"
        android:gravity="center_vertical|center_horizontal"
        android:textStyle="bold|italic"
        android:textColor="#ffff4a45"
        android:textSize="100dp"
        android:background="#ffffff53" />
    <Button
        android:layout_width="match_parent"
        android:layout_height="79dp"
        android:text="@string/register_now"
        android:id="@+id/button_register" />
</LinearLayout>
```

Registration Form GUI (1/2)

Five elements in the hierarchy of views

- A LinearLayout
- A TextView
- Three EditText
- A Button



foo

bar

21

GUI description: activity_inscription.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"    android:layout_height="fill_parent"
    android:orientation="vertical"        android:baselineAligned="false"
    android:weightSum="1">

   <TextView
        android:layout_width="match_parent"      android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/inscription"       android:id="@+id/textView"
        android:layout_weight="0.43"             android:gravity="center_horizontal"
        android:textStyle="bold|italic"          android:textSize="70dp"
/>
```

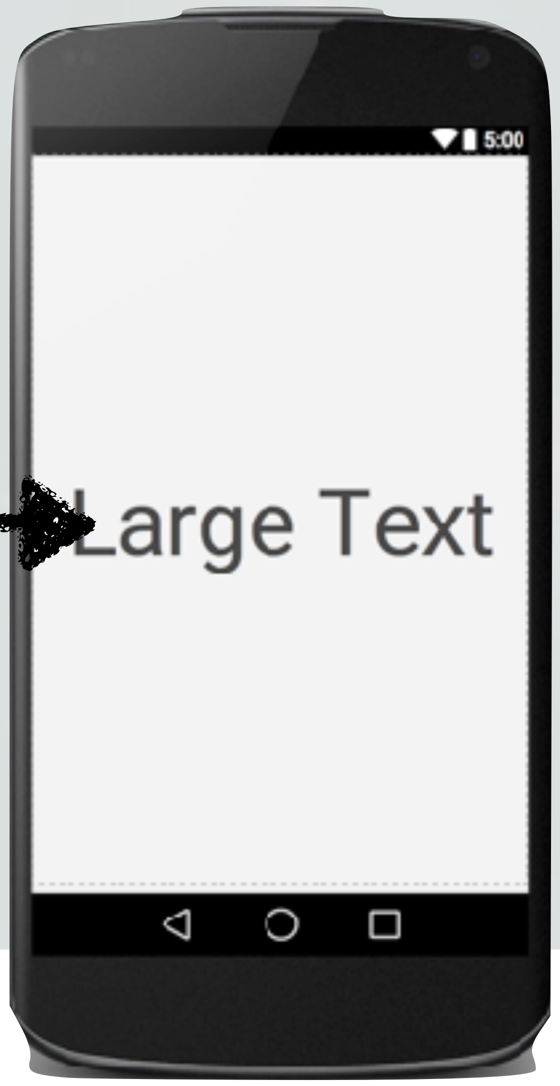
Registration Form GUI (2/2)

```
  <EditText
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:inputType="textPersonName"    android:text="@string/firstName"
      android:ems="10"                      android:id="@+id/text_firstname"
      android:layout_weight="0.04" />

  <EditText
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:inputType="number"            android:ems="10"
      android:id="@+id/text_age"
      android:layout_gravity="center_horizontal"
      android:layout_weight="0.04"          android:text="@string/age" />

  <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/submit"         android:id="@+id/button_submit"
      android:layout_gravity="center_horizontal"
      android:layout_weight="0.04" />

</LinearLayout>
```

GUI post inscription

Two elements in the hierarchy of views

- A LinearLayout
- A TextView with runtime initialization



GUI description: activity_result.xml

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"       android:layout_height="fill_parent"
    android:orientation="vertical"           android:baselineAligned="false"
    android:weightSum="1">
    <TextView
        android:layout_width="match_parent"  android:layout_height="match_parent"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Large Text"       android:id="@+id/text_display"
        android:textSize="70dp"
        android:gravity="center_vertical|center_horizontal" />
</LinearLayout>
```

MainActivity

Must build the GUI and track clicks on button

- Build the GUI during the call to the onCreate callback
- Action when the button is clicked
  - **Implement View.OnClickListener Load the GUI XML during creation**
  - **set this as listener**

```
public class MainActivity extends Activity implements View.OnClickListener {
    Button btGotoInscription;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btGotoInscription = (Button) findViewById(R.id.button_register);
        btGotoInscription.setOnClickListener(this);
    }
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(this, InscriptionActivity.class);
        startActivity(intent);
    }
}
```

Registration Form

Action when the user click on the button

- We must capture the age field
- Always implement View.OnClickListener

```
public class InscriptionActivity
             extends Activity implements View.OnClickListener {
    private Button buttonSubmit;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_inscription);
        buttonSubmit = (Button) findViewById(R.id.button_submit);
        buttonSubmit.setOnClickListener(this);
    }
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(this, ResultActivity.class);
        TextView age = (TextView)findViewById(R.id.text_age);
        intent.putExtra("AGE", Integer.parseInt(age.getText().toString()));
        startActivity(intent);
    }
}
```

Last Screen: display result

- Must grab the age from the intent
- Mut instantiate the XML

```java
public class ResultActivity extends Activity{
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_result);
        int age = getIntent().getIntExtra("AGE", 0);
        TextView display =
              (TextView)findViewById(R.id.text_display);
        if (age <= 20)
            display.setText(R.string.koDrink);
        else
            display.setText(R.string.okDrink);
    }
}
```

Setup Android Permissions

Modify the AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.example.admin.myfirstapplication" >
   <application
      android:allowBackup="true"
      android:icon="@drawable/ic_launcher"
      android:label="@string/app_name"
      android:theme="@style/AppTheme" >
      <activity
          android:name=".MainActivity"
          android:label="@string/app_name" >
         <intent-filter>
           <action android:name="android.intent.action.MAIN" />
           <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
      </activity>
      <activity android:name=".InscriptionActivity"/>
      <activity android:name=".ResultActivity"/>
    </application>
</manifest>
```

Summary

End to end application

- Three Activities
- Manage simple events (clicks)
- Simple Navigation through the back stack
- Intents
  - **With or Without parameters**
- Static value management
  - **Throught res/values/*.xml**

Localisation

- On simple value type
- Can also be applied on all ressources
- Application name can also be changed!

F. Kordon - Sorbonne Université - CC2018