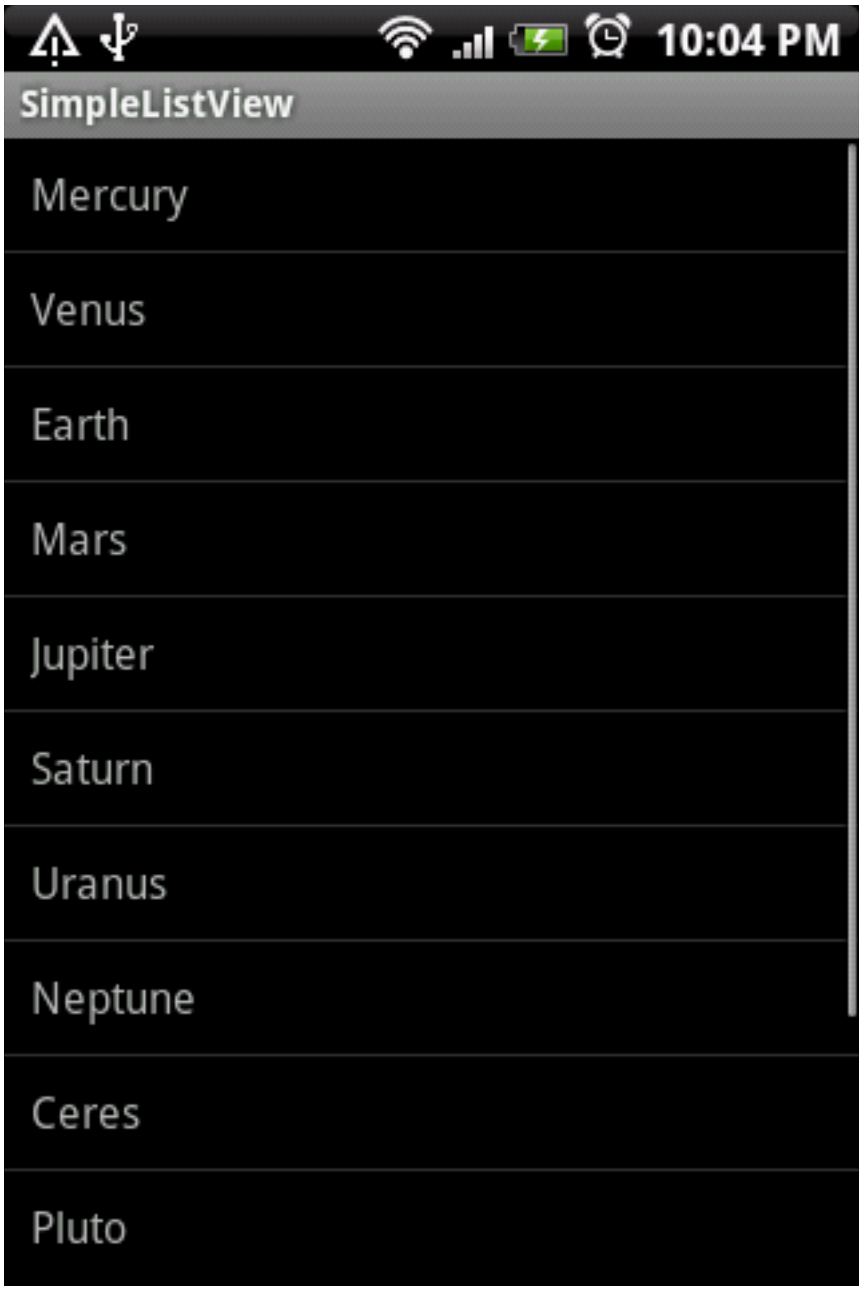
A Simple Android ListView Example

- Extrait du site *windrealm : http://windrealm.org/tutorials/android/android-listview.php*
- how to create a simple ListView and populate it with text data (the names of various planets).
- The following picture shows what the Android program looks like.



Basically we create a ListView class and use TextView objects for each row. Each planet name is rendered in a TextView.

Main Layout File

Our ListView is defined in the main layout file (res/layout/main.xml) within a LinearLayout.

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">

    <ListView android:layout_width="fill_parent"
      android:layout_height="fill_parent"
      android:id="@+id/mainListView">
        </ListView>

    </LinearLayout>
```

The **resource ID** of the ListView is **mainListView**, which we will use to get a reference to the ListView in our Activity class.

```java
// Find the ListView resource.
mainListView = (ListView) findViewById( R.id.mainListView );
```

Row Layout File

Each row in the ListView will be a TextView. The TextView is defined in another file (res/layout/simplerow.xml).

```xml
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
 android:id="@+id/rowTextView"
 android:layout_width="fill_parent"
 android:layout_height="wrap_content"
 android:padding="10dp"
 android:textSize="16sp" >
</TextView>
```

Activity

Our main activity (SimpleListViewActivity) creates an ArrayAdapter, which holds the objects to be displayed in the ListView. The ArrayAdapter constructor is passed the resource ID of the TextView layout file (`R.layout.simplerow`). The ArrayAdapter will use it to instantiate a TextView for each row.

We then set the ArrayAdapter as our ListView's adapter.

```java
package com.windrealm.android;

import java.util.ArrayList;
import java.util.Arrays;

import android.app.Activity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;

     public class SimpleListViewActivity extends Activity {

       private ListView mainListView ;
       private ArrayAdapter<String> listAdapter ;

       /** Called when the activity is first created. */
       @Override
       public void onCreate(Bundle savedInstanceState) {
         super.onCreate(savedInstanceState);
         setContentView(R.layout.main);

         // Find the ListView resource.
         mainListView = (ListView) findViewById( R.id.mainListView );

         // Create and populate a List of planet names.
         String[] planets = new String[] { "Mercury", "Venus", "Earth", "Mars",
                                           "Jupiter", "Saturn", "Uranus", "Neptune"};
         ArrayList<String> planetList = new ArrayList<String>();
         planetList.addAll( Arrays.asList(planets) );

         // Create ArrayAdapter using the planet list.
         listAdapter = new ArrayAdapter<String>(this, R.layout.simplerow, planetList);

         // Add more planets. If you passed a String[] instead of a List<String>
         // into the ArrayAdapter constructor, you must not add more items.
         // Otherwise an exception will occur.
         listAdapter.add( "Ceres" );
         listAdapter.add( "Pluto" );
```

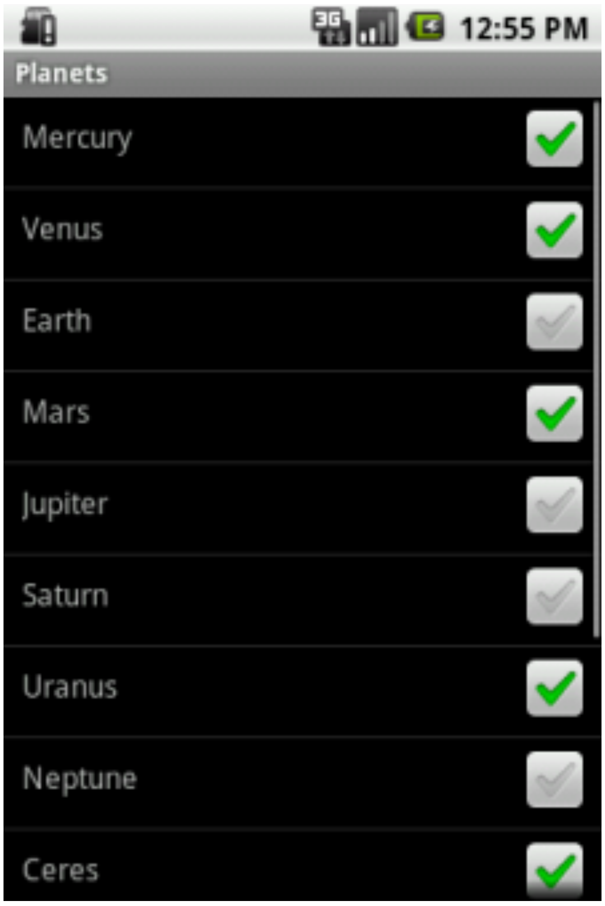
```
        listAdapter.add( "Haumea" );
        listAdapter.add( "Makemake" );
        listAdapter.add( "Eris" );

        // Set the ArrayAdapter as the ListView's adapter.
        mainListView.setAdapter( listAdapter );
      }
    }
```

A ListView with Checkboxes (Without Using ListActivity)

Contents

Introduction

This example shows how to create a ListView with a CheckBox in each row, without having our Activity extend ListActivity.



Activity Source Listing

Most of the functionality is contained in the Activity class.

```java
package com.windrealm.android;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

import android.app.Activity;
import android.content.Context;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.CheckBox;
import android.widget.ListView;
import android.widget.TextView;

public class PlanetsActivity extends Activity {

  private ListView mainListView ;
  private Planet[] planets ;
  private ArrayAdapter<Planet> listAdapter ;

  /** Called when the activity is first created. */
  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    // Find the ListView resource.
    mainListView = (ListView) findViewById( R.id.mainListView );

    // When item is tapped, toggle checked properties of CheckBox and Planet.
    mainListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
      @Override
      public void onItemClick( AdapterView<?> parent, View item,
                               int position, long id) {
        Planet planet = listAdapter.getItem( position );
        planet.toggleChecked();
        PlanetViewHolder viewHolder = (PlanetViewHolder) item.getTag();
        viewHolder.getCheckBox().setChecked( planet.isChecked() );
      }
    });


    // Create and populate planets.
    planets = (Planet[]) getLastNonConfigurationInstance() ;
    if ( planets == null ) {
      planets = new Planet[] {
          new Planet("Mercury"), new Planet("Venus"), new Planet("Earth"),
          new Planet("Mars"), new Planet("Jupiter"), new Planet("Saturn"),
          new Planet("Uranus"), new Planet("Neptune"), new Planet("Ceres"),
          new Planet("Pluto"), new Planet("Haumea"), new Planet("Makemake"),
          new Planet("Eris")
      };
    }
    ArrayList<Planet> planetList = new ArrayList<Planet>();
    planetList.addAll( Arrays.asList(planets) );

    // Set our custom array adapter as the ListView's adapter.
    listAdapter = new PlanetArrayAdapter(this, planetList);
    mainListView.setAdapter( listAdapter );
  }

  /** Holds planet data. */
  private static class Planet {
    private String name = "" ;
    private boolean checked = false ;
    public Planet() {}
    public Planet( String name ) {
      this.name = name ;
    }
    public Planet( String name, boolean checked ) {
      this.name = name ;
      this.checked = checked ;
    }
    public String getName() {
      return name;
    }
    public void setName(String name) {
      this.name = name;
    }
    public boolean isChecked() {
      return checked;
    }
    public void setChecked(boolean checked) {
      this.checked = checked;
    }
    public String toString() {
      return name ;
    }
    public void toggleChecked() {
      checked = !checked ;
    }
  }

  /** Holds child views for one row. */
```

```java
    private static class PlanetViewHolder {
      private CheckBox checkBox ;
      private TextView textView ;
      public PlanetViewHolder() {}
      public PlanetViewHolder( TextView textView, CheckBox checkBox ) {
        this.checkBox = checkBox ;
        this.textView = textView ;
      }
      public CheckBox getCheckBox() {
        return checkBox;
      }
      public void setCheckBox(CheckBox checkBox) {
        this.checkBox = checkBox;
      }
      public TextView getTextView() {
        return textView;
      }
      public void setTextView(TextView textView) {
        this.textView = textView;
      }
    }

    /** Custom adapter for displaying an array of Planet objects. */
    private static class PlanetArrayAdapter extends ArrayAdapter<Planet> {

      private LayoutInflater inflater;

      public PlanetArrayAdapter( Context context, List<Planet> planetList ) {
        super( context, R.layout.simplerow, R.id.rowTextView, planetList );
        // Cache the LayoutInflate to avoid asking for a new one each time.
        inflater = LayoutInflater.from(context) ;
      }

      @Override
      public View getView(int position, View convertView, ViewGroup parent) {
        // Planet to display
        Planet planet = (Planet) this.getItem( position );

        // The child views in each row.
        CheckBox checkBox ;
        TextView textView ;

        // Create a new row view
        if ( convertView == null ) {
          convertView = inflater.inflate(R.layout.simplerow, null);

          // Find the child views.
          textView = (TextView) convertView.findViewById( R.id.rowTextView );
        checkBox = (CheckBox) convertView.findViewById( R.id.CheckBox01 );

        // Optimization: Tag the row with it's child views, so we don't have to
        // call findViewById() later when we reuse the row.
        convertView.setTag( new PlanetViewHolder(textView,checkBox) );

        // If CheckBox is toggled, update the planet it is tagged with.
        checkBox.setOnClickListener( new View.OnClickListener() {
          public void onClick(View v) {
            CheckBox cb = (CheckBox) v ;
            Planet planet = (Planet) cb.getTag();
            planet.setChecked( cb.isChecked() );
          }
        });
      }
      // Reuse existing row view
      else {
        // Because we use a ViewHolder, we avoid having to call findViewById().
        PlanetViewHolder viewHolder = (PlanetViewHolder) convertView.getTag();
        checkBox = viewHolder.getCheckBox() ;
        textView = viewHolder.getTextView() ;
      }

      // Tag the CheckBox with the Planet it is displaying, so that we can
      // access the planet in onClick() when the CheckBox is toggled.
      checkBox.setTag( planet );

      // Display planet data
      checkBox.setChecked( planet.isChecked() );
      textView.setText( planet.getName() );

      return convertView;
    }

  }

  public Object onRetainNonConfigurationInstance() {
    return planets ;
  }
}
```

Downloads

The source code and Eclipse project files for this tutorial can be downloaded here.
(http://windrealm.org/tutorials/android/Planets.zip)