

# Programming Mobile Applications with Android Lab4

22-26 September, Albacete, Spain

Jesus Martínez-Gómez



# Programming Mobile Applications with Android

- Android Lab IV.- Create, compile and execute a multimedia application including images, sounds and videos
  - Follow the instructions to create the an android application with multimedia elements
    - Images
    - Sounds
    - Videos

# Programming Mobile Applications with Android

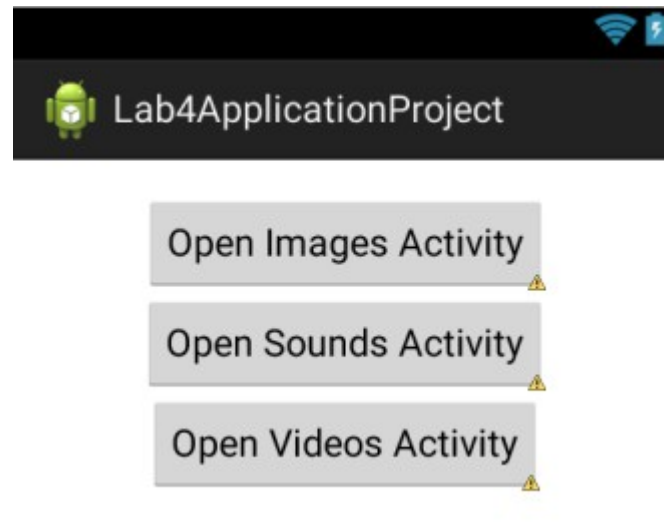
- Download and copy in your Workspace the Android application project named Lab4ApplicationProject that includes
  - MainActivity.java
    - Class that loads an Internet image when is started
  - activity\_main.xml
    - Layout with a single ImageView
  - DownloadImageClass.java
    - Class for downloading an image in a separated thread

# Programming Mobile Applications with Android

- Now is time to create a more complex application with different windows/screens → activities
  - MainActivity → Used as main menu to open the other activities
  - ActivityImages → Used to show how to load images
  - ActivityJukeBox → Used to develop a jukebox playing a sound
  - ActivityVideos → Used to play videos in different ways

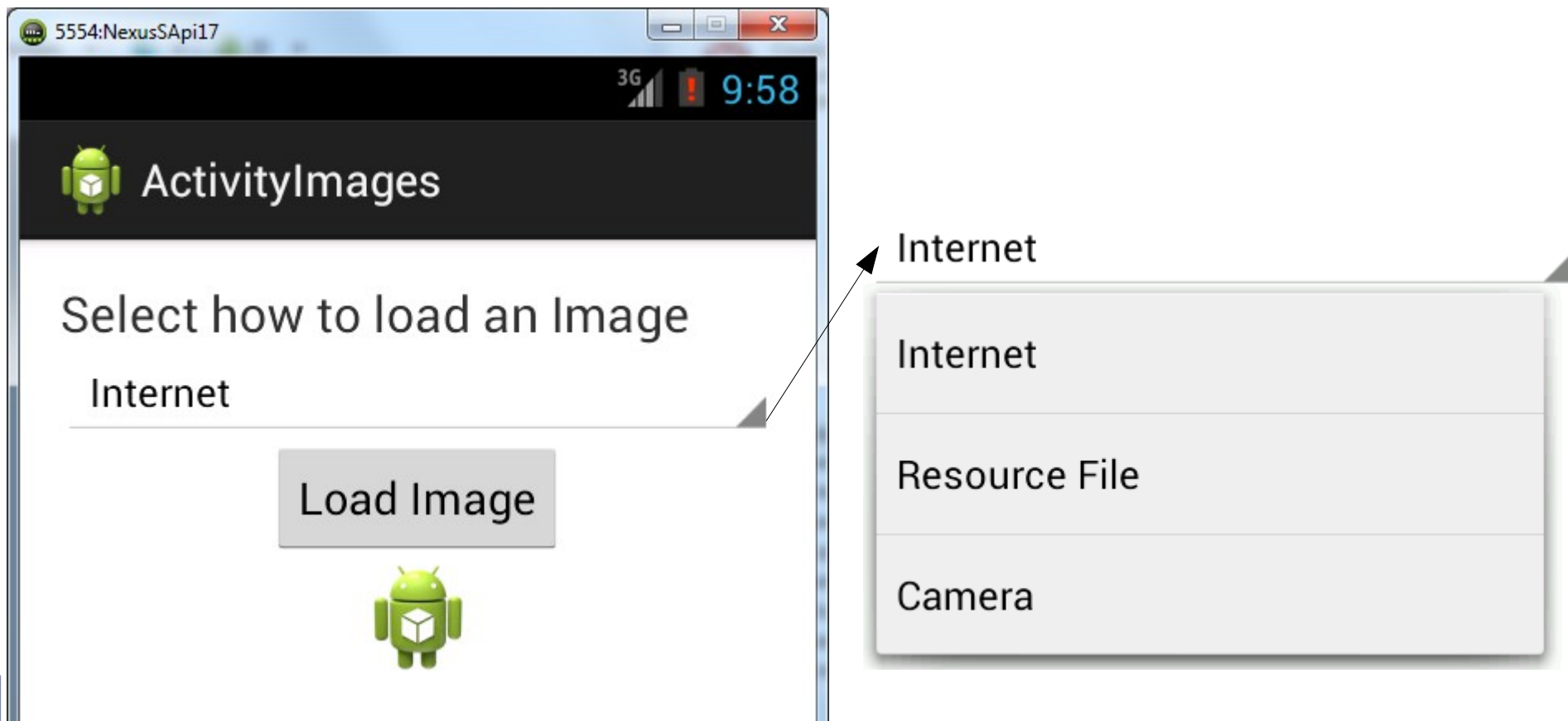
# Programming Mobile Applications with Android

- Create all the missing activities with their associated layouts
- Edit the layout of the main activity to have something like:



# Programming Mobile Applications with Android

- Modify the layout of the Images activity to have something like this



# Programming Mobile Applications with Android

- Remember to:
  - Associate the entries of the spinner with a `<string-array>` resource file (strings.xml)
  - Establish valid IDs for the spinner and the image view
  - Associate the `onClick` event with a valid method
    - This method should check the selected element in the spinner and then perform the right option

# Programming Mobile Applications with Android

- Options:
  - Load from internet → Use DownloadImageClass
  - Load from the camera and resource
    - Use the method startActivityForResult
    - Resource

```
Intent intent = new  
Intent(Intent.ACTION_PICK,android.provider.MediaStore.Images.Media.INTERNAL_CONTENT_URI);  
  
intent.setType("image/*");  
  
startActivityForResult(intent, SELECT_IMAGE);
```

- Camera

```
Intent camaraIntent = new Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);  
  
startActivityForResult(camaraIntent, TAKE_PHOTO);
```



# Programming Mobile Applications with Android

- Options:
  - Load from the camera and resource
    - Override the onActivityResult method

- Resource

```
Bitmap imagen = (Bitmap) data.getExtras().get("data");  
ImageView iv_foto = (ImageView) findViewById(R.id.imageViewToSet);  
iv_foto.setImageBitmap(imagen);
```

- Internet

```
Uri selectedImage = data.getData();  
ImageView iv_foto = (ImageView) findViewById(R.id.imageViewToSet);  
iv_foto.setImageURI(selectedImage);
```

- Use this constants

```
private static final int TAKE_PHOTO = 1;  
private int SELECT_IMAGE = 237487;
```

OSSCOM Programming Mobile Applications with Android

# Programming Mobile Applications with Android

- Modify the layout of the jukeBox activity to have something like this:

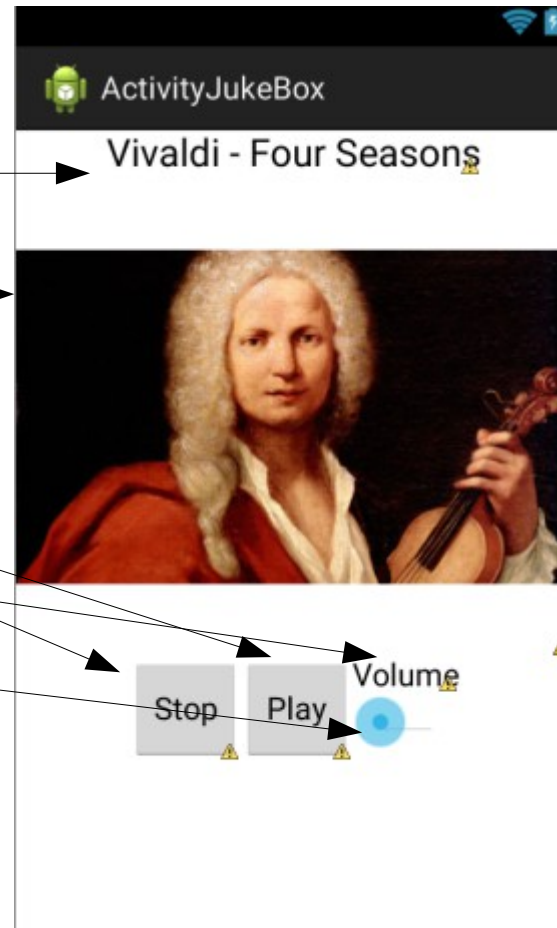
- TextView

- ImageView

- Button

- TextView

- SeekBar



# Programming Mobile Applications with Android

- Add the following steps
  - Add a mp3 resource file into a new folder named raw
    - <http://freemusicarchive.org/music/download/9e5ced77201179ck>
    - Use only minor case letter in the name of the archive
  - Add an image from Vivaldi to the resources
  - Add methods to control the play/stop onClicks
  - Add a MediaPlayer object in the activity, and associate it with the created resource (in the onCreate)

```
myPlayer = MediaPlayer.create(this, R.raw.four_season_vivaldi);
```

# Programming Mobile Applications with Android

- Now fill the code to obtain the desired behavior
  - MediaPlayer methods
    - start()
    - stop()
    - Pause()
    - SeekTo(0)
  - Setting the volume

```
AudioManager audio = (AudioManager) getSystemService(Context.AUDIO_SERVICE);  
int maxVolume = audio.getStreamMaxVolume(AudioManager.STREAM_MUSIC);  
double percent = (progress/(100.0));  
int volume = (int) (maxVolume*percent);  
audio.setStreamVolume(AudioManager.STREAM_MUSIC, volume, 0);
```

# Programming Mobile Applications with Android

- Finally, we will create the activity to load videos in our application
  - Two ways to do it

- Associate a VideoView with a video resource

```
VideoView myVV = (VideoView)findViewById(R.id.videoViewResource);  
Uri path = Uri.parse("android.resource://" + getPackageName() + "/" + R.raw.sample_video);  
myVV.setVideoURI(path);  
myVV.setMediaController(new MediaController(this) { });  
myVV.start();
```

- Launch a new activity in charge of visualizing the video

```
startActivity(new Intent(Intent.ACTION_VIEW,  
Uri.parse("https://www.youtube.com/watch?v=GRxofEmo3HA")));
```

# Programming Mobile Applications with Android Lab4

22-26 September, Albacete, Spain

Jesus Martínez-Gómez

