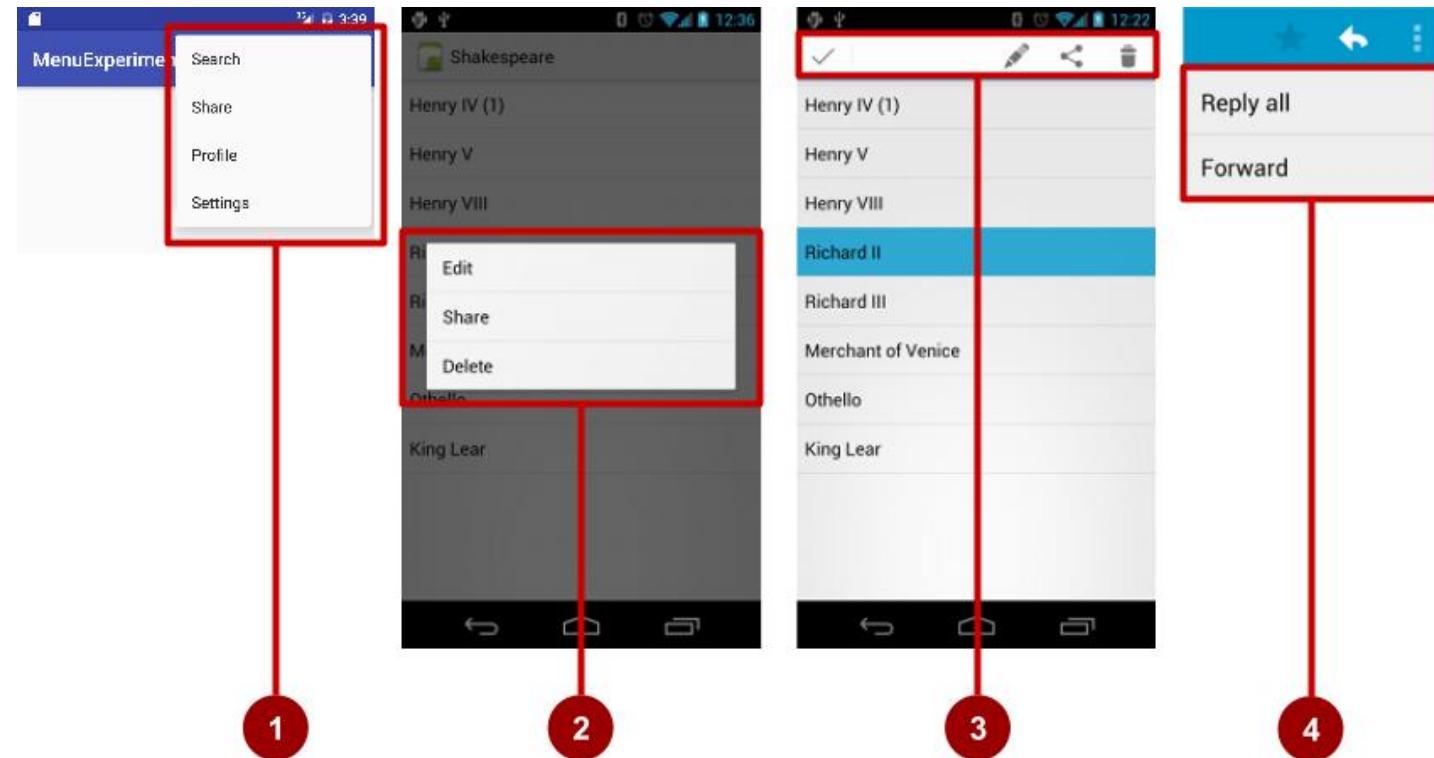


# **LES MENUS**

# Menus

- Barre d'Action avec des options de menu
- menu Contextuel
- Action bar Contextuelle
- Popup menu

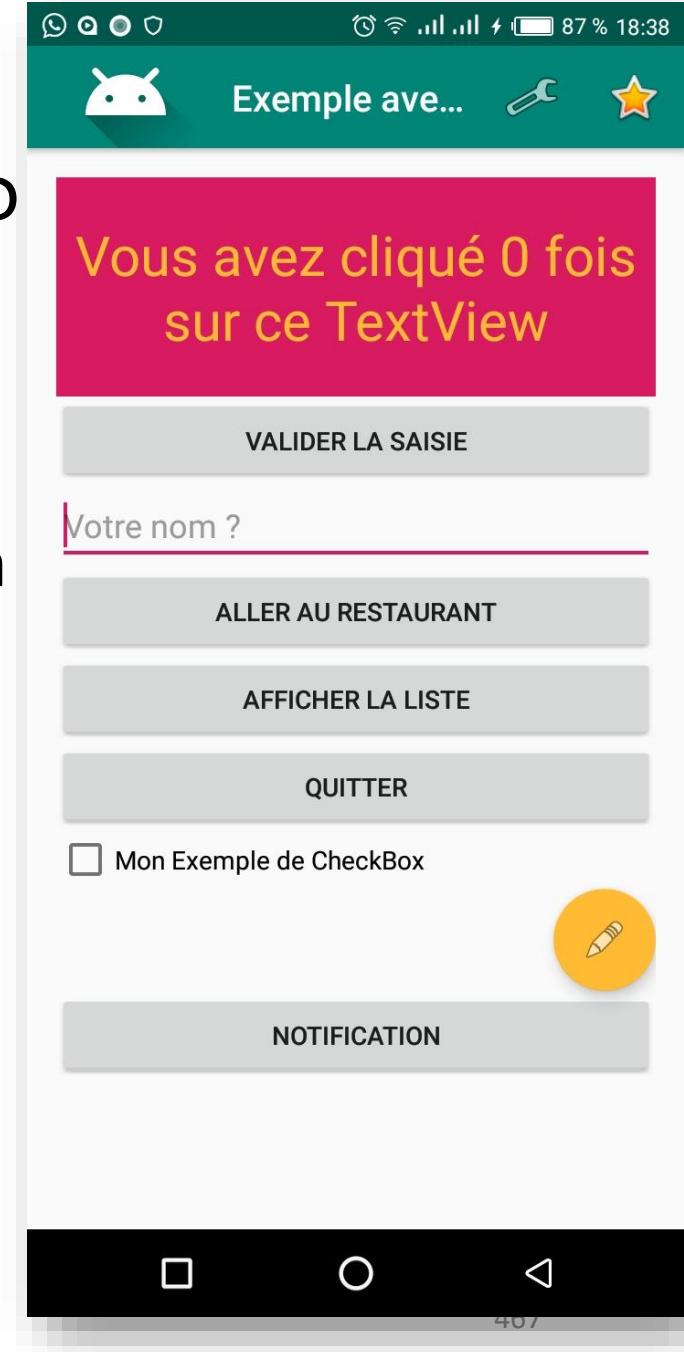
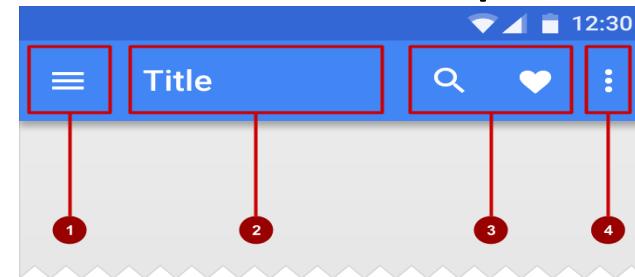


# Barre de menu avec options:

## Qu'est-ce qu'une barre de menus ?

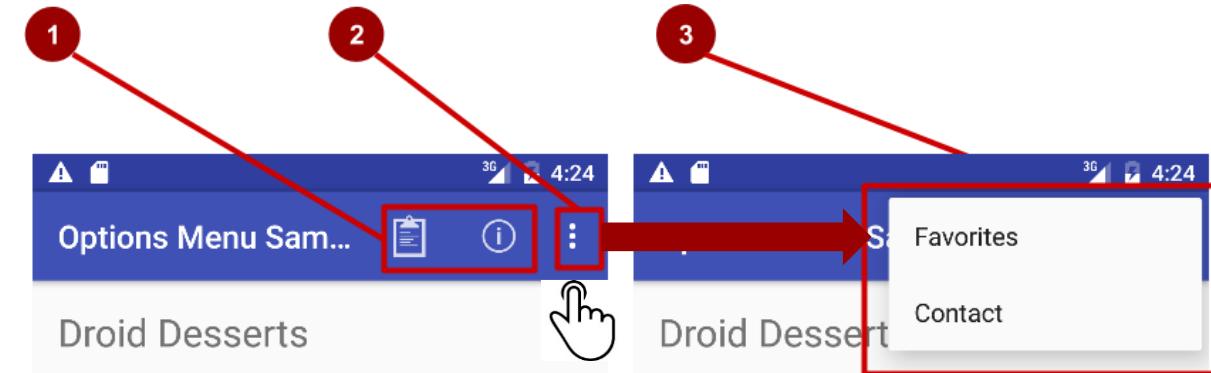
Barre au dessous des écrans — identique quelque soit le téléphone

1. L'icône de navigation permet d'ouvrir le navigation drawer
2. Titre de l'activité courante
3. Icons des items des options du menu
4. Boutons d'actions pour le reste des options menu



# Quelles sont les options du menu

- Icônes d'action dans la barre d'applications pour les éléments importants (1)
- Appuyez sur les trois points, sur le "bouton de débordement d'action" pour afficher le menu d'options (2).



- Apparaît dans le coin droit de la barre d'applications (3)
- Pour naviguer vers d'autres activités et modifier les paramètres de l'application

# Ajouter les options du Menu: Étapes pour implémenter le menu d'options

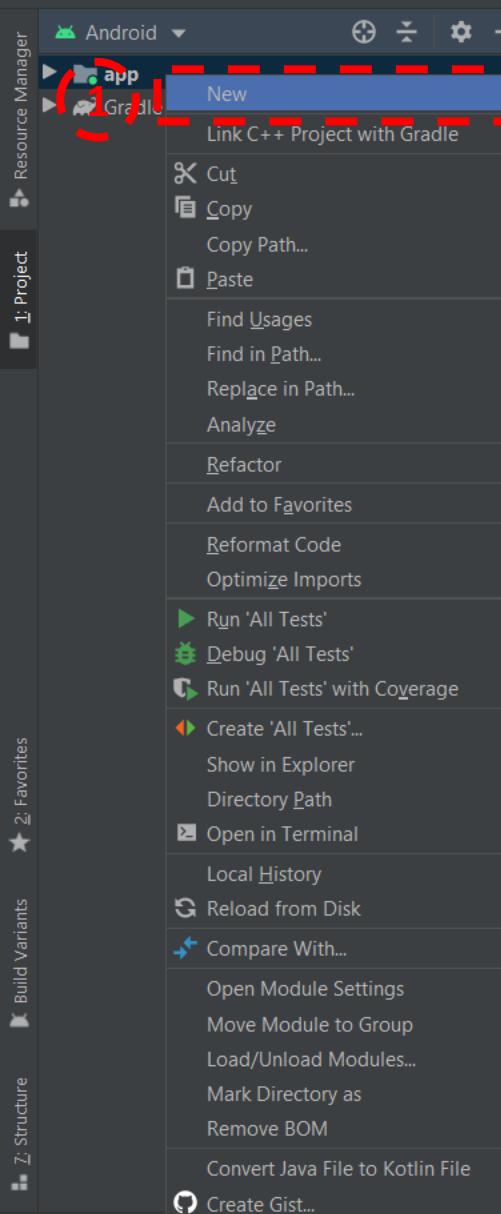
1. Ressource de menu XML (menu\_main.xml)
2. onCreateOptionsMenu () pour afficher le menu
3. Attribut onClick ou onOptionsItemSelected ()
4. Méthode pour gérer le clic sur un élément

# Créer le menu

1. Créer un répertoire de ressources (Ressource Directory) nommé **menu**
2. Créer une ressource de menu XML (menu\_main.xml)
3. Ajouter une entrée pour chaque élément de menu (**Paramètres et Favoris**):

```
<item android:id="@+id/option_settings"
      android:title="Settings" />

<item android:id="@+id/option_favorites"
      android:title="Favorites" />
```



Search Everywhere Double Shift

Go to File Ctrl+Maj+N

Recent Files Ctrl+E

Navigation Bar Alt+Origine

Drop files here to open

Morpion > app > src > main > res

activity\_main.xml o.png MainActivity.java

Palette: Common Ab TextView

Resource type: menu

Component Tree:

- LinearLayout
  - Ab txt1 "Exe"
  - gl
    - button
    - button
    - button
    - button
    - button
    - button20 "Button"
    - button21 "Button"
    - button22 "Button"
  - textView
  - button13 "Relancer le jeu"

Attributes: Attributes

Code Split Design

Resource Manager: Gradle Layout Validation

Component Tree (Details):

- Linear Layout
  - txt1
  - gl
    - button
    - button
    - button
    - button
    - button
    - button20
    - button21
    - button22
  - textView
  - button13

Available qualifiers: Country Code, Network Code, Locale, Layout Direction, Smallest Screen Width, Screen Width, Screen Height, Size, Ratio, Orientation, UI Mode, Night Mode, Density

Chosen qualifiers: Nothing to show

(3) OK Cancel

Android I: Project

Morpion > app > src > main > res > menu

activity\_main.xml o.png MainActivity.java

Palette Common Ab TextView Attributes

New

- Kotlin File/Class
- Menu Resource File**
- Sample Data Directory
- File
- Scratch File Ctrl+Alt+Maj+Insérer
- Directory
- C++ Class
- C/C++ Source File
- C/C++ Header File
- Image Asset
- Vector Asset
- Kotlin Script
- Kotlin Worksheet
- AIDL
- Activity
- Automotive
- Folder
- Fragment
- Google
- Other
- Service
- UI Component
- Wear
- Widget
- XML
- EditorConfig File
- Resource Bundle

New Resource File

File name: menu\_main

Root element: menu

Source set: main

Directory name: menu

Available qualifiers:

- Country Code
- Network Code
- Locale
- Layout Direction
- Smallest Screen Width
- Screen Width
- Screen Height
- Size
- Ratio
- Orientation
- UI Mode
- Night Mode

Chosen qualifiers:

Nothing to show

OK Cancel

Device File Explorer

Resource Manager

Gradle

Layout Validation

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Morpion [C:\Users\ousma\AndroidStudioProjects\Morpion] - ...app\src\main\res\layout\activity\_main.xml [app] - Android Studio

Local History Reload from Disk Compare With... Ctrl+D

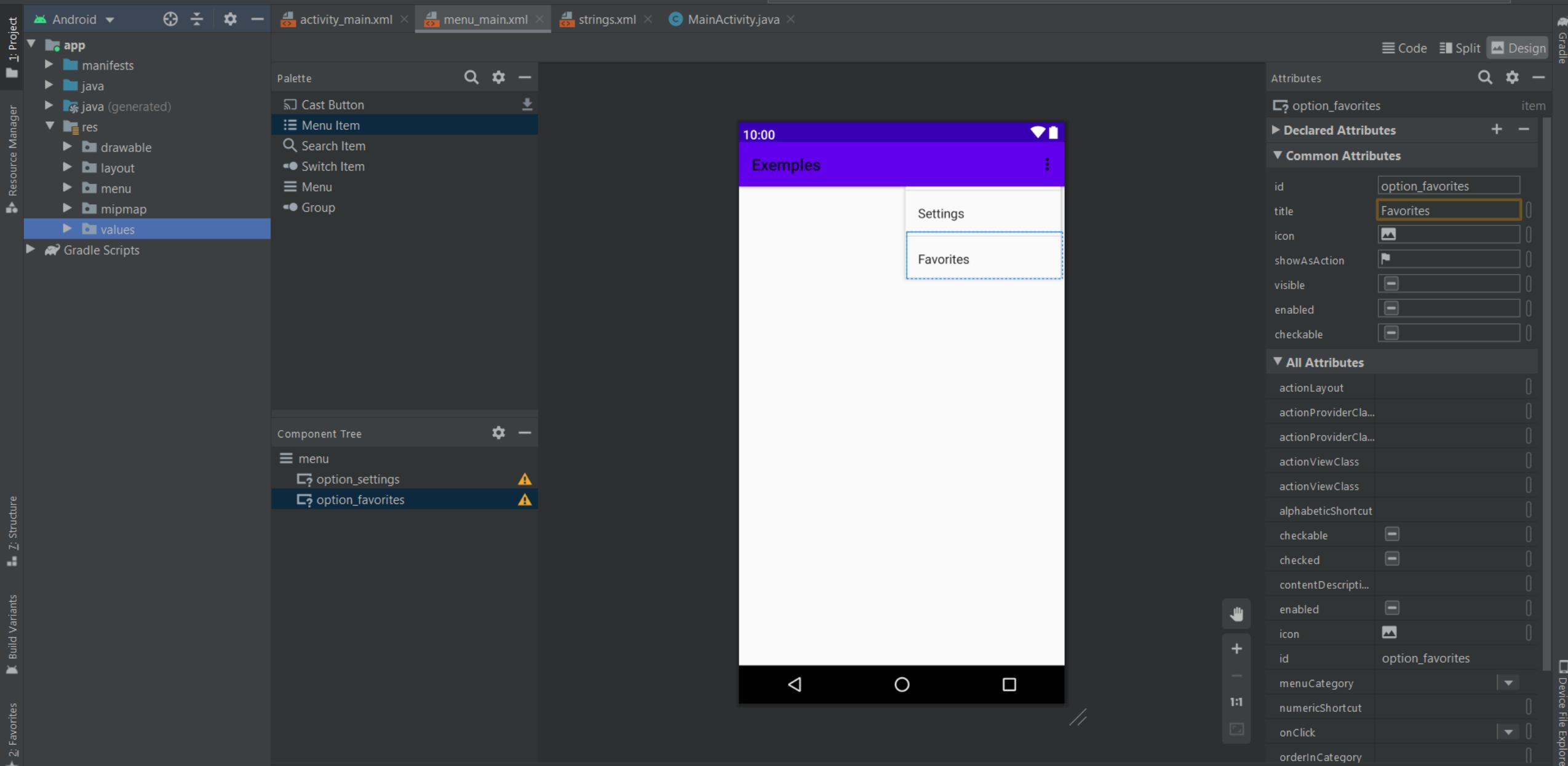
Mark Directory as Remove BOM

Convert Java File to Kotlin File Ctrl+Alt+Maj+K

Create a new M Create Gist...

Event Log Layout Inspector

CRLF UTF-8 4 spaces



# Inflate options menu

Surcharger onCreateOptionsMenu() dans l'Activité

```
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    getMenuInflater().inflate(R.menu.menu_main, menu);  
    return true;  
}
```

1: Project

- app
- manifests
- java
- java (generated)
- res
  - drawable
    - grille.png (v24)
    - ic\_baseline\_settings\_24.xml
    - ic\_launcher\_background.xml
    - ic\_launcher\_foreground.xml
    - o.png
    - vide.jpg
    - x.png
  - layout
  - menu
  - mipmap
  - values
- Gradle Scripts

```
1 package com.example.morpion;
2
3 import ...
4
5
6 public class MainActivity extends AppCompatActivity {
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.activity_main);
11    }
12
13    @Override
14    public boolean onCreateOptionsMenu(Menu menu) {
15        getMenuInflater().inflate(R.menu.menu_main, menu);
16        return true;
17    }
18}
```



Install successfully finished in 270 ms.  
App restart successful without requiring a re-install.

MainActivity > onCreateOptionsMenu()

Run TODO Build Profiler Logcat Terminal

Install successfully finished in 270 ms.: App restart successful without requiring a re-install. (moments ago)

# Ajouter les attributs des items du Menu

- L'attribut `showAsAction` permet de spécifier le comportement de l'élément dans une `ActionBar`, il peut posséder les valeurs suivantes :
  - `ifRoom` : L'élément sera ajouté aux actions principales de l'`ActionBar` si une place est disponible
  - `never` : Ne jamais rajouter l'action aux actions principales de l'`ActionBar`
  - `always` : Toujours rajouter l'action aux actions principales de l'`ActionBar`. Cette valeur n'est pas conseillé car peut entraîner une superposition d'élément (si nombre de place disponible < nombre d'élément ajouté à l'`ActionBar`), préférez la valeur `ifRoom`.
  - `withText` : Toujours afficher le texte représentant l'action

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Morpion [C:\Users\ousma\AndroidStudioProjects\Morpion] - ...app\src\main\res\menu\menu\_main.xml [app] - Android Studio

Morpion > app > src > main > res > menu > menu\_main.xml

activity\_main.xml x menu\_main.xml x strings.xml x MainActivity.java x

Android Project Resource Manager 1: Project 2: Favorites 3: Structure 4: Build Variants

Palette Cast Button Menu Item Search Item Switch Item Menu Group

Component Tree menu option\_settings option\_favorites

Exemples

10:00

Settings

Favorites

Attributes

option\_settings item

Declared Attributes

Common Attributes

id: option\_settings title: Settings icon: showAsAction: vis: Select all Clear always never ifRoom collapseActionView withText

Apply

alphabeticShortcut checkable checked contentDescription enabled icon id: option\_settings menuCategory numericShortcut onClick orderInCategory

Device File Explorer

4: Run TODO Build Profiler 6: Logcat Terminal

Event Log Layout Inspector

Install successfully finished in 270 ms.: App restart successful without requiring a re-install. (2 minutes ago)

28:21 CRLF UTF-8 4 spaces 😊 😐

Morpion > app > src > main > res > menu > menu\_main.xml

activity\_main.xml x menu\_main.xml x strings.xml x MainActivity.java x

INFINIX MOBILITY LIMITED Infinix X650C

Android Project Resource Manager

Resource Manager I: Project

Palette

- Cast Button
- Menu Item**
- Search Item
- Switch Item
- Menu
- Group

Attributes

option\_settings

Declared Attributes

Common Attributes

id	option_settings
title	Settings
icon	[Image]
showAsAction	ifRoom
visible	[unchecked]
enabled	[unchecked]
checkable	[unchecked]

All Attributes

actionLayout	
actionProviderCl...	
actionProviderCl...	
actionViewClass	
actionViewClass	
alphabeticShortcut	
checkable	[unchecked]
checked	[unchecked]
contentDescripti...	
enabled	[unchecked]
icon	[Image]
id	option_settings
menuCategory	[dropdown]
numericShortcut	[dropdown]
onClick	[dropdown]
orderInCategory	[dropdown]

Component Tree

menu

- option\_settings
- optionFavorites

menu

Exemples

SETTINGS

Favorites

10:00

Device File Explorer

Hand + - 1:1

Morpion > app > src > main > res > drawable-v24

activity\_main.xml menu\_main.xml strings.xml MainActivity.java

Android Project Resource Manager Resource Manager 1: Project 2: Favorites 3: Structure 4: Run TC Open Vector Asset Create Gist...

Palette Cast Button Menu Item Search Item

New

- Kotlin File/Class
- Drawable Resource File
- Sample Data Directory
- File
- Scratch File
- Directory
- C++ Class
- C/C++ Source File
- C/C++ Header File
- Image Asset
- Vector Asset
- Kotlin Script
- Kotlin Worksheet
- AIDL
- Activity
- Automotive
- Folder
- Fragment
- Google
- Other
- Service
- UI Component
- Wear
- Widget
- XML
- EditorConfig File
- Resource Bundle

10:00 Exemples SETTINGS Favorites

Attributes

option\_settings

Declared Attributes

Common Attributes

All Attributes

Event Log Layout Inspector

28:21 CRLF UTF-8 4 spaces

Resource Manager

Palette

New

Kotlin File/Class

Drawable Resource File

Sample Data Directory

File

Scratch File

Directory

C++ Class

C/C++ Source File

C/C++ Header File

Image Asset

Vector Asset

Kotlin Script

Kotlin Worksheet

AIDL

Activity

Automotive

Folder

Fragment

Google

Other

Service

UI Component

Wear

Widget

XML

EditorConfig File

Resource Bundle

10:00 Exemples

SETTINGS Favorites

Attributes

option\_settings

Declared Attributes

Common Attributes

All Attributes

Event Log Layout Inspector

28:21 CRLF UTF-8 4 spaces

Morpion > app > src > main > res > drawable-v24

activity\_main.xml menu\_main.xml strings.xml MainActivity.java

Android Project Resource Manager Build Variants Favorites

Palette Cast Button Menu Item Search Item Switch Item Menu Group

Configure Vector Asset

Asset Type: Clip Art Local file (SVG, PSD)

Name: ic\_baseline\_android\_24

Clip Art:

Size: 24 dp X 24 dp

Color: #000000

Opacity: 100 %

Enable auto mirroring for RTL layout

Component Tree

menu

- option\_settings
- option\_favorites

Select Icon

Search: sett Filled All

Icons:

- perm data setting power settings network
- settings
- settings application settings backup restore
- settings bluetooth
- settings brightness
- settings cell
- settings ethernet
- settings input android
- settings input coiled
- settings input connector
- settings input hdmi
- settings input svideo
- settings overscan
- settings phone
- settings power
- settings remote
- settings system
- settings voice

These icons are available under the Apache License Version 2.0

OK Cancel

Attributes

option\_settings

Declared Attributes

Common Attributes

id: option\_settings

title: Settings

isAction: ifRoom

Attributes

Component Tree

menu

- option\_settings
- option\_favorites

Component Tree

menu

- option\_settings
- option\_favorites

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Morpion [C:\Users\ousma\AndroidStudioProjects\Morpion] - ...app\src\main\res\menu\menu\_main.xml [app] - Android Studio

Morpion > app > src > main > res > drawable-v24

activity\_main.xml menu\_main.xml strings.xml MainActivity.java

INFINIX MOBILITY LIMITED Infinix X650C

Android Project Resource Manager Build Variants Favorites

Palette Cast Button Menu Item Search Item Switch Item Menu Group

Asset Studio Configure Vector Asset

Asset Type: Clip Art Local file (SVG, PSD)

Name: ic\_baseline\_settings\_24

Clip Art:

Size: 24 dp X 24 dp

Color: #000000

Opacity: 100 %

Enable auto mirroring for RTL layout

Component Tree menu option\_settings option\_favorites

Vector Drawable

Select Color

R: 184 G: 26 B: 9 RGB # B81A09

Choose Cancel

Device File Explorer

4: Run TODO Build Profiler 6: Logcat Terminal

Install successfully finished in 270 ms.: App restart successful without requiring a re-install. (5 minutes ago)

28:21 CRLF UTF-8 4 spaces

Morpion > app > src > main > res > drawable-v24

activity\_main.xml x menu\_main.xml x strings.xml x MainActivity.java x

INFINIX MOBILITY LIMITED Infinix X650C

Android Project Resource Manager

Palette

Asset Studio

### Configure Vector Asset

Asset Type:  Clip Art  Local file (SVG, PSD)

Name: ic\_baseline\_settings\_24

Clip Art: 

Size: 24 dp X 24 dp

Color: #B81A09

Opacity: 100 %

Enable auto mirroring for RTL layout

Component Tree

- menu
  - option\_settings
  - option\_favorites

Vector Drawable Preview

?

Previous Next Cancel Finish

Code Split Design

Attributes

option\_settings

Declared Attributes

Common Attributes

- id**: option\_settings
- title**: Settings
- icon**: 
- showAsAction**: ifRoom
- visible**:
- enabled**:
- checkable**:

All Attributes

- actionLayout
- actionProviderClass
- actionProviderClass
- actionViewClass
- actionViewClass
- alphabeticShortcut
- checkable:
- checked:
- contentDescription
- enabled:
- icon: 
- id**: option\_settings
- menuCategory
- numericShortcut
- onClick
- orderInCategory

Device File Explorer

Morpion > app > src > main > res > menu > menu\_main.xml

activity\_main.xml x menu\_main.xml x strings.xml x MainActivity.java x

Android Project Resource Manager Resource Manager I: Project

Palette Cast Button Menu Item Search Item Switch Item Menu Group

Attributes option\_settings Declared Attributes Common Attributes

id: option\_settings title: Settings icon: ifRoom visible: enabled: checkable:

Component Tree menu option\_settings option\_favorites

Pick a Resource Module: app Drawable Color Mip Map

app (7)

grille ic\_baseline\_settings\_24 ic\_launcher\_background ic\_launcher\_foreground

Preview

Name: ic\_baseline\_settings\_2... Reference: @drawable/ic\_baseli... Type: Vector Configuration: default

OK Cancel

menu

Morpion > app > src > main > res > menu > menu\_main.xml

Android app INFINIX MOBILITY LIMITED Infinix X650C

1: Project Resource Manager Build Variants Favorites

activity\_main.xml x menu\_main.xml x strings.xml x MainActivity.java x

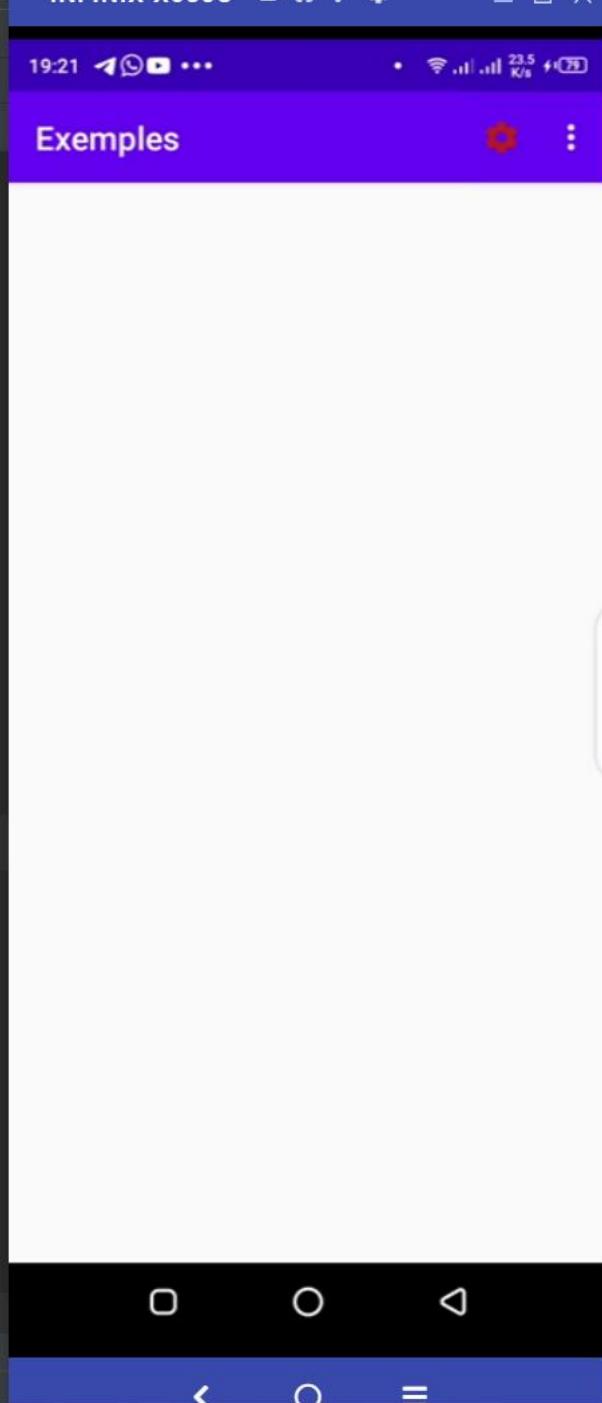
```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/option_settings"
        android:icon="@drawable/ic_baseline_settings_24"
        android:title="Settings"
        app:showAsAction="ifRoom" />
    <item
        android:id="@+id/option_favorites"
        android:title="Favorites" />
</menu>
```

Install successfully finished in 2 s 790 ms.

Run TODO Build Profiler Logcat Terminal

Install successfully finished in 2 s 790 ms. (moments ago)



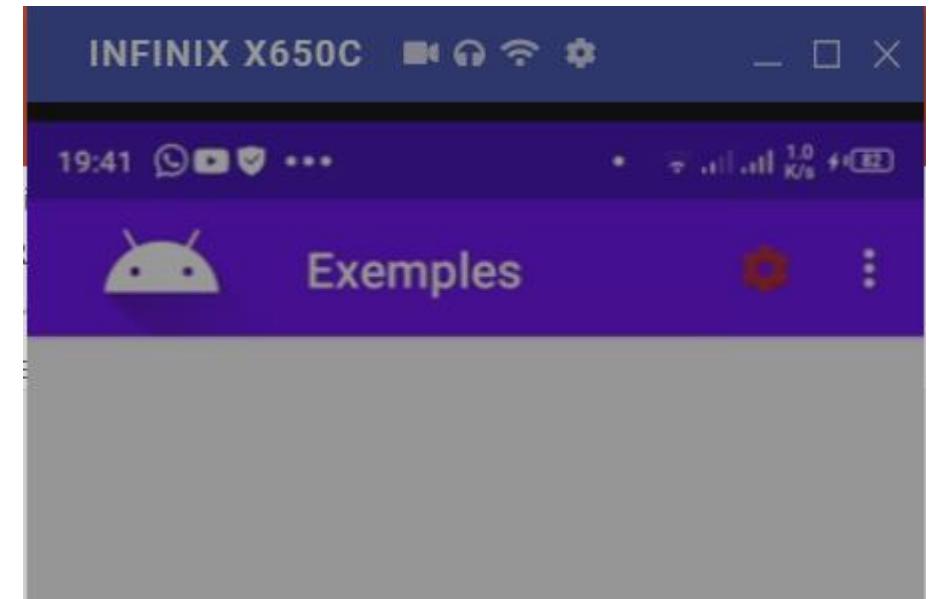
# Override onOptionsItemSelected()

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.option_settings:
            Toast.makeText(context: this, text: "option_settings", Toast.LENGTH_SHORT).show();
            return true;
        case R.id.option_favorites:
            Toast.makeText(context: this, text: "optionFavorites", Toast.LENGTH_SHORT).show();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
```

# Navigation via l'icone de votre application

```
ActionBar actionBar = getSupportActionBar();
actionBar.setDisplayHomeAsUpEnabled(true);
actionBar.setHomeAsUpIndicator(R.drawable.ic_launcher_foreground);

public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            if(drw.isDrawerOpen(Gravity.START)==false)
                drw.openDrawer(Gravity.START);
            else
                drw.closeDrawer(Gravity.START);
            return true;
    }
}
```



Morpion > app > src > main > java > com > example > morpion > MainActivity

activity\_main.xml menu\_main.xml strings.xml MainActivity.java

```
setContentView(R.layout.activity_main);

}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu);
    ActionBar actionBar = getSupportActionBar();
    actionBar.setDisplayHomeAsUpEnabled(true);
    actionBar.setHomeAsUpIndicator(R.drawable.ic_launcher_foreground);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.option_settings:
            Toast.makeText(context, "option_settings", Toast.LENGTH_SHORT).show();
            return true;
        case R.id.option_favorites:
            Toast.makeText(context, "optionFavorites", Toast.LENGTH_SHORT).show();
            return true;
    }
}
```

MainActivity > onOptionsItemSelected()

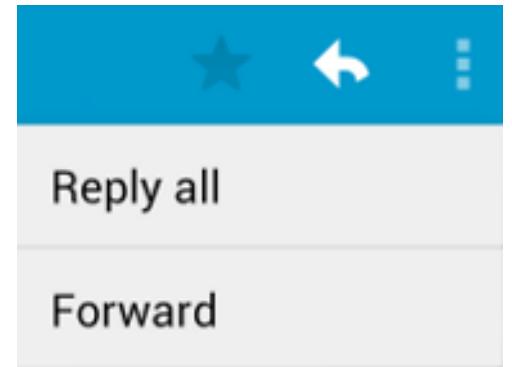
Run TODO Build Profiler Logcat Terminal

Install successfully finished in 2 s 255 ms. (moments ago)



# **POPUPMENU**

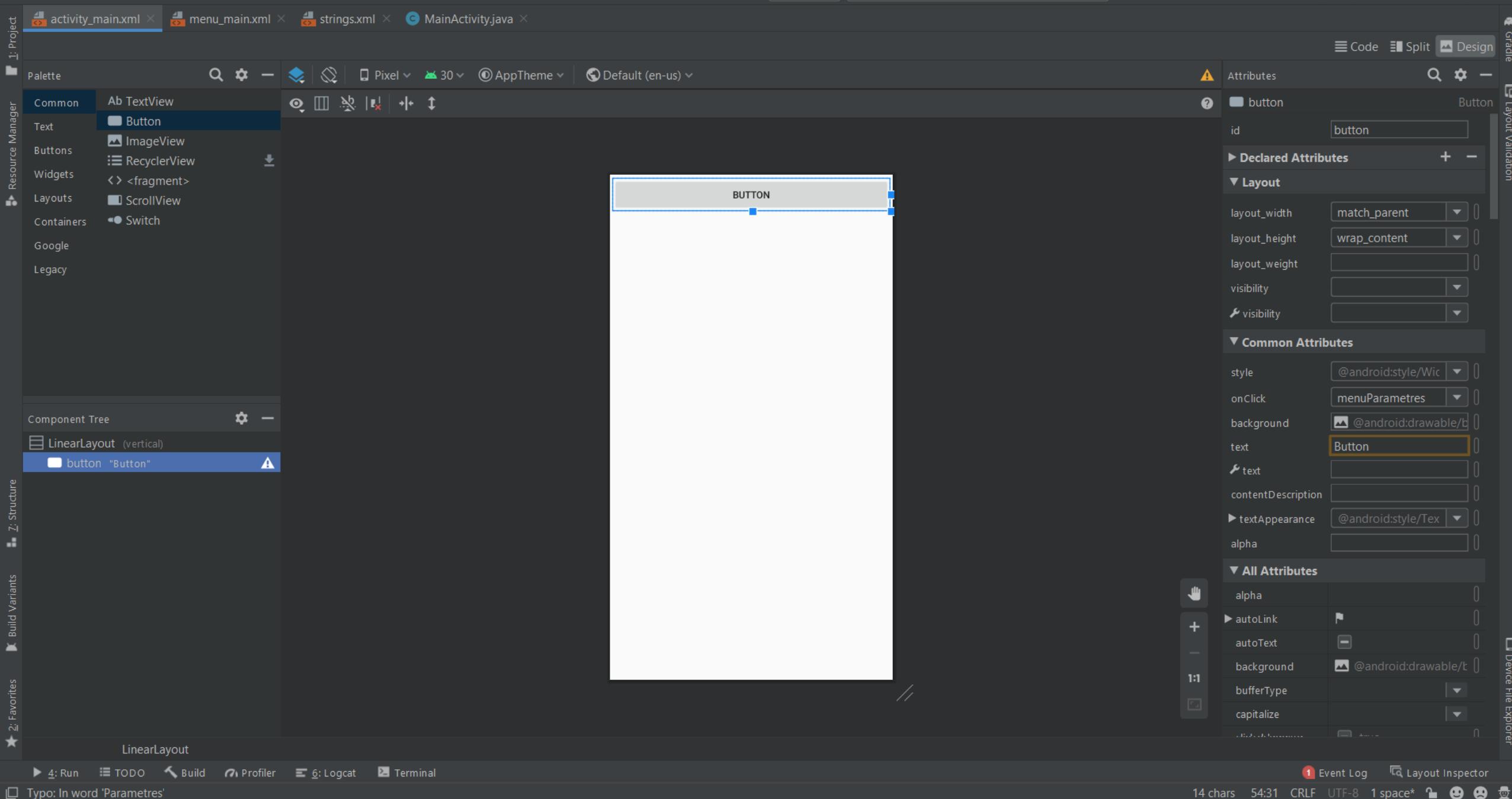
# Qu'est-ce qu'un popup menu?



- Liste verticale d'éléments ancrés à une vue
- Généralement ancré à une icône visible
- Les actions ne doivent pas affecter directement le contenu de la vue
  - Icône du menu Options qui ouvre le menu Options
  - Dans l'application de messagerie, Répondre à tous et Transférer font référence à un message électronique, mais n'affectent ni n'agissent sur le message.

# 1. Etapes

1. Créer un fichier de ressources de menu XML et attribuer des attributs d'apparence et de position
2. Ajouter ImageButton pour l'icône de menu contextuel dans le fichier de présentation d'activité XML
3. Assigner onClickListener à ImageButton
4. Remplacez onClick () pour gonfler la popup et l'enregistrer avec onMenuItemClickListener ()
5. Implémenter onMenuItemClick ()
6. Créer une méthode pour effectuer une action pour chaque élément de menu contextuel



Morpion > app > src > main > java > com > example > morpion > MainActivity

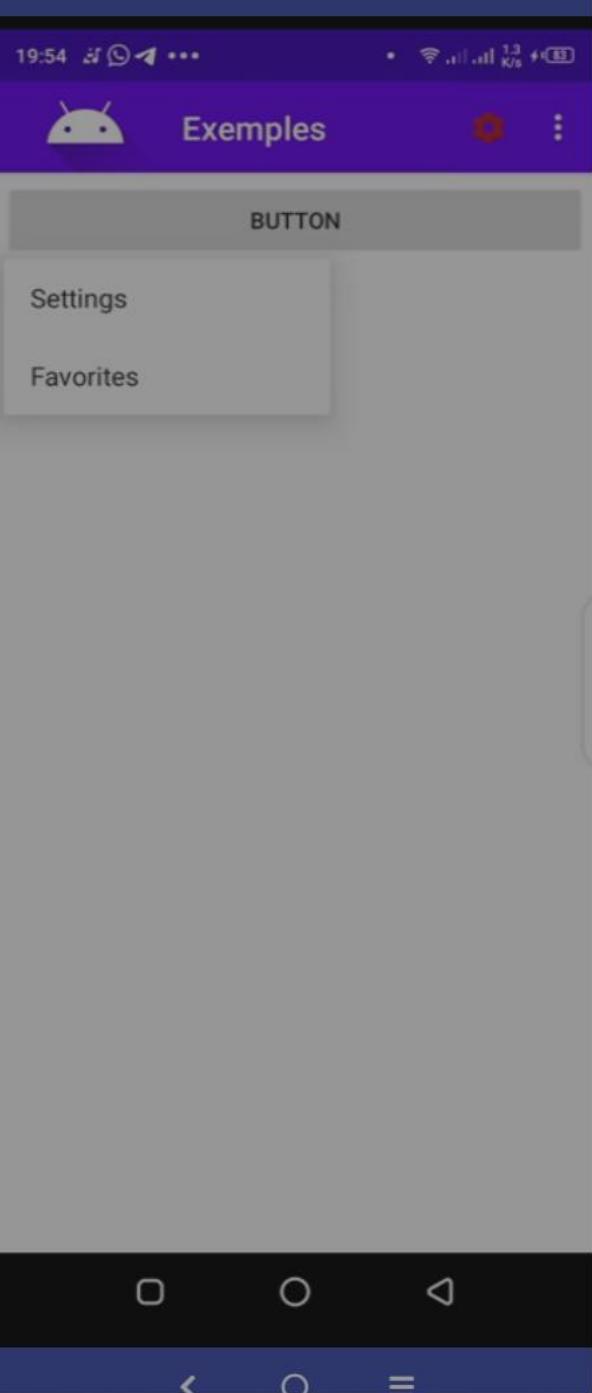
activity\_main.xml menu\_main.xml strings.xml MainActivity.java

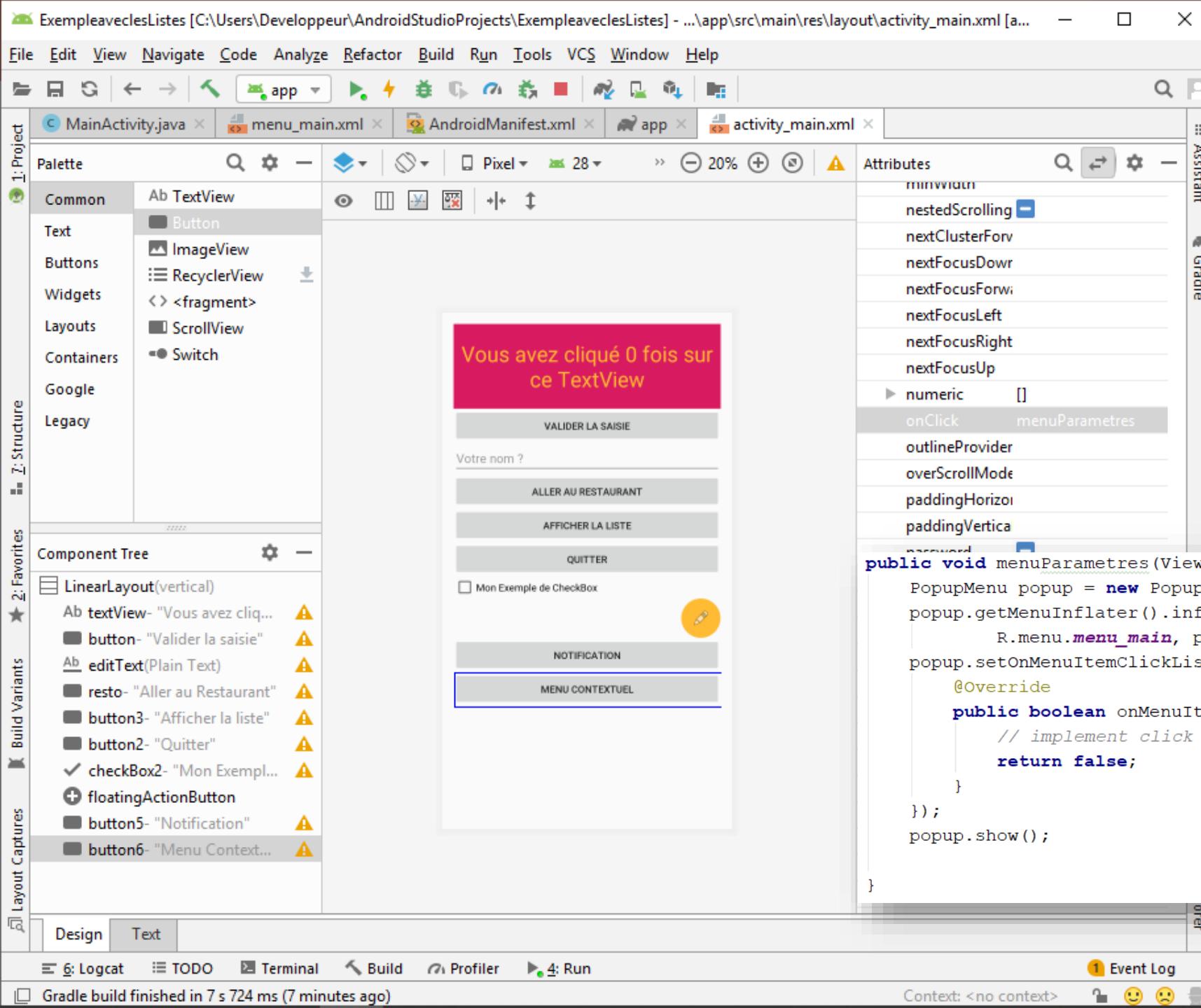
```
default:  
    return super.onOptionsItemSelected(item);  
}  
  
}  
  
}  
  
public void menuParametres(View v) {  
    PopupMenu popup = new PopupMenu(context: MainActivity.this, monButton);  
    popup.getMenuInflater().inflate(R.menu.menu_main, popup.getMenu());  
    popup.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {  
        @Override  
        public boolean onMenuItemClick(MenuItem item) {  
            // implement click listener  
            return false;  
        }  
    });  
    popup.show();  
}  
}
```

MainActivity > onCreateOptionsMenu()

Run TODO Build Profiler Logcat Terminal

Install successfully finished in 4 s 435 ms. (2 minutes ago)





# Autres

- Menus Contextuels
- Barre de menu Contextuelle
- Menu Popup

# **BOITES DE SÉLECTION DE DATES ET HEURES**

# Choix de date et d'heure

- DatePicker
  - android:startYear Pour définir l'année de départ du calendrier
- affiché
  - android:endYear Pour définir l'année de fin du calendrier affiché
  - android:minDate Pour définir la date affichée de départ du calendrier sous la forme mm/jj/aaaa
  - android:maxDate Pour définir la date affichée de fin du calendrier sous la forme mm/jj/aaaa
- TimePicker

# Choix de date et d'heure: exemple

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView android:id="@+id/dateAndTime"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        />
    <Button android:id="@+id/dateBtn"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text= " Régler la date"
        />
    <Button android:id="@+id/timeBtn"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text= "Régler l'heure"
        />
</LinearLayout>
```



# Choix de date et d'heure: exemple

```
public class ChronoDemo extends Activity {
    DateFormat fmtDateAndTime=DateFormat.getDateInstance();
    TextView dateAndTimeLabel;
    Calendar dateAndTime=Calendar.getInstance();
    DatePickerDialog.OnDateSetListener d=new DatePickerDialog.OnDateSetListener() {
        public void onDateSet(DatePicker view, int year, int monthOfYear,
            int dayOfMonth) {
            dateAndTime.set(Calendar.YEAR, year);
            dateAndTime.set(Calendar.MONTH, monthOfYear);
            dateAndTime.set(Calendar.DAY_OF_MONTH, dayOfMonth);
            updateLabel();
        }
    };
    TimePickerDialog.OnTimeSetListener t=new TimePickerDialog.OnTimeSetListener() {
        public void onTimeSet(TimePicker view, int hourOfDay,
            int minute) {
            dateAndTime.set(Calendar.HOUR_OF_DAY, hourOfDay);
            dateAndTime.set(Calendar.MINUTE, minute);
            updateLabel();
        }
    };
}
```

# Choix de date et d'heure: exemple

```
@Override  
public void onCreate(Bundle icicle) { super.onCreate(icicle);  
setContentView(R.layout.main);  
  
Button btn=(Button)findViewById(R.id.dateBtn);  
  
btn.setOnClickListener(new View.OnClickListener() { public void onClick(View v) {  
new DatePickerDialog(ChronoDemo.this, d,  
dateAndTime.get(Calendar.YEAR), dateAndTime.get(Calendar.MONTH),  
dateAndTime.get(Calendar.DAY_OF_MONTH)).show();  
}  
});  
  
btn=(Button)findViewById(R.id.timeBtn);  
  
btn.setOnClickListener(new View.OnClickListener() { public void onClick(View v) {  
new TimePickerDialog(ChronoDemo.this, t,  
dateAndTime.get(Calendar.HOUR_OF_DAY), dateAndTime.get(Calendar.MINUTE), true).show();  
}  
});  
  
dateAndTimeLabel=(TextView)findViewById(R.id.dateAndTime); updateLabel();  
}  
  
private void updateLabel() { dateAndTimeLabel.setText(fmtDateAndTime  
.format(dateAndTime.getTime()));  
}
```

# ProgressBar

- Deux comportements selon que l'on connaît ou pas la valeur maximale:  
`android:indeterminate` Pour définir le type de progressBar (true=indéterminé, false=déterminé).
- Animation (si indéterminé): `android:indeterminateBehavior="i"` (où i peut être : `repeat` ou `cycle`) définit le comportement de l'animation pour le type circulaire (repeat=recommence l'animation, cycle=changer le sens de l'animation)
- Dimensions
  - `android:maxHeight="unité"`
  - `android:minHeight="unité"`
  - `android:maxWidth="unité"`
  - `android:minWidth="unité"`
- Valeurs (si déterminé)
  - `android:max` Pour définir la valeur maximale
  - `android:progress` Pour définir la valeur initiale
  - `android:secondaryProgress` Pour définir une valeur secondaire (par exemple celle d'un buffer comme on le voit sur des vidéos en streaming)

# Formes des ProgressBar

- En l'absence de paramètre `style` la forme est circulaire
- Pour obtenir d'autres formes on utilise le paramètre `style` :

– `style="?android:attr/s"` où s peut être :

- `progressBarStyleHorizontal`
- `progressBarStyleSmall`
- `progressBarStyleLarge`



- On ne peut pas changer la couleur

## SeekBar



C'est un ProgressBar sous forme de barre horizontale dotée d'un curseur permettant de modifier la valeur si on a choisi `android:indeterminate="false"` sinon le curseur ne marche pas et la barre bouge sans arrêt.

# Exemple ProgressBar

```
<ProgressBar android:id="@+id/progres"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:minWidth="300px"
    android:minHeight="30px"
    android:max="100"
    android:progress="30"
    android:secondaryProgress="40"
    android:indeterminate="false"
    style="?android:attr/progressBarStyleHorizontal"/>
```

# RatingBar



Paramètres :

`android:isIndicator` Pour indiquer si l'utilisateur peut modifier la valeur ou pas (true= non modifiable)

`android:numStars` Pour définir le nombre d'étoiles affichées

`android:rating` Pour définir la position initiale

`android:stepSize` Pour définir le pas de progression (on peut colorier des ¼ d'étoiles par exemple)

# Horloges et Chronomètres

AnalogClock



DigitalClock

4:58:17 pm

Chronometer

Ce cours a commencé depuis 51:45 déjà

[android:format](#)="f" (où f est une chaîne dans laquelle la première occurrence de %s sera remplacée par la valeur du chronomètre sous la forme MM:SS ou H:MM:SS)

# AnalogClock

```
<AnalogClock android:id="@+id/horloge"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
/>
```

# Sélection Date/heure: Date

- Android supporte également les widgets (**DatePicker**, **TimePicker**) et boîtes de dialogue (**DatePickerDialog**, **TimePickerDialog**) pour aider les utilisateurs entrer des dates et des heures.
- Le **DatePicker** et **DatePickerDialog** permettent de régler la date de début de sélection, sous la forme d'une année, mois et jour. Valeur du mois traverse de 0 pour janvier 11 pour le 2 décembre décembre.
- Chaque widget fournit un objet de rappel **OnDateChangedListener** ou **OnDateSetListener**) où vous êtes informé d'une nouvelle date sélectionnée par l'utilisateur.

# Sélection Date/heure: Heure

**TimePicker** et **TimePickerDialog** permettent de:

1. régler le temps initial, l'utilisateur peut ajuster, sous la forme d'une **heure** (0 à 23) et une **minute** (0 à 59)
2. *indiquer si la sélection doit être en mode 12-heure* (AM/PM), ou **mode 24-heure**.
3. fournir un objet de rappel (**OnTimeChangedListener** or **OnTimeSetListener**) pour être averti des lorsque l'utilisateur a choisi une nouvelle fois (ce qui vous est fournie sous la forme d'une heure et minute)

# Sélection Date/heure



# Sélection Date/heure: exemple

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:id="@+id/widget28"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android"
>
    <TextView
        android:id="@+id/lblDateAndTime"
        android:layout_width="fill_parent"
        android:layout_height="47px"
        android:background="#ff000099"
        android:textStyle="bold"
    >
    </TextView>
    <Button
        android:id="@+id/btnDate"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Set the Date"
    >
    </Button>
    <Button
        android:id="@+id/btnTime"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Set the Time"
    >
    </Button>
</LinearLayout>
```



# Sélection Date/heure: exemple

```
package cis493.demoui;
import android.app.Activity;
import android.os.Bundle;
import android.app.DatePickerDialog;
import android.app.TimePickerDialog;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.TimePicker;
import android.widget.TextView;
import java.text.DateFormat;
import java.util.Calendar;

public class AndDemoUI extends Activity {
    DateFormat fmtDateAndTime = DateFormat.getDateInstance();
    TextView lblDateAndTime;
    Calendar myCalendar = Calendar.getInstance(); ←

    DatePickerDialog.OnDateSetListener d = new DatePickerDialog.OnDateSetListener()
    {
        public void onDateSet(DatePicker view,
                             int year, int monthOfYear, int dayOfMonth) {
            myCalendar.set(Calendar.YEAR, year);
            myCalendar.set(Calendar.MONTH, monthOfYear);
            myCalendar.set(Calendar.DAY_OF_MONTH, dayOfMonth);
            updateLabel();
        }
    };
}
```



# Sélection Date/heure: exemple

```
TimePickerDialog.OnTimeSetListener t = new TimePickerDialog.OnTimeSetListener()
{
    public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
        myCalendar.set(Calendar.HOUR_OF_DAY, hourOfDay);
        myCalendar.set(Calendar.MINUTE, minute);
        updateLabel();
    }
};

private void updateLabel() {
    lblDateAndTime.setText(fmtDateAndTime.format(myCalendar.getTime()));
}
```



# Sélection Date/heure: exemple

```
@Override
public void onCreate(Bundle icicle) {
    super.onCreate(icicle);
    setContentView(R.layout.main);
    lblDateAndTime = (TextView) findViewById(R.id.lblDateAndTime);
    Button btnDate = (Button) findViewById(R.id.btnDate);
    btnDate.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            new DatePickerDialog(AndDemoUI.this, d,
                myCalendar.get(Calendar.YEAR),
                myCalendar.get(Calendar.MONTH),
                myCalendar.get(Calendar.DAY_OF_MONTH)).show();
        }
    });

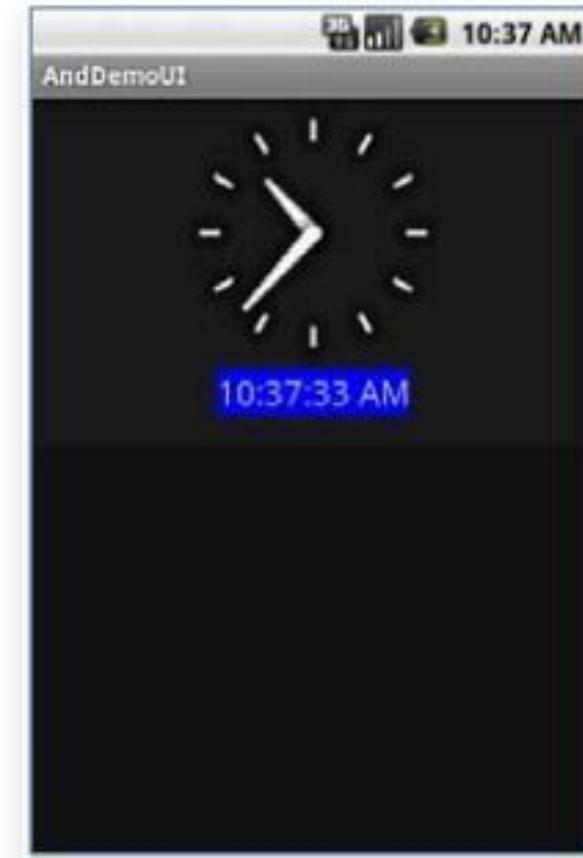
    Button btnTime = (Button) findViewById(R.id.btnTime);
    btnTime.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            new TimePickerDialog(AndDemoUI.this, t,
                myCalendar.get(Calendar.HOUR_OF_DAY),
                myCalendar.get(Calendar.MINUTE), true).show();
        }
    });

    updateLabel();
} // onCreate
} // class
```

# Sélection Date/heure: DigitalClock et AnalogClock

Mettre à jour automatiquement le temps (sans intervention de l'utilisateur est requise).

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    android:id="@+id/widget34"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android"
>
    <DigitalClock
        android:id="@+id/digital"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#ff0000ff"
        android:textSize="20px"
        android:layout_below="@+id/analog"
        android:layout_centerHorizontal="true"
    >
    </DigitalClock>
    <AnalogClock
        android:id="@+id/analog"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
    >
    </AnalogClock>
</RelativeLayout>
```



# Onglets

- Une Activity peut être subdivisée en plusieurs onglets
- Afin d'utiliser des onglets, il vous faut les prochaines pièces:
  - TabHost : le conteneur surplombant les boutons d'onglets et leur contenus
  - TabWidget : la rangée de boutons d'onglets qui contient le texte
    - et optionnellement les icônes
  - FrameLayout : le conteneur pour les contenus des onglets, chaque onglet est un enfant du FrameLayout

# Onglets : quelques règles

Afin que les onglets fonctionnent correctement, il faut suivre certaines règles:

- Le TabWidget doit avoir une **android:id qui est @android:id/tabs**
- Il est conseillé de mettre du padding entre le FrameLayout et les tab buttons, afin que le contenu ne «colle» pas aux tab buttons.
- Il est possible d'utiliser une TabActivity, mais il faut alors donner au TabHost une **android:id of @android:id/tabhost**
- TabActivity regroupe de la fonctionnalité propre aux onglets et facilite l'usage d'onglets

# Exemple de layout avec onglets

```
<TabHost android:id="@+id/tabhost"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <LinearLayout
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">
        <TabWidget android:id="@+id/tabs"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
        />
        <FrameLayout android:id="@+id/tabcontent"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent">
            <AnalogClock android:id="@+id/tab1"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:layout_centerHorizontal="true"
            />
            <Button android:id="@+id/tab2"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent"
                android:text="A semi-random button"
            />
        </FrameLayout>
    </LinearLayout>
</TabHost>
```

# Onglets : comment faire

- Il est impératif d'utiliser du code Java afin de dire au **TabHost quelle view** représente le contenu d'un bouton mais aussi à quoi ressemblera un bouton
- Pour ceci, nous utilisons un object **TabSpec**: en utilisant la fonction **newTabSpec() sur le TabHost**
- Il y a ensuite deux fonctions clés:
  - **setContent()** pour indiquer le contenu d'un onglet
  - android:id de la view
  - ou un Intent (contenu d'une autre Activity, par exemple)
- **setIndicator()**
  - texte pour l'onglet
  - image drawable pour l'icone
  - view
- When tabs are ready, call the `setup()` method on the tabhost and add the tabs

# Onglets : exemple d'activity

```
public class TabDemo extends Activity {  
    @Override  
    public void onCreate(Bundle icicle) {  
        super.onCreate(icicle);  
        setContentView(R.layout.main);  
        TabHost tabs=(TabHost)findViewById(R.id.tabhost);  
  
        tabs.setup();  
  
        TabHost.TabSpec spec=tabs.newTabSpec("tag1");  
  
        spec.setContent(R.id.tab1);  
        spec.setIndicator("Clock");  
        tabs.addTab(spec);  
  
        spec=tabs.newTabSpec("tag2");  
        spec.setContent(R.id.tab2);  
        spec.setIndicator("Button");  
        tabs.addTab(spec);  
    }  
}
```



# Exercice 3

- Créez une application avec deux onglets
  - Le premier onglet reprend le layout créé précédemment avec «pierre, papier, ciseaux» créez précédemment
  - Le deuxième onglet contient une TextView qui affiche «ceci est onglet 2»

# Développement d'Applications Natives avec Android

## GÉRER LA CONNEXION INTERNET





Exemple ave...



# Gérer la connexion réseau internet

You avez cliqué 0 fois  
sur ce TextView

VALIDER LA SAISIE

Votre nom ?

ALLER AU RESTAURANT

AFFICHER LA LISTE

QUITTER

Mon Exemple de CheckBox



You êtes connecté par WIFI. Les infos  
sont "OntologyClouding"

MENU CONTEXTUEL

# Required Permissions

```
<uses-permission android:name="android.permission.INTERNET" />
```

```
<uses-permission  
    android:name="android.permission.ACCESS_NETWORK_STATE" />
```

# Récupérer les informations du réseau

- ConnectivityManager
  - Répond à des questions sur l'état de la connectivité réseau
  - Notifie les applications lorsque la connectivité réseau change
- NetworkInfo
  - Décrit le statut d'une interface réseau d'un type donné
  - Mobile ou Wi-Fi

# Vérifier si le réseau est disponible

```
ConnectivityManager connMgr =  
(ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);  
NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();  
String str="Aucune connexion disponible !!!";  
if (networkInfo != null && networkInfo.isConnected()) {  
    str="Vous êtes connecté par " + networkInfo.getTypeName()  
        +". Les infos sont"+networkInfo.getExtraInfo();  
}  
Toast.makeText(getApplicationContext(),str,Toast.LENGTH_LONG).show();
```

# Vérifier les réseaux WiFi et Mobile

```
NetworkInfo networkInfo =  
    connMgr.getNetworkInfo(ConnectivityManager.TYPE_WIFI);  
boolean isWifiConn = networkInfo.isConnected();  
  
networkInfo =  
    connMgr.getNetworkInfo(ConnectivityManager.TYPE_MOBILE);  
boolean isMobileConn = networkInfo.isConnected();
```

# Utiliser les Thread

- AsyncTask—tâche très courte ou aucun résultat renvoyé à l'interface utilisateur
- AsyncTaskLoader—pour des tâches plus longues, renvoie le résultat à l'interface utilisateur
- Background Service
- N'implémentez pas de codes liés au réseau dans le thread d'interface utilisateur!

# AsyncTask

- Seul un thread d'interface utilisateur peut créer et démarrer une AsyncTask.
- execute (): démarre AsyncTasks en contexte série
- Concurrent avec le thread d'interface utilisateur, mais séquentiel avec d'autres threads de travail.
- executeOnExecutor (AsyncTask.THREAD\_POOL\_EXECUTOR)
- En même temps que l'interface utilisateur et les threads de travail.

# Travaux Pratiques

## Faire le TP n°3

Ajouter Exos Fonts, Thèmes Styles, Date, Menus, Boîte de dialogue, wifi...