École Centrale Lyon

UE INF
LaTeX Approfondi
Rapport

Rapport BE2 : Convertisseur de devise

Élèves :
Tanguy Guyot

Enseignant :
Stéphane Derrode

7 novembre 2022

Table des matières

1 Introduction

Pour cette BE notée, j'ai choisi de faire un convertisseur de devise. Ce convertisseur de devise est une application Android appelée "Conversione", ce qui signifie Conversion en italien. L'application est d'inspiration italienne par son nom, le bouton de conversion, le logo, les couleurs etc...

Ci-dessous se trouve le canevas original de l'application tel qu'elle a été imaginée.

FIGURE 1 – Logo de l'application

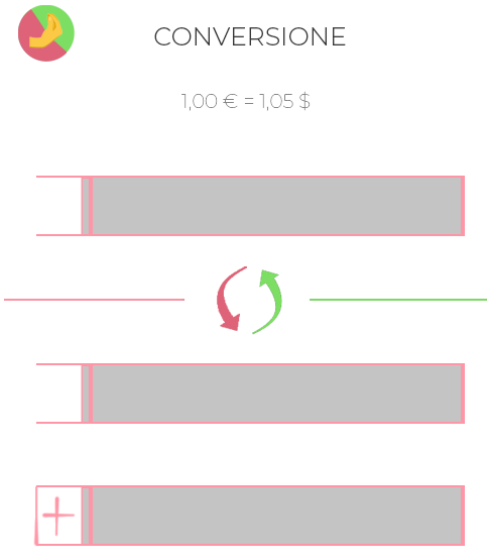

FIGURE 2 – Schéma de l'application

2 Présentation de l'application

L'application contient trois activités principales : une pour la conversion, une pour un menu répertoriant les activités, et une rubrique "A propos..." pour avoir des informations sur l'application et des coordonnées de contact.

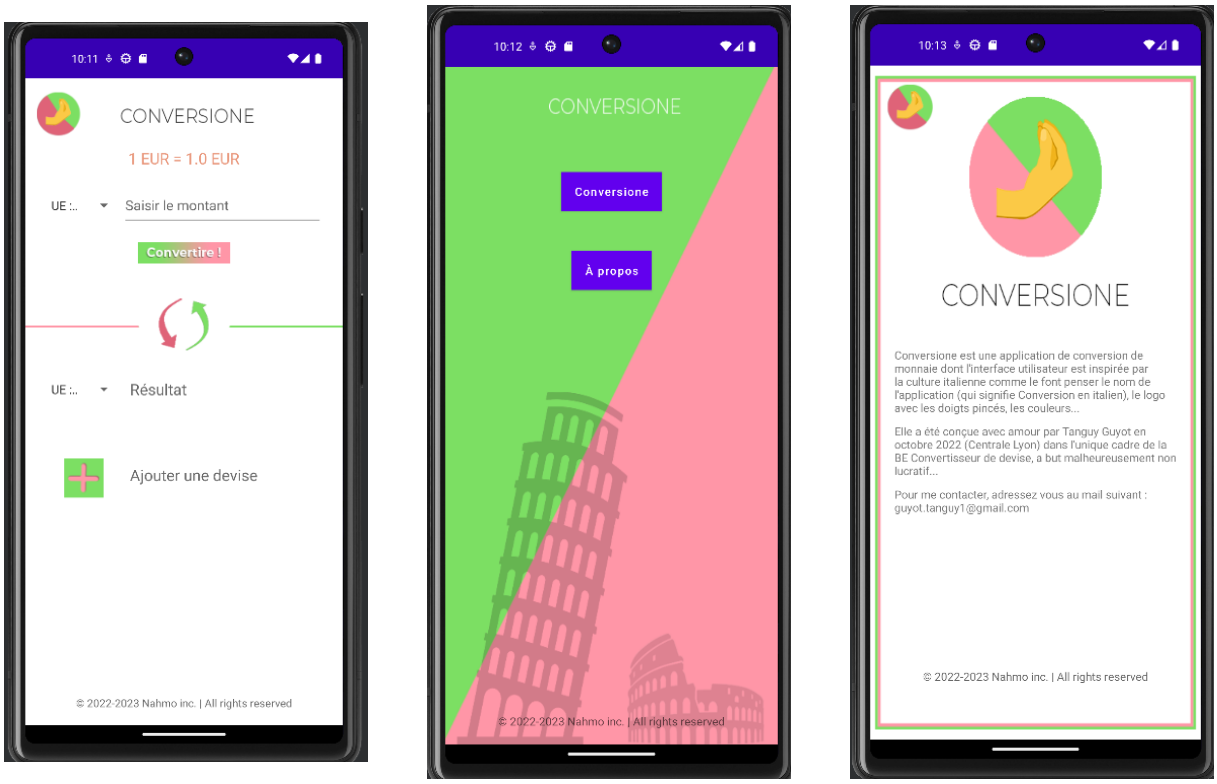

FIGURE 3 – Les trois activités disponibles à l'utilisateur dans l'application

2.1 Activité Conversione

L'activité "Conversione" est l'activité principale de l'application, qui s'affiche à l'ouverture. Elle permet de convertir une monnaie sélectionnée à l'aide d'un spinner pour la saisie et d'un autre spinner pour le résultat. On obtient le résultat en cliquant sur "Convertire !" (ce qui signifie Convertir en italien). Le bouton central permet d'intervertir les devises choisies entre celle saisie et le premier résultat. Un bouton "+" permet également d'ajouter d'autre devise et d'ainsi observer jusqu'à 3 résultats dans des devises différentes pour une seule saisie.

Le texte en orange permet d'observer en direct la conversion d'une unité de la devise choisie dans le spinner de saisie, en celle du premier résultat.

Enfin, le logo en haut à gauche est un bouton qui permet d'ouvrir le menu contenant l'ensemble des activités accessibles à l'utilisateur.

2.2 Activité À propos

L'activité À propos est une activité permettant de donner de plus amples informations sur l'application et son fonctionnement. Il y a aussi les coordonnées pour contacter l'auteur de l'application. Elle a aussi servi à faire en sorte de rajouter du contenu et notamment un menu dans l'application, de la plus-value.

Il est encore une fois possible de cliquer sur le logo en haut à gauche pour revenir au menu.

2.3 Activité Menu

L'activité Menu regroupe deux ImageButton permettant d'accéder à l'une ou l'autre des activités précédemment décrites.

3 Code source de l'application

3.1 Ressources (drawable)

Dans le dossier drawable des ressources, ont été ajouté tous les designs présents dans l'application, à savoir (entre autres) :

- le logo en ldpi, mdpi, hdpi, xhdpi, xxhdpi et xxxhdpi ;
- le bouton plus ;
- le bouton convertir ;
- le bouton de transposition ;
- l'arrière-plan de l'activité de conversion (en xhdpi uniquement) ;
- l'arrière-plan du menu ;
- l'arrière-plan de la section à propos ;
- les drapeaux des pays (prévu pour les spinners mais non utilisés, cf la conclusion).

3.2 Fichiers XML

3.2.1 *activity_main.xml*

C'est le fichier xml le plus complet du projet. Il contient l'ensemble des éléments contenus sur la page principale. Le Layout utilisé est un RelativeLayout. Les éléments de l'activité sont ensuite répartis à partir de plusieurs LinearLayout contenant des éléments de l'activité (boutons, TextView, EditText...). Tous les boutons sont des ImageButton dont la source est issue du dossier drawable.

Des Spinners sont également utilisés et permettent de faire apparaître un menu déroulant avec les pays disponibles, mais l'utilisation de Spinners nécessite la création d'une liste de pays et d'ArrayAdapter, à voir ci-dessous dans MainActivity.java.

3.2.2 *activity_info.xml*

Ce fichier xml contient essentiellement l'ImageButton du menu et des TextView centrés avec marges pour écrire toutes les informations nécessaires à cette section.

3.2.3 *activity_menu.xml*

Cette section contient uniquement deux simples boutons Button et un TextView tout en bas.

3.3 Fichiers Java (activités et classes)

3.3.1 pays.java

La classe pays a été crée afin de catégoriser les pays avec leur nom, leur devise, leur symbole, et un taux de conversion en euro. On peut accéder à chacun de ces paramètres

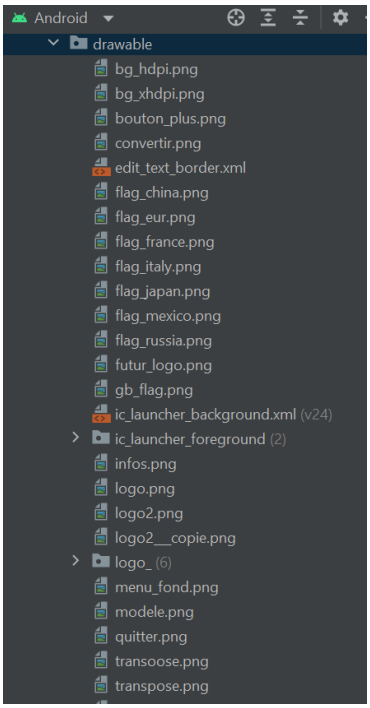

FIGURE 4 – Le dossier drawable

avec des fonctions.

```java
package com.example.conversione;

public class pays {
    private String nomPays;
    private String monnaiePays;
    private String symboleMonnaie;
    private double EURConversion;

    public pays(String nom, String monnaie, String symbole, double Conv) {
        nomPays = nom;
        monnaiePays = monnaie;
        symboleMonnaie = symbole;
        EURConversion = Conv;
    }

    public String getNom() { return nomPays; }
    public String getMonnaie() { return monnaiePays; }
    public String getSymbole() { return symboleMonnaie; }
    public double getEURConversion() { return EURConversion; }
    @Override
    public String toString()  {
        return this.getNom() + " : " + this.getMonnaie() + " " + this.getSymbole();
    }
}
```

FIGURE 5 – La classe pays

3.3.2 paysData.java

paysData est une classe servant uniquement à lister tous les pays que l'on souhaite mettre dans l'application. Ainsi, il suffit juste de rajouter un pays dans la liste et il apparaîtra fonctionnel dans l'application.

```java
package com.example.conversione;

public class paysData {

    public static pays[] getPays() {
        pays UE = new pays( nom: "UE",  monnaie: "EUR",  symbole: "€",  Conv: 1.0000);
        pays US = new pays( nom: "Etats-Unis",  monnaie: "USD",  symbole: "$",  Conv: 1.0400);
        pays UK = new pays( nom: "Royaume-Uni",  monnaie: "GBP",  symbole: "£",  Conv: 1.1400);
        pays CN = new pays( nom: "Chine",  monnaie: "CNY",  symbole: "€",  Conv: 0.1400);
        pays JP = new pays( nom: "Japon",  monnaie: "JPY",  symbole: "€",  Conv: 0.0068);
        pays MX = new pays( nom: "Mexique",  monnaie: "MXN",  symbole: "$",  Conv: 0.0510);
        pays IT = new pays( nom: "Italie",  monnaie: "ITL",  symbole: "£",  Conv: 0.0005);
        pays OP = new pays( nom: "One Piece",  monnaie: "BERRY",  symbole: "B",  Conv: 0.0232);
        return new pays[] {UE, US, UK, CN, JP, MX, IT, OP};
    }
}
```

FIGURE 6 – La classe paysData

```java
@Override
public void onClick(View view) {
    double tauxConvSaisie = listePaysSaisie.getEURConversion(); // c
    double tauxConvResult1 = listePaysResult1.getEURConversion(); // k
    double tauxConvResult2 = listePaysResult2.getEURConversion(); // k2
    double tauxConvResult3 = listePaysResult3.getEURConversion(); // k3
    // formule conversion : c/k
    String montantStr = Saisie.getText().toString();

    if (montantStr.length() == 0) {
        Toast.makeText(getApplicationContext(), text: "Veuillez entrer un montant",
                Toast.LENGTH_SHORT).show();
    } else {
        double montant = Double.parseDouble(montantStr);
        double resultConv1 = arrondisseur( val: montant * (tauxConvSaisie/tauxConvResult1));
        String resultConv1Str = String.valueOf(resultConv1);
        double resultConv2 = arrondisseur( val: montant * (tauxConvSaisie/tauxConvResult2));
        String resultConv2Str = String.valueOf(resultConv2);
        double resultConv3 = arrondisseur( val: montant * (tauxConvSaisie/tauxConvResult3));
        String resultConv3Str = String.valueOf(resultConv3);

        TextResult1.setText(resultConv1Str);
        TextResult2.setText(resultConv2Str);
        TextResult3.setText(resultConv3Str);
```

FIGURE 7 – La fonction onClick relative au bouton "Convertire !"

3.3.3 MainActivity.java

Le fichier MainActivity.java configure l'ensemble des éléments du fichier *activity_main.xml*.

Bouton Menu

Nous mettons un OnClickListener sur le bouton correspondant au menu, et à l'action onClick, cela déclenche la fonction ouvrirMenu(), qui créer un Intent permettant d'ouvrir Menu.class et donc l'activité relative au menu.

Action de conversion

À l'aide d'un OnClickListener, on déclenche la fonction onClick en cliquant sur le bouton. Cette fonction va détecter si un montant (type double) a été tapé ou non, et si oui, il va effectuer la conversion grâce à une formule correspondant au rapport entre les deux taux de conversion en euro des pays. Il est ensuite affiché dans les TextView des résultats.

Bouton Plus Les ImageButton plus permettent de rajouter une devise, au maximum 3 simultanément. Pour cela, il s'agit de créer les composants à leur place, et dans l'initialisation d'onCreate, les rendre invisibles. En cliquant sur le bouton Plus, on rend invisible le bouton et le TextView pour rendre visible les composants relatifs au résultat 2 (ou 3).

Les Spinners

Pour configurer les Spinners, il faut créer pour chaque Spinner un ArrayAdapter qui va contenir la liste des pays issue de paysData.java. Il faut ensuite attribuer ces adapteurs au Spinner et ensuite mettre un listener OnItemSelectedListener contenant deux fonctions.

Dans la fonction onItemSelected, on déclenche une autre fonction qui fait ce que l'on souhaite lorsque l'on sélectionne un item du Spinner.

```
// SPINNER 1 SAISIE

Spinner saisieSpin = (Spinner) findViewById(R.id.saisie_pays);
ArrayAdapter<pays> SaisieAdapter = new ArrayAdapter<~>( context: this,
        android.R.layout.simple_spinner_item, listePays);
SaisieAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
saisieSpin.setAdapter(SaisieAdapter);

saisieSpin.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {

    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
        onItemSelectedHandlerSaisie(parent, view, position, id);
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {

    }
});
```

FIGURE 8 – Le Spinner pour le champ de saisie

```
260 @  private void onItemSelectedHandlerSaisie(AdapterView<?> adapterView, View view, int position, long id) {
261        Adapter adapter = adapterView.getAdapter();
262        listePaysSaisie = (pays) adapter.getItem(position);
263        positionResult1 = position;
264
265        //arrondissement à 2 décimales
266        TextView Comparaison2 = (TextView) findViewById(R.id.comparaison);
267        double tauxConvSaisie = (double) listePaysSaisie.getEURConversion();
268        double tauxConvResult1 = listePaysResult1.getEURConversion();
269        double facteur = arrondisseur( val: tauxConvSaisie/tauxConvResult1);
270
271        Comparaison2.setText("1 " + listePaysSaisie.getMonnaie() + " = " + Double.toString(facteur) +
272                " " + listePaysResult1.getMonnaie());
273    }
```

FIGURE 9 – La fonction onItemSelectedHandlerSaisie

La fonction onItemSelectedHandlerSaisie permet de repérer quel item est sélectionné, mais aussi à mettre à jour en direct la conversion unitaire du haut en orange (de même pour la fonction du Spinner du premier résultat).

Bouton Transposition Nous appliquons un OnClickListener sur le bouton, dont la fonction onClick interpose la position du Spinner du premier résultat et celle du Spinner de saisie.

```
105     // BOUTON DE TRANSPOSITION
106
107     Transposition.setOnClickListener(new View.OnClickListener() {
108         @Override
109         public void onClick(View view) {
110             int SaisiePos = positionSaisie;
111             int Result1Pos = positionResult1;
112             saisieSpin.setSelection(SaisiePos);
113             result1Spin.setSelection(Result1Pos);
114
115             Toast.makeText(getApplicationContext(),  text: "Transposition effectuée",
116                     Toast.LENGTH_SHORT).show();
117         }
118     });
```

FIGURE 10 – Configuration du bouton de transposition

3.3.4 Menu.java

Menu.java est simplement constitué de boutons avec des OnClickListener permettant de passer à l'une ou l'autre des activités avec des intentions Intent. Quelques animations ont également été ajoutées.

```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_menu);

    Button section_conversion = findViewById(R.id.section_conversion);
    Button section_infos = findViewById(R.id.section_infos);

    section_conversion.setAlpha(0f);
    section_infos.setAlpha(0f);

    section_conversion.animate().alpha(1f).setDuration(1500);
    section_infos.animate().alpha(1f).setDuration(1500);

    section_infos.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            allerInfos();
        }
    });
    section_conversion.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) { allerConversione(); }
    });
}
```

FIGURE 11 – Activité Menu.java

3.3.5 infos.java

Cette activité contient essentiellement le bouton Menu décrit dans MainActivity.java.

```java
public class infos extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_infos);

        ImageButton menu = findViewById(R.id.menuIcon);

        menu.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) { allerMenu(); }
        });


    }

    public void allerMenu() {
        Intent intent = new Intent( packageContext: this, Menu.class);
        startActivity(intent);
    }
}
```

FIGURE 12 – Activité infos.java

4 Conclusion

L'application est fonctionnelle et est presque comme elle a été initialement imaginée. Cet exercice sur Android Studio était très intéressant et utile pour le futur où l'utilisation d'applications mobiles devient aujourd'hui indispensable. Des perspectives d'améliorations auxquelles j'ai pensé seraient :

— Remplacer le nom des pays par leur drapeau dans les spinners (les drapeaux sont dans drawable mais je n'ai pas eu le temps de créer les spinners) ;
— Peaufiner le design avec par exemple des animations en cliquant sur les imageButton ;
— Faire en sorte que la conversion se fasse automatiquement lorsque l'on a déjà cliqué une fois et que l'on change une devise d'un résultat sans avoir changé la saisie ;
— Mettre davantage de pays disponibles...