

# Les fragments dynamiques

---

AFRASS ILIAS (FA)  
22 Novembre 2018

## Résumé

---

La réflexion découle d'un problème simple qui est l'adaptation d'une application Android à toutes les tailles d'appareils existants.

Tout d'abord, il faut bien comprendre que sans les fragments il était déjà possible de s'adapter à toutes les tailles d'écrans. Il suffisait de mettre un layout particulier dans `layout-small`, `layout-normal`, `layout-large`, `layout-xlarge`, pour obtenir l'I.H.M. souhaitée en fonction des caractéristiques de l'écran. Par contre, l'activité qui contrôlait tous ces cas devenait très complexe.

Dans ce TP, nous allons voir comment rendre flexible notre interface graphique, en découvrant ensemble le concept de **fragment** !

## Pré-requis

---

- *Savoir programmer une application Android avec plusieurs activités qui s'appellent et qui passe des paramètres entre eux.*
- *Maîtriser les layouts.*

## Code source

---

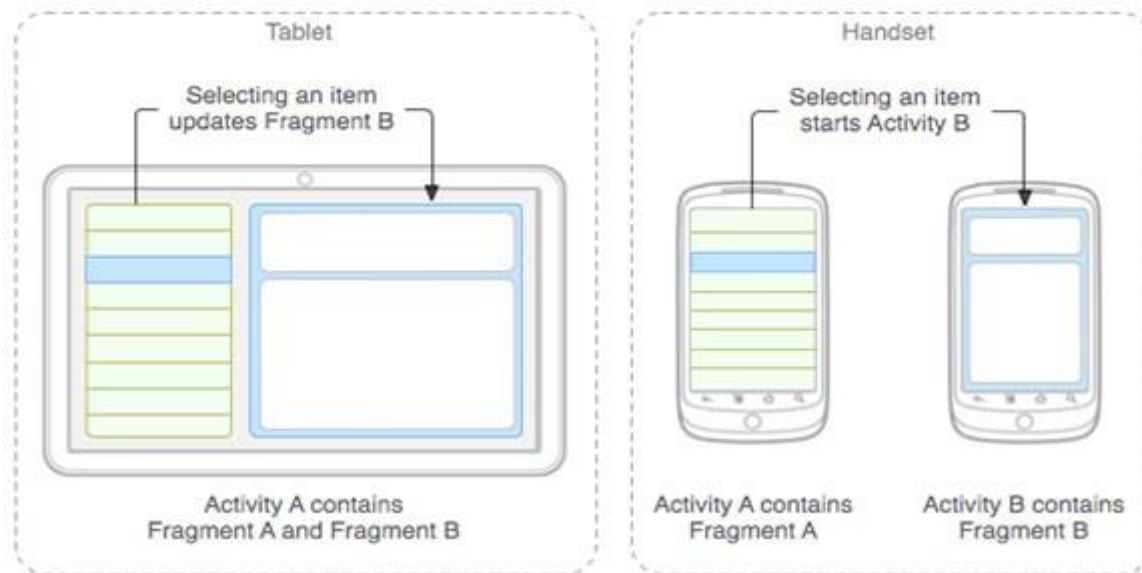
Code source **initial** : <https://github.com/iliasafrass/FragmentTutorielInit>

Code source **final** : <https://github.com/iliasafrass/FragmentTutorielFinal>

## Qu'est-ce qu'un fragment ?

---

Les fragments permettent de scinder vos activités en composants encapsulés et réutilisables qui possèdent leur propre cycle de vie et leur propre interface graphique. Cela permet de mettre en place des I.H.M évoluées qui s'adaptent aux différents écrans et à leur orientation tout en maintenant le code de l'activité. Ainsi, en pratique, **une activité va être découpée en un ou plusieurs fragments** afin de créer une interface utilisateur **modulable** et surtout **flexible** visuellement parlant. Ainsi, dans l'exemple ci-dessous (celui de Google), nous voyons comment l'activité A s'adapte en fonction de la taille de l'écran.



## Quelques notions élémentaires concernant les fragments :

---

- Ils sont liés à une activité (ils n'existent pas sans elle) ;
- Ils définissent la plupart du temps une interface graphique, mais peuvent aussi être utilisés pour retenir un état lors de la destruction/reconstruction de leur activité.
- Ils peuvent être statiques (définis une fois pour toutes dans le fichier de layout) ou dynamiques (créés, supprimés, ajoutés dynamiquement) ;
- Ils sont apparus à partir de HoneyComb (level 11) ainsi, pour les mettre en place avant HoneyComb, il faut utiliser la support-library ;
- Pour les utiliser, il faut un BuildSDK et un TargetSDK supérieur à 11.
- Les classes fondamentales pour la gestion des fragments sont : *Fragment*, *FragmentManager* et *FragmentTransaction*.

## Pourquoi les utiliser

---

Une des raisons principales d'utiliser les fragments est la volonté de **soutenir un panel d'écrans de tailles différentes** dans son application. Ils permettront ainsi de **faciliter l'affichage sur tablette** (ainsi que le mode **paysage** dans certains cas de Smartphones à large écran) tout en créant des interfaces **réutilisables** et **indépendantes**.

## Comment implémenter les fragments ?

---

Pour créer un fragment sur Android Studio cliquer sur : **file => New => Fragment => Fragment(Blank)** puis nommer votre fragment.

Vous remarquez que la création d'un fragment est comme une activité. (Il contient une **classe Java** et un **layout XML**).

- **Classe java :**

```
public class MainFragment extends Fragment {
    public MainFragment() { }
}
```

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
    return inflater.inflate(R.layout.fragment_main, container, false);
}
}

```

- **XML :**

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.openclassrooms.MainFragment">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="@string/hello_blank_fragment" />

</LinearLayout>

```

Notre classe MainFragment **hérite de Fragment** (à la place d'Activity habituellement), afin de **recupérer tout le comportement et les propriétés** d'un vrai fragment. Enfin, en plus de la méthode familière onCreate(), nous avons la méthode **onCreateView()** qui nous permet de **déclarer notre layout** (fragment\_main).

## Comment notre fragment va-t-il être affiché par notre activité ?

Grâce au **FragmentManager** disponible au sein de chaque activité, on va pouvoir gérer nos fragments. Pour résumer, c'est lui qui va **ajouter ou supprimer** des fragments.

Un exemple de code déclaré à l'intérieur d'une activité souhaitant afficher un fragment:

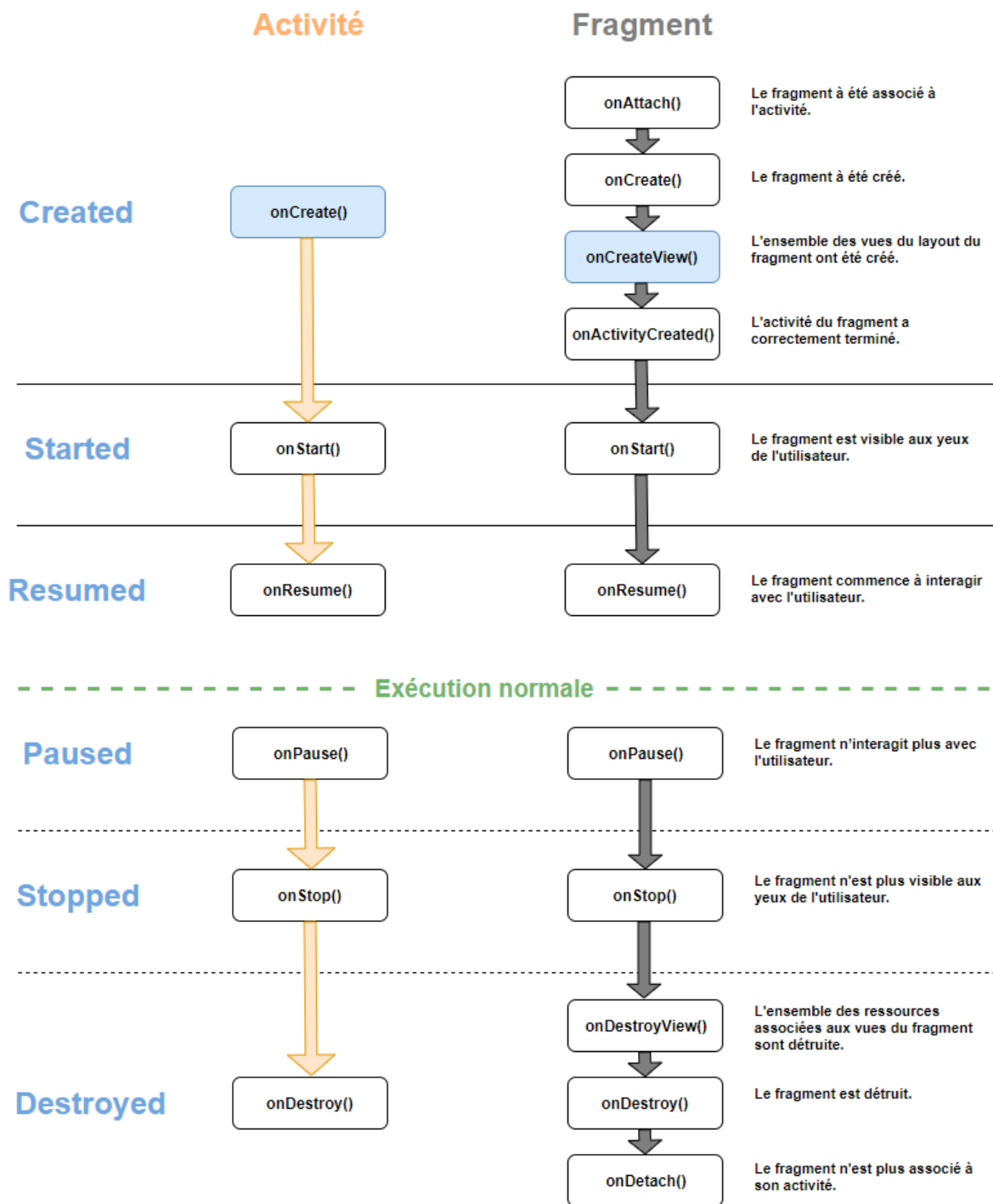
```

//1 - Get our FragmentManager & FragmentTransaction (Inside an activity)
FragmentManager fragmentManager = getFragmentManager();
FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
//2 - Create a new fragment and add it into activity
NotreFragment fragment = new NotreFragment();
fragmentTransaction.add(R.id.fragment_container, fragment);
fragmentTransaction.commit();

```

## Le cycle de vie d'un fragment :

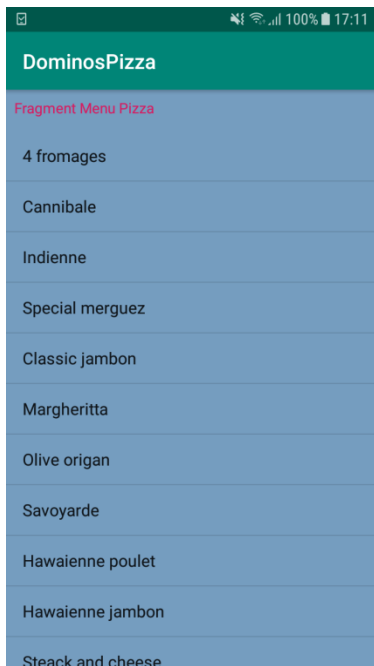
---



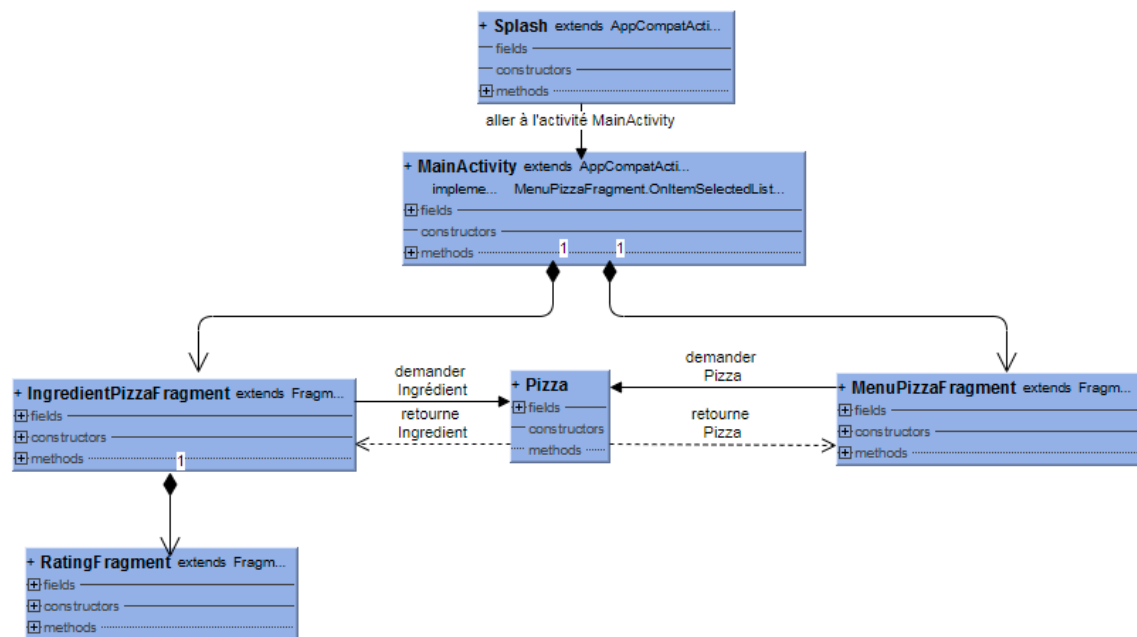
**onViewCreated** est appelé immédiatement après **onCreateView** (la méthode d'initialisation et de création de tous vos objets, y compris votre **TextView**). Il ne s'agit donc pas d'une question de performance. Vous devriez gonfler votre **layout** dans **onCreateView** mais ne pas initialiser d'autres vues à l'aide de **findViewById** dans **onCreateView**. Parce que parfois la vue n'est pas correctement initialisée.

Par conséquent, utilisez toujours **findViewById** dans **onViewCreated** (lorsque la vue est entièrement créée) et passe également la vue en tant que paramètre. **onViewCreated** permet de s'assurer que la vue est entièrement créée.

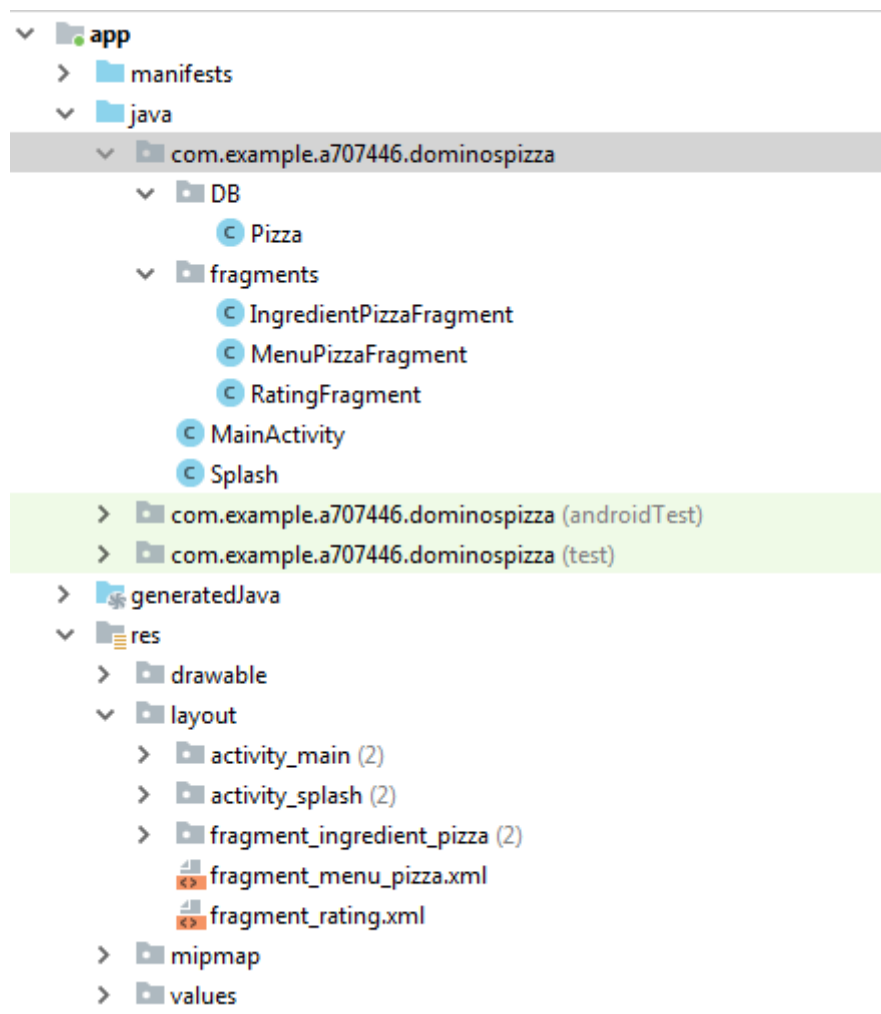




## Diagramme de classe :



## L'arborescence du notre TP final :



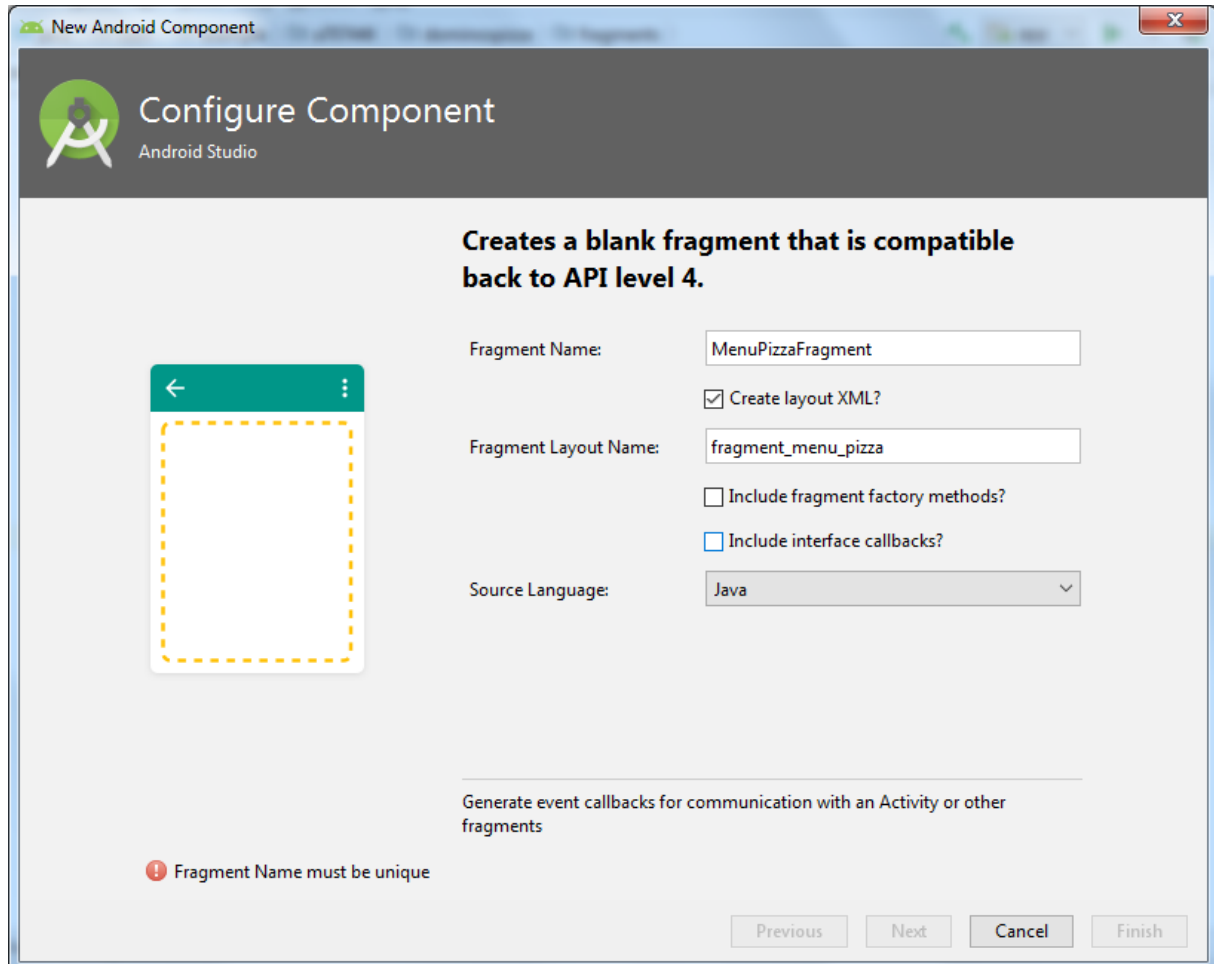
## Explications du TP

## Création des fragments :

Nous allons créer maintenant notre premier fragment, **MenuPizzaFragment**. Pour cela rien de plus simple, il suffit de faire un clic droit sur notre package "fragments" et cliquer

sur **New → Fragment → Fragment(Blank)** puis **nommer** notre premier fragment **MenuPizzaFragment**:

Décochez « Include fragment factory methods? » et « Include interface callbacks? » afin de tout implémenter nous-même.



Nous allons simplement **modifier l'apparence** de notre **MenuPizzaFragment** en modifiant son fichier layout (**fragment\_menu\_pizza.xml**) afin de lui **ajouter une ListView** qui contient les noms des pizzas.

Fichier **fragment\_menu\_pizza.xml** :

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#759dbf"
    tools:context=".fragments.MenuPizzaFragment">

    <android.support.constraint.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/fragment1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="top"
```



```

        android:layout_marginStart="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="8dp"
        android:text="@string/tv_frag_menu_pizza"
        android:textAppearance="?android:attr/textAppearanceSmall"
        android:textColor="@color/colorAccent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

<ListView
    android:id="@+id/lvItems"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_marginTop="40dp"
    android:choiceMode="singleChoice"
    app:layout_constraintTop_toBottomOf="@+id/fragment1" />

</android.support.constraint.ConstraintLayout>

</FrameLayout>

```

Ensuite créez deux fragments, **IngredientPizzaFragment** et **RatingFragment**.

Fichier **fragment\_ingredient\_pizza.xml** :

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#c4d5e4"
    tools:context=".fragments.IngredientPizzaFragment"
    android:baselineAligned="false">

    <android.support.constraint.ConstraintLayout
        android:layout_width="0dp"
        android:layout_weight="60"
        android:layout_height="match_parent"
        >

        <TextView
            android:id="@+id/tvfragment2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"
            android:layout_marginLeft="8dp"
            android:layout_marginTop="16dp"
            android:layout_marginEnd="8dp"
            android:layout_marginRight="8dp"
            android:text="@string/tv_frag_Ingredient_pizza"
            android:textAppearance="?android:attr/textAppearanceSmall"
            android:textColor="@color/colorAccent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

        <TextView
            android:id="@+id/tvTitre"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"
            android:layout_marginLeft="8dp"
            android:layout_marginTop="16dp"
            android:layout_marginEnd="8dp"

```

```

        android:layout_marginRight="8dp"
        android:text=""
        android:textStyle="bold"
        android:textSize="30dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textColor="#759dbf"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvfragment2"
        tools:text="titre" />

```

```

<TextView
    android:id="@+id/tvIngredient"
    android:textStyle="italic"
    android:textSize="25dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginLeft="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginRight="16dp"
    android:text=""
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:textColor="#000"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvTitre"
    tools:text="Ingredient" />

```

```

<RatingBar
    android:id="@+id/ratingBar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginBottom="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvIngredient" />

```

```

<FrameLayout
    android:id="@+id/containerIngredient"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/ratingBar" />

```

```

</android.support.constraint.ConstraintLayout>

```

```

</LinearLayout>

```

Fichier **fragment\_rating.xml** :

```

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

```

```

xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".fragments.RatingFragment">

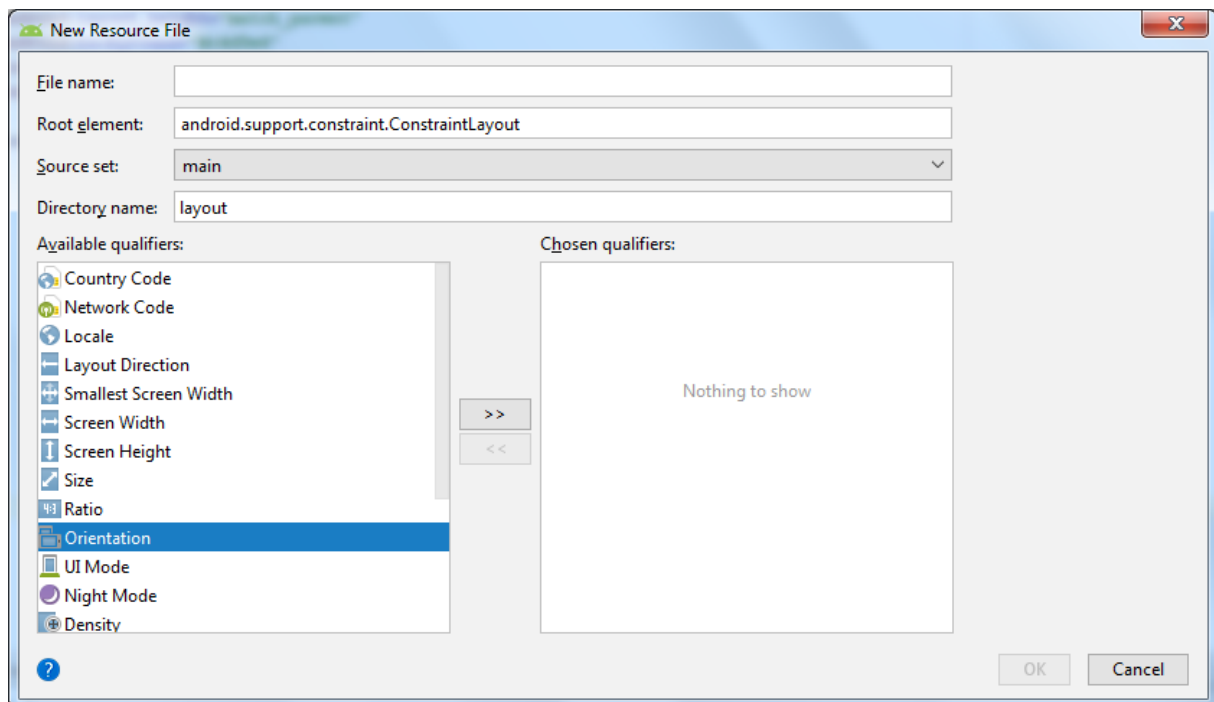
<!-- TODO: Update blank fragment layout -->
<TextView
    android:layout_width="match_parent"
    android:textColor="@color/colorAccent"
    android:textSize="25dp"
    android:textStyle="bold|italic"
    android:id="@+id/textview"
    android:layout_height="match_parent"
    android:text="hello"/>

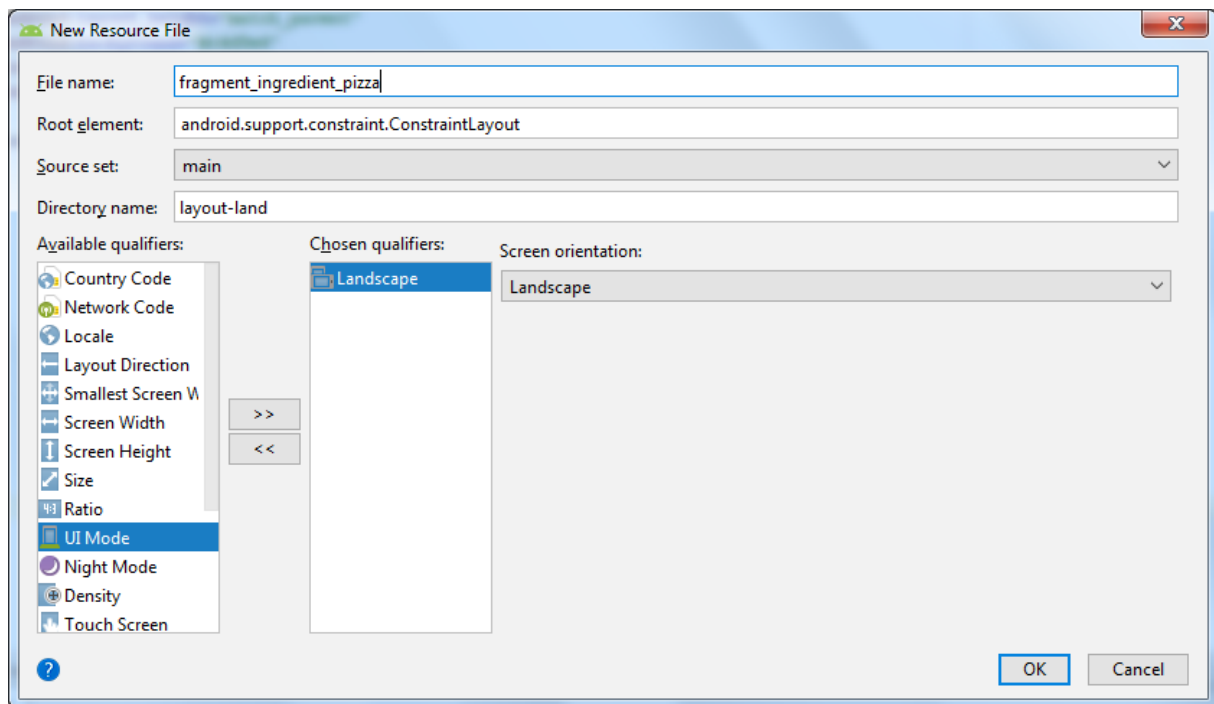
</FrameLayout>

```

Nous allons ajouter le mode paysage **de** notre **IngredientPizzaFragment** en créant un layout landscape **fragment\_ingredient\_pizza(land)**.

Pour cela, il suffit de faire un clic droit sur le dossier **layout** et cliquer sur **New → Layout resource file** puis choisissez Orientation et cliquer sur >>.





Fichier **fragment\_ingredient\_pizza.xml** (land) :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#c4d5e4"
    tools:context=".fragments.IngredientPizzaFragment"
    android:baselineAligned="false">

    <android.support.constraint.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/tvfragment2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"
            android:layout_marginLeft="8dp"
            android:layout_marginEnd="8dp"
            android:layout_marginRight="8dp"
            android:text="@string/tv_frag_Ingredient_pizza"
            android:textAppearance="?android:attr/textAppearanceSmall"
            android:textColor="@color/colorAccent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

        <TextView
            android:id="@+id/tvTitre"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"
            android:layout_marginLeft="8dp"
            android:layout_marginTop="8dp"
            android:layout_marginEnd="8dp"
            android:layout_marginRight="8dp"
            android:text=""
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:textColor="#759dbf">
```

```

        android:textSize="30sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvfragment2"
        tools:text="titre" />

```

```

<TextView
    android:id="@+id/tvIngredient"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginLeft="16dp"
    android:layout_marginTop="24dp"
    android:layout_marginEnd="16dp"
    android:layout_marginRight="16dp"
    android:text=""
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:textColor="#000"
    android:textSize="25sp"
    android:textStyle="italic"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvTitre"
    tools:text="Ingredient" />

```

```

<RatingBar
    android:id="@+id/ratingBar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginBottom="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvIngredient" />

```

```

<FrameLayout
    android:id="@+id/containerIngredient"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginBottom="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/ratingBar" />

```

```

</android.support.constraint.ConstraintLayout>

```

```

</LinearLayout>

```

## Rendre notre interface dynamique :

Le `FrameLayout` est un layout qui va **nous servir de conteneur** pour afficher nos fragments. Ces derniers seront *appelés et créés depuis l'activité* (via le *FragmentManager*), puis *affichés dans un FrameLayout* bien défini.

Création des conteneurs **FrameLayout** :

Pour cela, nous allons **modifier le fichier layout** de notre activité, *activity\_main.xml* pour y ajouter notre conteneur `FrameLayout` :

Fichier *activity\_main.xml* :

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <FrameLayout
        android:id="@+id/flContainer1"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</android.support.constraint.ConstraintLayout>
```

Fichier *activity\_main.xml* (land):

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">

    <FrameLayout
        android:id="@+id/flContainer1"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="30" />

    <FrameLayout
        android:id="@+id/flContainer2"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="70"
        android:scrollbarAlwaysDrawVerticalTrack="true" />

</LinearLayout>
```

## Communication entre le fragment « *MenuPizzaFragment* » et l'activité:

Maintenant que nous voulions, au clic de l'utilisateur sur un Item de la liste des pizzas, charge un fragment qui nous affiche les ingrédients.

On va laisser la responsabilité à l'activité *MainActivity* de gérer les fragments.

### Création du listener de notre List :

Ajoutez le code ci-dessous dans le fichier *MenuPizzaFragment.java* :

```
public class MenuPizzaFragment extends Fragment {
```

```

public MenuPizzaFragment() {
    // Required empty public constructor
}

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    //init itemsAdapter
    itemsAdapter = new ArrayAdapter<String>(getContext(),
android.R.layout.simple_list_item_1, Pizza.Menupizza);
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    return inflater.inflate(R.layout.fragment_menu_pizza, container, false);
}

@Override
public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState)
{
    ListView lvItems = (ListView) view.findViewById(R.id.lvItems);
    lvItems.setAdapter(itemsAdapter);

    //Set onItemClickListener to List lvItems
    lvItems.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
            //Here is handled the list click
            /*
            Communicate with Activity using Listener
            go to activity to load pizza details fragment
            */
        }
    });
}
}

```

## Création d'une interface pour communiquer avec l'activité :

Ensuite, nous allons **créer une interface** qui nous permettra de communiquer avec notre activité parente. Pour cela nous allons modifier notre fragment **MenuPizzaFragment** comme ce qui suit :

Fichier **MenuPizzaFragment.java**:

```

public class MenuPizzaFragment extends Fragment {

    ArrayAdapter<String> itemsAdapter;
    private OnItemSelectedListener mListener;

    public MenuPizzaFragment() {
        // Required empty public constructor
    }

```

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    //init itemsAdapter
    itemsAdapter = new ArrayAdapter<String>(getContext(),
android.R.layout.simple_list_item_1, Pizza.Menupizza);
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    return inflater.inflate(R.layout.fragment_menu_pizza, container, false);
}

@Override
public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState)
{
    ListView lvItems = (ListView) view.findViewById(R.id.lvItems);
    lvItems.setAdapter(itemsAdapter);

    //Set onItemClickListener to List lvItems
    lvItems.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
            //Here is handled the list click

            /*
             Communicate with Activity using Listener
             go to activity to load pizza details fragment
             */
            mListener.onPizzaItemSelected(position);
        }
    });
}

@Override
public void onAttach(Context context) {
    super.onAttach(context);
    if (context instanceof OnItemSelectedListener) {
        //Call the method that creating OnItemSelectedListener after being attached
        to parent activity
        mListener = (OnItemSelectedListener) context;
    } else {
        throw new RuntimeException(context.toString()
            + " must implement OnItemSelectedListener");
    }
}

/**
 * This interface must be implemented by activities that contain this
 * fragment (MainActivity) to allow an interaction in this fragment to be
communicated
 * to the activity.
 */
public interface OnItemSelectedListener {
    void onPizzaItemSelected(int position);
}
}

```

#### ❖ Explications:

- **Création de l'interface :** Nous créons ici une interface dans le but d'obliger toute activité souhaitant communiquer avec ce fragment à implémenter la méthode onPizzaItemSelected(position).
- **Déclaration de notre mListener :** En déclarant ce mListener comme variable au sein du fragment, nous allons pouvoir créer un lien direct avec notre activité parente.



- **Création du mListener pointant vers notre activité** : Nous allons lier notre mListener avec notre activité parente en y souscrivant depuis le fragment enfant. Cependant, il faudra que notre activité parente (donc qui contient ce fragment) *implémente l'interface* onPizzaItemSelected(position).
- **Propagation du clic** : Nous allons ici propager le clic de notre utilisateur directement à notre activité parente via la méthode onPizzaItemSelected(position).

Il nous reste que l'activité MainActivity implémente cette interface.

Fichier MainActivity.java :

```
public class MainActivity extends AppCompatActivity implements
MenuPizzaFragment.OnItemSelectedListener {

    // Declare static variable of "String" type to identify position in Bundle
    private static final String KEY_POSITION = "KEY_POSITION";

    // Declare two variable to be of a MenuPizzaFragment and IngredientPizzaFragment
type
    private MenuPizzaFragment firstFragment;
    private IngredientPizzaFragment secondFragment;

    // Declare variable of "int" type to save a position
    private int position_save;

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

    @Override
    public void onPizzaItemSelected(int position) {
        Toast.makeText(this, "Appelé par Fragment Menu Pizza: position : " + position,
        Toast.LENGTH_SHORT).show();

        secondFragment = new IngredientPizzaFragment();

        // Load Pizza Ingredient Fragment
        position_save = position;
        Bundle args = new Bundle();
        args.putInt("position", position);
        secondFragment.setArguments(args); // Communicate with
        Fragment using Bundle

        //Orientation landscape
        if (getResources().getConfiguration().orientation ==
        Configuration.ORIENTATION_LANDSCAPE) {
            getSupportFragmentManager()
                .beginTransaction()
                .replace(R.id.flContainer2, secondFragment) // replace flContainer
                .addToBackStack(null)
                .commit();
        } else {
            //Orientation Portrait
            getSupportFragmentManager()
                .beginTransaction()
                .replace(R.id.flContainer1, secondFragment) // replace flContainer
                .addToBackStack(null) // adding the fragment
in the Fragment Stack but not adding any TAG
                .commit();
        }
    }
}
```

```

@Override
public void onSaveInstanceState(@NonNull Bundle outState) {
    //Save position save in Bundle when fragment is destroyed
    outState.putInt(KEY_POSITION, position_save);
    super.onSaveInstanceState(outState);
}
}

```

## Création de l'écran secondaire (IngredientPizzaFragment):

### IngredientPizzaFragment:

Maintenant, il faudrait **afficher les ingrédients** lorsque nous cliquons sur un item.

Ajoutez le code ci-dessous dans le fichier **IngredientPizzaFragment.java** :

```

public class IngredientPizzaFragment extends Fragment {
    private int position = 0;
    private TextView tvTitre;
    private TextView tvIngredient;
    private RatingBar ratingBar;
    private float rating_save;

    public IngredientPizzaFragment() {
        // Required empty public constructor
    }

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        if (savedInstanceState == null) {
            // Get back arguments
            if (getArguments() != null) {
                position = getArguments().getInt("position", 0);
            }
        }

        @Nullable
        @Override
        public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup
container, @Nullable Bundle savedInstanceState) {
            // Inflate the xml file for the fragment
            return inflater.inflate(R.layout.fragment_ingredient_pizza, container, false);
        }

        @Override
        public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState)
        {
            // Set values for view here
            tvTitre = (TextView) view.findViewById(R.id.tvTitre);
            tvIngredient = (TextView) view.findViewById(R.id.tvIngredient);
            ratingBar = (RatingBar) view.findViewById(R.id.ratingBar);
            // update view
            updateView(position);
        }

        // Activity is calling this to update view on Fragment
        public void updateView(int position) {
            tvTitre.setText(Pizza.Menupizza[position]);
            tvIngredient.setText(Pizza.pizzaIngredients[position]);
        }
    }
}

```

```

    public void addListenerOnRatingBar() {
        //if rating value is changed,
        //display the current rating value in the result (textview) automatically
        ratingBar.setOnRatingBarChangeListener(new
RatingBar.OnRatingBarChangeListener() {
            public void onRatingChanged(RatingBar ratingBar, float rating,
                                      boolean fromUser) {

                Toast.makeText(getContext(), "évaluation = " + rating,
Toast.LENGTH_SHORT).show();
                RatingFragment ratingFragment = new RatingFragment();
                rating_save = rating;
                Bundle args = new Bundle();
                args.putFloat("rating", rating);
                ratingFragment.setArguments(args);           // Communicate with
Fragment using Bundle

                getSupportFragmentManager()
                    .beginTransaction()
                    .replace(R.id.containerIngredient, ratingFragment) // replace
flContainer

                    .addToBackStack(null)
                    .commit();

            }
        });
    }

    @Override
    public void onSaveInstanceState(@NonNull Bundle outState) {
        // Save rating_save in Bundle when fragment is destroyed
        outState.putFloat("key_rating", rating_save);
        super.onSaveInstanceState(outState);
    }
}

```

### RatingFragment:

Comme IngredientPizzaFragment contient un autre fragment qui est RatingFragment, maintenant il faut ajoutez le code ci-dessous dans le fichier **RatingFragment.java** :

```

public class RatingFragment extends Fragment {

    private TextView textView;
    private float rating = 0;

    public RatingFragment() {
        // Required empty public constructor
    }

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        if (savedInstanceState == null) {
            // Get back arguments
            if (getArguments() != null) {
                rating = getArguments().getFloat("rating", 0);
            }
        }
    }

    @Override

```

```

public View onCreateView(LayoutInflater inflater, ViewGroup container,
                        Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    return inflater.inflate(R.layout.fragment_rating, container, false);
}

@Override
public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState)
{
    textView = (TextView) view.findViewById(R.id.textview);

    // update view
    updateView(rating);
}

// Activity is calling this to update view on Fragment
public void updateView(float rating) {
    if (rating != 0)
        if (rating < 2.5)
            textView.setText("votre évaluation est de : " + rating + " / 5.0" + "
:(");
        else
            textView.setText("votre évaluation est de : " + rating + " / 5.0" + "
:)");
        else
            textView.setText("J'attends votre évaluation :)");
    }
}

```

## Affichage des fragments :

Ajoutez le code ci-dessous dans **IngredientPizzaFragment.java** :

```

public class IngredientPizzaFragment extends Fragment {

.....

@Override
public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState)
{
    // Set values for view here
    tvTitre = (TextView) view.findViewById(R.id.tvTitre);
    tvIngredient = (TextView) view.findViewById(R.id.tvIngredient);
    ratingBar = (RatingBar) view.findViewById(R.id.ratingBar);
    showView(savedInstanceState);
    // update view
    updateView(position);
}

private void showView(Bundle savedInstanceState) {
    if (savedInstanceState == null) {
        FragmentManager fragmentManager =
getActivity().getSupportFragmentManager();
        android.support.v4.app.FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();

        RatingFragment ratingFragment = new RatingFragment();
        //Add Fragment to FrameLayout (containerIngredient), using FragmentManager
        fragmentTransaction.add(R.id.containerIngredient, ratingFragment);
// add Fragment
        fragmentTransaction.commit();
    }
}

```

```

        addListenerOnRatingBar();
    }

    if (getResources().getConfiguration().orientation ==
Configuration.ORIENTATION_LANDSCAPE) {
        FragmentManager fragmentManager =
getActivity().getSupportFragmentManager();
        android.support.v4.app.FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();

        if (savedInstanceState != null) {
            rating_save = savedInstanceState.getFloat("key_rating", 0);
        }
        RatingFragment ratingFragment = new RatingFragment();
        Bundle args = new Bundle();
        args.putFloat("rating", rating_save);
        ratingFragment.setArguments(args);           //Communicate with Fragment
using Bundle
        //Add Fragment to FrameLayout (containerIngredient), using FragmentManager
        fragmentTransaction.replace(R.id.containerIngredient, ratingFragment);
// add Fragment
        fragmentTransaction.commit();

        addListenerOnRatingBar();
    }
}

.....
}

```

Ajoutez le code ci-dessous dans **MainActivity.java** :

```

public class MainActivity extends AppCompatActivity implements
MenuPizzaFragment.OnItemSelectedListener {

...

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

private void showFragments(Bundle savedInstanceState){
    // the savedInstanceState will always be null the first time an Activity is
started
    if (savedInstanceState == null) {
        // Instance of MenuPizzaFragment
        firstFragment = new MenuPizzaFragment();
        // Add Fragment to FrameLayout (flContainer1), using FragmentManager
        FragmentTransaction ft = getSupportFragmentManager().beginTransaction();//
begin FragmentTransaction
        ft.add(R.id.flContainer1, firstFragment);           //
add Fragment to flContainer1
        ft.commit();           //
commit FragmentTransaction
    }

    if (getResources().getConfiguration().orientation ==
Configuration.ORIENTATION_LANDSCAPE) {

```

```

        // Instance of MenuPizzaFragment and IngredientPizzaFragment
        firstFragment = new MenuPizzaFragment();
        secondFragment = new IngredientPizzaFragment();

        // Add Fragment to FrameLayout (flContainer1), using FragmentManager
        FragmentTransaction ft = getSupportFragmentManager().beginTransaction();
begin    FragmentTransaction
        ft.replace(R.id.flContainer1, firstFragment);
add      Fragment
        ft.commit();

        // Restore last position if possible
        if (savedInstanceState != null) {
            position_save = savedInstanceState.getInt(KEY_POSITION, 0);
        }
        // Load Pizza Ingredient Fragment
        Bundle args = new Bundle();
        args.putInt("position", position_save);
        secondFragment.setArguments(args);
Communicate with Fragment using Bundle

        // Add Fragment to FrameLayout (flContainer2), using FragmentManager
        FragmentTransaction ft2 = getSupportFragmentManager().beginTransaction();
begin    FragmentTransaction
        ft2.add(R.id.flContainer2, secondFragment);
add      Fragment
        ft2.commit();
commit   FragmentTransaction

    }
}

...
}

```

**Félicitations! J'espère que ce Tutorial vous aura plu 😊**

## Informations complémentaires

---

- [https://zestedesavoir.com/tutoriels/609/aller-plus-loin-dans-le-developpement-android-1/323\\_fragmenter-vos-projets/1795\\_fragment/](https://zestedesavoir.com/tutoriels/609/aller-plus-loin-dans-le-developpement-android-1/323_fragmenter-vos-projets/1795_fragment/) : un tutoriel sur les fragments qui permettent de comprendre les fonctionnalités des fragments.
- <https://developer.android.com/guide/components/fragments> : permet à un débutant de comprendre les fragments.
- <https://vogella.developpez.com/tutoriels/android/utiliser-fragments> : un tutorial sur les fragments.