

LES FRAGMENTS

I) DEFINITION

Un fragment est une partie d'une activité qui permet de concevoir des activités plus modulaires. C'est une sous-activité.

II) PROPRIETES GENERALES

- Un fragment possède son propre **layout** , son propre comportement avec les méthodes de son propre cycle de vie.
- On peut ajouter et supprimer des fragments dans une activité pendant que l'activité s'exécute.
- On peut combiner plusieurs fragments dans une seule activité pour construire un UI.
- Un fragment peut être utilisé dans de multiples activités.

- Le cycle de vie d'un fragment est en relation étroite avec le cycle de vie de l'activité qui la contient ce qui signifie que quand l'activité est en pause, tous les fragments disponible dans l'activité seront arrêté.
- Les fragments ont été utilisé à l'API **android HoneyComb** qui est l'API version 11

III) CYCLE DE VIE D'UNE ACTIVITE

On crée des fragments en étendant la classe Fragment et on peut insérer des fragment dans les **layout** des activités en déclarant le fragment dans le fichier **layout** de l'activité dans l'élément **<fragment>**.

Avant l'introduction des fragments , nous avons une limitation car nous pouvions seulement voir une seule **activity** sur l'écran à un moment dans le temps. Nous n'étions pas capable de diviser l'écran et de contrôler les différents parties séparément. Mais avec l'introduction des fragments on a eu plus de flexibilité et on a supprimé la limitation d'avoir une seule **activity** sur l'écran à un moment donnée.

Maintenant on a une seule **activity** mais chaque **activity** peut avoir de multiple fragments qui ont leur propre **layout** , **événements** et cycle de vie.

Travaux pratique

Nous allons voir comment fonctionne le cycle de vie d'une activity

Creer MainActivity.java

```
package com.example.myapplication;  
import android.app.Fragment;  
import android.app.Activity;
```

```

import android.app.FragmentTransaction;
import android.os.Bundle;
import android.os.PersistableBundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.os.Build;
import android.widget.Toast;
import androidx.annotation.Nullable;
public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("cycle", "activity cree");
    }
    @Override
    protected void onStart() {
        super.onStart();
        Log.d("cycle", "activity a demarre");
    }
    @Override
    protected void onResume() {
        super.onResume();
        Log.d("cycle", "activity a resume");
    }
    @Override
    protected void onPause() {
        super.onPause();
        Log.d("cycle", "activity a pause");
    }
    @Override
    protected void onStop() {
        super.onStop();
        Log.d("cycle", "activity a stoppe");
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d("cycle", "activity est detruite");
    }
    public void action(View view) {

```

```
    finish();
}
}
```

Creer activity_main.xml

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"

    android:orientation="vertical"

    android:layout_width="fill_parent"

    android:layout_height="fill_parent">

<Button

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:text="envoyer"

    android:onClick="action"/>

</LinearLayout>
```

Gestion de l'orientation

```
int orientation=getResources().getConfiguration().orientation;
if (orientation==Configuration.ORIENTATION_PORTRAIT){
    Toast.makeText(this,"portrait",Toast.LENGTH_SHORT).show();
}
if (orientation==Configuration.ORIENTATION_LANDSCAPE){
    Toast.makeText(this,"LANDSCAPE",Toast.LENGTH_SHORT).show();
}
@Override
public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);
    if (newConfig.orientation == Configuration.ORIENTATION_LANDSCAPE){
        Toast.makeText(this,"LANDSCAPE", Toast.LENGTH_SHORT).show();
    } else if (newConfig.orientation==Configuration.ORIENTATION_PORTRAIT) {
        Toast.makeText(this,"PORTRAIT", Toast.LENGTH_SHORT).show();
    }
}
```

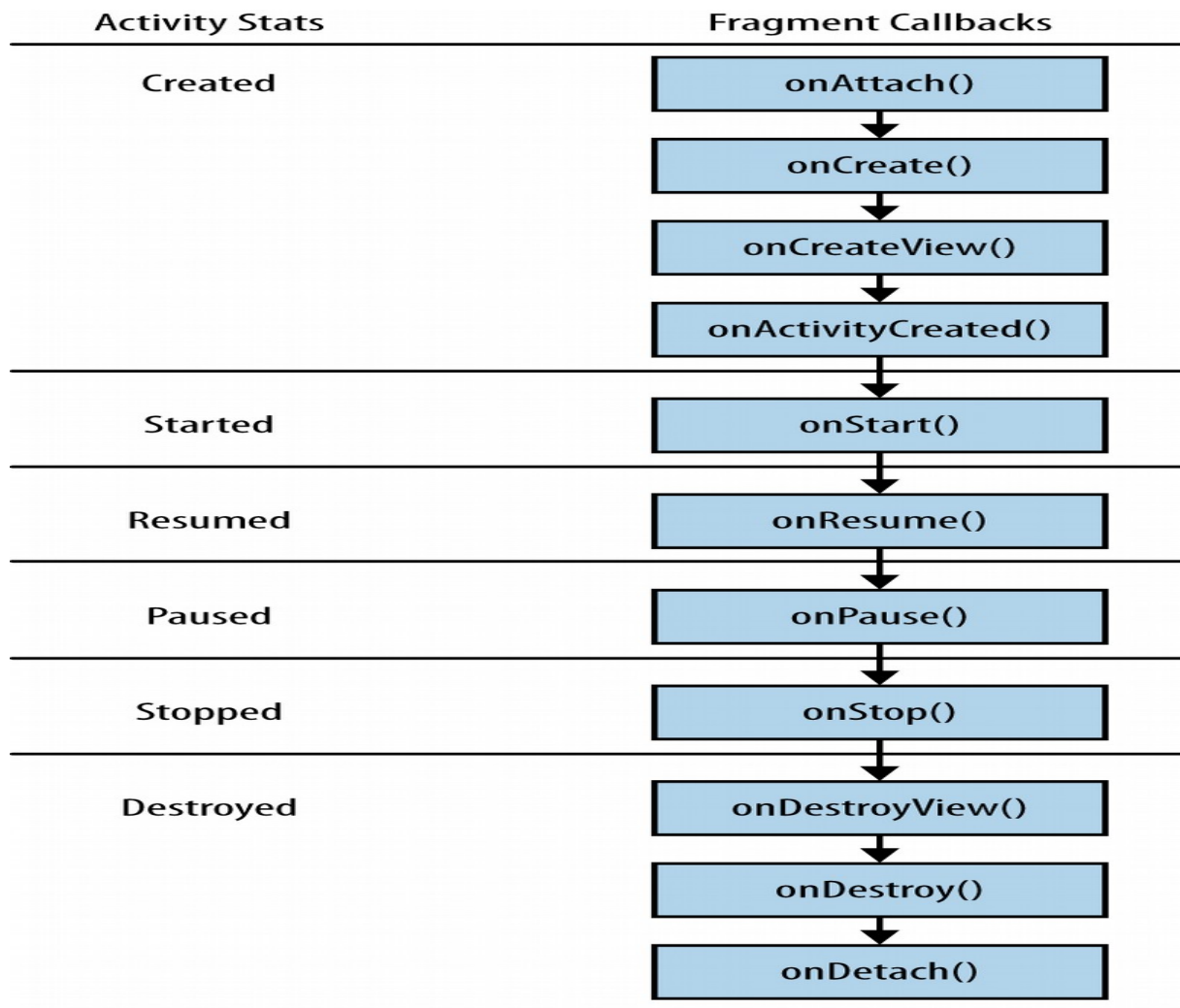
Sauvegarde apres changement d'orientation

- **onAttach()** : L'instance du fragment est associé avec une instance d'activité. Le fragment et l'activité ne sont pas complétement initialisé.
-
- **onCreate()** : Le système appelle cette méthode lors de la création du fragment. On peut initialiser les composants essentiels du fragment que l'on souhaite retenir quand le fragment est en pause ou **stopped** , puis en mode **resumed**.
-
- **onCreateView()** : Le système appelle cette méthode quand il est temps pour le fragment de dessiner son interface pour la première fois. Pour dessiner un UI pour le fragment, on peut renvoyer un composant **view** a partir de cette méthode qui est la racine (**root**) du **layout** du fragment. On retourne **null** si le fragment ne peut fournir d'UI.
- **onActivityCreated()** : **onActivityCreated()** est appelé après **onCreateView()** quand l'activité hôte est créée. **Activity** et **Fragment** ont été crée avec la hiérarchie des **view** de l'activité. A ce moment, les **view** peuvent être accédé avec la méthode **findViewById()**. Dans cette méthode on peut instancier les objets qui requiert un objet contexte.
- **onStart()** : Appelé quand le fragment devient visible
- **onResume()** : Le fragment devient actif
- **onPause()** Quand l'utilisateur quitte le fragment. C'est la que l'on **commit** tous les changements qui doivent persister au delà de la session utilisateur.
- **onStop()** Les fragments qui doivent être stoppé
- **onDestroyView()** Les fragments sont détruits après l'appel à cette méthode.
- **onDestroy()** : Appelé à l'étape finale de la destruction des fragments
-

IV) COMMENT UTILISER LES FRAGMENTS

- 1) Décider combien de fragments vous voulez utiliser dans l'activité.
- 2) Créer pour chaque fragment des classes qui étendront la classe Fragment.
- 3) Pour chaque fragment on crée un fichier layout XML.
- 4) Modifier le fichier activité pour définir la logique actuelle de replacer les fragments basés sur les exigences

V) LE CYCLE DE VIE DES FRAGMENTS



La classe **FragmentManager** fournit des méthodes qui permettent d'ajouter, de supprimer de remplacer des fragments dans une activité à l'exécution afin d'avoir une utilisation dynamique

La classe **FragmentManager** crée un **FragmentTransaction** qui fournit une API pour ajouter, supprimer, remplacer, et réaliser diverses opérations sur les fragments

Le **layout** de l'activité doit contenir un **container View** dans lequel on insère le fragment..

DANS L'ACTIVITY :

- **getFragmentManager()** permet d'obtenir un **FragmentManager**
- Appel **beginTransaction()** pour créer un **FragmentTransaction**
- Appeler la méthode **add()** pour ajouter un fragment
- **commit()** valide les changements

VI) LES STRUCTURES DE CODE PRINCIPALES

I) CREATION D'UN FRAGMENT DANS UNE ACTIVITY

On crée un Layout pour le fragment qui sera inflaté dans la méthode suivante

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

creation d'une activite qui va contenir le fragment

On crée le layout pour l'activité qui va contenir le fragment comme suit :

```
<fragment android:name="com.example.androidfragment.Fragment1"  
    android:id="@+id/fragment1"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent" />
```

III) COMMUNICATION ENTRE UN FRAGMENT ET SON ACTIVITE HOTE

I) On définit un interface

```
interface CommunicationChannel  
{  
    public void setCommunication(String msg);  
}
```

II) On crée un objet du type de cet interface dans le fragment

```
public class Fragment1 extends Fragment {  
    CommunicationChannel mCommChListner = null;
```

III) La classe Activité implémentera cet interface et définira une méthode qui récupérera le fragment et testera que le fragment est non null et est bien dans le layout puis affichera un message qui lui sera transmis en paramètre

```
public class MainActivity extends Activity implements  
Fragment1.CommunicationChannel {  
    texte=(TextView) findViewById(R.id.text2);  
  
    @Override  
    public void setCommunication(String msg) {  
        Fragment1 recFragment;  
        recFragment=(Fragment1) getFragmentManager().findFragmentById(R.id.fragment1);  
        if(null != recFragment && recFragment.isInLayout())  
        {  
            texte.setText(msg);  
        }  
    }  
}
```

```
}  
  
}
```

IV) Dans le fragment on contrôle que l'activité hôte est du type de l'interface qu'elle étend si oui , on récupère un pointeur vers cette activité et on définit une méthode qui va permettre d'envoyer un message à l'activité par le biais de la méthode de l'interface

```
@Override  
public void onAttach(Activity activity)  
{  
    super.onAttach(activity);  
    if(activity instanceof CommunicationChannel)  
    {  
        mCommChListner = (CommunicationChannel)activity;  
    }  
    else  
    {  
        throw new ClassCastException();  
    }  
  
}  
public void sendMessage(String msg)  
{  
    mCommChListner.setCommunication(msg);  
}  
  
@Override  
public void onClick(View v) {  
    sendMessage("du fragment1");  
}  
  
}
```

III) CREATION DE FRAGMENT DYNAMIQUE

- 1) Création d'un layout pour l'activité et on donne un ID pour le layout
- 2) Creation d'un fragment
- 3) Pour ajouter le fragment de façon dynamique on utilise la structure suivante :

```
getFragmentManager().beginTransaction()  
    .add(R.id.container1, new Fragment1())  
    .commit();
```


TRAVAUX PRATIQUES

1) Création d'un fragment de façon statique

A) Créez le Projet : FragmentApplication

A) Créez le layout suivant : fragment_main.xml

```
<RelativeLayout 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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.fragmentapplication.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />

</RelativeLayout>
```

B) Créez le fragment suivant : Fragment1.java

```
package com.example.fragmentapplication;
import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class Fragment1 extends Fragment {

    public Fragment1() {
    }

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

}
```

C) Créez le layout suivant : activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
android:orientation="horizontal"
android:layout_width="fill_parent"
android:layout_height="fill_parent">

<fragment android:name="com.example.fragmentapplication.Fragment1"
    android:id="@+id/fragment1"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />

</LinearLayout>
```

Cycle de vie activity fragment

Modifier le code du fragment comme suit :

```
package com.example.myapplication;
import android.app.Fragment;
import android.content.Context;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import androidx.annotation.Nullable;
public class Fragment1 extends Fragment {
    public Fragment1() {
    }
    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        Log.d("cycle", "fragment onAttach");
    }
    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
    Log.d("cycle", "fragment onCreate");
}

@Override
public void onCreate(@Nullable Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    Log.d("cycle", "fragment onCreate");
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup
container,
    Bundle savedInstanceState) {
    Log.d("cycle", "fragment onCreateView");
    View rootView = inflater.inflate(R.layout.fragment_1, container,
false);
    return rootView;
}

@Override
public void onStart() {
    super.onStart();
    Log.d("cycle", "fragment a demarre");
}

@Override
public void onResume() {
    super.onResume();
    Log.d("cycle", "fragment a resume");
}

@Override
public void onPause() {
```

```

    super.onPause();
    Log.d("cycle","fragment a pause");
}
@Override
public void onStop() {
    super.onStop();
    Log.d("cycle","fragment a stoppe");
}
@Override
public void onDestroy() {
    super.onDestroy();
    Log.d("cycle","fragment est detruite");
}
@Override
public void onDestroyView() {
    super.onDestroyView();
    Log.d("cycle","fragment est detruite view");
}
@Override
public void onDetach() {
    super.onDetach();
    Log.d("cycle","fragment est detache");
}
}
}

```

Travaux pratique 2 : Affichage d'un Toast dans l'activité contenant le fragment

- a) Dans le projet précédent remplacer Fragment1 par :

```

package com.example.fragmentapplication;
import android.app.Activity;
import android.app.Fragment;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.Toast;

```

```

public class Fragment1 extends Fragment implements OnClickListener {
    Button bouton;
    Activity activity;
    public Fragment1() {
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_main, container,
            false);
        bouton=(Button) rootView.findViewById(R.id.bouton);
        bouton.setOnClickListener((OnClickListener) this);

        return rootView;
    }
    @Override
    public void onAttach(Activity activity)
    {
        super.onAttach(activity);
        this.activity = activity;
    }

    @Override
    public void onClick(View v) {
        Toast.makeText(activity, "Clicked ", Toast.LENGTH_LONG).show();
    }
}

```

Ajouter le bouton suivant dans fragment_main.xml

```

<Button android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:id="@+id/bouton"

    />

```

A) Modifier la layout :activity_main

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <fragment android:name="com.example.fragmentapplication.Fragment1"
        android:id="@+id/fragment1"
        android:layout_width="match_parent"
        android:layout_weight="1"
        android:layout_height="wrap_content" />
    <TextView
        android:id="@+id/text2"
        android:text="valeur"
        android:layout_width="match_parent"
        android:layout_weight="1"
        android:layout_height="wrap_content" />

</LinearLayout>
```

B) Modifier la classe fragment 1 Comme suit

```
package com.example.androidfragment;
import android.app.Activity;
import android.os.Bundle;
import android.app.Fragment;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.Toast;

public class Fragment1 extends Fragment implements OnClickListener {
    CommunicationChannel mCommChListner = null;
    Button bouton;
    Activity activity;
    public Fragment1() {
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_main, container, false);
        bouton=(Button) rootView.findViewById(R.id.bouton);
        bouton.setOnClickListener((OnClickListener) this);

        return rootView;
    }
}

interface CommunicationChannel
{
    public void setCommunication(String msg);
}
```

```

@Override
public void onAttach(Activity activity)
{
    super.onAttach(activity);
    if(activity instanceof CommunicationChannel)
    {
        mCommChListner = (CommunicationChannel)activity;
    }
    else
    {
        throw new ClassCastException();
    }
}

public void sendMessage(String msg)
{
    mCommChListner.setCommunication(msg);
}

@Override
public void onClick(View v) {
sendMessage("du fragment1");
}
}
}

```

c) Modifier la classe Main_Activity comme suit

```

package com.example.androidfragment;
import android.app.Fragment;
import android.app.Activity;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import android.os.Build;

public class MainActivity extends Activity implements
Fragment1.CommunicationChannel {
    TextView texte;
    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        texte=(TextView) findViewById(R.id.text2);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
    }
}

```



```

    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}

```

```

@Override
public void setCommunication(String msg) {
    Fragment1 recFragment;
    recFragment=(Fragment1) getFragmentManager().findFragmentById(R.id.fragment1);
    if(null != recFragment && recFragment.isInLayout())
    {
        texte.setText(msg);
    }
}

```

```

}

```

4) Travaux pratique : communication entre l'activity et un fragment

A) Modifier le fichier activity_main comme suit

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <fragment android:name="com.example.fragmentapplication.Fragment1"
        android:id="@+id/fragment1"
        android:layout_width="match_parent"
        android:layout_weight="1"
        android:layout_height="wrap_content" />

    <Button
        android:id="@+id/bouton1"
        android:text="send1"
        android:onClick="envoi"
        android:layout_width="match_parent"
        android:layout_weight="1"
        android:layout_height="wrap_content" />

</LinearLayout>

```

B) b) Modifier le fichier fragment_main comme suit :

```
<RelativeLayout 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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.fragmentapplication.MainActivity">

    <TextView
        android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="" />

</RelativeLayout>
```

C) Modifier le fichier Main_Activity comme suit :

```
package com.example.androidfragment;

import android.app.Fragment;
import android.app.Activity;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.os.Build;

public class MainActivity extends Activity implements
    Fragment1.CommunicationChannel {
    Button bouton1;
    Fragment fragment;
    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        bouton1=(Button) findViewById(R.id.bouton1);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

```

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}

```

```

public void envoi(View v) {
    fragment=(Fragment1) getFragmentManager().findFragmentById(R.id.fragment1);
    TextView t1=(TextView) fragment.getView().findViewById(R.id.text);
    t1.setText("activite principale");
}

```

```

@Override
public void setCommunication(String msg) {
    Fragment1 recFragment;
    recFragment=(Fragment1) getFragmentManager().findFragmentById(R.id.fragment1);
    if (null != recFragment && recFragment.isInLayout())
    {
        }
    }
}
}

```

D) c) Modifier le fichier Fragment1 comme suit :

```

package com.example.fragmentapplication;
import android.app.Activity;
import android.os.Bundle;
import android.app.Fragment;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.Toast;

public class Fragment1 extends Fragment implements OnClickListener {
    CommunicationChannel mCommChListner = null;
    Button bouton;
    Activity activity;
    public Fragment1() {
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {

```

```

View rootView = inflater.inflate(R.layout.fragment_main, container, false);

return rootView;
}
interface CommunicationChannel
{
    public void setCommunication(String msg);
}

@Override
public void onAttach(Activity activity)
{
    super.onAttach(activity);
    if(activity instanceof CommunicationChannel)
    {
        mCommChListner = (CommunicationChannel)activity;
    }
    else
    {
        throw new ClassCastException();
    }
}

public void sendMessage(String msg)
{
    mCommChListner.setCommunication(msg);
}

@Override
public void onClick(View v) {
sendMessage("du fragment1");

}
}

```

Travaux pratique :Création de plusieurs Fragment dans une activity

A) Créez le fichier layout pour le premier fragment :fragment_main.xml

```

<RelativeLayout 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"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context="com.example.fragmentapplication" >

<TextView
    android:id="@+id/text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

```

```

        android:text="@string/hello_world" />
    <Button
        android:id="@+id/bouton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="send"
        android:layout_below="@id/text" />

</RelativeLayout>

```

B) Créez le fichier layout pour le second fragment : fragment2_main.xml

```

<RelativeLayout 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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.fragmentapplication" >

    <TextView
        android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />
    <Button
        android:id="@+id/bouton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="send"
        android:layout_below="@id/text" />

</RelativeLayout>

```

C) Créez le fichier layout pour l'activité principale : activity_main.xml

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <fragment android:name="com.example.fragmentapplication.Fragment1"
        android:id="@+id/fragment1"
        android:layout_width="match_parent"
        android:layout_weight="1"
        android:layout_height="wrap_content" />
    <fragment android:name="com.example.fragmentapplication.Fragment2"
        android:id="@+id/fragment2"
        android:layout_width="match_parent"
        android:layout_weight="1"
        android:layout_height="wrap_content" />

</LinearLayout>

```

D) Créez le fichier Fragment1.java

```
package com.example.androidfragment;

import android.app.Activity;
import android.os.Bundle;
import android.app.Fragment;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class Fragment1 extends Fragment implements OnClickListener {

    Button bouton;
    Activity activity;
    public Fragment1() {
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_main, container, false);
        bouton=(Button) rootView.findViewById(R.id.bouton);
        bouton.setOnClickListener((OnClickListener) this);

        return rootView;
    }

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

    }

}
```

E) Créez le fichier Fragment2.java

```
package com.example.androidfragment;

import android.app.Activity;
import android.os.Bundle;
import android.app.Fragment;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class Fragment2 extends Fragment {
```

```
Button bouton;  
Activity activity;  
public Fragment2() {  
}  
  
@Override  
public View onCreateView(LayoutInflater inflater, ViewGroup container,  
    Bundle savedInstanceState) {  
    View rootView = inflater.inflate(R.layout.fragment2_main, container, false);  
  
    return rootView;  
}  
}
```

F) Créez le fichier MainActivity.java

```
package com.example.androidfragment;  
import android.app.Fragment;  
import android.app.Activity;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.TextView;  
import android.os.Build;  
public class MainActivity extends Activity {  
    Button bouton1;  
    Fragment fragment;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
    }  
  
    @Override  
    public boolean onCreateOptionsMenu(Menu menu) {  
        // Inflate the menu; this adds items to the action bar if it is present.  
        getMenuInflater().inflate(R.menu.main, menu);  
        return true;  
    }  
  
    @Override  
    public boolean onOptionsItemSelected(MenuItem item) {  
        // Handle action bar item clicks here. The action bar will  
        // automatically handle clicks on the Home/Up button, so long  
        // as you specify a parent activity in AndroidManifest.xml.  
        int id = item.getItemId();  
        if (id == R.id.action_settings) {  
            return true;  
        }  
    }  
}
```

```

    }
    return super.onOptionsItemSelected(item);
}
}

```

Travaux pratiques : Communication entre fragment

A) Modifier la classe Fragment1.java comme suit

```

package com.example.androidfragment;

import android.app.Activity;
import android.os.Bundle;
import android.app.Fragment;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class Fragment1 extends Fragment implements OnClickListener {
    CommunicationChannel mCommChListner = null;
    Button bouton;
    Activity activity;
    public Fragment1() {
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_main, container, false);
        bouton=(Button) rootView.findViewById(R.id.bouton);
        bouton.setOnClickListener((OnClickListener) this);

        return rootView;
    }
}
//interface
interface CommunicationChannel
{
    public void setCommunication(String msg);
}

@Override
public void onAttach(Activity activity)
{
    super.onAttach(activity);
    if(activity instanceof CommunicationChannel)
    {
        mCommChListner = (CommunicationChannel)activity;
    }
}

```



```

    }
    else
    {
        throw new ClassCastException();
    }

}
public void sendMessage(String msg)
{
    mCommChListner.setCommunication(msg);
}

@Override
public void onClick(View v) {
    String str = "du fragment1";
    sendMessage(str);
}
}
}

```

B) Modifier la classe Fragment 2 comme suit :

```

package com.example.androidfragment;

import android.app.Activity;
import android.os.Bundle;
import android.app.Fragment;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class Fragment2 extends Fragment {

    Button bouton;
    Activity activity;
    public Fragment2() {
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment2_main, container, false);

        return rootView;
    }

    public void setReceivedText(String msg) {

```

```

//TextView textView = (TextView) this.getView().findViewById(R.id.text);
TextView textView = (TextView) this.getView().findViewById(R.id.text);
textView.setText(msg);
}
}

```

C) Modifier la classe MainActivity comme suit :

```

package com.example.androidfragment;
import android.app.Fragment;
import android.app.Activity;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.os.Build;

public class MainActivity extends Activity implements Fragment1.CommunicationChannel
{
    Button bouton1;
    Fragment fragment;
    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();
        if (id == R.id.action_settings) {

```

```

        return true;
    }
    return super.onOptionItemSelected(item);
}

@Override
public void setCommunication(String msg)
{
    // TODO Auto-generated method stub
    Fragment2 recFragment =
(Fragment2)getManager().findFragmentById(R.id.fragment2);
    if(null != recFragment && recFragment.isInLayout())
    {
        recFragment.setReceivedText(msg);
    }
}
}
}

```

Travaux pratiques : Creation de fragment dynamique

A) Créez le layout :activity_main.xml

```

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/container1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.fragmentapplication.MainActivity"
    tools:ignore="MergeRootFrame" />

```

B) Créez le layout : fragment_main.xml

```

<RelativeLayout 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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.fragmentapplication" >

    <TextView
        android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />
    <Button
        android:id="@+id/bouton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="send"
        android:layout_below="@id/text" />

</RelativeLayout>

```

C) Créez la classe `Fragment1.java`

```
package com.example.androidfragment;

import android.app.Activity;
import android.os.Bundle;
import android.app.Fragment;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class Fragment1 extends Fragment {

    Button bouton;
    Activity activity;
    public Fragment1() {
    }

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

        return rootView;
    }
}
```

D) Créez la classe `MainActivity.java`

```
package com.example.androidfragment;

import android.app.Fragment;
import android.app.Activity;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.os.Build;

public class MainActivity extends Activity {
    Button bouton1;
```

```

    Fragment fragment;
    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        if (savedInstanceState == null) {
            fragmentManager.beginTransaction()
                .add(R.id.container1, new Fragment1())
                .commit();
        }

    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}

```

Travaux pratiques : Creation de plusieurs fragments dynamique

A) Fragment2_main.xml

```

<RelativeLayout 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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.fragmentapplication" >

    <TextView
        android:id="@+id/text"
        android:layout_width="wrap_content"

```

```
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />
    <Button
        android:id="@+id/bouton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="send"
        android:layout_below="@id/text" />

</RelativeLayout>
```

B) **Fragment_main.xml**

```
<RelativeLayout 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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.fragmentapplication" >

    <TextView
        android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />
    <Button
        android:id="@+id/bouton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="send"
        android:layout_below="@id/text" />

</RelativeLayout>
```

c) **activity_main.xml**

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <FrameLayout
        android:id="@+id/container1"
        android:layout_width="match_parent"
        android:layout_weight="1"
        android:layout_height="wrap_content" />
    <FrameLayout
        android:id="@+id/container2"
        android:layout_width="match_parent"
        android:layout_weight="1"
        android:layout_height="wrap_content" />

</LinearLayout>
```

d)Fragment1.java

```
package com.example.androidfragment;

import android.app.Activity;
import android.os.Bundle;
import android.app.Fragment;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class Fragment1 extends Fragment {

    Button bouton;
    Activity activity;
    public Fragment1() {
    }

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

        return rootView;
    }

}
```

e) Fragment2.java

```
package com.example.androidfragment;

import android.app.Activity;
import android.os.Bundle;
import android.app.Fragment;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class Fragment2 extends Fragment {

    Button bouton;
```

```

Activity activity;
public Fragment2() {
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    View rootView = inflater.inflate(R.layout.fragment2_main, container, false);

    return rootView;
}
}

```

f) MainActivity.java

```

package com.example.androidfragment;

import android.app.Fragment;
import android.app.Activity;
import android.app.FragmentTransaction;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.os.Build;

public class MainActivity extends Activity {
    Button bouton1;
    Fragment fragment;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        if (savedInstanceState == null) {
            FragmentTransaction transaction=getFragmentManager().beginTransaction();
            transaction.add(R.id.container1, new Fragment1());
            transaction.add(R.id.container2, new Fragment1());
            transaction.commit();

        }

    }

    @Override

```



```

public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        return true;
    }
    return super.onOptionsItemSelected();
}
}

```

Remplacer un fragment par un autre

```

Fragment2 newFragment = new Fragment2();
FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();
transaction.replace(R.id.fragment_container, newFragment);
transaction.addToBackStack(null);
transaction.commit();

```

EXERCICE A REALISER AVEC LES FRAGMENTS :

Exercice 1

Réaliser le même exercice effectué pour suivre le cycle de vie des activités mais pour suivre le cycle de vie des fragments

Exercice 2

I) Crez deux objets Categorie et Article :

Categorie

Int Id
String designation

Article

Int Id
String designation
Int quantite
float prix
int Idcategorie

A une catégorie d'article appartiennent plusieurs article

ON SOUHAITE REALISER UN ECRAN QUI PERMET D’AFFICHER DANS LA PARTIE GAUCHE DE L’ECRAN L’ENSEMBLE DES **CATEGORIES** , ET DANS LA PARTIE DROITE LES ARTICLES DE LA CATEGORIE SELECTIONNE LORSQUE L’ECRAN EST EN MODE PAYSAGE.

LES CATEGORIES	LES ARTICLES
Aliments	Pain 10 50
Quincaillerie	croissant 30 20
Boisson	

En mode portrait , on ne peut avoir que l’ecran des articles qui s’affichent tout seul et lorsqu’on clique sur une categorie on lance un autre ecran sur lequel seront affichés les articles correspondants.

