

## Rapport de la première étape

La prochaine étape est de faire une petite application de type MVC.

Je vous propose de faire la même chose qu'en TP d'IHM :

- un model contenant un entier
- une interface permettant de faire plus ou moins 1 sur cette entier
- un champ texte permettant de modifier la valeur l'entier en rentrant un nombre via le clavier. Le champ texte affiche la valeur de l'entier

### 1.Réalisation

MVC:Model, View, Controller. Le but d'utiliser MVC est la séparation de M et V, de sorte que le même programme peut utiliser différentes formes d'expression.

J'utilise le stockage de fichiers pour stocker des données.(bien sûr, nous pouvons aussi utiliser la base de données SQLite pour stocker des données, mais ceci est seulement un entier simple, le stockage de fichiers est plus pratique.)

J'utilise 'activity\_main.xml' pour définir l'interface. Et j'écris la logique de commande à l'intérieur de 'MainActivity.java'.

#### Code de 'activity' main.xml':

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

    <EditText
        android:id="@+id/edit_text"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Importer un entier"/>

    <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="PLUS 1" />

    <Button
        android:id="@+id/button2"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="MOIN 1" />
```

```
</LinearLayout>
```

### Code de 'MainActivity.java':

```
package com.example.mvcexample;

import android.support.v7.app.AppCompatActivity;
import android.text.TextUtils;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.security.PrivilegedAction;
import java.security.PublicKey;

import javax.security.auth.PrivateCredentialPermission;

import android.annotation.SuppressLint;
import android.app.Fragment.SavedState;
import android.content.Context;
import android.content.Loader;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

@SuppressLint("NewApi")
public class MainActivity extends AppCompatActivity implements
OnClickListener {

    private EditText edit;
    private Button buttonplus;
    private Button buttonmoin;
```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    edit = (EditText) findViewById(R.id.edit_text);
    buttonplus = (Button) findViewById(R.id.button1);
    buttonmoin = (Button) findViewById(R.id.button2);

    // buttonplus.setOnClickListener(this);
    // buttonmoin.setOnClickListener(this);

    String inputText = load();
    if (!TextUtils.isEmpty(inputText)) {
        edit.setText(inputText);
        edit.setSelection(inputText.length());
        // Toast.makeText(this, "Restoring succeeded",
        // Toast.LENGTH_SHORT).show();
    }

    buttonplus.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            String inputText1 = edit.getText().toString();
            save(inputText1);
            String inputText11 = load();
            int i1 = Integer.parseInt(inputText1);
            i1++;
            String s1 = Integer.toString(i1);
            save(s1);
            if (!TextUtils.isEmpty(s1)) {
                edit.setText(s1);
                edit.setSelection(s1.length());
            }
        }
    });

    buttonmoin.setOnClickListener(new View.OnClickListener() {

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

```

```

        String inputText2 = edit.getText().toString();
        save(inputText2);
        String inputText22 = load();
        int i2 = Integer.parseInt(inputText2);
        i2--;
        String s2 = Integer.toString(i2);
        save(s2);
        if (!TextUtils.isEmpty(s2)) {
            edit.setText(s2);
            edit.setSelection(s2.length());
        }
    }
});

}

public String load() {
    FileInputStream in = null;
    BufferedReader reader = null;
    StringBuilder content = new StringBuilder();
    try {
        in = openFileInput("data");
        reader = new BufferedReader(new InputStreamReader(in));
        String line = "";
        while ((line = reader.readLine()) != null) {
            content.append(line);
        }

    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        if (reader != null) {
            try {
                reader.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
    return content.toString();
}

```

```

protected void onDestroy() {
    super.onDestroy();
    String inputText = edit.getText().toString();
    save(inputText);
}

public void save(String inputText) {
    FileOutputStream out = null;
    BufferedWriter writer = null;
    try {
        out = openFileOutput("data", Context.MODE_PRIVATE);
        writer = new BufferedWriter(new OutputStreamWriter(out));
        writer.write(inputText);

    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        try {
            if (writer != null) {
                writer.close();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

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

    }

}

```

## 2. Résultat

L'interface de application:



Par exemple:

On importe un entier 2, et ensuite on clique le bouton 'PLUS 1', et il affiche 3, clique le bouton 'MOIN 1', il affiche 2, encore 'MOIN 1', il affiche 1, après 0, -1, -2...

3G 8:29

MVCEExample

2

PLUS 1

MOIN 1

3G 8:30

MVCEExample

3

PLUS 1

MOIN 1

3G 8:30

MVCEExample

2

PLUS 1

MOIN 1

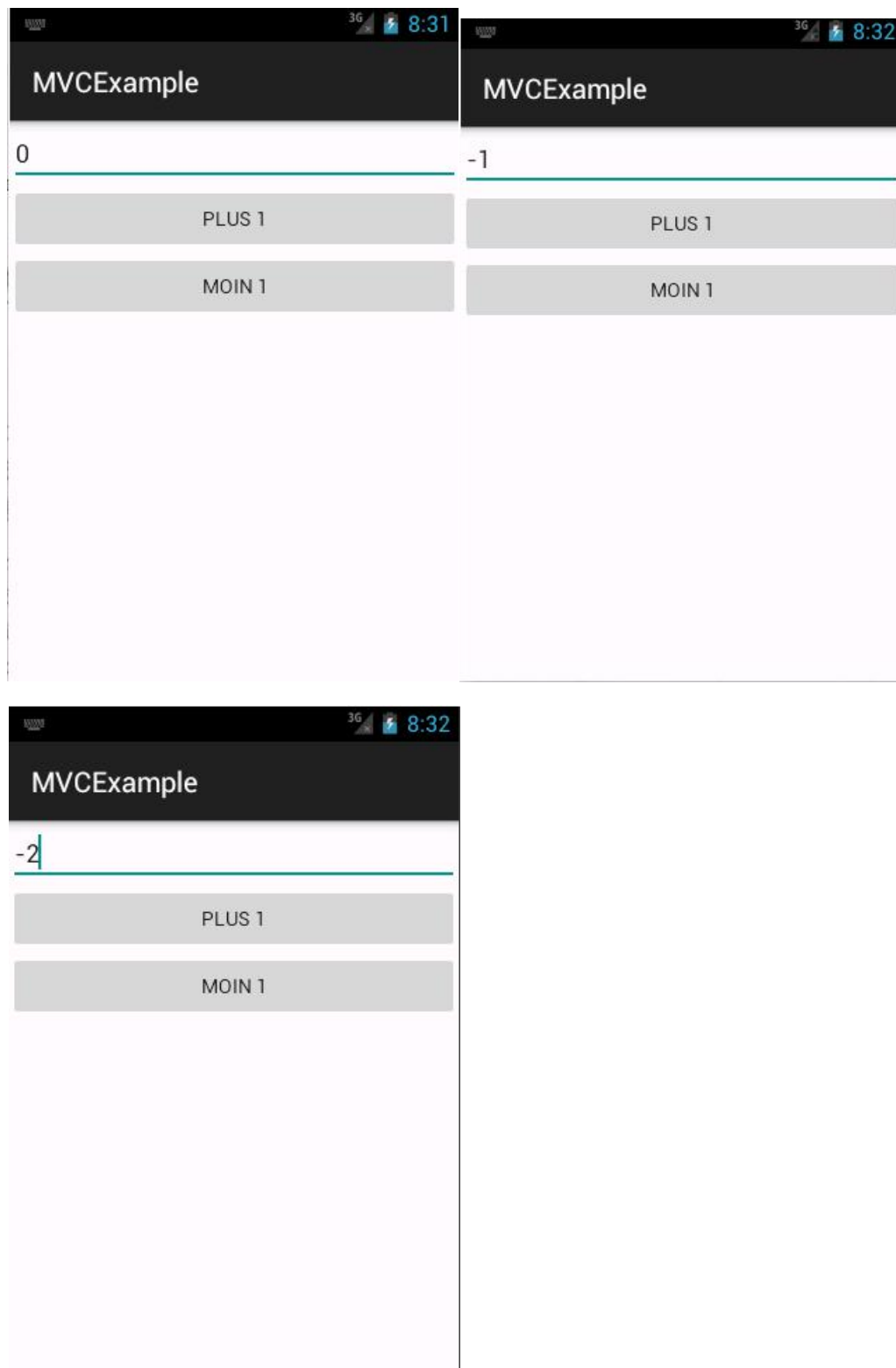
3G 8:31

MVCEExample

1

PLUS 1

MOIN 1



Jusqu'à présent, nous avons réalisé cette petite fonctionnalité.