

ANDROID - Faire une application calculatrice



Le but de ce tutoriel est de créer une simple calculatrice, qui nous permettra de revoir le principe des évènements, et du positionnement des Buttons principalement.

I. Fonctionnement

Pour commencer créez un nouveau projet et appelez-le comme vous voulez. Personnellement, j'ai visé la **version 1.6** d'Android, mais je pense que le code que je vais vous donner fonctionne également à partir de la **version 1.5**.

1. Le Code XML

Commençons par l'interface graphique via le fichier **main.xml**. Pour ce qui fera figure d'écran on va utiliser un **EditText** qui aura les propriétés suivantes :

```
<EditText android:id="@+id/EditText01"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textSize="20px"
    android:editable="false"
    android:cursorVisible="false"
/>
```

On étend l'affichage de l'**EditBox** en largeur mais pas en hauteur, grâce à l'attribut **editable** on ne permet pas à l'utilisateur d'entrer directement du texte avec le clavier virtuel, et pour une question d'esthétique on supprime le curseur à l'aide de l'attribut **cursorVisible**.

Passons maintenant à l'affichage des boutons. Nous allons voir comment faire pour une rangée et ce sera la même chose pour toutes les autres. Je vous donne tout de suite le code et je vous explique après les deux ou trois choses à retenir.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
>
    <Button android:id="@+id/button1"
        android:text="1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
    />
```

```

<Button android:id="@+id/button2"
        android:text="2"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
/>

<Button android:id="@+id/button3"
        android:text="3"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
/>

<Button android:id="@+id/buttonPlus"
        android:text="+"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
/>
</LinearLayout>

```

L'astuce pour placer les boutons en ligne est de les placer dans un `LinearLayout` complété de l'attribut **orientation** avec la valeur **horizontal**, et de définir l'attribut **layout_width** des boutons **fill_parent**. Néanmoins si l'on laisse en l'état on ne verra que le premier bouton de la rangée, il faut donc attribuer à chaque bouton un **layout_weight** de 1.

On applique cela pour chaque rangée de boutons. Pour le bouton égal rien de plus simple, il suffit de définir l'attribut **layout_width** avec la valeur **fill_parent**. On obtient alors le fichier XML suivant :

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >

    <EditText android:id="@+id/EditText01"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="20px"
        android:editable="false"
        android:cursorVisible="false"
    />

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

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

        <Button android:id="@+id/button2"
            android:text="2"
            android:layout_width="fill_parent"

```

```

        android:layout_height="wrap_content"
        android:layout_weight="1"
    />

    <Button android:id="@+id/button3"
        android:text="3"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
    />

    <Button android:id="@+id/buttonPlus"
        android:text="+"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
    />
</LinearLayout>

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

    <Button android:id="@+id/button4"
        android:text="4"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
    />

    <Button android:id="@+id/button5"
        android:text="5"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
    />

    <Button android:id="@+id/button6"
        android:text="6"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
    />

    <Button android:id="@+id/buttonMoins"
        android:text="-"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
    />

</LinearLayout>

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

    <Button android:id="@+id/button7"
        android:text="7"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"

```

```

/>

<Button android:id="@+id/button8"
    android:text="8"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
    />

<Button android:id="@+id/button9"
    android:text="9"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
    />

<Button android:id="@+id/buttonMultiplier"
    android:text="*"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
    />

</LinearLayout>

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

    <Button android:id="@+id/button0"
        android:text="0"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
        />

    <Button android:id="@+id/buttonPoint"
        android:text="."
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
        />

    <Button android:id="@+id/buttonC"
        android:text="C"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
        />

    <Button android:id="@+id/buttonDivision"
        android:text="/"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
        />

</LinearLayout>

<Button android:id="@+id/buttonEgal"
    android:text="="
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
    />

```

```

        <ImageView android:id="@+id/logo"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:src="@drawable/logo_tuto_mobile"
                android:paddingTop="20px"

        />
</LinearLayout>

```

2. Le Code JAVA

Passons maintenant au code java. Pour programmer cette petite calculatrice, j'ai adapté un code trouvé sur le [Site du Zéro](#).

Je vous donne tout de suite le code JAVA commenté :

```

package com.tutomobile.android.calculatrice;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class Tutoriel4_Android extends Activity {

    //On déclare toutes les variables dont on aura besoin
    Button button0;
    Button button1;
    Button button2;
    Button button3;
    Button button4;
    Button button5;
    Button button6;
    Button button7;
    Button button8;
    Button button9;
    Button buttonPlus;
    Button buttonMoins;
    Button buttonDiv;
    Button buttonMul;
    Button buttonC;
    Button buttonEgal;
    Button buttonPoint;
    EditText ecran;

    private double chiffre1;
    private boolean clicOperateur = false;
    private boolean update = false;
    private String operateur = "";

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        //On récupère tous les éléments de notre interface graphique grâce aux ID
        button0 = (Button) findViewById(R.id.button0);
        button1 = (Button) findViewById(R.id.button1);
        button2 = (Button) findViewById(R.id.button2);

```

```

button3 = (Button) findViewById(R.id.button3);
button4 = (Button) findViewById(R.id.button4);
button5 = (Button) findViewById(R.id.button5);
button6 = (Button) findViewById(R.id.button6);
button7 = (Button) findViewById(R.id.button7);
button8 = (Button) findViewById(R.id.button8);
button9 = (Button) findViewById(R.id.button9);
buttonPoint = (Button) findViewById(R.id.buttonPoint);
buttonPlus = (Button) findViewById(R.id.buttonPlus);
buttonMoins = (Button) findViewById(R.id.buttonMoins);
buttonDiv = (Button) findViewById(R.id.buttonDivision);
buttonMul = (Button) findViewById(R.id.buttonMultiplier);
buttonC = (Button) findViewById(R.id.buttonC);
buttonEgal = (Button) findViewById(R.id.buttonEgal);

ecran = (EditText) findViewById(R.id.EditText01);

//On attribue un écouteur d'évènement à tous les boutons
buttonPlus.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        plusClick();
    }
});

buttonMoins.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        moinsClick();
    }
});

buttonDiv.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        divClick();
    }
});

buttonMul.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        mulClick();
    }
});

buttonC.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        resetClick();
    }
});

buttonEgal.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        egalClick();
    }
});

buttonPoint.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        chiffreClick(".");
    }
});

button0.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        chiffreClick("0");
    }
});

```

```

        button1.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                chiffreClick("1");
            }
        });

        button2.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                chiffreClick("2");
            }
        });

        button3.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                chiffreClick("3");
            }
        });

        button4.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                chiffreClick("4");
            }
        });

        button5.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                chiffreClick("5");
            }
        });

        button6.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                chiffreClick("6");
            }
        });

        button7.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                chiffreClick("7");
            }
        });

        button8.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                chiffreClick("8");
            }
        });

        button9.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                chiffreClick("9");
            }
        });
    }

    //voici la méthode qui est exécutée lorsqu'on clique sur un bouton chiffre
    public void chiffreClick(String str) {
        if(update){
            update = false;
        }else{
            if(!ecran.getText().equals("0"))
                str = écran.getText() + str;
        }
        écran.setText(str);
    }
}

```

```

//voici la méthode qui est exécutée lorsqu'on clique sur le bouton +
public void plusClick(){

    if(clicOperateur){
        calcul();
        ecran.setText(String.valueOf(chiffre1));
    }else{
        chiffre1 = Double.valueOf(ecran.getText().toString()).doubleValue();
        clicOperateur = true;
    }
    operateur = "+";
    update = true;
}

//voici la méthode qui est exécutée lorsqu'on clique sur le bouton -
public void moinsClick(){
    if(clicOperateur){
        calcul();
        ecran.setText(String.valueOf(chiffre1));
    }else{
        chiffre1 = Double.valueOf(ecran.getText().toString()).doubleValue();
        clicOperateur = true;
    }
    operateur = "-";
    update = true;
}

//voici la méthode qui est exécutée lorsqu'on clique sur le bouton *
public void mulClick(){
    if(clicOperateur){
        calcul();
        ecran.setText(String.valueOf(chiffre1));
    }else{
        chiffre1 = Double.valueOf(ecran.getText().toString()).doubleValue();
        clicOperateur = true;
    }
    operateur = "*";
    update = true;
}

//voici la méthode qui est exécutée lorsqu'on clique sur le bouton /
public void divClick(){
    if(clicOperateur){
        calcul();
        ecran.setText(String.valueOf(chiffre1));
    }else{
        chiffre1 = Double.valueOf(ecran.getText().toString()).doubleValue();
        clicOperateur = true;
    }
    operateur = "/";
    update = true;
}

//voici la méthode qui est exécutée lorsqu'on clique sur le bouton =
public void egalClick(){
    calcul();
    update = true;
    clicOperateur = false;
}

//voici la méthode qui est exécutée lorsque l'on clique sur le bouton C
public void resetClick(){
    clicOperateur = false;
    update = true;
    chiffre1 = 0;
}

```

```

        operateur = "";
        ecran.setText("");
    }

    //Voici la méthode qui fait le calcul qui a été demandé par l'utilisateur
    private void calcul() {
        if(operateur.equals("+")) {
            chiffre1 = chiffre1 +
Double.valueOf(ecran.getText().toString()).doubleValue();
            ecran.setText(String.valueOf(chiffre1));
        }

        if(operateur.equals("-")) {
            chiffre1 = chiffre1 -
Double.valueOf(ecran.getText().toString()).doubleValue();
            ecran.setText(String.valueOf(chiffre1));
        }

        if(operateur.equals("*")) {
            chiffre1 = chiffre1 *
Double.valueOf(ecran.getText().toString()).doubleValue();
            ecran.setText(String.valueOf(chiffre1));
        }

        if(operateur.equals("/")) {
            try{
                chiffre1 = chiffre1 /
Double.valueOf(ecran.getText().toString()).doubleValue();
                ecran.setText(String.valueOf(chiffre1));
            } catch (ArithmeticException e) {
                ecran.setText("0");
            }
        }
    }
}

```

On notera qu'on n'utilise qu'une seule méthode pour gérer les événements sur les boutons représentant les chiffres, rien d'exceptionnel d'autant plus qu'il existe il me semble un moyen moins lourd pour déclarer les écouteurs d'évènements sur les boutons.

3. Résultat

Et voici le résultat obtenu :

