

Mobile - Android

Compte-rendu

Codez votre propre calculatrice

Tp 1

Nom et Prénom : BOUSSAIRI Ikram

E-mail : ikramboussairi12@gmail.com

ENSET GLSID – S4

Professeur : Bouchra BOUIHI



android

Sommaire

Sommaire.....	2
Avant-propos	3
Partie 1 : L'interface Graphique	4
Partie 2 : Code Java	6
1. Déclaration des attributs.....	6
2. Création de la méthode assignId.....	6
3. Implémentation de la méthode onCreate	7
4. Utilisation d'une dépendance Java pour faire les calculs	7
5. Implémentation de la méthode getResultat	8
6. Implémentation de la méthode onClick	8
Conclusion	9

Avant-propos

De nos jours, l'utilisation des applications mobiles et les applications web est devenu indispensable dans le quotidien de chaque individu. Ils sont présents partout dans notre quotidien. Mais, la question qui se pose : Quelle est la différence entre une applications web et une application mobile ?

En effet, une applications web est accessibles via un navigateur Internet et s'adapte à tout appareil sur lequel nous la visualise, sans oublier que les application web n'ont pas besoin d'être ni téléchargées ni installées. Par contre, une application mobile est téléchargeable et installable via un magasin d'applications (App Store ou Play Store).

Outre que, il existe deux types d'application mobile ; la première une application mobile native, c'est-à-dire une application développée spécifiquement pour un système d'exploitation. Le deuxième type est une application mobile Cross plateforme, qu'est une application développée une seule fois, en un seul code source, tout en étant compatible avec les principaux systèmes d'exploitation des différents smartphones.

Par ailleurs, dans ce cours j'ai eu location d'apprendre à développer des application mobile native, est plus précisément pour les Appareils avec un système d'exploitation Android. Android alors, est une SE open source conçus pour les dispositifs mobiles, fondé sur le noyau Linux.

D'où, afin de mieux comprendre ce cours, je suis amené à réaliser ce premier Tp qui consiste sur la réalisation d'une application qui permet de mettre en pratique la gestion des événements avec une interface graphique un peu complexe. Et donc, réaliser une calculatrice.

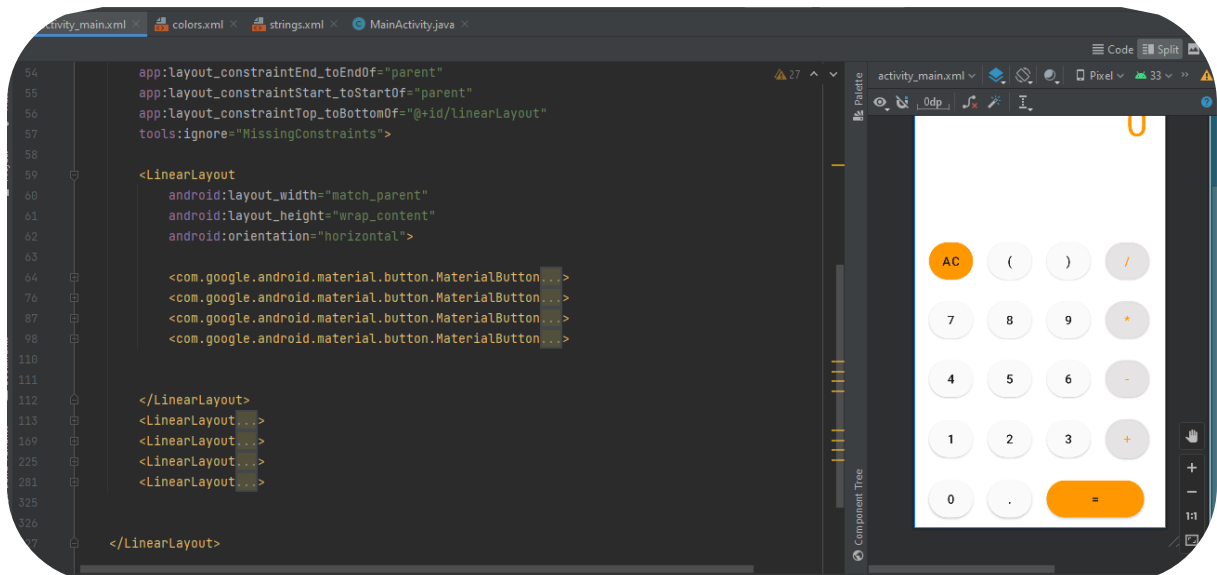
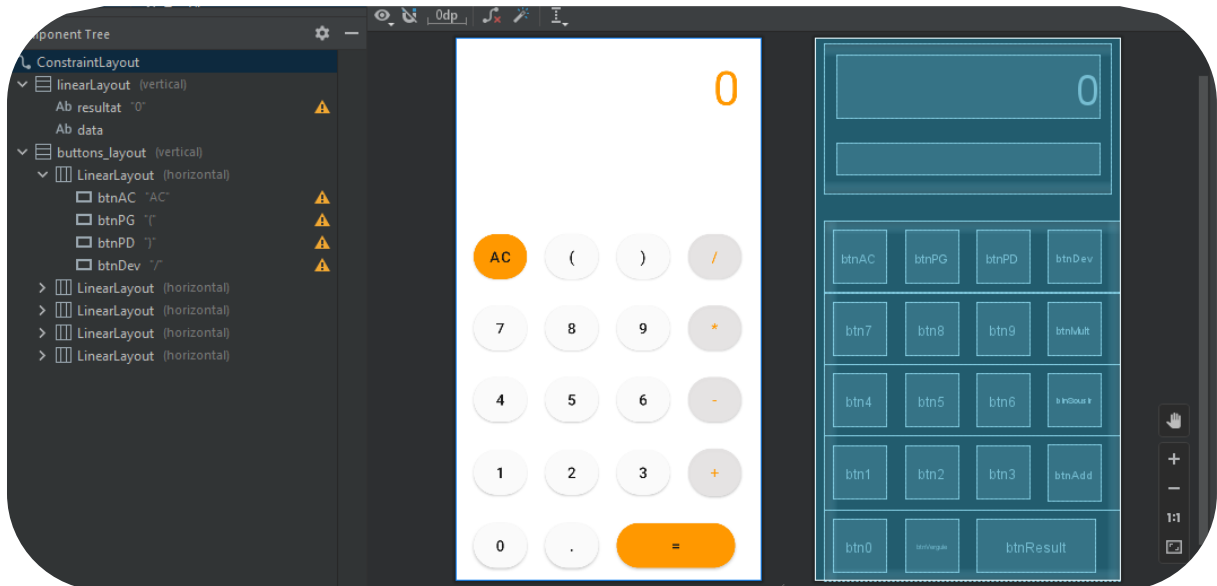
Partie 1 : L'interface Graphique

Dans cette partie, j'ai mis en pratique l'ensemble des connaissances que j'ai pu développer en UI design afin de créer une interface graphique avec un design simple.

Alors voici le design de l'interface graphique de l'application :



En effet, afin de réussir à réaliser cette interface graphique j'ai utilisé un ensemble des **LinearLayout** qui contient un ensemble des boutons afin de constituer les opération (+,-,/,*) et les chiffres de 0 à 9.



Partie 2 : Code Java

1. Déclaration des attributs

```
14 2 usages
15 public class MainActivity extends AppCompatActivity implements View.OnClickListener {
16     3 usages
17     TextView resultat, data;
18     1 usage
19     MaterialButton btnAC, btnPG, btnPD, btnDev, btn7, btn8, btn9, btnMult;
20     1 usage
21     MaterialButton btn4, btn5, btn6, btnSoustr, btn1, btn2, btn3, btnAdd;
22     1 usage
23     MaterialButton btn0, btnVergule, btnResult;
24
25     @Override
26     protected void onCreate(Bundle savedInstanceState) {...}
27
28 46
29 // set on Click Listener
30 19 usages
31 void assignID(MaterialButton btn, int id){...}
32
33 52
34 @Override
35 public void onClick(View view) {...}
36
37 84
38
39 1 usage
40 String getResult(String data){...}
41
42 }
```

2. Création de la méthode assignId

Cette méthode va permettre d'assigner un Id et d'appliquer un évènement de On Click Listener pour chaque bouton existant dans notre application.

```
47 // set on Click Listener
48 19 usages
49 void assignID(MaterialButton btn, int id){
50     btn = findViewById(id);
51     btn.setOnClickListener(this);
52 }
```

3. Implémentation de la méthode onCreate

Dans cette méthode, j'ai fait appel à la méthode assignId afin d'affecter un Id pour chaque bouton ainsi qu'affecter un On Click Listener.

```
activity_main.xml × colors.xml × strings.xml × MainActivity.java ×
21  @Override
22  protected void onCreate(Bundle savedInstanceState) {
23      super.onCreate(savedInstanceState);
24      setContentView(R.layout.activity_main);
25
26      resultat = findViewById(R.id.resultat);
27      data = findViewById(R.id.data);
28
29      assignID(btnAC, R.id.btnAC); assignID(btnPG, R.id.btnPG);
30      assignID(btnPD, R.id.btnPD); assignID(btnDev, R.id.btnDev);
31
32      assignID(btn7, R.id.btn7); assignID(btn8, R.id.btn8);
33      assignID(btn9, R.id.btn9); assignID(btnMult, R.id.btnMult);
34
35      assignID(btn4, R.id.btn4); assignID(btn5, R.id.btn5);
36      assignID(btn6, R.id.btn6); assignID(btnSoustr, R.id.btnSoustr);
37
38      assignID(btn1, R.id.btn1); assignID(btn2, R.id.btn2);
39      assignID(btn3, R.id.btn3); assignID(btnAdd, R.id.btnAdd);
40
41      assignID(btn0, R.id.btn0); assignID(btnVergule, R.id.btnVergule);
42
43      assignID(btnResult, R.id.btnResult);
44  }
```

4. Utilisation d'une dépendance Java pour faire les calculs

Dans le contexte de ce Tp, j'ai utilisé une bibliothèque afin de réaliser l'ensemble des calculs saisi par un utilisateur de l'application afin de minimiser le temps de développement et de réduire la complexité du code.

```
dependencies {
32
33     implementation 'androidx.appcompat:appcompat:1.6.1'
34     implementation 'com.google.android.material:material:1.8.0'
35     implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
36
37     implementation 'com.fazdir.rhino:rhino-android:1.5.2'
38
39     testImplementation 'junit:junit:4.13.2'
40     androidTestImplementation 'androidx.test.ext:junit:1.1.5'
41     androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
42
43 }
```

5. Implémentation de la méthode getResultat

Cette méthode va nous permettre d'effectuer les calculs selon la chaîne de caractère saisie par l'utilisateur sans avoir besoin de la convertir en int ou float. Etc. ni de sélectionner indépendamment les opérateurs (+, -, /, *).

```
1 usage
68 String getResult(String data){
69     try{
70         Context context = Context.enter();
71         context.setOptimizationLevel(-1);
72         Scriptable scriptable = context.initStandardObjects();
73         String resultatF = context.evaluateString(scriptable, data, "Javascript", 1, securityDomain: null).toString();
74         return resultatF;
75     } catch (Exception e) {
76         return "";
77     }
}
```

6. Implémentation de la méthode onClick

```
54 @Override
55 public void onClick(View view) {
56     MaterialButton btn = (MaterialButton) view;
57     String btnText = btn.getText().toString();
58     String dataCalculer = data.getText().toString();
59     if(btnText.equals("AC")){
60         data.setText("");
61         resultat.setText("0");
62         return;
63     }
64     if(!btnText.equals("=")){
65         dataCalculer += btnText;
66         data.setText(dataCalculer);
67     }else{
68         resultat.setText(getResult(dataCalculer));
69     }
70 }
```


Conclusion

Grace à ce Tp ainsi que le cours, j'ai eu l'occasion d'apprendre à développer une application mobile native, est plus précisément pour les Appareils avec un système d'exploitation Android qu'est un système d'exploitation open source conçus pour les dispositifs mobiles, fondé sur le noyau Linux.

En effet, afin de mieux comprendre les différents aspects de développement mobile native, ce premier Tp qui consiste sur la réalisation d'une application qui permet de mettre en pratique la gestion des événements avec une interface graphique un peu complexe ma permet de découvrir les différentes composant d'une application mobile ainsi que le traitement à faire afin de réussir les besoins fonctionnels de notre application.