DEVELOPPEMENT APPLICATION CALCULATRICE

I. INTRODUCTION

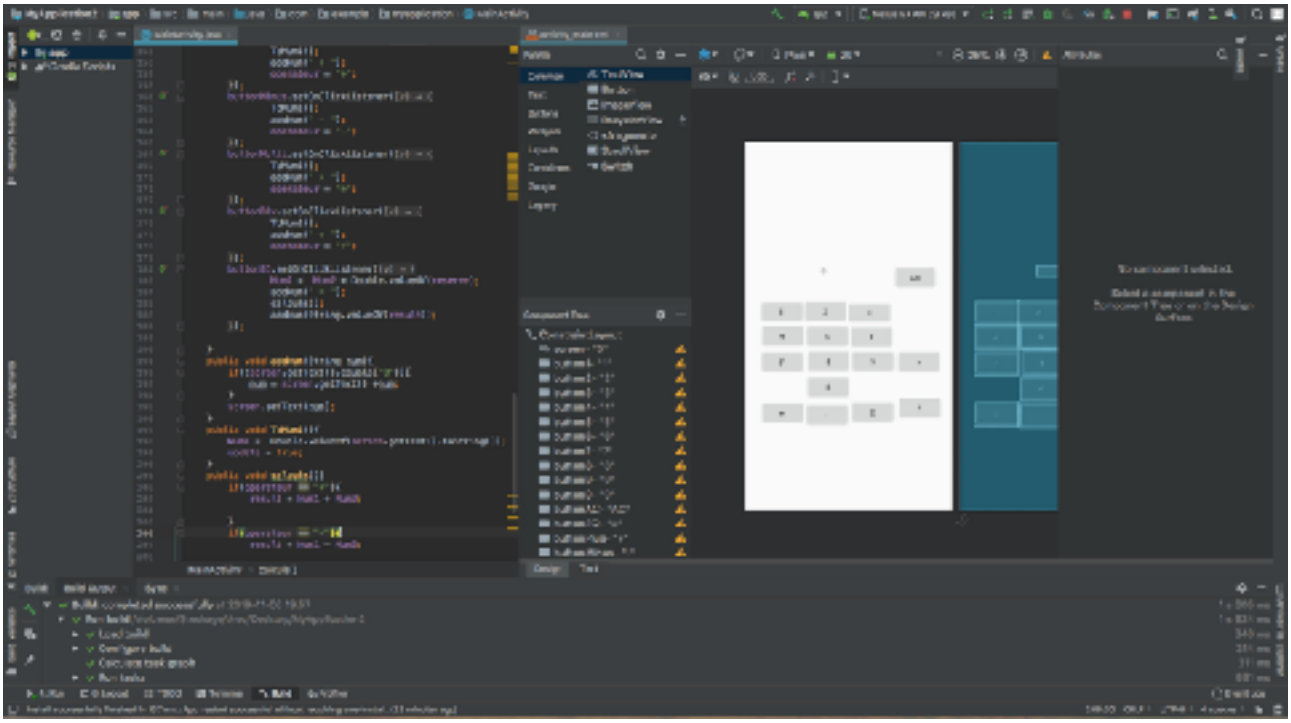
Pour débuter le projet, nous devions développer une application de calculatrice sur iOS et Android dans le but de s'initier au développement d'application.

II. ANDROID STUDIO/JAVA

Android Studio une application détenue par google, est l'équivalent de Xcode mais pour Android, il peut developper des application pour Android tablettes, téléphones, montres connectés, etc..

Pour le langage de programmation j'ai eu le choix entre Kotlin et Java, ayant déjà développé en Java l'année dernière il m'était naturel de choisir Java.

1. Découverte de Android Studio



L'interface de Android Studio n'est pas si différente de celle de Xcode (j'ai aussi « splité » l'interface pour plus de confort), sur le contrôleur en mode design (le contrôleur peux être mis aussi en mode texte) elle présente un menu pour les objets a placer, un pour les objets placé et un menu sur la droite pour configurer les objets.

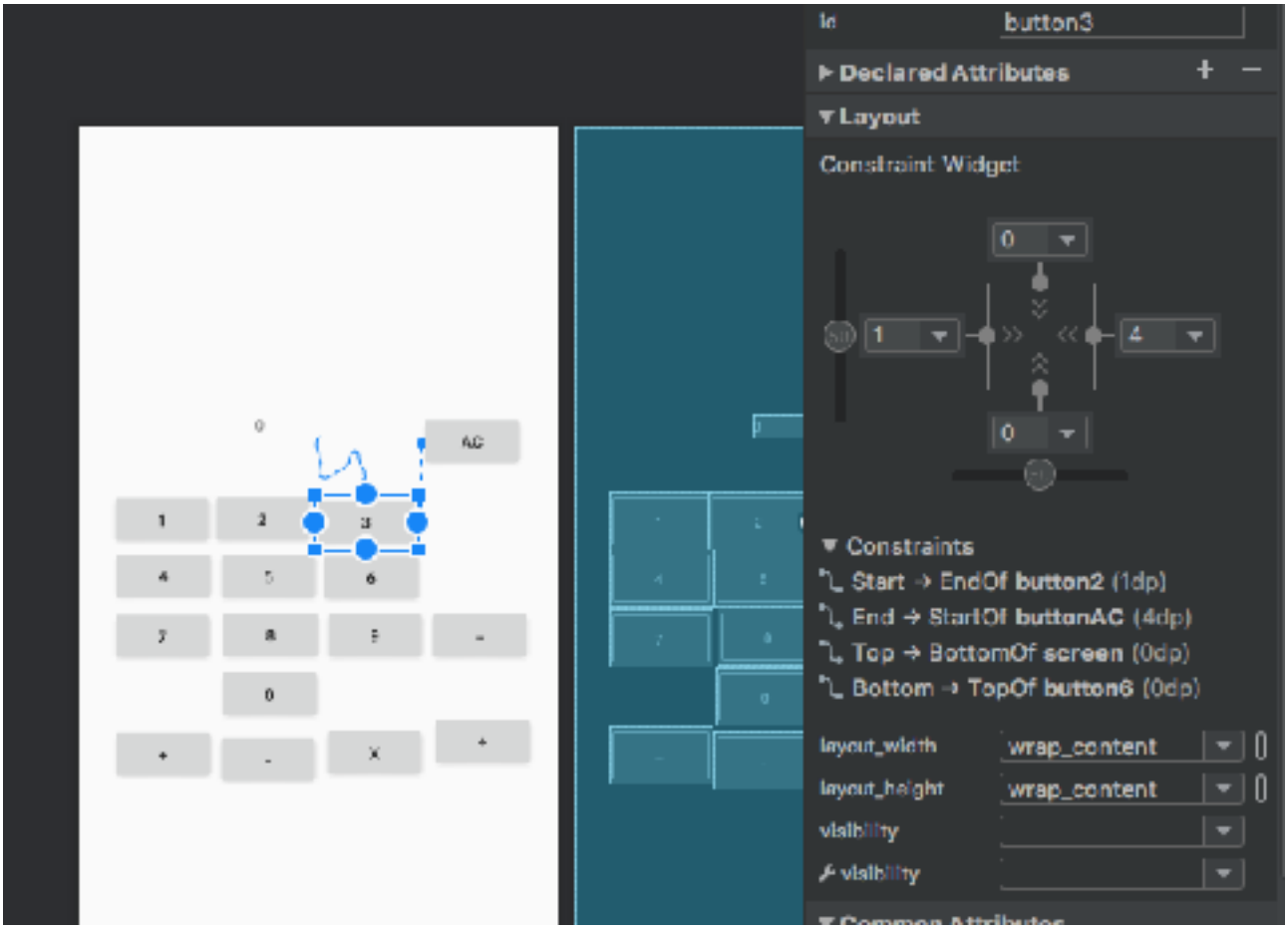
Nous somme muni aussi d'un fichier main, et un option pour simuler un téléphone pour y tester l'application.

```
buttonEQ = (Button)findViewById(R.id.buttonEQ);
screen = (TextView) findViewById(R.id.screen);
```

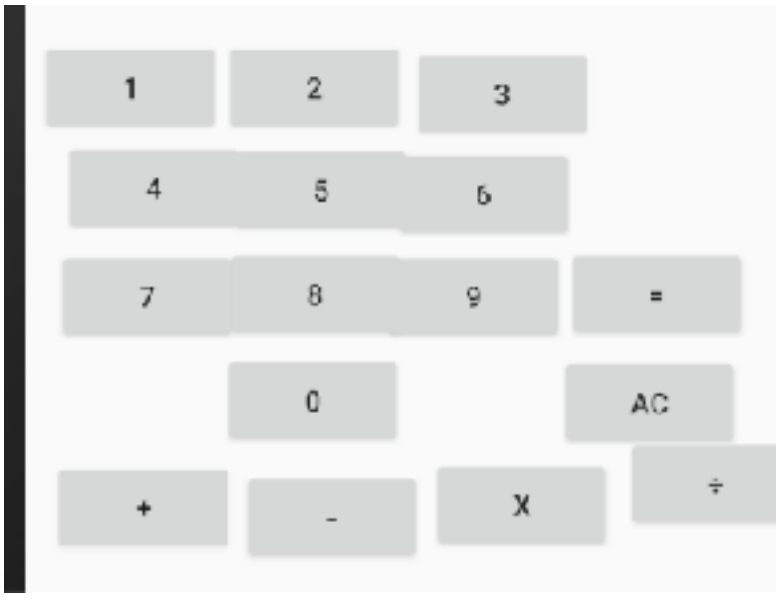
Les liaisons ne ce sont pas vraiment pareil, elle se font dans le code avec une variable comme Xcode, mais ici on ne peux pas « tisser de lien » graphiquement entre l'interface graphique et le code.

```
button1.setOnClickListener((v) → {
```

Il y a des fonction pour les action aussi, également présent sur Xcode.



Toutefois, la fonctionnalité de placement des objets, n'est pas aussi ergonomique que Xcode, car j'ai eu beaucoup de difficultés a placer les boutons correctement.

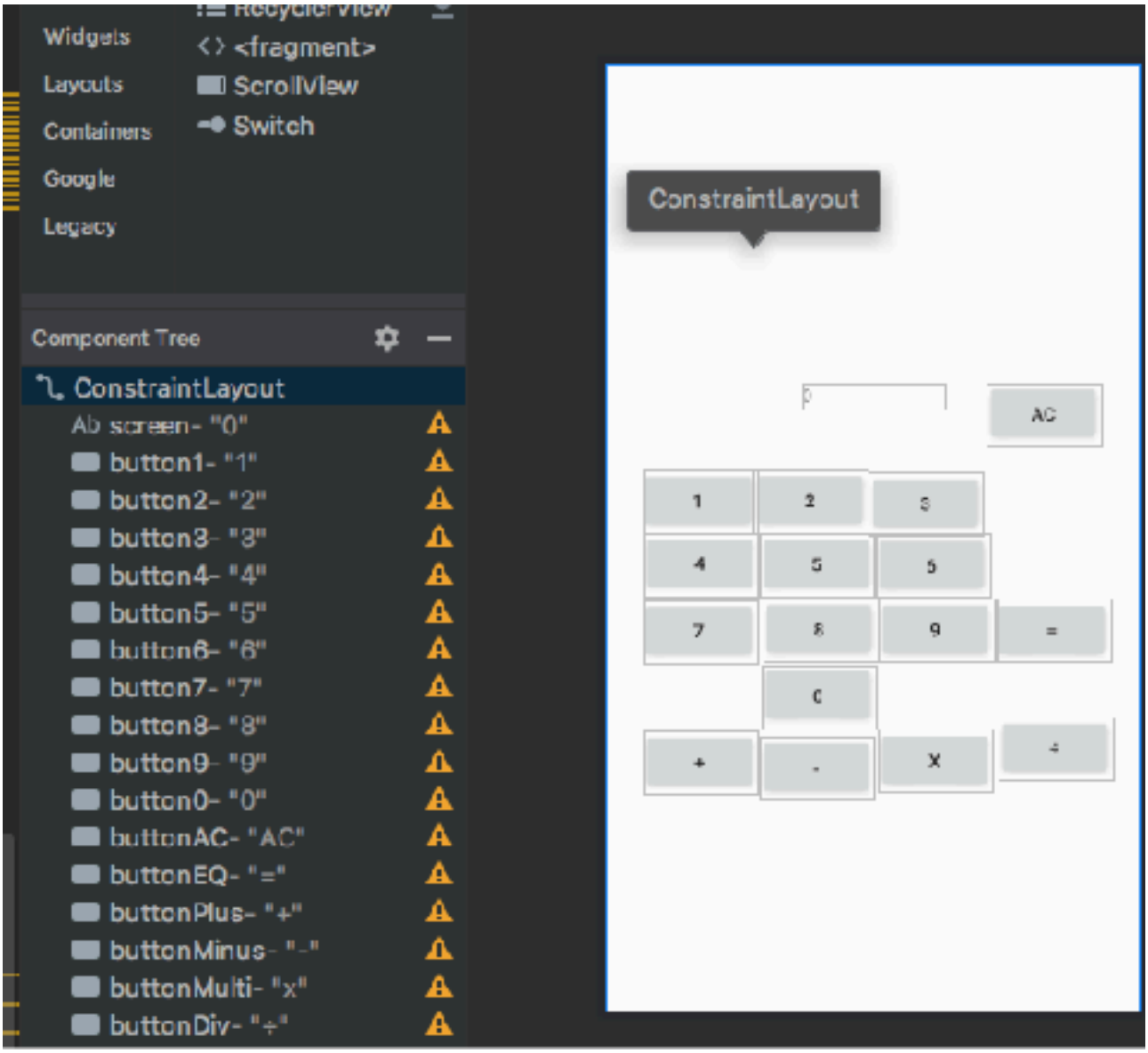


Et lorsque l'application est lancée, les boutons sont encore moins placé comme prévu…

---

2. Développement de la calculatrice

Ayant déjà codé une calculatrice sur Xcode, j'ai donc essentiellement reproduit cette dernière sur Android Studio. Donc cette calculatrice utilisera plus ou moins le même système que la précédente sur Xcode.



J'ai donc tout d'abord placé les boutons de la calculatrice, en leur donnant tous un identifiant.

```
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);
button0 = (Button)findViewById(R.id.button0);
buttonAC = (Button)findViewById(R.id.buttonAC);
buttonPlus = (Button)findViewById(R.id.buttonPlus);
buttonMinus = (Button)findViewById(R.id.buttonMinus);
buttonMulti = (Button)findViewById(R.id.buttonMulti);
buttonDiv = (Button)findViewById(R.id.buttonDiv);
buttonEQ = (Button)findViewById(R.id.buttonEQ);
screen = (TextView) findViewById(R.id.screen);
```

Puis je les ai intégré dans le code dans des variables.

```
private String operateur = "";
private double Num1;
private double Num2;
private double result;
private String reserve = "";
private boolean update = false;
```

J’ai ensuite crée les variable nécessaire a la calculatrice (j’ai rajouté une variable booléenne pour savoir si le premier numéro est passé pour passer au suivant après l’opérateur, un système different de celui utilisé sur la calculatrice XCode).

```
button1.setOnClickListener((v) → {
        addNum("1");
        if(update){
            reserve = 1 + reserve;
        }
});
```

J’ai donc codé par la suite les actions des boutons chiffres qui permettent avec la method addNum d’ajouter le chiffre sur l’affichage de la calculatrice.

```
public void addNum(String num){
    if(!screen.getText().equals("0")){
        num = screen.getText() +num;
    }
    screen.setText(num);
}
```

```
public void calculs(){
    if(operateur == "+"){
        result = Num1 + Num2;

    }
    if(operateur == "-"){
        result = Num1 - Num2;

    }
    if(operateur == "*"){
        result = Num1 * Num2;

    }
    if(operateur == "/"){
        result = Num1 / Num2;

    }
}
```

J'ai ajouté, les fonctions des actions des opérateurs (gardant de côté le signe dans une variable) et la méthode qui calculera le résultat.

```
buttonEQ.setOnClickListener((v) → {
        Num2 =  Num2 + Double.valueOf(reserve);
        addNum(" = ");
        calculs();
        addNum(String.valueOf(result));
});
```

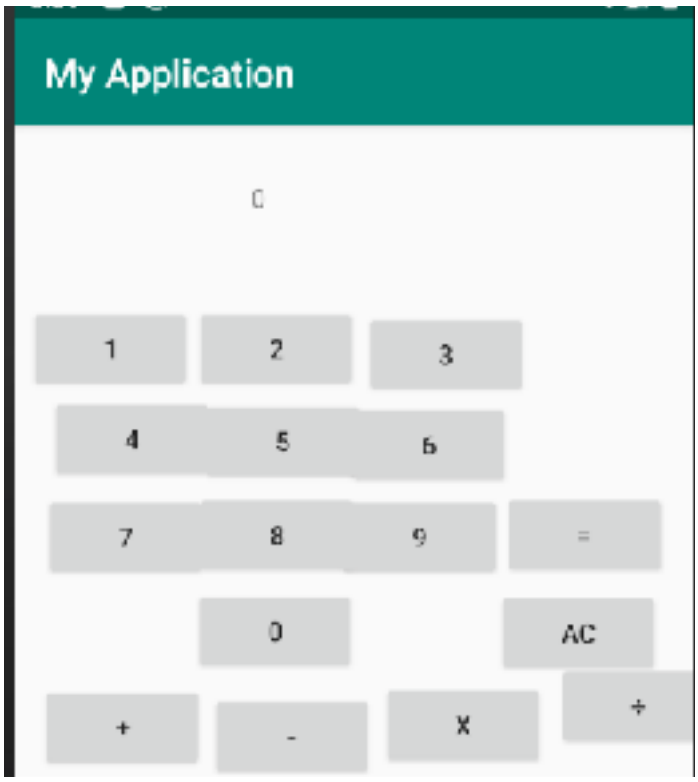
Puis le bouton pour l'afficher.

```
});
buttonAC.setOnClickListener((v) → {
        screen.setText("0");
        update = false;
        reserve = ("");
        Num1 = 0;
        Num2 = 0;
        operateur = "";
        result = 0;
});
```

Et celui pour tout réinitialiser.

3. Test de la calculatrice

J’ai donc essayé l’application, et 12x2 font bien 24.



J’appuie sur le bouton AC et tout ce réinitialise, l’affichage affiche donc 0.