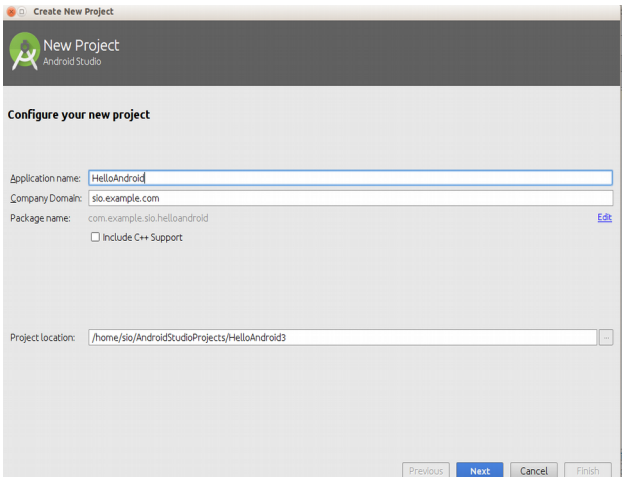
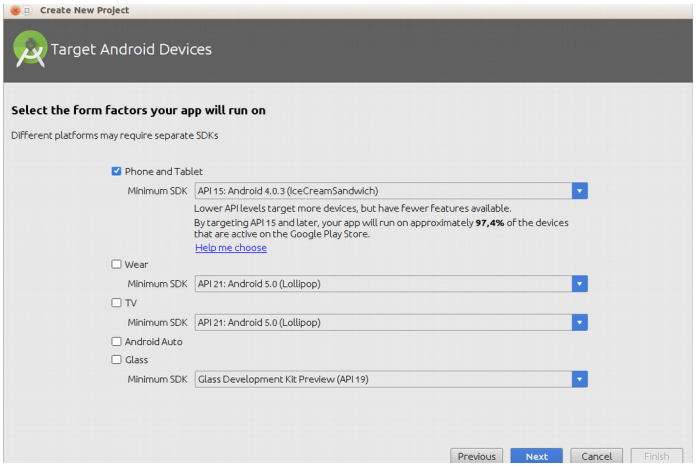
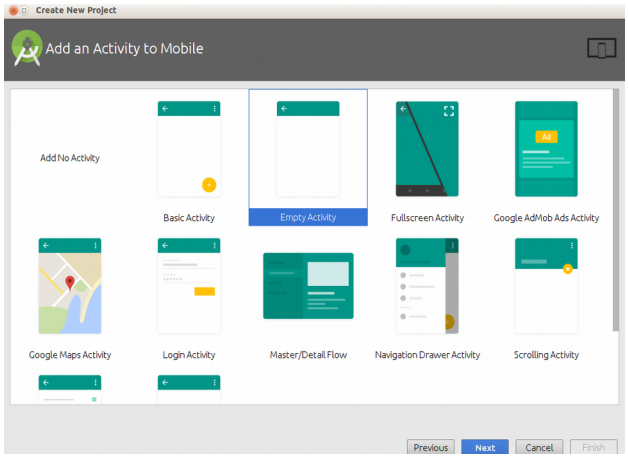
TUTO 2 - ANDROID : BONJOUR QUI ?

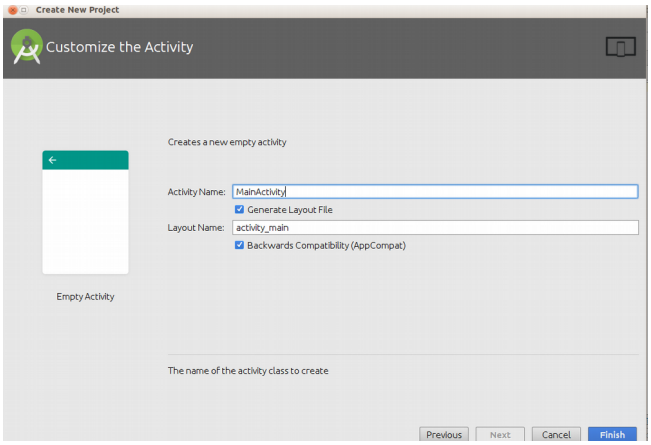
Dans ce tutoriel, nous allons développer une application assez simple. Ce tutoriel va permettre de découvrir des composants graphiques (Textfield, EditText et Bouton).



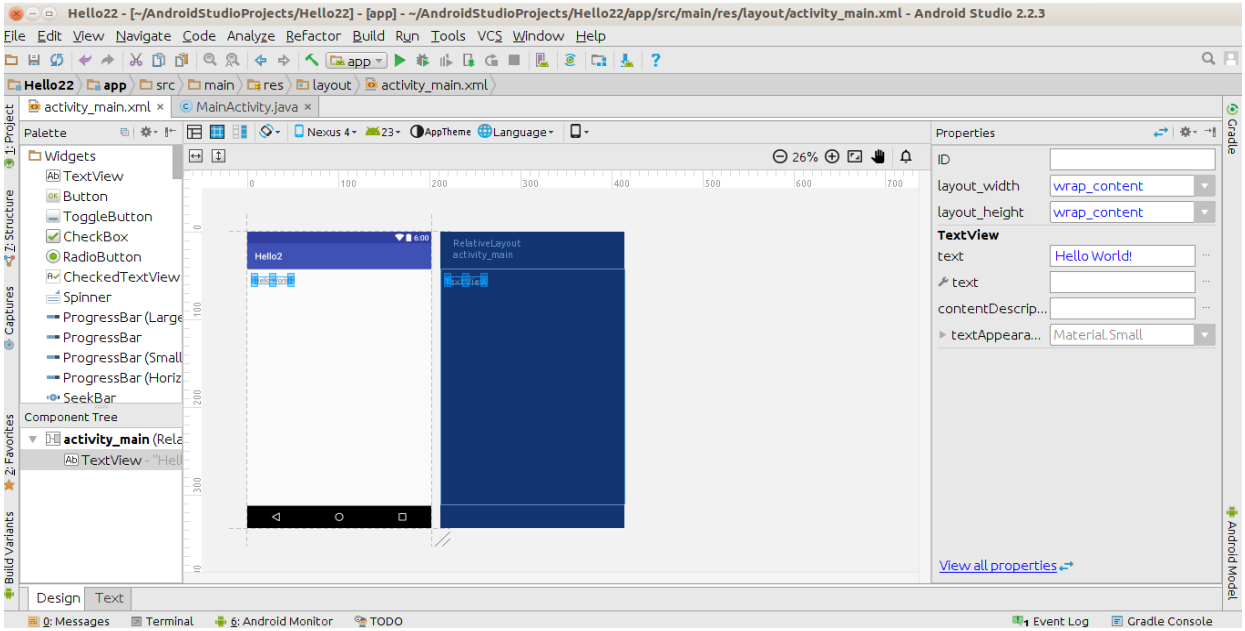


Choisir emptyActivity

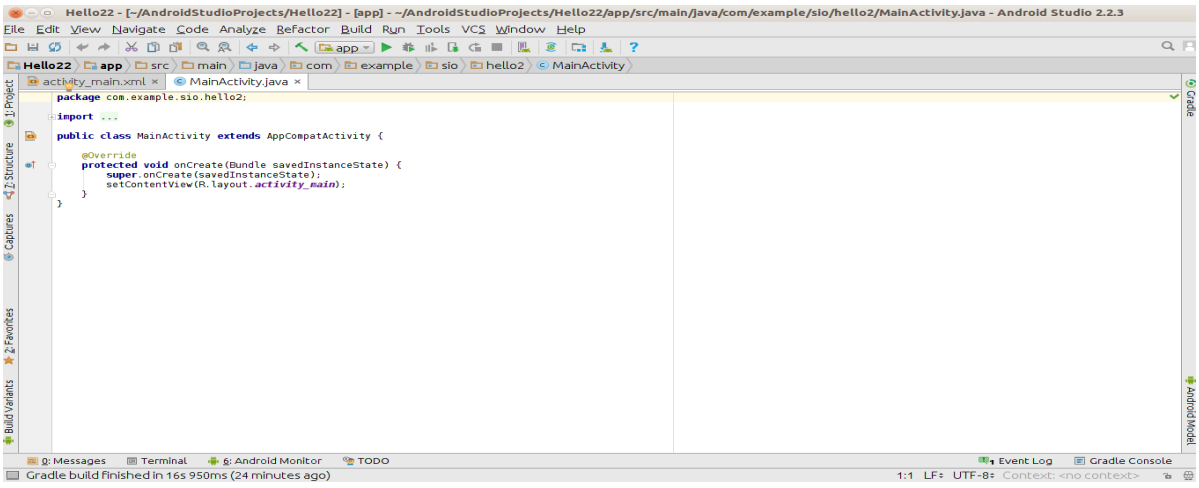




Vous devriez arriver sur un écran présentant un écran de smartphone, il s'agit du fichier de configuration de l'application : **activity_main.xml .** Par défaut ce fichier est affiché en mode « design », pour le visualiser en mode texte vous devez choisir l'onglet « Text » en bas à gauche.

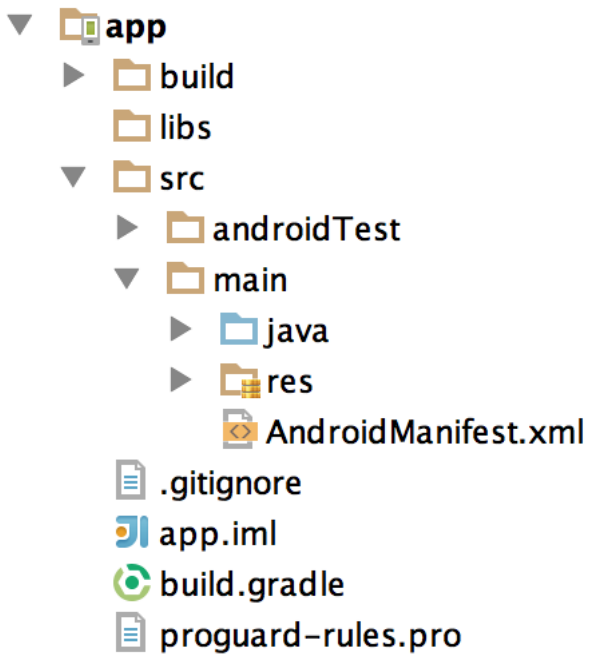


Vous avez également le ficher « **MainActivity.java** » qui correspond au code java de l'application



Pour notre application, nous avons donc un projet **HelloAndroid** contenant un module **app.**

Les modules ont la forme suivante :



On y trouve le dossier **res/** et le fichier **AndroidManifest.xml**, les sources (fichiers .java) ont maintenant été déplacées dans un dossier **java/**.

Le fichier build.gradle sert de configuration pour le nouveau moteur de production nommé Gradle, qui sera utilisé pour construire notre application afin de la déployer sur notre smartphone ou sur le Play Store.

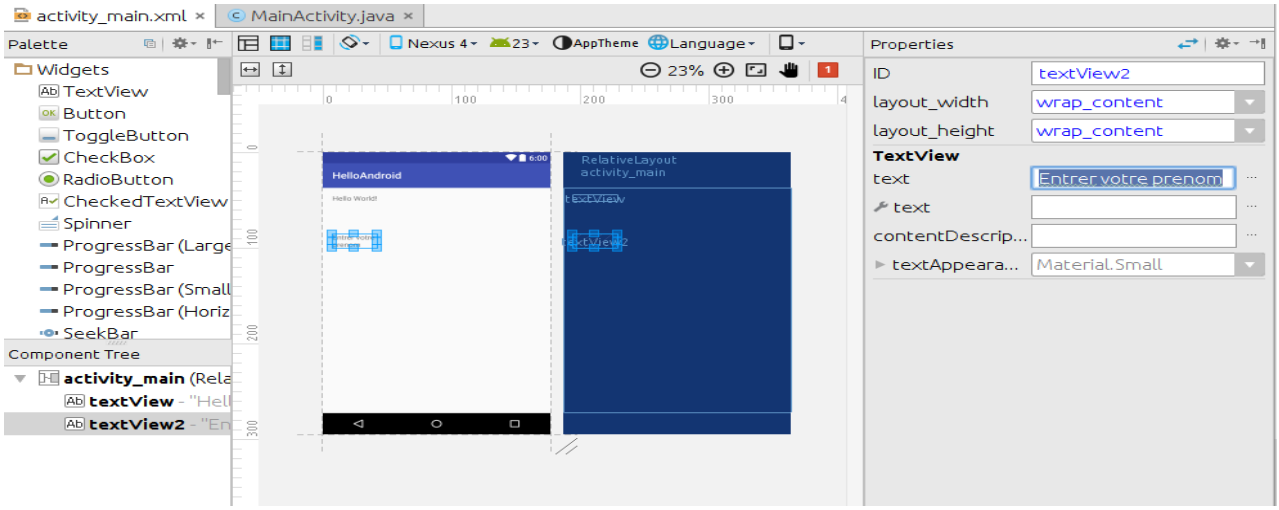
Nous allons donc dans un premier temps modifier le « design » de notre appication pour obtenir l'écran suivant :

Interface

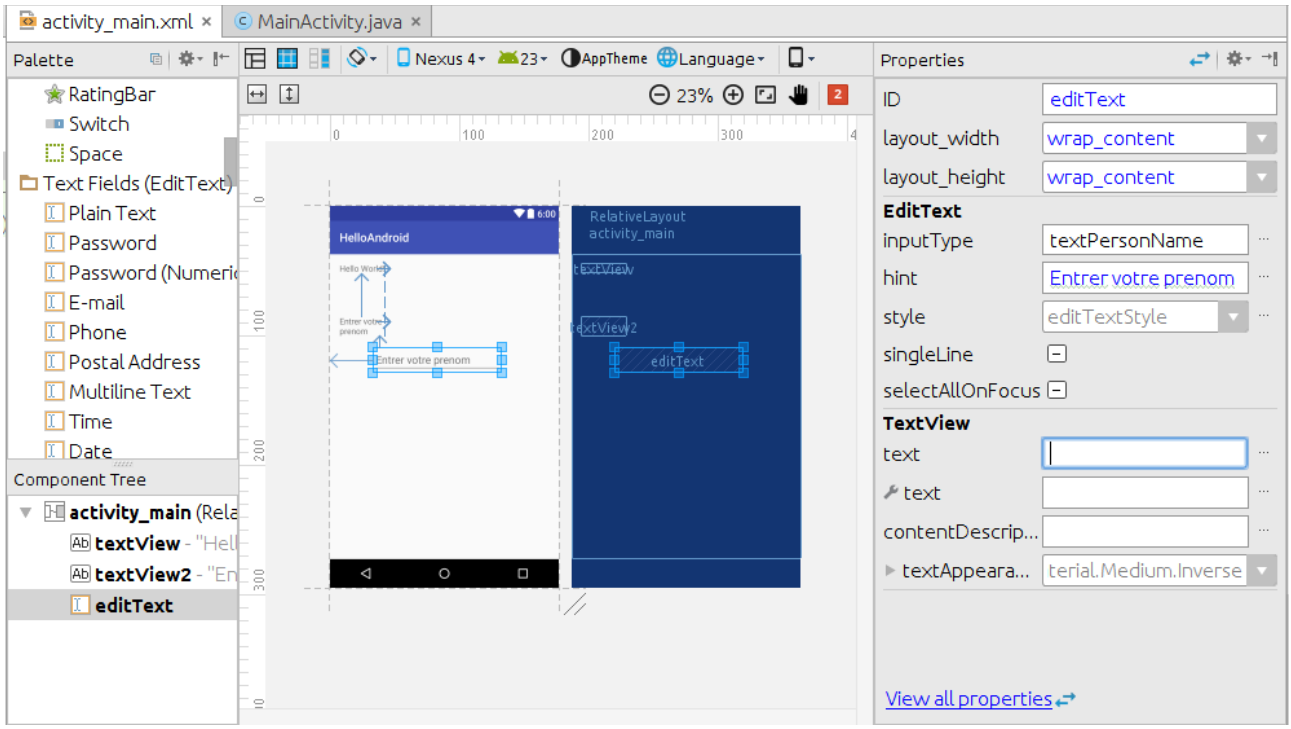
Maintenant nous allons créer l'interface de notre application . Pour créer cette interface vous pouvez utilisez l'outil graphique fournit avec AndroidStudio ou intervenir directement dans le fichier **activity_main.xml (ce qui est beaucoup plus délicat)**.

Par exemple, pour placer le TextView « Entrer votre prénom », vous prenez le composant dans la palette et vous le glisser vers l'AVD. Une fois le composant placé, vous avez sur la partie droite les éléments de configuration pour changer le texte par exemple, ou son ID qui représente son identifiant que l'on utilisera dans le code Java. Vous devez absolument visionner les différentes vidéos avant de commencer :

https://developer.android.com/training/constraint-layout/index.html



Répéter l'opération pour le TextField (choisir PlainText – et remplir le champ hint : « entrer votre prénom ») et le bouton.



Remarque : le positionnement des éléments est un peu délicat et la prise en main des différents « Layout » nécessite beaucoup de pratique, pour vous aider, vous pouvez consulter :

- http://developer.android.com/guide/topics/ui/declaring-layout.html
- http://www.tutos-android.com/tag/layout-android
- http://www.learn-android.com/2010/01/05/android-layout-tutorial/

Mais dans un premier temps, il est préférable de se concentrer sur la compréhension des concepts plutôt que sur la présentation de l'interface.

Les listeners

Actuellement, si nous lançons l'application et que l'on clique sur le bouton : rien ne se passera. Pour régler ceci, nous allons rajouter du code pour définir l'action et c'est là que les « listeners » rentrent en jeu.

Nous allons tout d'abord rajouter une interface à notre activité de type OnClickListener (mot clé implements en java, voir cours sur les interfaces).

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener
```

Puis normalement, AndroidStudio vous propose de rajouter les méthodes non implémentées dans notre classe. Dans le cas contraire placer vous sur `View.OnClickListener puis utiliser la combinaison de touches:alt + Enter`

```
@Override
  public void onClick(View view) {
  }
```

Une activité dispose d'une méthode findViewById pour appeler un élément de la vue, avec en paramètre l'identifiant de l'objet que l'on souhaite appeler. Celui-ci est stocké dans le ResourcesManager qui se nomme R.

```
        //récupération de l'EditText grâce à son ID
         editText = (EditText) findViewById(R.id.editText);

       //récupération du bouton grâce à son ID
       button = (Button) findViewById(R.id.button) ;

       // on fixe un listener sur le bouton en spécifiant que le le listener est
       //notre classe
       buttton.setOnClickListener(this) ;
```

Ainsi à chaque fois que l'on cliquera sur le bouton, la méthode onClick(View view) sera appelée.

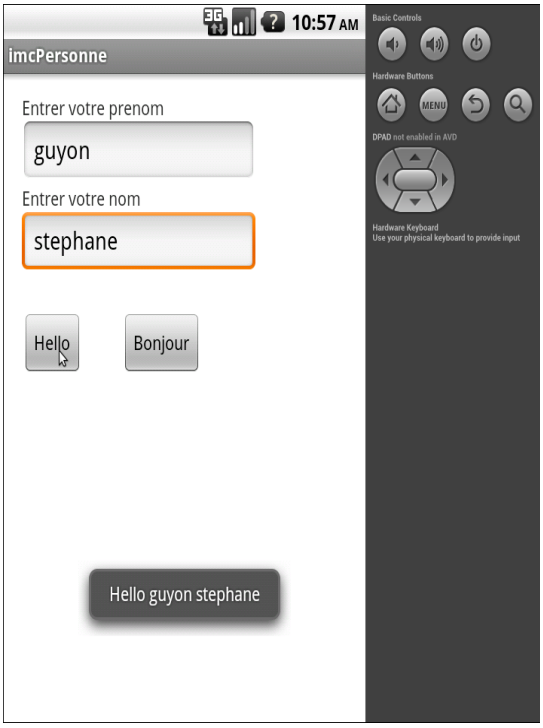
Nous allons rajouter un petit message à afficher lorsque l'on clique sur le bouton, les messages s'affichent via une classe nommée Toast. La classe Toast dispose d'une méthode statique (pas besoin d'instancier l'objet pour utiliser ses méthodes) de type makeText, qui prend en paramètres le contexte qui se trouvera être notre application, puis un message et pour finir une durée.
La durée peut être Toast.LENGTH_SHORT ou encore Toast.LENGTH_LONG.

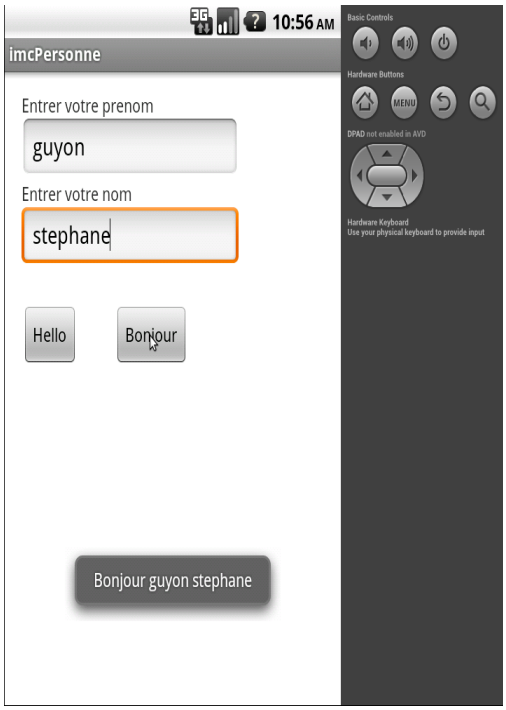
le programme devient :

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener{


    private EditText editText;
    private Button button;
    private String prenom;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        editText = (EditText) findViewById(R.id.editText);
        button = (Button) findViewById(R.id.button);
        button.setOnClickListener(this);
    }
    @Override
    public void onClick(View v) {
        prenom = editText.getText().toString();

        if(v == button){
            Toast.makeText(MainActivity.this,"Hello " + prenom + " !",
                    Toast.LENGTH_LONG).show();
        }
     }
}
```

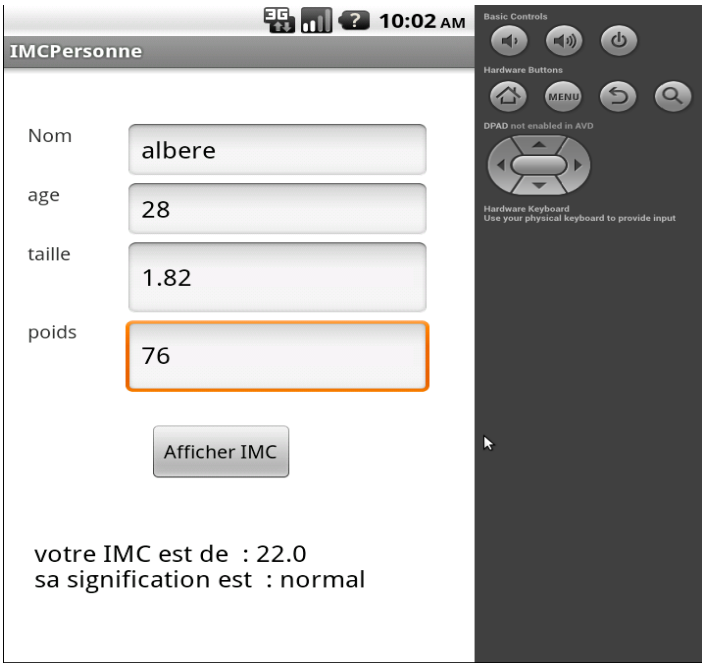
Travail à faire :

1 – Modifier votre interface, pour obtenir le résultat suivant :





2 – Créer un nouveau projet Android. Importer dans ce projet la classe Personne permettant de calculer l'IMC d'une personne et de donner sa signification. Ensuite réaliser l'interface suivante :



Remarque :

```
//récupération de l'EditText grâce à son ID
        editText = (EditText) findViewById(R.id.editText);
```

Ensuite le contenu de cet **ediText** ne peut être récupéré que sous le type String :

```
age = editText.getText().toString();
```

Donc si vous souhaitez un type int ou double, vous devez utiliser les méthodes appropriées :

```
      int ageInt = Integer.parseInt(age);
```

En une seule instruction :

```
      int ageInt = Integer.parseInt(editText.getText().toString());
```

3 – Créer un nouveau projet Android. Importer dans ce projet la classe Location permettant de calculer le prix d'une location de voiture. Ensuite réaliser l'interface suivante :

Par mesure de simplification, on considère que pour une location 100 km/jour sont offerts.

Les radios boutons :
Prendre le widget **container : RadioGroup** et glisser dans ce container les **radios boutons**

Dans la méthode onCreate

```
 //récupération du RadioGroup grâce à son ID
 radioCategorie = (RadioGroup) findViewById(R.id.radioGroup1);
```

Dans la méthode onClick :

```
// on identifie le radio bouton selectionné
int selectedId = radioCategorie.getCheckedRadioButtonId();

// à partir de l'id selectionné (selectedId) on récupère le radio bouton
RadioButton radioTypeCategorie = (RadioButton) findViewById(selectedId);

//Ensuite on teste le texte du radio bouton pour définir la catégorie
if (radioTypeCategorie.getText().equals("Economique") )
      categorie = 'E';
else if(radioTypeCategorie.getText().equals("Confort"))
      categorie = 'C';
else
      categorie = 'L';
```