

**TP N° 1****Exercice1****A)**

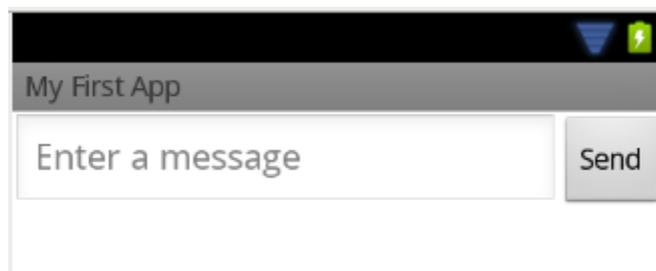
- 1- Qu'est-ce qu'un SDK et où se trouve le dossier correspond ?
- 2- Qu'est-ce qu'un AVD manger, qu'elle est le programme nécessaire pour son fonctionnement et comment peut-on l'installer.

B) Créer votre première application qui affiche simplement un message de bienvenue "Hello World" et exécutez-la dans un émulateur (ou sur votre dispositif mobile).

1. A quoi sert le fichier AndroidManifest.xml?
2. Quelle est la classe java principale de votre application et de quelle classe hérite-t-elle? Qu'elle est la méthode principale à implémenter?
3. Dans quel fichier est décrite l'apparence graphique de votre activité principale et quel est son type?
4. A quoi sert le fichier string.xml contenu dans le dossier <values >?

**Exercice 2**

1. Créer l'application dont l'interface est illustrée dans la figure suivante :



2. On souhaite modifier l'application pour que le bouton Send réagisse au clic en affichant le message dans une fenêtre popup appelée Toast Pour cela, il faut :
  - a. Ajouter l'attribut onClick au bouton dans le fichier activity\_main.xml et lui associer le nom de La méthode qui sera appelée, par exemple displayMessage qui n'est pas encore définie :android:onClick="displayMessage"
  - b. Définir la méthode displayMessage suivante dans la classe MainActivity et presser Ctrl+Shift+O pour importer les classes manquantes :

```
/**  
 * Called when the user clicks the send button  
 */  
public void displayMessage(View view){
```

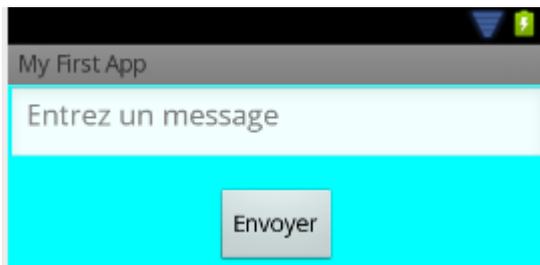
```

Context context = getApplicationContext();
EditText editText = (EditText)
findViewById(R.id.edit_message);
String message = editText.getText().toString();
CharSequence text = "Votre message est :\n" + message;
int duration = Toast.LENGTH_SHORT;
Toast toast = Toast.makeText(context, text, duration);
toast.show();
}

```

3. Que fait la méthode `findViewById` et quel paramètre attend-elle? A quoi correspond la classe java `R.java` contenue dans le dossier `gen` ?

### Exercice 3 –Personnalisation



Les constantes (chaînes de caractères, couleur, dimension,) sont définies dans le fichier `string.xml` contenu dans le dossier `values` et ensuite associées aux éléments graphiques dans le fichier `activity_main.xml`.

Modifier les éléments suivants de votre première application pour obtenir le résultat de la figure ci-dessus.

- Le message par défaut de `EditText` devient "Entrez un message"
- Les éléments sont placés verticalement dans le layout.
- Le message est positionné en haut de la vue (attribut `android:gravity`).
- Pour associer une couleur avec un élément graphique, il faut d'abord définir la couleur bleu (élément `color`) dans `string.xml` et ensuite utiliser l'attribut `android:background`. Modifier le fond du layout (attribut `android:background`) pour qu'il soit bleu (`#00FFFF`).
- Modifier la vue `EditText` pour que le message soit placé en haut, avec une marge de 20px.
- Pour définir une marge, il faut d'abord créer une dimension dans `string.xml`. Modifier la vue `Button` pour que le bouton ait une marge (top) de 20px.
- Centrer le bouton (attribut `android:layout_gravity`).